CARNEGIE MELLON UNIVERSITY

School of Architecture

College of Fine Arts

A COLLABORATIVE FRAMEWORK FOR MACHINE LEARNING-BASED TOOLMAKING FOR CREATIVE PRACTICES

A Dissertation in

Computational Design

By

Ardavan Bidgoli

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

September 2022

CARNEGIE MELLON UNIVERSITY

School of Architecture

College of Fine Arts

Thesis

Submitted in Partial Fulfillment of the requirements for the degree of

Doctor of Philosophy

TITLE:

A Collaborative Framework for
Machine Learning-Based Toolmaking for Creative Practices

AUTHOR:

Ardavan Bidgoli

ACCEPTED BY ADVISORY COMMITTEE:

Daniel Cardoso Llach	Principal Advisor	Sep 27, 2022 DATE
Eunsu Kang	Advisor	Sep 27, 2022 DATE
Golan Levin	Advisor	9/27/2022 DATE
Barnabás Póczos	Advisor	Sep 27, 2022 DATE

Abstract

The latest boom of Machine Learning (ML) in the early-2010s has raised a new wave of interest among creative practitioners to explore the intersection of Art and Artificial Intelligence (AI), specifically Generative Machine Learning. A growing number of artists, designers, and architects appropriated these algorithms to make new tools for their creative practices.

This dissertation introduces and documents a collaborative framework to make machine learning-based tools for creative practices. The framework embraces the idiosyncratic nuances and elements of the physical context of the creative practice. It takes a new point of view on data and data curation as the primary method of interacting with ML algorithms. The framework achieves this goal by utilizing small user-generated datasets, which are biased toward the creative practitioners' personal preferences, subjective measures, and elements of the physical context of their practice. Through collaboration with machine learning expert toolmakers, the framework makes ML algorithms more accessible to these creative practitioners. It highlights the affordances of ML algorithms, specifically Conditional Variational AutoEncoders (C-VAE), that can be efficiently trained and overfit on small datasets to produce outcomes that are closely tied to the creative practitioners and their context.

In the two case studies, the framework serves as a high-level blueprint to develop bespoke tools that support various stages of machine learning-based toolmaking for creative practitioners. In SecondHand, I collaborated with a group of participants to develop handwriting typeface generation tool. A dashboard, based on Dash Plotly, featuring interactive data visualization and data curation tools, was developed for this study. In ThirdHand, I collaborated with a musician to create a robotic tool to play santur, a traditional Persian musical instrument, using an ABB IRB 120 robotic arm and a real santur.

The case studies demonstrated that the proposed collaborative framework meaningfully brings ML experts' technical literacy to complement creative practitioners' domain knowledge and skills to overcome the technical ML challenges, and help integrate various idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process.

In Memory of My Father, Ghasem Ali Bidgoli

1949-2020

Acknowledgments

It was a delightful September morning in 2003, I walked through the iconic gates of the University of Tehran to start my journey as a freshman at the School of Fine Arts. Young and full of dreams, I didn't know the journey would take 19 years. Yet, in another September day, on the other side of the world, I finally finished the journey. I was a lucky guy who found wise mentors, supporting friends, smart colleagues, and of course, the love of my life, in these years. I will always be grateful for this opportunity.

Here, on the verge of the finish line, I would like to take the opportunity to express my gratitude to the people who supported me throughout my years at CMU.

First, I would like to thank Daniel, my advisor, and the chair of the committee. I had the privilege to work with him at the Stuckeman Center for Design Computing at Penn State and later at CMU. I still remember our first Skype call, when I was still in Iran, struggling with serious challenges to come to the US. His generous support has always been abundant throughout these years. His rigorous academic standards taught me the meaning of being a scholar.

I would like to express my gratitude to my committee members: Eunsu Kang, Golan Levin, and Barnabás Póczos. Your insight, wisdom, and constructive criticism helped me shape my research and added to my scholarship.

My years at CMU were brighter because of my friends, colleagues, and faculty members who were always there when I needed them. I would like to thank my friends, fellow Ph.D. students in room 402; Pedro Veloso, Jinmo Rhee, Manuel Rodriguez Ladron De Guevara, Emek Erdolu, and friends at CodeLab, especially, Atefeh Mahdavi and Ozguc Bertug. It was a privilege to work with you and to learn from you. I would like to express my gratitude towards Omar Khan, Azadeh Sawyer, Josh Bard, and Madeline Gannon for their support throughout these years.

This research owes to all the people who helped me with the two case studies. First, I would like to thank my students from *Introduction into Machine Learning and Design* who played a pivotal role in my research. I would like to thank them for their contribution.

Moreover, I would like to thank Mahtab Nadalian for patiently bearing with me through ThirdHand. I cannot overemphasize her contribution to this research. It was such a pleasing surprise to befriend Mahtab, Amir, and Kija. The early ideas of ThirdHand were nourished by several brainstorming sessions that I had with Ebrahim Poustinchi. His knowledge of Persian music, robotics, and creative computing encouraged me to lay out this complex study.

Before joining CMU, I used to call Penn State home. It was the place I met some of my most precious friends. I would like to express my gratitude towards Mehrdad Hadighi, the head of the Department of Architecture at the Stuckeman School of Architecture and Landscape Architecture. Loukas Kalisperis has always been a trusted mentor, and at the very same time, a compassionate friend for me. Jose Pinto Duarte, who has always been there to support me and Vina in all these years. Finally, I would like to thank all my

friends and colleagues at SCDC, you helped me become what I am today; Shokufeh Darbari, Ali Ghazvinian, Paniz Farrokhsiar, and Shiva Puntanthanbaker.

I would like to thank other scholars and researchers that I had the opportunity to work with and learn from. I want to thank Rebecca Fiebrink, her research on creative computing and AI has been an inspiration for me since the first time I attended her workshop at the Studio for Creative Inquiry. It was also my great pleasure to work with Jose Luis Garcia del Castillo Lopez, he never hesitated to offer his help when I needed it and taught me the value of synergy and collaboration in the research community. And finally, my friend, Mohammad Keshavarzi, whose bright mind never ceases to bring new ideas, energy, and passion to push the boundaries.

I would like to thank my managers at Bentley Systems and Autodesk; Volker Mueller, Evan Atherton, and Matt Jezyk. They opened a new door for me to learn how computational design research can find its audiences beyond the academic environment.

Finally, I would like to thank my family, who stood beside me throughout all these years. Although they could never visit me here in the United States, their unconditional love and support have always been with me. My mom, the one who always motivated me with his never-ending passion for life and joyfulness. Thank you *maman* for standing for me and helping me live the life that I wanted. And my father, *baba*. He was the one who inspired me to pursue my future career in architecture. I learned from him what I couldn't find in any classroom. Every morning that we spend on construction sites and every afternoon in his *daftar*, was a master class for me. I wish you were here on this day. We lost him in the dark winter of 2020, but his memory will always brighten my days. I dedicate this thesis to his creative soul.

And at last, who can endure such a long journey without a trusted friend who is ready to offer everlasting love on every turn of the road? For me, it was Vina, my love of life, beacon of hope, and as she puts it, "my partner in crime." Congratulation *bamshee*, we finally made it together. It is time to open a new chapter in our life!

Glossary

Tool: An apparatus or instrument required in the practice of activity or profession (Merriam-Webster n.d.). In the context of this research, the word *tool* is used as a broad term referring to computational tools used by creative practitioners.

Toolmaking: The process of studying, designing, and prototyping a tool for a specific activity.

Toolmaker and ML Expert Toolmaker: In this thesis, the term *machine learning expert toolmaker* refers to an individual or group of people with experience in 1) computational design, computational toolmaking and 2) experience in various fields of machine learning, from theory to design and implementation of an end-to-end ML pipeline. The ML expert toolmaker collaborates with creative practitioners to design, implement, and refine the technical aspects of the meta-tool. Throughout this document, I use *toolmaker* and *ML expert toolmaker* interchangeably.

Creative Practitioner: Individuals or groups of people who create ideas or artifacts in various fields, including, but not limited to, fine arts, music, composition, performance, design, architecture, or any other domain where creative engagement is a key element of the practice.[1]

Creative Computing: refers to the inquiries into the intersection of creativity and computing by using computers to make novel creative work. It "seeks to reconcile the objective precision of computer systems (mathesis) with the subjective ambiguity of human creativity (aesthesis)" (Hugill and Yang 2013, 5).

***The Framework*:** In the context of this study, the *framework* refers to a high-level guideline for ML-based toolmaking for creative practitioners, casting it as a collaboration between creative practitioners and ML-expert toolmakers. This framework focuses on integrating the idiosyncratic nuances and elements of the physical context of creative practices into the toolmaking process and serves as a blueprint for developing the meta-tools and defining collaborative toolmaking workflow.

---

[1] The definition of creative practitioner is borrowed from Rebecca Fiebrink's, a pioneer of Art and ML and senior lecturer at the Department of Computing at Goldsmiths University. She defines creative practitioners as "people creating ideas or artifacts in a broad set of domains. They include creators in the fine arts, music composition and performance, and theater and performance art, as well as creators of new indie games and "makers" of other hard-to-pigeonhole artifacts and experiences." She also defines creative domains as domains in which creative expression is a primary goal (Fiebrink 2019).

***Meta-tool*:** The meta-tool—a term derived from Rebecca Fiebrink's Meta-instrument—refers to the tool used to make another tool (2009; 2017), in this case, ML-based tool for creative practitioners. The meta-tool provides a software/hardware platform for the ML expert toolmaker and the creative practitioner to collaborate on the tool development. It enables them to collect and process data, train the machine learning model, and fine-tune the results.

***Machine Learning*:** A subfield of artificial intelligence which studies algorithms that can complete a given job by observing a series of solution samples rather than explicitly programming the answer. [2] ML algorithms rely on statistical methods to gradually improve their performance through an iterative cycle of experiences. In this sense, experience may refer to observing an unlabeled data set (unsupervised learning), a labeled data set (supervised learning), a series of simulations (reinforcement learning), or a series of demonstrations (learning from demonstration).

***Generative Machine Learning Models*:** In machine learning, a generative model is a model that can be trained on an unlabeled subset of a distribution p_{data} and learns an estimated representation of that distribution, p_{model}. By drawing samples from this distribution, users can generate new instances that closely resemble those in the training set. In this regard, generative models differ from discriminative models, which map features to labels and have been widely used for tasks such as image classification.[3]

Latent Space (in Generative Models): In generative models, latent space or latent feature space, or embedding space, refers to the machine learning model's internal representation of the data set. Embedded in the hidden layers of the model, it represents the encoded learned features of the training data. For example, in a Variational Autoencoder, the latent space is the bottleneck layer. Navigating the latent space refers to the act of replacing the latent vector with different values to explore the outputs of the generative model.

Interactive Machine Learning: Interactive machine learning refers to "algorithms that can interact with agents and can optimize their learning behavior through these interactions where the agents can also be human" (Holzinger 2016, 119). In iML, the training process is cast as human-computer interaction (HCI) (Dudley and Kristensson 2018), and the computer is a part of the human design process rather than the human being in the loop of an algorithmic process (Gillies et al. 2016). The user can iteratively add new learning samples to steer the learning direction until the desired outcome is achieved.

---

[2] Tom Mitchel defines learning algorithms as "[a] computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E." (Mitchell 1997).

[3] There are other definitions of generative modeling, i.e., Doersch defines it as "… a broad area of machine learning which deals with models of distribution $P(X)$, defined over datapoints X in some potentially high-dimensional space X. Some researchers refer to the generative aspects of machine learning models as the unconventional application of ML. For example, Fiebrink describes unconventional applications, as the use of generative models to "produce new content that is "similar" to the training set" (Fiebrink 2019, 3). In contrast, she lists applications such as processing, reasoning, prediction, or classification of data as examples of conventional applications of ML. This research avoids this terminology, to prevent any confusion in the future.

Table of Contents

Table of Figures

Table of Tables

Chapter 1. Introduction

1.1 ML-Based Tools for Creative Practices[1]

The latest boom of Machine Learning (ML) in the early 2010s has raised a new wave of interest among creative practitioners to explore the intersection of Art and Artificial Intelligence (AI), specifically Generative Machine Learning. A growing number of creative practitioners, such as artists, designers, and architects, appropriated these machine learning algorithms to make ML-based tools to support their creative practices. They tap on off-the-shelf online data sets, ready-to-use ML models, and powerful hardware to tame ML algorithms and create a variety of tools. These factors allowed creative practitioners to inquire into the application of ML in writing, visual arts, painting, music composition, game design, choreography, interactive installation, and architecture to name a few. They demonstrated that generative ML models can open new opportunities to make a new generation of tools for creative practices.[2]

To fathom the unique affordance of generative machine learning algorithms, it is enlightening to compare them with conventional computational toolmaking that are based on the conception of knowledge as an object that can be acquired from the practitioner, embedded in algorithms, and transferred from one machine into another.[3] Toolmaking procedures that are built on this conception require toolmakers to encode creative practitioners' skill, knowledge, and creative process in algorithms.[4] However, this is a

---

[1] This document uses the following stylistic guidelines:

- The first-person singular form is used to express the author's point of view and personal remarks.
- The second-person plural form is used to reflect on the collaborative work with the creative practitioner(s).
- The third-person plural form is used as the third-person singular gender-neutral form when referring to a generic person whose gender is unknown or irrelevant to the context.

[2] It is worth mentioning that a considerable body of literature in ML-based tools for creative practices use discriminative models—i.e., classifying data and associating each data sample to a specific label. For instance, Fiebrink's bow-gesture classifier (Fiebrink 2011) can observe a high-dimensional vector of input data from sensors—accelerometers and pressure sensors—to map it into a one-dimensional vector that defines the gesture class.

[3] I thoroughly address the conception of skill as object and attainable Appendix II: The Context.

[4] To delve deeper into the challenges of conventional computational toolmaking for creative practitioners, one can study parametric modeling, which is quite popular among architects and computational designers. In 2007, Rick Smith, who played a crucial role in developing Digital Project software at Gehry Technology in the 1990s and 2000s and worked on parametric modeling for decades (Davis 2013), wrote a technical note on using parametric modeling software. He lists several issues, most notably: 1) necessity of making many decisions upfront, or as he calls it upfronting, 2) necessity of anticipating future design changes, 3) incapability of dealing with major design changes,

challenging expectation as creative practitioners usually do not think algorithmically. They work on wicked problems which can be unique, ill-defined, and influenced by a vast array of parameters while having no definitive formulation, clear stopping rule, or ultimate test (Buchanan 1992). Quantifying success metrics for such wicked problems is a complex task. Even where there are established quantifiable measures of success, i.e., in architectural design, there is still a wide range of subjective and qualitative aspects corresponding to the creative elements of the practice that cannot be easily evaluated, quantified, and integrated into an algorithm.

To overcome these challenges, computational toolmakers rely on abstraction, eliminating factors in favor of simplicity and reducing the computational complexity of their tools. While abstraction is a key concept in AI and a powerful tool that makes representing real-life problems in computers' world possible, it is a double-edged sword. Excessive abstraction can eliminate the personal and subjective aspects of the practice, discard elements of the physical context, and "…factor out all aspects of perception and motor skills" (Brooks 1991b, 142). This can result in decontextualization, where the resulted tool barely represents the idiosyncratic and nuances of the practice or elements of its physical context.

Another critical issue with conventional computational toolmaking approaches comes from their conception of knowledge and skill. The idea of encoding skill and knowledge of a practitioner into an algorithm, tool, or machine stems from the conception of knowledge as an object that can be extracted from people to be transferred and embedded in machines.[5] However, toolmaking is beyond a mere technical endeavor; it is a critical step in a creative process that should be informed by personal preferences, subjective measures, elements of the physical context, users' conception of the activity, and the proposed solution. Simultaneously, toolmaking informs the creative practitioners' conception of the activity and affects the solution. This dynamic relationship is a crucial element in a creative process that should not be interrupted.[6]

The promising proposition of machine learning algorithms to address these issues resides in their ability to automatically find the patterns by learning from a meticulously curated dataset and use them to generate novel samples that did not exist in the dataset. I found this a unique opportunity for creative practitioners as it liberates them from encoding their knowledge, skill, and creative processes into abstract algorithms. Instead, they can use ML algorithms, as their machinic surrogates while interfacing with them through what they are mastered in, providing instances of their practice in the context of their practice.[7]

However, some challenges hinder creative practitioners' ability to adopt machine learning algorithms in their tools. First, designing, implementing, training, and optimizing ML models is a technically complex

---

4) challenge of foreseeing the effects of changes in one part of the model on the other parts, and 5) limitations on sharing and usability due to over-complicated models than cannot be comprehended by anybody other than the original designers (Smith 2007 cited in Davis 2013).

[5] In Appendix II: The Context, I argue how the social, technical, political, and economic turmoils of the past two centuries contributed to the conception of skill as an object or commodity that can be captured, stored, and transferred, rather than being situated in its social and physical context.

[6] The discussions on the dynamic relationship between humans and their tools are gaining more attention in the shadow of the recent AI advancement. Iyad Rahwan, the director of the Center for Humans and Machines at the Max Planck Institute for Human Development, argues that humans and machines can inform, mold, and alter each other's behavior in various ways. Humans change the machines' behavior by engineering algorithms, actively providing training data, or being observed by data collection machines. On the other hand, machines affect our behavior, social fabric, and the political landscape through their omnipresent role in our routine decision-making procedures (Rahwan et al. 2019, 483).

[7] I discussed the notion of machinic surrogates in (Bidgoli, Kang, and Cardoso Llach 2019).

process. The lion's share of current ML-based toolmaking efforts for creative practitioners is developed based on frameworks primarily aimed at niche audiences, such as researchers and developers with prior exposure to machine learning (Roberts, Hawthorne, and Simon 2018). While creative practitioners are experts in their field of work, extensive knowledge of machine learning is a rarity among them. This lack of ML knowledge forces creative practitioners toward safe options, i.e., off-the-shelf datasets, pre-trained models, or black-box toolkits which push them toward deeper abstractions. A consequence of this issue is a detachment between these ML-based toolmaking and the idiosyncratic aspects, elements of the physical context, and nuances of creative practice, which is a reminiscence of the decontextualization that I discussed above.

Breaking this technical barrier and going beyond these safe options will increase the odds of encountering technical problems that novice ML users cannot easily address. In Chapter 2, I will discuss how this issue reflects itself in the current state of ML-based toolmaking for creative practices, where contextual factors are generally neglected in favor of making ML-based tools more accessible. This missing context breaks the dynamic relationships between creative practitioners and their tools.[8]

To address these challenges—technical barriers and missing context—I propose, implement, and document a collaborative framework to develop ML-based tools for creative practices. In this framework, the idiosyncratic aspects, and elements of the physical context of creative practices are taken into account in ways that generic tools or frameworks have so far neglected to do. Through close collaboration with ML expert toolmakers, this framework makes ML algorithms more accessible to creative practitioners without requiring them to directly engage with the complexities of the backend ML algorithms.

The framework follows four principles:[9]

1. Working with and from the context, the framework is built around the conception of skill and knowledge as situated in the context of its practice. It contrasts with conceptions of skill as an object, or as a commodity which has been prevalently used in the realm of AI and ML.

2. Meaningful extended collaboration between creative practitioners and toolmakers, the framework makes ML algorithms more accessible to creative practitioners without requiring them to engage with the technical complexities of ML algorithms.

3. Using data to interface with the machine learning algorithms, extending the prior efforts in interactive machine learning for toolmaking for creative practices, most notably (Fiebrink 2011), this framework utilizes real-time data visualizations, interactive data curation tools, fast-training machine learning algorithms, and interactive sampling tools to allow the creative practitioners to control the learning process. This approach to data offers a counterpoint to the currently dominant approach in creative machine learning, where 1) data is usually rigid, pre-determined, and externally sourced, and 2) the primary method of controlling the learning model is coding.

---

[8] This issue is rooted in a broader gap in the ML literature. A majority of efforts in the machine learning research community are focused on developing novel algorithms and improving efficiency (Simard et al. 2017), leaving fewer resources to inquire into integrating inputs from the user and the context into the process.

[9] In Chapter 3, The Framework, I will elaborate on these four principles and two case studies, where I collaborate with creative practitioners to make tools based on this framework, are documented in Chapter 4, The SecondHand and Chapter 5, The ThirdHandChapter 4.

4. Using generative machine learning models and taking advantage of overfitting on small datasets, the framework encourages using user-curated datasets, which are inevitably small, biased, and skewed toward those specific users. It suggests using machine learning algorithms that can be efficiently and quickly trained on a limited set of data that the creative practitioner can feasibly provide.

1.2 Research Question

The two primary questions of this research are:

- How do interfaces for data generation and curation for generative machine learning offer new pathways for toolmaking for creative practitioners?
- How can a collaborative approach mitigate the lack of technical machine learning experience among creative practitioners and help them to integrate the idiosyncratic aspects, elements of the physical context, and nuances of their creative practice in the toolmaking process?

1.3 Hypothesis

I hypothesize that interfaces for data generation that emphasize user-generated data to integrate elements of the physical context and users' subjective preferences can reveal new potentials of generative machine learning to support creative practices.

I hypothesize that a collaborative approach to developing ML-based tools for creative practices can meaningfully bring ML experts' technical literacy to complement creative practitioners' domain knowledge and skills, overcome the technical ML challenges, and help integrate various idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process.

1.4 Methodology Overview

This thesis documents two case studies focused on the collaborative development of ML-based tools for creative practices. In the first study, the *SecondHand*, a group of students with an introductory knowledge of machine learning developed handwriting typeface generation tools based on their handwriting samples. The toolmaking process empowers them to interface the machine learning model not through code, but mostly through providing samples, curating the training dataset, and in some cases, playing with a handful of training parameters.

In the second study, the *ThirdHand*, I collaborated with a musician to create a robotic tool to play santur, a traditional Persian musical instrument. The musician has no computer science, programming, or machine learning background. The study was a collaboration between me, the machine learning expert toolmaker, and them, the creative practitioner, where they provided data samples, evaluated the results, and provided feedback in the process. Meanwhile, my role was to oversee, develop and maintain the technicalities of the toolmaking process.

The toolmaking process, dynamics between the toolmaker and creative practitioners, and the results were documented through digital field notes, video recordings, written reports, and unstructured interviews, collectively forming the basis for the discussions and conclusions on each case study.

1.4.1 Framework and Meta-Tool

The framework, as described earlier, is a high-level guideline to make ML-based tool for creative practices. The meta-tool—a term derived from Fiebrink's Meta-instrument (Fiebrink, Trueman, and Cook

2009)[10] —refers to an implementation of the framework designed and fine-tuned for a specific case study. It provides the software/hardware platforms for the ML expert toolmaker and the creative practitioner to collaborate on the tool development. The meta-tool allows them to:

- Embrace creative practitioners' subjective measures and personal aspects of their craft in the toolmaking process. This goal is achieved by:
    - Creating and curating bespoke datasets: The creative practitioner can generate data samples and curate hybrid datasets of user-generated samples. This process empowers them to introduce desired features to the learning model and steer the learning direction based on personal preferences and subjective assessments,
    - Enriching the evaluation loops: The creative practitioner can provide personal and subjective evaluation and feedback iteratively to supervise the toolmaking process,
- Integrating elements of physical context of the craft in the toolmaking process through:
    - Physical involvement: The creative practitioner can introduce material behaviors and tool affordances by creating data samples in close-to-real-life demonstrations.

1.5 Motivations and Importance

During my Ph.D. studies, I have been studying the new opportunities that machine learning brings to creative practitioners, specifically by introducing new means of toolmaking. This interest is the primary motivation behind the major themes of this research:

Examining how machine learning algorithms can help creative practitioners play a central role in the toolmaking process. I am curious to study how ML can help addressing the shortcomings of current ML efforts in creative fields, most notably decontextualization.[11]

Making ML algorithms more accessible to creative practitioners. I am interested in examining a collaborative framework that brings ML experts' technical literacy to complement creative practitioners' domain knowledge and skills, which can potentially render ML-based toolmaking more accessible.

Exploring new forms of dynamic collaboration between creative practitioners and ML experts, who otherwise work separately from each other.[12] I have been curious to explore the unexpected interactions between the ML-expert toolmaker and creative practitioner using the proposed toolmaking framework. Across this research, I seek to design and create opportunities for such dynamic interactions and reflect on them, i.e., the unexpected event of an outlier sample in the training data set, a surprising, generated sample, or a never-seen-before design space. These unexpected incidents require the toolmaker's intervention and help them iteratively refine the process and build a personal perception of the tool and its properties.

Exploring new forms in which toolmakers interface with their ML-based toolmaking system. I look forward to testing how data can serve as the main means of interaction between the creative practitioners

[10] Fiebrink describes meta-instrument as "… an instrument for creating instruments" (Fiebrink 2017, 11). In this study, I opted for meta-tool as I use it not only in the realm of music, but in the broader realm of creative practices.
[11] I will discuss these shortcomings in more details in Chapter 2, ML-Based Toolmaking for Creative Practitioners.
[12] While not being a primary contribution of this dissertation, this framework can potentially help ML experts engage in creative activities using their custom-made tools.

and the ML model.[13] This notion of data offers a counterpoint to the currently dominant approach in creative machine learning, where 1) data is usually rigid, pre-determined, and externally sourced, and 2) the primary method of controlling the learning model is coding. I investigate how trusting the creative practitioners to thoughtfully decide on the inclusion of contextual, subjective, and personal inputs allow them to interface with the learning algorithm and introduce elements of the physical and personal context of practice in the toolmaking process. Moreover, documenting and analyzing the shift from code to data as the main method of interfacing with the ML algorithm and its effects on the creative practitioner, its tool, and the craft is another motivation for this study.

Contributing to the body of knowledge on using machine learning algorithms with small datasets for bespoke toolmaking. Machine learning research is generally associated with large-scale datasets that might be biased toward specific races, gender, geographic regions, or art genres. Generalizing the application of ML models trained on these data sets is a concerning matter and a source of debate. In contrast, this research explores the positive side of biased datasets and overfitting a machine learning model on them. I am interested in exploring the generative potential of bias that resides in each creative practitioner's judgments and subjective metrics. I work with very small user-generated datasets, which are inevitably biased and skewed toward those specific users. The ML models trained on these datasets are prone to overfitting, and their outcomes are not generalizable, as they closely tied to the user and its context.[14]

Finally, I envision this research to be an opportunity for creative practitioners to meaningfully get involved in the toolmaking process. For them, the benefit of this approach is twofold, on the one hand, it allows them to introduce various aspects of their experience and knowledge to the toolmaking process and to make better tools. On the other hand, this allows them to gain a better understanding of their tools and find inspiration to explore new frontiers of creativity that wasn't in reach before. In that capacity, the primary aspiration for the creative practitioners will reside in the opportunity to create tools to help them explore new creative experiments.

1.6 Novelty

This research documents the development and evaluation of a novel framework for ML-based toolmaking that allows creative practitioners to collaborate with ML experts and introduce various idiosyncratic aspects, elements of the physical context, and nuances of creative practice. This research also explores novel approaches to make this process accessible for creative practitioners with limited technical literacy. As the body of literature in ML-based toolmaking for creative practices has been steadily growing in the past few years, other scholars have elaborated on some of the methods and approaches that I adopted in

---

[13] This notion of data is primarily inspired by the works of Rebecca Fiebrink on *Training Data as Interface* (2016) which is also elaborated and discussed in (Cardoso Llach 2017).

[14] Interestingly, this also helps defining ML-based tools with respect to the context and people—toolmakers and creative practitioners—rather than as stand-alone and autonomous agents. It is commonly believed that with the introduction of autonomous machines in manufacturing, these machines will take over certain jobs. At the same time, the current human labor force will shift to occupy newly created jobs, i.e., training, maintaining, and supervising these autonomous machines. However, most of the jobs that these machines will take over are the ones that require less training and experience. In most cases, workers in these jobs are the most vulnerable in the job market with the lowest wages. It is naïve to assume that these workers can smoothly prepare themselves for the new complex responsibilities of autonomous systems. These new jobs usually demand educated workers with longer training and more experience, which is a different demographic section than those who are losing their jobs.

this research. I acknowledge these prior efforts and build upon their contributions while distinguishing this research from the literature by:

- Centering the creative practitioner in the toolmaking process and embracing the idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process,
- Developing and evaluating a framework for dynamic collaboration between the creative practitioner and the ML expert toolmaker and qualitative documentation of this process using qualitative and quantitative metrics,
- Developing novel interfaces allowing creative practitioners to interactively collect, curate, and manipulate their user-generated data as a primary means of interacting with ML models,
- Developing a series of generative machine learning models (C-VAE), optimized to train rapidly on small datasets for bespoke toolmaking.
- Investigating the potentials of small user-generated datasets, which are inevitably biased and skewed toward those users, and documenting the generative potential of bias that resides in each creative practitioner's judgments and subjective metrics.
- Combining latent space exploration and data curation to control generative models' behavior,
- Developing a physical-to-digital-and-back-to-physical process around a generative ML model.

1.7 Deliverables

This research aims toward three final deliverables:

- First, a framework for collaborative machine learning-based toolmaking for creative practices,
- Second, two implementations of the framework:
    - The SecondHand meta-tool sets several key characteristics of a collaborative and interactive ML-based toolmaking workflow. Most notably, the real-time data visualizations and data curation tools, fast-training machine learning backend, and interactive sampling tools.
    - The ThirdHand meta-tool defines fundamental aspects of a ML-based toolmaking process that utilized two data modalities—sequence of motion and sound—including machine learning model, data pipeline for temporal six-degree-of-freedom data, the physical-to-digital-and-back-to-physical pipeline, and the robotic implementation.
- Third, thorough documentation and analysis of the two case studies, where the framework and derived meta-tools were put into practice. These two reports are guidelines for other researchers to explore the realm of ML-based toolmaking for creative practitioners.

1.8 Contribution

This research contributes to the field of ML-based toolmaking for creative practices in multiple ways:

- Defining a framework for creative practitioners to collaborate with ML expert toolmakers to integrate ML in their toolmaking process,
- Enabling creative practitioners to be at the center of the toolmaking process and establishing methods to introduce context to this procedure,
- Expanding on the body of knowledge on using machine learning algorithms with user-generated small datasets for bespoke toolmaking,
- Exploring the generative potential of bias in small user-generated datasets that reflect creative practitioners' judgments and subjective metrics.

1.9 Thesis Structure

This dissertation is organized into six chapters, including this introduction:

Chapter 2, ML-Based Toolmaking for Creative , is a review of state-of-the-art machine learning in the realm of toolmaking. This chapter is organized into three sections: first, I briefly introduce some of the creative practitioners who have explored making bespoke tools with deep learning algorithms in the late 2010s. This section aims to illustrate the landscape of independent efforts to work with AI/ML in creative practices. The chapter proceeds to discuss some of the technical barriers that stand in front of creative practitioners to harness ML algorithms in their tools. I investigate three major problems: lack of ML technical knowledge, data logistics, and evaluation challenges. Finally, I review the precedents in the literature which have considered context as their primary matter of study.

Chapter 3, The Framework, introduces the framework for designing ML-based tools for creative practitioners. This framework allows creative practitioners to build their tools, for their specific physical and personal contexts, without requiring them to engage with the complexities of the backend ML algorithms. It aims to make ML models more accessible to these creative practitioners through close collaboration with ML expert toolmakers. This framework serves as a high-level guide to design and implement meta-tools in the two case studies.

Chapter 4, The SecondHand, and Chapter 5, The ThirdHand, are comprehensive reports and documentation of two case studies which I conducted to test the hypothesis of this study. Each chapter encompasses its methodology section, followed by an in-depth discussion on the context, participants, research components, bespoke meta-tool implementation, and study progress. I conclude each chapter with a discussion.

The first case study, The SecondHand, is a vehicle to investigate the potentials of bespoke data collection methods, interactive data curating tools, and generative models in machine learning-based creative toolmaking. It addresses a remote collaboration between a group of students from 48-770: Inquires into Machine Learning and Design and me to create a handwriting typeface generator. Through this study, I examine and document two primary subjects: 1) how creative practitioners interact with the meta-tool for curating their own data sets and 2) how the meta-tool helps with the accessibility of ML-based toolmaking for creative practitioners.

The second case study, The ThirdHand, is a collaboration between a musician and me to develop a robotic musical instrument to play santur, a traditional Persian musical instrument. In addition to the subjects that I investigated in the first case study, this study delves deeper on 1) dynamic one-on-one collaboration between the creative practitioner and the toolmaker, and 2) working with data complexities, such as translating data from physical to digital and then reproduce them in physical world, and working with two different data modalities, one as six-degree-of-freedom motion sequence, and one as sound.

Chapter 6, Conclusion, provides an in-depth discussion of the study, a summary of contributions, unaddressed challenges, and future steps.

Appendix I: Conditional Variational AutoEncoders is a brief technical discussion on machine learning algorithms that are used in this study.

Appendix II: The Context, is a discussion on the social, economic, and historical aspects of skill, learning, and toolmaking. Throughout this appendix, I walk back in time and revisit various conceptions of skill, as a form of knowledge, and explain how the mere act of toolmaking has long been a matter of political,

social, and economic debate. This brief context is a prelude to introducing ML-based toolmaking that centers the creative practitioners and elements of their physical context.[15]

1.10 Delimitations

To set this research within the boundaries of a Ph.D. dissertation, the scope of the proposed framework is narrowed down to focus on these subjects:

The general domain of work: This study exclusively focuses on making tool for creative practices. It is not addressing computational creativity which advocates using computational methods to mimic human creative actions.

Autonomy: The proposed framework helps creative practitioners execute their plans— predefined or exploratory—by utilizing the generated outcomes of the machine learning model. This research does not focus on the intermediator agent[16] nor an autonomous generative system to serve as a so-called AI artist. It is the creative practitioners' responsibility to utilize this tool to express their creative intentions.

Activities: This framework focuses on the primary actions that creative practitioners can compose together to accomplish a more complex task. These actions can range from rough fabrication activities—i.e., basic subtractive or additive manufacturing—to primary elements of performance art that can be actuated using robotic arms or generating stand-alone graphical elements. As illustrated in Figure 1, this research doesn't address the process of generating novel creative pieces nor an autonomous process to plan a sequence of actions to embody such pieces. This thesis intentionally distances itself from these two phases, as they are more inclined toward autonomy and computer creativity. Instead, it focuses on providing a creative practitioner, who already has addressed these two steps, with a tool to embody its artistic expressions.

Datatype: Based on the domains of activities defined for this thesis, the data types are limited to two categories that foster a wide range of applications in the field of design and performative art: 1) temporal three-dimensional data—i.e., motion capture streams, and 2) 2D images.

Limitation: Considering the wide range of creative activities, it is virtually impossible to develop a universal framework to address them all. Accordingly, the framework's scope of functionality is limited to specific activities adapted in the two case studies.

---

[15] While the discussion on context contributes to the understanding of a situated approach to toolmaking, the in-depth discussions in this appendix could disrupt the fellow of this dissertation. Accordingly, I found it more suited to include it as an appendix rather than a chapter within the body of this dissertation.

[16] In some use case scenarios, a stand-alone action is the goal of activity. For example, in cinematography a single camera motion is sufficient for the cinematographer. In some other activities, a sequence of actions leads to the activity's goal. In such cases, an intermediator agent is required to convert the transition between the initial state and the final state into a series of discrete actions. For example, in painting an agent (the artist or an AI agent) should decide on the sequence of brush strokes to convert a blank canvas into a finished painting.

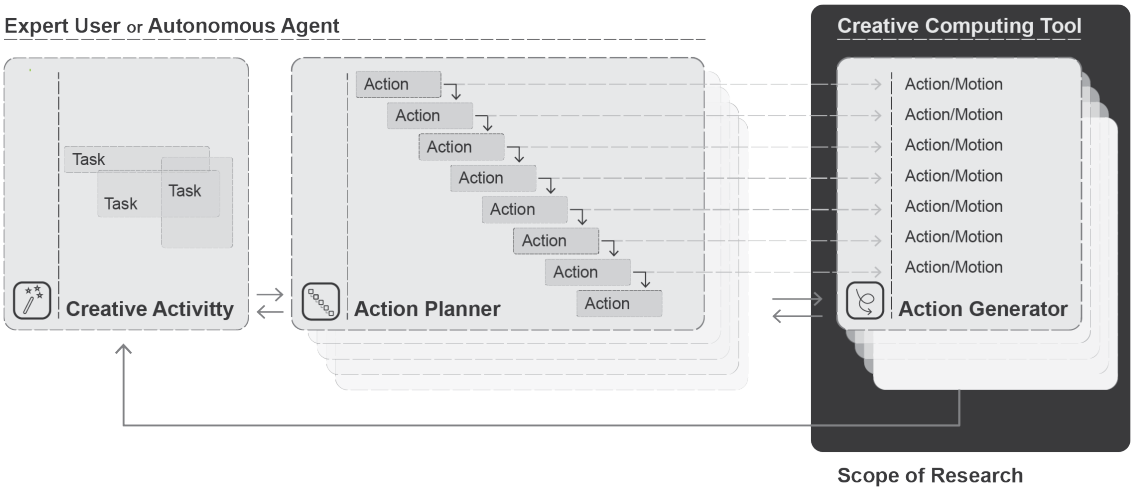


Figure 1. Scope of research concerning activities.

Note on the COVID-19 Pandemic Impact: The first case study, the SecondHand, was conducted during the peak period of the COVID-19 pandemic. In that period, in-person contact was strictly prohibited, and educational activities were limited to remote learning. That study was inevitably adjusted to accommodate these special circumstances. For instance, all activities were organized and conducted remotely, communications and discussions were transferred into Zoom video calls, and data collection methods were designed to utilize accessible and simple equipment. However, the other case study was conducted when the restrictions were partially removed, and in-person meetings were allowed.

Chapter 2. ML-Based Toolmaking for Creative Practitioners

In the previous chapter, I introduced major challenges ahead of using machine learning to make tools for creative practices. The missing context and lack of ML technical knowledge were the two topics that I found most relevant to the research questions of this thesis.

In this chapter, I will review the collective efforts of creative practitioners, interdisciplinary researchers, and engineers to integrate ML algorithms into their toolmaking process and investigate the challenges that they face in their journey.

This chapter is organized as follow: first, I briefly introduce some of the artists who have explored toolmaking with deep learning algorithms in the late 2010s. This section aims to illustrate the landscape of independent efforts to work with AI/ML in creative practices. Second, I discuss some of the technical challenges that stand in front of creative practitioners to harness ML algorithms in their toolmaking processes. I investigate three major problems: lack of ML technical knowledge, data logistics, and evaluation challenges. Third, I introduce frameworks, tools, and products that each aim to address a sub-section of these technical challenges. Following this, I focus on the efforts to address the missing context and examine this issue through the lens of researchers such as Rebeca Fiebrink, who recast the machine learning toolmaking as a human-computer interaction process with a strong emphasis on the user inputs and interactions, as well as Giulio Brugnaro, who developed a method to introduce elements from the physical context into the machine learning process.

The purpose of this chapter is not to conduct an exhaustive review of the literature, but it aims to illustrate an overall image of the shortcomings in the current efforts and learn from their experience. Learning from this literature, I will lay out my own approach, which will be presented in Chapter 3, The Framework.

Figure 2. Left: Memo Akten, "Learning to See," installation at "AI: More than Human", London, UK, 2019. Image from (Akten n.d.). *Middle: Mario Klingemann, "Butcher's son." Image from* (Artamonovskaja 2021). *Right: Tom White, "Cello," from the "Perception Engines" series. Image from* (White n.d.). *All images belong to the artists, reproduced here under fair use.*

2.1 ML Artists

Since the mid-2010s, alongside the new bloom of machine learning, a growing number of artists, scholars, and interdisciplinary teams have been exploring the new boundaries of ML and Art. Gene Kogan (Kogan n.d.), Memo Akten (Memo Akten n.d.), Mario Klingemann (Klingemann n.d.), Kyle McDonald (McDonald n.d.), Anna Ridler (Ridler n.d.) were just a few examples of a thriving community who adapted state-of-the-art ML algorithms to serve as a means of creative expression.[1]

These artists used ML models such as Pix2Pix (Isola et al. 2017) and CycleGANs (Zhu et al. 2017) and modified them to create artworks. These ML models are generally available through public GitHub repositories or even open-source projects with flexible licensing that allows artists and developers to modify and repurpose them for their specific goals. For instance, in his ongoing series "Learning to See," Memo Akten used various machine learning models, i.e., variations of GANs, on different bespoke datasets to analyze a camera feed. In one of his installations, these models were connected to a live feed from a camera pointed at a scene, where audiences could manipulate the items and observe the response from the model (Akten, Fiebrink, and Grierson 2019) (Figure 2, left). Mario Klingemann, a German artist who has been actively exploring ML and art, explains that he used a chain of GANs models to create his work "Butcher's Son." He controlled the process by curating the training dataset, fine-tuning the hyperparameters, evaluating, and selecting one out of the thousands of generated variations (Klingemann 2018) (Figure 2, middle). Tom White, a New Zealand-based artist, and lecturer at the University of Wellington, developed a series of machine learning models to make a "perception engine," which could generate abstract representations of various objects (Figure 2, right).

[1] I discussed a few examples of their works previously in (Bidgoli, Kang, and Cardoso Llach 2019).

2.2 Technical Barriers

Despite the efforts that I briefly introduced here, the adoption of ML algorithms in creative toolmaking is not a trivial process. The creative practitioners that I mentioned above dedicated significant resources to curating the training datasets, fine-tuning the hyperparameters, proposing new user interaction models, tweaking the model's architecture, or chaining models together. Adapting such ML algorithms that are originally geared toward the ML research community is a process that requires experience in computer programming, familiarity with ML algorithms, knowledge of data pipelines, and experience with methods to reliably evaluate the results.

In this section, I will first introduce three major challenges that collectively hinder creative practitioners' ability to work with ML algorithms in their toolmaking process. Then, I will introduce some of the efforts to address these challenges by reviewing the current state of ML-based toolmaking for creative practices.

2.2.1 Lack of ML Technical Knowledge

Creating tools for creative practitioners using machine learning algorithms is not new phenomenon (Bernardo, Grierson, and Fiebrink 2018). However, efforts to make these tools, with a few exceptions, rely on frameworks aimed at niche audiences such as researchers and developers with extensive prior exposure to machine learning (Roberts, Hawthorne, and Simon 2018).

While creative practitioners and toolmakers are experts in their field of work, extensive knowledge of machine learning is a rarity among them. This lack of ML technical knowledge affects their ability to integrate machine learning algorithms into their toolmaking workflows. Thus, they are more inclined toward off-the-shelf toolkits, online public datasets, and pre-trained models. Moving beyond these safe options requires a steep learning curve and a significant level of computational resources, while increasing the odds of encountering technical problems and data challenges. These creative practitioners can still use ML models in their workflow; however, their flexibility is usually limited to feeding different datasets, combining pre-set features, or in the case of Natural Language Processing (NLP) models, crafting prompts.

An interesting example of creative practitioners using ML in their work without getting deeply engaged with the technical side comes from the early days of Obvious, a Paris-based trio of French artists. When in 2018 they had one of their pieces sold in the Christie's auction (Christie's 2018), they did not need to go deep into the details of the ML model that they were using. What they found most suitable for their work was a public GitHub repository that allowed them to re-train a variation of DC-GAN (Radford, Metz, and Chintala 2015), named Art-DCGAN (Barrat 2017), with a dataset of classic portrait paintings to generate their piece *Portrait of Edmond Belamy*. When I reached out to the team, Hugo Caselles-Dupré confirmed that their role was limited to moderating the training data, from online datasets, and selecting the best ones among the pool of outputs.[2, 3]

2.2.2 Data

Machine learning algorithms run on data, either collected in the wild, or synthetically produced. Curating valid and reliable data is a delicate task that can easily go off the rails and undermine the credibility of the whole effort. In Appendix II: The Context, I thoroughly discuss how traditional methods of data collection are incapable of addressing different aspects of skill and, consequently, toolmaking. Here, I

---

[2] Hugo Caselles-Dupré, email message to author, April 4, 2019.

[3] I should assert that in their more recent works, the Obvious Art team demonstrated a drastically sophisticated technical literacy that signifies their development since then.

define the problem of data from a technical point of view to address two issues that are closely tied to the topic of this research: representativeness and scale.

Representativeness

Representativeness signifies if a dataset represents what it is intended to represent. This is especially critical when the subject has multiple independent factors involved. For instance, to create a representative dataset for a robotic wood carving system, Brugnaro and Hanna incorporated various data to account for the complexity of the physical context, i.e., tools and materials. They captured the tool motions, torque level, type of wood, direction of wood grain, and carving results (Brugnaro and Hanna 2017).[4] Meanwhile, it should be noted that overloading the dataset with a flood of features from the physical context is not necessarily a reliable solution, and may overwhelm the ML algorithm.

When it comes to toolmaking for an expert user, the physical context should be complemented with the information from the expert users' inputs. It is impossible to capture or store tacit knowledge, hence the name tacit. Designing and curating a dataset to, even partially, represent a specific skill and its tacit aspects is not a trivial task. In such scenarios, an AI/ML expert alone cannot decide on which signals to pick and which ones to factor out. However, a close collaboration between the ML expert and knowledge domain experts can considerably facilitate the process. I will discuss this matter in more detail through the two case studies.

Scale

In some branches of ML, specifically deep learning—which is abundantly popular among the creative computing community—the scale of data overwhelmingly influences the process of data collection, labeling, and auditing. Deep learning models require large, diverse, and balanced datasets and should contain enough samples to allow data-intensive models, such as GANs, to learn hidden patterns in the dataset without overfitting. Powerful models tend to rapidly overfit on smaller datasets, introducing complex challenges into the training phase. Also, the datasets should be inclusive enough to cover a wide range of samples with a balanced distribution over all categories or clusters.[5]

Scale is also a significant challenge issue when incorporating physical context into the machine learning process, where the costs associated with generating samples can snowball rapidly. Some researchers collect and process samples through a manual process. [6] For instance, Bard et al., at the Digital

---

[4] They used this dataset to train a machine learning model to predict the sequence of motions to carve a piece of wood using a robot-mounted chisel.

[5] To get a better grasp of this issue, it is enlightening to have a closer look at some of the recent research efforts on the intersection of ML and Computer-Aided Architectural Design (CAAD). The CAAD community has been actively exploring affordances of ML in architectural design, analysis, and simulation. A recurring issue in ML/CAAD literature is the negligence of curating a sufficiently large, diverse, and balanced dataset to begin with. There are several samples of GANs models being trained on samples of 100 or less (Huang and Zheng 2018; Chan and Spaeth 2020; Newton 2020; Cho et al. 2020). Researchers use pre-trained models or apply data augmentation to compensate for this issue. One exception in this category is (Nauata et al. 2020), where the authors developed an elaborated pre-processing workflow to convert over 117,000 samples from the LIFULL HOME's dataset ("Informatics Research Data Repository LIFULL HOME'S Dataset" n.d.) to train their model. Even in this case, the dataset is still biased toward a tiny section of possible data points. The dataset contains over 5 million plans of rental properties in Japan. Thus, it is already missing all the samples that are not for rent, are not located in Japan, and has not been surveyed.

[6] Building the research on a poorly curated dataset is a problematic issue, especially when creating new samples requires users' participation, consuming materials, and time on the equipment. In one example, a dataset with a minimal size of samples was used to train a model, presumably due to the fact that creating new samples would require large metal sheets going through an incremental forming process (Rossi and Nicholas 2018). To mitigate this issue,

Fabrication Lab at Carnegie Mellon University, manually took pictures of rendered surfaces and assigned an appropriate label to each picture (2018). They utilized data augmentation methods (Perez and Wang 2017) to increase the size of the dataset to train their model.

In contrast, some researchers develop automated workflows to generate samples on scale. Notably, Luo et al. set up an automated pipeline using a robotic arm and a computer-vision post-processing pipeline to bend 34 plastic strips under various forces and record their transformations to generate a dataset of 360 samples with 162 frames per sample (2018). Chen et al. developed a 3D printer setup to automatically print stand-alone curves made of polylactide (PLA) between two points in space. They leveraged this apparatus to print several samples and collected the data to create a sizable dataset to train and test different machine learning models (Chen et al. 2020).

Some researchers adopt a hybrid approach. For example, Brugnaro and Hanna collected the initial dataset directly through user demonstrations. They collected multiple variables to form a dataset of 1500 samples directly from the user demonstrations. To increase the dataset size, they managed to apply some variations to the collected samples, replay them on a robotic arm, and record the cut made by the robot. The result was a larger dataset that could be expanded further without directly engaging the user (Brugnaro 2020).

It is also possible to use a combination of physical demonstrations and reinforcement learning models to curate their training datasets. For example, (Liang, Kamat, and Menassa 2020) use Imitation Learning to train a robotic arm to perform quasi-repetitive tasks in construction setups. In this approach, the model only observes a limited number of samples from the user and then applies the reinforcement learning method to learn the policy.

Scale is particularly challenging when working one-on-one with expert users to make tools for them. In such scenarios, it is not feasible to collect and curate a large dataset. This limitation determines which ML architectures can be used. Models that can be trained on smaller datasets, such as variational autoencoders (VAE), are more suitable for such scenarios. Moreover, we can be more forgiving about overfitting when training an ML model for a specific expert user with the sole goal of using it for that specific person and specific task. In such a scenario, overfitting the model over the small dataset is, in fact, the goal. In the next chapters, I will elaborate on these two points and explain how combining a small dataset, a bespoke ML model, and intentional overfitting can address the problem of scale.

2.2.3 Evaluation

The other important challenge of machine learning in creative activities is the lack of explicit metrics to evaluate the performance of a model. Evaluating the outcomes of a machine learning model, specifically generative models, is a challenging task and has been a subject of study among machine learning researchers. Evaluation metrics in creative practices and crafts are primarily qualitative, which renders evaluation subjective and inconsistent. In general, there are two major factors that are needed to be assessed when evaluating the outcomes of a generative model, 1) the fidelity, which describes how closely the generated samples resemble the training data, and 2) the coverage, which explains what fraction of the training data is being represented by the generated samples.

---

some researchers adopt data augmentation methods without considering the principles behind them. In one case, only eight samples were collected, then augmented to 8000 to retrain a Pix2Pix model (Ramsgaard Thomsen et al. 2020). Such poorly curated datasets result in overfitting and eventually weak generalizability of the models. This shows itself as a large gap between training and test errors. In the case mentioned above, these numbers were 94% and 75% for training and test accuracy, respectively. In such cases, the model is simply impractical for any further application.

Researchers suggest different methods to assess the outputs of generative models (Salimans et al. 2016), for example, using human users to check the outputs' fidelity. It is a common practice to use crowd-sourcing methods, such as Amazon Mechanical Turk, to accelerate and scale up the process. This approach is highly dependent on the subjective decisions of human evaluators. Therefore, the process is sensitive to evaluators' demography, motivations, and background. This approach may return inconsistent results, however, researchers observed that auditing the results and providing feedback on the mistakes made by the evaluators can significantly enhance the reliability of the process (Salimans et al. 2016).

Moreover, this approach is most effective when the subject of the study is commonly understandable by non-expert evaluators. For instance, evaluating the human faces generated by an ML model, or assessing the visual fidelity between the brushstrokes a human user has drawn and brushstrokes generated by an ML model (Bidgoli et al. 2020), or testing a model's performance in detecting urban objects from aerial images (Koh and Huang 2019). However, in niche use cases, where evaluation requires expert-level knowledge, finding the right crowd is a concerning issue.

The other method is based on using pre-trained discriminative models to detect the presence of meaningful objects and check the distribution of those objects over the training dataset. The intuition behind this method is that if the quality of outputs is good enough, then the classifiers trained on real images should be fooled by the generated samples and classify the synthesized results as real ones (Isola et al. 2017). Using this method, users could evaluate a generative model's ability to generate meaningful samples and compare the distribution of those samples over the training dataset.

However, the state-of-the-art classifiers are designed, developed, and trained for very specific data types and tasks, i.e., face detection in images, semantic segmentation in 3D models, voice recognition in sound clips, etc. For a wide range of creative activities, there is no suitable classifier available.

More importantly, in creative practices, toolmaking evaluation is a subjective procedure that is closely coupled with the expert users' preferences and subjective measures. The current approaches that were mentioned above cannot pertain to these metrics. Thus, in this study, I will focus on methods to allow the expert user to collaborate in the toolmaking process and provides its subjective evaluation during various stages of the toolmaking process.

2.3 Tools of Creative ML

Computational artists and toolmakers have tried to compensate for the technical barriers and allow users with different levels of experience in machine learning to adopt ML-based tools to facilitate their workflow. They materialized their efforts in the form of simplified APIs and libraries, integrating ML libraries to popular software packages, designing user-friendly graphical user interfaces (GUI), simplifying the data pipelines, providing pre-trained models, integrating cloud computing services for faster computation, or in some cases, black boxing the backend ML models.

These collective efforts resulted in toolkits, libraries, plug-ins, and APIs, including but not limited to, Magenta—and the family of tools based on Magenta, i.e., Magenta.js (Roberts, Hawthorne, and Simon 2018), and Magenta Studio (Roberts, Engel, et al. 2019)—, ml5.js ("Ml5.Js: Friendly Machine Learning For The Web" n.d.)—and tools developed based on ml5.js such as Teachable Machines ("Teachable

Machine" n.d.)—, mlLib (Bullock and Momeni 2015), Unity ML-Agents (Juliani et al. 2018), Runway (RunwayML 2019), Wekinator (Fiebrink, Trueman, and Cook 2009; Fiebrink 2011) (Figure 3). [7]

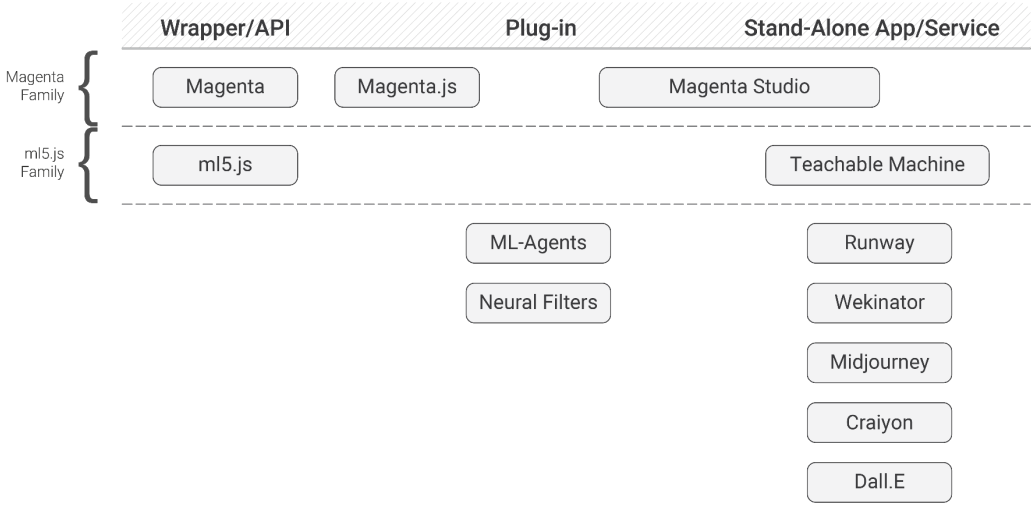


Figure 3. The spectrum of ML-based tools geared towards creative practitioners.

Simplified APIs and libraries: Some of the tools mentioned in the previous section are targeted toward users with prior knowledge of programming and machine learning. Those are usually developed as libraries for programming languages that are already popular among the creative computing community. For instance, ml5.js and Magenta.js[8] are two open-source libraries developed for JavaScript. The decision to use JavaScript is very interesting and deserves a closer look. First, it should be noted that JavaScript is a popular programming language in the creative computing community. This popularity makes it easier for Magenta.js and ml5.js to serve a broader range of audiences. Second, from a technical point of view, JavaScript is easy to set up and use. It allows users to run their code directly in the browser on a wide range of platforms and devices. Moreover, it helps users bypass the technical issues associated with other machine learning frameworks written for Python and C++, which require advanced setup procedures to leverage hardware integrations.

APIs for popular software packages: Another approach is aimed toward bringing ML algorithms into environments that creative practitioners are already familiar and comfortable with. This would encourage and empower them to explore the potential of the ML in their workflow without getting deeply involved with complex backend problems (Roberts, Hawthorne, and Simon 2018). It is common to see these tools accommodate integration with input/output protocols to interface with other software packages and various pieces of hardware. This is a significant incentive to make these tools more desirable for a specific group of creative practitioners, such as musicians, visual artists, and creative computing artists.

[7] Additionally, there is also a movement to inquire into machine learning and art, mostly focused image-based synthesis. A notable example of this movement is the Machine Learning for Art (ml4a) which is a collection of tools and educational resources which apply techniques from machine learning to arts and creativity developed and maintained by Gene Kogan (Kogan n.d.).
[8] Magenta.js is a simplified JavaScript API based on Tensorflow.js.

Bullock and Momeni followed this approach while developing ml.lib, a cross-platform[9] and open-source machine learning tool for Max and Pure Data (Bullock and Momeni 2015).[10] Ml.lib is geared toward musicians with minimal prior knowledge of machine learning but eager to experiment with it in the context of interactive art.

Another interesting example in this category is Unity ML-Agents. This toolkit is aimed to bring a handful of reinforcement learning algorithms to this popular game engine. Although Unity natively supports C# scripting, the ML-Agents library is written in Python, which is the language of choice for a wide range of ML researchers and enthusiasts. This resulted in a less-than-optimal workflow, forcing the user to switch back and forth between Unity's native game development space and ML-Agents reinforcement learning algorithms.

User-friendly GUI: A common theme among the machine learning toolkits for creative practitioners is to wrap ML algorithms, evaluation methods, and supporting tools with a user-friendly and simplified GUI. Such interfaces can improve the usability of machine learning-based tools and subsequently increase their adoption by creative practitioners in their workflows (Bernardo, Grierson, and Fiebrink 2018). Wekinator and Runway are two examples of this approach. Magenta Studio, developed based on the Magenta.js backend and wrapped in a minimal GUI, falls under this category too. It is developed as a plug-in for Max for Live for Ableton Live (Ableton n.d.)—a popular software package for music creation and live performance—but it is also available as a stand-alone application with its own GUI.

Black box ML tools: Software packages such as Photoshop treat machine learning-based features as black boxes. Users can use them to achieve certain functionalities with close-to-no engagement with the underlying ML algorithm. This simplicity and ease of use come with a tradeoff; the user has no direct method to interface with the backend to curate data, retrain the model, or fine-tune it for novel use cases.[11]

Simplified data pipelines: Simplifying the process of collecting, labeling, and auditing data is another effective approach to make ML tools more accessible to novice users. For instance, Teachable Machines ("Teachable Machine" n.d.) provides an intuitive interface to collect and label data samples in the browser, which allows the user to curate a data set in a few minutes. Wekinator is also a good example that supports various input formats and modalities to interactively curate a dataset.

Pre-trained models: Training a machine learning model from scratch is a time-consuming and resource-hungry process.[12] One method to mitigate this issue is to provide the users with a pre-trained model and let them re-train the model based on a new dataset provided by the user. This is an efficient and effective method to re-purpose large and deep models in a relatively short time with less-capable hardware setups. Teachable Machines and Magenta Studio follow this approach and provide pre-trained models for image and sound classification, pose estimation in Teachable Machines, and music generation in Magenta Studio.

---

[9] ml.lib supports both x86 and ARM processors and works on Mac OS, GNU/Linux, and Windows. This range of supported platforms renders ml.lib accessible on a wide range of modern hardware and music devices.
[10] ml.lib is developed as a wrapper for the C++ library developed by Nicholas Gillian (Gillian and Paradiso 2014).
[11] Projects which are mostly designed as a tech-demo, fall under this category. A notable examples is GauGAN family, i.i., ("GauGAN2" n.d.), that are now integrated with NVIDIA Canvas (NVIDIA n.d.).
[12] Specifically, deep CNN and massive NLP models are extremely expensive to train. Training these models from scratch is not financially feasible nor environmentally sustainable.

Cloud-computing: As discussed earlier, training machine learning models is a computationally heavy task. It also requires installing various software packages and applying the necessary settings.[13] Some toolkits tackled these two issues by integrating cloud-based computing services. In such cases, the training process is off-loaded into a cloud computing server, with all the required libraries and software packages pre-installed. For instance, Runway (RunwayML 2019) offers cloud computing services integrated into their app, and users can purchase time on their servers to train or use heavy models.

2.3.1 Magenta

Magenta is a free and open-source project that includes libraries and code snippets to help artists who work with deep learning based on the Tensorflow deep learning framework (Magenta n.d.; Abadi et al. 2016). In 2016, the Google Brain team released the initial version of Magenta. Since then, several artists have used it as the backend library to train their ML-based musical tools, mostly over open-source datasets (Magenta n.d.). This was followed by the introduction of Magenta.js, based on Tenserflow.js, to allow users to run the models, or in limited cases train, their models in a web browser.

Using Magenta requires a moderate background in ML and programming, and it is most suited to serve as the backend for other tools. Brain team later developed the Magenta project one step further and introduced Magenta Studio as an independent application and a series of deep learning-based plug-ins for the popular live music production software package, Max For Live for Ableton Live (Roberts, Engel, et al. 2019) (Figure 4).

The researchers behind the Magenta project followed a series of design principles from (Kayacik et al. 2019) to make the tools more desirable for the target users. They advanced toward this goal by designing a simplified UI with vocabulary that the users are already familiar with, simplifying the installation process on the operating systems and hardware, and designing modular components that could be chained together similar to musical instruments. Magenta has multiple plug-ins that were supported by deep learning models such as MusicRNN and MusicVAE (Roberts et al. 2018).[14] The ML models used in Magenta Studio are pre-trained and the users can use them to produce desired results based on their taste.

[13] Virtual environments (for Python) and Dockers are some of the methods to eliminate the setup issues. However, they cannot address the hardware limitations. Another approach is to use cloud services such as Google CoLab, which is optimized to work with interactive Python Notebooks, has most of the useful libraries pre-installed, and provides different tiers of hardware over different tiers of subscription (Google, n.d.).

[14] There are other machine learning toolkits for creative practitioners, including but not limited to, XMM and Gesture Variation Follower which are not discussed in this section. For a closer look at these and other tools please look at (Bernardo et al. 2017; Caramiaux et al. 2014; Françoise, Schnell, and Bevilacqua 2013).

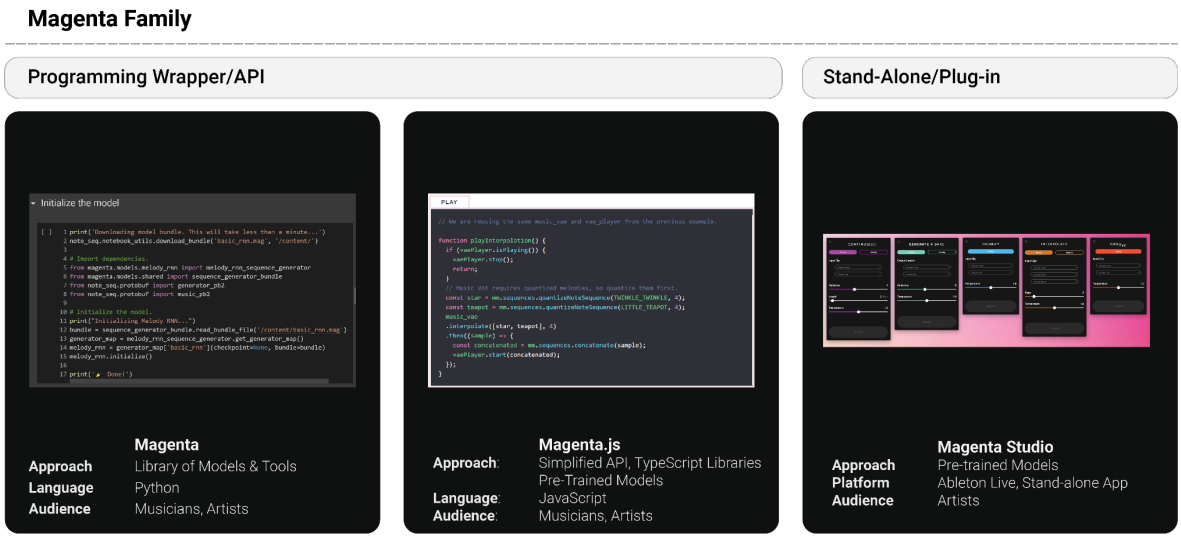


Figure 4. Magenta, Magenta.js, and Magenta Studio, image from left to right from ("Hello Magenta" n.d.; "Making Music with Magenta.Js" n.d.; Roberts, Mann, et al. 2019).

2.3.2 ML-Agents

ML-Agents is a package for the Unity game engine that includes a library of RL tools written in PyTorch (Juliani et al. 2018) (Figure 5, left). [15] It taps on the extensive 3D capabilities and physics simulation engine of Unity to create environments to train its pre-defined RL models. The familiar environment of Unity, a widely popular engine among game developers worldwide, made this library accessible to a wide range of audiences with prior knowledge of design, 3D modeling, game development, and scripting.[16] Working with ML-Agents still requires a significant programming background to develop the game in Unity's natively supported C#. However, it has reduced the machine learning technical interface into a simplified set of hyperparameter adjustments organized in a structured setup file.[17]

2.3.3 Wekinator

Wekinator is open-source software for interactive machine learning developed by Fiebrink in 2009 (Fiebrink, Trueman, and Cook 2009) (Figure 5, middle). It is developed based on Weka, an open-source and free ML library written in Java (Witten and Frank 2002).

---

[15] Although Unity is primarily geared toward game developers, it is also well-received among researchers in other fields, such as robotics where RL is an active field of research.

[16] In 2020, when I was working on the early draft of this chapter, the ML-Agents repository on git-hub had seen a steady stream of updates and revisions. While the early versions were implemented with Tensorflow deep learning library, the latest versions are developed using PyTorch. The constant updates and changes of this toolkit negatively impacted its usability. As of 2022, the changes are less frequent, and the library is more stable.

[17] The library is designed to serve game developers' purposes primarily. However, researchers in other fields such as architecture are also enthusiastically following its application in their practice (Hosmer and Tigas 2019). Unity's advantage resides in its ability to export the environments as stand-alone applications for desktop computers, mobile devices, and web-based apps, making it possible to run the machine learning model directly on a wide range of devices. From this point of view, it resembles some of the affordances of JavaScript-based libraries that were discussed earlier.

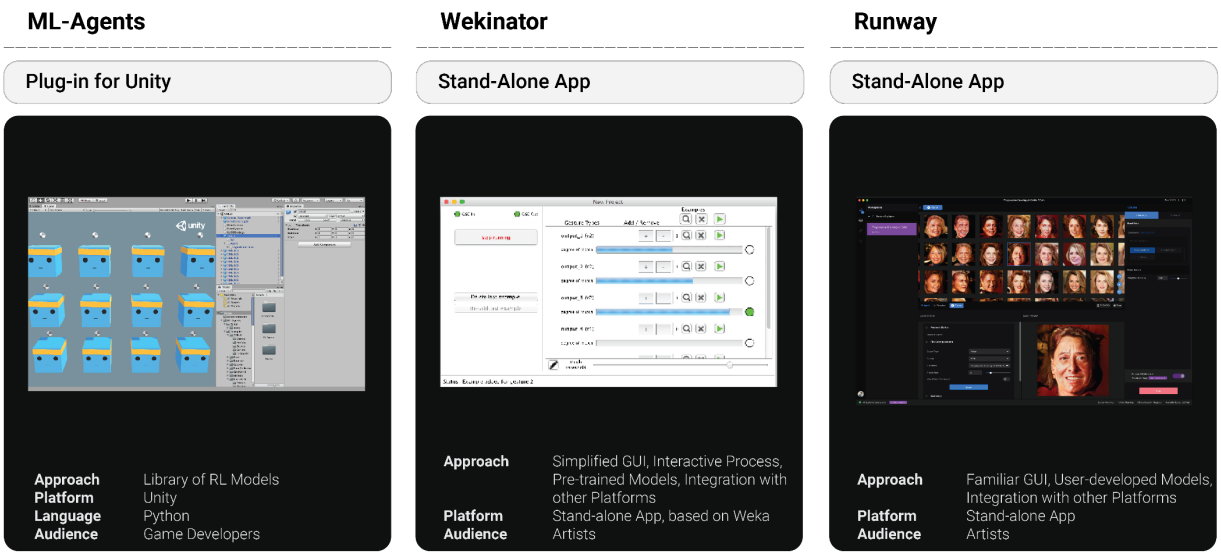


Figure 5. ML-Agents, Wekinator, and Runway interface, images from left to right from (Unity-Technologies 2021)*,* (Wekinator n.d.)*, and* (Valenzuela 2019)*.*

Fiebrink describes Wekinator as a meta-instrument, an instrument that can be used to create other instruments. As a stand-alone app, it packs a handful of machine learning models with features for collecting training data samples, changing miss-classification costs, and changing the weights of each sub-model in an ensemble model. Wekinator also supports a wide range of input and output protocols. These features allow the users to curate their own dataset interactively and adjust the selected model's behavior to steer its learning direction toward their desired direction (Bernardo, Grierson, and Fiebrink 2018).

2.3.4 Runway[18]

Runway was initially started as a thesis project at the New York University and evolved into a creative machine learning platform with a graphical and user-friendly interface that closely resembles popular graphic design software packages that visual artists are already accustomed to (Figure 5, right). Runway incorporates several open-source ML models developed either by the core team or the community of users. Models can run or train either on the Runway servers on the cloud or on the users' local machines using a Ducker container.[19] Following this approach, novice users can select a model from the library and start using it without complicated installation and setup procedures. The user interactions are also simplified and implemented in a familiar visual vocabulary of creative practitioners.

For expert users, Runway provides a software development kit (SDK) to port custom machine learning models. These models can be designed, implemented, trained, and fine-tuned by the user, then interfaced through Runway's GUI. The package is also equipped with various plug-ins/APIs to integrate with other popular toolkits among artists, such as Open Framework, Processing, Rhino, and Photoshop (RunwayML 2019).

---

[18] What is discussed here as Runway, refers to the original product made by RunwayML company. As of now, summer 2022, RunwayML has pivoted Runway from what is described in this thesis into an ML-based video editing tool.
[19] "A Docker container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another" (Docker n.d.).

2.3.5 Teachable Machines

Teachable Machines is a web-based tool developed based on ml5.js. It sports a simple and intuitive interface to curate datasets directly in the browser and a set of pre-trained machine learning models for image and audio classification as well as pose estimation (Figure 6, right). The trained models can be used directly in the browser, or they can be deployed on the cloud and accessed through a simplified API over the internet (Figure 6, left).

The significant advantage of Teachable Machines is its clear and intuitive interface. Users can run it in a web browser without any setup and curate their dataset interactively without any engagement with technical details. Although some basic training hyperparameters for fine-tuning are provided, the default settings are quite sufficient for most cases, and users can train their models with a few clicks.[20, 21]

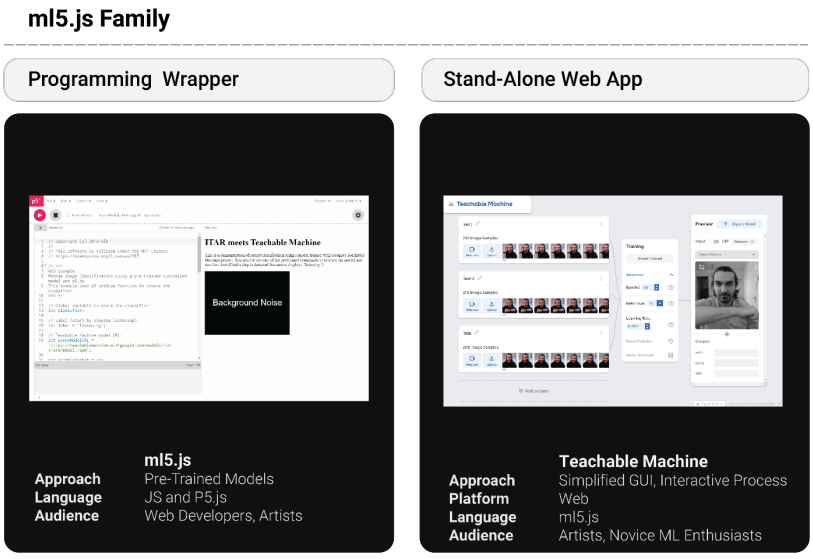


Figure 6. ml5.js and Teachable Machines interfaces, screenshots from author's projects.

[20] I found Teachable Machine a valuable pedagogical tool to teach basic concepts of machine learning models to architecture students. For instance, in 2019, students of 48-755/48-755 *Introduction to Architectural Robotics* used Teachable Machines to make a voice control system for a robotic fabrication system without any prior ML experience.

[21] Finally, it is worth mentioning the absence of a prominent toolkit in the list, Grasshopper, the visual programming plug-in for the Rhinoceros modeling software package. Along with Dynamo, it is the most popular visual programming interface among architects. Although there are a few machine learning add-ons available for Grasshopper, their presence in the literature is very sparse in pedagogical (Khean et al. 2018) and research studies (Rossi and Nicholas 2019; Brugnaro 2020).

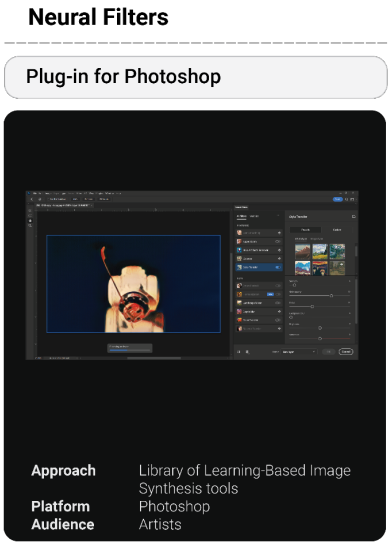


Figure 7. Neural Filters interface in Photoshop, image by the author.

2.3.6 Neural Filters

Neural Filter is a set of learning-based image synthesis tools organized as a plug-in for Photoshop. The package contains several tools to apply visual style transfer, background removal, and object selection. The developers kept all the technical details inside a black box. Users have no means to directly provide data, train models, or customize the models. However, like other tools in Photoshop, it is possible to manipulate some input parameters to adjust and fine-tune the results (Figure 7).

2.3.7 The Dawn of Text-to-Image Models

The landscape of art and machine learning has changed in the past two years with the rapid development of multi-modal machine learning algorithms, specifically models than use a combination of natural language processing and computer vision. In January 2021, OpenAI introduced Dall.E, a transformer-based language model that could translate text prompts into close-to-photo-realistic images (Ramesh et al. 2021). Following that, projects such as Dall.E 2 (Ramesh et al. 2022), Imagen and Parti by Google (Saharia et al. 2022; Yu et al. 2022), and Midjourney by Midjourney lab (Midjourney Lab 2022) have become the face of AI-generated "art" (Figure 8).

The machine learning algorithms behind these projects are mostly proprietary and closed source. Accessing them is usually through invitation-based programs.[22] Training these models usually requires massive computational resources, only available to a handful of technology giants and state-level organizations. Accordingly, the possibility of adapting or fine-tuning these models for creative practices by individual artists is slim. However, some efforts to replicate their results have been made with different levels of success. For instance, to make Dall-E Mini, Dayma et al. used a down-sized model compared to

[22] At the time of writing this chapter, these projects are rapidly entering the public beta stage. The pace of development is so fast, that even between the revisions of this document, researcher managed to partially replicate some of these models and deploy them on platforms such as Hugging Face.

the one behind Dall-E and utilized pre-trained models instead of training the model from scratch (2022). It was a technically complex effort conducted by a group of ML experts.

The primary interface between the users and these models is a prompt, a body of text that expresses what the user wants the model to generate. Once a batch of images is generated, users can either pick one and delve deeper in its direction or enhance the resolution of their choice. These models behave as black boxes; users cannot obtain any meaningful grasp of their internal workflow. Thus, crafting "the prompt" and navigating the results are the new skills that users are mastering through a trial-and-error process. [23]

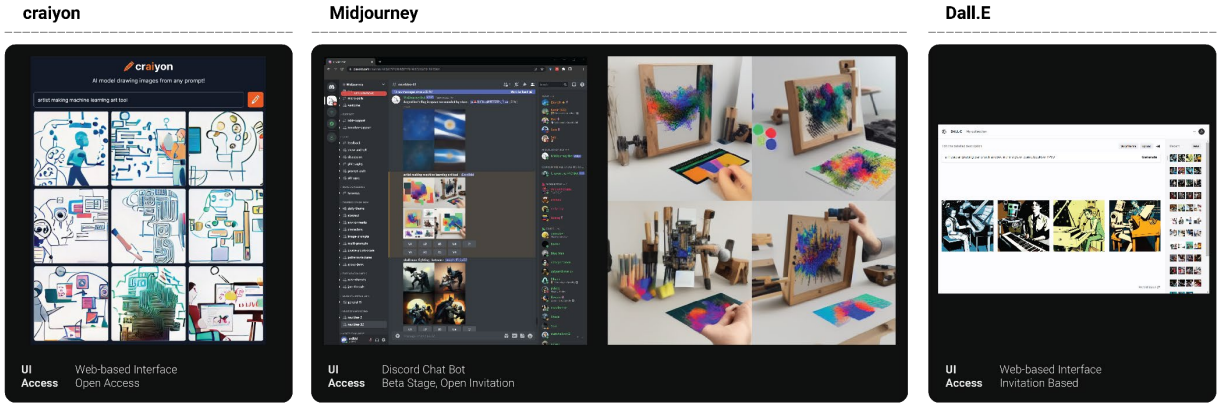


Figure 8. Left: Interface of craiyon, formerly known as Dall.E Mini, with the prompt: "artist making machine learning art tool." Middle: Midjourney UI implemented as a Discord chat bot. Right: example of results with the same prompt. Right: Dall.E interface. Screenshots by the author.

2.3.8 Accessibility Dilemma

The tools discussed above represent a spectrum of solutions to make ML tools accessibility to creative practitioners, which are summarized in Figure 9. They reduce the necessity of engagement with the technical details while focusing on exploring the process and the outputs. Thus, they allow creative practitioners to focus on the creative process rather than the technical challenges that working with ML models entails.

However, better accessibility comes with the trade-off of losing flexibility and control over the machine learning algorithms. On one end of the spectrum sits Magenta and similar programming APIs that let the user have almost full control and flexibility over the models and training process. But they are best suited for seasoned users with adequate hands-on experience with programming and ML concepts. On the other end is Photoshop-like applications, which provide an easy and off-the-shelf solution to use ML in a creative workflow. However, they provide the least level of control over the underlying ML algorithm.

Finding a sweet spot between these two extremes is a delicate task that requires a thorough understanding of the users and the specific use case scenario. Creative practitioners should have the option to create and customize their own learning systems by training the models themselves (Bernardo et al. 2017). Thus, a limited number of pre-trained models, simplified data pipelines, and restricted training fine-tuning might not be enough for all users.

[23] Some of these tools come with extra functionalities and editing tools as well. For instance, Dall.E 2 has some image editing functionalities.

Moreover, simplification for the sake of accessibility weakens creative practitioners' understanding of their tool, which is a critical factor in users' ability to control and harness their potentials (Rahwan et al. 2019). Although these efforts help users with limited experience and technical knowledge to have the opportunity to modify existing models or, to some extent, develop and deploy their own models, this simplicity and accessibility are not always positive. Novice users who are not necessarily familiar with ML's fundamental concepts are more likely to fall into the common pitfalls of machine learning. For instance, experienced users can easily detect and avoid issues such as misjudgment of a model's performance based on the training results instead of test results, overfitting models in the training phase, or model-specific problems such as mode collapse in GAN models (Veloso et al. 2022).[24]

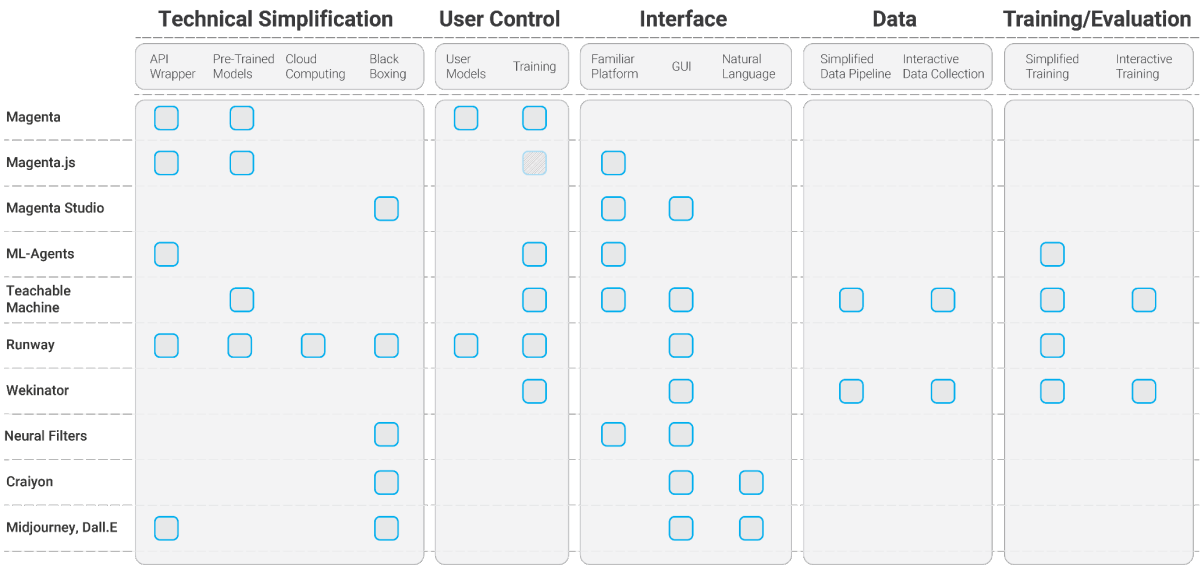


Figure 9. Approaches to make ML tools accessible to creative practitioners in the literature.

2.4 The Missing Context

While reviewing the literature on ML-based toolmaking for creative practitioners, it became clear that with a few exceptions, most of these efforts ignored the contextual relationships between creative practitioners, their tools, and their practices. As the lion's share of efforts in machine learning communities is focused on developing novel algorithms and improving their efficiency, there are fewer resources dedicated to informing the design process and the results with inputs from the users and the contexts in which users will eventually use them (Simard et al. 2017).

[24] To reflect on this issue, I tap on the CAAD/ML literature again. A glance at the growing number of published papers and dedicated sessions to ML in CAAD conferences signifies the trend around the topic. However, a thorough review of the literature highlights fundamental issues, i.e., lack of a clear and transparent methodology or a comprehensive report about the process and results. In some cases, machine learning methods were developed to accomplish tasks that could be otherwise addressed efficiently without it, or confusion between machine learning and statistics. These issues signify the absence of a thorough understanding of machine learning paradigms and technical knowledge to design, conduct, and report ML-based research among the authors of such papers.

To illustrate the potential role of context in the toolmaking process, it is helpful to mention one of the precedents at the CMU School of Architecture. In 2014, Bard et al. published their research on robotic surface rendering. Their approach entails capturing skilled workers' hand motions and then replaying them with an articulated robotic arm (Figure 10, left). Later in 2019, a similar approach was practiced as part of the *Human Robot Virtuosity* class, where students collaborated with a group of local creative practitioners and skilled workers, including graffiti artists, wood printers, and plastering experts (Figure 10, right).[25] Although Bard did not use any ML algorithm in this research, the overarching theme of centering practitioners and allowing them to introduce their tools and methods helped them to incorporate various elements of context into the toolmaking process.

Figure 10. Robotic plastering (left) and graffiti (right), images from (Bard et al. 2014; Naseck, Ng, and Tsai 2019).

2.5 ML and Embracing the Context

2.5.1 Elements of Physical Context

Returning to the literature of ML-based tools for creative practitioners, there are a few examples where researchers who have investigated methods to study the potentials of ML in working with material behavior and tool characteristics to improve robotic fabrication methods for wood carving and metal forming (Brugnaro and Hanna 2017; Rossi and Nicholas 2018). Giulio Brugnaro, by that time a Ph.D. researcher from the University College of London, developed a learning system to inform the robotic fabrication workflow on the material's specific properties and behavior. In this system, an expert user demonstrates several samples of carving a wooden piece with a chisel (Figure 11). The demonstration is recorded as a sequence of motions in space with six degrees of freedom—representing both location and orientation of the tool at each timestamp. A series of machine-generated samples later augmented these examples to form a learning dataset. Brugnaro designed a neural network to map these chisel motions into the cut they made on wood pieces. The final ML models could predict the result of a chisel motion on a specific piece of wood, and inversely, predict the necessary chisel motion to create a given cut on the wooden piece.

In his workflow, the robotic arm, fabrication system, data collection tools, and material feedback sensors are intertwined to inform the machine learning model about the expert user's techniques, the tool, and the material in use. Brugnaro describes such systems as "… soft systems, both adaptable and continuously

[25] For more information, please refer to (Bard et al. 2014) and the web site of Human Machine Virtuosity course in Spring 2019 ("Human-Machine Virtuosity – An Exploration of Skilled Human Gesture and Design, Spring 2019." n.d.).

evolving, whose dynamism is constantly fed by a flow of information" (Brugnaro, Figliola, and Dubor 2019).

Despite its very interesting approach to physical context, this research touches close to the Taylorist point of view to AI.[26] The authors describe it as "… a robotic fabrication system where the instrumental and material knowledge of skilled human craftsman is captured, transferred, robotically augmented and finally integrated into an interface that make this knowledge available to the designer" (Brugnaro, Figliola, and Dubor 2019, 151).

Figure 11. Data collection apparatus, training samples, and execution of learned motions, image from (Brugnaro and Hanna 2017).

2.5.2 Idiosyncratic Elements

Brugnaro's research, as discussed above, was focused chiefly on the material-tool behavior and leaves the personal or social context unaddressed. For instance, the evolution of toolmakers' conception of the tool during the toolmaking process is not a primary focus. The toolmaker's subjective evaluations are also missing from this project. Once the learning samples are generated, the toolmaker loses its control and agency over the augmented samples, learning direction, or evaluation of the results. Meanwhile, other researchers have examined the possibility of including these factors in their toolmaking process.

Rebecca Fiebrink, a computer scientist, pioneer of Art and ML, and currently a reader at Creative Computing Institutes at the University of the Arts London, has been working on machine learning not from a mere technical point of view but the human-machine interaction (HCI) point of view. Her work on interactive machine learning is one of the most interesting examples of allowing creative practitioners to engage with the toolmaking process and introducing their subjective measures and personal preferences in data curation, training, and evaluation.

She developed Wekinator (Fiebrink, Trueman, and Cook 2009) to allow an artist to develop their tool in collaboration with an ML expert through an interactive supervised machine learning workflow (Fiebrink 2011). In this scenario, Wekinator serves as an interface to the ML backend allowing the artist to focus on creating samples as well as evaluating and assessing the tool's performance. Fiebrink, as the ML expert toolmaker, could focus on moderating the technical aspects of the process (Figure 12).

As a case study, Fiebrink collaborated with a professional composer/cellist to make a classifier that was able to recognize standard bowing gestures for live performance or composition. Throughout this collaboration, the creative practitioner contributed in two major ways: 1) providing the learning samples

[26] From this point of view, human skill could be reduced into abstract data points, which then can be acquired, contained in databases, and eventually transferred. This conception of skill as data resonates with the knowledge decontextualization that I previously discussed. I discuss the Taylorist point of view to skill, and subsequently AI, with more detail in Appendix II: The Context.

by playing her cello using a custom-made bow. They demonstrated a range of techniques on the real instrument and recorded them as a sequence of data by the sensory tools installed on the bow, 2) introducing their perception of the tool, personal feelings, and subjective assessments as a part of the evaluation and feedback process.

This project is one of the few examples that manages to incorporate physical context alongside personal preferences and subjective measures of the creative practitioners in the toolmaking process.[27] It also introduces a dynamic collaboration between the creative practitioner and the ML expert toolmaker, which has been a source of inspiration for my own work. I will revisit this form of collaboration and explain how this serves as a basis for the proposed toolmaking framework in Chapter 3.

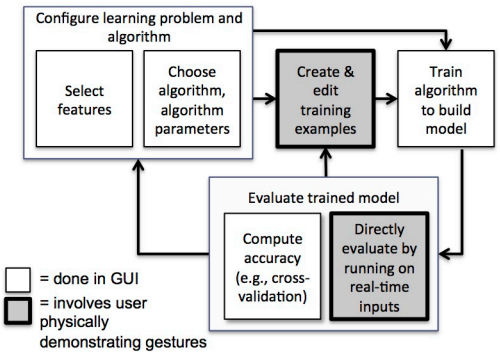


Figure 12. Interactive learning workflow in Wekinator, image from (Fiebrink 2017, 163).

2.6 Discussion

The review of literature signifies two major issues in the current state of ML-based toolmaking for creative practitioners. First, the technical barriers and second, the missing context. I discussed how the current methods of addressing technical barriers can reduce creative practitioners' control over the toolmaking process and force them to use off-the-shelf solutions and subsequently forfeit their personal preferences, subjective measures, and elements of the physical context in favor of simplicity and accessibility.

In the next chapter, I introduce a framework for ML-based toolmaking for creative practitioners that allows them to closely collaborate with ML-experts in various stages of the toolmaking process to integrate idiosyncratic aspects, elements of the physical context, and nuances of their creative practice in the toolmaking process.

[27] In her 2017 paper, "Machine Learning as Meta-Instrument: Human-Machine Partnerships Shaping Expressive Instrumental Creation," Fiebrink elaborates the idea of using interactive machine learning as a meta-instrument. She discusses how supervised learning can be leveraged to design new tools in real-time for creative activities and emphasizes the relationship between the builder and the toolmaking procedures as a key factor in understanding the new instrument. She argues that this can result in "… an exploratory, playful, embodied, and expressive …" toolmaking process (Fiebrink 2017, 137).

Chapter 3. The Framework

In the previous two chapters, I argued that the lack of attention to the context alongside the technical barriers are among the most critical challenges making AI/ML-based tools for creative practices. We observed that this phenomenon, with a few exceptions, reflected itself in the current state of ML-based tools. It became evident that the common methods to mitigate technical barriers come with the trade-off of losing control over the toolmaking process, reducing creative practitioners' flexibility, and consequently detaching from the context of the practice.

In this chapter, I propose a collaborative framework to make ML-based tools for creative practices that embraces the personal preferences and subjective measures of the practitioners as well as elements of the physical contexts of the practice. This framework will be used as a high-level guide to design and implement toolkits for the two case studies.

In this thesis, I present and discuss a collaborative framework to make ML-based tools for creative practices, *the framework* from here on. This framework aims to make ML models more accessible to creative practitioners to build bespoke tools tailored to their specific personal and physical context of practice without requiring them to engage with the complexities of the backend ML algorithms.

The framework defines a collaborative effort between the creative practitioners and the ML expert toolmaker to achieve this goal. In the two case studies presented in the following chapters, I demonstrate how the framework serves as a blueprint for defining collaborative toolmaking workflow and developing meta-tools.

The meta-tool—a term derived from Fiebrink's Meta-instrument (Fiebrink, Trueman, and Cook 2009)—refers to an implementation of the framework designed and fine-tuned for a specific case study. It provides the software/hardware platforms for the ML expert toolmaker and the creative practitioner to collaborate on the tool development (Figure 13).

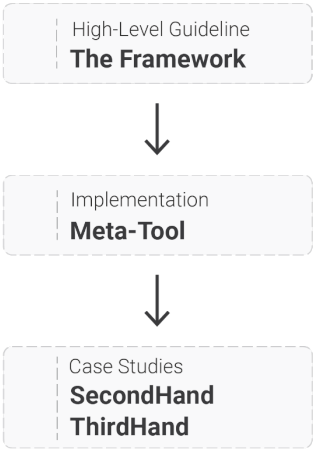


Figure 13. Framework, meta-tool, case studies

3.1 Framework Principles

This framework is designed in accordance with the two hypotheses of this research. First, it allows creative practitioners to utilize user-generated data to interface with the learning algorithm and integrate their subjective preferences and elements from the physical context to explore potentials of ML algorithms in supporting their creative practices. Second, it allows creative practitioners to collaborate with ML expert toolmakers and overcome the technical challenges in the toolmaking process.

The goal of this framework is not to make a tool to replicate the creative practitioner in any given situation. It is poised to augment the creative users' abilities and allow them to further explore their practice with this new tool.

Working with and from the Context:

The framework is built around the conception of skill, and knowledge, as situated in the context of its practice. This framework aims at centering creative practitioners in the toolmaking process. It also encourages data collection in the close-to-real-life context of the practice, with real tools and materials. The creative practitioner can introduce material behaviors and tool affordances by creating data samples in close-to-real-life demonstrations. For example, collecting hand gestures of a painter while they are

working on a painting with a brush, paint, and canvas, or recording the camera motions of a cinematographer on the set, or observing dancers' motions in a performance or rehearsal. In each case, the user can improvise, repeat, modify, or remove samples to sculpt the dataset, one sample at a time. That is a departure from the conventional approach in which an ML expert would decide on these matters. I expect that this approach reduces the chance of critical data being ignored in favor of simplicity and abstraction.

Meaningful Extended Collaboration between Creative Practitioner and Computational Toolmaker

This framework helps make ML algorithms more accessible to creative practitioners by proposing a collaborative workflow that allows them to work with ML expert toolmakers to develop their ML-based tools. The framework acknowledges the critical role of a multidisciplinary expert with sufficient ML and computational toolmaking experience. This workflow opens new opportunities for novel forms of dynamic collaboration between creative practitioners and toolmaking experts, who otherwise work separately from each other.

The framework suggests that the two parties engage in a dynamic collaboration to design, develop, and test the meta-tool. They collaboratively 1) engage in the development of the meta-tool, 2) decide on the inclusion or exclusion of data, 3) provide data samples, 3) curate the training dataset, and 4) evaluate the results. This process entails making various decisions over general aspects of work, data collection and curation workflows, user interfaces, modes of interaction, and implementations. This process helps both the creative practitioner and toolmaker gradually evolve their conception of the tool, based on a mutual understanding of each other's work, the context, and the affordances of the meta-tool. Through iterative design, prototyping, and testing cycles, the two sides learn to adjust and improve their work to make the tool.

The framework empowers the creative practitioner to introduce desired features to the learning model and steer the learning direction based on personal preferences and subjective assessments while the toolmaker addresses the technical aspects. Accordingly, the resulting tools are tailored to the specific creative practitioners and their creative activity, as they are shaped based on their experience, skills, personal judgments, preferences, and subjective measures.

Using Data to Interface with the Machine Learning Algorithms

In this framework, data is treated as a means of interaction with the ML model in the hands of creative practitioners.[1] This notion of data offers a counterpoint to the currently dominant approach in creative machine learning, where 1) data is usually rigid, pre-determined, and externally sourced, and 2) the primary method of controlling the learning model is coding. This new form of design material helps users interface with the ML backend by generating, collecting, and curating training data and incrementally shaping the learner's behavior.

Using Generative Machine Learning Models and Taking Advantage of Overfitting on Small Datasets

Machine learning research is generally associated with large-scale datasets that might be skewed and biased toward specific races, gender, geographic regions, or art genres. Generalizing the results of ML models trained on these datasets is one of the most complex challenges of ML in recent years. In contrast,

[1] This notion of data is primarily inspired by the works of Rebecca Fiebrink on *Training Data as Interface* (2016) which is also elaborated and discussed in (Cardoso Llach 2017).

this research explores the positive side of biased datasets and overfitting a machine learning model on them.

The framework encourages using user-curated datasets, which are inevitably small, biased, and skewed toward those specific users. As creative practitioners play more with this new design material, their dataset becomes more skewed and biased toward their own special and unique tastes. The framework takes advantage of such curated datasets and suggests using machine learning algorithms that can be efficiently trained on a limited set of data. The model inevitably overfits the training samples. However, in this study, this is an intentional outcome, despite being a heretic in virtually any other field of machine learning.[2] Working with such small datasets allows creative practitioners to explore the generative potential of bias that resides in each user's judgments and subjective metrics.

3.2 Machine Learning Models[3]

The framework suggests using generative machine learning algorithms that can be trained efficiently and quickly on small datasets. Variational AutoEncoder (VAE) (Kingma and Welling 2013) is an optimal choice for this scenario. VAE is an architecture of generative models that uses an encoder-decoder architecture (Figure 14, left). Encoding/decoding refers to mapping, or embedding, an input data into a latent representation, usually of lower dimension, then decoding it into the same or another representation. While the input and output of the model might be of the same modality, hence the name AutoEncoder, mapping between two different modalities is also quite common, i.e., from text to image.

In a VAE, the encoder, usually denoted as $q_\theta(z|x)$, trains on the input data x to learn the features and encodes them in a latent representation space z which is usually of lower dimension, referred to as the bottleneck. The latent space itself is a distribution, usually normal distribution, represented by two vectors for means and standard deviation. To generate a sample of the latent space z, we can draw a sample from this distribution. The decoder, denoted as $p_\emptyset(x|z)$, is another network that receives the samples from z and outputs samples from the distribution of the x.

While VAEs can generate random new samples by feeding with a random z vector, for this study, the creative practitioners need to have control over what model generates. For this purpose, I opted to use Conditional VAEs, which share the same architecture as VAE plus a condition input before the decoder model (Figure 14, right).

---

[2] At the moment of writing this document, this approach is most notably utilized in NeRF (Neural Radiance Fields) for view synthesis (Mildenhall et al. 2020). A NeRF model can look at a handful of images from one single scene and learn to generate new images from the same scene, but from different directions. In this scenario, the model is trained on only one specific scene, using only a few images taken from different angles. After training, it will only work on the same scene it was trained on, but no other scene. In the case of ThirdHand, the musician is comparable to the scene, and new mezrab motions are equivalent of the new views.

[3] For an in-depth discussion on the machine learning technical details, please refer to Appendix I: Conditional Variational AutoEncoders.

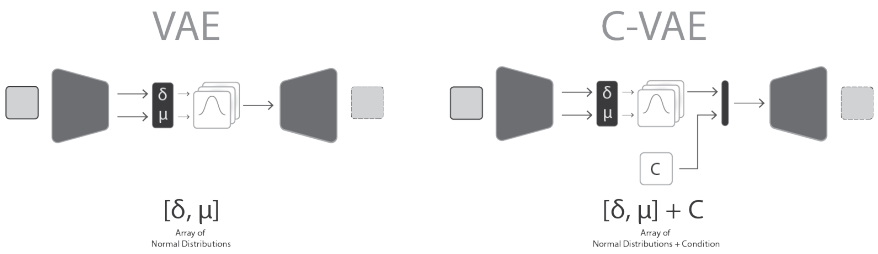


Figure 14. VAE (left), C-VAE (right) architecture.

In the training process, the encoder reduces the data dimension and embeds it in the latent space z. The decoder aims to get the latent representation and reconstruct the original input data. Inevitably, some pieces of information will be lost in the encoding process as the input is compressed into a lower dimension. When the decoder reconstructs the input from the latent representation, the outcome will not be identical to the input.

The objective of VAE is to reduce the lost information between the input and the reconstructed output while keeping the latent space distribution as close as possible to the standard normal distribution. Accordingly, the loss function for a VAE consists of two parts, 1) a reconstruction loss that observes the decoder performance in reconstructing samples, and 2) a Kullback-Leibler Divergence (KLD) that describes how close the latent space distribution is to a standard normal distribution (Figure 15).

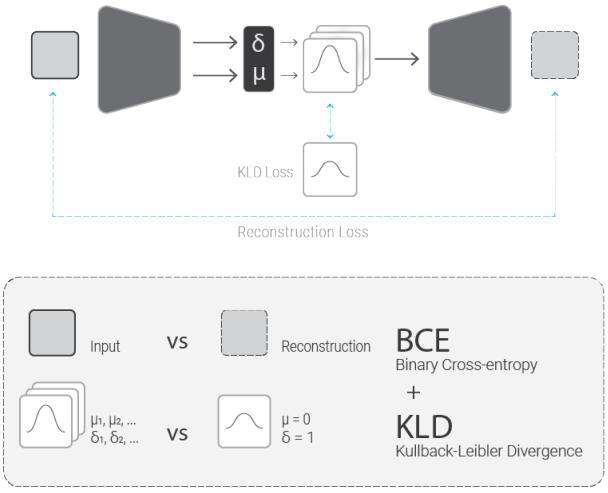


Figure 15. VAE loss function components.

We can use a C-VAE to generate new samples by feeding it with a latent vector and a deliberate condition signal. The regulated latent space of C-VAE makes it easier to create this latent vector by sampling from a multivariate standard normal distribution. The condition signal is usually a one-hot vector that can be concatenated to the latent vector before being fed to the decoder (Figure 16). The regulated latent space—sanctioned by the KLD loss—renders this approach a viable solution to get meaningful results. The KLD loss forces the encoder model to map the input samples as close as possible to a standard normal distribution with $\mu = 0$ and $\delta = 1$. As such, we are sampling the latent vector randomly from a multivariate standard normal distribution will result in generating a meaningful sample.

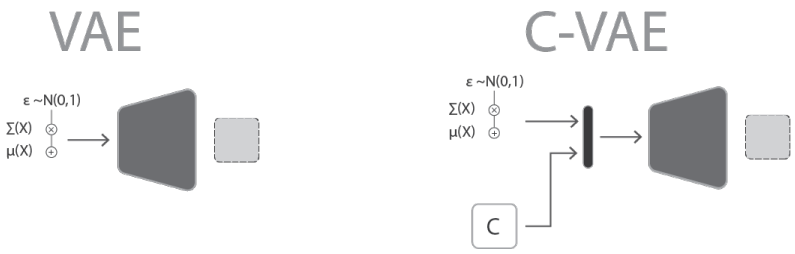


Figure 16. Drawing samples from VAE (left) and C-VAE (right).

As the data modality, size of the dataset, and purpose of case studies are different, for each of them a different C-VAE model is designed, implemented, and optimized. The specificity of each model is discussed in the corresponding chapter.

3.3 Interactive Machine Learning

While not strictly implemented in this thesis, the framework, as it is used in the SecondHand study, follows the basic guidelines of Interactive Machine Learning (iML).[4] As such, it is necessary to briefly discuss iML in this chapter.

Interactive machine learning—first introduced by Fails and Olsen (Fails and Olsen 2003)—refers to "algorithms that can interact with agents and can optimize their learning behavior through these interactions where the agents can also be human" (Holzinger 2016, 119). In iML, the training process is cast as human-computer interaction (HCI) process (Dudley and Kristensson 2018), and the computer is a part of the human design process rather than the human being in the loop of an algorithmic process (Gillies et al. 2016). The user can iteratively add new learning samples to steer the learning direction until the desired outcome is achieved. Thus, the "[i]ML workflow is inherently co-adaptive in that the user and the target model directly influence each other's behavi[o]r" (Dudley and Kristensson 2018, 8:2).

iML is particularly useful for creative applications and developing custom-made tools and shines the best when there is a user who can generate reliable training examples (Fiebrink 2019) and iteratively curate the training dataset and provide new samples until the desired results are achieved. Thus, it empowers the creative practitioner to be the primary driver of the training process and allows them to introduce their subjective measures and personal preferences into the learning process without requiring them to possess a comprehensive understanding of the underlying machine learning algorithm. While an expert ML toolmaker is required to develop the backend ML algorithm and support different parts of the ML pipeline, the training process can be handled by the creative practitioner.

Scholars pointed out some of the challenges of iML, for instance, 1) users might be inconsistent across samples that they provide and their inputs might be different from their intentions, 2) the training process can be open-ended with no clear ending criteria, and 3) interacting with ML model is not straightforward and the responses might not be clearly conceived by the users (Dudley and Kristensson 2018). While these challenges can be problematic in general machine learning applications, they mostly work in favor of this research's goals. The uncertainty and inconsistency in the user inputs are inherent characteristics of

[4] The SecondHand dashboard allowed the participants to interactively curate the dataset, retraining the learning algorithm, and evaluate the results, to achieve their desired results, from this point of view, it follows the basic principles of iML. However, in the ThirdHand, as I will discuss in detail, the almost-real-time feedback loop, which is critical in iML, could not be achieved. As such, I did not use iML in the ThirdHand study.

creative practices that I intend to acknowledge and embrace in the framework. Moreover, the open-ended nature of training would allow the creative practitioners to engage in an iterative cycle of train, feedback, correct, and finally to come up with their desired results. In such scenarios, it is the creative practitioner who will decide on the progress and success.[5]

However, the last issue, interpreting the ML algorithm behavior and comprehending the causal relationships between users' inputs and ML algorithm responses, is a particularly hard challenge to tackle. To compensate for this issue, the machine learning algorithm should complete the training process rapidly enough to allow the user to get feedback as quickly as possible (Fails and Olsen 2003).[6] Moreover, the user interface should provide the users with means to understand the behavior of the learning algorithm. For instance, real-time interactive visualizations can help users understand the effect of inputs on the model's behavior.

Fails and Olsen also refer to overfitting as another issue with iML. Methods such as cross-validation, which are commonly used to mitigate overfitting, can increase the training time. Thus, iML relies on the user to intervene and provide samples to balance the dataset and correct the training process (2003). In the case of this study, bias in the data, as well as overfitting, are both intentionally considered as part of the process.

Finally, it should be noted that iterative cycles of training in interactive learning may result in catastrophic forgetting, a condition in which the process of learning something new suddenly erases what the model has learned previously (French 1999; McCloskey and Cohen 1989). In this study, it was not necessary to implement methods to mitigate catastrophic forgetting as the datasets were relatively small with a low level of variance, and the number of iterative training cycles for each model was very limited.

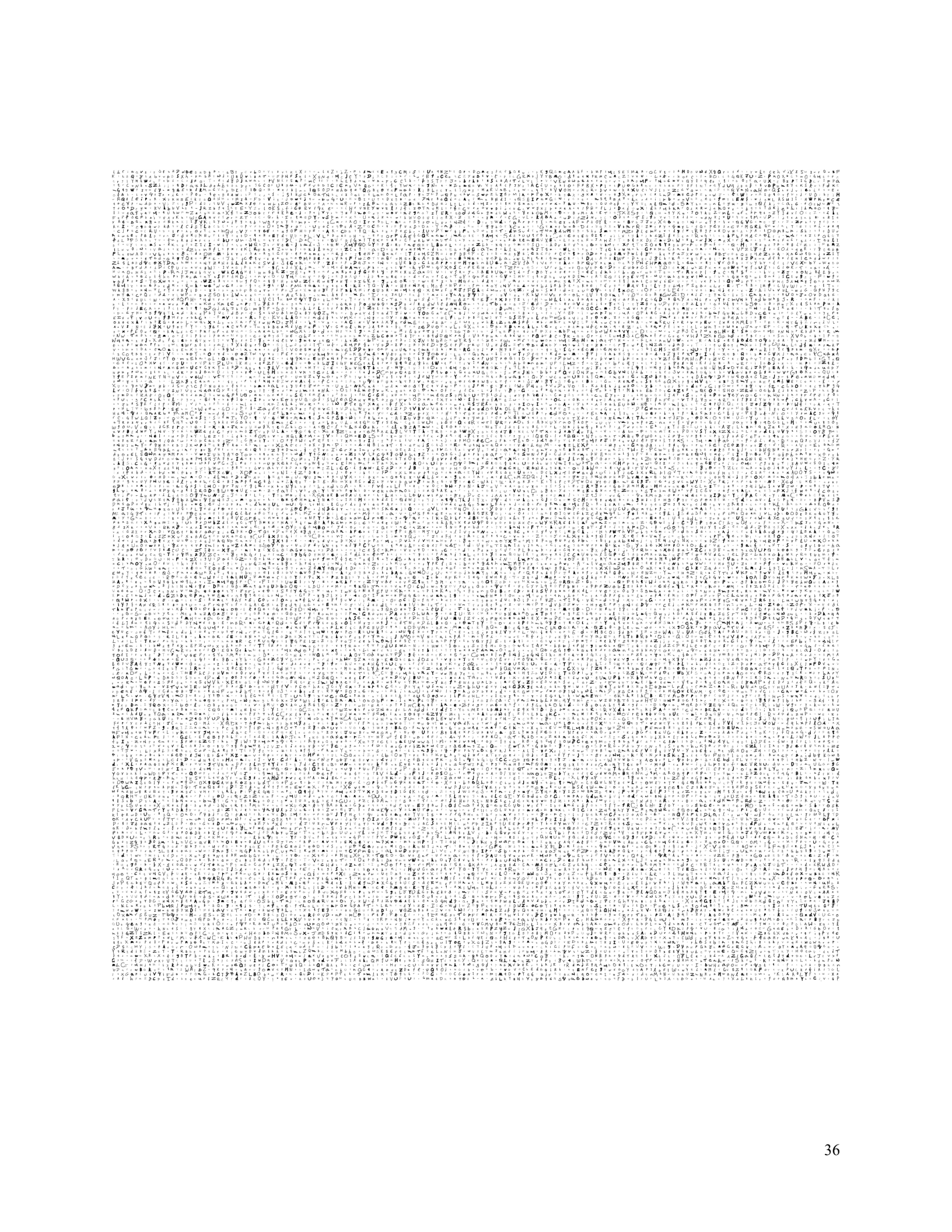
A note on the toolmaker

Throughout this research, I wore different hats at various stages of each case study. My background in computational design, toolmaking, and machine learning came into play at every turn of this study and informed my decisions. Inevitably, I constantly had to change my hats, and at some points, I wore all three simultaneously. However, to keep this document concise and clear, I will refer to myself as the "toolmaker," an umbrella term to include my computational design, toolmaking, and machine learning background. As the toolmaker, I collaborate with different creative practitioners to help them develop their machine learning-based tools without engaging with the complexities associated with machine learning algorithms. Instead, they provide the training samples, curate their desired datasets, and monitor the learning process to achieve their desired outcomes.

---

[5] Interestingly, Dudley and Kristensson specifically refer to the application of iML in craetive practices and exploratory applications, where the accuracy of the model or its performance are not the primary goal (2018).

[6] Fails and Olsen introduce a *Fast and Focus* UI principle for their iML system and argue that "[t]o be interactive the training part of the loop must take less than five seconds and generally much faster" (2003, 40).



Chapter 4. The SecondHand

During the Spring and Fall semesters of 2021, in the middle of the COVID-19 pandemic, students of my class, 48-770: *Learning Matters*, participated in a toolmaking exercise integrated with one of the technical modules of the course. In the previous modules of this course, students worked with pre-collected and pre-processed data and used quantitative measures to assess and evaluate their machine learning models' behavior. In this exercise, in contrast, they focused on creating bespoke datasets, using subjective measures, and using personal preferences to control the direction of the toolmaking process.

In this chapter, I document this study, which serves as a pilot to investigate the potential of bespoke data collection methods, interactive data curating tools, and generative models in ML-based tools for creative practices.

As the ML expert toolmaker, I designed and implemented a meta-tool that allows participants to create and curate bespoke datasets, train the backend machine learning model, and navigate the latent spaces to create a handwriting typefaces generator based on their handwriting. The meta-tool provided data collection and curation tools and the necessary interactive visual interfaces which allowed the participants to utilize data to control the model's training and manipulate the generation process.

A primary objective of this study was to examine how students interact with the meta-tool to curate their own data sets. The other objective of this study stems from the question of the accessibility of ML-based toolmaking for creative practitioners. On this topic, my goal was twofold: 1) investigating the dynamics between the ML expert toolmaker (me), creative practitioners (participants), and the meta-tool, and 2) the possibility of utilizing data as the primary interface between the creative practitioners and the ML backend of the meta-tool.

4.1 Study Framework

4.1.1 Hypothesis

This study hypothesizes that interfaces for data generation that emphasize user-generated data to integrate elements of the physical context and users' subjective preferences can reveal new potentials of generative machine learning to support creative practices.

The hypothesis addresses the two primary topics of this research, 1) making machine learning-based tools accessible to creative practitioners and 2) embracing elements of the context in the toolmaking process. The validity of this hypothesis is investigated through a toolmaking practice conducted by a group of participants using a purpose-built meta-tool. In this study, participants created their machine learning-based tool to generate handwriting typefaces. The provided meta-tool allowed them engage in an iterative cycle of a) data generation, b) data curation, 3) training, and 4) evaluation to create their tools with limited engagement with the technical details of the ML algorithm.

4.1.2 Goals and Objectives

A primary objective of this study was to examine how participants work with the meta-tool and observe the dynamics between them, toolmaker, elements from the physical context, and the underlying technology.

This study also aimed to examine the potential of data as an interface to make ML-based toolmaking more accessible to creative practitioners.[1] Participants in this study had a preliminary level of experience with computer programming and ML. However, prior to this study, they only worked with pre-processed datasets.[2] In those experiences, participants did not engage in generating, preparing, or curating any dataset. Accordingly, their only means of interacting with the machine learning models was coding in interactive Python programming environments, such as Jupyter Notebooks. Observing this shift from code to data and its effects on the users' experience is another objective of this study.

I designed the toolmaking process to allow me, as the ML expert toolmaker, and the creative practitioners to engage in a series of dynamic interactions to iteratively refine the meta-tool and make it more accessible, transparent, and understandable. Investigating these interactions is another motivation behind this study.

It is important to clarify that creating a handwriting typeface serves as a vehicle to put the research hypothesis to the test. Thus, this study's primary measure of success is beyond creating visually appealing handwriting typefaces. This study seeks to identify the potential affordances of integrating elements of the context into the machine learning-based toolmaking process. Such affordances may crystalize in various forms, including, but not limited to:

- Affordances of the machine learning-based meta-tools in allowing creative practitioners to engage elements of the physical context, such as specific tools and materials, in the toolmaking process,
- Affordances of the machine learning-based meta-tools in reflecting the users' personal context, i.e., subjective measures, personal preferences, specific personal style, and social contexts such as interactions with the toolmakers and colleagues,

---

[1] The notion of data as the interface is based on Rebecca Fiebrink's long standing works (Rebecca Fiebrink 2016).
[2] In a previous module of *Learning Matters*, students worked with *Bubble2Floor*, a generative tool for converting adjacency bubble diagrams into architectural layout plans (Veloso et al. 2022). Bubble2Floor backend was a conditional GAN architecture based on Pix2Pix model (Isola et al. 2017). In that process, the GANs model was trained on a synthetic dataset created by the instructors of the class and provided to the students as-it-is.

- Affordances of the proposed toolmaking process (meta-tool, data pipeline, machine learning model, interfaces, and user interactions) in helping creative practitioners understand the behavior of the machine learning model and align it with creative practitioners' workflow.
- Affordances of data as an effective interface between the creative practitioners and a machine learning model in the process of toolmaking,
- Affordances of collective data curation among colleagues to enrich the toolmaking process.

4.2 Relevant Work

This study relies on two well-established research fields: 1) ML-based toolmaking for creative practitioners and 2) machine learning research on handwriting recognition and generation. The former has been discussed in Chapter 2. Here, I focus on the latter to put this study in the broader landscape of machine learning research. This introduction will help the reader distinguish between this study and state-of-the-art handwriting recognition and generation.

For decades, recognizing handwritten text has been an underlying technology behind many services around us. Mail services have utilized algorithms to read the handwritten addresses on envelopes at a fascinating pace in their sorting facilities. ATMs could read your deposit checks accurately well before the current machine learning boom. This range of practical applications attracted many governmental organizations, research institutions, and research teams to allocate resources to Handwritten Text Recognition (HTR) and to develop infrastructures for such research efforts, i.e., large datasets of handwritten text. These datasets later helped to boost the data-hungry machine learning research on HTR.

One of the most popular datasets of handwritten text comes from The National Institute of Standards and Technology (NIST) databases. The widely popular dataset, dubbed MNIST,[3] is very well-known among machine learning experts who work on computer visions, machine learning, and classification systems. The MNIST dataset contains 70K handwritten digits from 0 to 9 in the format of 28x28 pixel images. It has been extensively used as a benchmark to evaluate the performance of different machine learning models. Another variant of the full NIST dataset called Extended MNIST—or EMNIST for short—gained attention among machine learning researchers. This dataset has a significantly larger sample size of around 814K, covering both letters and numbers. It is organized in two variants: 1) the first one comprises 62 classes to represent lowercase letters (26), upper-case letters (26), and digits (10). The other variant has ten classes for digits and only 37 classes for all the letters based on their visual similarities. As per NIST suggestion, the authors of EMNIST merged the samples for the letters C, I, J, K, L, M, O, P, S, U, V, W, X, Y, and Z (Figure 17 and Figure 18) (Cohen et al. 2017).[4, 5]

---

[3] The MNIST dataset was compiled from two different NIST datasets in the late 1990s by Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. It contains 70K thousand images of handwritten digits split as 60K training samples and 10K test samples. MNIST also is a very popular dataset for entry-level tutorials on generative machine learning models regarding its standard format, having a limited number of classes, and ease of training models with acceptable results.

[4] This dataset is based on NIST Special Database 19, which contains all NIST handprinted data. Over 810K samples were collected from 3600 participants as separate digit, upper and lower case, and free text fields. The data was then processed and manually checked and labeled. ("NIST Special Database 19" n.d.)

[5] There are other popular datasets such as IAM (Marti and Bunke 2002) and CVL (Kleber et al. 2013). This list is by no means exhaustive, for instance there are handwritten datasets for different languages and different styles of writing which are not discussed here.

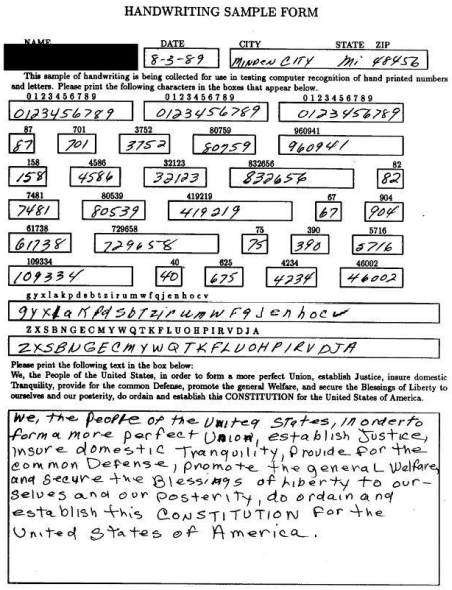


Figure 17. A handwriting sample form (HSF) from the NIST Special Database 19, source: ("NIST Special Database 19" n.d.).

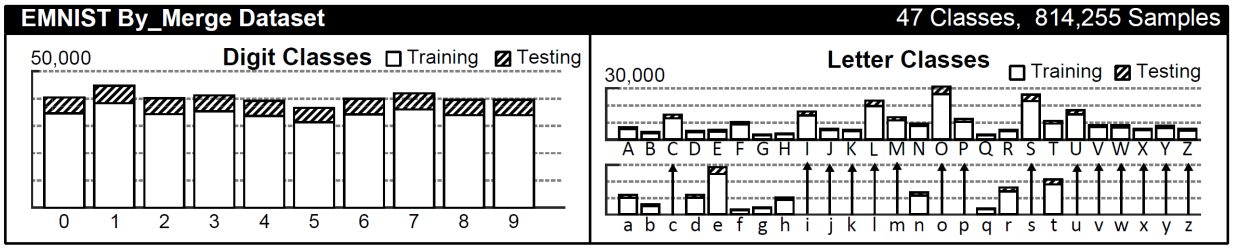


Figure 18. EMNIST by_merge dataset, notice the merger of upper- and lowercase letters for C, I, J, K, L, M, O, P, S, U, V, W, X, Y, and Z, source: (Cohen et al. 2017).

While HTR is a topic of interest for many research efforts on handwriting, some studies inquire into handwriting synthesis, which is concerned with the generation of handwriting for a given text. The motivation behind this practice might be purely technical. For instance, researchers have used the EMNSIT dataset to train a Data Augmenting GANs (DAGANs) model. This model could generate new handwritten samples for data augmentation, a method to increase the dataset size without collecting more real samples (Antoniou, Storkey, and Edwards 2018).

Meanwhile, some studies treat handwritten text synthesis as a hybrid of engineering and creativity. In 2016, Haines et al. from the University College of London published their paper "My text in your handwriting," where they introduced a model that could render a text closely resembling the style of a specific user. Their model could account for glyph selection, spacing, ligature, and text texture. They demonstrated their work by rendering quotes in the handwriting style of famous figures, such as Abraham Lincoln, Frida Kahlo, and Sir Arthur Conan Doyle (Figure 19) (Haines, Mac Aodha, and Brostow 2016).

Figure 19. Text rendered in the style of Abraham Lincoln (left) and Frida Kahlo (right) (bottom), based on the original samples of their handwriting (top), source: (Haines, Mac Aodha, and Brostow 2016).

In another study named ScrabbleGAN, researchers used a semi-supervised learning approach to generating handwritten text (Fogel et al. 2020). The model, developed based on GANs architecture, can generate words with different lengths and manipulate the writing style. GANwriting, a model developed by Kang et al. and trained on a subset of the IAM dataset, can generate handwriting samples conditioned on calligraphic style as well as the textual context (Kang et al. 2020).

The studies that are mentioned above all utilize image representation of handwriting samples. However, some researchers went one step further and studied the temporal representation of handwriting. Their models are based on machine learning architectures designed for sequential inputs, most notably Recurrent Neural Networks (RNN). For instance, Alex Grave used Long Short-term Memory (LSTM)—a variation of RNN—for handwriting synthesis (Graves 2013). Later on, Chung et al. developed a Variational RNN (VRNN) (Chung et al. 2015) and trained it on the IAM-OnDB dataset (Liwicki and Bunke 2005).[6] Their model could capture the diversity of samples while keeping a consistent style during the generation phase.

For these studies, handwriting samples should be collected in a particular format to represent time steps, which requires special hardware, i.e., a digital stylus pen. For instance, for the IAM-OnDB dataset, the researcher team utilized an eBeam interface. eBeam consisted of a special casing for a standard marker that could communicate with an infrared signal receiver installed on the corner of the whiteboard. This assembly could accurately register and record the coordinates of the tip of the marker at any point. Then this data was labeled to associate each data point with the corresponding drawing on the whiteboard.

It is worth mentioning that using additional equipment to collect temporal representations of handwriting may affect the quality of collected samples. For instance, the eBeam system adds extra weight and momentum to the marker, changing one's handwriting characteristics. However, the popularity of digital styluses and pen-like accessories for tablets has made collecting temporal data more accessible and less intrusive in recent years.

4.3 Methodology

To investigate the validity of this study's hypothesis, I designed a toolmaking process where participants could use a meta-tool to generate new handwriting typefaces based on their bespoke datasets. This study can be summarized at a high level as follows: 1) participants provide sets of handwriting samples, 2) they use these samples for training a generative machine learning model, 3) this model was used as the backend of the handwriting typeface generator tool, 4) the typeface generator tool was used to create new

[6] The IAM On-Line Handwriting Database contains 13049 samples from 221 writers (Liwicki and Bunke 2005). The model was trained on sequences of (x, y) coordinates with the pen-on/pen-up label.

handwriting typefaces. While developing their tools, participants worked with the meta-tool, engaged in discussions, provided feedback, and expressed their opinions, which were documented as video recordings and written reports.

4.3.1 Study Context

The study was designed in parallel with one of the four main modules of *Learning Matters: Exploring Artificial Intelligence in Architecture and Design*, [7] an introductory course to machine learning and creative practices with a special focus on architecture and design. Throughout the 16-week curriculum, students got familiar with the fundamental concepts of machine learning, gained hands-on experience with popular programming tools in ML, and developed their machine learning pipelines. This study was designed in tandem with the class's second module, which focused on generative machine learning models, working with user-generated data, and utilizing Conditional Variational AutoEncoders (C-VAE).[8]

4.3.2 Scope

For this case study, the scope is narrowed down, and some abstractions were implemented in four aspects: 1) field of work, 2) data modality, 3) participants' ML experience, and 4) level of details. Table 1 summarizes the scoping, simplifications, and abstractions.

When discussing the scope of this study, it is inevitable to address the context in which it was conducted. The Spring and Fall semesters of 2021 were at the peak of the COVID-19 pandemic. Students were still attending their classes remotely in accordance with the pandemic restrictions. This heavily influenced this study's design and mostly reflected itself in the data type and data collection. It was not feasible to work with any form of data requiring advanced data collection hardware. I had to design the data collection setup, required hardware, and accompanying software tools with accessibility and availability for all students in mind. I chose to work with handwriting and pixel-based representation, which was accessible to all participants; a simple digital pen, or a printer and a digital camera would suffice for this study.

Table 1. Scope of study, forces, and mitigation plans

Field	*Decision*	*Driving Factor*	*Advantage*
Skill	Opting for handwriting	Skill must be shared among all participants Data collection hardware should be available to all	All participants are familiar with this skill Only minimum hardware was required to collect data
Data modality	Opting for image-based rather than 6 DoF, temporal, or multi-modal representation	Unified data collection method for all participants Remote data collection limitations	Unified data collection using off-the-shelf and widely available tools such as tablets, digital/mobile cameras, or scanners
Prior exposure to ML	Introductory level of knowledge of ML Students from the 48-770 class was recruited	Limitation on remote moderation of study Restriction on recruiting in the pandemic era	Compliance with safety regulation Enhanced study moderation

[7] I developed and initiated 48-770 for the Spring 2021 with the generous support from the CMU School of Architecture. Since then, I have been the lead instructor of this course. During its three iterations, my colleagues in the Ph.D. in Computational Design program Manuel Rodriguez Ladron De Guevara, Jinmo Rhee, and Pedro Veloso have taught their dedicated modules. Initially titled *Learning Matters*, the second and third iterations of this course were titled as *Inquiries into Machine Learning and Design*, and *Introduction to Machine Learning and Design* respectively.

[8] This study was reviewed and approved by CMU's internal review board (IRB). Due to its integration with the curriculum of *Learning Matters*, it falls under IRB category-1 exemption.

Simplified details	Details such as half-spaces, kerning, and cursive handwriting were excluded for simplicity	Reducing the complexity of data collection Creating a unified data collection method	The letters were collected using unified charts for each letter (iteration 1) Letters were later collected in words (iteration 2)

4.3.3 Study Procedure and Timeline

The participants went through a three-step cycle (Figure 20):

1- Data generation, preparation, and curation:
   The participants were asked to create raw data samples, process them, and prepare them for the machine learning model
2- Training the generative machine learning model:
   The participants trained the machine learning model interactively through cycles of data curation, training the model, and observing the results. Thus, they steered the model's behavior by manipulating the training data instead of the machine learning architecture or its hyperparameters.
3- Generating new handwriting typefaces by drawing samples from the model.

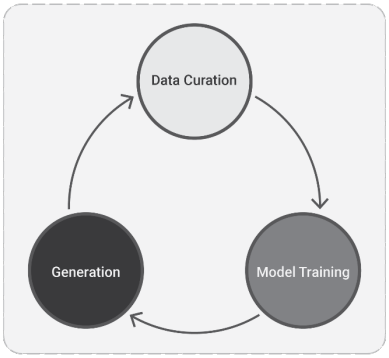


Figure 20. The iterative toolmaking cycles.

The participants repeated this cycle three rounds, as illustrated in Figure 21:

Round one: This round was designed to familiarize the participants with the process from providing handwriting samples to generating new typefaces,

Round two: The goal of this round was to allow participants to retrospectively review the first round and improve their workflow based on the experience gained during the first round,

Round three: Participants were instructed to share their samples on a shared database and engage in a collective data curation activity. Each participant could use this database to curate their bespoke dataset, train the model, and generate a handwriting typeface.

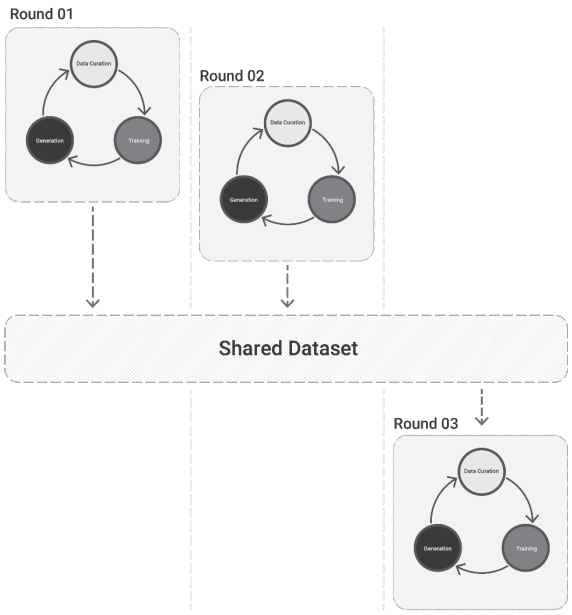


Figure 21. Three-step toolmaking process.

During the span of this study, participants were 1) introduced to the main concepts, tools, and platforms that were used in the study, 2) provided hands-on training to use them properly, 3) made their handwriting typeface generator tool, and eventually 4) presented their work.

4.4 The SecondHand Meta-Tool

Meta-tool collectively refers to the set of tools I developed to allow the participants to 1) collect, process, and curate data, 2) train the machine learning algorithm, and finally, 3) navigate its latent space to generate new handwriting typefaces. I iteratively refined and improved the SecondHand meta-tool to incorporate feedbacks from the participants. Moreover, the participants occasionally modified and adapted parts to match their specific workflow and requirements.

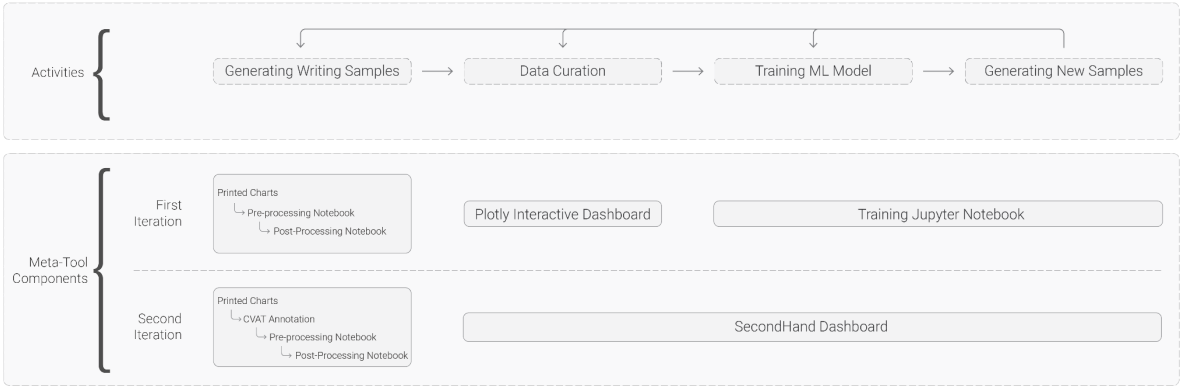


Figure 22. Activities and their corresponding components in the meta-tool implementation during the first and second iterations of the study.

While in the first iteration of this study, the meta-tool was organized as a series of Jupyter notebooks running on the Google CoLab (Google, n.d.)service as well as an early version of the interactive dashboard for data curation. This resulted in various issues regarding the interoperability of various meta-tool components. However, for the second iteration, I implemented the meta-tool as a web-based interactive dashboard, named SecondHand Dashboard, using Dash Plotly (Plotly 2022), except the data collection process that still relied on external tools such as CVAT (Figure 22).

4.5 Meta-Tool Components

The meta-tool consists of three main components: 1) data pipeline, 2) machine learning algorithm, and 3) generating interface. Figure 23 shows the three components as implemented in the SecondHand Dashboard.

4.5.1 Data Pipeline

The data pipeline was designed to support a unified workflow that all participants could reproduce at home using simple and accessible tools, such as a printer, a scanner, or a cell phone camera. The pipeline was also flexible enough to accommodate more advanced pieces of hardware, i.e., tablets, styluses, and digitizers.

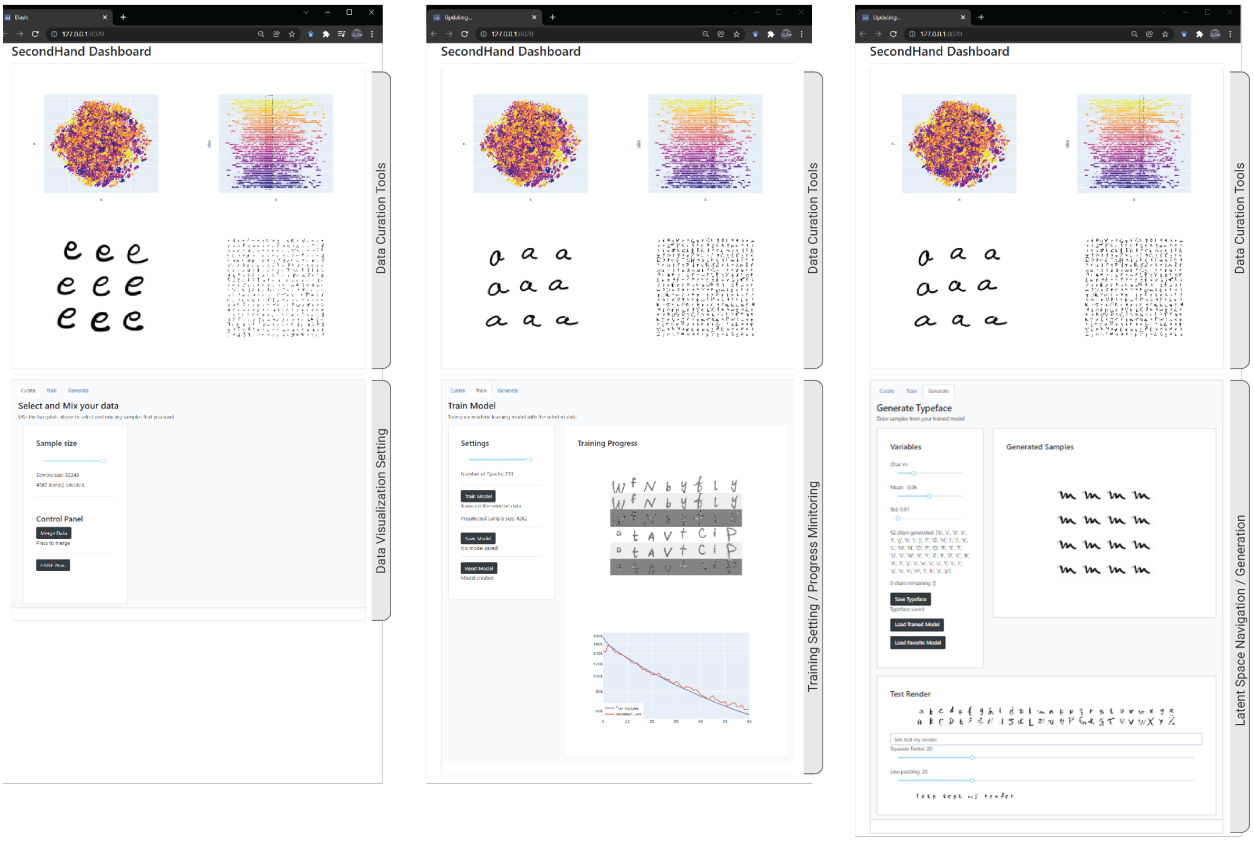


Figure 23. The integrated dashboard, as used in the second iteration of this study in Fall 2021, with data curation, training, and generation tools integrated into one platform.

As discussed earlier in this chapter, there are two approaches to handwriting representation: 1) temporal and 2) pixel based. The pixel-based approach has several advantages over the other one: 1) the data collection process is significantly easier as it only requires image representation of data instead of a sequence of steps, and 2) pixel-based data can be used with Convolutional Neural Networks (CNNs) while temporal data is more suited with recurrent machine learning models. The training process for non-recurrent models is usually more efficient,[9] as the samples can be generated at once, and the problems of learning long-range dependencies are avoided (Kang et al. 2020).

Based on these advantages, I decided to use pixel-based representation and avoid temporal representation. This decision resulted in a simplified data collection pipeline to accommodate the requirements of this study. It made it possible to utilize a CNN-based machine learning model with very efficient training cycles.

In the first iteration of this study, participants were provided with an 11-page set of grids with 36 cells for each letter of the English alphabet, totaling 1872 data entries. Participants could fill out the charts using a desired writing tool—pen, pencil, marker—and digitalize them using a scanner or a digital camera (Figure 24, left). Participants had access to a data pre-processing CoLab notebook with the necessary functions and a detailed video demonstration to facilitate this process. Alternatively, participants could use a digital medium with stylus support to directly generate the data in digital format. In both cases, participants could use the same CoLab notebook to slice the input images, extract each letter, remove the boundaries, and format the results as NumPy arrays with pre-defined shapes and dimensions.

Participants' feedback from the first iteration of the study indicated that collecting handwriting samples using isolated cells negatively impacted their handwriting. Accordingly, I implemented a different data collection method to address this issue in the second round. This time, I asked the participants to write letters in words instead of writing each letter separately. One thousand words were selected and organized into 20 pages. Half of the words were written in capital letters, and the other half in lowercase. Participants could write each word in the blank space below each printed word (Figure 24, right). While closer to a typical writing setup, this method imposed different challenges. For instance, defining the boundary of each letter should be done manually;[10] once the writing was over, participants had to use an online computer vision annotation tool to annotate the boundary of each letter. [11]

[9] Recurrent neural networks are usually slower to train as they cannot take advantage of parallelism as much as convolutional neural network can.

[10] During the development of this phase, I tested various automated segmentation methods. However, the accuracy of these methods was not satisfactory and could add more challenge in the labeling process.

[11] For this part Computer Vision Annotation Tool (CVAT), an online free platform was used ("CVAT" n.d.).

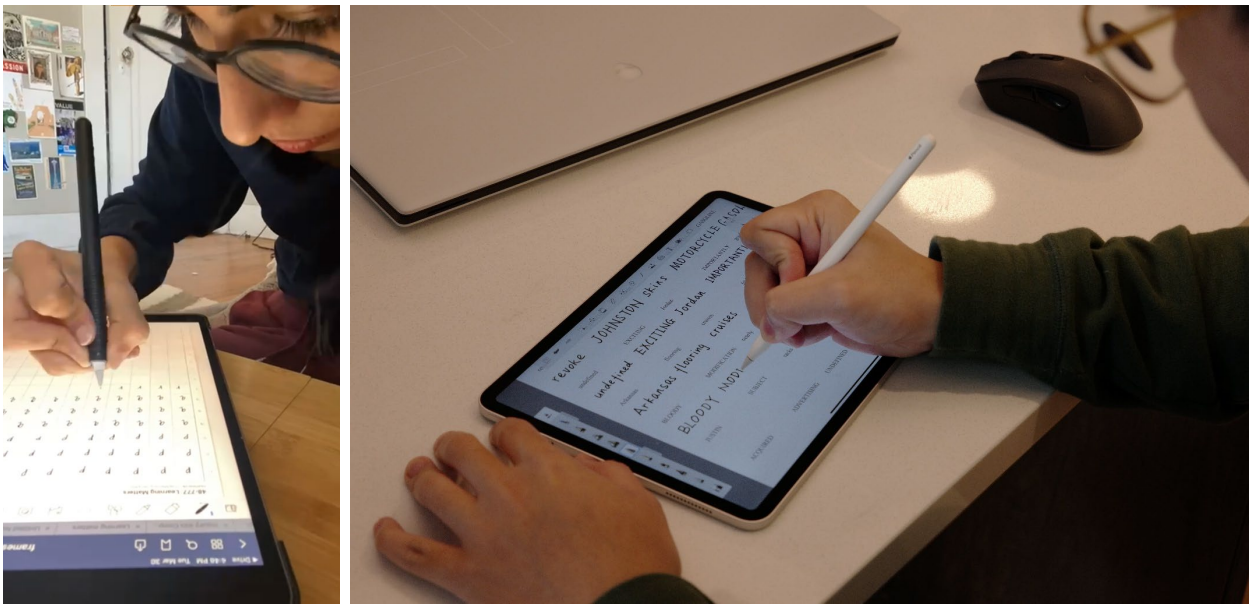


Figure 24. Collecting samples using a digital pen and tablet: iteration one in separate cells (left), and iteration two, in words (right). Images courtesy of 48-770 students, reproduced here with permission.

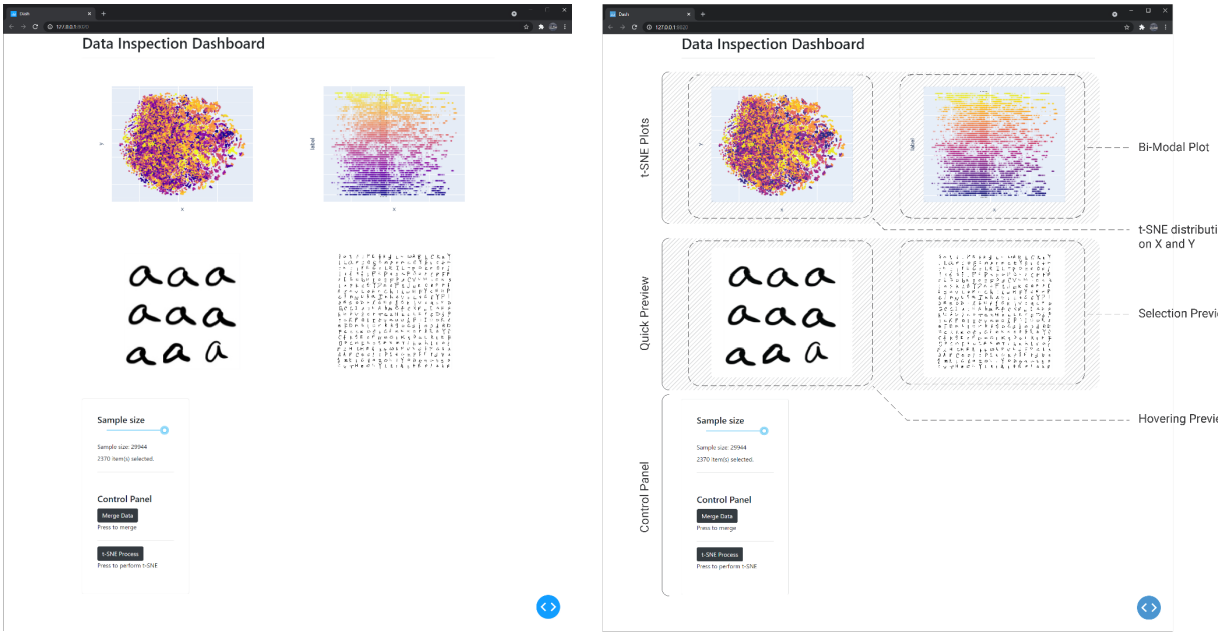


Figure 25. The data dashboard (left), UI elements (right).

4.5.2 Data Visualization/Curation Dashboard:[12]

A series of interactive data visualizations were the core elements of the meta-tool's data curation. These visualizations proved to be essential tools for the collective data curation process, where participants had to curate their training dataset from over 28000 shared samples which were created by other participants (Figure 25).

To create these plots, the data samples were processed by the t-SNE algorithm (Van der Maaten and Hinton 2008) to reduce their dimensions from 64×64 to only two.[13] Mapping a dataset of high-dimension to a very low-dimension space is a common practice in data visualization. When used effectively, it helps users comprehend data distribution, hidden patterns, and relationships between the samples, which were otherwise hard to recognize.[14] The resulting two-dimensional mapping was a distribution of samples based on their visual features. A color scheme representing the label of each sample—i.e., a, b, X, Q, …—was also applied to each plot. Users could hover over the two scatter plots to visually inspect them one by one or review them in bulk using the selection tools.

The first plot—positioned on the top left side of the dashboard—visualizes the t-SNE distribution of samples based on their visual features (Figure 25, left). This plot made it easier for the participants to inspect the visual trends in the dataset and get more familiar with the sample space.

The second plot—positioned on the top right side of the dashboard—combines two data modalities: 1) one of the two elements of the t-SNE manifold on the X-axis, and 2) labels on the Y-axis (Figure 25, left). This bi-modal representation helps participants conveniently explore the dataset based on visual features and labels. The benefits of this plot were later crystallized in the study. Several participants used it to select a subsection of the whole dataset while controlling the number of each label in the set.

Hovering the mouse over a point in each of the two plots triggers a quick preview function. This function shows the hovered sample and the four samples before and after it in the dataset. Participants can also use click-drag gestures on both plots to select a subset of the dataset, using rectangle or lasso-style selection in real time and see the selected samples. These two options allow the participants to rapidly check any region of the t-SNE distribution and add/remove samples to curate the desired dataset.

4.5.3 Machine Learning Backend[15]

The Conditional Variational AutoEncoder used in this study follows the basic architecture of VAEs, an encoder, a variational sampling, and a decoder chained one after the other. The C-VAE architecture can disentangle the visual features of the input data from the label. It means that with the same label input, it is possible to manipulate the latent vector and generate different samples from the same label but with

---

[12] This dashboard as described here was only used in the first iteration of this study. In the second iteration, this dashboard, the training interface, and sample generation tools were all integrated in one unified dashboard that I will discuss in more details in section 4.6.3, Integrated Dashboard.

[13] In this study, I opted to use an implementation of t-SNE, called Open t-SNE (Poličar 2020). All samples were first converted to a vector of 1×4096 shape and then fed to the t-SNE algorithm to get mapped into a two-dimensional space as 1×2 vectors.

[14] It is worth mentioning that this method has also been used by several artists and researchers to visualize art collections, for a few interesting samples please visit project Pix Plot with a Uniform Manifold Approximation and Projection (UMAP) backend, (Duhaime 2017) and t-SNE Map that utilizes t-SNE algorithm (Diagne, Barradeau, and Doury 2018).

[15] For more information on the ML backend algorithm, please refer to please refer to Appendix I: Conditional Variational AutoEncoders and 3.2 Machine Learning Models sections.

different visual features. Similarly, with a fixed latent vector, it is possible to feed different label vectors to generate various glyphs with similar visual features.

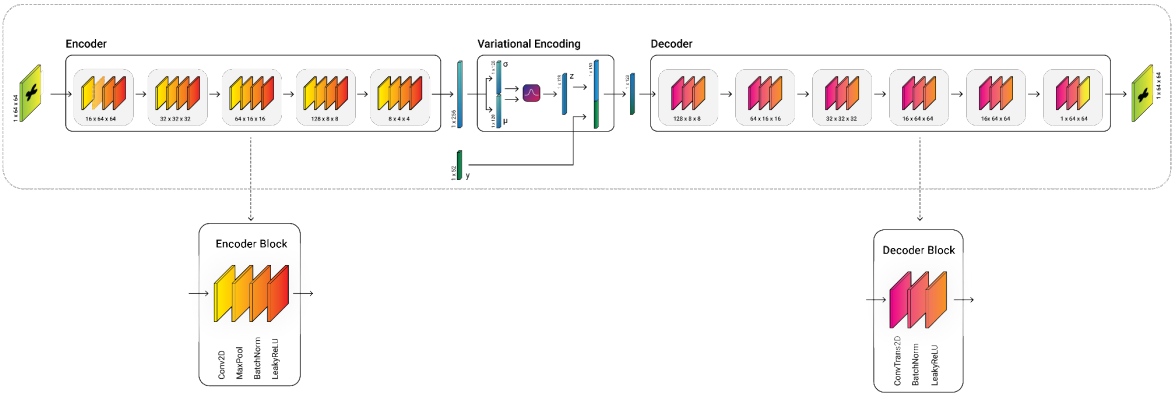


Figure 26. Architecture of the conditional variational autoencoder (top), the encoder block (bottom left), and the decoder block (bottom right), minor modification is applied.

The encoder and the decoder models use a cascade of modular blocks (Figure 26, top). The encoder blocks consist of convolutional 2D layers, followed by max-pooling, batch normalization, and leaky ReLU activation function (Figure 26, bottom left). In the decoder model, the block uses a convolutional transpose to sample up the input, followed by batch normalization and leaky ReLU (Figure 26, bottom right). The only exceptions are the first module of the encoder—which skips the max pooling—and the last module of the decoder—which substitutes the leaky ReLU with sigmoid to keep the results in the 0.0 to 1.0 range. The model's architecture was fixed for all the participants, and a set of suggested hyperparameters were available to the participants (Table 2). However, the participants had the option to adjust a few training parameters, most notably the number of training epochs.

Table 2. Suggested training parameters.

Variable	*Default value*
Batch size	64
Number of epochs	250
Latent dimension	128
Learning rate	$1e-3$
Validation to train ratio	0.1

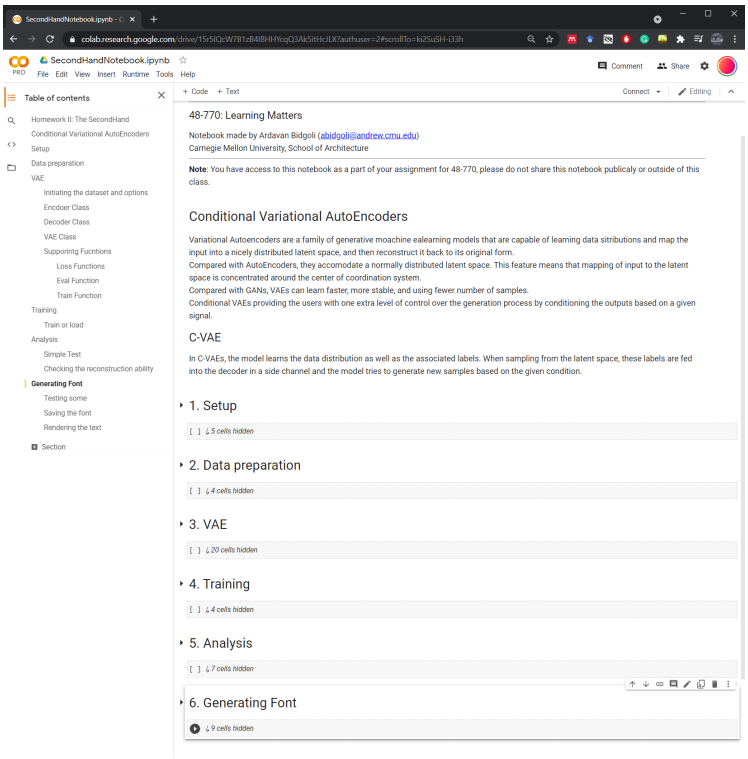


Figure 27. Training/generating notebook, the notebook structure with comments and notes.

Training

In the first iteration of the study, the training process was implemented as a Jupyter Notebook running on Google's CoLab server. The notebook contained the Python codes and supporting materials such as descriptions and guides. It was organized into six sections: 1) introduction, 2) code setup, 3) the C-VAE model, 4) training workflow, 5) analysis, and 6) generating typeface (Figure 27). Participants could load the data files from the pre-processing notebook or the data curation dashboard to this notebook, then train the model and monitor its progress through visualizations. During the next iterations of this study, the Jupyter notebook was revised and updated based on participants' feedback and used in combination with the dashboard.

The training notebook was designed to provide high-level control using hyperparameters and a series of visualizations to help with qualitative observations and quantitative evaluations. Most notably, it included a three-row plot to monitor the model's performance during the training process (Figure 28). The first row of the plot shows a random set of samples from the validation dataset. The second row shows the same set of samples passing through the model, encoded to the latent space, and then decoded to get reconstructed. The third row shows the differences between the first and second rows; yellow regions indicate the most similarity, while red spots highlight the largest discrepancies. This combination of plots visualizes the model's performance intuitively. Participants can make a qualitative evaluation based on the fuzziness of results on the second row and the smaller regions covered with red spots in the third row. Sharper results in the second row and fewer red regions in the third row were indicative of a better performance. The combination of the three-row plot and the classic loss-per-epoch plot provides the two means of supervising the training process.

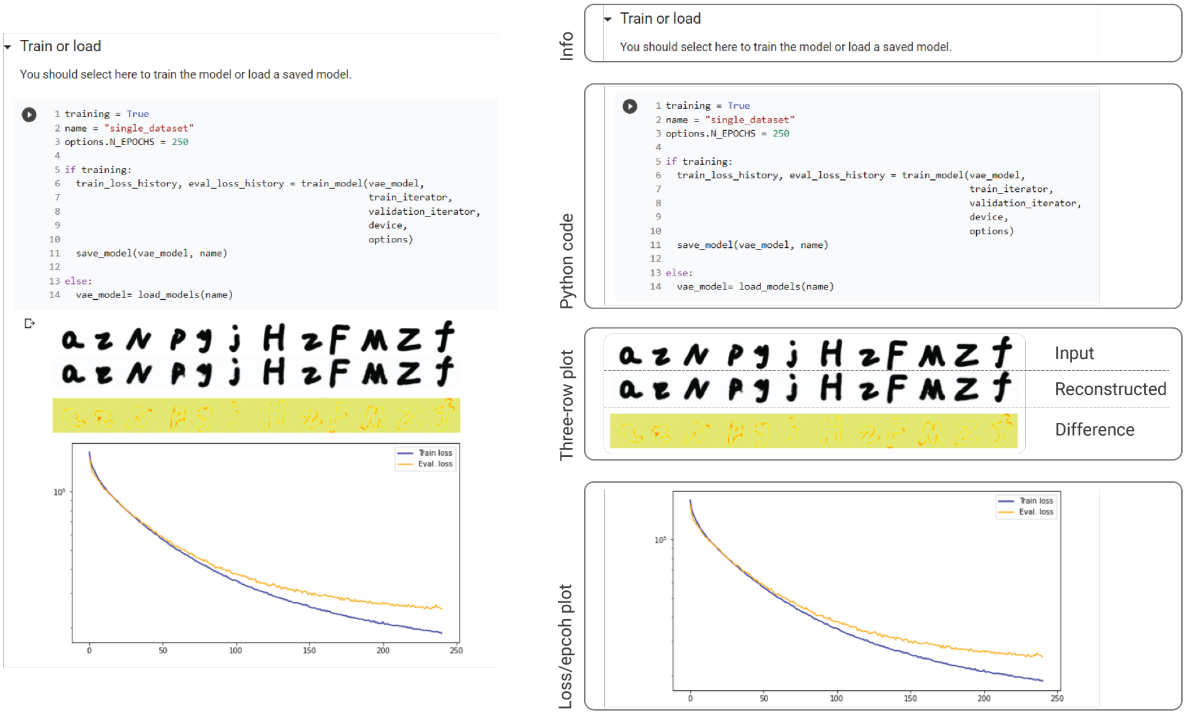


Figure 28.Training interface as it appeared on the Jupyter notebook (left), interface details (right), minor edits applied.

In the second iteration, I implemented the training process as a tab in the integrated dashboard. The user interface provided participants with the same three-row plot and a simplified interface to adjust training parameters and monitor the process (Figure 29, left). In this setting, the model runs on the participants' local devices rather than cloud servers. Accordingly, each participant's experience could vary based on their hardware.

Integrating the training process with the dashboard helped the participants interactively curate the training data set. By design, the data curation visualizations were always visible on the top half of the dashboard, even while working with the Train tab (Figure 29, right). In this workflow, participants could select a dataset, train the model for an arbitrary number of epochs, evaluate the model's performance, revisit data selection, modify it, and continue training for more epochs.

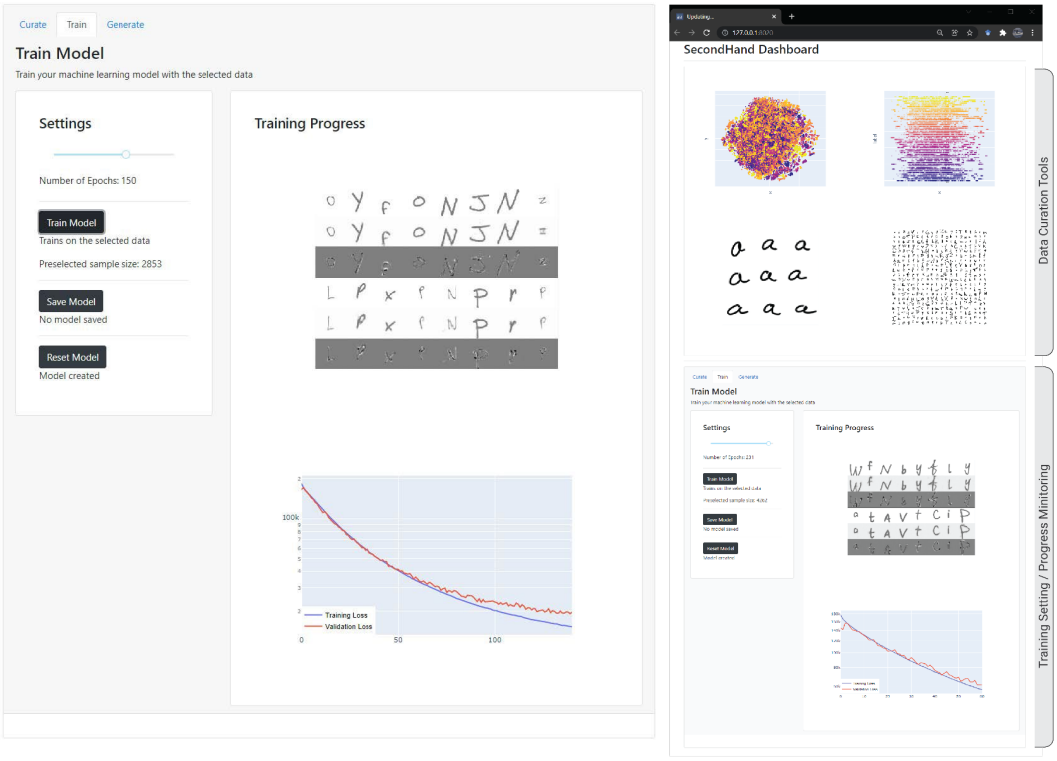


Figure 29. Training tools in the integrated dashboard.

4.5.4 Latent Space Navigation and Sampling

The last step in the iterative toolmaking cycle is navigating the latent space of the trained model and generating new samples to create a handwriting typeface. Participants could use interactive widgets to draw samples from the model's latent space and generate new samples for each letter. Sampling from the latent space was implemented in two methods: 1) generating each glyph with refined control over the sampling distribution, and 2) generating the whole 52 glyphs of the alphabet at once—without any fine-tuning (Figure 30, top).

I iteratively revised and improved the sampling process to make it more intuitive, more understandable, and easier to navigate for the participants. For instance, in the early version—v.1 from now—the primary sampling method was complicated, and the interface was not communicating enough with the user. The over-simplified method to rapidly generate the "whole alphabet" with one click was intended to allow the participants to quickly generate the whole 52 glyphs in the alphabet and then come back and fine-tune each glyph individually using the other tools in the notebook.

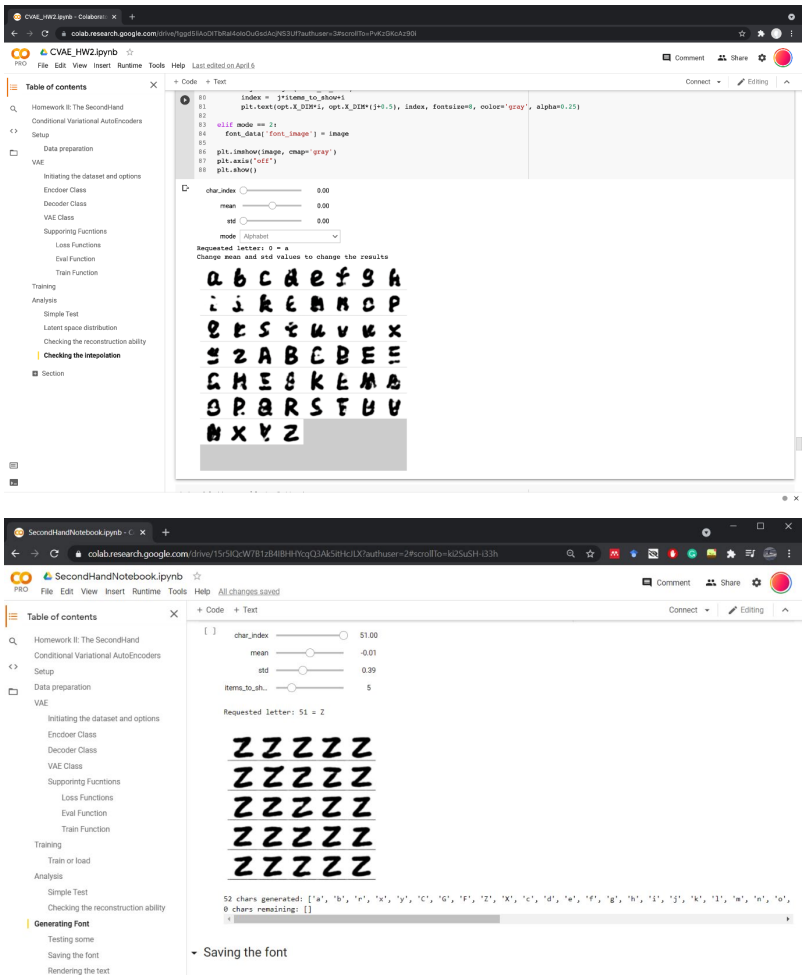


Figure 30. The v.1 sampling interface with the "whole alphabet" method, note the poor quality of samples (top). Interactive sampling widgets in v.3, note the slight variations between the samples (bottom).

In the revised notebook—v.3 from here after—I replaced this sampling method with a more efficient implementation[16] accompanied by two series of visualization: 1) showing an array of generated samples to let the user inspect them in real-time, and 2) a list of generated glyphs to demonstrate the progress toward all 52 glyphs (Figure 30, bottom). The four sliders were responsible for selecting a specific glyph to work on, adjusting the mean and standard deviation of sampling distributions, and the number of samples to be generated for each glyph. To encourage the participants to play with the sampling parameters—instead of leaving the setting on default values and skipping through each glyph—the sampling parameters were intentionally defaulted to produce borderline results. The results submitted after these changes were noticeably improved compared to those created previously.

[16] The main change was the modification that were applied to the random distribution behind the sampling method. This resulted in generating visually more appealing samples while requiring fewer adjustments in the sampling parameters.

The last variable was designed to save a larger pool of samples for each glyph with the desired level of variations between them. This pool of samples for each glyph helped create a more natural text rendering at the end, similar to human handwriting; each glyph instance follows the same style but with slightly different details. In the last iteration of the study, this sampling method was integrated with the dashboard and let participants quickly generate the typeface and render sample text with it (Figure 31).

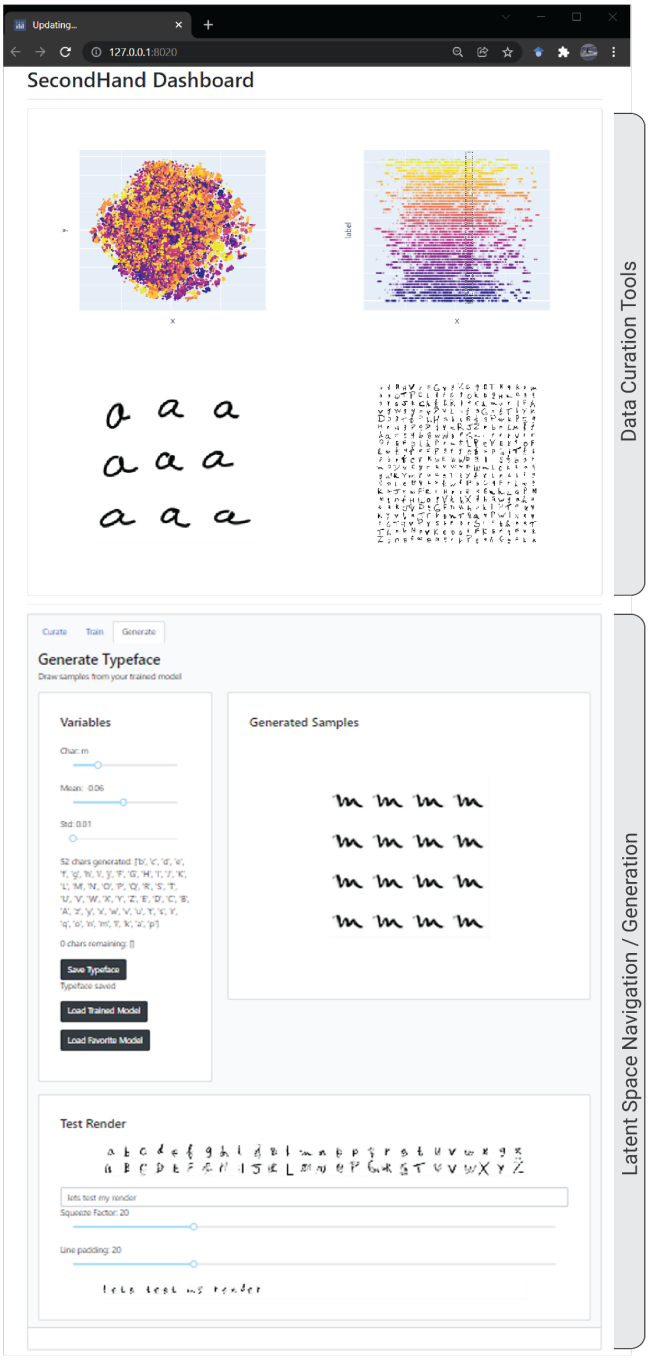


Figure 31. Sampling methods as unified in the integrated dashboard

Eventually, participants could use the interactive rendering widgets to convert an input text to a handwriting form while adjusting the spacing between the letters and lines (Figure 32). The generated typeface is saved as a series of 2D images in NumPy array of dimension $52 \times n \times size \times size$ were 52 reflects the number of glyphs in the typeface, n is the number of samples for each glyph, and $size$ defines the number of pixels in each glyph width and height. The final NumPy array is then saved from being used later.

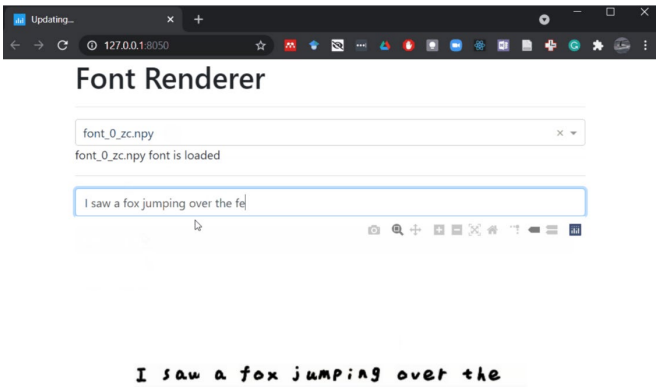


Figure 32. The real-time text to handwriting toolkit.

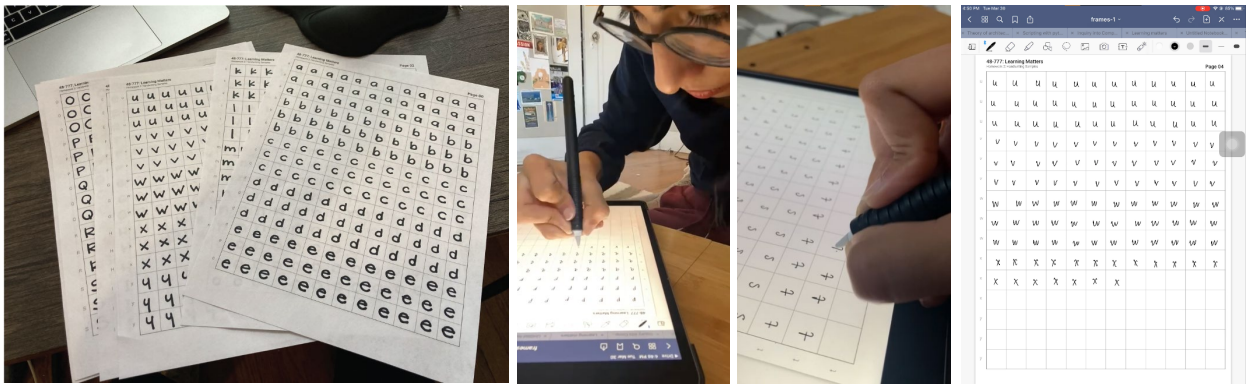


Figure 33. Charts filled using hand and Sharpie marker (left) and digital pen (right). Images courtesy of Learning Matters students, reproduced here with permission.

4.6 Study Report

This section is a comprehensive report on the two iterations of this study, which were conducted in the Spring and Fall semesters of 2021. It is based on my firsthand observations, conversations with the participants, and review of their submitted materials. During this study, the participants had several opportunities to share their thoughts and views on the process, their experiences, and the results. Discussions during class time, work sessions, office hours, and presentations were the main means of communication. Moreover, at the end of this study, participants submitted a brief report and an essay to reflect on their experience with the data collection process, using data as an interface, interacting with a generative model, and navigating its latent space using interactive tools.

4.6.1 Data Collection Process

The first step in this study was data collection, which proceeded with data preparation. One of the earliest decisions for each participant was to choose a platform for data collection. Participants could use pen and paper or any digital pen/stylus setup to generate their input data. Only three participants opted to use physical pen and paper, while the rest used different digital devices combined with a handful of software packages to post-process the samples (Table 3).

Table 3. Mediums of data collection[17]

Format	*Hardware*	*Software*	*Number of participants using (first/second iterations)*
Pen and paper	Sharpies, pen	-	2/1
Digital tablets	Apple iPad	GoodNotes and others	4/3
	Wacom tablets	Adobe Photoshop	2/0
	Other digitizers	Adobe Acrobat	1/0

Using digital pens helped the participant have a more flexible workflow, allowing them to zoom, rotate, and edit the written samples. The choice between pen and paper and digital tools was not merely associated with hardware availability. Interestingly, one participant found using a digital pen unnatural after testing it briefly and decided to use a physical pen and paper. They stated that using pen and paper results in messier samples and a less forgiving process, but it offers the most realistic examples of one's handwriting.

Some participants were curious to know the outcomes had they used different media. There were indications about the influence of the data collection medium on the quality of the samples. For instance, participant #8 mentioned that the different tiers of Adobe Acrobat software register and handle the edges noticeably differently, which was inconsistent with their original handwriting.[18] Additionally, a few participants mentioned that using a digital medium helped them quickly update their samples and replace the undesired ones. This process could be more challenging with a traditional pen and paper or marker and paper.

In the first iteration, the data collection procedure required the participants to write down around 1900 individual letters, which equals two-thirds of a letter-sized page (Figure 33, Figure 34, Figure 35). Some participants considered this phase "straightforward" and "fun." In contrast, some participants found this process more time-consuming than initially expected and sometimes "monotonous." Participant #8 states that this specific data collection approach is "unnatural" and "time-consuming."

In the second iteration, where letters were organized in 1000 words rather than individual cells, participants could "… provide more natural inconsistencies"[19] to the dataset while reflecting the participant's "writing habits"[20] (Figure 36). However, they find the process tedious and painstaking. The annotation was also time-consuming and prone to subtle mistakes that could trigger serious challenges down the road (Figure 37). Interestingly, participants managed to find impromptu solutions for these

[17] For more information about the software packages and pieces of hardware, please refer to developers' or manufacturers' websites (Apple 2022), (Wacom 2022), (GoodNotes 2022), (Adobe 2022b), (Adobe 2022a).
[18] Participant #8, reflection paper.
[19] Participant #10, reflection paper.
[20] Participant #12, reflection paper.

issues—notably, developing Python code to enhance pre-processing stage, manually editing labels, using Photoshop to correct their handwriting errors— and then communicated this solution among their fellow participants. Some of these solutions also helped me enhance the meta-tool and allowed me to fine-tune it for the next iteration.

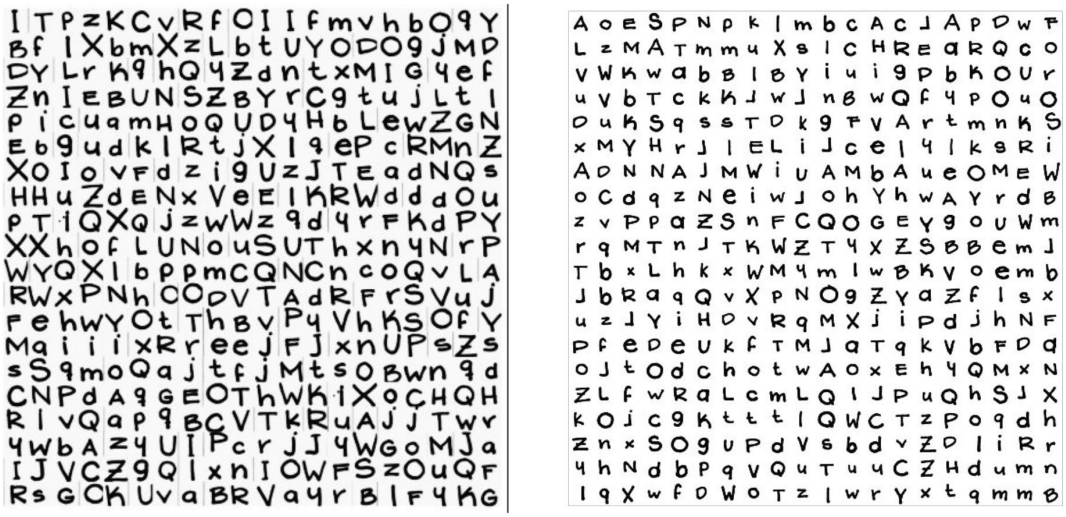


Figure 34. Samples of handwritten letters from the first iteration of study, after preprocessing. Note that the grid lines were also captured around many of the glyphs in round one (left) while in the second round this issue was corrected by the participant (right), images from 48-770 students, reproduced here with permission.

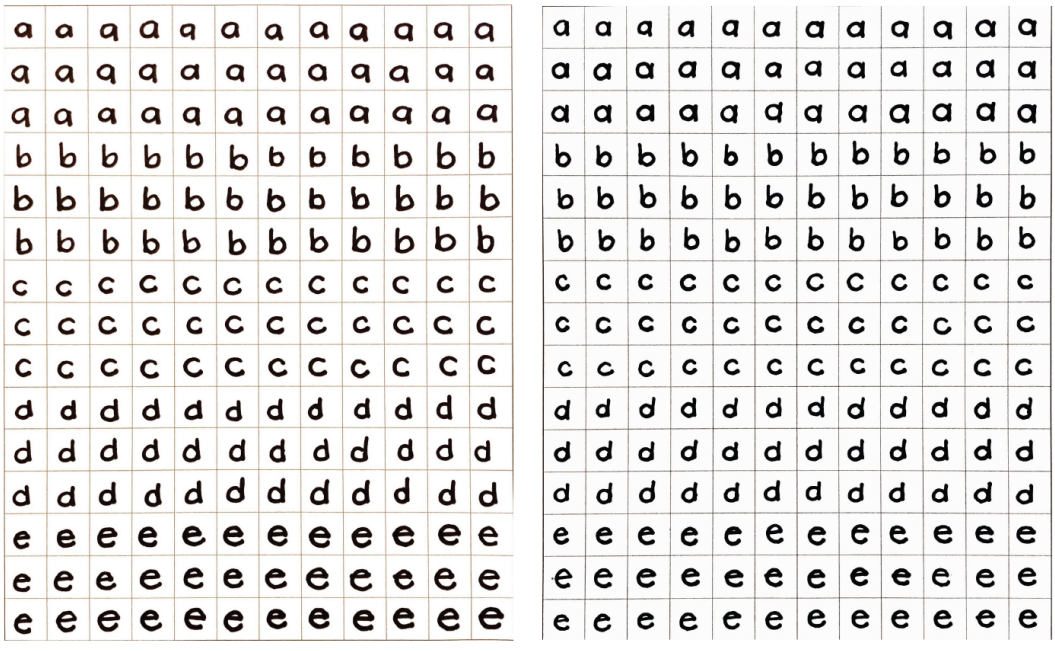


Figure 35. Samples from the first iteration of study, after the initial preparations. The first round (left) contains letters that were not correctly aligned with the grid, this later produced undesirable artifacts in the results. In the second round (right) letters were organized and aligned more consistently. Images courtesy of participants, reproduced here with permission.

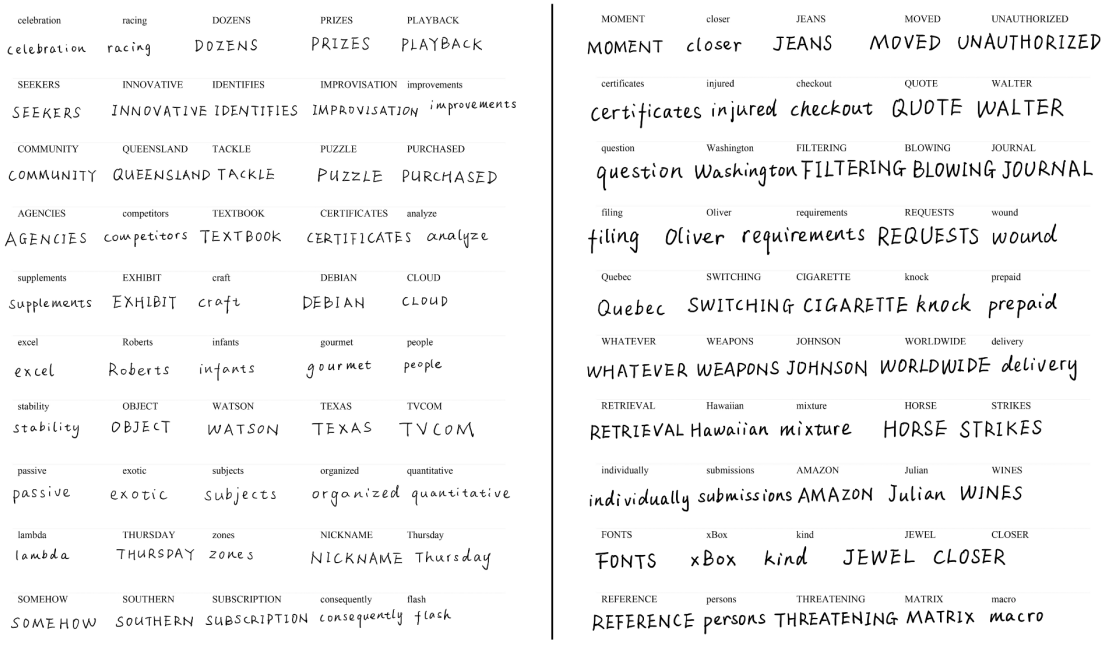


Figure 36. Sample collection in words for the second iteration, note that the samples are written in words rather than isolated letters. The left page is from the first round of data collection, while the page on the right is from the second round. Note the difference in spacing and thicknesses, images from 48-770 students, reproduced here with permission.

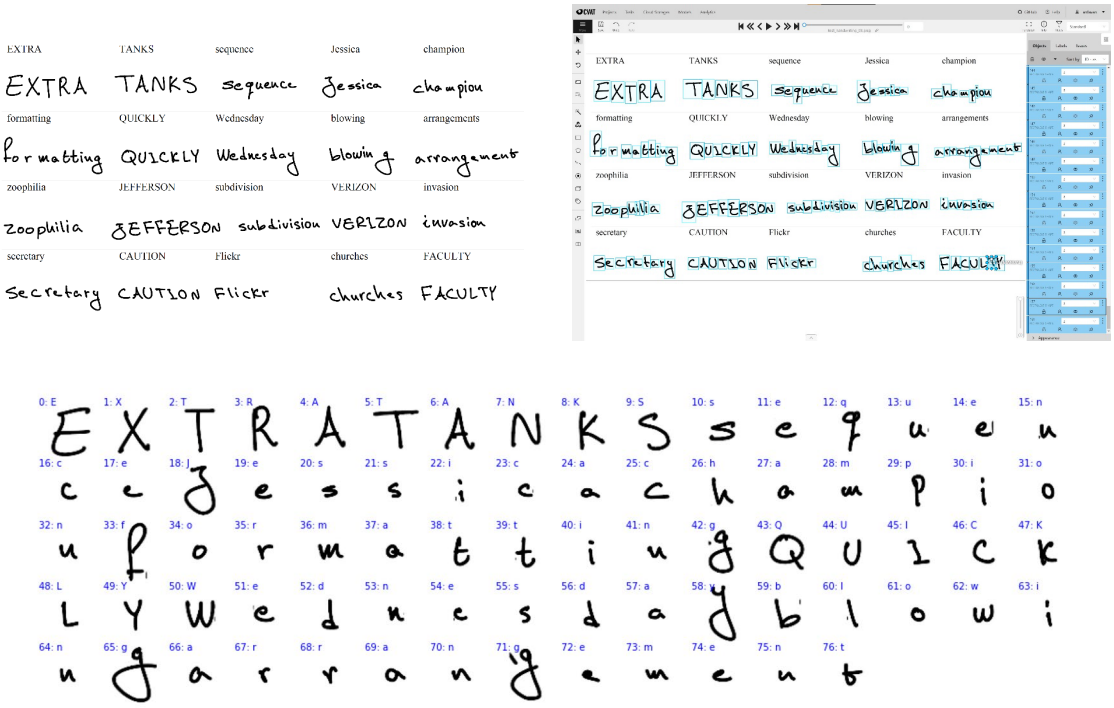


Figure 37. The segmentation and annotation process, writing the words (top-left), annotating each letter in CVAT (top-right), using the Python code to extract each letter as a fixed-size image (bottom).

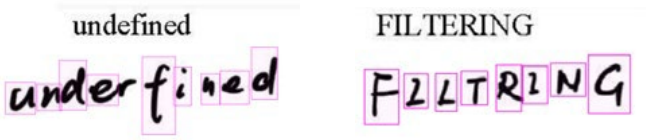


Figure 38. Samples of common, but hard-to-catch mistakes, images from 48-770 students, reproduced here with permission.

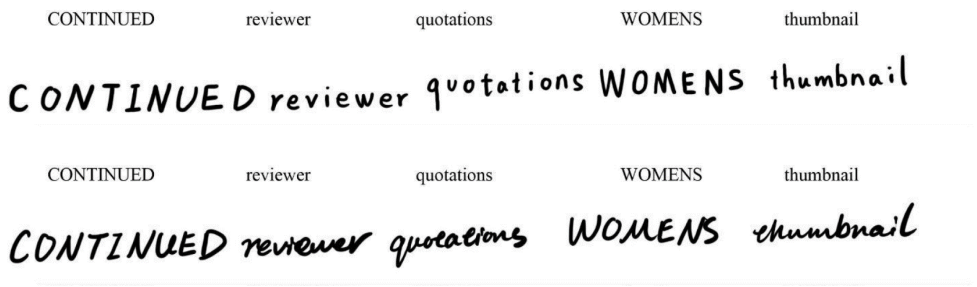


Figure 39. Sample of original handwriting of a participant (top), samples of the same participant's handwriting adapted to keep letters separated, image from 48-770 students, reproduced here with permission.

Although the revised method felt more natural during the writing stage, some participants highlighted new issues. Most importantly, the tedious annotation and segmentation process to find and mark each letter individually. Spotting and addressing issues in the samples—common mistakes such as a missing letter or a typo (Figure 38)—became too challenging for the participants.

All participants reported prolonged time spent on the first ten pages. Moreover, collecting the exact same number of letters using meaningful words is impossible. The frequency of some letters, i.e., e and a, is significantly larger than other letters, such as q and x. To compensate for this issue, a larger set of samples were collected, then the surplus samples were discarded while certain samples were duplicated to match the intended dataset size. For each letter with more than 36 samples, only 36 of them were randomly selected. But, for the letter with fewer than 36 samples available, the existing ones were duplicated to match the number of other letters.

Another issue was the impact of the segmentation method on the participants' handwriting. They had to modify their routine handwriting to accommodate the segmentation method. For instance, each letter had to be clearly spaced from the adjacent ones (Figure 39). Also, participants who routinely write letters slightly inclined had to straighten up their handwriting to let the letters fit inside the straight rectangles with no overlap.

While discussing the data collection process with the participants, the most interesting discussions were probably formed around the relationship between the participants and the data. The process helped one participant to "…identify the relationship between each step [and] the influencing factors that potentially affect the results."[21] There were several positive reflections on the bonds between the participants and their datasets. For instance, some expressed their satisfaction with the level of familiarity with the data.

[21] Participant #13, reflection paper.

This satisfaction was partially derived from the level of control they had over the quality of samples, which could be out of their control when using datasets from external sources.

The reflection papers also discussed the difference between a bespoke data collection approach and an off-the-shelf one. On the one hand, participants refer to the convenience of having a ready-to-use large and diverse dataset as an advantage of off-the-shelf datasets over a bespoke data pipeline. On the other hand, a common concern over the off-the-shelf approach was the lack of control over data quality and losing the agency over what was being fed to the learning model. For instance, participant #2 states that through the bespoke data collection process, "… I could control the quality of the dataset I was going to build … I would be very familiar with the dataset."[22] Another participant mentioned that the user-generated data "…give[s] a lot of agenc[ies] in decision-making for the one making the dataset," while off-the-shelf datasets provide more variations with the tradeoff of less control.

The other major area of discussion was devoted to different notions and understandings that participants gained through this process. Some participants found this bespoke data curation an enlightening process that helps them feel connected to the process and "makes things which may have seemed quite abstract much more tangible and 'real.'" [23]

Some participants reported that they developed a notion about correlations between the decisions in the data curation process and how the machine learning model behaves. For instance, a deliberate choice of stroke thickness would eventually reflect in the sharpness of the results. Several participants referred to a similar causal relationship that they observed between the data collection process and the machine learning model behavior.

The data collection experience not only allowed the participants to learn about an end-to-end machine learning pipeline but also to understand their skills and abilities better. An enlightening observation came from participant #7, who realized their handwriting was not precisely as perfect and persistent as they used to perceive: "I have a great deviation in a few characters, and that is where the generated data was inefficient."[24] They had to adjust their handwriting and data collection discipline to produce more uniform samples, making learning easier for the model.

This was an important point. Despite my efforts toward making the data collection closer to real-life scenarios, there was still an inevitable level of detachment between the two. From the technical point of view, using more advanced handwriting recognition and automated segmentation methods could help address these issues, with the tradeoff of less transparency in the toolmaking workflow. However, from a broader perspective, this is an example of bi-directional dynamics between the meta-tool and the creative practitioner. The meta-tool empowers the creative practitioner, while the creative practitioner needs to embrace the affordances of its meta-tool and diverge from its routine workflow to achieve a greater goal.

Other technical challenges and unexpected results were observed, especially in this study's first iterations and early stages. An interesting result came from participant #5, where a wrong padding value resulted in a series of chopped-off, yet exciting, artifacts (Figure 40).

---

[22] Participant #2, reflection paper.
[23] Quotation marks inside the quote are from the participant.
[24] Participant #7, reflection paper.

.

Figure 40. An unexpected artifact resulted from wrong padding parameters in the data pre-processing phase, images from 48-770 students, reproduced here with permission.

4.6.2 Data as Interface

Interacting with the learning model and controlling its outputs by data helped participants understand the importance of a vetted and well-curated dataset in a machine learning procedure. It is also an informative way for both novice and expert users to interact with machine learning generative models. One participant described the "… rather creative effort [that] is required to compose a means to get useful data that can be trained by the machine …" compared to code which is a "…matter of experimentation and analyzing results."[25] Another participant addressed the relationship they developed with the dataset through this process. They mentioned their prior experience with ML but "never had the opportunity to have such a close relationship with dataset before." They then elaborated, "[t]he experience of manipulating models with data is a great way to understand the importance of a well[-]curated and robust dataset." This helped them realize the sheer significance of data in the process. As they put it, "[t]his fact is reinforced and "lived" through this kind of "data-first" project, so I really commend the end[-]user impression and respect I've gained through this learning experience."[26]

One interesting participant observation was the possibility of combining data curation and parameter adjustments to control the learning procedure on different scales. Participants mentioned that working with data as an interface empowered them to steer the direction of outcomes significantly and let them take more significant steps toward desired results—or, as one participant described it, "coarse grain."[27] Working with code and hyperparameters gave them granular control over the behavior of the model, fine-tuning results, and making small improvements, or as the same participant put it, "fine grain."

Despite primarily positive opinions, one participant did not find the approach rewarding. While working on this project, participant #8 had multiple issues with the data pipeline and, consequently, the learning model. As a result, they could not observe any meaningful interaction between the data and the learning model and mentioned that they were unsure if they could control the model through data. Although they were optimistic about the affordances of data since "… it felt more engaging than tweaking arbitrary

[25] Participant #10, reflection paper.
[26] Participant #3, reflection paper.
[27] Participant #2, reflection paper.

parameters," they believed that writing code is a more "immediate" and "direct" way of controlling the model.[28]

At the end of the semester, the same participant revisited the toolmaking process using the same dataset, but this time using the v.3 notebook. This time, the results were very satisfying. This was one of the biggest lessons of this study; proper tools and implementations that support user-friendly and real-time interactions can drastically change the user experience and the results.

4.6.3 Integrated Dashboard

Participants found the integrated dashboard a valuable tool in their process that gathered all three steps of the toolmaking process in one unified interface. The interactive data curation tools that helped them visualize and comprehend the diversity of samples in the dataset were received very well as a step beyond the widgets available in the CoLab notebooks. Some participants even found them helpful in inspecting their own datasets. As one participant put it: "the more I used the dashboard, the more intuitive it became." The impact of the dashboard on participants' intuition of data and its potential in ML went beyond my initial expectation: "…naturally, made clear by the dashboard, data in its vastness and diversity is what determines the code, rather than the other way around."[29] However, it was not the same experience for everyone. Some believed that the dashboard was more interactive than responsive and informative.

The hovering overview was the favorite interaction on the data dashboard (Figure 41). Some participants indicated that the plot with both t-SNE and label distribution was more helpful than the plot with only the t-SNE distribution. It helped them understand the distribution of data by quickly hovering over the X-axis to see style changes and over the Y-axis to observe various glyphs in relatively similar shapes. The selection tools were also well-received by the participants. They appreciated the ability of the selection tool to quickly select and slice sub-sections of the dataset in a few clicks rather than applying cumbersome NumPy operations to split and select data (Figure 42).

The organization of the dashboard as a one-page web app (Figure 23), with the data curation tools constantly available, made the overall process cleaner and smoother. While finding a specific function in the notebooks required constant scrolling and more attention, "… in the dashboard, [one] find[s] the buttons, labels, sliders without any effort; the interface on the dashboard [was] so clean." One participant mentioned that "[t]he dashboard provides an unobstructed interface for training; users can entirely focus on selecting data, training models, designing fonts, and rendering paragraphs." This user concluded that as novice ML user, they prefer the dashboard over the notebook.

[28] Participant #8, reflection paper.
[29] Participant #10, reflection paper.

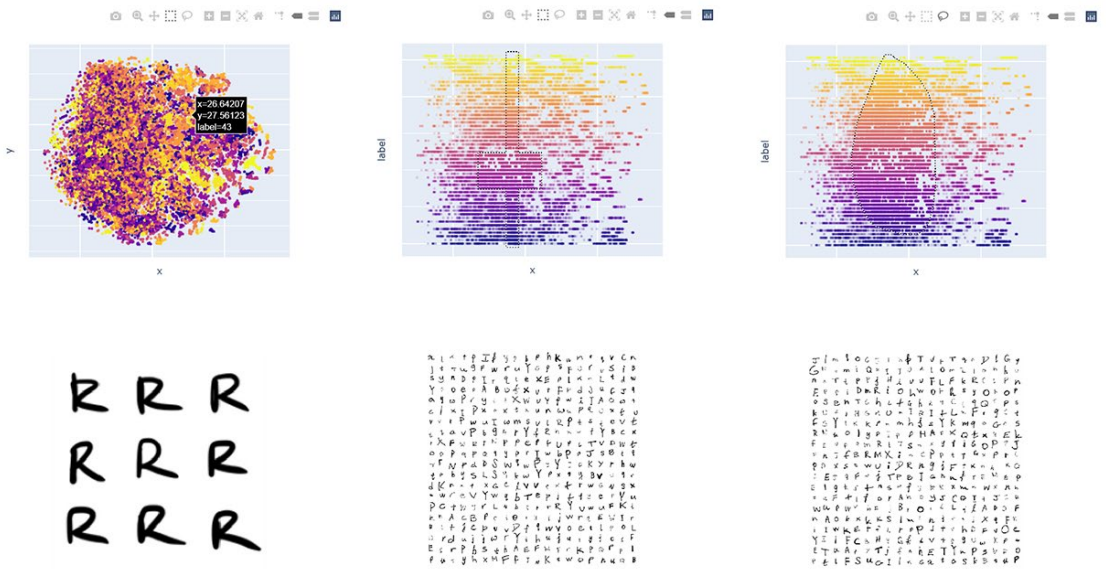


Figure 41. Dashboard's user interactions: hovering preview (left), rectangle selection (center), lasso freeform selection (right).

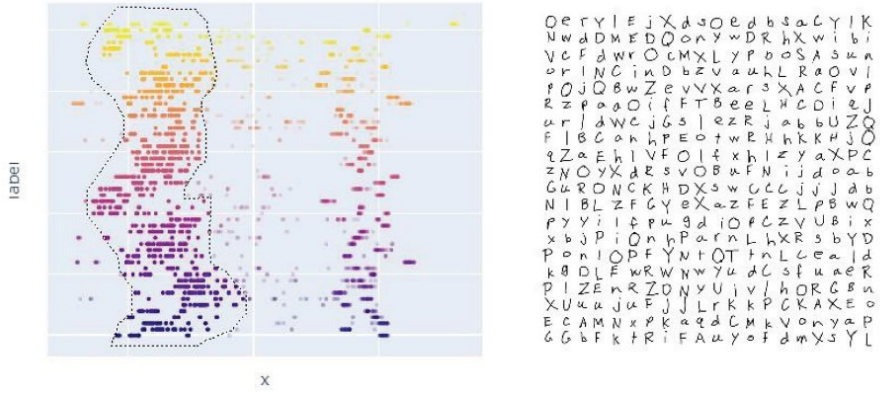


Figure 42. A data curation sample, note the effort to keep the samples visually close while covering the full range of glyphs using the t-SNE/Label plot (left), a view of 400 samples from the selection (right), image courtesy of Learning Matters students, reproduced here by permission.

There were suggestions to improve the dashboard. For instance, adding a brief description for each dashboard section was suggested. Although the data curation dashboard was accompanied by a comprehensive video tutorial where each functionality was discussed in detail, adding such descriptions as tooltips would provide participants with a quick refresher on each section's functionality.

Another suggestion was to replace the 2D plots with 3D plots. This idea was tested in the early stages of dashboard development. Despite the exciting visual appearance, it became clear that interacting with a 3D plot on a 2D screen is frustrating. Adding the extra dimension introduced several user interaction challenges while adding limited values. Accordingly, it was replaced by the current 2D plot in the final version.

While most comments on the data dashboard were positive, some participants pointed out a fundamental issue with the dashboard workflow. One participant pointed out that while the visualizations were

interesting, it was unclear what kind of data should be selected to return good results. This is a critically valid issue. Each action on the dashboard was followed by a real-time response which would guide the user to adjust its subsequent decisions accordingly. However, since the training process was detached entirely from the dashboard, participants could not predict the consequences of their decisions on the training process and results. This issue was addressed in the second iteration of this study by integrating the training phase into the dashboard. Participants could add, remove, or completely change the training dataset after each designated number of epochs.

It is worth mentioning that the C-VAE model used in this study is relatively shallow,[30] and the dataset is very small. Therefore, the training process is quite fast. For more complex models with slower training processes, such integrations would not greatly benefit the user experience. In such cases, the time gap between the users' decisions and their effects on the results can easily surpass any meaningful real-time interaction threshold.[31]

Finally, I noticed that the data curation tool had a rather unexpected drawback. The tool was so easy to use that it encouraged participants to create wildly diverse datasets. To their disappointment, the C-VAE model could easily be overwhelmed with such diverse datasets, and the results were abysmal. As observed by one of the participants, compared with a regular dataset—coming from a single participant—with the same sample size and the same number of epochs, a blend dataset would result in lower quality and less desirable results due to the limited capacity of the C-VAE model.

4.6.4 Training Process

As discussed earlier in this chapter, the training process was intentionally stripped to its barebones to let participants focus on the data pipeline. Participants used a pre-defined model with mostly fixed hyperparameters. Only playing with the number of epochs, participants tested a wide range between a few hundred and a few thousand epochs and eventually found values between 150 and 250 as the optimal range.

The participants described the training process as "extremely quick"—compared to other machine learning architectures—notably GANs—and "straightforward."[32] As the provided model was optimized to train over a short period, its ability to learn complex data distributions was limited. A participant noticed this issue, highlighting that they had overestimated the model's generalization ability. This forced them to return to the data generation phase and create a more uniform dataset that matched the model's capacity.

Multiple participants noticed that the dataset's size, quality, and distribution significantly impacted their results. The model tends to overfit when training over large subsets of the shared dataset. Some

---

[30] Here, I use the term shallow in contrast with deep, as in deep learning. It is worth mentioning that referring to a model as deep is a subjective decision, based on the current state of the art in machine learning. The model that was used in this study, could be considered a deep model a decade ago.

[31] There were also a few references to the cosmetic issues in the dashboard implementation, i.e., the plot sizes. While the dashboard was designed to act responsive to different screen sizes, the visualizations were best viewed in large desktop screens, not mobile devices, nor laptops. In the second iteration, some level of responsiveness was added to the layout of the dashboard to compensate for this issue.

[32] It is worth mentioning that the two iterations of this study were conducted in two different stages of the class during the two semesters. In Spring 2021, this study followed the GAN module. Participants found the C-VAE model quick and straightforward after working with a derivation of Pix2Pix for three weeks. In contrast, during the Fall 2021 semester, this study preceded the GAN module. Participants from this cohort did not find the model quick and straightforward as they only had the experience with simpler AutoEncoder models based on multi-layer perceptron.

participants adjusted the input data to overcome this issue, while others ignored that and used the overfitted model to generate their samples. Switching from thin to thicker strokes could also help the model achieve better results. Several participants noticed this point and took advantage of this finding, especially after it was brought up during the presentations by other participants. Such information exchanges between the participants became a habit across this study, allowing them to swiftly overcome challenges in their process.

4.6.5 Latent Space Navigation

Navigating the latent space of a C-VAE is slightly different from other types of AE and VAE models. The model could disentangle the labels from the visual characteristics through the conditioning method in the latent space. On the one hand, users should feed the label signals, and on the other hand, they should explore the latent space to find the most desirable style for the given label. Participants could use a series of interactive widgets to navigate the latent space and generate their desired typeface.[33]

During the early stages of the first iteration of this study, some participants could not successfully train their model. Even though these participants could successfully curate their datasets, they had issues with the training process and generating new samples to create the typeface. Further observations and discussions with the participants clarified that the methods to draw samples from the latent space were quite hard to understand and use. Some participants reported the process to be tedious and complained that the sliders in the UI were either over-sensitive in some cases or almost ineffective in other cases.

Several participants highlighted the widgets' real-time visual feedback as an intuitive feature that helped them make their design decisions more efficiently. The tools helped participants visualize and explore the latent space without directly comprehending it: "Although we can't see the actual latent space, changing the mean and standard deviation really gives me [an] understanding of how the letters cluster in the latent space…I could really see how different types of handwriting emerge with comparatively large standard deviation input."[34] The interactions made the latent space navigation more "tangible" and "visual" for users who only possess a high-level understanding of the workflow. One of the participants mentioned that "… there is a design agency in generating the font based on changing values of the latent space."[35]

There were also comments on the shortcomings of the interactive widgets; some participants were confused because the same input parameters could result in slightly different results. This is a direct result of the random algorithm behind the sampling method. Even with the same standard deviation and mean values, a virtually unique Z vector would be sampled each time. Thus, the generated samples would be different. From the users' perspective, it means that if they move the widget sliders over a good example, they cannot reproduce that example by setting the sliders back to the same values, which can be very frustrating when going through 52 glyphs.

---

[33] As discussed earlier, the first version of these interactive widgets which were available through the training notebook v1 proved to be less effective and it was later replaced with a new set of widgets with supporting visualizations in notebook v3. The reflection papers were written based on the v3.

[34] Participant #9, reflection paper, all reflection papers are available in Appendix III.

[35] Participant #10, reflection paper.

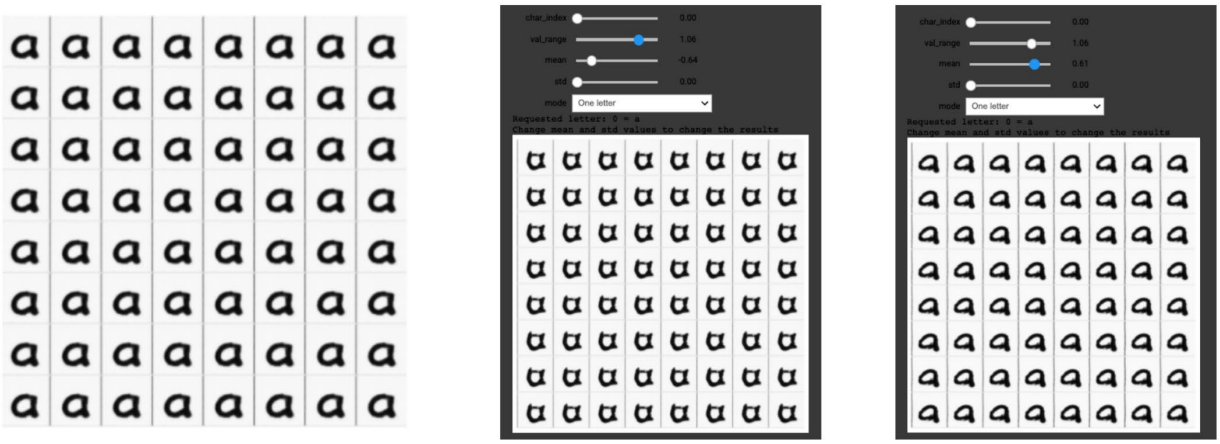


Figure 43. Exploring the latent space to find a desired set of glyphs using widgets in notebook v1, images from 48-770 students, reproduced here with permission.

Multiple participants mentioned that they could not clearly comprehend the effects of each variable on the results. For instance, the fact that changing the values in one direction would not necessarily improve the outcomes was challenging to comprehend for some participants. Participants came up with a range of suggestions to address this issue, most notably 1) adding a series of brief descriptions to each widget to clarify the variable's effect on the results and 2) including some form of visualization to depict the effects of changing std and mean values on the sampling distributions.

Meanwhile, some users managed to decipher some form of causalities between the variables and the results. Participants opted to set the variables to their extreme values and figure out their effect on the results, which helped some of them decide on variable combinations very quickly. One participant carefully observed the effects of variables on some letters and tried to draw a pattern of the relationships between variables and results through observation: "I believe, the greater the mean is, it allows for normal transitions of styles without distorting the letter much … when the mean slide[r] is in -1, it gave the 'a' character a gothic look, with the stretched pointy edges. And for the other side of the spectrum, the 'a' was just a little bit squished down but still looking like an 'a' character, which is difficult to spot from the lower mean value" (Figure 43).[36]

In general, due to the stochastic nature of the process, it is not an easy task to standardize latent space navigation. While it is relatively easy to predict the general effect of each variable in the sampling process, fine-grain control is not available before careful observations and examination of each variable's effect. One possible method to improve users' experience in latent space navigation is using an existing sample as a "seed" to start with. In this scenario, the user selects a desired sample from the dataset and passes it through the encoder model to find its vicinity in the latent space. This latent representation can be the starting point for generating other glyphs and variations. In this scenario, the user starts from a visual sample, and then the sample helps find the abstract mediums (std, and mean values).

The v.3 notebook followed the opposite approach: the sampling method would start with abstract variables (std and mean), then these variables would be used to generate visual results. In this method, users should start from random visual results and change the variables to find the desired solutions. On

[36] Participant #5, reflection paper.

.

the other method, users can pick the desired sample, or draw the desired sample, generate its corresponding variables, and continue exploring the vicinities of that sample in the latent space.

4.6.6 The Three-Round Process

The three-round process was executed in a two-week period. For the first round, participants had one week to get familiarized with the process and share their data with the cohort. They were given another week to work on the second and third rounds combined.

Round one:

The first round of data collection was a warmup and learning opportunity for the participants, and they had a week to finish this phase. It was the first experience in data generation, pre-processing, and curation for many of them and became a bumpy journey for some of them. Some of the common mistakes were prevalent among participants, including, but not limited to, 1) issues with the alignment of glyphs in the cells, 2) lack of consistency between the size of uppercase and lowercase letters, 3) discrepancies between the style of glyphs across the charts, and 4) some issues with the post-processing procedures. Moreover, some participants reported issues with glyph with close visual features, such a "c", "b", "d", "q", "p", "O", "Q".

In the second iteration, where participants could write letters in words, the most common issues could be traced back to the annotation and segmentation process, as discussed earlier (Figure 38). Students completed ten pages, covering around 500 words. While this is a relatively small piece of writing, segmentation, annotation, post-processing, and checking the data took significantly more than the first method. Participants reported over 5-6 hours spent in this phase.

Round two:

For the second round, participants put the experience of the first round into practice and generated more consistent data with thicker strokes, clearer edges, and adjusted alignments to overcome the previously observed issues. Other techniques, such as applying different pre-processing steps and changing the software for data collection, were also used by participants to improve their toolmaking process (Figure 44). Thus, many participants found the second round's results superior to the first round's due to these modifications in the dataset.

Some participants also reported that the experience they gained in the first round helped them work with the interactive widgets with more dexterity and navigate the latent space more productively. However, one participant reported that they could not improve the results remarkably between the two rounds.[37]

In the second iteration, participants were asked only to complete another five pages (250 words) rather than all ten remaining pages due to time constraints. Accordingly, instead of working with a completely new set of samples, they combined the data from the 15 pages. This did not increase the total number of samples in their training dataset, as the total number of samples was set to 1872. However, it helped them balance the number of samples for each letter.

Round three:

In the last round, participants engaged in a collective data curation activity, where each participant could tap on a pool of shared data samples, collectively made by all participants, to curate a new training and generate a fusion handwriting typeface. In several cases, students failed to create legible results on the

[37] This participant revisited the pipeline later in the semester and came back with very interesting results.

first try. Once they had access to the notebook v3, they combined it with the data curation dashboard and made very interesting results.

Some participants managed to train their model with relatively small datasets—2x to 6x the size of a single dataset— while some others tried to mix all the available data at that point—up to 15x larger than a single dataset. The results were quite interesting, and some participants reported that it was the best outcome of the three. In contrast, some participants found that the results were not as consistent as they expected to the point that it was possible to see traces of multiple handwriting in the results. One participant tried a mixture of two completely different handwriting samples to create its typeface; the results were not visually appealing, and "it looks like a right-handed person trying to write with their left hand."[38]

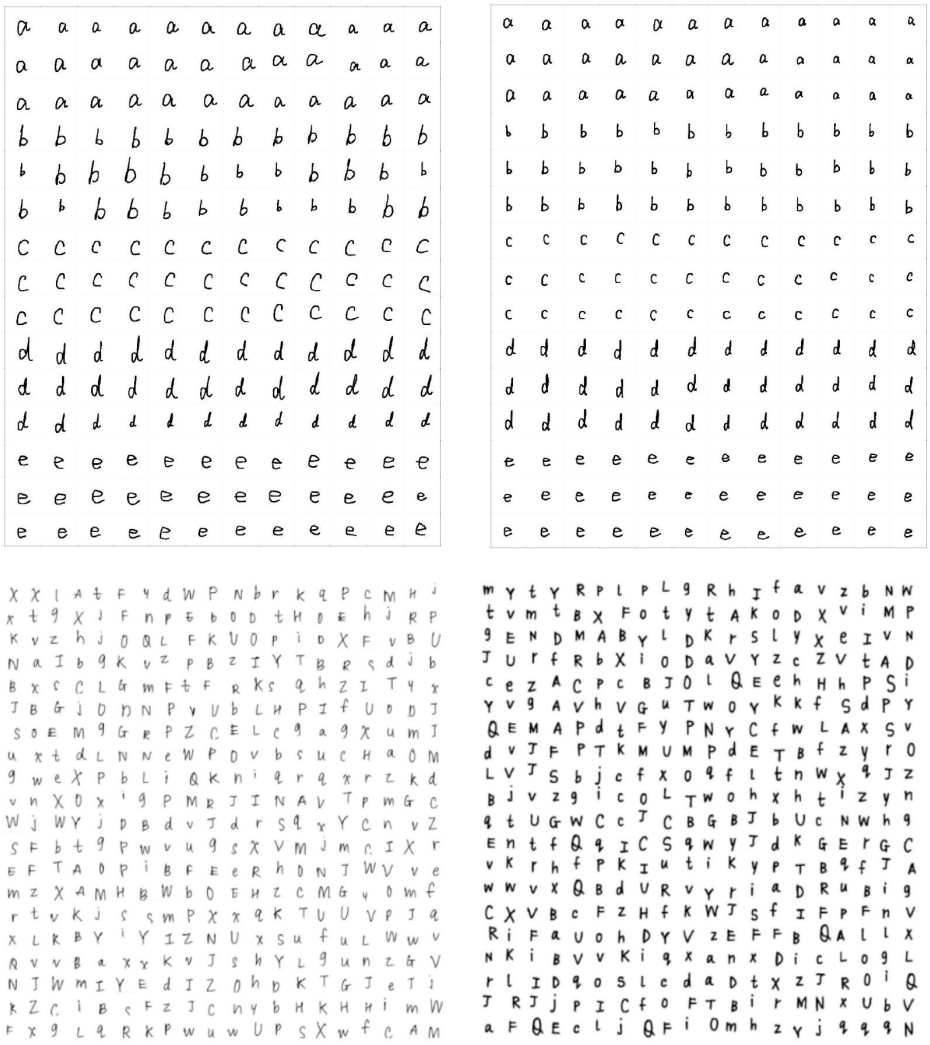


Figure 44. First dataset (left column) and the second dataset (right column), notice the consistency in size and style of the second set compared to the variances of the first dataset (top row) and change of thickness (bottom row), images from 48-770 students, reproduced here with permission.

[38] Participant #5, reflection paper.

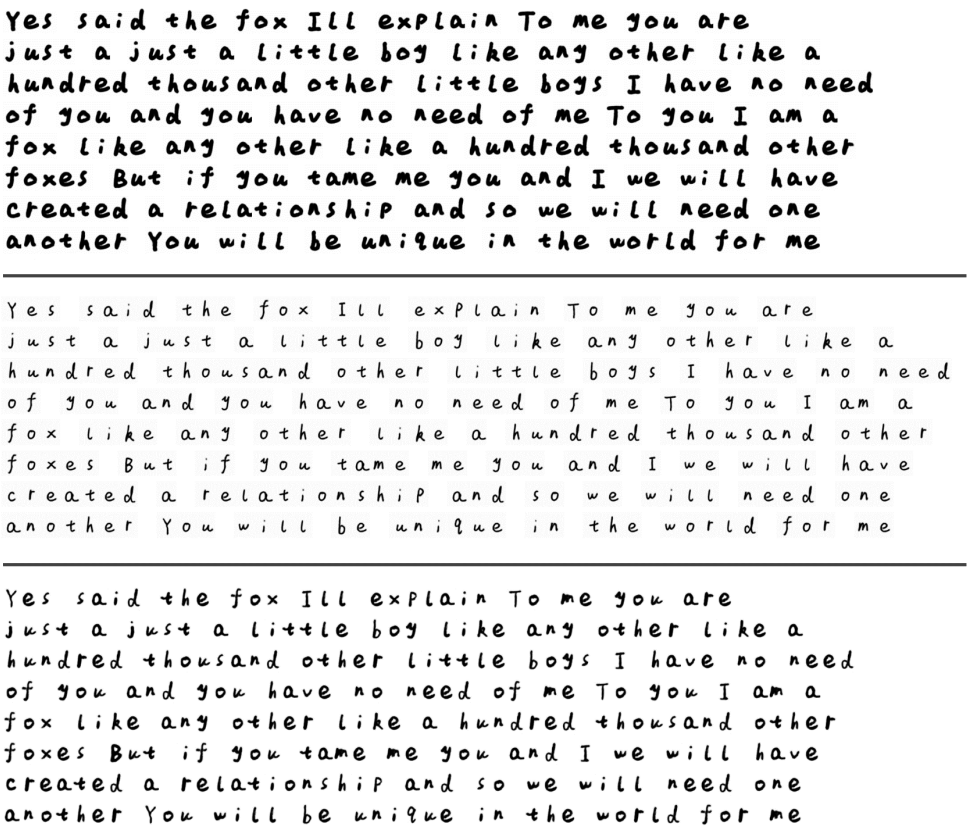


Figure 45. Typeface designed and generated by participant #2: round one (top), round two (middle), round three (bottom), notice the consistency of results on round one and two, where all the samples were generated by one user and round three where data was curated from different users' inputs, images from 48-770 students, reproduced here with permission.

4.7 Discussions

The study, in general, was a successful example of engaging users in the machine learning-based creative toolmaking workflow. The participants were profoundly engaged in the process, from creating their datasets to using the machine learning model for generating the typeface. In this section, it is enlightening to revisit the objectives set at the beginning of this study and reflect on them based on the results, observations, and participant feedback.

4.7.1 Data as Interface

One of the primary objectives of this study was to inquire about the affordances of data as an interface to interact with and control a machine learning model. Throughout the discussions and reflection papers, participants' feedbacks were positive. They found data as a more intuitive interface than coding. Data allowed them to sense the relationship between the input data and the results. This made it easier to control the model and improve the quality of generated samples by taking course-grain steps toward the

desired results. One participant described the design of the data collection process as a "creative process in itself" through which users can "get into a conversation with the tool through the data."[39, 40]

4.7.2 Physical Context

Another objective of this study was to investigate if the machine learning-based meta-tools allow the users to engage with elements of their physical contexts, such as specific tools and materials, in the toolmaking process. In several cases, participants referred to their medium of choice—pen and paper, tablets, digitizers—as well as the software packages that they used for data processing. It becomes clear from this repeated pattern that the data collection process, and subsequently the whole toolmaking process, were heavily affected by the medium and the software package. Even changes within the same medium turn out to have noticeable effects. For instance, one participant expressed satisfaction with the results when using thick Sharpie markers rather than thin pens. Another participant mentioned using Adobe Acrobat Pro instead of Adobe Acrobat Reader to have more desirable samples (Figure 46). These discussions prove one of the primary points of this study, the affordances of the toolmaking apparatus to capture and reflect elements of the physical context.

Figure 46. Effects of software used in the data collection on the samples, notice the way the software handles the corners in different samples, image from 48-770 students, reproduced here with permission.

The COVID-19 restrictions heavily influenced this study's design mostly reflected in the data collection phase. As it was brought up in the reflection papers, the level of abstraction had impacted the participants' natural workflow. Most notably, collecting samples in secluded cells introduced undesirable side effects, i.e., unnatural kerning.[41] Several participants referred to kerning as a significant issue with their generated typeface. After observing the first round of presentations, participant #8 concluded that the best results—probably the most legible and closest to the original dataset—are the outcomes of almost "robotic-looking" and uniform samples. They continue to bring up a very interesting point: "…to really capture the individual style of a writer, woodworker, one would need to capture their movements and results when they are in a flow and operating without self-conscious manipulation of the tool."[42] Their point of view directly pointed at the abstract methods implemented for data collection.

4.7.3 Idiosyncratic Elements

Several participants reported that their generated typefaces were visually close to their handwriting. Participants generated samples based on their handwriting throughout the data collection process and adjusted the dataset based on their personal preferences. However, it must be noted that as they learned the model's behavior, they gradually adjusted their samples to the affordances of the machine learning model. This bi-directional interaction was one of the most interesting observations of this study;

---

[39] Participant #8, reflection paper.
[40] It is worth mentioning that one participant described the process of controlling the model through code and hyperparameter as a type of "craft" that takes time and practice to achieve the expected results. This is an interesting observation that resonances the notion of craft in the realm of information technology (Mccullough 1996).
[41] Kerning is the adjusted space between two letters that can be different between specific pairs of letters.
[42] Participant #8, reflection paper.

participants molded the meta-tool to fit their workflow and concurrently adjusted their workflow to fit the meta-tool's affordances.

On the training side, the visualizations—most notably, the three-row plot—helped participants see beyond the classic machine learning plots and decide on the training progress based on their preferences. Additionally, the interactive widgets in the training notebook, with the real-time plots, allowed them to incorporate their personal preferences in the generative process. The same applies to the data curation dashboard, where they could select and train their model based on the desired dataset instead of an off-the-shelf one.

4.7.4 Understanding the Behavior of the Machine Learning Models and Aligning It with Creative Practitioners' Workflow

The three-round process was an eye-opening experience. One could observe the participants' learning curves, starting from the bottom, learning from mistakes, and exploring the uncharted territories of other participants' data. In every round, participants learned new lessons and faced new challenges, and they managed to improvise solutions to overcome them in the next round. Glyph alignments, size and style consistency, the model's tendency to learn thicker strokes, and adjustments to improve the notebooks are only a few challenges that participants faced initially and solved successfully in the following rounds. I found the three-round approach one of the most successful components of this study.

An objective of this study was to observe the affordances of collective data curation in the toolmaking process. The initial assumption was that the participants could improve their toolmaking process by enriching their data set with the samples provided by their fellow participants. The data curation dashboard was developed around this idea and let the users select a desired sub-section of the collective database to generate a new typeface. The data curation dashboard and its implantation in the integrated dashboard were also very well-received among the participants. The real-time interactions helped several participants select data, inspect their dataset, and explore the patterns and similarities in the dataset. The t-SNE plots, specifically the t-SNE/label plot, with their real-time previews, were very helpful in this case.

However, from the first presentation, it became clear that most participants did not fully comprehend its applications and potential in the data curation process. It was only after the presentation that participants learned more about the affordances of the dashboard from each other's process and took advantage of it for the final submissions. It became clear that the users would not effectively adopt an abstract dashboard without concise yet engaging documentation and demonstration.

During the study and after the first round of presentations, it was clear that this method of mix-and-matching data introduces significant challenges to the learning model. To make the participants' experience closer to a real-time design experience, the C-VAE model was optimized to reduce the training time. This approach resulted in a well-adapted model for small datasets with a low level of variance in the training samples. Some participants pushed the model to its limits by training it over a wide range of samples from multiple other participants at once. Upon facing a training dataset with such a wide range of variations, the model failed to learn the data distribution, and the results were far from legible.

The fascinating part came after the first round of reviews; participants observed how the model collapsed in such cases for everyone and embraced this shortcoming. They switched gears to redefine their approach based on the affordances of their machine-learning model and opted for carefully curated datasets. Most participants successfully generated their handwriting typeface using smaller subsets of the shared database. A model with deeper architecture could potentially learn the data distribution in larger

datasets without overfitting. However, this achievement would come at a potentially steep increase in training time, which defies the goals of this study.

While all the participants had prior exposure to the details of the machine learning model and the means to adjust various hyperparameters, only a few of them found it necessary to get engaged with the model's details. In contrast, the others found the number of epochs the only parameter that needed further adjustments. Meanwhile, all participants got heavily engaged with the interactive widgets and real-time visualizations to explore the latent space and found it very helpful to generate desired results from the model. The widgets were not perfect, especially in the v1 of the notebook. However, changes I made to the v3 improved users' experiences.

Participants reported on their experience with the process of creating bespoke datasets, curating data, training a model, and making their typeface generator tools. They described this process as building a "real human-AI relationship" through "physical-to-virtual data conversion." There were multiple indications of a common understanding about how specific visual features of some glyphs in the dataset can significantly improve the behavior of the learning model or which portions of the dataset will create the most challenging bottlenecks in the learning process. On a few occasions, participants mentioned what they learned from observing other participants' datasets and workflows to enhance their own.

4.8 Lessons from this Study

As I reviewed the submissions and reflection papers, I realized that even though all the participants had an equal chance to receive training and supporting materials, it was only after the first presentation that almost all participants could accomplish all the objectives of this study. The leap from the presentations to the final submission can be credited partially to the improved tools and widgets. However, multiple participants highlighted that they learned from the other participants and then revisited their work and improved it. It is an interesting observation that resembles the TEA Set story (H. Collins 1974).[43] Accordingly, a series of one-on-one working and training sessions will be designed for the subsequent studies where the expert user and the researchers get engaged in a collaborative dialogue to learn from each other and develop different pieces of the meta-tool.[44]

Interactive visualizations are critical elements for communicating with users with limited technical knowledge. These visual mediums help users intuitively correlate their decisions and the behavior of the machine learning model. Even for the users with prior exposure to machine learning, these visualizations were more effective than abstract activities such as coding.

4.8.1 Participant, Toolmakers, Meta-Toolmaker

Throughout this study, it became clear that my role is best described as a meta-toolmaker. I interfaced with the participants through the meta-tool and provided the underlying technology to allow them to create their handwriting typeface generator tool. I sculpted the meta-tool primarily based on my decisions and understanding of the study's question. Indeed, the bi-directional dynamics between the participants and me enriched the process and smoothened the experience.

---

[43] TEA Set is a now-classic example of Collins studies on tacit knowledge. In *The TEA Set: Tacit knowledge and scientific network* (1974), he emphasizes the importance of tacit knowledge, even in cutting-edge scientific efforts. This study is mentioned earlier in Discussion on the .

[44] Building on the lessons learned in this study, I opted for one-on-one sessions with the participant on the second case study.

Meanwhile, participants were free agents who were exploring uncharted territories when it came to toolmaking. I was not in charge, but I guided them when they needed clarification. I provided the infrastructure and the meta-tool, and they built their tools and used the tool to generate several iterations of new typefaces. As such, they made their own handwriting typeface generator tool.

In the next study, I examined a different approach to the meta-tool and toolmaking process. I aimed to blur the lines between the meta-tool and the tool and invited the expert user to engage directly in the toolmaking process.

4.8.2 Limitations and Future Steps

The scope of this study was narrowed down to individual letters. However, as mentioned by the participants, this data collection method forced them to deviate from their routine handwriting to satisfy the technical limitation of the data pipeline. Even in the second iteration, participants had to adjust their handwriting to match the data annotation method. This issue weakened the argument of this study to work with real data generated by the users. A potential next step for this study is to work on methods to address this issue.

Another direction to expand this study is to explore different rendering approaches, such as rendering words or sentences instead of one letter at a time. This expansion requires integrating proper kerning and letter spacing. Also, the scope of this study can expand to include ligatures and try languages that use cursive scripts as the default form of writing, i.e., Persian scripts.

Shifting from pixel-based representation into traces and vector-based representation is another area that can be explored further. This approach opens new opportunities to study the physical toolmaking aspects using computer-controlled actuators.

A critical limitation of this study, specifically in its first iteration, was the lack of integration between the meta-tool's components. Throughout the review process, I realized that the fragmentation of the machine learning pipeline and lack of interoperability[45] was a potential bottleneck in making a comprehensible workflow. Interoperability hinders the design process, as switching between platforms for data curation, training, and generation phases is time-consuming and distracting (Veloso et al. 2022).

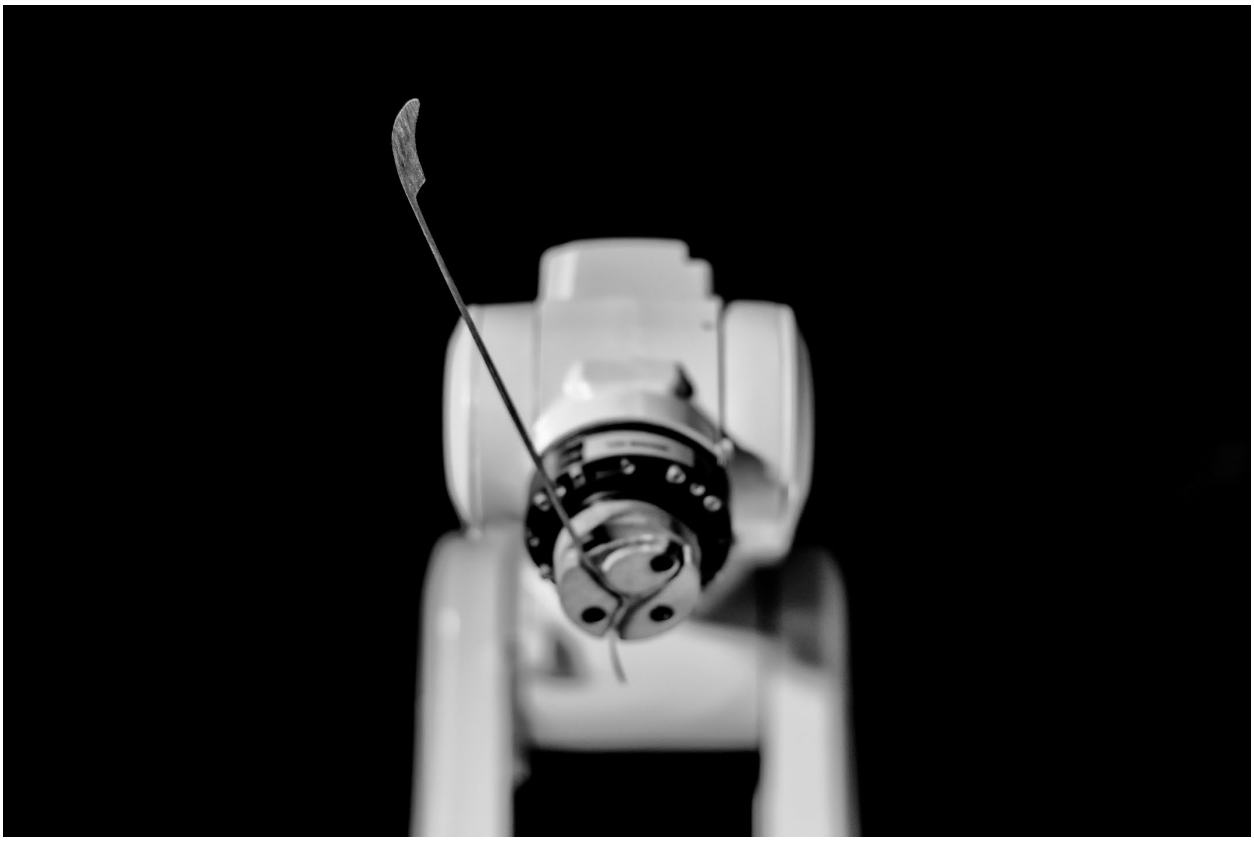
In the first iteration of this study, the three main steps of the machine learning pipeline—data collection, data curation, and model training—were completely separated from each other. The second iteration partially addressed this issue by integrating the data curation, training, and sample generation phases. However, the data collection and annotation were still detached from the rest of the toolmaking process. When participants generate the learning samples, it will be more intuitive to see the model trained on the data in real time and provide instant feedback to the user, opening the way for an iterative loop of data curation, training, and generation.

Integrating all phases of a machine learning pipeline—data collection, data curation, training, and generating new samples—and improving interoperability can help users oversee all elements of their toolmaking process as a unified system and comprehend how it works. An end-to-end toolmaking apparatus with readable and communicative visualizations can help users draw lines between their decisions through the data collection and outcomes and see the connections between their decisions and

[45] Interoperability is the ability of two or more software components to cooperate despite differences in language, interface, and execution platform (Wegner 1996, 285).

the results. This understanding of a complex system shapes a new form of tacit knowledge for the process of machine learning-based creative toolmaking.

Working with C-VAE models helped this study in various ways. Most notably, I found the fast and reliable training process one of the biggest advantages of this architecture. Also, the trained models demonstrated good coverage of the training dataset. Finally, the model's simplicity made it easier for the participants to understand how it works. However, the blurry results, a known characteristic of VAE models, were the main downside of using this architecture. Also, the model was designed to be light and fast to train, so its ability to learn complex data distributions was limited. Accordingly, another possible direction for future work is to explore other generative models that could learn more complex data distribution and generate sharp-looking results.



Chapter 5. The ThirdHand

This chapter is a comprehensive report on the second case study of this thesis, where I collaborated with a musician to develop a robotic musical instrument to play *santur*, a traditional Persian musical instrument.

Building on the lessons learned from the first case study, ThirdHand explores how a collaborative approach to machine learning can mitigate the lack of technical machine learning experience among creative practitioners and help them to integrate the idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process.

My experience and familiarity with santur were quite limited when conducting this study. Both of us were venturing into new worlds that we had not explored previously. For me, santur was a black box, and machine learning was a mystery for the musician. The process aims to bridge the two worlds through conversation, interaction, and collaboration.

This chapter is organized into six sections: I will 1) introduce the framework of this case study, then, through the lens of other scholars and creative practitioners' work, I will locate this study in the broader perspective of musical toolmaking. The proceeding section is an in-depth discussion of the 2) methodology and technical aspects of this research, followed by a thorough description of the 3) meta-tool and 4) early tests. Dividing the last three sections was a challenging task. As we worked collaboratively for several months, we iteratively revisited research methods and refined the tools based on the musician's feedback, comments, and recommendations. This workflow, based on the research-by-design approach, lasted during the entirety of the study, and gradually shaped the methodology toward its final form. Accordingly, most pieces of study reports found their homes between the lines of the methodology section. The final section reports on the 5) demonstration. I conclude the chapter with a 6) discussion of the study and its outcomes.

5.1 Study Framework

This study addresses the two primary topics of this thesis: 1) accessibility of machine learning-based tools for creative practitioners and 2) embracing the idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process. Compared with the first case study, this one delves deeper into the collaborative aspect of the toolmaking process between the musician and toolmaker.

5.1.1 Question and Hypothesis

The primary question in this study is:

- How do interfaces for data generation and curation for generative machine learning offer new pathways for toolmaking for creative practitioners?

The secondary question is:

- How can a collaborative approach mitigate the lack of technical machine learning experience among creative practitioners and help them to integrate the idiosyncratic aspects, elements of the physical context, and nuances of their creative practice in the toolmaking process??

To answer these two questions, I hypothesize that:

- Interfaces for data generation that emphasize user-generated data to integrate elements of the physical context and users' subjective preferences can reveal new potentials of generative machine learning to support creative practices.
- A collaborative approach to developing ML-based tools for creative practices can meaningfully bring ML experts' technical literacy to complement creative practitioners' domain knowledge and skills, overcome the technical ML challenges, and help integrate various idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process.

5.1.2 Goals and Objectives

A primary objective of this study is to examine an approach to ML-based toolmaking process which embraces the bi-directional dynamics between 1) the social context—i.e., relationships and interactions between human agents—, 2) elements of the physical context—i.e., tools and instruments—, 3) the personal context, and 4) the underlying technology.

Documenting and reflecting upon the collaboration between the toolmaker (me) and the creative practitioner are other goals. In this study, Mahtab Nadalian, the participating musician, had no prior exposure to computer programming, robotics, or machine learning. Therefore, her primary means of interacting with the machine learning models was through data and collaboration with me, the toolmaker. Thus, the proposed toolmaking process primarily relied on these dynamic interactions between us to iteratively refine the meta-tool and fine-tune it while keeping it accessible, transparent, and understandable.

This study also aims to explore the potential of data as an interface to make ML-based toolmaking more accessible to creative practitioners. Observing this approach's effects on the musician and its robotic musical instrument is another objective of this study.[1]

The input data modality differs from the output in this study; the musician provides *mezrab*[2] stroke samples, and the robotic musical instrument plays the notes on a santur to produce the notes. In such a

[1] The notion of data as the interface is based on (Rebecca Fiebrink 2016).
[2] مِضراب

setup, inspecting and curating data samples is not direct and explicit as I developed and implemented in the SecondHand study. Exploring the data curation methods and tool refinement in this scenario is another objective of this study.

5.1.3 Context

Robotic Musical Instruments

Creating robotic musical instruments is by no means a new quest.[3] It has been a topic of interest for at least two centuries. Barrel-operated stringed instruments, as well as roll-operated pianos, have been to some extent accessible in the market since the mid-19th century. Notably, "Pianista"—originally a pneumatically operated piano-playing machine by French innovator Fourneaux—was first patented in 1863 and soon became the synonym for player instruments (Bowers 1972).

At the dawn of the 21st century, the passion for making music-playing machines was still strong. Robotic musical instruments—sound-making devices that automatically create music with mechanical parts—to play piano, percussion, string instruments, wind instruments, and even turntable were not rare (Kapur 2005, 1) (Figure 47). While most of these machines were designed to mimic the way human musicians play instruments, some researchers went beyond the traditions and designed robotic systems to play instruments in novel ways to produce "… additional sonic variety and playability" (Weinberg et al. 2020, 8). For example, the magnetic resonator piano, a hybrid acoustic-electronic instrument based on the grand piano, could directly manipulate the piano strings using electromagnetic actuators (McPherson 2010) (Figure 48).

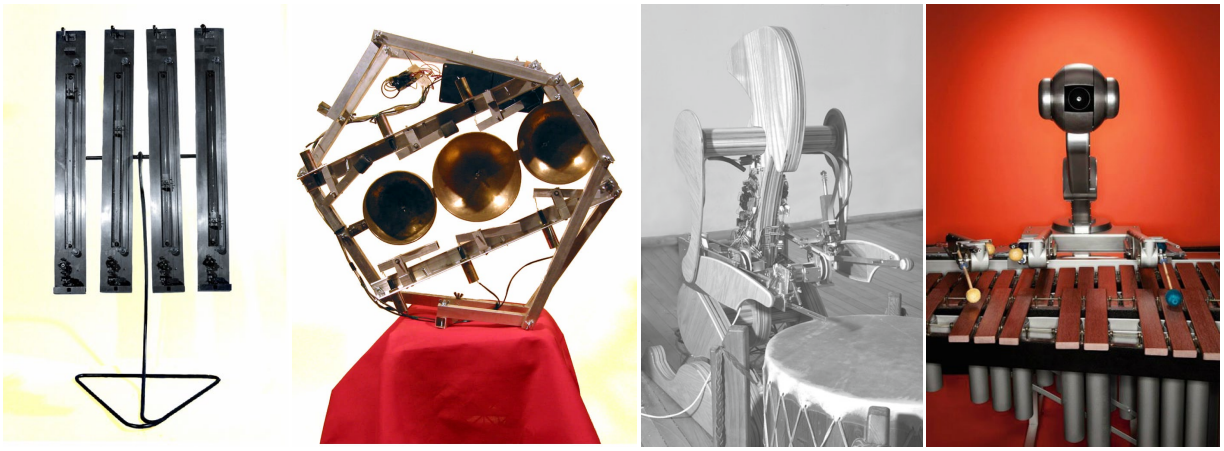


Figure 47. Robotic musical instruments, from left to right: TibetBot and GuitarBot by LEMUR, images from (Singer et al. 2004)*, Hail on percussion and Shimon on marimba, images from* (Weinberg et al. 2020)*.*

[3] While not a topic of this study, creating machines to generate music is a subject of interest in the creative computing society. Inquiring into the affordances of artificial intelligence in music creation has also been a subject of interest for researchers since the mid-1950s. One of the earliest examples is Lejaren Hiller and Leonard Isaacson's work published in the *Journal of the Audio Engineering Society*, titled "Musical Composition with a High-Speed Digital Computer" (1958). In that paper, they introduce the project they started as early as 1955 and discuss their technique for writing music using Illinois Automated Computer, better known as Illiac I. Fast forward to recent years, a significant body of literature on AI-assisted musical creation has been published. More recently, machine learning has become a vehicle for music generation research, as reflected in the current efforts of the Magenta team at Google (Chris Donahue, Ian Simon, and Sander Dieleman 2018). *Handbook of Artificial Intelligence for Music* (Miranda 2021) is a valuable resource in that matter for the eyes of interested readers.

Figure 48. The magnetic resonator piano, images from (McPherson 2010).

ML-Based Musical Instruments:

Another approach to music toolmaking is crystalized through the works of Rebecca Fiebrink. While not incorporating a physical instrument to produce sound, she has developed a series of machine learning-based musical tools in collaboration with musicians (Fiebrink 2011). For instance, she collaborated with a cellist, utilizing K-Bow[4] and Wekinator[5] to train a discriminative machine learning model. Once trained and evaluated by the artist, the model could classify seven different bow gestures performed by the artist. Fiebrink also worked with other artists to help them make their own machine learning-based musical tools, where an interactively trained machine learning model could recognize the signals from the artist and pass these prompts to a sound-producing algorithm (Figure 49).[6]

Figure 49. The MARtLET instrument, by Michelle Nagai, using Wekinator developed by Rebecca Fiebrink, images from (Fiebrink 2011)

[4] K-Bow is a sensor-packed bow that could collect the data from the artist in real-time. Developed by Keith McMillen labs, KMI Labs, in early 2010s, it is described as a "… MIDI controller that takes information from the gestures and movements of a violin bow and translates it into MIDI Bluetooth controller data" (Keith McMillen Instruments 2020).

[5] Wekinator, as I introduced it in Chapter 2, ML-Based Toolmaking for Creative Practitioners is an open-source software for real-time interactive machine learning developed by Fiebrink in 2009 (Fiebrink, Trueman, and Cook 2009). It is developed based on Weka, an open source Java ML library (Witten and Frank 2002).

[6] Fiebrink's Ph.D. thesis is a valuable resource for interactive machine learning for creative practices (Fiebrink 2011).

Figure 50. Robotic drumming prosthetic, image from (Weinberg et al. 2020).

Motivations For Robotic Musical Instruments [7]

Artists and creators develop robotic musical instruments with different motivations in mind. While an external observer may interpret such efforts as a means toward automation, de-skilling, or replacing human musicians, they might be fueled by different goals and objectives in the creators' minds. In the mid-2000s, Ajay Kapur—by then a Ph.D. student at the University of Victoria's Music Intelligence and Sound Technology Interdisciplinary Centre and most recently the associate provost for creative technologies at California Institute of the Arts—interviewed some of the robotic musical instrument creators and inquired into their motivations. The spectrum of answers is interesting to explore; one artist expressed frustration in working with other artists and envisioned a band that they could play with forever. Another artist found the robotic musical instrument as a vehicle to overcome its allergy that barred them from playing any reed instruments (Kapur 2005). Researchers even developed a robotic prosthesis for an amputee artist to help them play drums again (Bretan et al. 2016) (Figure 50).

Another motivation behind robotic musical instruments is rooted in the quest to supplement the creativity of human agents and enrich the creative practice instead of merely mimicking and replacing the artist. An example of such an approach is to make robotic musical instruments that allow musicians to perform passages that human players cannot produce. Such systems can be envisioned as extension and elaboration of musicians' creative expression, not a vehicle to substitute human artists. The juxtaposition of a human artist with such capable machines "…establish[es] an environment in which human creativity can grow, thus, enriching human musical culture rather than replacing it" (Weinberg et al. 2020, 3).

An interesting example of this approach is demonstrated by Mohammad Jafari and Gil Weinberg (2021). They developed a robotic instrument to play santur and augment the musician's ability to play scores that were otherwise impossible to play. They designed a robotic arm for striking the strings with a mezrab.

[7] I intentionally left out the commercial efforts to create mechanical music-reproduction devices that are merely designed for music playback and/or entertainment. Instead, I focused on the artists, musicians, and makers who sought the creative aspects of robotic musical instruments.

Their demo video shows Mohammad and the robotic instrument playing the same instrument simultaneously but on two sides of the instrument (Figure 51).[8]

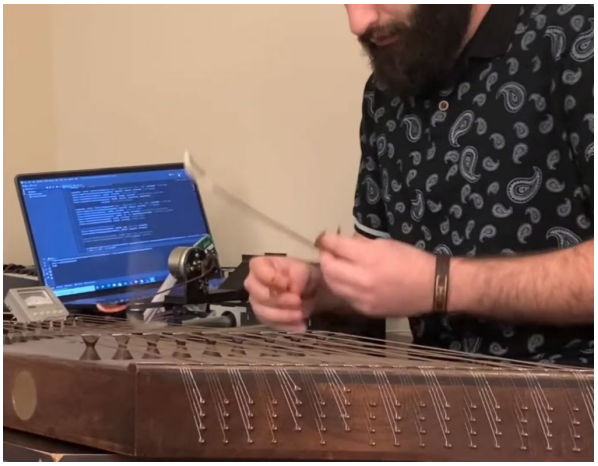


Figure 51. Santur Bot by Mohammad Jafari, screen capture from the demo video (Jafari 2021).

While this list is by no means exhaustive, it signifies the possible range of motivations and goals that fuel the endeavors of robotic musical instruments.

Question of authorship

When recorded sound first made its debut in the cinema in the 1920s, it started a wave of skepticism. It was perceived as machines taking over humans' role in creative practices. At the time, some musicians in the US formed a coordinated effort to stand against it under Music Defense League. In their ad campaign, the recorded music was referred to as "canned music" and "mechanical music," among other names. The face of the ad campaign against the recorded sound was a soulless humanoid robot singing, "O, soul of my soul, I love thee, …" (Figure 52).

Concerns about machinic music are still relevant today. Weinberg et al. cite multiple articles from Wired, Smithsonian, and New York magazine (Dutton 2012; Novak 2012; Morgan 2013) to reflect the popular media point of view toward machines getting into the realm of arts and creativity. This dissatisfaction with machinic music might be associated with different factors, notably the common belief that the growth of musical software programs will pose a credible threat to the human musicians' livelihood through automation and de-skilling (Rowe 2001, 4).

Automation and de-skilling are not the only topics that raise concerns toward robotic musicianship. There are other concerns that stem from the sentiment that mechanical machines, robots, or computers can be creative and make music on their own. The question of human authorship in machinic musicianship is addressed by Steve Coons. In his handwritten notes on machine creativity, he argues that even if the computer program generates novel examples in the style of a human expert, "… the creative act has already been performed" (cited in Cardoso Llach 2015, 62). Although he specifically utilizes music generation in his notes, his conceptualization of human authorship in the digital age can be generalized to other aspects, potentially music performance.[9] This case study aligns more with Coons's point of view on authorship and creativity.

[8] This setup became a source of inspiration for the demo of this case study which I will introduce later in this chapter.
[9] I discussed this matter in another publication (Bidgoli, Kang, and Cardoso Llach 2019).

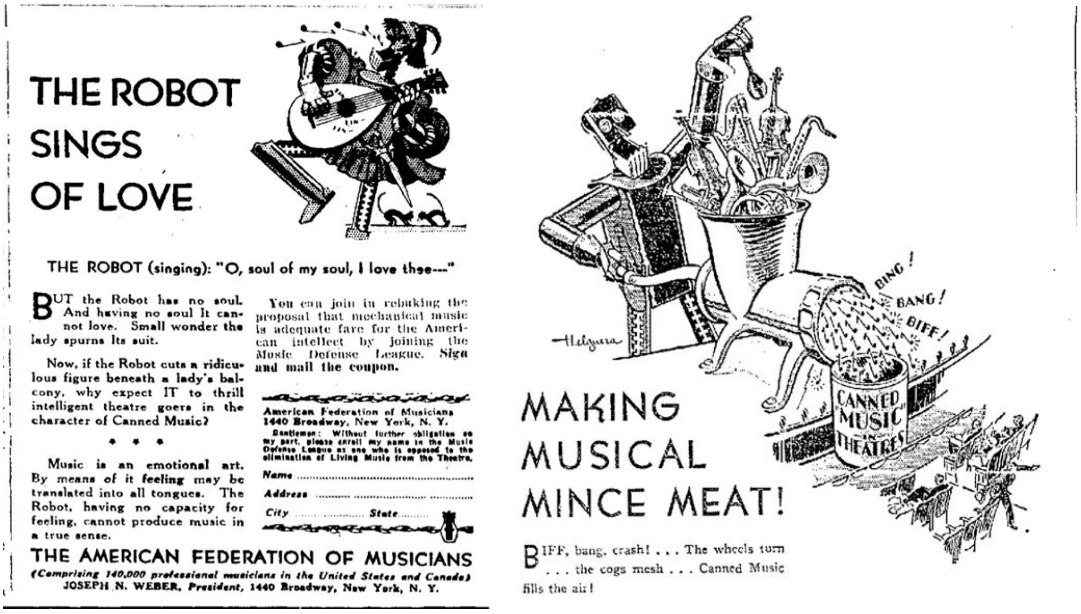


Figure 52. The soulless robot singing (left) and a robot grinding musical instruments (right) as published in the Smithsonian Magazine website (Novak 2012)*, originally published in Oelwein Daily Register on August 17th, 1930 (left) and Syracuse Herald November 3rd, 1930.*

5.1.4 Positioning the Study in the Context of Robotic Musical Instruments

Building on the ideas discussed through the lens of Weinberg et al., this study proposes a framework for making robotic musical instruments to augment an artist's capability to play santur. It is not destined to replace the artist, but it is an effort to explore the affordances of machine learning for musical toolmaking.

This study distinguishes itself from the previously mentioned projects by focusing on the artist, its direct inputs, and feedback and embracing them as the centerpiece of their own toolmaking process. Collaborating with the toolmaker, the musician can directly influence the process by providing data samples and subjective evaluation. The musician can improvise, repeat, modify, or remove samples to sculpt the dataset, one sample at a time, to gradually shape the final tool. Following these steps, this study distances itself from the conception of skill and knowledge as commodity/object and builds on the notion of skill as situated in the context of its practice.[10]

This study utilizes a dataset of six-degree-of-freedom motions provided by the musician as a vehicle to convey the specific idiom of the artist. The samples presented earlier in this chapter did not focus on this aspect. For instance, Santur Bot (Jafari and Weinberg 2021) utilizes a brushless direct current (BLDC) motor to actuate the strokes. The authors pre-designed various speeds and torque levels to produce different notes. Although their instrument is fine-tuned to create a desirable sound, it bears no variations based on the musician's idiom. As such, while the physical contextual factors were embedded in the system, the subjective measures were not integrated into the process.

Although this study is inspired by Rebecca Fiebrink's works on interactive machine learning for musical toolmaking, I explore new frontiers that have not been addressed in Fiebrink's work. The current research

[10] For further discussion on this topic, please refer to section 8.1. Skill.

aims at playing music directly on a musical instrument. Instead of discriminative models for interpreting the musician's signals, the machine learning model in this study is trained on user-generated samples and directly plays novel generated instances on the physical instrument. This aspect of research entails using generative machine learning models to produce high-dimensional mezrab strokes.

Similar to Fiebrink's Wekinator, I developed a meta-tool that can be adjusted and modified for specific use cases. However, the level of variations that comes with playing on a physical instrument requires developing a separate bespoke data collection and machine learning model for each use case. Accordingly, in contrast with Wekinator, which was later adopted by different creative practitioners, in my study, the generalizability of the meta-tool has not been a primary objective.

5.1.5 Scope and Abstractions

The scope of this study is narrowed down to developing a robotic musical instrument to play santur based on the samples provided by the participating musician. Accordingly, composing, generating, or improvising music is out of the scope of this study. The data modality is restricted to mezrab strokes, encoded as sequences of six-degree-of-freedom motions.

Elements of this study were designed to avoid abstractions while maintaining the similarity between the study setup and the artist's regular performances. For instance, we used an unmodified instrument and pairs of standard mezrabs for the data collection and demonstrations. The artist tuned the instrument to match hers, and the motion capture trackers were designed and integrated into the mezrabs to produce the least interference with her workflow.

However, some degrees of abstraction were inevitably introduced to the study. These abstractions were examined to ensure that their effects on the workflow were negligible. For instance, after analyzing the collected samples, it became clear that the mezrab motions are not significantly influenced by the note being played. But they depend heavily on the playing technique and the hand that plays them. Thus, the focus of the study was shifted toward techniques and each hand's strokes.

5.1.6 Participant Artist

During the summer of 2021, when this study was undergoing, the relaxation of COVID-19 precautions allowed me to change the gears from remote collaboration to in-person interactions. The new circumstances made it possible to collaborate with a musician in person, collect samples on a real instrument, and eventually showcase the instrument in an experimental demo session.

The search to find a musician started in early summer by reaching out to the community of Persian artists. An ideal candidate for this study was an expert musician with limited machine learning and computer science knowledge. I was introduced to a Persian musician in Pittsburgh, Mahtab Nadalian, who met these criteria, and invited her for an initial interview. Mahtab holds a B.A. degree in santur performance from the University of Tehran, School of Fine Arts, and an associate degree in audio recording technology. She has been tutoring santur students for several years. She does not have any background in machine learning or computer programming. Her profile was a perfect match for this study, and I invited her to join this study.

5.1.7 Santur[11]

Santur is a traditional Persian stringed musical instrument with common roots with the hammered dulcimer (Figure 53). The instrument is usually played while putting stationary on an inclined platform.

[11] Other alternative forms such as *santūr*, *santour*, *santoor* are commonly used. However, the latter being used mostly to refer to the Indian instrument. In Persian, the spelling is سَنتور.

Its trapezoidal frame seats in front of the musician and provides a flat framework for its 72 strings. The strings are made of two different metal alloys, 1) steel (silver-colored wires) for lower pitches which require more tension in the strings, and 2) brass or copper for higher pitches, which require less tension. An array of metal hooks on the left side of the instrument holds the strings, while the tuning pegs on the other side help tune each string individually (Figure 54, bottom left and middle).

Strings are organized in groups of four, each tuned to the same pitch, stretched horizontally over the instrument, and raised by a single wooden bridge called *kharak*[12] (Figure 54, top left). The special arrangement of the wooden bridges allows every other set of strings to get elevated on the same side (Figure 54, top-right). The strings raised on the right side produce lower pitches, while the ones raised on the left side produce the higher ones. Strings are beaten by a pair of identical wooden hammers, or mallets, called *mezrabs,*[13] held between the index and middle fingers and thumb. It is common to pad a mezrab's tip with a piece of felt to soften the impact.

Santur physical characteristics simplified the data collection implementation. The instrument remains stationary on a fixed inclined platform during the performance, eliminating challenges associated with pose tracking and transformation during the data collection and performance (Figure 55 and Figure 56). Moreover, the long and slim shape of mezrabs allows to accommodate motion capture trackers with enough space to incorporate variations on the trackers' distribution on each mezrab. In comparison, a hand-held instrument played by bare hands, fingers, or a small pick, would pose significantly more complex data collection and robotic implementation challenges.

[12] خَرَک

[13] For a comprehensive study of historic background and musical-acoustic analysis of santur, please refer to the Shahab Mena's studies at the University of Tehran (Mena 2006; 2010) as well as *The Grove Dictionary of Musical Instruments* under dulcimer (During, Hassan, and Dick 2001).

Figure 53. Top view of a Persian santur used in this study and its mezrabs. Note the motion capture spherical markers on the instrument and sticker markers on the mezrabs. Also, note the fine felt padding on mezrabs' tips, image by the author.

Figure 54. Strings elevated using wooden bridges (top row), tuning pegs, metal nails, and the vertical arrangement of strings (bottom row), images by the author.

Figure 55. Mahtab, photographed in her home studio, tuning her santur before a practice session, image by the author, edited for better visual quality.

Figure 56. General santur posture, image by the author, edited for better visual quality and removing branding signs.

5.2 Methodology

The case study design followed the principles of research by design, cast as design-led research. In this approach, design is both the object of study and the means of conducting the study through developing a project and inquiring about different design elements. The design process in this method is a pathway to create insight, knowledge, and products (Roggema 2016).

This method dictates a non-linear approach that is not destined for a solid solution. The outcomes are documentation of the process rather than a conclusive result. "It is an exploration and testing of ideas by means of design" (ibid, 11). In research by design, the design process should embrace the context, allow unexpected exploration, and develop knowledge that serves broader audiences. New knowledge emerges through constant interaction between researchers and the participants who communicate through non-textual materials.

To adapt the research by design for this study, I introduced a twist to the framework; instead of using conventional design mediums, i.e., sketches, drawings, and models, I opted for computational implementation and physical prototyping. Thus, instead of sketching design ideas, we developed iterations of the meta-tool, instead of making models, we prototyped a functional musical instrument to play santur, and instead of visual presentation, we orchestrated a performance demonstration.

Thinking and doing were intertwined in this process. We returned to the drawing board iteratively and revisited our decisions based on the new knowledge we gained at each step of this study. Every decision in the meta-tool development process and technical implementation detail was part of the research inquiry. We kept the design process as a conversation between me, the toolmaker, and the musician. We stayed flexible, explorative, and innovative, as we do in the design process, to explore different ways of addressing the research question. This allowed us to overcome the uncertainties of this research on the intersection of creativity and technology.

The research method in this study followed a three-stage process (Roggema 2016) (Figure 57):

1- Analysis (pre-design)
   In this phase, I focused on defining the elements of this study and curating the research question and hypothesis. At this stage, I reached out to musicians, creative computing experts, and technical advisors who could guide me in developing the study. Once *I* decided on the primary elements of this study, most notably, the musical instrument and the participating musician, *we* proceeded with a thorough analysis of the task, context, and potential avenues for research and design.
2- Projection (design)
   The projection was the core phase of this research method, where we aimed to address the research question using non-textual artifacts. Here, we worked closely to develop a meta-tool based on the *framework* and the knowledge gained through the analysis phase. We collaborated to create a robotic musical instrument and studied it in a demonstration. Throughout this phase, I iteratively revisited our previous decisions and revised them based on the musician's feedback, comments, and assessments. I documented this process as digital field notes, video, audio, and motion captures.
3- Synthesis (post-design)
   As the method's name implies, it delivers two sets of outcomes, 1) the artifact of design, in this case, the robotic musical instrument, and 2) the research outcomes, in this case, a new understanding of collaborative ML-based toolmaking. During this final stage, I continued the conversation with the musician to reflect on the process and crystallized the results from both aspects of the outcomes.

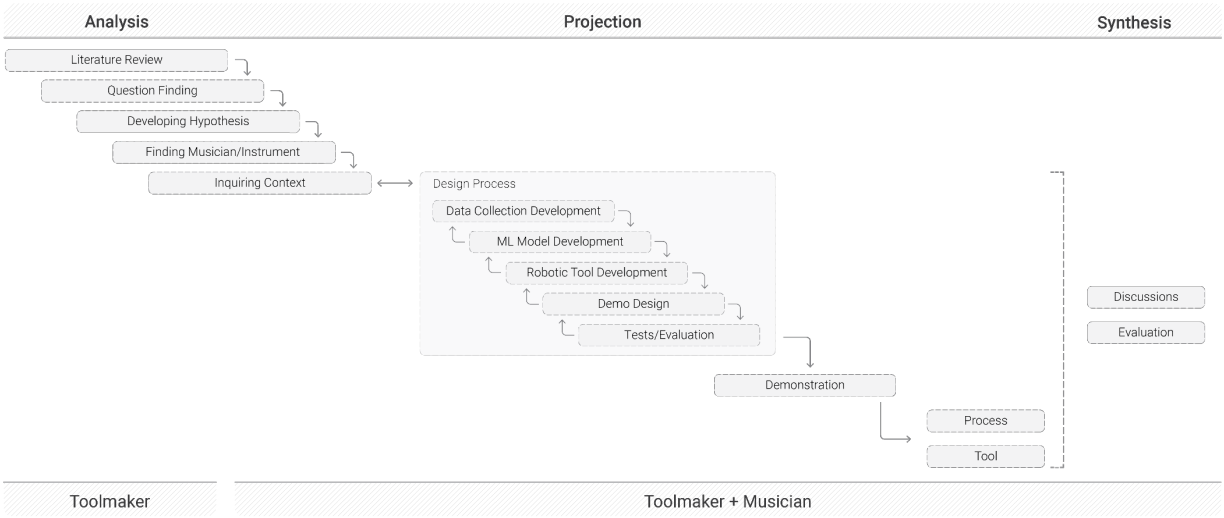


Figure 57. Research method schematic diagram.

Note on the Expectations

Throughout this process, our conception of the tool gradually evolved as we gained a better understanding of each other's workflow, santur, ML algorithms, and the available hardware. Learning from the first case study, this time, instead of making initial decisions by myself, I worked collaboratively with the musician from the beginning to shape a representation that reflects her perception of the tool. As I will elaborate in the following sections, during the early stages of this study, we were expecting to create a musical instrument to play santur with Mahtab's idiom. During the final stages of this study, we developed a mutual understanding that the tool should be able to play santur in a way that Mahtab could use in experimental performance.

Note on Toolmaker

As I mentioned earlier in Chapter 3, The Framework, in this study, I wore different hats at various stages of the study development. My background in computational design, toolmaking, and machine learning came into play at every turn of this study and informed my decisions. Inevitably I constantly had to exchange my hats, and I wore all three simultaneously at some points. However, to keep this document concise and clear, I will refer to myself as the "toolmaker," an umbrella term to include my computational design, toolmaking, and machine learning background.

5.3 Meta-Tool

The meta-tool allowed me and the musician to collaborate on the tool development. It enabled us to collect and process data, train the machine learning model, and implement and fine-tune our robotic musical instrument (Figure 58 and Figure 59).

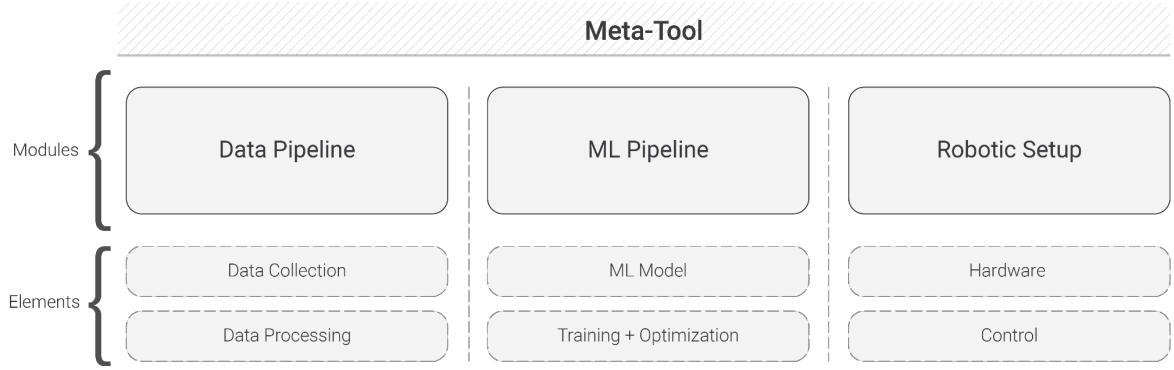


Figure 58. Meta-tool schematic diagram.

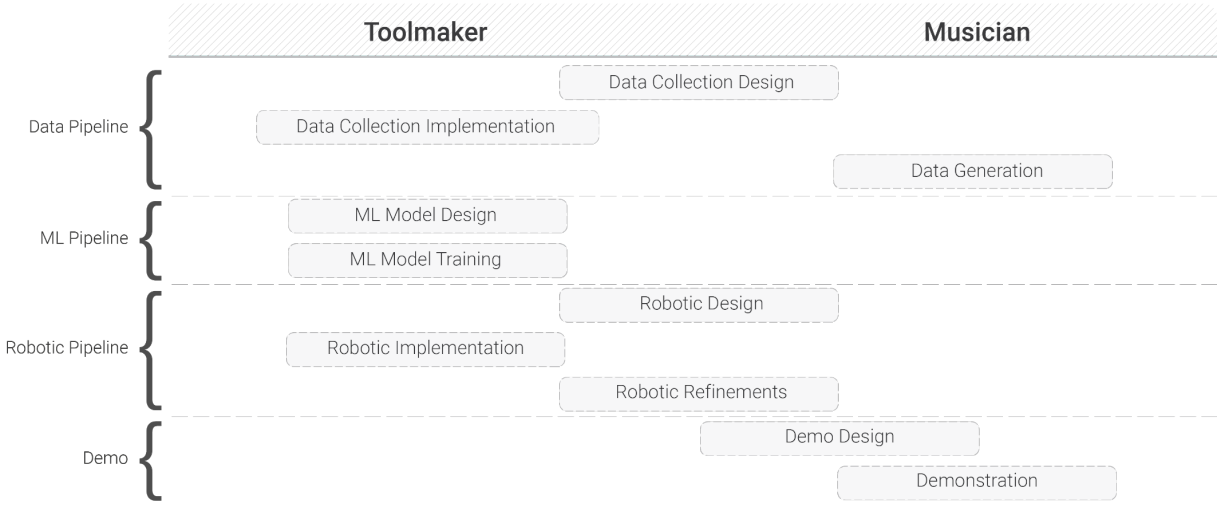


Figure 59. Contribution of the toolmaker and the musician in the collaborative toolmaking process.

5.3.1 Data Pipeline

The data pipeline collectively refers to all software and hardware pieces of the meta-tool that support data collection, data processing, and data curation activities. The pipeline details were developed in collaboration with the musician (Figure 59).[14] The data collection process was designed and implemented to have the least interference with her regular performance. For instance, the instrument used in the data collection was purchased based on the musician's recommendation, and each modification to the instrument was reviewed and approved by her.

The primary data modality in this study was motion, represented as sequences of six degrees of freedom data points. Here motion serves as a vehicle to convey the musician's idioms to the robotic musical instrument. I used an *OptiTrack Flex 13* motion capture system, a medium-volume motion capture camera system with sub-millimeter precision, and tailored software tools to process the captured data. The motion capture system recorded the musician's mezrab motions to record samples in her specific idiom of playing.

[14] Throughout this chapter, I will discuss several cases of such interactions and decisions that we collaboratively made to adjust our workflow with the technological framework and adjust the technological framework to match the musician's preferences. This form of bi-directional interaction was one of the most eye-opening experiences of this study.

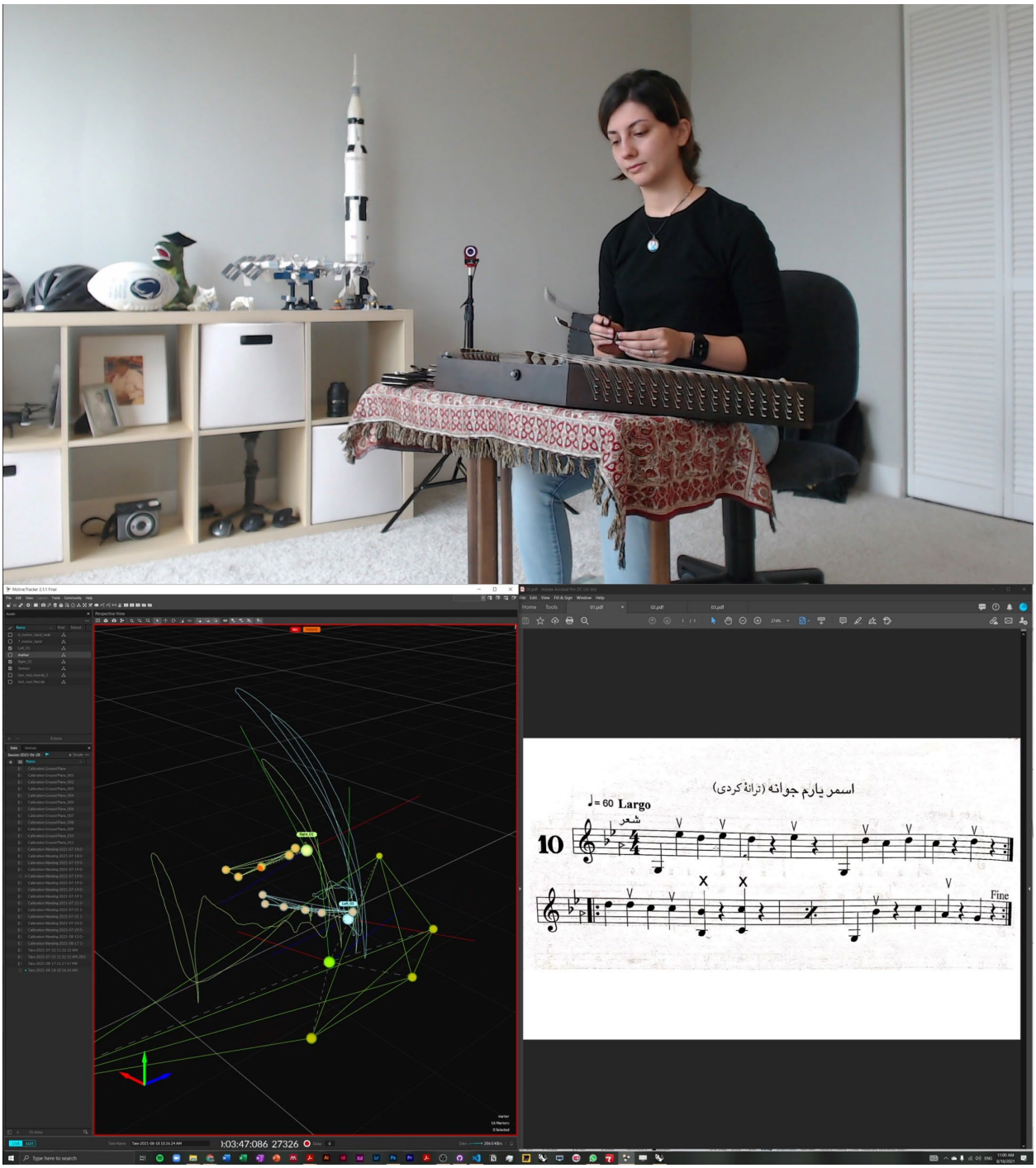


Figure 60. A snapshot from the video recordings during the second data collection session. The video feed (top) is accompanied by the Motive interface (bottom left) and notes being played (bottom right).

Data Collection Sessions

Between July and August 2021, we conducted three data collection sessions in my home office equipped with the motion capture setup (Figure 60, top). Each session lasted between 30 to 90 minutes and included initial setup, discussions on the session's objectives, data collection, and debriefing. Each session was recorded alongside a screen capture of the Motive software (Figure 60, bottom).

We dedicated the first session to communicating the technical points of the data collection schema to ensure that it matches her playing method. During this session, we identified some technical issues, i.e., problems with tracker marker arrangements and low sampling rate for specific techniques. Learning from the first session, I had the opportunity to adjust and improve the motion capture setup. During the next sessions, we iteratively improved the process based on the lessons we learned and the mutual understanding of each other's workflow. Eventually, we shaped a data pipeline where she could demonstrate samples of her choice on a standard santur, using a set of minimally modified mezrabs, in an environment like her at-home rehearsal environment.

In the first session, we had multiple rounds of data collection, each dedicated to one specific playing technique. The rationale behind this decision was twofold. First, it was a trivial task to manually segment and annotate the resulting data stream. This allowed me to use this subset of data as a toy dataset and sketch various data post-processing methods. Second, the special techniques we collected in this phase are not as common as the basic strokes. Having them collected separately helped us make a more balanced dataset. During the second and third sessions, she curated a library of mezrab strokes in her idiom of santur playing. The samples were provided in the context of seven short songs she has chosen from santur exercise books based on her experience teaching santur to novice learners. In total, we collected over 1400 samples of mezrab strokes played different notes, with different techniques, played by both hands

Motion Capture Setup

A motion capture system tracks a given object by locating a series of markers installed on the object. Designing a reliable array of trackers is the first step in a motion capture pipeline. Trackers can be active, which uses special LEDs to emit light, or retro-reflective (passive), which are coated with reflective materials. Passive trackers come in different geometries, such as spherical, circular, and square-shaped. The trackers must stay fixed on the object to ensure consistent and reliable capture.

The main factors influencing the quality of capture are the number of trackers, their placement on the object, and their physical properties (size, shape, and reflectivity). The motion capture system needs to see a tracker with at least three of its cameras to triangulate its position in space accurately. Larger trackers are easier to spot over longer distances. However, they are unsuitable for delicate motions—i.e., facial tracking and fine hand gestures. Smaller markers can improve accuracy. But they also introduce other challenges, most notably, the higher chance of occlusion and loss of tracking over long distances from the cameras. Sphere-shaped trackers can be tracked from a wide range of angles; in contrast, flat trackers, made of reflective tape, are best tracked when directly facing the motion capture cameras.

The arrangement of trackers is also a critical factor in motion capture accuracy and consistency. The motion capture software uses the spatial relationship between the markers to identify the tracked objects from each other and calculate their locations and orientations in the 3D space. Motive,[15] the preparatory OptiTrack software package, needs at least three trackers to form a *rigid body* that can be tracked. Increasing the number of trackers and distributing them at larger distances from each other and in different planes will increase the chance of accurate measurements. Motive also uses the trackers as a pseudo-fingerprint to automatically identify rigid bodies. When there are multiple objects in a scene, i.e., two mezrabs, it is essential to have different tracker placements on each object to allow the software to distinguish them from each other (OptiTrack n.d.).

Early tests aimed at comparing various arrangements of markers using both spherical and flat trackers. While the tracking consistency with spherical markers was higher, installing three or more trackers was

[15] Motive is the optical motion capture software, developed by OptiTrack (OptiTrack 2022).

impossible without adversely affecting the physical features of the mezrabs. Santur mezrabs are feather-weight wooden pieces with very slim stems (Figure 61, left), and we wanted to find a solution that could maintain these characteristics (Figure 62). This eliminates the use of spherical markers as an option since their weight negatively impacts the user experience.

One solution for this issue was to design and fabricate custom-made mezrabs with thicker stems and enough flat surfaces to install spherical or flat markers (Figure 61, right). As a benchmark, three spherical trackers were installed on a custom-made mezrab and tested (Figure 63, left). When these prototypes were presented to the musician, she found them usable but far from ideal. Testing them for a few minutes, she highlighted the different sound signatures of these prototypes, stemming from their thick head compared with the delicate heads of standard mezrabs. Accordingly, these prototypes were only used as the benchmark to assess the trackability of other alternatives.

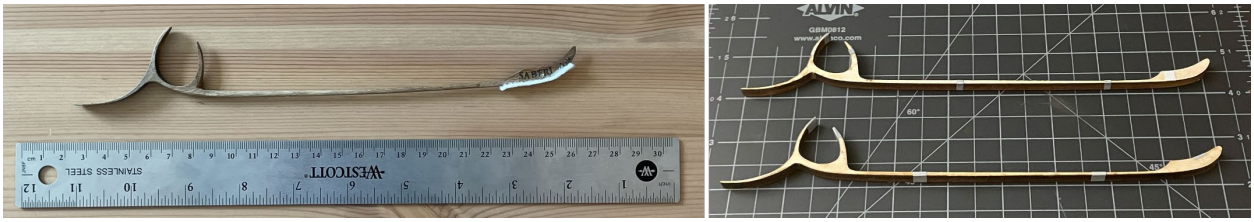


Figure 61. A standard mezrab (left), custom-made mezrabs cut from 1/4" balsa sheet, with retro-reflective flat markers (right).

Figure 62. The musician comparing the markers on her own mezrab (left) and her unmarked mezrab(right) by holding them in a standard idle position during an initial data collection session.

Switching to flat trackers slightly reduced the consistency and accuracy of tracking. However, an increase in the number of trackers could compensate for this issue. The latest iteration of mezrabs equipped with flat trackers was on par with the one with spherical ones (Figure 63, right) while keeping the feather-weight status of the mezrab intact.

The next iterations were based on a pair of mezrabs crafted by an expert in Iran (Figure 65, right). The trackers were cut from a retro-reflective one-sided tape and attached to the mezrab in various locations. The tape was cut into $^1/_8$ inch and $^1/_4$ inch strips and wrapped around various spots on each mezrab. Patches of $^1/_4$ inch squares are added to the mezrabs' heads to gain maximum distance between the markers. The pattern of trackers on each mezrab was designed slightly differently, allowing the Motive software to distinguish between the two.

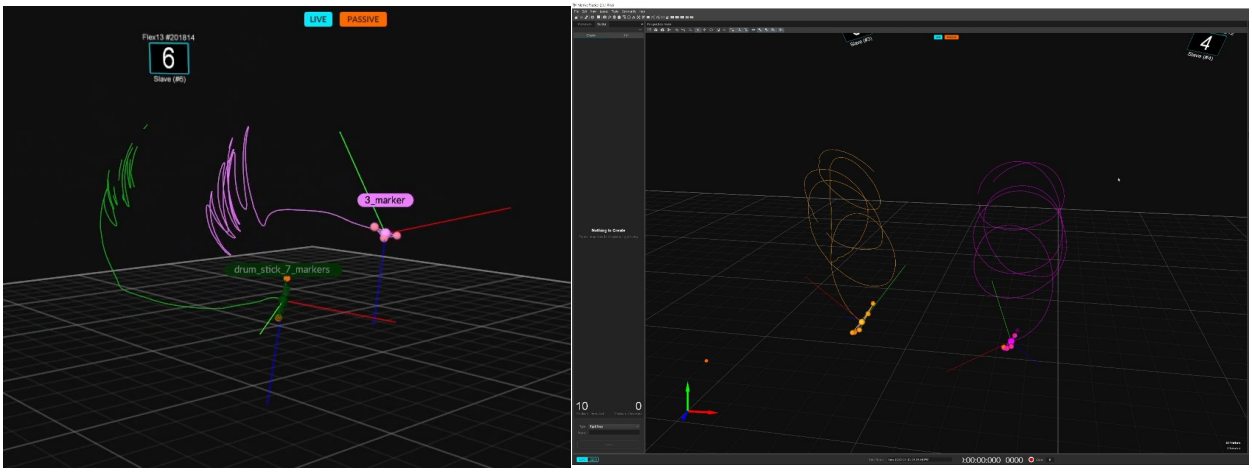


Figure 63. Left: Capture results for two mezrab tracking designs. Spherical trackers (magenta) vs. flat trackers (green). The missing segments in the green trace show where the cameras are missing the trackers. Right: Revised flat trackers comparison: six trackers made of thin strips and square patches (orange) vs. five thicker trackers (magenta).

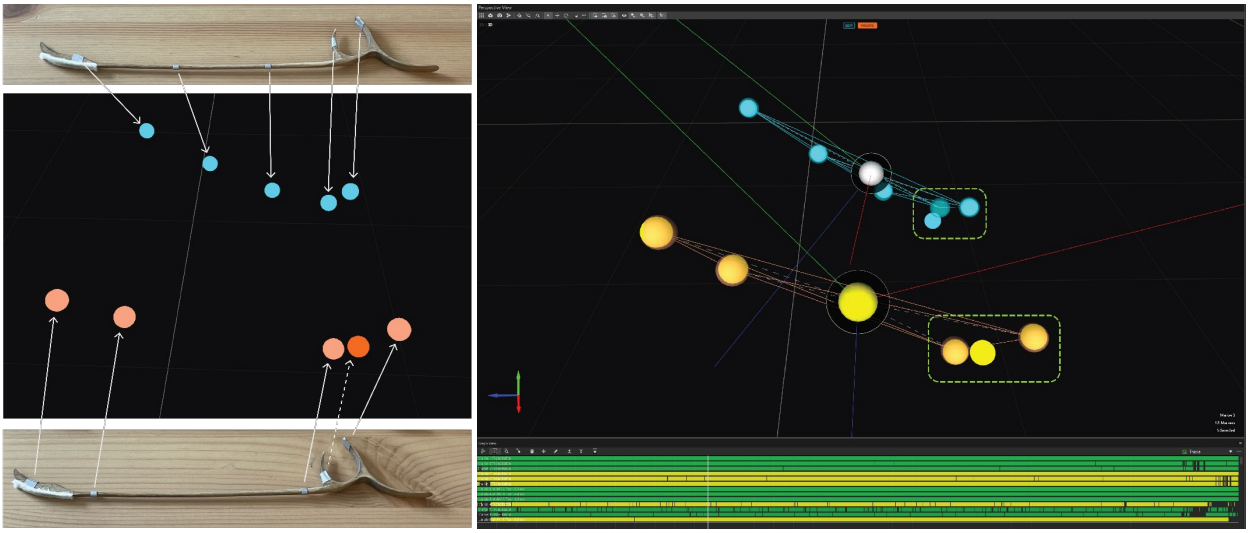


Figure 64. Flat retro-reflective trackers on the mezrab (left), corresponding registered ones in the Motive app (right).

Figure 65. Early retro-reflective marker arrangement (left) and revised version (right), notice the slight differences between the marker arrangement on each pair of mezrabs.

Figure 64, left, shows these trackers as they were registered by Motive.[16] During the early tests, this combination of trackers resulted in smooth and consistent tracking. Figure 64, right, depicts a snapshot of the motion capture software. The lower panels' green and yellow bars visualize the tracking history on each tracker. The blank marks on each track indicate the missing data points.

A closer look at the tracks in Figure 64 shows that the ratio of missing data points is negligible compared with all collected points. The tracks with the highest rate of missing samples correspond to the markers on the rear handle close to the gripping point. This might be due to a higher chance of occlusion behind the musician's fingers or opposite hand. In the final marker arrangement, these markers were omitted in favor of more markers on the stem of mezrabs (Figure 65, right). Motive registers the missing data points as empty placeholder rows marked with the corresponding time stamp. This makes it easier to find them in the post-processing phase. These missing data placeholders later get filled to bridge the gap between the previous and next available data point.

Another observation in the initial data collection sessions was the problem with trackers located close to each other. The motive algorithm could not confidently identify closely located trackers on the same mezrab. In the revised versions, the spaces between markers were increased. The musician did not face any issues after this design refinement and could play her notes without adjusting her finger placement.

Based on these tests, feedback from the musician, and several iterations, we settled on a pair of standard mezrabs with flat markers (Figure 65, right). These mezrabs were almost identical to those she used routinely (Figure 65, left). Nevertheless, the musician pointed out a remarkably interesting point about a well-crafted mezrab: it is suggested that a mezrab should have consistent color across its length to prevent vision fatigue. From this point of view, the flat trackers on the stem of the mezrab may introduce some fatigue in long-term use. Using this mezrab for short data collection sessions renders this fatigue negligible.

Data Processing

Decisions on data cleaning methods, filters, and data segmentation significantly influence the following steps down the stream. I, as the toolmaker, took the most responsibility in this phase. However, we communicated the process to ensure data processing and annotation correctness. We assigned multiple labels and vectors to each data point: 1) mezrab motions (including position and rotation), 2) playing hand, 3) note being played, and 4) technique.

[16] In the snapshot, captured from the first data collection session, one tracker is not correctly labeled to the mezrab, marked with the dark orange, and dotted line.

The Motive software provided algorithms and tools to clean the motion data and fill the gaps or missing trackers. It could also automatically identify the left and right mezrab and organize the motion data in separate channels. The other features, notes and techniques, were manually annotated and added to the dataset later.

Each recorded session had to be broken down into single mezrab strokes associated with corresponding labels to prepare them for the training pipeline. Thus, each motion capture stream was broken down into smaller sections corresponding to each song and stored as .csv files.[17] The filtering method was implemented in Python 3 using the GH_CPython add-on in Grasshopper. The .csv files were fed into the Python script, and the outputs were passed to Grasshopper components in real-time to generate 3D visualizations which could be easily interpreted (Figure 66).

While all the data pre-processing steps were implemented in Python, it was the easy-to-use and flexible visualization tools of Grasshopper that allowed me to swiftly inspect the data, find edge cases, and update the Python code to address them. This real-time feedback loop allowed me to adjust the hyperparameter easily and fine-tune the process beyond what I could achieve by other methods, i.e., Python static plots, Jupyter Notebooks, or customized interactive data dashboards.

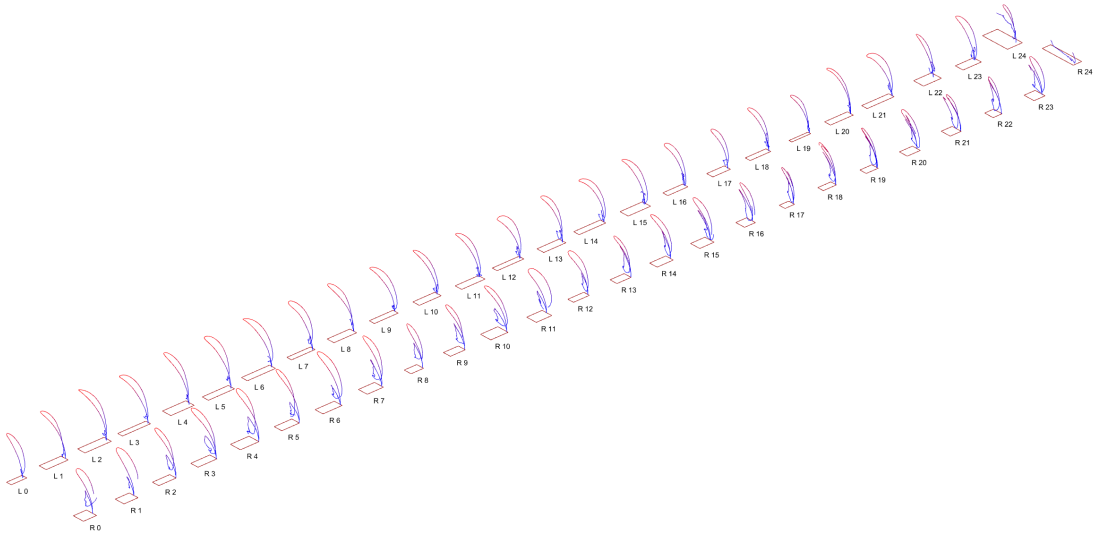


Figure 66. Fifty motion samples, recorded during the first data collection session. Each sample is illustrated as a curve interpolating across 272 points. Each point represents a frame of motion capture stream (1.94 sec). The red rectangles show the XY boundaries of each motion.

[17] Comma-Separated Values, more commonly referred to as csv, is a common format to store tabular data in plain text.

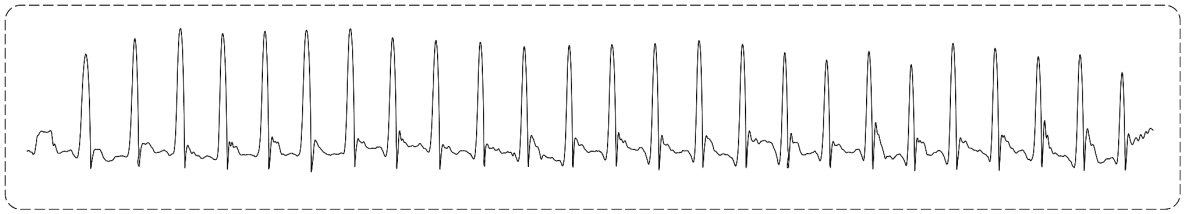


Figure 67. The z value for one mezrab over 60 seconds, playing the same note. Notice the repeated patterns and slight variation over time.

Data Segmentation/Annotation

The first step in the data processing pipeline was to break down a stream of data into single motions. This process begins with finding the frame in each stroke where the mezrab hits the strings—the *touching frame* from here on—and using it to slice each mezrab stroke. The touching frame occurs where the *z* value of a data point is at its minimum. However, finding the exact touching frame is not straightforward. Due to the nuance motions of the musician's hands, several local minimums between two touching frames might be flagged as false touching frames.

Upon further analysis of data visualizations in Grasshopper, a repeated pattern emerged: the artist always moves the mezrabs higher than the resting position right before moving them down rapidly to hammer the strings. This vertical hike was the most pronounced and consistent among all motion signals before each touching frame (Figure 67). Accordingly, instead of searching for the touching frame, I shifted the focus to spotting the peak points on the z channel to find the touching frames that proceed them accurately.

Several hyperparameters were used to filter out the false high peaks and false touching frames. First, two hyperparameters, *tresh_h* and *tresh_l,* set the threshold for the peak amplitude. These two parameters helped eliminate any peak that is not strong enough within a specific neighborhood. One parameter, *dist*, is solely dedicated to finding the true top peaks by limiting the minimum acceptable distance between two high peaks. The other hyperparameter, *peak_dist*, is responsible for removing false low peaks by setting the maximum distance between a high peak and the following touching frame. The last hyperparameter checks the ratio between *z* and *y* to distinguish between touching frames and the short rest after that. This hyperparameter works because true touching frames are always followed by some rapid z motions. In contrast, the resting phases are usually followed by swing motions in the horizontal plane. The value for each hyperparameter was determined through a series of tests and observations in the Rhinoceros 3D/Grasshopper environment (Table 4, Figure 68, and Figure 69).

Table 4. Data filtering hyperparameters.

Hyperparameter	*Effect*	*Value*
tresh_l	Eliminating weak local minimums (false touching frames)	0.289
tresh_h	Eliminating weak local maximums (false high peaks)	0.408
dist	Eliminating the false touching frames which are too far from a high peak	15
peak_dist	Eliminating the high peaks that are located too close to each other	50
motion_fix_length	Defining the number of frames in each motion sequence	30

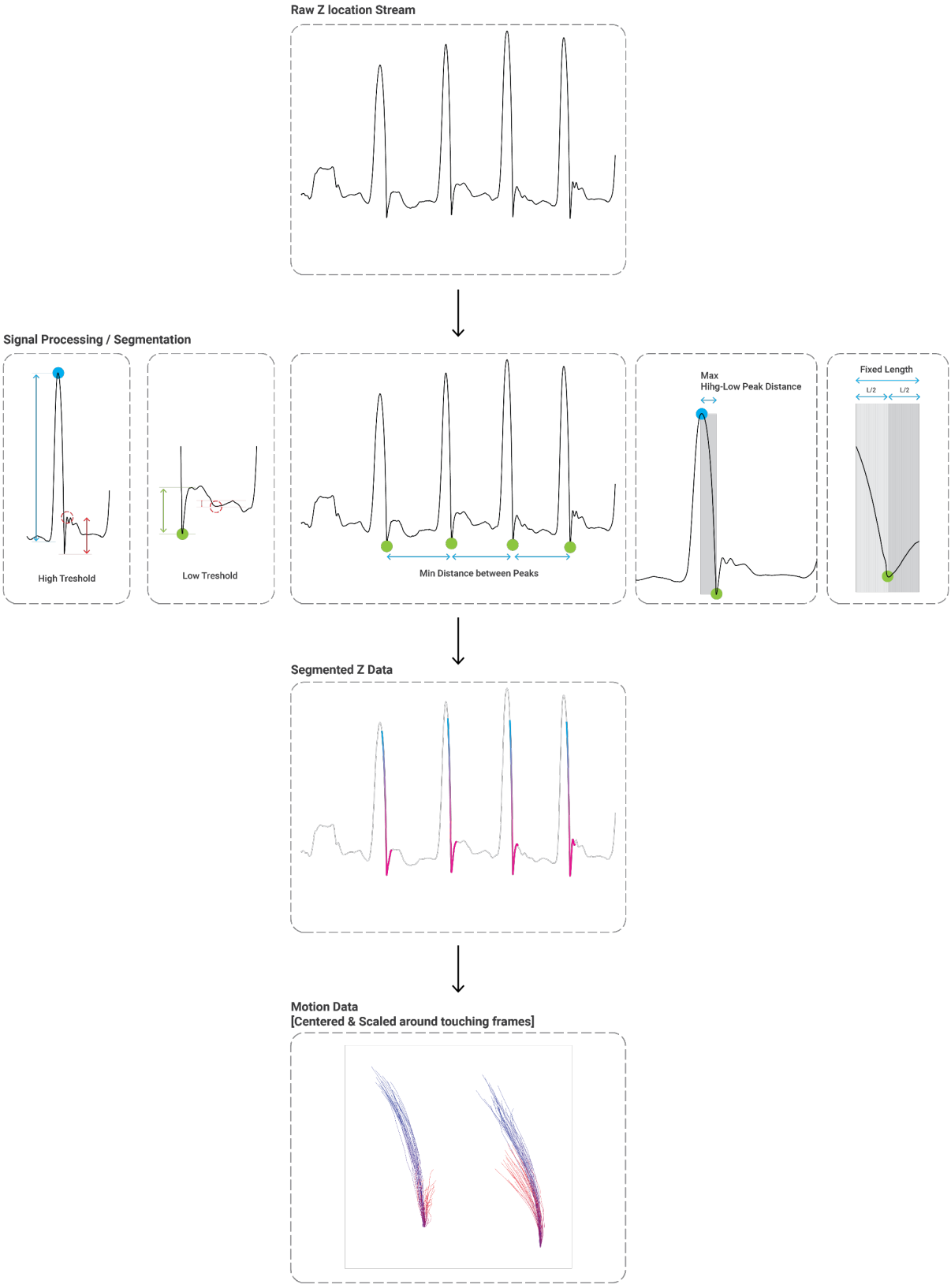


Figure 68. Signal segmentation process.

Figure 69. Results of data processing and segmentation method applied to streams of motions.

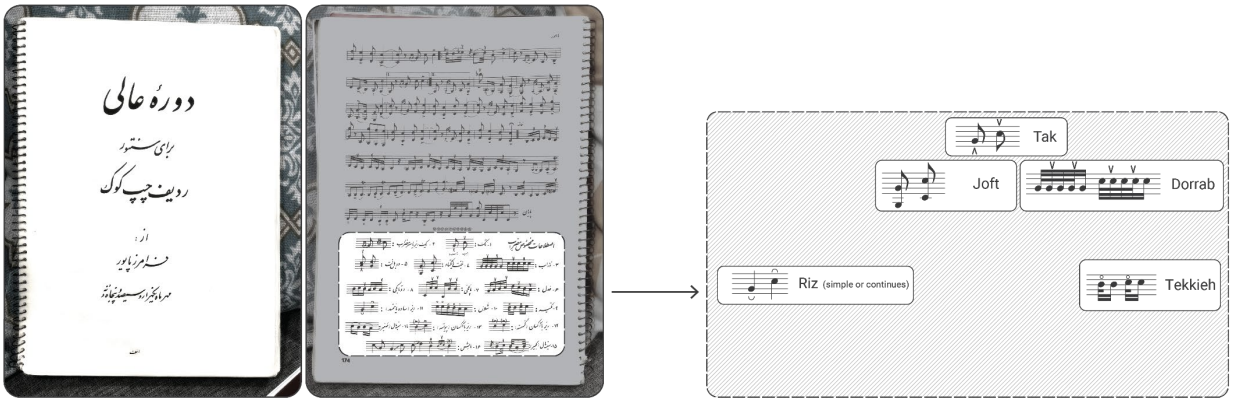


Figure 70. Santur playing techniques as presented in Faramarz Payvar's Santur workbook. Cover page (left), techniques (middle), techniques used in this study (right) (پایور 1359)

We collected samples for various techniques of combining the mezrab strokes with two hands: simple right or left, *Dorrab* (right, left, right), *Tekkieh* (left, right), *Joft* (left and right at the same time), *Riz*[18] to name a few (Figure 70). We found riz[19] and dorrab the most challenging techniques as they are fast-paced or follow unequal timing between their elements. In such cases, the importance of hyperparameter fine-tuning was even more pronounced.

In a stream of data, most frames represent the period in which the mezrab is in its resting position before or after playing a note. These frames bear non-essential information and can be trimmed from each sequence's beginning and end. Thus, a fixed length is set as the hyperparameter for all motions, with ½ of the frames lined up before the touching frame and the rest after—including the touching frame (Figure 71). As the last touch, any sequence of motion with low variance in z was also discarded as it most likely represents the resting position of a mezrab rather than a strike.

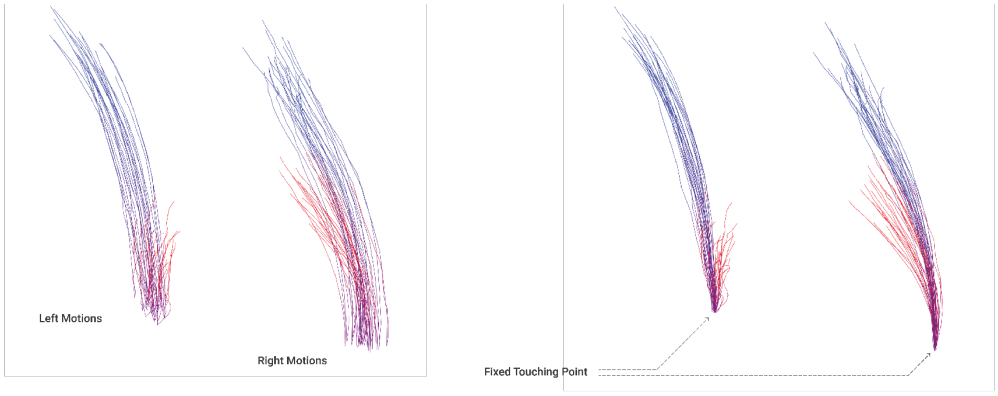


Figure 71. Samples from the left and right hand, distributed on the real touching points (left pair) and moved to a fixed touching point (right pair). The red color indicates the beginning of the sequence.

[18] Dorrab: دُرّاب, Tekkieh: تِکیّه, Joft: جُفت, Riz: ریز

[19] Riz technique is the repetition of the same note in a rapid pace, like tremolo. The number of notes being played depends on the performer's convenient pace. In the scores we chose for data collection, each riz note was denoted for 7-9 strokes. Mezrab's motions in this technique are quite fast to the point that the number of registered steps per motion was not adequate for motion processing or automatically isolating it from the neighbor notes.

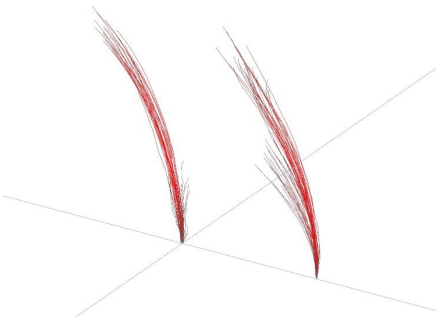


Figure 72. Analysis of mezrab's velocity, red parts represent the fastest sections of each motion. Note that the time of each motion is fixed. Accordingly, the longer curves represent faster motions.

A closer visual analysis of recorded motion revealed some interesting facts. In all cases, the speed of mezrab moving down toward the strings is faster than retrieving. While both hands gain similar speed to hit the string, there is a significant difference between the left and right hands regarding retraction (Figure 72). The right-hand bounces back faster than the left hand, and the variation in its final position is more significant. Such pronounced differences render the classification based on hand an easy task. Once the segmentation of motions from the data stream was done, each sample was formatted as a 20×9 vector, representing the 20 poses in space, each defined by a point and two vectors (Figure 73).

$$\boldsymbol{X_t} = [\boldsymbol{x_0}, \boldsymbol{x_1}, \dots, \boldsymbol{x_{19}}]$$

$$\boldsymbol{x_i} = [px_i, py_i, pz_i, v_x x_i, v_x y_i, v_x z_i, v_y x_i, v_y y_i, v_y z_i]$$

The point component of each motion is measured in meters, following the Motive application units. Accordingly, these values range between positive and negative, possibly beyond the range of (-1, 1). Meanwhile, the vector components do not have any units and have different ranges. To facilitate the learning process, some data pre-processing methods were applied: 1) the strokes were moved in 3D space to have their lowest point located on the origin of the coordination system, 2) the stroke data was scaled on each feature to map them between 0 and 1 (Figure 74).

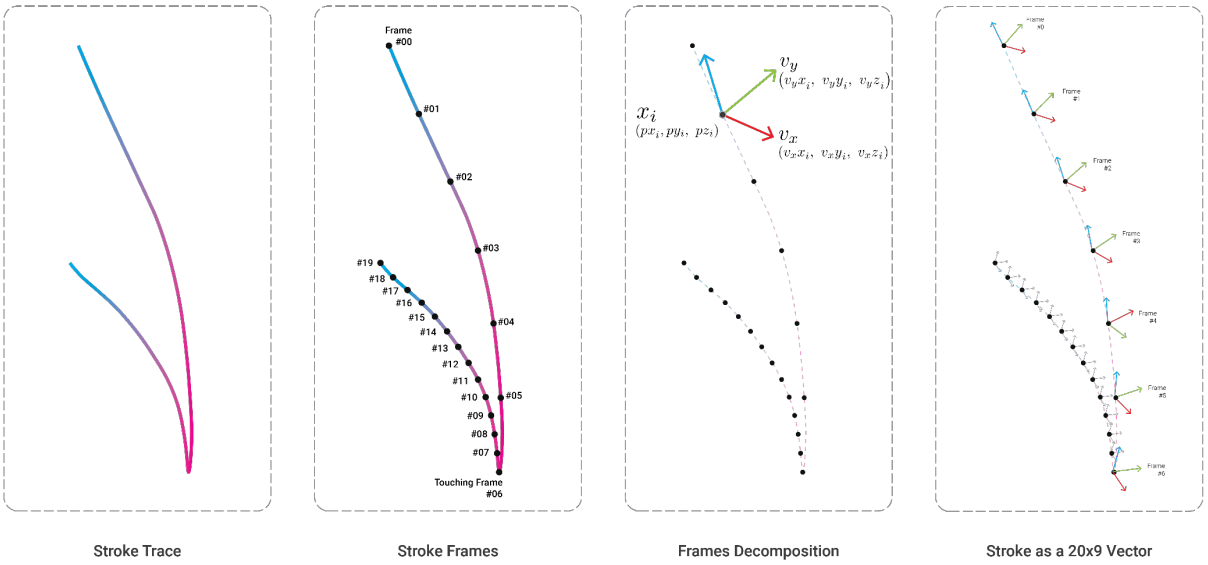


Figure 73. Stroke representation as a 20×9 vector.

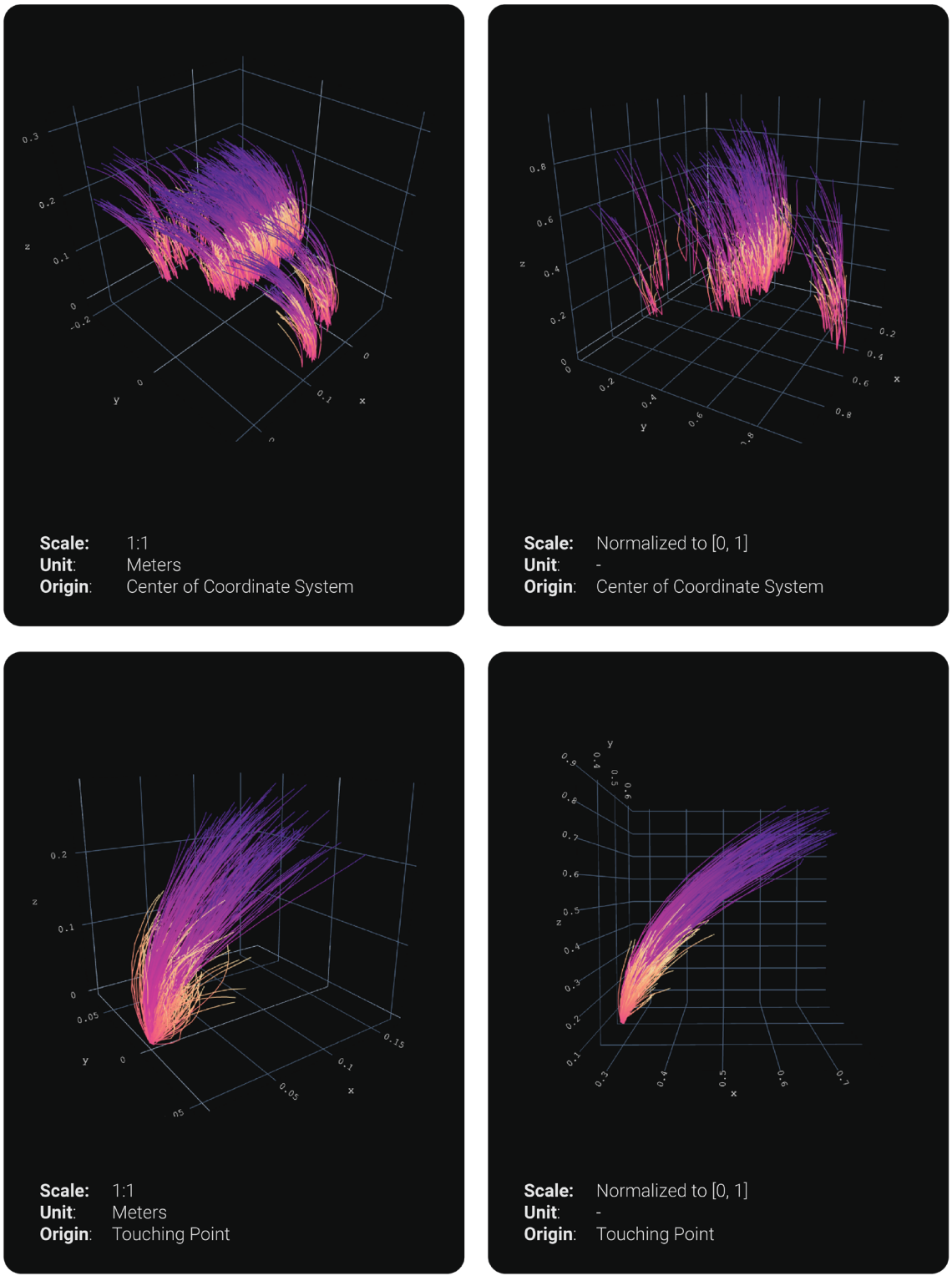


Figure 74. All recorded motions, in the original scale (top-left), scaled to 0 and 1 (top-right), centralized based on the touching point (bottom-right), and scaled-centralized (bottom-left). The gradient signifies the sequence (purple represents the early frames.

5.3.2 Machine learning backend [20]

Like the previous study, the machine learning backend was built around the Variational AutoEncoder architecture. The sequential nature of motion data renders it suitable for Recurrent Neural Networks (RNN), i.e., Long Short-Term Memory (LSTM). However, CNNs have also been tested for sequence-to-sequence tasks in combination with RNNs or pure CNN encoder-decoder architectures. What makes CNN layers a suitable choice for this study is the temporal-spatial relationship between the data points. Li et al. proposed a CNN-based encoder-decoder architecture for human dynamics in 3D, a complex spatial and sequential data type (2018). They argue that the hierarchical structure of CNNs renders them powerful tools to capture such temporal-spatial patterns, which RNNs cannot properly learn. Moreover, as discussed in the previous chapter, RNNs are slower in the training phase and less efficient compared with CNNs. Accordingly, I chose to use CNN layers to capture the spatial and temporal features of the dataset.

While training the machine learning model, I intentionally left the model to overfit the training dataset. In many fields of machine learning, this practice is frowned upon. However, in this specific scenario, we wanted to have the model overfit on the relatively small and quite biased dataset. There was no intention to make a generalizable model. In contrast, it was trained to serve one specific task: generating novel samples based on the dataset collected from one specific person.

In this architecture, the convolutional filters process each pose at time step t and its neighbors at $\{t - (2n+1)\ \ldots, t + (2n+1)\}$, where $2n+1$ is the convolution filter size, to find the correlations between them.[21] The one-dimensional convolutional filters span nine features and stride over the 20 frames in each stroke (Figure 75).

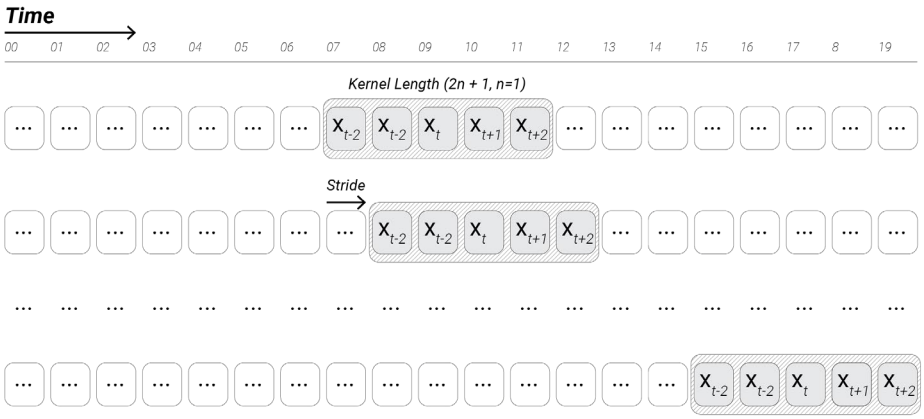


Figure 75. One-dimensional convolution filter striding over the sequence of data with a stride value 1.

Encoder-Decoder Models

A Conditional Variational AutoEncoder (C-VAE) model with One-Dimensional Convolutional layers in both the encoder and decoder was used for this study (Figure 76). The encoder uses multiple CNN blocks to map the input motion into a latent representation. Each CNN block in the encoder includes a Conv1d

[20] For an in-depth discussion on the machine learning technical details, please refer to Appendix I: Conditional Variational AutoEncoders.

[21] For more complex and diverse datasets, finding a solution for both short-term and long-term dependencies is essential. For instance, Li et al. designed different CNN layers for short-term and long-term mapping and concatenated their latent representation (2018). For this study, the dataset has significantly smaller variances and does not require such complex solutions.

layer followed by batch normalization, ReLu activation function, and a Dropout layer. It is worth noting that the input dimension is of shape 20×9, but the latent dimension could be as high as 256. In this scenario, the main purpose of having a latent representation is not mere dimension reduction; its value resides in the generative capabilities of the model.[22]

The decoder model's role is to collect the latent vector z from the variational latent space and the conditioning signal to reconstruct the motions. Throughout the model design stage, it became clear that stacking any layer on top of the deconvolutional layer will adversely affect the decoder's ability to reconstruct the motions. Accordingly, the decoder model only uses PyTorch's convolutional transpose 1d layers.

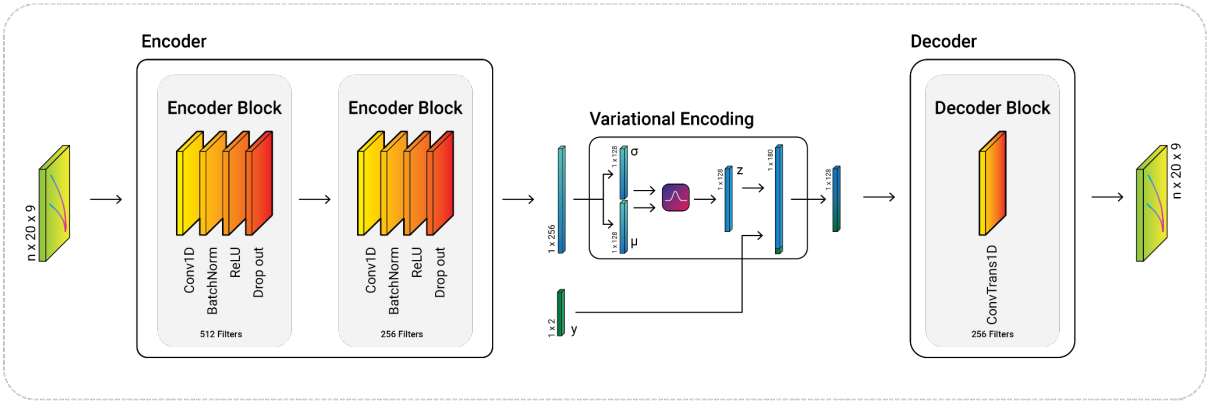


Figure 76. The schematic diagram of the C-VAE model.

Loss Function

The loss function in VAE architecture consists of two elements, 1) reconstruction loss and 2) KLD loss. The former calculates the difference between the input data and the model's output, using L1 or L2 loss. The KLD loss determines the difference between the learned latent distribution and a multivariant standard normal distribution. It regulates the latent distribution by penalizing the model if the learned latent distributions differ from a normal distribution with a mean of zero and standard deviation of one ($\mu = 0,\ \sigma = 1$).

It is also possible to improve the model's behavior by fine-tuning the balance between the reconstruction loss and KLD loss by adding a weight value to one of them, usually KLD. Applying higher weights to KLD will increase its impact on the learning process and results in a more regulated latent space, with the tradeoff of reducing the reconstruction accuracy. I tested different configurations of reconstruction loss—L1 and L2—, deduction method—sum and average—and weights of KLD loss to optimize the model.

Hyperparameter Fine-Tuning

In the previous study, the SecondHand, I provided the participants with an almost-fixed machine learning model. It was an informed decision to accommodate a specific expert user-ML toolmaking expert

[22] The encoder does not map the motions into an unregulated latent space, as per variational autoencoder definition, the encoder maps the input data over a series of normal distributions. For an in-depth discussion on the VAEs, please refer to Appendix I: Conditional Variational AutoEncoders.

interaction. In this study, I took the same approach and tailored the model based on the dataset that the musician and I curated during the data collection phase.

During the fine-tuning process, different values for variables such as kernel size, number of filters per layer, dropout rate, depth of model, and latent space dimension were tested. I utilized an automated hyperparameter optimization tool to accelerate ML model development. Hundreds of AE, VAE, and C-VAE models were tested and fine-tuned using WandB's Sweep toolkit to find a range of optimal models (WandB 2020) (Figure 77).

Eventually, a simple yet efficient model was selected; a C-VAE model with two layers of 1D-CNN blocks in the encoder and the decoder models and a latent space of size 256. Training the model for 150 epochs took around 27 seconds, and it took 22 seconds more to get into 300 epochs (Figure 78).[23] However, the results on the 150 epochs were quite satisfying, and further training was not necessary (Figure 79).

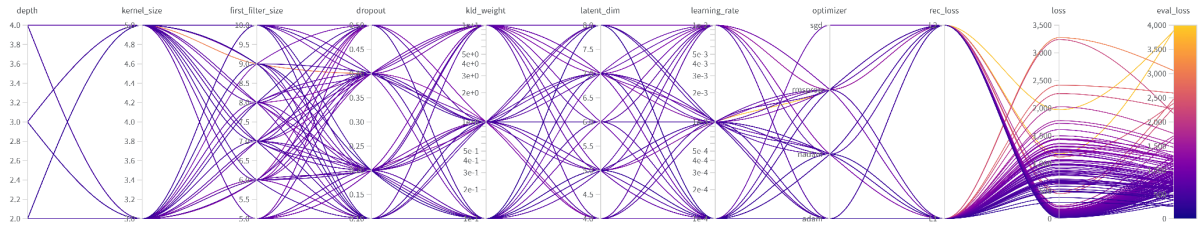


Figure 77.Hyperparameter fine-tuning for the C-VAE model: 200 different combinations of parameters were tested. Failed cases and models with eval_loss over 4500 are omitted for clarity.

Table 5. Range of hyperparameters for optimization

Hyperparameter	*Tested values*	*Used for the model*
Depth of encoder/decoder model	2, 3, 4	**E: 2, D: 1**
Filters in the first encoder layer	$2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}$	$\mathbf{2^9}$
Latent dimension	$2^4, 2^5, 2^6, 2^7, 2^8$	$\mathbf{2^8}$
Kernel size	3, 5	**5**
Dropout rate	1e-1, 2e-1, 3e-1, 4e-1, 5e-1	**1e-1**
Reconstruction loss function	L1, L2	**L1**
Reconstruction loss function reduction	Sum, Average	**Sum**
KLD weight	1e-1, 1, 1e+1	**1e-1**
Learning rate	1e-2, 1e-3, 1e-4	**1e-4**

[23] All the trainings were conducted on a local machine, using a NVIDIA 2080Ti GPU.

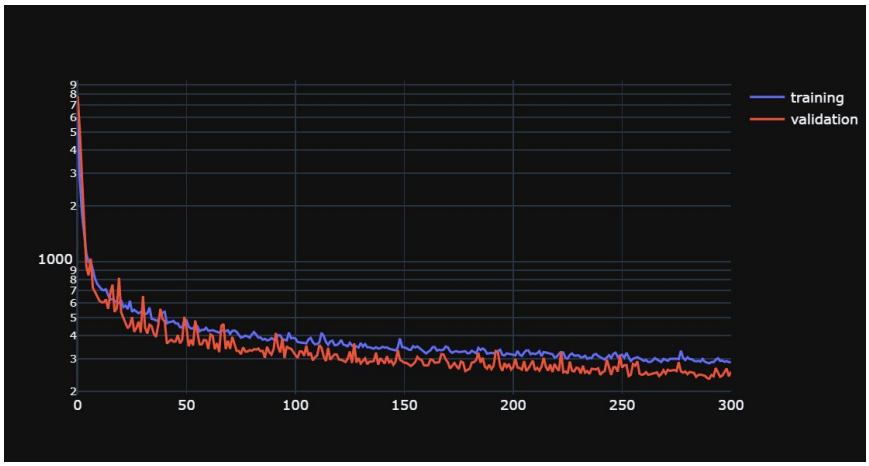


Figure 78. Training loss vs. validation loss for the C-VAE model: the x-axis represents the number of epochs, and the y-axis (logarithmic scale) represents the loss (weighted KLD + reconstruction loss).

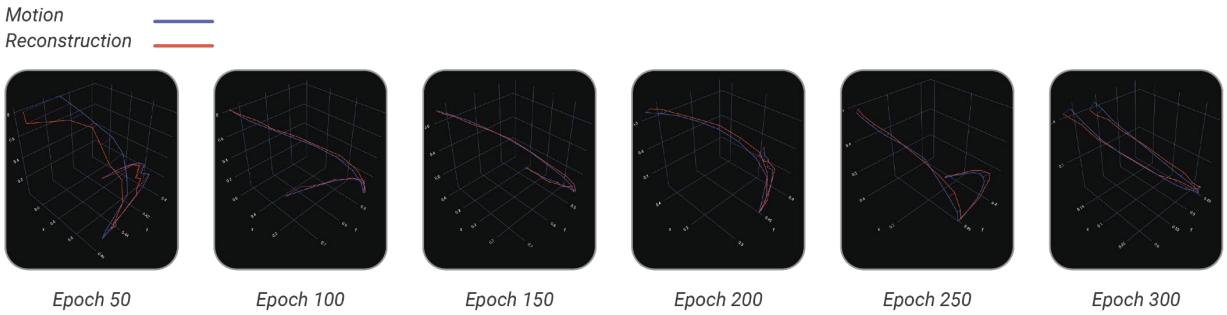


Figure 79. Reconstruction of the validation samples during the training process over 300 epochs. Plots depict a random validation sample. Only the three first values representing the location are visualized.

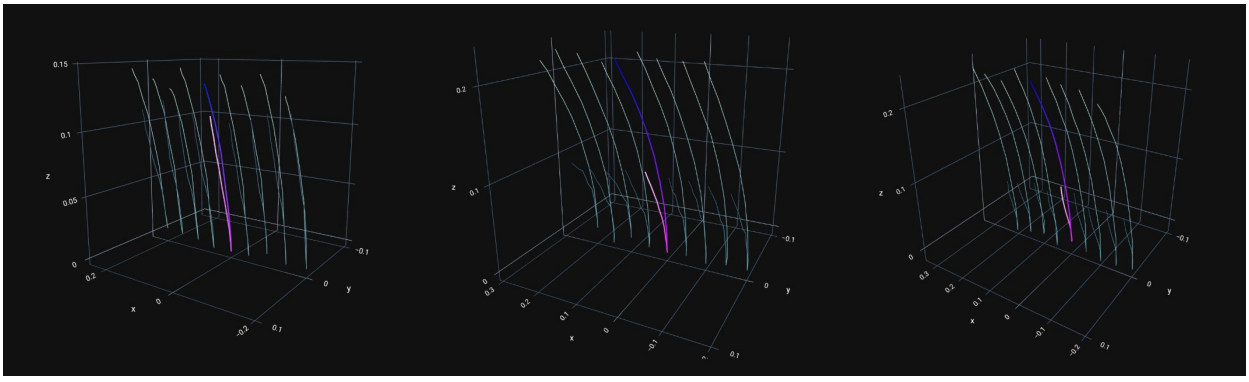


Figure 80. Samples of generated motions (teal) and the initial seed (magenta gradient).

To generate new samples, the encoder model (p_θ) was fed with a randomly selected motion from the dataset x as the seed to find its latent vector z. Then a range of random noise vectors was generated and added to create z'. These new vectors were passed to the decoder model (q_ϕ) to generate a range of new motions x':

$$z = p_\theta(x)$$

$$z' = z + \mathcal{N}(\mu, \Sigma)$$

$$x' = q_\emptyset(z')$$

Figure 80 presents three sets of generated motions. Comparing the seed motion (rendered in magenta/pink gradient) with the generated samples (rendered in teal) shows a close resemblance between the samples, with slight variations resulting from the noise function.

5.3.3 Robotic Implementation

The robotic implementation resulted from a close collaboration between Mahtab and me. Through dry runs and demonstration sessions, the robot/santur setup, audio recording tools, and the actuation methods were all adjusted and fine-tuned based on her feedback and comments.

Hardware Setup

The robotic implementation was designed around an ABB IRB-120 articulated six-degree-of-freedom industrial robotic arm with 580 mm horizontal reach and a maximum load of 3.5kg (ABB Robotics 2022)(Table 6 and Table 7). While there were three robotic arms available at the School of Architecture dFab Lab, the smallest of the squad, was more agile and better suited for this specific task. The hardware setup included the robotic arm, an adjustable adapter to hold the mezrab, and the santur (Figure 81, Figure 82, and Figure 83).

The robot was programmed using ABB's proprietary language, RAPID. Converting the motion sequences—either collected by motion capture or generated using the machine learning model—into robot-executable RAPID code followed this process: 1) converting the targets into Grasshopper plane objects, 2) converting the planes into HAL targets, 3) generating and processing RAPID code, 4) adding supporting motions and UI in RAPID, 5) uploading the code to the robot controller (Figure 84).

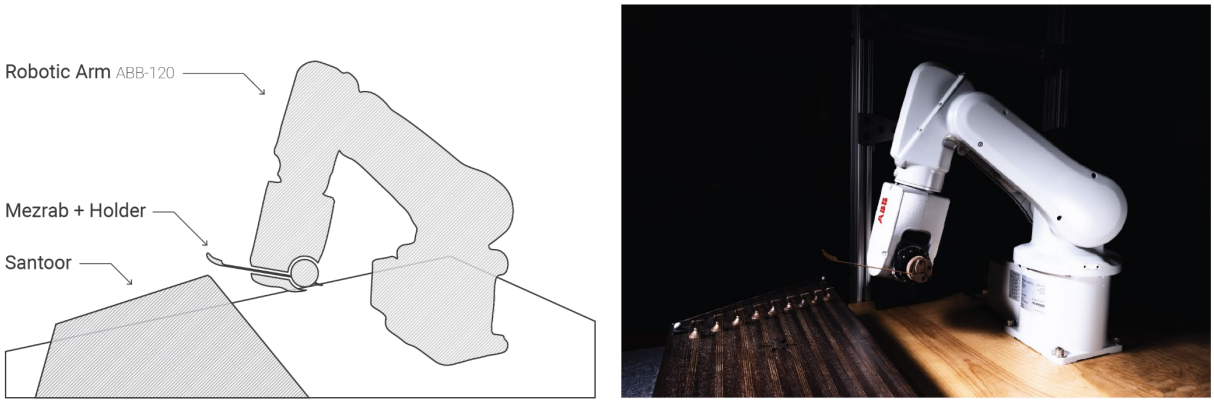


Figure 81. Schematic diagram of the robotic setup.

Figure 82. Mezrab holder, detailed view (left), installed on the robot (right).

Figure 83. Robotic setup details.

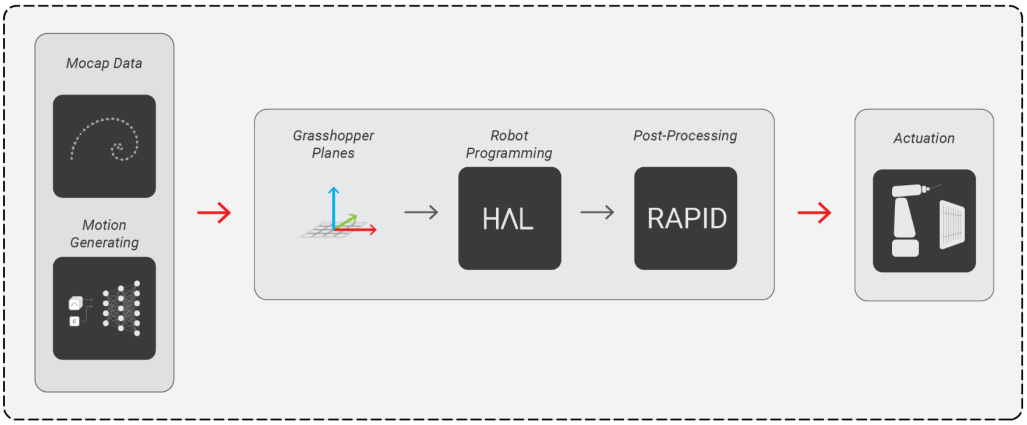


Figure 84. Schematic diagram of generating motion for robotic actuation.

Table 6. ABB IRB 120 General Specifications (ABB robotics 2021).

Specification	*Value*
Max Payload	3kg
Max Reach Span	0.58m
Number of Axis	6
Max TCP Velocity	*6.2 m/s*
Max TCP Acceleration	28 m/s²
0-1 m/s Acceleration	*0.07 s*

Table 7. ABB IRB 120 Movement Specifications (ABB robotics 2021).

Axis Movement	*Rotation Range*	*Velocity*
Axis 1	+165° to -165°	250°/s
Axis 2 Arm	+110° to -110°	250 °/s
Axis 3 Arm	+70° to -110°	250 °/s
Axis 4 Wrist	+160° to -160°	320 °/s
Axis 5 Bend	+120° to -120°	320 °/s
Axis 6 Turn	+400° to -400°	420 °/s

Control and Path Planning

While developing the robotic implementation, it became clear that a mere accurate reconstruction of strokes would not help us create the robotic instrument for Mahtab. We understood that fine-tuning the model to meet her expectation of sound quality is what we should aim at. To achieve this, I revised the pipeline to achieve: 1) accurate motion reconstruction while 2) collaborating with the musician to fine-tune the results during the test sessions.

To address the first goal, it was essential to address two factors: 1) accurate recreation of poses in six degrees of freedom and 2) maintaining the correct velocity throughout each sequence. Accurate positioning is a routine and trivial task in industrial robotic arm programming. Standard programming procedures are sufficient to meet that goal. However, maintaining the accurate velocity while passing through closely located targets was a challenge and mandated a tradeoff between accuracy and speed.

Timing
To meet the exact timing of each motion, a special feature of ABB robotic arms was used to enforce a constant timing between each target point.[24] The following tests revealed that the robot could not accurately keep up with the acceleration required for this study. Specifically, the robotic arm could not rapidly bounce back after the touching frame. This issue was exacerbated when all six robot joints were engaged in the motion.

To improve the robot's agility, with the tradeoff of losing some accuracy, the motions were programmed to fly by targets; the robot did not need to touch the targets, but it was satisfactory to get close enough to each target before proceeding to the next ones. The threshold of fly-by is independently adjustable for each target by *zone value*; if the tool center of the robot gets closer than the zone value to the target, then it has the green light to proceed to the next ones. The fly-by moves result in smoother motions, which was more desirable for this project.[25] For this study, I defined a zone range of 50 mm for all the targets, except for the touching frame set at 0.3 mm.[26]

Reduction
During the motion capture process, the samples were collected at 120 frames per second. Accordingly, each target point corresponds to a period of $1/120$ second (Figure 85, left). Forcing the robotic arm to process 120 targets per second can potentially overload the robot controller and result in software failure.

To compensate for this issue, two different methods were tested: 1) downsampling the number of targets during the execution time; reducing the number of targets by keeping every n^{th} target helps keep up with the $n/120$ of second time stamps (Figure 85, center), and 2) reducing the number of targets by truncating steps before and after the touching moment; during the replay tests we realized that reducing the motions for each note to five steps before and two-three steps after the touch, will not significantly affect the outputs (Figure 85, right). This reduction also helps reduce noise as it demands a shorter travel distance per note.

Figure 85. Sample reduction methods.

Motion type
Another critical factor in robot programming is how the robot travels between the targets. While each target represents the pose of the mezrab's tip at each time stamp, it is also important to adjust the trajectory the robot follows between these targets. A smooth and seamless motion requires the robot to

[24] As the target points were all captured on a constant frame rate (120Hz) the robot had to traverse the distance between the two targets in a fixed time.
[25] In contrast, the robot can be directed to strictly touch a target and then steer toward the next. This approach to path planning inevitably renders the robot more sensitive to noisy paths and results in jittery motions and vibration.
[26] RAPID language has two different z values with a 0 mm radius 1) fine, which forces the robot to touch the target and hold on to that position for a fraction of a second, and 2) z0, which applies a 0.3 mm zone value with no enforced short stop. For this study, the z0 is used to keep up with speed and prevent any adverse effect on the produced sound.

rotate all six joints simultaneously. Between multiple motion types available for ABB robotic arms, the MoveJ method can accommodate this study's requirements. MoveJ method gives the robot the flexibility to calculate and follow any deliberate combination of joint rotations to traverse between the targets. Simultaneously, it can synchronize the six joints to arrive at the designated target on time. This movement results in a smooth motion and helps avoid singularity and joint rotation limit issues that may occur in a linear motion. Moreover, the strict timing method helps keep up with the time stamp and follow the musician's pace. Table 8 presents a summary of motion planning details.

Table 8. Robotic motion tuning

Factor	*Primary method*	*RAPID Method*
Accuracy	Assigning different zone values for trail and touch	*z* values
Consistent Velocity	Down-sampling / truncating motions Enforcing time limit for each move	*t* flag
Motion smoothening	Using joint-based motion	*MoveJ*

Motion Sample Variations

During the early discussions, Mahtab pointed out that each stroke, regardless of hand and note, should be the same. However, a closer analysis revealed slight differences between each stroke sample that she provided. Upon closer inspection of the collected data, I noticed that the left-hand strokes and the right-hand ones are quite different and easily distinguishable. However, the differences between strokes with different notes are barely noticeable, and classification methods had a tough time classifying them.

While working on this study, I had the opportunity to discuss my work with Mohammad Jafari, the researcher behind the Santur Bot project at Georgia Tech Center for Music Technology (2021). When I discussed this observation with him, Mohammad provided an interesting explanation. He mentioned that the reason behind this observation is twofold. He explained that the almost-indistinguishable strokes for different notes is possibly associated with the modern style of playing santur. This style encourages musicians to apply the same force on both hands for all notes and keep the strokes uniform. Meanwhile, he elaborated, right-handed artists play the notes on their dominant hand louder and more pronounced, making the difference between the two hands noticeable.

His explanations confirmed what I had observed during the early data visualizations; while left-hand motions were shorter and more diverged, the right-hand strokes were more converged, longer, and faster (Figure 71). After discussing these points with Mahtab, we concluded that the differences between notes are negligible, and we can focus on the technique and hands as the labels for the strokes.

5.4 Preliminary tests

We calibrated the robotics system and evaluated its basic characteristics at the CMU dFab lab (Figure 86). During these test and demo sessions, we had the opportunity to debug the system, adjust the robot/santur setup, test various sound recording arrangements, and test different noise reduction methods.

Before using the physical machine, I used RobotStudio, ABB's modeling and simulation software, to simulate the process. The simulations were critical to catching common robot programming bugs such as collisions, singularities, and reachability issues. During these simulations, I developed and tested a user interface to control the operation using the robot's teaching pendant.

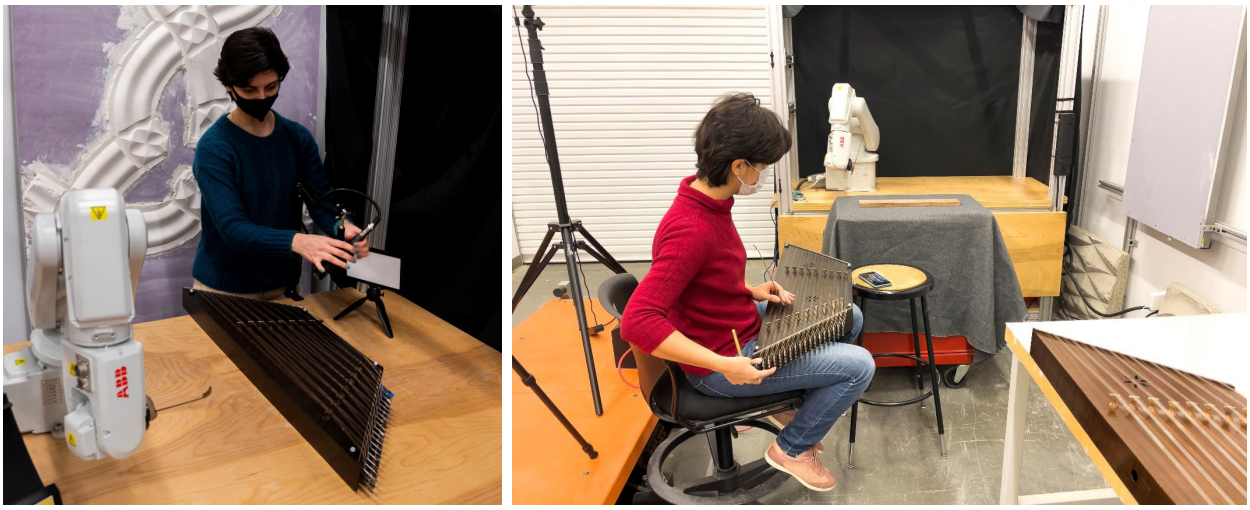


Figure 86. Testing sound recording setup (left), tuning santur for the demo (right), images by the author.

After this phase, a series of tests were conducted on the real robot to examine all system components, specifically the sound recording setup.[27] The musician iteratively evaluated the quality of the results based on her experience and personal preferences. Meanwhile, I focused on designing and implementing methods to accommodate her inputs. Mahtab's inputs were essential to achieving her desired sound characteristics through adjustments, modifications, and fine-tuning of the robot-santur setup.

5.4.1 Motion Studies

We tested the robotic system with three sets of motion samples:

1. Single-joint strokes actuated using only the sixth joint of the robotic arm, serving as a benchmark for sound signature and actuation,
2. Strokes samples randomly selected from the motion capture data[28] to examine the robotic setup's ability to reproduce notes.
3. Strokes generated by the C-VAE model to examine the machine learning model's ability to generate valid motion data.

The three sets of tests allowed us to improve the physical setup, debug the robotic programming pipeline, and, last but not least, evaluate the sound signature and noise level in each type of motion.

The single-joint motions were the most trivial ones to program, debug, and test. These strokes were simple, repetitive, and strictly consistent. The robot was programmed to rotate the sixth joint 20 degrees clockwise and then counterclockwise. Since only the smallest motor of the robot was engaged during these motions, it emitted the lowest noise level. However, the outcomes were repetitive, monotone, and of low amplitude.

For the replay tests, a set of random samples was picked from the dataset. For tests on the generated samples, a set of strokes was generated by sampling from the C-VAE model, conditioned on the right hand. The following steps were similar between these two tests. All strokes were passed to

---

[27] The robot was set to operate at its peak speed in automated mode to keep its pace with the stoke samples. Special safety precautions were in place to guarantee the safety of the users. Following the safety guidelines, all individuals in the room were kept out of the robot's working envelope.

[28] Accordingly, the tests at this stage are also called replay tests.

Grasshopper/HAL[29] definition to generate the RAPID code and then passed to RobotStudio for simulation. Some strokes that resulted in singularity or reachability issues were flagged and removed. On each set of successful samples, one was selected and inserted into the RAPID code on the robot controller for the tests and play. These motions were used to play notes where we could observe the robot's behavior and sound characteristics. During these tests, we applied various adjustments to the robotic setup, path planning process, sound recording system, and santur's tune.

Replaying mezrab motions at the correct pace pushes the robotic arm to its acceleration limits.[30] Such vigorous accelerations echoed as a barrage of noises over various frequencies and amplitudes. Motors, joints, vibrations, and the cooling fans in the controller cabinet all contributed to the noise. Therefore, the produced sound was a mixture of santur sound and various mechanical noises.

Eliminating undesired noise from the recordings was a complicated challenge. We implemented several methods to compensate for this issue, 1) using mezrabs with no felt tip to increase the amplitude of generated sound, 2) testing various microphones to isolate santur sound and reduce ambient noises, 3) isolating the microphone and santur from the robot framework, 4) applying noise reduction in post-processing, and 5) fine-tuning an equalizer to cut certain frequencies. Despite all these measures, some level of robotic noise is still present in the final recordings.

5.5 Demonstration

While working on this study, we developed a better intuition of the toolmaking process and the robotic musical instrument we were making. It gradually became clear to both of us what could be done and what fell beyond the capacity of this tool. This mutual understanding became the keystone for designing the demonstration, where Mahtab could play a score of her choice alongside the robotic musical instrument.

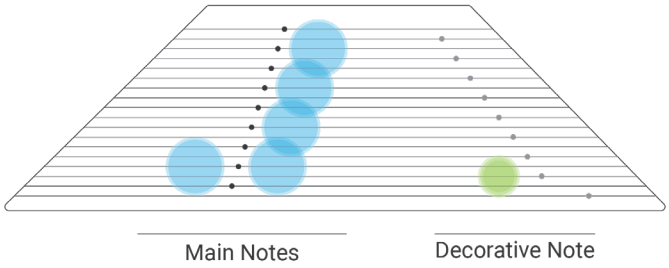


Figure 87. Distribution of notes played by the musician (blue) and the decorative notes played by the robotic arm (green).

We sketched ideas about treating the robotic instrument as the musician's augmented third hand. She proposed modifying an existing score and elaborating on it by adding decorative notes, where playing them with two hands was virtually impossible. The original score, a Kurdish piece named *Pepoo Soleimani*, included a repetitive pattern of F notes played with riz technique on the left side of the instrument on white strings. She proposed to complement these notes with decorative F notes played on

[29] HAL is a framework for programming industrial robotic arm, with powerful add-on for Rhino/Grasshopper (Schwartz 2013; HAL Robotics 2022).

[30] The RAPID code was adjusted to manually override the default acceleration limits on the robot's joints to keep up with mezrabs motions.

the opposite side of santur on yellow strings. The original song was perfectly fine without these decorations, but adding these matching notes could elevate it to another level. In this scenario, the robotic musical instrument was responsible for playing the decorative notes and keeping the tempo.

As riz technique occupies both hands, playing another note, especially on the other side of the instrument, is a demanding, if not impossible, task (Figure 87). This is where the robotic arm serves as the third hand and plays the decorative note. In this scenario, the role of the tool is to complement the musician's performance, in contrast with the automation approach, where the goal is to replace the musician. She described this scenario as the human playing the "delicate details" while the robotic instrument was tasked to place "pleasing" notes between them.

We framed the interactions between the musician and the robotic tool by a fixed shared musical context that included 1) the modified score and 2) a set tempo. We intentionally left more advanced models of interaction, i.e., improvisation, variable timing, and imperfect tempo, to focus on the main topic of this study. In this scenario, the musician and the instrument followed the shared musical context while contributing to the higher-level musical outcome: the robotic instrument keeps the beat and plays the decorative notes while the musician plays the main notes (Figure 88).

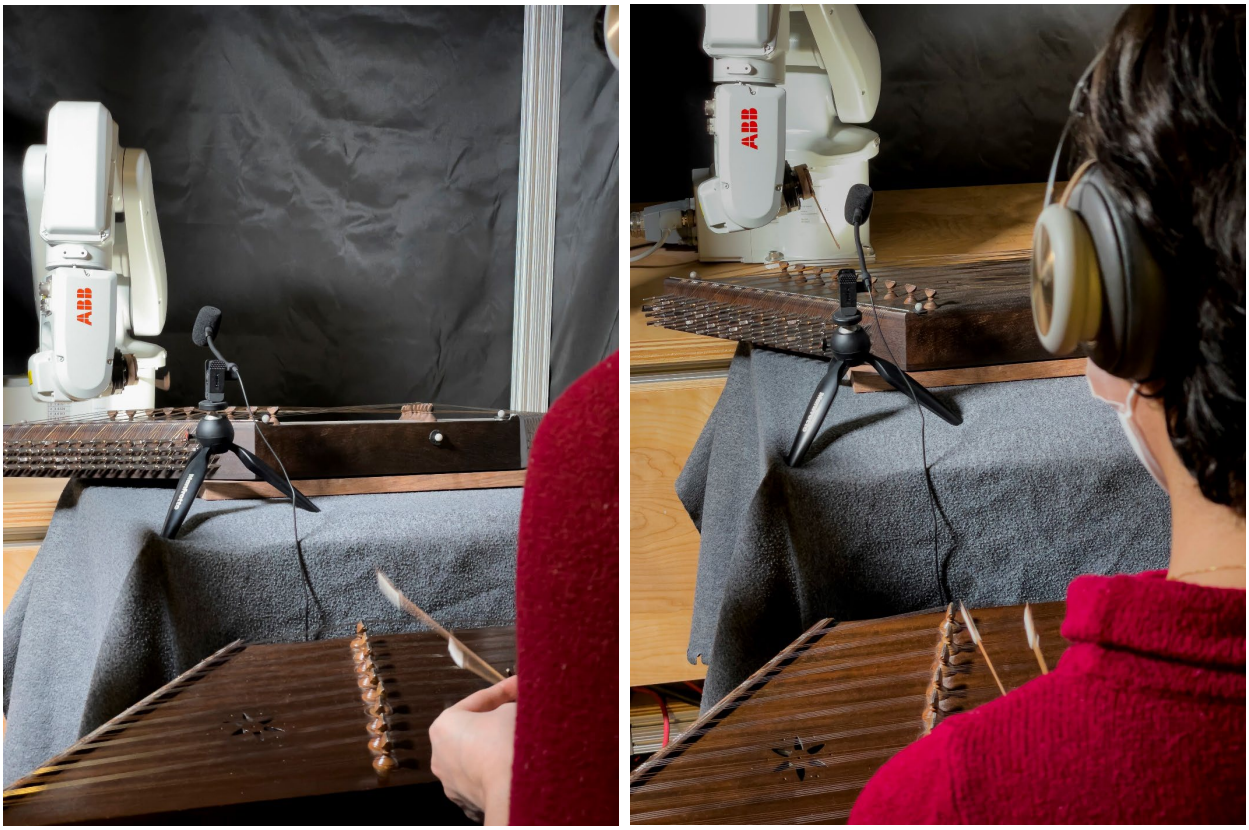


Figure 88. Artist and its robotic musical instrument during the demonstration, images by the author.

Mahtab decided on the notes being played, her role in the demonstration, and tasks that she gave the robotic musical instrument (i.e., keeping the beat, playing the base note, and playing solo notes (Bretan et al. 2016)). Her decisions about tempo and techniques, which were highly dependent on the technical aspects of the robotic arm, were informed by our mutual understanding of the system's affordances and limitations. For instance, during the initial stages of data curation, we noticed the hardware limitation of

our motion capture and robotic setup to handle riz technique. Thus, we collectively decided not to use this technique on the robotic instrument.

5.5.1 Discussions on the demonstration

The demo session was set in the dFab Lab at the School of Architecture, Carnegie Mellon University. The room was packed with different pieces of hardware, tools, models, and samples from other ongoing research projects, and three robotic arms. The two larger robots were stationary on the other side of the room, giving us just enough space to set our impromptu studio space. The ABB-120 robot was installed on a portable cart, making it easy to move around the room. The aluminum frame around it allowed us to install a black drape around the robot, reducing visual clutters and subtle noises around the setup.

Initial Setup

Before starting, Mahtab tuned both santurs. We fixed the test santur and the microphone on a tool cabinet in front of the robot cart and organized the sound recording hardware on another table. The other santur was placed on a stool in front of the robot (Figure 88).

I ran the robot for a couple of test runs. "The sound has changed since the last time. The noises [are different],"[31] said Mahtab immediately after listening to the robot replaying her mezrab motions on the monitor headphones. She had adjusted the recording setup during the last test session, and now she could notice the slight change in the sound signature and the robot noises. She was right; I had made a few changes to the robot motion planning and managed to reduce vibrations. Also, I just swapped the original microphone with a new one based on her suggestion.[32]

The early tests—that we conducted prior to this demo—were quite helpful in revealing challenges in sound recording and acoustic characteristics of the produced sound. As Mahtab mentioned, santur, by design, is a challenging instrument for sound engineers to record. In this study, it became even more challenging due to the noises generated by the robot.

Motion Types and Reduction Methods

Once she was on board with the adjustments, she started playing the modified song alongside the robot, and we worked together to test various motion types and reduction methods. She carefully monitored the instrument's sound signature on different motion types and reduction methods, trying to produce a sound signature that could meet her bar.

The behavior of the robotic arm and the sound signature when playing generated strokes were adequately close to the replaying tests, with the same noise level. Upon further review, Mahtab could not distinguish between the replay sets and the generated motions. Assessing only the motion generation, it was a successful test.

Although it was reassuring to find such a close resemblance between the sound signature of the two tests, it was not clear if she would feel the same under a better sound recording condition. Using down-sampling and truncation methods helped reduce the noise level significantly. Mahtab preferred the truncation method as it emitted noise in shorter bursts than the downsampling method.

After these tests, I played some notes using only the robot's 6th joint. After several rounds of full 6-joint motions, listening to the robot playing only with the sixth joint was a surprise for her. "The sound is much

---

[31] Sound recordings of the demo session, 0:50 time stamp, Dec. 21st, 2021.

[32] The first microphone was a Shure SM57-LC cardioid instrument microphone, like the one she was using at her home studio. The new one was Shure PGA98H-XLR cardioid condenser gooseneck instrument microphone.

better!" she reacted when hearing the robot playing the three count off notes.[33, 34] This holds with our expectations; the sixth joint of the robot is considerably quieter than the other joints, and its motion makes the least noise, resulting in a crisper sound. Nevertheless, this was a repetitive mechanical motion with no variation rendering it the least desirable outcome.

Sound Signature and Noise

I was curious to know Mahtab's opinion on the test. Later, during the debriefing interview, I asked her if the robotic musical instrument that we made could pass her bar. She was affirmative. The sound signature was on par with her expectations. I was confused as the recordings were flooded with various noises, but she was fine with that. She clarified, "[…] at that moment, my effort was to only listen to the strokes being played. This happens in the human brain naturally. But when you record the sound, it is different from what you hear." At that point, by applying "… noise reduction, it can produce the sound that is expected," she asserted.[35]

Mahtab's remark was quite interesting. As a person with limited experience in music, my criteria and success measures were quite different from hers. While I was completely distracted by the technical challenges and the surrounding noises, she could isolate the notes from the background noise in her mind, forming a unique perception of the instrument and its performance in her mind. However, beyond that point, we needed to utilize post-processing and noise reduction methods to flesh out those notes from the flood of noises.

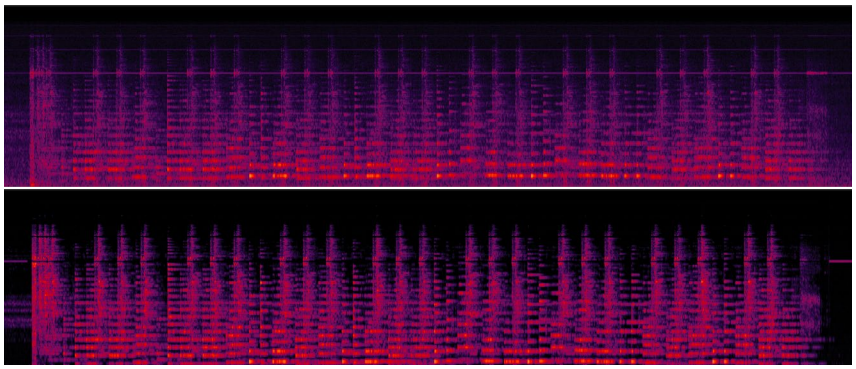


Figure 89. Recording Spectral Frequency before (top) after (bottom) applying noise reduction and removing unwanted frequencies.

Machinic [In]accuracy

We rehearsed the song several times with different motion types and reduction methods and documented the process. She played her notes on the white strings and the robotic instrument was tasked to play the decorative notes precisely after her, but on the yellow strings on the opposite side of the instrument. After several attempts with different reduction methods, Mahtab put off her pair of headphones and nodded her head in approval. I put on the headphone and listened to the recording for a few seconds. "Is it synced?" I asked. She was affirmative, "It is correct. It plays correctly. It is supposed to play at the end of my note."[36]

---

[33] Count off, or count in, is a verbal or instrumental cue which helps musician, in a performance or recording session, start at the right time with the right tempo.
[34] Demonstration video recordings, December 21st, 2022.
[35] Mahtab Nadalian, debriefing interview by author, February 5th, 2022.
[36] Demonstration video recordings, December 21st, 2022.

It was the first time the robot played the notes at the exact time she had composed. However, for several rounds, the timing was not correct. Upon closer inspection of the recorded footage, I noticed that the robot's timing was incorrect. I tracked down the issue to the closely-timed notes, i.e., quarter notes. When actuating a stroke without downsampling, the robot could not finish the stroke in the designated time slot, dragging them into the next bar. Since the robot was programmed to use a global timer, it could fade these subtle drags during the rest periods and prevent them from accumulation. Accordingly, the overall timing was correct, but individual close notes were slightly dragged.

I programmed the robot to count off by playing three beats at the beginning of the song. As the robot could not catch up with the pace of these three count off beats, each beat was slightly skewed. However, it was enough to confuse Mahtab, making it challenging for her to keep up with these subtle fluctuations. When we watched the videos of two different runs side-by-side, it became obvious that the robot's skewed tempo had affected her timing and performance throughout the song.

It took her by surprise when I mentioned this issue in our debriefing session. Her perception of the robot as a mechanical system gave her the impression that it could keep up with the tempo perfectly and any rush or drag was from her side. She elaborated that "… I still think that when you order a robot to do something, it can repeat that task several times [with no variation]. But a human cannot do that. This is the difference between electronic or computer [generated] music and an orchestra."[37] She trusted the mechanical system to the extent that she attributed all errors to herself.

5.6 Discussion

Research by design is a dynamic process nurtured by non-textual artifacts and the constant realignment of objectives and approaches. In such a non-linear process, the outcomes contribute to form documentation of the process rather than a conclusive report on a specific solution (Roggema 2016, 11). In this study, the framework, meta-tool, robotic musical instrument, and demonstration were all vehicles for a research effort, which helped us to inquire about ML-based machine learning toolmaking for creative practitioners. Thus, I organized this section as a discussion on the process, the musician's experience, the inclusion of the context, and the dynamics between the musician and the meta-tool.

5.6.1 Musician's Role in the Toolmaking Process

One of the primary objectives of this study was to examine a toolmaking approach that embraces the dynamics between the social context—i.e., relationships and interactions between people —, the physical context—i.e., instruments—, personal context, and the underlying technology.

My approach to embracing the musician's personal and subjective measures was to actively engage her in various stages of the project, from purchasing the instruments to fine-tuning the robotic setup and adjusting the sound engineering details. Our dialogues shaped different elements of this study and guided us through design, implementation, and demonstration.

This dynamic and collaborative toolmaking process was a new experience for the musician and understanding her point of view was critically important. During the debriefing interview, I asked Mahtab about her journey. She mentioned that at the beginning, she was curious to learn what can the robotic instrument do as a stand-alone agent and what type of problems it would face. As we proceeded in the study, she directed her questions beyond mere technical aspects: "what can this instrument convey?" "Is it capable of conveying the same feeling and aspirations as she does just by replicating her hand motions?"

[37] Mahtab Nadalian, debriefing interview by author, February 5th, 2022.

"Having only one robotic arm available, how is it possible to achieve the complexity that comes with two hands?" and "if the robotic instrument successfully produces notes, will audiences recognize this?"[38]

Our discussion continued into her sense of engagement in the process. She clearly drew a line between two periods: 1) the data collection and 2) the demonstration. She felt more engaged when we started to work in the lab with a physical robot in the room. She became cognizant of the "interactive" work that she was doing with my help to make a robotic tool to play music. "… [D]uring the data collection phase, I felt I was just playing, and someone was just recording. Your explanations were clear, but what would happen at the end was not foreseeable or tangible. But when the robot started to play, I realized and felt what was happening."[39]

The first part of her statement was enlightening. I tried to limit the level of intervention in the data collection phase and keep the musician in an environment and workflow as close as possible to her working environment. In practice, the process was so close to her routine workflow that she could not feel any difference. To my surprise, this approach came at a price; she felt less engaged in the preliminary stages. However, she regained her sense of engagement upon the introduction of the robotic instrument and moving to the robotic lab. She mentioned that for her, the lab environment was an "experimental space," and she was there for a new experience, not to stay, create, and work for a long time. Having that mindset, she found herself comfortable in the lab environment for the brief period of this study.[40]

5.6.2 Real-time Interaction with ML Model and Data Curation

A pivotal moment in this study was when we realized that it is impossible to form a meaningful real-time interaction between the musician and the machine learning model. This issue derived from the data modalities we worked with, motion and sound. As discussed earlier, motion was merely a vehicle to convey the musician's idiom into the robotic musical instrument. The mismatch between the two modalities made it impossible to directly associate the motion samples and the sound created by the robot. In this situation, we could not rely on the real-time interactions I previously utilized in the SecondHand study.[41]

Thus, we went back to the drawing board and revised the way the musician could interface with the meta-tool. In the revised plan, the musician focused on 1) data generation, i.e., making decisions on samples, techniques, and notes to be collected, 2) supervising the robotic apparatus fine-tuning, and 3) contributing to the sound recording and post-processing. As the toolmaker, I took over the training phase and trained the model to maximize the resemblance between the collected samples and the generated ones. This pivot allowed us to explore a different form of a collaborative toolmaking process.

This challenge resembled the issues I faced in the first iteration of the SecondHand study, where participants could not comprehend the direct impact of their decisions on the outcomes. In the second iteration, I integrated the data curation and generation phase in one unified interface, where they could observe the results in real time. Following the same line of thought, in future efforts, it might be possible

---

[38] ibid.
[39] ibid.
[40] However, she also expressed some of her concerns about the lab environment. She pointed out that the lab space was too large for sound recording, and its environment was hard to control with all the mechanical equipment running around the room.
[41] In the SecondHand study, the input and output data were both images. Participants could observe and comprehend the results of their data curation decisions almost in real-time. They had the opportunity to update their decisions and steer the learning process in their desired direction.

to integrate the training process with the robotic instrument and observe the results of data curation in real time with the robot replays.

5.6.3 Contributions, Limitations, and Future Works

The key contributions of this study are: 1) introducing a framework for ML-based musical robotic toolmaking integrating the user inputs, preferences and subjective measures, as well as aspects of physical context, i.e., musical instrument, 2) developing an implementation of the framework as a meta-tool and its main components: a) data pipeline, b) machine learning model, c) hardware implementation, 3) demonstrating the affordances of a collaborative toolmaking process to render machine learning-based toolmaking accessible to creative practitioners.

The meta-tool developed for this study is designed around the specific use case and tailored for the musician and her instrument. However, the framework provides a blueprint for further explorations and can serve as the underlying principle for making bespoke meta-tools for other musicians and their instruments. A keen researcher can expand this approach to other musical instruments or other creative practices with adequate modifications and adaptations. This requires further efforts to find the proper machine learning model, pick the right data modalities, develop a matching data collection method, and make a suitable robotic setup.

An underlying assumption of this study was that six-degree-of-freedom motion is sufficient to learn the artist's idiom of playing, as we developed the robotic instrument, it became clear that with the current hardware available, it is not possible to test this assumption. A potential direction for further investigation is to have the machine learning model adjusted to incorporate other forms of sensory data, such as torque—for better representation of motions. Testing other machine learning algorithms to study other characteristics of playing santur—such as timing—is another technical challenge that can be tackled in the future.

Integrating the robotic instrument in the data collection phase is also an interesting field of research. For instance, making a real-time bridge between the data collection pipeline and the robotic instrument is one in this direction. This will help the musician instantly evaluate samples in real time by replaying them on the robot. The robotic instrument can also help augment the dataset by playing slightly changed motions and allowing the musician to evaluate them. Another possible avenue of research is integrating various real-time fine-tuning methods for robotic actuation and motion generation. Such real-time interactions can allow the musician to tune the tool—like how they tune their instrument.

This study signified some of the technical challenges that physical actuation will bring to the equation. Throughout this study, the robot's mechanical limitations were quite challenging. Keeping up with the velocity and acceleration of mezrab pushed the robot to its limits while producing undesired noises. Developing a bespoke robotic system to match mezrab motions and potentially provide control over the torque is another field to explore further.

Chapter 6. Conclusion

The final chapter of this thesis reflects on the objectives of this research, the questions, and my journey to answer them. It includes a summary of lessons I learned from each case study and the concluded results. Throughout this chapter, I will put these outcomes in the broader picture of computational toolmaking for creative practices. Finally, I will propound the next steps and the directions that future research can take.

6.1 Research Summary

This thesis started with the question of how creative practitioners can harness the power of machine learning algorithms in their tools. In chapter one, I explained how the latest bloom of machine learning in the mid-2010s has raised a new wave of interest among creative practitioners to explore the intersection of art and artificial intelligence. We observed how a growing number of creative practitioners, such as artists, designers, and architects, are actively exploring ML affordances in creating tools that support their creative practices.

The literature review revealed that the current efforts to create tools for creative practitioners using machine learning could not meaningfully integrate the idiosyncratic aspects, elements of the physical context, and nuances of creative practice into the toolmaking process, leaving creative practitioners with limited options. Through this discussion, it became clear that although the recent wave of ML-based tools promises a bright future for creative practitioners, the path toward this future is inevitably riddled with challenges. I explained how creative practitioners' lack of ML technical knowledge in computer programming and ML had confined them to off-the-shelf options. I proceeded to open the discussion on how this issue contributed to the detachment of the toolmaking process from its physical and personal context.

This led to formalizing the two primary questions of this research:

- How do interfaces for data generation and curation for generative machine learning offer new pathways for toolmaking for creative practitioners?
- How can a collaborative approach mitigate the lack of technical machine learning experience among creative practitioners and help them to integrate the idiosyncratic aspects, elements of elements of the physical context, and nuances of their creative practice in the toolmaking process?

To address these questions, I hypothesize that interfaces for data generation that emphasize user-generated data to integrate elements of the physical context and users' subjective preferences can reveal new potentials of generative machine learning to support creative practices.

Following this, I also hypothesize that a collaborative approach to developing ML-based tools for creative practices can meaningfully bring ML experts' technical literacy to complement creative practitioners' domain knowledge and skills, overcome the technical ML challenges, and help integrate some idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process. The two case studies explored these two hypotheses.

In Chapter 2, ML-Based Toolmaking for Creative Practitioners, I reviewed the recent efforts in ML and toolmaking for creative practitioners. I identified that a combination of lack of ML technical knowledge, limitation on data, and challenges in evaluation renders ML-based tools inaccessible for many creative practitioners. I proceeded to review various methods to bridge these technical barriers and audited some of the tools and software packages that adopted these methods. It became clear that applying those methods to mitigate the technical barriers comes with the trade-off of losing control over the toolmaking process, increased chance of making technical mistakes, and getting confined to the already available options, and crucially missing the personal and physical context of the practice.

To overcome the technical barrier, I suggested a collaborative approach to toolmaking between toolmakers and creative practitioners and using data as a primary means to interface the ML algorithm. To address the missing context, I worked toward incorporating elements of the context in the toolmaking

process for creative practices: being physical, i.e., tools and materials, personal, i.e., user's subjective measures and personal preferences, or social, i.e., interactions with peers and other experts.

In chapter 3, I proposed a framework for making collaborative ML-based tools for creative practices by combining the two approaches. I designed this framework to embrace the pivotal role of the creative practitioners and the context in the toolmaking process, evoking new means to avoid decontextualization and abstraction. It centers creative practitioners in the toolmaking process and embraces data collection in the close-to-real context of the practice.

This framework supports a dynamic collaboration between creative practitioners and ML expert toolmakers to overcome the technical barriers and render ML algorithms more accessible. It encourages them to collaborate on creating bespoke data curation workflows to integrate data in the actual context of practice. Decisions over the inclusion or exclusion of factors are governed by creative practitioners, based on the context and the task at hand, and by the toolmaker based on technical limitations and affordances.

Moreover, this framework sought to allow the creative practitioners to use data to interface with the ML backend by generating, collecting, and curating training datasets to incrementally shape the learner's behavior.

I investigated the validity of the hypotheses through two case studies, *SecondHand* and *ThirdHand*. For each, I developed a meta-tool based on the proposed framework and collaborated with creative practitioners in the toolmaking process. Chapter 3 and Chapter 4 were comprehensive reports on these case studies.

In *SecondHand*, I investigated one of the hypotheses of this research: *interfaces for data generation that emphasize user-generated data to integrate elements of the physical context and users' subjective preferences can reveal new potentials of generative machine learning to support creative practices*. I collaborated with a group of participants to create their machine learning-based tools to generate handwriting typefaces. I investigated how creative practitioners worked with the meta-tool and documented the bi-directional dynamics between the human agent, physical context, and the underlying technology. This study also attempted to evaluate the potential of data as a form of interface to make ML-based toolmaking more accessible to creative practitioners.

I observed that the participants were profoundly engaged in the process, from creating their datasets to using the machine learning model for generating the typeface. Several participants expressed how the choice of medium or software package to process the data have changed their results. It becomes clear from this repeated pattern that the data collection process and the whole toolmaking process were heavily influenced by the medium and the software package.

Participants also reported that their generated typefaces were visually close to their handwriting. However, it was also interesting to know that many of them gradually changed their data samples to match the affordances of the meta-tool. This bi-directional interaction was one of the most interesting observations of this study; participants molded the tool to fit their workflow and concurrently adjusted their workflow to fit the meta-tool's affordances.

Throughout the discussions and reflection papers, participants also mentioned that they found data to be a more intuitive interface than coding. Data, and its representation as interactive visualizations in the dashboard, allowed them to sense the relationship between the input data and the results. This made it easier to control the model and to improve the quality of generated samples by taking course-grain steps toward the desired results.

Also, they mentioned that the data visualization widgets in the meta-tool helped them to introduce their personal preferences and subjective measures in the data curation, training, and drawing samples. The three-round process was a window to observe each participant's learning process. In every round, participants found new lessons and faced new challenges, and they managed to improvise solutions to overcome them in the next round.

This study made it clear that there is a fine balance between the complexity of the machine learning algorithm, its training time, and the scale/variance in the training dataset. Participants reported that the mix-and-match strategy to make hybrid handwriting typefaces from very large or high-variance and diverse datasets was beyond the capacity of the provided model. Increasing the complexity of the model to mitigate this issue could increase the training time and reduce the usability of the meta-tool. As such, participants came up with creative solutions, i.e., adjusting their data to match the ML model and averting complex data combinations.

In the second case study, *ThirdHand*, I collaborated with Mahtab Nadalian, a seasoned musician and professional santur player. We collaboratively developed a bespoke meta-tool, including the robotic hardware and corresponding software tools, to make a robotic musical instrument for her. This case study validated the first hypothesis of this study: *interfaces for data generation that emphasize user-generated data to integrate elements of the physical context and users' subjective preferences can reveal new potentials of generative machine learning to support creative practices*. It also backed the second hypothesis: *a collaborative approach to developing ML-based tools for creative practices can meaningfully bring ML experts' technical literacy to complement creative practitioners' domain knowledge and skills, overcome the technical ML challenges, and help integrate various idiosyncratic aspects, elements of the physical context, and nuances of creative practice in the toolmaking process.*

I designed this case study based on the principles of research by design, cast as design-led research. It allowed me to examine the dynamic and bi-dimensional interactions between me as the toolmaker and the musician. I narrowed down the scope of this case study to developing a robotic musical instrument to play santur based on the samples provided by the participating musician and actively avoided topics such as composing, generating, or improvising music.

I also took ThirdHand as an opportunity to further integrate elements of the physical context into the toolmaking process. I introduced a physical-to-digital-and-back-to-physical process into the meta-tool. Working with Mahtab, we tried eradicating any interference in the data curation phase, using a minimally modified pair of mezrabs and a real santur. Our dialogues shaped different elements of this study and guided us through design, implementation, and demonstration. The meta-tool allowed us to collect and curate her mezrab motion samples in an environment that closely resembled her routine practice sessions.

As we were working on this study, it became evident that we could not create a meaningful real-time interaction between the musician and the machine learning model. It was a challenging endeavor, as the inputs and outputs of the meta-tool were of two different modalities, motion, and sound.[1] The mismatch between the two modalities made it impossible to directly associate the motion samples and the sound created by the robot. This challenge prevented us from having a meaningful real-time interaction between Mahtab's demonstrations and the meta-tool outputs. We embraced her personal and subjective measures,

[1] To illustrate some of the challenges, I can refer to data visualization for the ThirdHand study. It was visually overwhelming to represent both time and 6DoF motion sequences on a 2D screen. Even more problematic was to represent sound, time, and 6DoF data as 2D visualizations.

and she was actively engaged in various stages of the project, from purchasing the instruments to adjusting the robotic setup and the sound engineering details. Then I trained a machine learning model based on the dataset to produce novel examples of mezrab strokes and play them on a real santur by a robotic arm.

We curated a demonstration to showcase the robotic instrument. The blueprint was to let Mahtab play a score of her choice alongside the robotic musical instrument. She personally decided on what she wanted to do, what tasks were assigned to the instrument, and the notes being played. During the performance, the robot was tasked to play the assigned notes and keep the tempo while she was playing her notes.

It became clear that translating data from physical to digital and then back to physical introduces significant challenges to the process. At the initial stages of this study, I assumed that six-degree-of-freedom motions are sufficient to represent the artist's idiom of playing. Due to the robotic arm's mechanical limitations, we could not accurately reproduce her idiom of playing. As such, I could not back this assumption. However, we focused on other aspects of the toolmaking process. Fine-tuning the robotic and santur setup, selecting proper sound equipment, and sound post-processing were all executed based on Mahtab's inputs, and the demonstration made it clear that Mahtab was satisfied with the final sound signature.

During the debriefing, I realized that the data collection process was so close to her regular practice sessions that Mahtab almost lost his sense of engagement with the process. She felt more engaged when we started to work in the lab with a physical robot in the room and became cognizant of the "interactive" work she was doing with my help in making a robotic tool to play music.

Another interesting observation of this study was the musician's perception of the robotic instrument. She perceived the robotic arm as an accurate mechanical system with an extreme level of accuracy. This gave her the impression that any imperfection, such as rush or drag in the performance, was from her side. However, it was a bug in the programming and, indeed, an issue with the robotic system.

The meta-tool developed for this study was designed around the specific use case and tailored for the musician and her instrument. It is not directly applicable to any other instrument or artist. However, the collaborative research and design process described above provides a blueprint for further explorations and can serve as the underlying principle for making bespoke meta-tools for other musicians and their instruments.

6.2 Discussion

6.2.1 Abstraction

This study made it clear that abstraction is still an inevitable step in making a practically feasible representation. A researcher can find methods to mitigate its effects but eliminating all forms of abstraction is a Herculean task. The adverse effects of abstraction were quite tangible in the SecondHand study, where I had to work remotely with two cohorts of participants. In that study, I made the initial decisions on various forms of abstractions, then revisited my decisions based on the feedback from the participants. This iterative workflow gradually shaped the polished meta-tool that was used at the end of the second iteration.

This experience guided me throughout the ThirdHand study. This time, instead of making the initial decision by myself, I worked collaboratively with the musician from the beginning to shape a representation that could reflect her perception of the tool. At this point, the notion of referring to the real world and the immediate context evolved into referring to the expert and the physical context. We

managed to limit the abstractions and intervention to the point that the data collection sessions felt like an imitation of her routine practice sessions.

Nevertheless, some of the decisions on abstraction were not this successful. Abstracting the idiom of playing into sequences of six-degree-of-freedom motions was an example of such cases. It was only toward the end of this study, when we tried to actuate the motions on the robotic arm, that we realized that it might not be sufficient.

6.2.2 Personal Context

In 2018, when I was working with my fellow Ph.D. student, Pedro Veloso, on the project DeepCloud (2018), we examined the power of machine learning algorithms in data-rich domains. It was fascinating to see how users can observe the real-time effects of their inputs on the outcomes. Users could truly see a cloud of points morphing from an SUV to a sedan, or anything in between, in real time. They could pick one, or thousands, based on their personal preferences and subjective measures.

Nevertheless, DeepCloud was rigid and inflexible. Neither users nor we had much control over what could be done with it. Pedro and I downloaded an off-the-shelf dataset, borrowed an ML model, retrained it, and wrapped it in a web-based interface for this use-case scenario. However, in this thesis, I invited the users to be active collaborators who sit behind the wheel and steer the toolmaking process by their subjective measures and personal preferences at every corner. My goal was to embrace their skill and knowledge.

I observed how the users used the meta-tools to serve their goals and simultaneously learned its behavior and gradually managed to adjust their own work to the affordances of the meta-tool. These bi-directional and dynamic interactions were some of the most interesting observations of this study; participants molded the meta-tool to fit their workflow and concurrently adjusted their workflow to fit the meta-tool's affordances.

Despite working remotely, participants of the SecondHand study found various ways to collaborate with each other. They mixed and matched data created by their fellow participants and shared their experiences to improve their work. As such, each outcome was not only the result of an individual's work, but to some extent, the result of the whole cohort's efforts. It was fascinating to see this social context being reflected in the outcome. Even for the ThirdHand study, the close collaboration between me and Mahtab is found its footprint in all stages of the project.

I found the real-time response and communicative visualizations as the two key factors that allowed the SecondHand participants to assess their progress based on their subjective measures. The interactive widgets in the dashboard for data curation, training, and generation of samples allowed them to measure the process against their subjective metrics iteratively. In the ThirdHand study, we substituted this form of visual signals with physical and auditory ones. Mahtab could listen to the notes on the robotic instrument to fine-tune her tool based on her measures. Observing and documenting this transition from pure digital representation on the screen to notes being played on a real instrument was a novel experience for me.

6.2.3 Physical Context

I have discussed the physical context in chapters Chapter 4 and Chapter 5 in detail. Here, I would like to open the discussion on one aspect of physical context that I believe deserves more attention. While designing the SecondHand study, I allowed the participants to decide on their medium of choice, pen and paper, or digital stylus and tablet. Participants mentioned how this choice of physical tool affected their process and results. On the one hand, there were several indications of achieving better results by using thick Sharpie markers rather than thin pens. On the other hand, participants mentioned how using Adobe

Acrobat Pro instead of Adobe Acrobat Reader resulted in cleaner and better handwriting samples. I found these notes as a sign of success, as they could successfully use the meta-tool to capture and represent some aspects of the physical context, i.e., the writing tool, as well as the software that they used. At the very same time, these were signals for me to revisit the definition of context. Reviewing the reflection papers, I realized that limiting the physical context to physical tools and materials was insufficient. I should have also considered the digital/software context. Participants' software packages to scan and edit their handwriting samples were as influential as physical mediums. One could see how the choice between two versions of Adobe Acrobat could result in different results. In the ThirdHand study, we utilized Adobe Audition to post-process the recorded session, and it left its footprint on the sound signature. These software packages were part of the context that I should have taken into consideration.

6.2.4 Data as Interface

In 2016, Dr. Rebecca Fiebrink visited Studio for Creative Inquiry at CMU to deliver a workshop on ML and Art. By then, I had close-to-no experience with ML, but I still managed to make my first ML-based tool in a few minutes using Wekinator. It was my first encounter with interactive ML and using data as interface, and it stayed with me while working on this research.

Throughout the SecondHand study, participants found data curation as a more intuitive interface than coding. Data let them sense the relationship between the input data and the results. This made it easier to control the model and to improve the quality of generated samples by taking course-grain steps toward the desired results. It was interesting to read a reflection paper where a participant described the design of the data collection process as a creative process in itself through which users can get into a conversation with the tool through data.

In the ThirdHand, we could not completely replicate the same experience. The difference between input and output data modalities and the digital-to-physical translation were the barriers in front of us. Nevertheless, we utilized data as the means to steer the training process: Mahtab focused on data generation, and I took over the training phase. This pivot allowed us to explore a different form of a collaborative toolmaking process.

6.3 Benefits for the Creative Practitioners

In the introduction of this thesis, I called this research an opportunity for creative practitioners to meaningfully get involved in the toolmaking process. I argued that the benefit of this approach is twofold, for one, I expected this collaborative toolmaking process to allow them to introduce various aspects of their experience and knowledge to the toolmaking process and to make better tools. Moreover, I expected it to allow creative practitioners to develop a better understanding of the tool and find inspiration to explore new frontiers of creativity that wasn't in reach before. In that capacity, the primary aspiration for the creative practitioners was the opportunity to make tools to serve their creative experiments.

While working on the ThirdHand study, these expectations materialized. The collaborative workflow allowed Mahtab to be an active part of the toolmaking process. Meanwhile, it was also interesting to see how the conversations that helped us form the toolmaking process also helped Mahtab develop a better image of the tool's affordances which later reflected in her demonstration performance. She comprehended the robotic instrument not as a replicator or a replacement for herself, but an experimental tool, an extension of her body, to perform a musical experiment which otherwise she couldn't conduct without help from another musician. This sentiment was well-aligned with my initial expectation of this research.

On the Secondhand study, I observed another form of this experimental creative exploration. Participants, who already had prior exposure to generative ML algorithms, utilized the toolkit to develop an understanding of the C-VAE model and took advantage of this new information to push the ML model to its limits. While some of their efforts ended up in gibberish glyphs, they learned from these and kept on experimenting to find the right direction. I found their efforts to create new typefaces from large conglomerate of data samples as the best representative of my vision for this research.

6.4 Limitations and Future Steps

Before discussing the limitations of this study, I would like to reemphasize the nature of this research that ties it to the specificities of the participants and the context. As each meta-tool was tailored for its unique context, any utilization of it or the results by a different person or for a different purpose defies its original intentions. As such, this research is not directly scalable or generalizable. I do not believe that it is a limitation of this research, but it is its inherent characteristics of it. Nevertheless, the methodology and the implementations that I developed can serve as blueprints for further research efforts.

As we went through a unique situation between 2020 and 2021, external limitations became determining factors in this research. I concluded this research with the two case studies presented here. However, I would prefer to explore other territories as well. I initially intended to collaborate with craftsmen and skilled workers and make tools for skilled trades. For that matter, I reached out to the Pittsburgh Glass Center to collaborate with them on one of the case studies dedicated to hot glass sculpting. Another domain that I wanted to inquire into was choreography. I envisioned that case study to explore a new realm of natural language prompts, motion sequences, and complex relationships between choreographers and performers to push my conception of toolmaking to another level. Both opportunities suddenly vanished with the outburst of the COVID-19 pandemic.

I have already discussed the next steps for each case study inChapter 4 and Chapter 5. Here, I would like to focus on the bigger picture and highlight the challenges that are open to further investigation. The following topics are not limitations but challenges that I could not address with the resources at my disposal.

First, I found bridging the digital/physical barrier a particularly critical issue. In retrospect, when comparing the two case studies, the leap from mere digital presentation to real-world physical recreation brought a barrage of technical challenges to the equation. If it was not for the close collaboration with Mahtab and the mutual understanding of the mechanical system limitations, the ThirdHand study could not be concluded.

A potential next step in this field is to explore bridging the digital/physical barrier. I believe that the solution does not reside in better robotic systems or more sophisticated mechanical contraptions, but we may find it in a different form of collaboration between human agents and machines. We may not need to offload the execution into machines but allow creative practitioners to interpret and execute them. The choreography case study that I mentioned earlier was aimed at addressing this topic. I envisioned the meta-tool allowing the choreographer to translate her expressive prompts into a series of motions, which could facilitate communication between her and the performers. There was a strong case for *physical to digital and back to physical* in that proposal, but the execution was eventually left on the shoulders of the performers, not a set of mechanical contraptions. Unfortunately, the restrictions of the pandemic forced me to put that idea down.

I limited the scope of this research to the atomic actions that form basic tasks. It was a deliberate choice that stemmed from the nature of this research, which values the users' agency over the algorithm's autonomy. A potential field for future investigation is to go beyond these atomic actions while preserving creative practitioners' authority over the process.[2]

In the two case studies, I chose the machine learning models based on what was available at the time, both on the ML algorithms and the hardware side. With the almost daily progress in this field, more powerful models with close to real-time training are getting more accessible to researchers. I am eager to see new frontiers being explored with these new models. I am specifically excited about the affordances that Natural Language Processing (NLP) and learning-based image synthesis methods can bring to the table for creative practitioners, specifically the text-to-image models. The ability to express concepts and prompts in plain words and map them to the latent space of a deep learning model is an exciting opportunity for creative practitioners to interact with their tools. I am curious to see how researchers combine this new opportunity with data-as-interface to let creative practitioners interact more intuitively with their ML-based tools.

In the two case studies, I focused on toolmaking for creative practitioners. However, I see an untouched landscape to explore in the craftsmanship and skilled labor domain. Creating tools for craftspeople with them is a domain with considerable social and economic impact, and I am looking forward to seeing researchers take on that field from a human-centered point of view. I should reiterate that I see the most potential in developing tools that assist craftspeople in their work rather than a stand-alone robotic system that can accomplish a given task.

Finally, collaborative toolmaking between multiple creative practitioners and craftsmen is another territory that I would like to explore. This will allow future researchers to explore different social aspects of toolmaking within the communities of practice.

6.5 Contributions

I believe that the key contribution of this thesis resides in the framework for ML-based toolmaking for creative practitioners. I articulated this approach as a high-level guide for further inquiries in creative-computing toolmaking where the creative practitioner can:

1. collaborate with ML expert toolmakers to integrate ML into their toolmaking process,
2. be at the center of the toolmaking process and establish methods to introduce context to this procedure.

The second contribution of my research is the qualitatively detailed documentation of collaborative ML-based toolmaking for creative practices. This documentation can guide keen readers to comprehend this toolmaking approach in its proper context.[3]

---

[2] This is a very sensitive territory to navigate. The realm of AI and ML is saturated with promises of an autonomous future, where machines take over the tasks that humans once used to fulfill. A consequence of this situation is the skepticism toward any attempts to create ML-based tools due to the fear of replacing people with machinic counterparts. An inquiry geared toward more autonomy in ML-based toolmaking for creative practices may inevitably spark questions about replacing creative practitioners with "creative machines."

[3] I would like to reiterate Collins' point on the tacit aspect of knowledge, even in scientific and highly technical fields by referring the reader to (H. Collins 1974).

The third contributions of this research are the two implementations of the meta-tool. These are blueprints that guide future researchers to design their meta-tools, machine learning algorithms, data pipelines, and user interfaces for collaborative and ML-based toolmaking for creative practitioners.

This research also contributes to the body of knowledge on using generative machine learning algorithms with user-generated small datasets for bespoke toolmaking. It also further contributes to the literature on the generative potential of bias in such datasets that reflect creative practitioners' judgments and subjective metrics.

Bibliography

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. "TensorFlow: A System for Large-Scale Machine Learning." In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA*.

ABB robotics. 2021. "Product Specification - IRB 120." ABB. https://new.abb.com/products/robotics/industrial-robots/irb-120.

ABB Robotics. 2022. "IRB 120." 2022. https://new.abb.com/products/robotics/industrial-robots/irb-120.

Ableton. n.d. "Max for Live." Accessed May 11, 2022. https://www.ableton.com/en/live/max-for-live/.

Adobe. 2022a. "Adobe Acrobat." 2022. https://acrobat.adobe.com/us/en/.

———. 2022b. "Adobe Photoshop." 2022. https://www.adobe.com/products/photoshop.html.

Akten, Memo. n.d. "Learning to See." Accessed July 11, 2022. https://www.memo.tv/works/learning-to-see/.

Akten, Memo, Rebecca Fiebrink, and Mick Grierson. 2019. "Learning to See: You Are What You See." In *ACM SIGGRAPH 2019 Art Gallery*, 1–6.

Amazon.com. n.d. "Amazon Mechanical Turk." Accessed June 5, 2022a. https://www.mturk.com/.

———. n.d. "Amazon Rekognition." Accessed July 27, 2018b. https://aws.amazon.com/rekognition/.

Antoniou, Antreas, Amos Storkey, and Harrison Edwards. 2018. "Augmenting Image Classifiers Using Data Augmentation Generative Adversarial Networks BT - Artificial Neural Networks and Machine Learning – ICANN 2018." In , edited by Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, 594–603. Cham: Springer International Publishing.

Apple. 2022. "IPad." 2022. https://www.apple.com/ipad/.

Argall, Brenna D., Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. "A Survey of Robot Learning from Demonstration." *Robotics and Autonomous Systems* 57 (5): 469–83. https://doi.org/10.1016/j.robot.2008.10.024.

Artamonovskaja, Aleksandra. 2021. "Meet Mario." The Lumen Prize. 2021. https://www.lumenprize.com/blog/meet-mario.

Asada, Haruhiko, and Sheng Liu. 1991. “Transfer of Human Skills to Neural Net Robot Controllers.” In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference On*, 2442–48. IEEE.

Bahnisch, Mark. 2000. “Embodied Work, Divided Labour: Subjectivity and the Scientific Management of the Body in Feredrick W.Taylor’s 1907 ‘Lecture on Management.’” *Body & Society* 6 (1): 51–68.

Bard, Joshua, Ardavan Bidgoli, and Wei Wei Chi. 2018. “Image Classification for Robotic Plastering with Convolutional Neural Network.” In *Robotic Fabrication in Architecture, Art and Design 2018*, 3–15. Springer, Cham. https://doi.org/10.1007/978-3-319-92294-2_1.

Bard, Joshua, Madeline Gannon, Zachary Jacobson-Weaver, Mauricio Contreras, Michael Jeffers, and Brian Smith. 2014. “Seeing Is Doing : Synthetic Tools for Robotically Augmented Fabrication in High- Skill Domains.” In *ACADIA 2014 Proceeding*, 409–16.

Barely, Stephen R., and Julian E. Orr, eds. 1997. *Between Craft and Science, Technical Work in U.S. Settings*. Ithaca and London: Cornell University Press.

Barrat, Robbie. 2017. “Art-DCGAN.” 2017.

Bernardo, Francisco, Mick Grierson, and Rebecca Fiebrink. 2018. “User-Centred Design Actions for Lightweight Evaluation of an Interactive Machine Learning Toolkit.” *Journal of Science and Technology of the Arts* 10 (2): 25–38. https://doi.org/10.7559/citarj.v10i2.509.

Bernardo, Francisco, Michael Zbyszyriski, Rebecca Fiebrink, and Mick Grierson. 2017. “Interactive Machine Learning for End-User Innovation.” *AAAI Spring Symposium - Technical Report*.

Bidgoli, Ardavan, Manuel Ladron De Guevara, Cinnie Hsiung, Jean Oh, and Eunsu Kang. 2020. “Artistic Style in Robotic Painting; a Machine Learning Approach to Learning Brushstroke from Human Artists.” In *Proceedings of the 29th International Conference on Robot and Human Interactive Communication (RO-MAN)*. Naples.

Bidgoli, Ardavan, Eunsu Kang, and Daniel Cardoso Llach. 2019. “Machinic Surrogates: Human-Machine Relationships in Computational Creativity.” In *Proceedings of 25th International Symposium on Electronic Arts, ISEA 2019*. Gwangju, South Korea.

Bidgoli, Ardavan, and Pedro Veloso. 2018. “Deepcloud the Application of a Data-Driven, Generative Model in Design.” In *Recalibration on Imprecision and Infidelity - Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2018*, 176–85. Mexico City: IngramSpark.

Bowers, Q. David. 1972. *Encyclopedia of Automatic Musical Instruments*. Vestal, New York, USA: Vestal press.

Bretan, Mason, Deepak Gopinath, Philip Mullins, and Gil Weinberg. 2016. “A Robotic Prosthesis for an Amputee Drummer,” 1–14. http://arxiv.org/abs/1612.04391.

Brooks, Rodney A. 1991a. “New Approaches to Robotics.” *Science* 253 (5025): 1227–32. https://doi.org/10.1126/science.253.5025.1227.

———. 1991b. “Intelligence without Representation.” *Artificial Intelligence* 47 (1–3): 139–59. https://doi.org/10.1016/0004-3702(91)90053-M.

Brown, John Seely, Allan Collins, and Paul Duguid. 1989. “Situated Cognition and the Culture of Learning.” *Educational Researcher* 18 (1): 32–42.

Brugnaro, Giulio. 2020. “Robotic Training for the Integration of Material Performances in Timber

Manufacturing." University College of London. https://discovery.ucl.ac.uk/id/eprint/10114936.

Brugnaro, Giulio, Angelo Figliola, and Alexandre Dubor. 2019. "Negotiated Materialization: Design Approaches Integrating Wood Heterogeneity through Advanced Robotic Fabrication." In *Digital Wood Design: Innovative Techniques of Representation in Architectural Design, Lecture Notes in Civil Engineering Book Series*, edited by Fabio Bianconi and Marco Filippucci, 135–58. Springer International Publishing. https://doi.org/10.1007/978-3-030-03676-8_4.

Brugnaro, Giulio, and S Hanna. 2017. "Adaptive Robotic Training Methods for Subtractive Manufacturing." In *Proceedings of the 37th Annual Conference of the Association for Computer-Aided Design in Architecture (ACADIA)*, 164–69. Acadia Publishing Company.

Buchanan, Richard. 1992. "Wicked Problems in Design Thinking." *Design Issues* 8 (2): 5–21. https://www.jstor.org/stable/1511637.

Bullock, Jamie, and Ali Momeni. 2015. "Ml.Lib: Robust, Cross-Platform, Open-Source Machine Learning for Max and Pure Data." In *New Interfaces for Musical Expression, NIME 2015*, 3–8. Louisiana, USA.

Buolamwini, Joy, and Timnit Gebru. 2018. "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification." *Proceedings of Machine Learning Research* 81: 1–15. http://proceedings.mlr.press/v81/buolamwini18a/buolamwini18a.pdf.

Caramiaux, Baptiste, Nicola Montecchio, Atau Tanaka, and F R Ed. 2014. "Adaptive Gesture Recognition with Variation Estimation." *ACM Transactions on Interactive Intelligent Systems* 4 (4). https://doi.org/http://dx.doi.org/10.1145/2643204.

Cardoso Llach, Daniel. 2015. *Builders of the Vision Software and the Imagination of Design*. New York: Routledge.

———. 2017. "Data as Interface: The Poetics of Machine Learning in Design." In *Machine Learning – Medien, Infrastrukturen Und Technologien Der Künstlichen Intelligenz*, edited by Christoph Engemann and Andreas Sudmann, 195–218. [transcript-verlag].

Chan, Yick Hin Edwin, and Benjamin Spaeth. 2020. "Architectural Visualisation with Conditional Generative Adversarial Networks (CGAN): What Machines Read in Architectural Sketches." In *Anthropologic: Architecture and Fabrication in the Cognitive Age - Proceedings of the 38th ECAADe Conference*, edited by L Werner and D Koering, 2:299–308. Berlin, Germany.

Chen, Dechen, Dan Luo, Weiguo Xu, Chen Luo, Xia Yan, Liren Shen, and Tianjun Wang. 2020. "Re-Perceive 3D Printing with Artificial Intelligence" 1: 443–50. https://doi.org/10.5151/proceedings-ecaadesigradi2019_034.

Chernova, Sonia, and Andrea L. Thomaz. 2014. "Robot Learning from Human Teachers." *Synthesis Lectures on Artificial Intelligence and Machine Learning* 8 (3): 1–121. https://doi.org/10.2200/S00568ED1V01Y201402AIM028.

Cho, Dahngyu, Jinsung Kim, Eunseo Shin, Jungsik Choi, and Jin Kook Lee. 2020. "Recognizing Architectural Objects in Floor-Plan Drawings Using Deep-Learning Style-Transfer Algorithms." In *RE: Anthropocene, Design in the Age of Humans - Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia, CAADRIA 2020*, 2:719–27. Hong: Association for Computer-Aided Architectural Design Research in Asia (CAADRIA).

Chris Donahue, Ian Simon, and Sander Dieleman. 2018. "Piano Genie: An Intelligent Musical Interface." Magenta. 2018. https://magenta.tensorflow.org/pianogenie.

Christie's. 2018. "Is Artificial Intelligence Set to Become Art's next Medium?" 2018. https://www.christies.com/features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx.

Chung, Junyoung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. 2015. "A Recurrent Latent Variable Model for Sequential Data." *Advances in Neural Information Processing Systems* 28: 2980–88.

Clancey, William J. 1997. *Situated Cognition: On Human Knowledge and Computer Representations*. *Cambridge University Press*. Cambridge, UK: Cambridge university press.

Cohen, Gregory, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. "EMNIST: Extending MNIST to Handwritten Letters." In *2017 International Joint Conference on Neural Networks (IJCNN)*, 2921–26. IEEE.

Collins, Allan, John Seely Brown, and Ann Holum. 1991. "Cognitive Apprenticeship: Making Thinking Visible." *American Educator* 15 (3): 6–11.

Collins, Harry. 1974. "The TEA Set: Tacit Knowledge and Scientific Networks." *Science Studies* 4 (2): 165–85.

———. 2010. *Tacit and Explicit Knowledge*. Chicago: University of Chicago Press.

Csikszentmihalyi, Christopher. 2002. "Tacit Knowledge, Flickering Lasers, and Sweaty Tango." In *Digital Dialogues: Technology and the Hand*. Haystack Mountain School of Crafts.

"CVAT." n.d. Accessed January 2, 2022. https://cvat.org/.

Davis, Daniel. 2013. "Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture." *RMIT University*. http://www.danieldavis.com/papers/danieldavis_thesis.pdf.

Dayma, Boris, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phúc Lê, Luke, and Ritobrata Ghosh. 2022. "DALL-E Mini Explained." W&B. 2022. https://wandb.ai/dalle-mini/dalle-mini/reports/DALL-E-Mini-Explained-with-Demo--Vmlldzo4NjIxODA.

Deshpande, Aditya, Jiajun Lu, Mao-chuang Yeh, Min Jin Chong, and David Forsyth. 2017. "Learning Diverse Image Colorization."

Diagne, Cyril, Nicolas Barradeau, and Simon Doury. 2018. "T-SNE Map." Experiments with Google. 2018. https://experiments.withgoogle.com/t-sne-map.

Docker. n.d. "What Is a Container?" Accessed May 11, 2022. https://www.docker.com/resources/what-container/.

Doersch, Carl. 2016. "Tutorial on Variational Autoencoders." *ArXiv Preprint ArXiv:1606.05908*. https://doi.org/10.3389/fphys.2016.00108.

Dreyfus, Hubert L. 2007. "Why Heideggerian AI Failed and How Fixing It Would Require Making It More Heideggerian." *Philosophical Psychology* 20 (2): 247–68.

Dudley, John J., and Per Ola Kristensson. 2018. "A Review of User Interface Design for Interactive Machine Learning." *ACM Transactions on Interactive Intelligent Systems* 8 (2): 8:1-8:37. https://doi.org/10.1145/3185517.

Duhaime, Douglas. 2017. "Pix Plot." Yale DHLab. 2017. https://dhlab.yale.edu/projects/pixplot/.

During, Jean, Scheherazade Qassim Hassan, and Alastair Dick. 2001. "Santur." Oxford Music Online.

Oxford University Press. 2001. https://doi.org/10.1093/GMO/9781561592630.ARTICLE.51800.

Dutton, Judy. 2012. “Robots Are Already Replacing Us | WIRED.” Wired Magazine. 2012. https://www.wired.com/2012/12/ff-robots-are-already-replacing-us/.

Esteva, Andre, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. 2017. “Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks.” *Nature* 542 (7639): 115–18.

Fails, Jerry Alan, and Dan R. Olsen. 2003. “Interactive Machine Learning.” In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 39–45. ACM.

Fiebrink, Rebecca. 2011. “Real-Time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance.” Princeton University.

———. 2017. “Machine Learning as Meta-Instrument: Human-Machine Partnerships Shaping Expressive Instrumental Creation.” In *Musical Instruments in the 21st Century*, edited by Till Bovermann, Alberto de Campo, Hauke Egermann, Sarah-Indriyati Hardjowirogo, and Stefan Weinzierl, 137–51. Singabpore: Springer.

———. 2019. “Machine Learning Education for Artists, Musicians, and Other Creative Practitioners.” *ACM Transactions on Computing Education* 19 (4). https://doi.org/10.1145/3294008.

Fiebrink, Rebecca, Dan Trueman, and Perry R. Cook. 2009. “A Meta-Instrument for Interactive, On-the-Fly Machine Learning.” In *The Proceedings of the International Conference on New Interfaces for Musical Expression*, 280–85. https://doi.org/10.1016/s1474-6670(17)60085-5.

Fogel, Sharon, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roee Litman. 2020. “Scrabblegan: Semi-Supervised Varying Length Handwritten Text Generation.” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 4323–32. https://doi.org/10.1109/CVPR42600.2020.00438.

Forsythe, Diana E. 2001. *Studying Those Who Study Us: An Anthropologist in the World of Artificial Intelligence*. Stanford: Stanford University Press.

Françoise, Jules, Norbert Schnell, and Frédéric Bevilacqua. 2013. “A Multimodal Probabilistic Model for Gesture-Based Control of Sound Synthesis.” In *MM 2013 - Proceedings of the 2013 ACM Multimedia Conference*, 705–8. Barcelona, Spain. https://doi.org/10.1145/2502081.2502184.

French, Robert M. 1999. “Catastrophic Forgetting in Connectionist Networks.” *Trends in Cognitive Sciences* 3 (4): 128–35. https://doi.org/10.1016/S1364-6613(99)01294-2.

Gamble, Jeanne. 2001. “Modelling the Invisible: The Pedagogy of Craft Apprenticeship.” *Studies in Continuing Education* 23 (2): 185–200. https://doi.org/10.1080/01580370120101957.

“GauGAN2.” n.d. Accessed May 11, 2022. http://gaugan.org/gaugan2/.

Gillian, Nicholas, and Joseph A. Paradiso. 2014. “The Gesture Recognition Toolkit.” *Journal of Machine Learning Research* 15: 3483–87. https://doi.org/10.1007/978-3-319-57021-1_17.

Gillies, Marco, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frederic Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, Saleema Amershi, and Bongshin Lee. 2016. “Human-Centred Machine Learning.” In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 3558–65. ACM.

Goodfellow, Ian. 2016. “NIPS 2016 Tutorial: Generative Adversarial Networks.” *ArXiv Preprint ArXiv:1701.00160*.

Goodfellow, Ian Ij, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. “Generative Adversarial Networks.” In *Advances in Neural Information Processing Systems*, 2672–80. https://doi.org/10.1001/jamainternmed.2016.8245.

GoodNotes. 2022. “GoodNotes.” 2022. https://www.goodnotes.com/.

Google. n.d. “Colaboratory.” https://research.google.com/colaboratory/.

Graves, Alex. 2013. “Generating Sequences with Recurrent Neural Networks.” *ArXiv Preprint ArXiv:1308.0850*.

Ha, David, Jonas Jongejanl, and Ian Johnson. 2017. “Draw Together with a Neural Network.” Google AI. 2017. https://magenta.tensorflow.org/sketch-rnn-demo.

Haines, Tom S F, Oisin Mac Aodha, and Gabriel J Brostow. 2016. “My Text in Your Handwriting.” *ACM Transactions on Graphics (TOG)* 35 (3): 1–18.

HAL Robotics. 2022. “HAL Robotics.” 2022. https://hal-robotics.com/.

Haydu, Jeffrey. 1988. *Between Craft and Class, Skilled Workers and Factory Politics in the United States and Britain 1890-1922*. Berkeley and Los Angeles: University of California Press. https://publishing.cdlib.org/ucpressebooks/view?docId=ft9t1nb603&chunk.id=d0e52&toc.depth=1&toc.id=d0e52&brand=ucpress.

“Hello Magenta.” n.d. Accessed May 11, 2022. https://colab.research.google.com/notebooks/magenta/hello_magenta/hello_magenta.ipynb.

Hiller, Lejaren A., and Leonard M. Isaacson. 1958. “Musical Composition with a High-Speed Digital Computer.” *Journal of the Audio Engineering Society* 6 (3): 154–60. https://www.aes.org/e-lib/browse.cfm?elib=231.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. “Denoising Diffusion Probabilistic Models.” *Advances in Neural Information Processing Systems* 33: 6840–51.

Holzinger, Andreas. 2016. “Interactive Machine Learning for Health Informatics: When Do We Need the Human-in-the-Loop?” *Brain Informatics* 3 (2): 119–31. https://doi.org/10.1007/s40708-016-0042-6.

Hosmer, Tyson, and Panagiotis Tigas. 2019. “Deep Reinforcement Learning for Autonomous Robotic Tensegrity (ART).” In *Ubiquity and Autonomy - Paper Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2019*, 16–29.

Huang, Weixin, and Hao Zheng. 2018. “Architectural Drawings Recognition and Generation through Machine Learning.” In *Recalibration on Imprecision and Infidelity - Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2018*, 156–65.

Hugill, Andrew, and Hongji Yang. 2013. “The Creative Turn: New Challenges for Computing.” *International Journal of Creative Computing* 1 (1): 4. https://doi.org/10.1504/ijcrc.2013.056934.

“Human-Machine Virtuosity – An Exploration of Skilled Human Gesture and Design, Spring 2019.” n.d. Accessed June 19, 2022. https://courses.ideate.cmu.edu/16-455/s2019/.

“Informatics Research Data Repository LIFULL HOME’S Dataset.” n.d. National Institute of Informatics from LIFULL Co., Ltd. Accessed February 2, 2021. https://www.nii.ac.jp/dsc/idr/lifull/.

Isola, Phillip, Jun-Yan Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. “Image-to-Image Translation with Conditional Adversarial Networks.” In *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition, CVPR, 1125–34. Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/CVPR.2017.632.

Jafari, Mohammad. 2021. “GT Music Technology Research - Santoor Bot.” GT Music Technology Research YouTube Channel. 2021. https://www.youtube.com/watch?v=c6yfzXkd6Kg&t=20s&ab_channel=GeorgiaTechSchoolofMusic.

Jafari, Mohammad, and Gil Weinberg. 2021. “Santoor Bot Designing and Building a Robotic Santoor Musician.” 2021. https://music.gatech.edu/santoor-bot.

Juliani, Arthur, Vincent Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Danny Lange, and Marwan Mattar. 2018. “Unity: A General Platform for Intelligent Agents.” *ArXiv*.

Kang, Lei, Pau Riba, Yaxing Wang, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas. 2020. “Ganwriting: Content-Conditioned Generation of Styled Handwritten Word Images.” In *European Conference on Computer Vision*, 273–89. Springer.

Kapur, Ajay. 2005. “A History of Robotic Musical Instruments.” In *International Computer Music Conference, ICMC 2005*.

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. “Progressive Growing of GANs for Improved Quality, Stability, and Variation,” 1–25. https://doi.org/10.1002/joe.20070.

Kayacik, Claire, Sherol Chen, Signe Noerly, Jess Holbrook, Adam Roberts, and Douglas Eck. 2019. “Identifying the Intersections: User Experience + Research Scientist Collaboration in a Generative Machine Learning Interface.” In *Chi ’19 Extended Abstracts*. Glasgow, Scotland UK. https://doi.org/10.1145/3290607.3299059.

Keith McMillen Instruments. 2020. “K-Bow.” 2020. https://www.keithmcmillen.com/labs/k-bow/.

Khean, Nariddh, Alessandra Fabbri, M Hank Haeusler, Student-centred Learning, and Educative Framework. 2018. “Learning Machine Learning as an Architect , How To?” In *Computing for a Better Tomorrow - Proceedings of the 36th ECAADe Conference, AI for Design and Built Environment*, 1:95–102. Lodz, Poland: Lodz University of Technology.

Kingma, Diederik P., Danilo J. Rezende, Shakir Mohamed, and Max Welling. 2014. “Semi-Supervised Learning with Deep Generative Models.” In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 3581–89. Cambridge, MA, USA: MIT Press.

Kingma, Diederik P, and Max Welling. 2013. “Auto-Encoding Variational Bayes.” *ArXiv Preprint ArXiv:1312.6114*, 1–14. https://doi.org/10.1051/0004-6361/201527329.

Kleber, Florian, Stefan Fiel, Markus Diem, and Robert Sablatnig. 2013. “Cvl-Database: An off-Line Database for Writer Retrieval, Writer Identification and Word Spotting.” In *2013 12th International Conference on Document Analysis and Recognition*, 560–64. IEEE.

Klingemann, Mario. n.d. “Quasimondo.” Accessed May 10, 2022. http://quasimondo.com/.

———. 2018. “The Lumen Prize: The Butcher’s Son.” 2018. http://enter.lumenprize.com/node/187.

Kogan, Gene. n.d. “Gene Kogan.” Accessed May 10, 2022a. https://genekogan.com/.

———. n.d. “Machine Learning for Arts.” Accessed May 16, 2022b. https://ml4a.net/.

Koh, Immanuel, and Jeffrey Huang. 2019. “Citizen Visual Search Engine: Detection and Curation of Urban Objects.” *Communications in Computer and Information Science* 1028: 168–82.

https://doi.org/10.1007/978-981-13-8410-3_13.

Lave, Jean, and Etienne Wenger. 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge university press.

Levine, Sergey, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. 2018. "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection." *The International Journal of Robotics Research* 37 (4–5): 421–36.

Li, Chen, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. 2018. "Convolutional Sequence to Sequence Model for Human Dynamics." In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5226–5523. https://doi.org/10.1109/CVPR.2018.00548.

Liang, Ci Jyun, Vineet R. Kamat, and Carol C. Menassa. 2020. "Teaching Robots to Perform Quasi-Repetitive Construction Tasks through Human Demonstration." *Automation in Construction* 120 (December): 103370. https://doi.org/10.1016/j.autcon.2020.103370.

Liwicki, Marcus, and Horst Bunke. 2005. "IAM-OnDB-an on-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard." In *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, 956–61. IEEE.

Luo, Dan, Jingsong Wang, and Weiguo Xu. 2018. "Applied Automatic Machine Learning Process for Material Computation." In *Computing for a Better Tomorrow - Proceedings of the 36th ECAADe Conference, AI for Design and Built Environment*, 1:109–18. Lodz, Poland: Lodz University of Technology.

Maaten, Laurens Van der, and Geoffrey Hinton. 2008. "Visualizing Data Using T-SNE." *Journal of Machine Learning Research* 9 (11).

Magenta. n.d. "Demos." Accessed May 16, 2022a. https://magenta.tensorflow.org/demos.

———. n.d. "Magenta." Accessed April 26, 2019b. https://magenta.tensorflow.org/.

"Making Music with Magenta.Js." n.d. Accessed May 11, 2022. https://hello-magenta.glitch.me/.

Marti, U-V, and Horst Bunke. 2002. "The IAM-Database: An English Sentence Database for Offline Handwriting Recognition." *International Journal on Document Analysis and Recognition* 5 (1): 39–46.

McCloskey, Michael, and Neal J Cohen. 1989. "Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem." In *Psychology of Learning and Motivation*, 24:109–65. Elsevier.

Mccullough, Malcolm. 1996. "Abstracting Craft: The Practiced Digital Hand." In *Abstracting Craft:The Practiced Digital Hand*. MIT press Cambridge, MA.

McDonald, Kyle. n.d. "Kyle McDonald." Accessed May 10, 2022. https://kylemcdonald.net/.

McPherson, Andrew. 2010. "The Magnetic Resonator Piano: Electronic Augmentation of an Acoustic Grand Piano." *Journal of New Music Research* 39 (3): 189–202. https://doi.org/10.1080/09298211003695587.

Memo Akten. n.d. "Memo Akten | Mehmet Selim Akten | The Mega Super Awesome Visuals Company – Artist." Accessed May 10, 2022. https://www.memo.tv/.

Mena, Shahab. 2006. "Acoutstic-Musical Analysis of Santoor." Univeristy of Tehran.

———. 2010. "Historical Background of Santur and Its Performing Styles in Iran Upto 1928 (بررسی

(پیشینه تاریخی و شیوه های اجرایی سنتور در ایران تا 1307 هجری شمسی)." University of Tehran.

Merriam-Webster. n.d. "Tool Definition & Meaning." Accessed May 26, 2022. https://www.merriam-webster.com/dictionary/tool.

Mescheder, Lars, Sebastian Nowozin, and Andreas Geiger. 2017. "Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks." https://doi.org/10.1016/j.aqpro.2013.07.003.

Midjourney Lab. 2022. "Midjourney." 2022. https://www.midjourney.com.

Mildenhall, Ben, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis." In *Computer Vision – ECCV 2020*, edited by Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, 405–21. Cham: Springer International Publishing.

Mirza, Mehdi, and Simon Osindero. 2014. "Conditional Generative Adversarial Nets." *ArXiv Preprint ArXiv:1411.1784*.

Mitchell, Tom M. 1997. *Machine Learning*. McGraw-Hill.

"Ml5.Js: Friendly Machine Learning For The Web." n.d. Accessed May 10, 2022. https://ml5js.org/.

Morgan, Richard. 2013. "The (Robot) Creative Class." The New York Magazine. 2013. https://nymag.com/news/intelligencer/robot-jobs-2013-6/.

Naseck, Perry, Andey Ng, and Mary Tsai. 2019. "Dehumanized Graffiti." Human-Machine Virtuosity Website. 2019. https://courses.ideate.cmu.edu/16-455/s2019/1315/dehumanized-graffiti/.

Nauata, Nelson, Kai Hung Chang, Chin Yi Cheng, Greg Mori, and Yasutaka Furukawa. 2020. "House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation." *ArXiv Preprint ArXiv:2003.06988* 12346 LNCS: 162–77. https://doi.org/10.1007/978-3-030-58452-8_10.

Newton, David. 2020. "Deep Generative Learning for the Generation and Analysis of Architectural Plans with Small Datasets." In *Architecture in the Age of the 4th Industrial Revolution - Proceedings of the 37th ECAADe and 23rd SIGraDi Conference*, edited by JP Sousa, JP Xavier, and G Castro Henriques, 2:21–28. Porto, Portugal. https://doi.org/10.5151/proceedings-ecaadesigradi2019_135.

"NIST Special Database 19." n.d. Accessed September 6, 2021. https://www.nist.gov/srd/nist-special-database-19.

Novak, Matt. 2012. "Musicians Wage War Against Evil Robots | History." Smithsonian Magazine. 2012. https://www.smithsonianmag.com/history/musicians-wage-war-against-evil-robots-92702721/?no-ist.

NVIDIA. n.d. "NVIDIA Canvas: Turn Simple Brushstrokes into Realistic Images." Accessed May 11, 2022. https://www.nvidia.com/en-us/studio/canvas/.

Oord, Aäron Van Den, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. 2016. "WaveNet: A Generative Model for Raw Audio." *ArXiv Preprint ArXiv:1609.03499.*

Oord, Aaron van den, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. 2016. "Conditional Image Generation with PixelCNN Decoders." https://doi.org/10.1016/j.optmat.2013.09.026.

OptiTrack. n.d. “Markers - NaturalPoint Product Documentation Ver 2.2.” Accessed July 22, 2021. https://v22.wiki.optitrack.com/index.php?title=Markers.

———. 2022. “Motive.” 2022. https://optitrack.com/software/motive/.

Perez, Luis, and Jason Wang. 2017. “The Effectiveness of Data Augmentation in Image Classification Using Deep Learning.” http://arxiv.org/abs/1712.04621.

Plotly. 2022. “Dash.” 2022. https://plotly.com/dash/.

Polanyi, Michael. 1966a. “The Logic of Tacit Inference.” *Philosophy* 68 (155): 1–18. http://www.jstor.org/stable/3749034.

———. 1966b. *The Tacit Dimension*. *Knowledge in Organisations*. Garden City, New York: Doubleday & Company, inc.

———. 2005. *Personal Knowledge: Towards a Post-Critical Philosophy*. London: Routledge.

Poličar, Pavlin. 2020. “OpenTSNE: Extensible, Parallel Implementations of t-SNE.” 2020. https://opentsne.readthedocs.io/.

Pu, Yunchen, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. 2016. “Variational Autoencoder for Deep Learning of Images, Labels and Captions.” In *Advances in Neural Information Processing Systems 29*, 2352–60. https://doi.org/10.1109/ICCV.2017.245.

Radford, Alec, Luke Metz, and Soumith Chintala. 2015. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks.” *ArXiv Preprint ArXiv:1511.06434*. https://doi.org/10.1051/0004-6361/201527329.

Rahwan, Iyad, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean François Bonnefon, Cynthia Breazeal, Jacob W. Crandall, et al. 2019. “Machine Behaviour.” *Nature* 568 (7753): 477–86. https://doi.org/10.1038/s41586-019-1138-y.

Ramesh, Aditya, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. “Hierarchical Text-Conditional Image Generation with Clip Latents.” *ArXiv Preprint ArXiv:2204.06125*.

Ramesh, Aditya, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. “Zero-Shot Text-to-Image Generation.” In *International Conference on Machine Learning*, 8821–31. PMLR.

Ramsgaard Thomsen, Mette, Paul Nicholas, Martin Tamke, Sebastian Gatz, Yuliya Sinke, and Gabriella Rossi. 2020. “Towards Machine Learning for Architectural Fabrication in the Age of Industry 4.0.” *International Journal of Architectural Computing* 18 (4): 335–52. https://doi.org/10.1177/1478077120948000.

Razavi, Ali, Aaron Van den Oord, and Oriol Vinyals. 2019. “Generating Diverse High-Fidelity Images with vq-Vae-2.” *Advances in Neural Information Processing Systems* 32.

Rebecca Fiebrink. 2016. “Training Data as a User Interface.” Art with MI. 2016. https://youtu.be/zzadrm3SPrQ.

Reddy, Michael. 1979. “The Conduit Metaphor.” In *Metaphor and Thought*, edited by Andrew Ortony. Vol. 2. Cambridge: Cambridge University Press.

Ridler, Anna. n.d. “Anna Rridler.” Accessed May 10, 2022. http://annaridler.com/.

Roberts, Adam, Jesse Engel, Yotam Mann, Jon Gillick, Claire Kayacik, Signe Nørly, Monica Dinculescu,

Carey Radebaugh, Curtis Hawthorne, and Douglas Eck. 2019. “Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live.” https://storage.googleapis.com/pub-tools-public-publication-data/pdf/ab2b1a3cf41bd6c42b1d599382a40e05c22bea3b.pdf.

Roberts, Adam, Jesse Engel, Colin Raffel, Ian Simon, and Curtis Hawthorne. 2018. “MusicVAE: Creating a Palette for Musical Scores with Machine Learning.” 2018. https://magenta.tensorflow.org/music-vae.

Roberts, Adam, Curtis Hawthorne, and Ian Simon. 2018. “Magenta.Js: A JavaScript API for Augmenting Creativity with Deep Learning.” In *Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden. https://goo.gl/magenta/py.

Roberts, Adam, Yotam Mann, Jesse Engel, and Carey Radebaugh. 2019. “Magenta Studio.” 2019. https://magenta.tensorflow.org/studio-announce.

Roggema, Rob. 2016. “Research by Design: Proposition for a Methodological Approach.” *Urban Science* 1 (1): 2. https://doi.org/10.3390/urbansci1010002.

Rossi, Gabriella, and Paul Nicholas. 2018. “Modelling A Complex Fabrication System, New Design Tools for Doubly Curved Metal Surfaces Fabricated Using the English Wheel.” In *Computing for a Better Tomorrow: Proceedings of the 36th International Conference on Education and Research in Computer Aided Architectural Design in Europe*, 1:811–20.

———. 2019. “Haptic Learning: Towards Neural-Network-Based Adaptive Cobot Path-Planning For.” In *Architecture in the Age of the 4th Industrial Revolution - Proceedings of the 37th ECAADe and 23rd SIGraDi Conference*, edited by JP Sousa, JP Xavier, and G Castro Henriques, 2:201–10. Porto, Portugal: University of Porto.

Rowe, Robert. 2001. *Machine Musicianship*. Cambridge, Mass: MIT press Cambridge, MA.

RunwayML. 2019. “Runway: Artificial Intelligence for Augmented Creativity.” 2019. https://runwayapp.ai/.

Saharia, Chitwan, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, and Rapha Gontijo Lopes. 2022. “Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding.” *ArXiv Preprint ArXiv:2205.11487*.

Salimans, Tim, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. “Improved Techniques for Training GANs,” 1–10. https://doi.org/arXiv:1504.01391.

Schaffer, Simon. 1994. “Babbage’ s Intelligence : Calculating Engines and the Factory System.” *Critical Inquiry* 21 (1): 203–27.

Schön, Donald. A. 1992. “Designing as Reflective Conversation with the Materials of a Design Situation.” *Research in Engineering Design* 3: 131–47. https://doi.org/10.1016/0950-7051(92)90020-G.

Schwartz, Thibault. 2013. “HAL.” In *Rob| Arch 2012*, edited by Sigrid Brell-Çokcan and Johannes Braumann, 92–101. Vienna: Springer-Verlag Wien. https://doi.org/10.1007/978-3-7091-1465-0_8.

Seeman, Melvin. 1959. “On The Meaning of Alienation.” *American Sociological Review* 24 (6): 783–91. https://www.jstor.org/stable/2088565.

Sengers, Phoebe. 1996. “Socially Situated AI: What It Means and Why It Matters.” In *Proceedings of the 1996 AAAI Symposium, Entertainment and AI/A-Life. Technical Report WS-96-03*, 69–75.

———. 1998. “Anti-Boxology: Agent Design in Cultural Context.” *English*. Carnegie Mellon University. http://ra.adm.cs.cmu.edu/anon/usr/ftp/1998/CMU-CS-98-151.pdf.

Sennett, Richard. 2008. *The Craftsman*. Yale University Press.

Simard, Patrice Y., Saleema Amershi, David M. Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, et al. 2017. “Machine Teaching: A New Paradigm for Building Machine Learning Systems.” http://arxiv.org/abs/1707.06742.

Singer, E, J Feddersen, C Redmon, and B Bowen. 2004. “LEMUR’s Musical Robots.” In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 181–84.

Smith, Adam. 1776. *The Wealth of Nations*.

Suchman, Lucy A. 1985. *Plans and Situated Actions:The Problem of Human-Machine Communication*. Xerox.

Taylor, Feredrick Winslow. 1911. *The Principles of Scientific Management*. *Harper & Brothers*. The Floating Press, 2012.

“Teachable Machine.” n.d. Accessed May 10, 2022. https://teachablemachine.withgoogle.com/.

Unity-Technologies. 2021. “ML-Agents.” 2021. https://github.com/Unity-Technologies/ml-agents/blob/release_19_docs/docs/Getting-Started.md.

Valenzuela, Cristobal. 2019. “Porting a Machine Learning Model from GitHub to RunwayML in 5 Minutes ? | Runway Blog.” 2019. https://runwayml.com/blog/porting-a-machine-learning-model-from-github-to-runwayml-in-5-minutes/.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. “Attention Is All You Need.” *Advances in Neural Information Processing Systems* 30.

Veloso, Pedro, Jinmo Rhee, Ardavan Bidgoli, and Manuel Ladron De Guevara. 2022. “A Pedagogical Experience with Deep Learning for Floor Plan Generation.” In *POST-CARBON, Proceedings of the 27th International Conference of the Association for ComputerAided Architectural Design Research in Asia (CAADRIA) 2022*, 373–82. Hong Kong: Association for Computer-Aided Architectural Design Research in Asia (CAADRIA).

Wacom. 2022. “Wacom | Interactive Pen Displays , Pen Tablets and Stylus Products.” 2022. https://www.wacom.com/en-us.

Walker, Jacob, Carl Doersch, Abhinav Gupta, and Martial Hebert. 2016. “An Uncertain Future: Forecasting from Static Images Using Variational Autoencoders.” *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9911 LNCS: 835–51. https://doi.org/10.1007/978-3-319-46478-7_51.

WandB. 2020. “Hyperparameter Tuning - Documentation.” 2020. https://docs.wandb.ai/guides/sweeps.

Wegner, Peter. 1996. “Interoperability.” *ACM Computing Surveys (CSUR)* 28 (1): 285–87.

Weinberg, Gil, Mason Bretan, Guy Hoffman, and Scott Driscoll. 2020. *Robotic Musicianship: Robotic Musicianship Embodied Artificial Creativity and Mechatronic Musical Expression*. Springer International Publishing. https://doi.org/10.1007/978-3-030-38930-7.

Wekinator. n.d. “Downloads | Wekinator.” Accessed May 11, 2022. http://www.wekinator.org/downloads/.

White, Tom. n.d. “Cello.” Accessed July 11, 2022. https://drib.net/art/cello.

Witten, Ian H, and Eibe Frank. 2002. “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.” *Acm Sigmod Record* 31 (1): 76–77.

Yu, Jiahui, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, and Burcu Karagol Ayan. 2022. “Scaling Autoregressive Models for Content-Rich Text-to-Image Generation.” *ArXiv Preprint ArXiv:2206.10789*.

Zhao, Jieyu, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. “Men Also Like Shopping: Reducing Gender Bias Amplification Using Corpus-Level Constraints.” https://doi.org/10.18653/v1/D17-1323.

Zhu, Jun-yan, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. “Unpaired Image-to-Mage Translation Using Cycle-Consistent Adversarial Networks.” *ArXiv Preprint*, 2223–32. https://doi.org/10.1109/ICCV.2017.244.

پایور, فرامرز. 1359. *دوره عالی برای سنتور ردیف چپ کوک*.

Appendix I: Conditional Variational AutoEncoders (VAEs)

This chapter is a detailed discussion of the machine learning architecture that is used as the backend for the two case studies. The code and implementation for both case studies will be available on the GitHub repositories https://github.com/Ardibid/SecondHand and https://github.com/Ardibid/ThirdHand.[1]

[1] The two repositories will be publicly available at the end of the embargo period.

7.1 Deep Generative Machine Learning Models

A generative model (GM) in machine learning refers to a model that can be trained on an unlabeled subset of the distribution p_{data} and learns an estimates representation of that distribution, p_{model}.[2] We can draw novel samples from p_{model}, which did not exist in the distribution p_{data}, but closely resemble them. From this point of view, generative models differ from the discriminative models that map features to labels and have been widely used for tasks like image classification. GMs are specifically useful for working with high-dimensional data distribution, i.e., images, and multi-modal data spaces, i.e., natural language and images. Moreover, they can also be used for model-based reinforced learning and making predictions with missing inputs when trained with missing data (I. Goodfellow 2016).

While there are various general architectures for generative models (Figure 90), in this thesis, I will focus on machine learning deep generative models based on maximum likelihood. These models can be categorized based on the method by which they approximate the likelihood. One branch of models can explicitly estimate p_{data} and generate samples directly from it; the other can only generate samples from p_{model} (I. Goodfellow 2016). A handful of deep generative architectures have a track record of compelling synthesis of images, text, and other data formats in various fields: 1) Variational AutoEncoders (VAE) (Kingma and Welling 2013), 2) Autoregressive models (Aäron Van Den Oord et al. 2016), and 3) Generative Adversarial Networks (GANs) (I. I. Goodfellow et al. 2014).[3] Among these architectures, this research focuses on VAEs and specifically conditioned VAEs (C-VAEs).

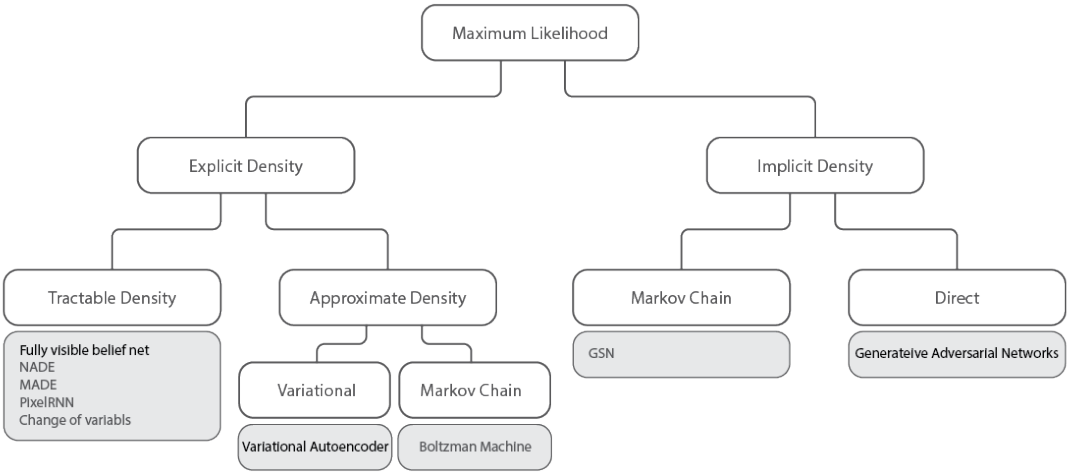


Figure 90. Deep generative models categorization (I. Goodfellow 2016).

[2] There are other definitions for generative modeling, i.e., Doersch defines it as "… a broad area of machine learning which deals with models of distribution $P(X)$, defined over datapoints X in some potentially high-dimensional space X" (2016, 1).

[3] In the summer of 2022, when I was working on the final versions of this document, deep generative models based on Diffusion (Ho, Jain, and Abbeel 2020) and Attention (Vaswani et al. 2017), best-known by the project such as Dall-E (Ramesh et al. 2022) and Midjourney (Midjourney Lab 2022) were the center of attention, not only among the researchers, but also public.

7.2 Encoder/Decoder Architecture

Before introducing VAEs, getting familiar with the Encoder/Decoder architecture and one of its commonly used examples, AutoEncoders (AE), is enlightening. Encoding/decoding refers to mapping or embedding an input data into a latent representation, then decoding it into the same or another representation. The input and output of the system might be of the same modality, i.e., from image to image, or of two different modalities, i.e., from text to image. AutoEncoder models form a subcategory of encoder/decoder architecture that is designed to receive a data point, map it to a latent dimension, and then reconstruct it back to the original format as accurately as possible.

An interesting feature of AutoEncoders is their latent representation of data distribution. Once the model is trained, the decoder can be tricked with synthesized latent representations to generate novel examples that resemble the original data distribution. For instance, Bidgoli and Veloso utilized an AutoEncoder to create an early stage-of-design prototyping tool with point cloud data (Bidgoli and Veloso 2018). The latent representation in an AE is not regulated, and data distribution can be quite sparse. This means that drawing an acceptable sample from the latent can be difficult.

7.3 Variational Autoencoders

Variational Autoencoders (Kingma and Welling 2013) is an architecture of generative models that leverage explicit representation of the likelihood. While the VAE architecture resembles Autoencoder architecture, they are not identical (Figure 91). Like AEs, VAEs have encoder and decoder networks. The encoder, usually denoted as $q_\theta(z|x)$, trains on the input data x to learn the features and encodes them in a latent representation space z which is usually of lower dimension, referred to as the bottleneck. In a VAE, this latent space itself is a distribution, usually normal distribution $p(z) = N(0, I)$, represented by two vectors for means and standard deviation. To generate a sample of the latent space z, we can draw a sample from this distribution. The decoder, denoted as $p_\emptyset(x|z)$, is another network that receives the samples from z and outputs samples from the distribution of the x.

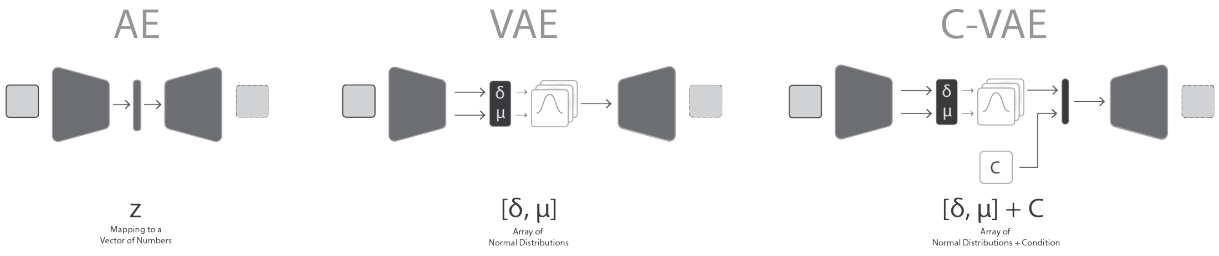


Figure 91. Autoencoder (left), VAE (middle), C-VAE (right) architecture.

In the training process, the encoder reduces the data dimension and embeds it in the latent space z. The decoder aims to get the latent representation and reconstruct the original input data. Some data will be lost in the encoding process as the input is compressed into a lower dimension. When the decoder reconstructs the input from the latent representation, the outcome will not be identical to the input. The objective of VAE is to reduce the lost data between the input and the reconstructed output while keeping the latent space distribution as close as possible to the standard normal distribution.

Accordingly, the loss function for a VAE consists of two parts, 1) a reconstruction loss which observes the decoder performance in reconstructing samples, and 2) a Kullback-Leibler Divergence (KLD) that describes how close q is to p (Equation 1 and Figure 92).

$$l_i(\theta, \emptyset) = -E_{z\sim q_\theta(Z|X)}[\log p_\emptyset(x^i|z)] + KL(q_\theta(z|x)||p(z)) \quad 1$$

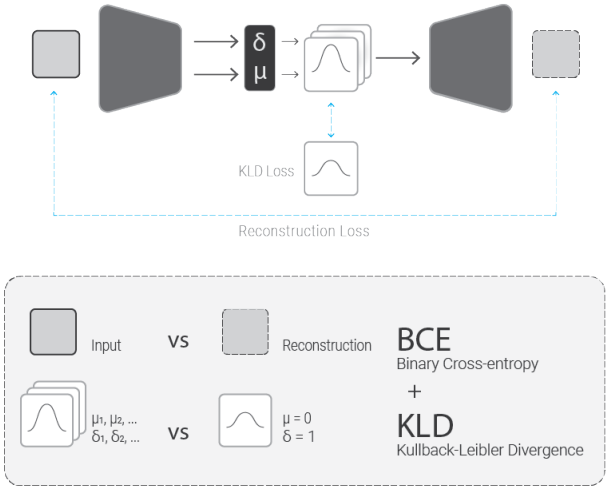


Figure 92. VAE loss function components.

VAEs can be trained with gradient descent. The goal is to optimize the loss w.r.t. encoder and decoder parameters $\emptyset$ and θ. However, in training, it is necessary to sample from the distribution, which blocks the gradient descent. To overcome this problem, a reparameterization trick is applied using $z = \mu + \sigma.\varepsilon$ where $\varepsilon \sim N(0,1)$. ε adds stochastic to the model, but since we do not need to optimize it, it will not block the gradient descent.

VAEs have been used for image processing, namely generating handwriting digits, faces, house numbers, etc., denoising images, segmentation, physical simulation, segmentation, inpainting, generating captions for images (Pu et al. 2016), image colorization (Deshpande et al. 2017), forecasting from static image (Walker et al. 2016), interpolating between sequence of drawings (Ha, Jongejanl, and Johnson 2017), and large-scale image generation (Razavi, Van den Oord, and Vinyals 2019) (Figure 93).

Figure 93. Variational Autoencoder used for large-scale image generation: class-conditional 256x256 image samples from a VQ-VAE-2 model trained on ImageNet. Images from (Razavi, Van den Oord, and Vinyals 2019).

7.4 Computational Complexity

In VAEs the data likelihood can be calculated with this equation:

$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz \qquad 2$$

To do so, it is necessary to 1) define the latent variable to capture the latent information and 2) handle the integral over z, which is intractable. It is assumed that the samples can be drawn from a simple distribution, like a normal distribution, and later map them by a complex function to overcome the first problem.

The second problem is more challenging, as the equation is intractable, it is not possible to compute $p(x|z)$ for every possible z. Accordingly, the posterior density is intractable. To overcome this issue, we need to define an additional encoder $q_\emptyset(z|x)$ to approximate the decoder $p_\theta(z|x)$. This trick allows us to calculate a lower bound of the data likelihood, which is tractable and can be optimized. It can be demonstrated that after this trick, the log-likelihood can be arranged as:

$$logp_\theta(x^{(i)}) = E_z[logp_\theta(x^{(i)}|z)] - \ D_{KL}(q_\emptyset(z|x^{(i)}) \parallel p_\emptyset(z)) + \ D_{KL}(q_\emptyset(z|x^{(i)}) \parallel p_\emptyset(z|x^{(i)})) \qquad 3$$

Only the third one is intractable among the three terms on the right side, but we know that it is always greater or equal to zero. So, this can be rearranged as an inequation:

$$logp_\theta(x^{(i)}) \geq E_z[logp_\theta(x^{(i)}|z)] - \ D_{KL}(q_\emptyset(z|x^{(i)}) \parallel p_\emptyset(z)) \qquad 4$$

Thus, the variational lower bound is:

$$logp_\theta(x^{(i)}) \geq \mathcal{L}\ (x^{(i)}, \theta, \emptyset) \qquad 5$$

And for training, we need to maximize the lower bound:

$$\theta^*, \emptyset^* = \arg\max_{\theta,\emptyset} \sum_{i=1}^{N} \mathcal{L}\ (x^{(i)}, \theta, \emptyset) \qquad 6$$

7.5 Conditional VAEs

Kingma et al. discussed the possibility of using a conditional approach in generative models. They propose that conditioning can be utilized as a tool to "explore the underlying structure of data" (Kingma et al. 2014, 3587). Their paper reports on two demonstrations of the affordances of conditional models as a tool to explore the latent space of generative models and the possibility of content/style separation through fixing the label (y) and navigating the latent space.

In the demonstrations, they utilized a C-VAE model with a simple 2-D latent space trained on the MNIST dataset. In the first demo, they navigated the latent space—with values between -5 and 5 while feeding the model with a fixed label. They observed that nearby regions of the latent space corresponded to a similar handwriting style while the content, the letter, was constant (Figure 94). In the second demo, the authors passed a sample to the encoder network and produced the latent representation vector z. Then

they passed this z vector with various label vectors to the decoder model. The results show that the generated samples corresponded to the style of the original sample, while the generated number was associated with the label vector, demonstrating a dismantlement of style and content (Figure 95).

Figure 94. Navigating the latent space with a fixed label, in each plot, the label (2, 3, and 4 from left to right) where kept fixed while the latent vector (z) was changing, image from (Kingma et al. 2014, 3588).

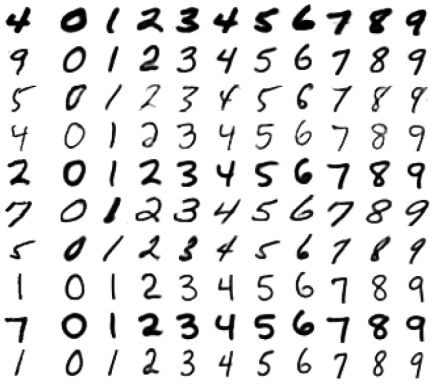


Figure 95. Original samples used to create z vectors (left), the samples generated based on the z vectors and various label vectors (right), image from (Kingma et al. 2014, 3588).

7.6 Navigating the Latent Space of VAE and C-VAE

We can use a VAE to generate new samples by feeding it with a latent vector. The regulated latent space of VAE makes it easier to create this latent vector by sampling from a multivariate standard normal distribution. In VAEs, the enforced regulation on the latent space—sanctioned by the KLD loss—renders this approach a viable solution to get meaningful results. The KLD loss forces the encoder model to map the input samples as close as possible to a standard normal distribution with $\mu = 0$ and $\delta^2 = 1$.[4] As such,

[4] Gaussian distribution is a deliberate choice in this case, but it might be possible to replace it with other distributions too.

we are sampling the latent vector randomly from a multivariate standard normal distribution will result in generating a meaningful sample. [5]

While the mere generation of a valid sample might be satisfying for a general-purpose generative model, the user needs to generate samples with special prior conditions in the case of this study. For instance, while generating a handwriting typeface, a user needs to generate a specific glyph—i.e., a, A, g, or ?—rather than generating a random glyph each time it taps on the generative model.

Extracting the latent vector from a given sample can partially mitigate this challenge. This sampling method utilizes the encoder model to map a given input x to the latent space and find the corresponding z vector. Users can use this vector as a starting point. For an observation x, z is drawn from the prior distribution $q_\theta\,(z|x)$ (Equation 7). The user can apply a deliberate sequence of latent space arithmetic on z to create a new vector z' (Equation 8). The new sample $\hat{x}$ can be generated by feeding z'through the decoder model $p_\theta(\hat{x}|z')$ (Equation 9).

$$z \sim q_\emptyset\,(z|x) \qquad 7$$

$$z' = z + a,\;\; z, a \in \mathbb{R}^N \qquad 8$$

$$\hat{x} \sim p_\theta\,(x, z') \qquad 9$$

An interesting feature of this approach is the possibility of conditioning the generative process on multiple input samples as the starting seeds. This feature allows the user to start from a deliberate set of samples, navigate the latent space between them, and observe the results in real time until it lands on a satisfying solution (Equations 10-12, Figure 96 middle).[6]

---

[5] One of the main challenges if using an AE as a generative model is that such sampling methods cannot always return acceptable results. Due to AutoEncoder's unregulated way of mapping inputs to the latent space, valid samples can be unevenly distributed at virtually any point in the n-dimensional latent space. Even an arbitrary point in the proximity of a legitimate sample might not be associated with a new valid sample.

[6] These methods are not exclusive to VAEs. AE models can also take advantage of these latent space navigation methods. Project DeepCloud (Bidgoli and Veloso 2018) takes advantage of both workflows with its AutoEncoder backend. In one mode, users could start from a sample, find its latent representation, and traverse the latent space by manipulating each element of the latent vector individually. The observations were quite interesting, i.e., when working on a dataset of chairs, an element of the latent vector was primarily responsible for growing handles, while another could control the number of legs. As an unsupervised generative workflow, this approach wasn't guaranteed to automatically find such interesting elements in the latent vectors. Consequently, users had to manually inspect the effects of each element of the latent vector to spot the interesting ones.

In its other mode, DeepCloud could map multiple input samples $X = \{x_1, x_2, \dots, x_n\}$ to the latent space and create the latent vectors $Z = \{z_1, z_2, \dots, z_n\}$. DeepCloud interface provides the user with tools to apply latent space arithmetic to interpolate between the latent representations of the inputs and create a new latent vector $\hat{z}$. In DeepCloud, the authors implemented this process by multiplying each latent vector by γ coefficients that apply deliberate weights to each latent vector individually. Users manually adjust γ coefficients for each latent vector z by changing sliders and knobs on a KORG midi controller. deck—a ubiquitous piece of hardware among musicians. This interpolated latent vector is then passed to the decoder model to generate novel hybrid instances.

$$Z \sim q_{\emptyset}(Z|X) \quad 10$$

$$\hat{z} = \sum_{i=0}^{n} \frac{\gamma_i z_i}{n}, \;\; z_i \in Z, \mathbb{R}^N, \gamma_i \in \mathbb{R} \quad 11$$

$$\hat{x} \sim p_{\theta}(\hat{x}|\hat{z}) \quad 12$$

Another approach to gain some control over the generative process is to incorporate the conditioning signals into the model and the training phase. Conditional generative models—such as C-GANs (Isola et al. 2017; Mirza and Osindero 2014) and C-VAEs (Aaron van den Oord et al. 2016; Kingma et al. 2014)—have proven the effectiveness of this approach through condition the sampling process to a piece of prior information. In general, a C-VAE works with pairs of data $(X, Y) = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ where $x_i \in \mathbb{R}^D$ is the i^{th} sample of data and $y_i \in \{1, \ldots, L\}$ is its designated label.

While C-VAEs do not need a starting observation to serve as a seed, they can still take advantage of the seeding method by drawing the z vector from $q_{\theta}(z|x)$ and then generate a new sample x, conditioned on this latent vector and a deliberate y signal $p_{\theta}(\hat{x}|y, z)$ (Equations 13-15, Figure 96 right).

$$z \sim q_{\emptyset}(z|x) \quad 13$$

$$z' = z + a, \;\; z, a \in \mathbb{R}^N \quad 14$$

$$\hat{x} \sim p_{\theta}(\hat{x}|y, z) \quad 15$$

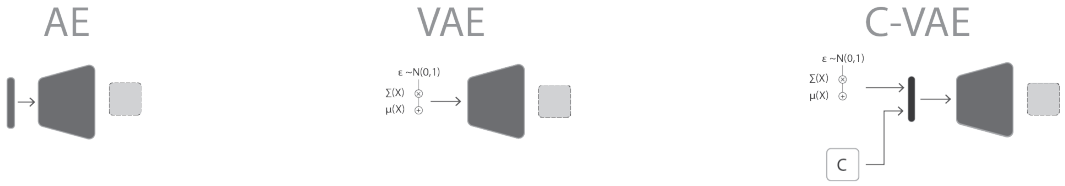


Figure 96. Drawing samples from AE, VAE, C-VAE.

7.7 VAE vs. GAN

It is common to compare VAEs and GANs as the two architectures of deep generative models. GANs are generally the preferred choice for ML tasks that involve natural-looking images, and there is a growing body of literature on using complex GAN architectures to generate photo-realistic, high-resolution, and sharp-looking images. Moreover, GANs only need one pass through the model to generate new samples, which results in quicker time-to-response (I. I. Goodfellow et al. 2014).

However, without a clear objective function, the training process is not stable and straightforward. GANs training process is prone to mode collapse, where the model concentrates on a limited region of the training set and fails to present other regions. This will result in clusters of similar instances among a pool of outputs. Another issue is that vanilla GANs cannot scale up in pixel resolution, and even with the modified architectures, they may start generating fractal artifacts. Finally, the lack of a well-structured latent space means optimization methods should be used to find the latent representation of a specific input signal.

In contrast, training a VAE model is usually faster, easier, and more stable, especially with small datasets. One obstacle in developing deep generative models is the notoriously challenging training process. It is common to see methods that rely on 1) assumptions on the structures in the data, 2) excessive approximation, or 3) computationally expensive inference models to overcome this issue. The application of Neural Networks as a powerful function approximation method, combined with the use of backpropagation to train them, helped address some of these issues and led to significant progress in GMs (Doersch 2016). A major advantage of VAE architecture is that it does not rely on strong assumptions about the data structure, and the model can be trained fast using backpropagation (Doersch 2016).

Moreover, VAEs are not prone to problems such as mode collapsing and spotting general training issues such as overfitting is trivial. They provide a regularized latent space that can be used to generate novel samples and apply latent space arithmetic. Accordingly, when more precise control over the generation process is critical, VAEs provide more control over the generation process.

With all the advantages of VAEs over GANs, it should be noted that VAEs generally produce blurry results compared to GANs, mostly because the inference model they rely on for training is not expressive enough to capture details of the distribution (Mescheder, Nowozin, and Geiger 2017). This issue can also be associated with the MSE element in the loss function (Karras et al. 2017). MSE usually results in blurry outcomes that may miss subtle but critical details, i.e., details of complex geometries. It also may fail to detect objects that are not big or bold enough in a given scene.

7.8 C-VAE Models Used for Case Studies

The C-VAE model used in the SecondHand and ThirdHand studies—based on the model introduced by Kingma et al. (2014), generates the samples based on a latent vector z as well as a latent label variable y (Figure 97 and Figure 98). It generates sample $\hat{x}$ from the conditional distribution with $p_\theta\ (\hat{x}|y, z)$, where y is a conditioning label vector, and z is the latent vector. The main advantage of this approach lies in the y signal, which can be a vector representing the one-hot-encoding of labels, an embedded signal, or other forms of information. This powerful yet flexible conditioning capability was a major derive behind choosing it as the backend for this study.

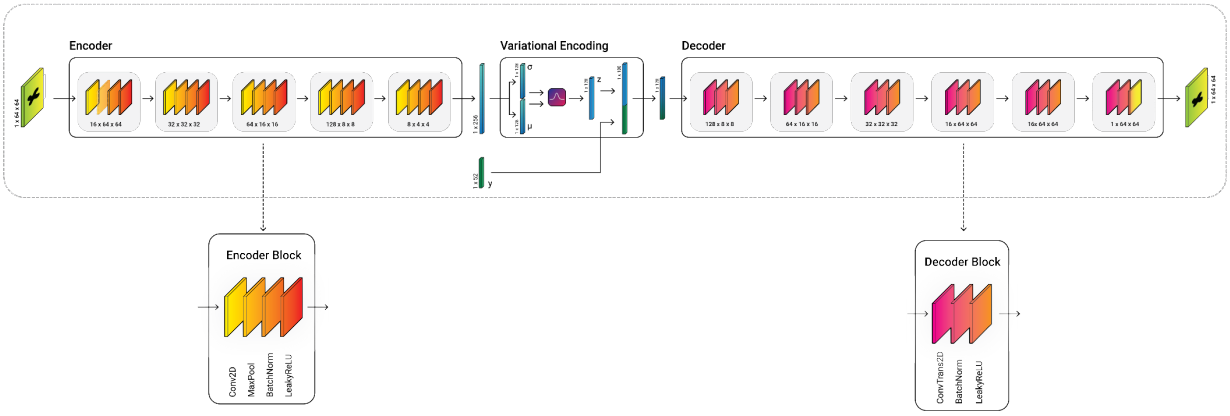


Figure 97. C-VAE model used in the SecondHand study. Notice the different blocks used in the encoder and decoder as well as the condition vector concatenated to the latent space output.

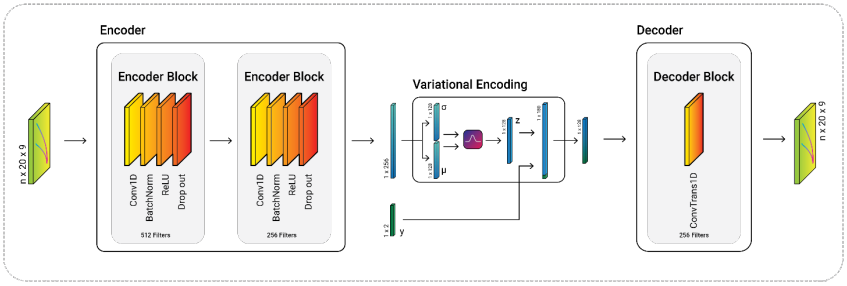


Figure 98. C-VAE model used in the ThirdHand study. Note the shallow encoder and single-layer decoder networks.

Appendix II: The Context

This chapter is a discussion on the underlying context of this thesis: the social, economic, and historical aspects of skill, learning, and toolmaking. Throughout this appendix, I walk back in time and revisit various conceptions of skill, as a form of knowledge, and explain how the mere act of toolmaking has long been a matter of political, social, and economic debate.

This brief context is a prelude to introducing situated ML-based toolmaking for creative practitioners that stands on two pillars, the creative practitioner, and elements of the physical context. While it contributes to the understanding of situated approach to toolmaking, the in-depth discussions that are presented here may disrupt the fellow of this dissertation. Accordingly, I found it more suited to be presented as an appendix rather than a chapter within the main body of this dissertation.

8.1 Skill

A tool without its user is a dormant object. What transforms a tool is the craftsman and their skill. In this section, I will introduce three conceptions of skill in their historical and social contexts [1] to answer three primary questions:

> Why codifying skills in tools and automating work have been enticing yet, elusive for centuries?
> How have attempts to codify and automate work reconfigured conceptions of learning and skill?
> How have these attempts influenced toolmaking procedures?

8.1.1 Skill Situated in its Context

According to American sociologist Richard Sennet, medieval workshops were the epitome of craftsmanship. A young apprentice would spend years practicing among a community of colleagues who follow the same goal, to elevate in the professional hierarchy (Sennett 2008). The journey began with trivial tasks, and the apprentice could gradually progress in the workshop organization to become a journeyman and eventually a master. In this context, the learning process did not rely on a pre-defined curriculum. Instead, the learners could learn by observing masters, interacting with pupils, and achieving hands-on experience while accomplishing various tasks. Direct interactions with the master, being in the physical context of the trade, and learning through time were inseparable elements of the apprenticeship model of learning. These three elements are also the pillars of situated learning theory that gained popularity in the late 1980s and 1990s through the works of Jean Lave, Etienne Wenger, John S. Brown, and Allan Collins. Jean Lave, an anthropologist at Berkley, and Etienne Wenger, a computer scientist, educational theorist, and practitioner, described the structure of a situated learning framework in their book *Situated learning: Legitimate peripheral participation* (1991). They argue that learning is a social activity deeply integrated with its social context. Situated learning suggests that learning relies on three essential elements: 1) time, 2) master-apprentice relationship, and 3) being situated in the community of practice—where the profession thrives.

They question the common belief of the superiority of verbal communication over direct demonstration (ibid, 22) and criticize the notion of knowledge decontextualization in pedagogy (ibid, 40). Lave and Wenger argue that mastery does not reside in the master itself but is embedded in a social structure that the master belongs to, referred to as the community of practice. They emphasize the importance of learning by thriving in such an environment and gradually proceeding through opportunities in the actual context of that practice.[2] From being assigned to trivial tasks to eventually being accountable for the critical ones, a journeyman builds the identity of a master.

Other scholars also profoundly contributed to the foundations of the situated theory. Notably, John Seely Brown,[3] Allan Collins, and Paul Duguid published an influential piece of literature in this field: “Situated

---

[1] Hereby I should clarify that the despite the order of these three conceptions of skill in the text, there is no intention to establish a chronological sequence among the three.

[2] A side benefit of this approach is the gradual growth in the level of responsibilities. In the early stages of learning, the learner repeatedly faces safe failures with no severe consequences. As the apprentice proceeds in the hierarchy, it shifts from the sideline to the center and gains more responsibility while constantly learning from the master and colleagues (Chernova and Thomaz 2014).

[3] John S. Brown is now a visiting scholar and advisor to the Provost at the University of Southern California. However, back in the day (1986-2002), he was the chief scientist and the director of the legendary Xerox Corporation’s Paolo Alto Research Center, better known as Xerox PARC. Paul Duguid, who is now an adjunct full professor at the School of Information at Berkeley, was also a member of the Xerox PARC family between 1989 and 2001. Allan M. Collins

Cognition and the Culture of Learning" (Brown, Collins, and Duguid 1989). The paper opens with a bold statement criticizing, by that time, the standard teaching practices that assume conceptual knowledge can be detached from the context in which it is learned, practiced, and thrived.

Throughout the paper, the authors scrutinize this notion and criticize how knowledge was understood and treated as an independent substance that can be decontextualized and transferred, advocating the notion of knowledge as situated. They emphasize the interwoven relationships between knowledge and its context. From their point of view, knowledge is a partial "product of the activity, context, and the culture in which it is developed and used" (ibid, 32), and as such, it is inseparable from the physical, personal, and social context.[4]

Other scholars assert this notion. For instance, William Clancey—by that time a member of the Institute for Research on Learning (IRL) and a computer scientist with a particular interest in cognitive science and AI[5]—argues that human thoughts and actions are both adapted to the surrounding environment because "what people *perceive*, how they *conceive* of their *activity*, and what they *physically* do develop together" (Clancey 1997, 343).[6]

These bounds between knowledge and its context result in various forms of interactions between the people, procedures, and the physical, personal, and social context. Such interactions allow each agent to evolve through time. The progressive improvement of situated knowledge is a critical factor in this discourse. Interestingly, Brown et al. tap on the concept of tool to clarify the situatedness and progressive nature of knowledge. They draw an analogy between conceptual knowledge and a set of tools, as "[t]hey can only be fully understood through use, and using them entails both changing the user's view of the world and adopting the belief system of the culture in which they are used" (Brown, Collins, and Duguid 1989). Their analogy implies that, like tools, knowledge iteratively improves in time through activities instead of being rigid or particular.[7]

---

is a professor emeritus at Northwestern University. During his academic career, he worked on psychology, AI, and education. Specifically, his work on situated learning in education is a subject of interest in this thesis.

[4] Collins, Brown, and Holum later proposed a model of instruction that they named "Cognition Apprenticeship" with roots in the apprenticeship traditions while holdings elements of schooling and could be practiced in the current educational platforms of the U.S. (A. Collins, Brown, and Holum 1991).

[5] Interestingly Xerox has yet another appearance in this section; Clancy was one of the founding members of the Institute for Research on Learning (IRL) in Menlo Park, California. The institution was a non-profit research organization which was initially supported with a grant provided by Xerox Foundation.

[6] Italics are from the source.

[7] There is a quoted paragraph, allegedly from a 1991 publication by Jean Lave, which I could not track to any of her works. The paragraph reads like this: "Lave (1991, p.84) clarifies:" Situated" does not imply that something is concrete and, or that it is not generalizable, or not imaginary. It implies that a given social practice is multiply interconnected with other aspects of ongoing social processes in activity systems at many levels of particularity and generality." This paragraph appears in the *Handbook of Educational Theories* (2012) as well as a paper titled "Situated Cognition in Theoretical and Practical Context" by Wilson and Myers, which is published as a chapter of *Theoretical Foundation of Learning Environment* (2012). The latter references the quote to page 84 of Lave's 1991 paper "Situated learning in communities of practice" and referenced as "Lave, J. (1991). Situated learning in communities of practice. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds). *Perspectives on socially shared cognition* (pp. 63-82). Washington, DC: American Psychological Association." The paper finishes on page 82, and the quote cannot be from page 84. Upon further inspection, I realized that this quote does not appear in that paper. However, a simple search on the internet shows that this exact quote has been requoted in several other places.

While situated cognition was initially conceived in the realm of education theory, it has broader implications beyond only education. It spans the social, behavioral, and neural aspects of knowledge and action (Clancey 1997). Researchers and scholars such as Phoebe Sengers inquired into the potentials of situatedness theory in AI and proposed three primary characteristics for a situated AI agent: 1) an AI agent should be evaluated with respect to its physical, personal, and social context, 2) the agent should be designed with a focus on its dynamics with the socio-physical context, and 3) it should be representative of its actual context (1996, 71).

Drawing on the ideas that I introduced above, in this dissertation, I take skill to be a form of knowledge that is best described in relation to its physical, personal, and social context. This conception of skill is in harmony with Lave and Wanger's social approach and Brown and Collins's physical point of view. It also reflects on Lucy Suchman's situated action that "comprises necessarily ad hoc responses to the actions of others and to the contingencies of particular situations" (1985) and the dynamic interactions with the social fabrics and the physical environment in which it thrives. It also resonates with the "Socially Situated AI" sentiment (Sengers 1996).[8]

8.1.2 Skill as Object/Commodity

Alongside the conception of skill as situated in its context, there are other schools of thought that treat skill as an object, a collectible substance that can be captured, transferred, or stored. From a historical point of view, one can argue that the social, industrial, and political changes of the late 18th and early 19th centuries facilitated the emergence of this conception of skill.[9,10] In this section, I will discuss how, in this context, the conception of skill as a managerial commodity and as data emerged.

Skill as Commodity

The last decade of the 19th century remarked a turning period in the social and political definition of work and skill. Jeffery Haydu, a professor of Sociology at UCS who has been focused on the historical aspects of labor and employer movements in the US, provides us with a detailed view of this transitional era through the lens of his comparative studies on the state of the US and British steel industry in the late 19th century (1988).

By that time, the apprenticeship was a de facto model for educating, regulating, and balancing the availability of the workforce in the metalworking market. This dominance granted craftsmen significant control over the workshop affairs. Between the 1890s and the 1920s, the newly emerging managerial practices challenged the craftsmen's dominance over the workshop affairs, aiming to break down works into specialized tasks and embedding skills in machines (ibid).

---

[8] In an early version of this thesis, I used "socio-materially situated secret" to refer to this conception of skill. The term "secret" reflects on the Japanese tradition of craftsmanship practice. In the Japanese craftsmanship tradition, masters were destined to guard their craft secrets even from the most gifted pupils. Apprentices must spend years observing the masters, mimicking their slightest moves, and replicating every nuance trick to gradually enrich their skillsets. Eventually, they will gain what they have invested their lives in, stealing the masters' secrets to become a master (Singleton, 1989 cited in Collins, 2010, p. 93). Additionally, I borrowed and alternated it from Jeanne Gamble, a researcher at the University of Cape Town who had been focused on vocational and professional education, describes tacit knowledge as a "manual mystery*"* while addressing the tacit aspects of craft (2001, 186). I eventually decided to drop "secret" for simplicity.

[9] To read more on the historical, political, economic, and in this specific case, military causes that initiated these new inceptions, please read (Schaffer 1994) and (Haydu 1988).

[10] It is critical to acknowledge the interactions between these two conceptions of skill, which is not the subject of this thesis and requires further discussion.

The forces behind this transition can be explained through the theories formulated by Frederick W. Taylor, an influential early 20th-century theoretician and a key figure in scientific management. He believed that "[the] management must know better than every workman in our place" (Bahnisch 2000, 62). He advocated the appropriation of workers' skills and reducing their role solely to accomplishing tasks assigned by the management. He suggested that the management must choreograph workers' actions in harmony with the manufacturing line and precise timing rather than autonomous acts (ibid, 54).

Applying this level of control was challenging as it could raise significant resistance from skilled workers who mastered their practice through years of apprenticeship. Thus, any effort to disturb this traditional order could face severe resistance. The transformation of the Portsmouth dockyard from a traditional workshop into the first Royal Navy site equipped with automatic machines between 1795 and 1807 (Schaffer 1994, 210) is an example of such conflict between skilled workers and managerial outreach. Simon Schaffer, a historian, and philosopher of science at the University of Cambridge, provides a detailed image of the Portsmouth dockyard transition by Charles Babbage.[11] The craftsmen and workers in the dockyard were protective of their skill, and resisted managers' efforts to observe and document their practice or use of tools. Subsequently, they attempted to keep their craft skills away from the managerial inspectors (Schaffer 1994, 214).

Nevertheless, workers' skills eventually have been eroded in favor of Taylor's scientific management practices. In the process of "deskilling," procedures that demanded skilled workers were replaced by pieces of machinery or menial tasks that could be operated or executed by less-skilled workers, with lower wages and shorter training periods. Taylor's approach resulted in detaching the conception of work from its execution, eliminating the cognitive and social tissues that were tying them together, eventually "dislocate[ing] the crafts skills from its original collective social base" (Gamble 2001, 190). Workers became alienated from their work, feeling powerless in the workspace, believing their own decisions and behavior will not determine the outcomes that they are looking for (Seeman 1959). The master-apprentice relationship and the hierarchy of "master, journeyman, apprentice" were replaced by "skilled, semi-skilled, unskilled" demarcation (Gamble 2001, 190).

Taylor's work was not the first effort to dethrone the craftsmen's control over the workshop floors. It was preluded by a century of preparation. Prior to Taylor, other scholars, economists, and philosophers have eroded its pillars. In 1776 Adam Smith, the famous Scottish economist, suggested the concept of division of labor as a critical factor in growth (1776). In *The Wealth of Nations*, he emphasizes the importance of breaking down works into smaller components and training workers to become experts in an isolated and specific task to improve their efficiency. Building on those ideas, Babbage proposed a set of principles that suggests breaking down work into separate procedures that each requires a different skill level, then hiring workers with a matching level of skill that is required for each task. This approach helped managers to *"*purchase exactly that precise quantity of [skilled and novice workers] which is necessary for each process*"* (Babbage 1835, p.175 cited in Schaffer, 1994, p. 209).

These efforts redefined the skilled labor market, where skill was attainable by managers at the precise amount needed at the moment (Schaffer 1994). It was a historical transition that also facilitated the transfer of production control from skilled workers to managers. By the 20th century in the United States, the craft was pushed to the side as a "secondary means of organizing work" (Barely and Orr 1997, 2) and was substituted by technical work, and technicians. Technical work entails a new type of formal

[11] Charles Babbage is the British mathematician, philosopher, and engineer who is also the designer of one of the earliest known instances of automatic computing engines.

education, reducing the role of long-term practice and distances from the traditional apprenticeship model (1997, 4–11).

Skill as Data

The conception of skill as a commodity, which I discussed above, implies that skill can be traded as an object, separated from its context, tools, and the skilled workers who are practicing it. This assumption is a cornerstone of the data-driven approach to skill. One of the earliest tracks of the conception of skill as data can be traced back to the early years of the 20th century. In 1907, in his influential work, "Lecture on Management," Fredrick Taylor proposed that every action and movement of workers could be captured as data for scientific management studies (Bahnisch 2000, 62). He proceeded to distinguish the concept of work from its execution, facilitating the systematical analysis of workers' actions as a prelude to the scientific process of finding the best way of performing a task, or in today's terms, optimization.

Taylor was one of the first authors to adopt the word 'data' in its current meaning. He repeatedly uses "data" throughout his 1911 book *Principles of scientific management* and utilizes it as a possible means to depict a precise image of "… what really constitutes a proper day's work for a workman.*"* From Taylor's point of view, such a data-driven image could help to tune harmonious cooperation between managers and workers (Taylor 1911, 42–43).[12]

The underlying assumption behind Taylor's proposition was that human skill could be reduced into abstract data points, which then can be acquired, contained in databases, and eventually transferred. This conception of skill as data resonates with the knowledge decontextualization that I previously discussed.

The last decade of the 20th century became the most thriving era of Taylor's point of view on skill, when it resurfaced as a cornerstone of artificial intelligence efforts and configured a Taylorist approach to AI.[13] Phoebe Sengers, a professor of Information Science and Science & Technology Studies at Cornell University, asserts that in the 1990s, AI was a reincarnation of Taylorism's human engineering and control practices (1998, 62). What helped this process was the leap in computing infrastructure of the late 20th century which made it possible to process large datasets and facilitate the conception of "information as an economic good" (Barely and Orr 1997, 9).[14] The dataset "owner*"* claims the ownership of skill and trades it at the desired cost. If in skill-as-commodity conception, an organization could acquire skills through human resource efforts, this conception offers raw datasets or off-the-shelf programs for quick implementation.

Fueled by optimism, some AI proponents were promising human-level intelligence "embedded" in computer programs. Several scholars reported on this promise. For example, Diana Forsyth—an anthropologist and science and technology studies (STS) scholar who devoted her short scholarly career to the domain of artificial intelligence and informatics—[15] observes that AI visionaries believe

---

[12] Taylor worked in the Midvales Steel Company for several years, started as a machine shop worker in 1870s and gradually proceeded to the top. He persuaded the owner of the company to fund a study about the time required to perform various tasks by the workers. It is interesting to know that according to Taylor, the owner was not convinced that the scientific study that he authorized would eventually return any valuable results, nevertheless he proceeded to support it.

[13] Asada and Liu cited several efforts has been focused on measuring and quantifying human dexterity and skill in working with tools as early as the 1970s (1991).

[14] Interestingly, "information as an economic good" makes a very well-suited segue between skill-as-commodity and skill-as-data conceptions.

[15] After her unfortunate death in an accident during a hiking trip in 1997, her unfinished works and essays were published in 2001 book, *Studying those who study us* by her colleagues.

"…computers will increasingly be able to duplicate human expertise"(2001, 35). It was generally accepted among the AI advocates that by collecting millions of data points about an object, we could even address the common sense knowledge problem (Dreyfus 2007)[16] or derive models to "elucidate" human skill and human actions from these datasets and transfer them to machines using artificial neural networks (Asada and Liu 1991, 2442–43). While Taylor's focus was centralized around optimizing the human labor force, Asada's work aimed to treat skill as a transferable substance from humans to machines.

The latest bloom of machine learning (ML) in the early 2010s, advancements in computer processing hardware, and the availability of large datasets resulted in several efforts to extract this new type of goods. Researchers have leveraged different methods to "acquire" this commodity by processing the available datasets, i.e., medical records (Esteva et al. 2017), conducting a large number of virtual simulations or real experiences (Levine et al. 2018), direct demonstrations by a human "teacher" (Argall et al. 2009).

Treating skill as an object is a common practice in the current state of the AI/ML economy. Here it is enlightening to review some examples of this phenomenon. Amazon.com, the e-commerce, cloud computing, and AI giant, offers two interesting services Mechanical Turk and Rekognition. Amazon Mechanical Turk (MTurk), is described as a "crowdsourcing marketplace" where employers can outsource tasks to "distributed workforce"s (Amazon.com n.d.). This closely resembles the conception of skill as a managerial asset, where managers can hire, and fire, the workforce on-demand for atomic tasks without the challenges of working with the craftsmen.

While MTurk is designed to hire workers, Rekognition is advertised as a quickly deployable image recognition service promising a substitute for skilled users in tasks that require visual inspection. As per Amazon's account, clients "… pay only for what [they] use … [Amazon] charges [them] only for the images processed, minutes of video processed, and faces stored" (Amazon.com n.d.). The same applies to a myriad of AI/ML services that take over tasks that were once dominated by skilled workers.

Comparing these two services crystalizes a key difference between the two forms of skill as object conception. In MTurk, which reflects the conception of skill as commodity, we observe the detachment of workers from the conception and understanding of their work. The hired workers are not completely aware of their employer or the broader goals of the task they are conducting. In Rekognition we see the tendency to detach the workforce from the execution of work and skill is presented as data embedded in computer vision algorithms.

8.1.3 Discussion on the Contrasts between the Two Conceptions of Skill

Reducing skill into abstracted data is an appealing concept in the realm of skill transfer and discourses on codifying skills into machines. However, this conception has several flaws that cannot be overlooked, specifically when it comes to embracing the context. First, I argue that the skill as object conception requires a perfect vehicle to encapsulate and transfer skill. Through the lens of Reddy's Conduit Metaphor and Toolmaker Paradigm, I reason that such a vehicle does not exist, and skill is woven into its physical, personal, and social context. Then I will explain how the skill as object conception is incapable of addressing the physical, personal, and social context, nor the tacit aspects of skill.

[16] Common sense problem refers to the challenge of recognizing and embedding facts, details, and underlying assumptions that a typical user is expected to know. While trivial for a human, it has been a hard challenge for AI experts.

The Impossible Perfect Vehicle

Thinking of knowledge as a transferable object inevitably implies that there exists a container, or vehicle, to capture and codify skill and effortlessly transfer it between individuals, or machines. This closely reminds us of Michael Reddy's Conduit Metaphor (1979) where he explains how linguistic structures trick us into making the same assumption about thoughts, as objects that can be captured, contained in linguistic expression, and then transferred through communication. In both skill as a commodity and skill as data, a similar mindset is prevalent (Table 9).

Table 9. Mapping Conduit Metaphor on skill as a commodity and skill as data

	Conduit Metaphor	*Skill as a Commodity*	*Skill as Data*
Source	Author	Craftsman	Craftsman
Object	Thoughts	Skill	Skill
Vehicle	Linguistic expressions	Managerial Methods	Data
Transfer	Communication	Mechanization	Programming
Target	Audience	Mechanical machines/Workers	Computers

Reddy, already aware of these mind tricks and their consequences, introduces the toolmaker paradigm. He argues that one's thoughts are isolated from the others and informed by the context it has flourished in. Thus, communication between individuals is not an effortless task as it is assumed in the conduit metaphor, but it entails an imperfect method of communication that urges constant efforts by both sides to overcome its shortcomings. Thus, any communication is prone to interpretation and subjective assumptions, which tie messages into each person's thoughts and the context.

Although Reddy does not directly address human-machine communication or AI in general, the toolmaker paradigm helps expose some of the shortcomings of efforts to codify human skills into the machine's language and mapping of skills between humans and machines. Following his logic, we can extrapolate the limitations of conduit metaphor to the conception of skill as object. First, the context plays a significant role in the process, being physical elements, social factors, or personal preferences in the conception of skill as data, context is reduced and abstracted, as I will discuss in the next section. Moreover, when it comes to transferring skills to machines, it is essential to consider the fact that humans and machines are inherently different, and they do not share the same physical context or intellectual characteristics (if we consider any intellectual power for machines). Finally, there is no perfect communication method, not between humans and machines nor among the machines. Thus, it is virtually impossible to directly map human capabilities into machine tasks due to such differences.

Representation of physical, Personal, and Social Context

The other prevalent issue of the conception of skill as an object stems from the representation of the context. Efforts to capture and codify knowledge require methods to reduce the complex context into a finite set of variables. We can observe this phenomenon in the AI discourses of the late 20th century, where some researchers believed that representation is the key to a successful AI. However, to achieve such representation, AI researchers needed to implement an abstraction schema, factoring out all aspects

they determined as non-critical and only including the relevant details to simplify the problems.[17] This form of abstraction is the cornerstone of many AI studies.

However, abstraction proved to be a problematic issue. An abstract representation of the context tends to factor out the dynamic coupling between the system and its world and eliminates various aspects of a creative practitioner's perception and motor skills. Thus, it is not as valuable as constantly referring to the real world and the immediate context. [18]

The downside of abstraction in representation is reflected in the efforts to extract knowledge from domain experts to build expert systems in the 1990s. Forsyth explains how AI researchers' were occupied with achieving this goal by leveraging knowledge acquisition methods— i.e., surveys, interviews, and observations (2001). Throughout this process, she observed a strong tendency to detach the expert users from the process. A consequence of this approach is the ignorance of expert users' personal inputs, social ties among the experts, as well as information about the physical context of their practice.[19, 20]

Tacit Knowledge

Another shortcoming of treating knowledge as an object arises when addressing the tacit aspects of knowledge.[21] Forsyth and Collins have observed and documented several instances of these issues, the efforts that shared the skill as an object conception.

Michael Polanyi, the Hungarian-British chemist, who later became a professor of Social Sciences at the University of Manchester and an influential figure in social science and philosophy, envisions tacit knowledge "… as a way to know more than we can tell" (Polanyi 1966b, 17–18). He elaborates further, stating that "… tacit knowledge can be discovered, without our being able to identify what it is that we

---

[17] At a closer look, it becomes clear that abstraction is a key concept behind the conduit metaphor, as it treats knowledge as a detached object from its surrounding context that can be transmitted from one point to another without any loss.

[18] I borrowed this idea from Rodney Brooks and his influential paper "Intelligence without Representation" (1991b). Brooks, an Australian roboticist and scholar, who is best-known for his years at MIT as the head of CSAIL, explains situatedness as a central idea for behavior-base robotic systems: "The robots are situated in the world, they do not deal with abstract description, but with the "here" and "now" of the environment that directly influences the behavior of the system" (Brooks 1991a, 1227). Although Brooks in this paper specifically discusses the robotic systems, his reasoning is also valid in the context of AI and toolmaking. Outside the academic world, we might be familiar with Brooks through his contributions to the consumer-grade robots. He is one of the three co-founders of iRobot, the company that, among a myriad of defense contracts, popularized robotic home vacuum cleaners by introducing Roomba. Needless to say, these vacuum robots constantly use their sensors to sense the environment and react to the context they work in.

[19] From a broader point of view, this problem is another consequence of ignoring the tacit aspect of knowledge in the 1990s era AI culture.

[20] Even more recent efforts, for example, ML models, which are usually trained on large datasets that have been collected and prepared through crowdsourcing, are prone to biased dataset flaws. Several researchers demonstrated that such datasets could be pre-loaded with assumptions and convey social and contextual biases, which will eventually be encoded to the trained models.

For example, Zhao et al. demonstrated that two popular visual semantic role labeling datasets contain significant gender bias. They highlight that "… the activity cooking is over 33% more likely to involve females than males in a training set, and a trained model further amplifies the disparity to 68% at test time"(Zhao et al. 2017). Similarly, some of the popular datasets for human face detection studies are heavily weighted on lighter-skinned subjects. They also evaluated three commercial systems and observed that they perform significantly better when the test sample is a lighter-skin male vs. a darker-skinned female (Buolamwini and Gebru 2018). When creative practitioners repurpose these tools in their work, these issues bleed into their creative process and intensify the situatedness gap.

[21] The Latin origin of "tacit" means "silent" and "unspoken. "Tacit" also means "secret." In "tacit knowledge," it is bears both meaning (Csikszentmihalyi 2002).

have come to know" (Polanyi 1966a, 5). One may possess tacit knowledge, which can be crystallized as a particular skill, but not be able to easily express or codify it.

Polanyi describes tacit knowledge by describing its distance from explicit knowledge: "… while tacit knowledge can be possessed by itself, explicit knowledge must rely on being tacitly understood and applied" (ibid, 7). A classic example of tacit knowledge is bicycle riding. One can possess this skill, but it is not easy to teach another person to do so just through verbal clues or even identify the steps to maintain balance while riding. Polanyi associates this with the fact that we are "… only subsidiarily aware of these things," which might not be enough to make the thing identifiable (ibid, 5).

Tacit knowledge plays an essential role in the master-apprentice pedagogical model and is one of the factors that urges a newcomer to spend years forging their own identity as a master. Harry Collins, a sociologist of science, has observed and documented numerous cases of tacit knowledge in different fields. A now-classic example of his studies is *The TEA Set: Tacit knowledge and scientific network* (1974), where he emphasizes the importance of tacit knowledge, even in cutting-edge scientific efforts. He elaborates an example of knowledge diffusion in a network of experts trying to reproduce a specific laser system. The experts were trying to replicate a series of experiments first successfully conducted in another research lab. Collins reports that only those who relied on direct human-to-human communications could eventually accomplish the task, while those who solely focused on formal written mediums, i.e., peer-reviewed publications, failed to achieve the same goal.[22]

Collins explains that tacit knowledge cannot be formally instructed and cannot be achieved through practice. The learner should gain it through demonstration, guided instruction, personal contact, and socialization with others who master it (ibid, 99). Accordingly, skill is not merely learned or transferred, but it is reconstructed by the learner, situated in a specific context. Tacit knowledge and the skills it entails may vanish if the face-to-face master-apprentice relationship fails, even for a brief period (Polanyi 2005, 55).

There are several factors that give tacit knowledge such a unique characteristic.[23] While some of these factors are inherent properties of knowledge, some are factors derived from the context, individuals, and relationships among them. For instance, individuals who master a domain of knowledge might not be aware of its importance or even its existence and, accordingly, cannot express or describe it. This phenomenon is known as unrecognized knowledge. [24]

Another complex yet interesting aspect of tacit knowledge is ostensive knowledge. It refers to the knowledge which is significantly easier to obtain through observation of an apparatus or practice rather than verbal communication or scripted guides. In such cases, the spoken explanation would be too

[22] It is worth mentioning that from the perspective of skill as a commodity, tacit knowledge is very well recognized, but it is treated as a commodity that is carried by individuals. This is best reflected in Collin's brief discussion on how experts in economics and knowledge management practices approach tacit knowledge. In this field, tacit knowledge is considered as an asset that can be acquired by hiring people who possess that knowledge or even acquiring the whole business in which it has been developed (H. Collins 2010, 3).

[23] For a comprehensive discussion on this topic, please refer to (H. Collins 2010).

[24] It also contributes to the importance of the time factor and mimicking in the master-apprentice learning process. Through time, an apprentice starts to copy every delicate detail of the master's action. Some of these unrecognized skills can evolve into unexplained rituals; practitioners are committed to observing them without being aware of their importance.

complex compared to watching the performance. Riding a bicycle is a classic example of ostensive knowledge: one will find it impossible to explain in words but easy to demonstrate.

Finally, one who masters a skill may assume that the audience also has a specific set of understanding, assumptions, and skills, which they may not have. In such cases, the two agents' isolated minds do not share the same context and common ground, and the communication methods can not effectively convey the message. This phenomenon is known as mismatched salience.[25]

These characteristics explain why learning tacit knowledge requires a prolonged period of time and living in the community of practice. It also signifies why conventional data collection methods, i.e., surveys and interviews, are prone to overlooking such nuances of tacit knowledge. These methods are incapable of recording unrecognized skills since the subject is unaware of their existence, or cannot verbally express them, or the audience does not have the common ground to understand them.

Interestingly, these characteristics also allow tacit aspects of knowledge to reside in any artifact made by human agents, even when the goal is to intentionally wipe them out. Thanks to Forsyth's work, we have well-documented examples of intentional efforts to eliminate the tacit aspects of knowledge in the 1990s' AI culture (2001). Through the lens of her work, we can observe how AI researchers tried to decouple the social and cultural aspects of knowledge from the expert system that they were trying to develop. Forsyth indicates their promise was not credible, and the resulting expert systems were not value-free, as they incorporated researchers' tacit assumptions about the nature of knowledge and work (ibid).

8.1.4 Discussion

Conceptualizing skill as an object, or as a commodity, can lead to problems. Notably, it can mislead us to overlook the contextual and tacit aspects of skill. Through the lens of Polanyi's work, we observe that even recognizing nuance and intangible aspects of tacit knowledge is a challenging task, let alone capturing, storing, and transferring them. The conception of skill as an object is fundamentally built around decontextualization and abstraction from the context. It assumes that knowledge is transferable from people and the physical, personal, and social context in which it thrives, into a machine. However, the contextual nature of knowledge ties it to the characteristics of its context and cannot be reproduced in a machine. Even though one can try to build a system based the conception of skill as an object, the outcomes will be the bearer of tacit knowledge of people who have contributed to the development of its components.

A review of literature on the intersection of design and machine learning[26] suggests that, with a few exceptions, most of these efforts ignored the contextual relationships between toolmakers, tools, and the creative task they are addressing. Addressing this issue requires a different perspective on AI and ML for toolmaking for creative practitioners. In this thesis, I suggest methods of toolmaking that embrace knowledge as it is situated in the physical, personal, and social context. In Chapter 3, The Framework, I discussed this approach in detail.

I would like to close this section with reference to Donald A. Schön, American philosopher and professor of urban planning at MIT. He opens his article, *Designing as reflective conversation with the materials of a design situation*, with a few propositions which help us understand the state of research on AI and

---

[25] The three characteristics that are mentioned here, unrecognized knowledge, ostensive knowledge, and mismatched saliences are only a few examples to demonstrate the complexity of tacit knowledge. A keen reader can refer to (H. Collins 2010) for further discussion on this topic.

[26] For this review, please refer to chapter 2, ML-Based Toolmaking for Creative Practitioners.

design with respect to the tacit aspects of knowledge at that time. The first proposition is that design research, from an AI point of view, “is an attempt to capture design knowledge by embodying it in procedures expressable in a computer program” (1992, 131), which reminds us of the conception of skill as an object. However, in the second proposition, Schön states that design knowledge is mostly tacit, and it is expressed in and by actual design, a hint to the situatedness of design knowledge in its context. Schön points out a few characteristics of tacit knowledge that reflect in designers, i.e., they know more than what they can tell, they cannot accurately describe what they know, and they can best represent their knowledge when they put it into practice.

In the third proposition, Schön argues that the efforts to embed design in symbolic and procedural representations are incomplete and inadequate. His opening ends with a critical question: is it even possible to use AI and ML to create tools for creative practitioners, and in this case, designers? Schön argues that it depends on the purpose of such tool. Creating a tool to generate design output without specifically presenting the solution and replacing a human designer with a “functional equivalent” might be an extremely hard problem to tackle. But making tools that assist and facilitate these creative practitioners in their workflow is an achievable objective.

Appendix III: Supporting Documents

Supporting materials for the case studies.

9.1 SecondHand Study Reflection Papers

Participant #2

The data collection process:

I think the data collection process for interactive machine learning is based on user's input. When I was writing the letters, sometimes I would erase the letters that I thought not good as an input for the dataset I would build. By doing so, I could control the quality of the dataset I was going to build and I would be very familiar with the dataset. Compared with the off-the-shelf approach, I think I would not be that familiar with the dataset from the off-the-shelf approach and I may concern about its quality. But I think the off-the-shelf approach saves user's time of collecting data and it can provide more diverse data to cover various situations instead of the data only collected by a user. I think if the idea is to create a unique or small dataset, I would prefer to collect data by myself, otherwise, I think I will use the off-the-shelf approach.

Your experience with the training process through providing various sets of data instead of only changing the architecture and hyperparameters:

I trained several sets of data and created according fonts. My first font is from my first data set which has a thick line weight. As a result, I got a font with thick line weight. Then I was thinking what if I create a set of data with light line weight and use it to train the model? As expected, I got a font with light line weight. Compared with the approach that only changes the architecture and hyperparameters, I think providing various sets of data can have total different outcomes. The outcomes may not be very different when using a single data set and changing the architecture and hyperparameters. I think if I am happy with the training result and only wants to improve it a little bit, I will change the architecture and hyperparameters because it is very close to the result I want. If I think the training result is way much away from what I expected, I will change the data set because it is easier to get the result I want.

Your experience with the navigation of the latent space using interactive widgets in the CoLab notebook:

I think it is easier to know what is the result of changing the number of mean/std value. The interactive widgets provide a visual representation of the result which helps me to identify the whether I accept the result of not. It is very easy to use and can show the result quickly. If the notebook does not have the widgets, I think it is really hard to tell whether I can accept the result and what is the difference when I slightly change the mean/std value. But I also find that when I change the mean/std value back to the previous one, the result is different.

Your experience with the data viewer/selection dashboard:

I am interested in the right part of the dash data view. I think the left one also provides many meaningful information, the right one clearly shows me how the letter changes when I sweep the mouse from left to right. But sometimes I will find another letter presents in the letter that I am focusing on. I am looking at the letter of 'c', but some 'b's also present (shown in the following image). When sweep the mouse from bottom to top, I think the letters in the path share some similarities which I find quite interesting because I do not think about that when I was writing the letters.

How was your experience with using data as a method of controlling an ML model compared with your experience of using code to modify an ML model?

I think the data collecting process is very important when using data as a method of controlling an ML model. Using code to modify an ML model may be seen as a craftsmanship, user has to change the values many many times to get the expected result and even cannot get the result if the data is not ideal. Using data as a method of controlling an ML model can quickly get the needed result (coarse grain), but in order to get a better result (fine grain), user has to either improve the quality of the data of modify the architecture/hyperparameters of the model.

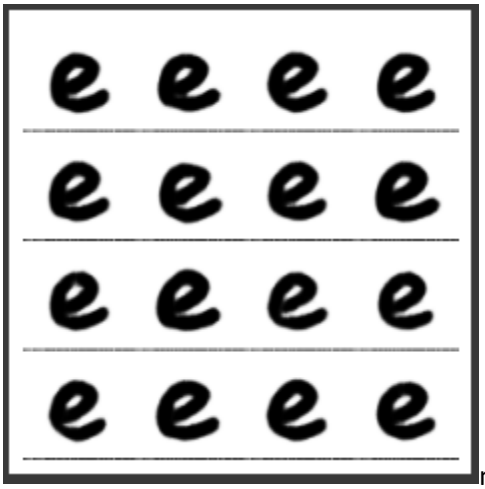
The overall process:

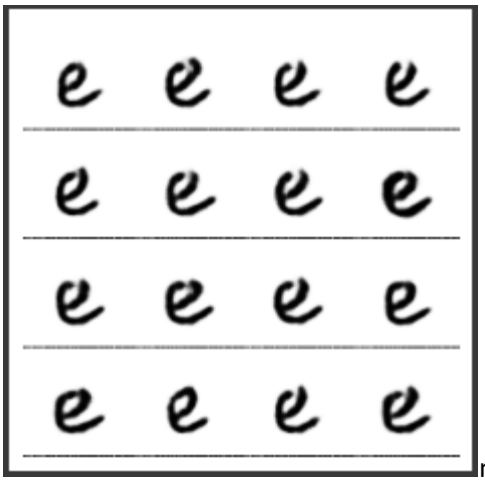
High mean value works better sometimes, low mean value does not work sometimes.

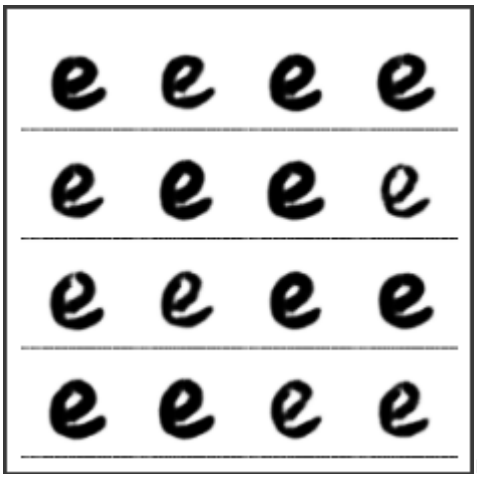
Adjust mean value affects greatly of the result, adjust std value changes slightly of the result.

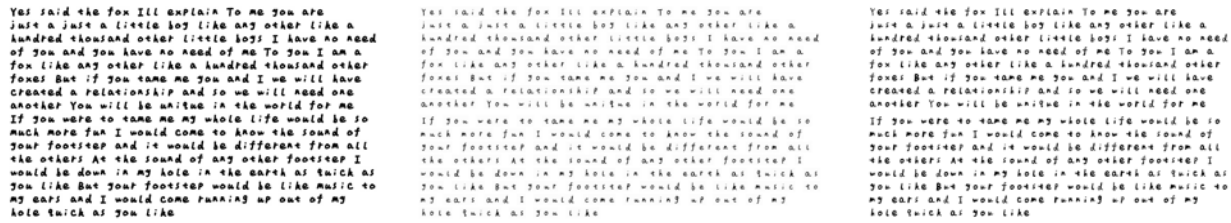
The same mean and std value will have different results when changing them to a value first and changing them back to the previous value.

The third data set I use is the combination of the my first (thick line weight) and second (light line weight) data sets. I was expecting the result would have a median line weight, but actually not (shown in the following images).


mean 0.18, std 0.20


mean 0.20, std 0.20


mean 0.19, std 0.20



The final results of the fonts I generated.

It is very interesting to train the model and generate my own font.

Participant #3

DATA COLLECTION PROCESS

Manual vs. Off-The-Shelf

Manually creating data was a very successful and fun process overall. The act of writing out each letter by hand slowed the data collection down and required physical labour, pain and sweaty hands to complete. Though somewhat arduous, a manual collection process allows you to physically understand the amount of data required for machine learning algorithms require for learning, and you also become very cognisant of the type and consistency of training material you need to provide. Writing out letters allows you to do this as each pen stroke is signifier and representation of data being fed into the model. This notion heightened my awareness and made me feel much more connected to the process, and in a sense, highly engrained within a human-computer system, something which I've never experienced before. I believe this physical - computational connected really changes your view and overall impression of AI / ML systems and makes things which may have seemed quite abstract much more tangible and "real". As a result, it becomes much easier to imagine how other "real world" things (singing, painting, moving, etc) can fluidly transport from the "real" world into the artificial through its breakdown through quantitative & computational digestion as conveyed through this project.

TRAINING PROCESS EXPERIENCE

Multiple Training Datasets

Training the model multiple times with multiple types and sizes of datasets allows a user to really understand the value of high quality, consistent and large datasets and its direct impact on the quality of synthesized output imagery, or in this case text. Even though the first dataset took quite some time to create by hand (32 minutes), it was clearly not robust enough to result in a well trained model. Perhaps if I were to have trained the model for additional iterations, my results may have improved. Nonetheless, the second training set with 3 times the amount of data ensured me that dataset quantity rather than an increase of epoch can have a quite positive effect on output quality instead. The third and final training dataset used was the class dataset which contained 15x more data than the 1st set, and approx. 5x more data than the second set resulted in arguablly the best results in terms of consistent legibility. However, I do have to say that this improvement may simply be due to the fact that my own lowercase handwriting is not particularily legible. That being said, the third dataset did not appear to be 6x better than the second. This may have been due to the disparity of styles included in the set as it was made up of approx. 10 student's handwriting. The algorithm may have had difficulty defining a consistent pattern. Nevertheless, training the model with multiple datasets was an extremely beneficial process as it helped me understand the importance of volume and consistency.

LATENT SPACE NAVIGATION

Pros and Cons

Latent space navigation was unforately fairly unintuitive and hard to understand even after extended use. Though the "mean" and "standard deviation" are terms I understand and can imagine, the navigation GUI could have benefited from a graphic representation of the current state of the normal curve as effected by user changes. This normal curve could change with user manipulation of the sliders and help visualize what is being done and how the curve and samples taken from it are being effected. Finally, the navigation GUI could have benefited from a brief explanation of what each control does and how exactly it effects the synthesized output letter. Again, even after much use and playing around, I still did not generally understand how the outputs were effected to the accuracy that I had hoped. As a result, latent space navigation was not as smooth as I had hoped. Finally, the character selector was overly sensitive and made it hard to change from letter to letter. For example, if I wanted to switch from "a" to "b", I would typically end up going to "c" or "d" by mistake. This made naviation quite difficult.

DATA VIEWER/SELECTION DASHBOARD

My Experience

The data viewer and dashboard were great tools to help visualize the datasets and understand the variety of samples within. It was also a good tool to get a quick overview of your datasets after processing. Selecting particular data to sample from was also easy and straight forward. However, it did take some time to understand how to interact with the dashboard and what the various buttons did. I'd suggest adding in a brief explanation above. Finally, I would recommend making this dashboard larger as I found everything too small given the amount of visible datapoints. As said on the Miro dashboard, a 3d rotation function would be great.

EXPERIENCE THROUGH DATA

Data vs. Code?

Using data as a method of controlling an ML model compared with using code to modify an ML model is an extremely beneficial way to work with ML / Generative models for both novices and experts alike. As I've been working with ML models for nearly two years, I've never had the opportunity to have such a close relationship with dataset before. The experience of manipulating models with data is a great way to understand the importance of a well curated and robust dataset. Though coding can improve models in their own way, it is widely agreed that the dataset is often the most important and significant contributor to a successful model. This fact is reinforced and "lived" through this kind of "data-first" project, so I really commend the end user-impression and respect I've gained through this learning experience.

OVERALL PROCESS

Pros and Cons

Reflecting on this project was a great opportunity to stop and really think about the values I've gained from the experience. Though its easy to say "oh, this is easy, I didn't even have to write much code to make it work!", it is clear that this exercise provided something much more valuable than learning a couple numpy tricks or some cool new neural net architecture. In the end, it was a fantastic way to build a real human-AI relationship through hands-on data creation, physical-to-virtual data conversion and training. As previously mentioned, this intimate relationship between physical and virtual through dataset creation recognizes the importance that AI and ML is not just a computational process that exists virtually, but is rather a tool that can interact with real things, connect to real people, and have real effects on our physical world. In a sense, it links the real world to the virtual, which is a thing that typical ML / AI projects rarely do. That being said, I'd make hand-written datasets a requirement as I believe that "physical-virtual" relationship is the most important aspect of this project. After showing my wife the model in action creating writing in my own hand-written style, she immediately wanted me to make a dataset of hers! We then went on and discussed created datasets for other things like her painting style, crocheting style, and so on. I don't think the same discussion would have arisen if I had created a dataset on my computer and not by hand.

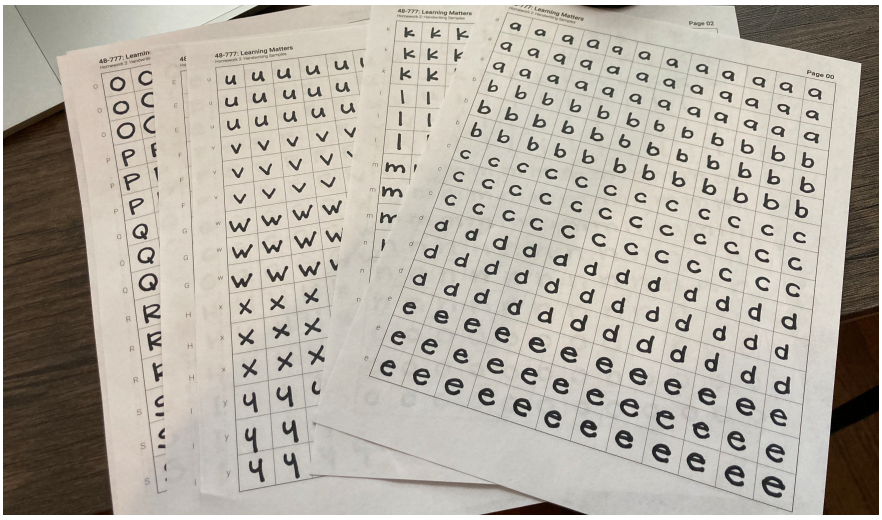
Participant #5

IDENTIFIABLE DATA REMOVED

Interactive ML HW Reflection

Data collection

This data collection process was really interesting and enlightening for me, because I wrote all of the letters by hand twice and it made me realize how convenient is to have digital data instead. Yet, there was some magic of doing the handwriting on paper, specially when I saw how the digital versions of the other students worked out. Understanding that the average of the pixels covered and learned by the model were working better on bold characters was something I realized after I trained the class dataset. So, I'm glad I took a more conservative approach and that I wrote everything with a sharpie, it turned out looking great.



Additionally the first time I trained the model with my first dataset, it was a little bit squished and almost spray-looking. I thought it was because the characters weren't cleaned up correctly from the data pre-processing notebook, and because the letters weren't all aligned in the center.

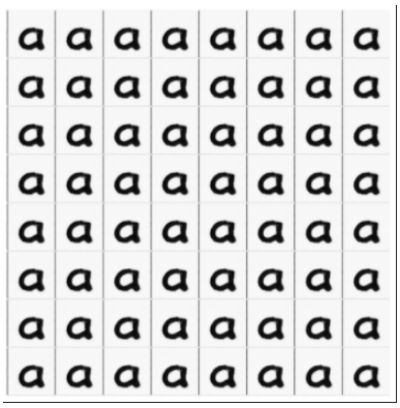
So I corrected that in my second batch of data collection, which was more uniform and centered, which made the pre-processing easier and allowed for clean cut characters. In the end, I believe the second batch was better and I did it based on the experience from the first one.

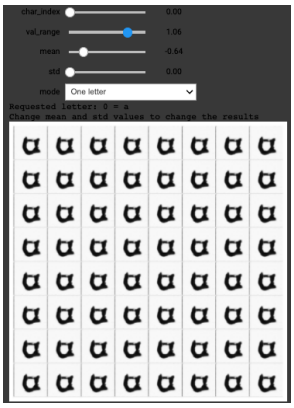
For the third dataset from the class, I purposely picked one that looked different to mine, just because I wanted to train the model with a thinner handwriting, and curiously I worked terribly.

As I described it, it looks like a right-handed trying to write with their left hand, yet it was fascinating to see this result compared to my earlier tries. Made me wonder if others had the same result. But yeah, the best training I saw was on my first 2 fonts, instead of the last one from the class dataset, which made think about how this is still just picking up pixels, and making a mean out of it, so the less pixels you give per character, the less it can work with, making it harder to learn.

Navigation of latent space

When I first trained the model, I only looked at the alphabet image, instead of the character mode where the sliders of latent space could be worked. But after the presentation I explored some of the possible options in my font, the one that had more difference between the mean slider was the first font (the grungy one), because as Ardavan mentioned in the presentation… my second font was so consistent that there was no room for change, but the first one showed really interesting results.





The original character is the one on the far left, and when the mean slide is in -1, it gave the “a” character a gothic look, with the stretched pointy edges. And for the other side of the spectrum, the “a” was just a little bit squished down but still looking as an “a” character, which is something that is difficult to spot from the lower mean value. I believe, the greater the mean is, it allows for normal transitions of styles without distorting the letter much.

Selection Dashboard

For the selection dashboard, I didn't have any problem installing the selection app in the terminal. And something I highly appreciated was that the selection already came all cleaned up and there was no pre-processing required.

It was super easy to use and work with, I really liked this selection tool, I loved how it previewed the selection and the data points. It was an amazing interface.

Data as ML model control instead of code

I think what I saw different from the last hw, was the variety of outputs you could have, since everyone's handwriting is different, there is a ton of possibilities of font mix n'match, which makes it fascinating, but still you need to have people willing to put the effort to build a dataset of 18K characters. So, I did like this exploration of working dataset instead of working the model's code to get different results, it makes it easy to control and improve it if necessary.

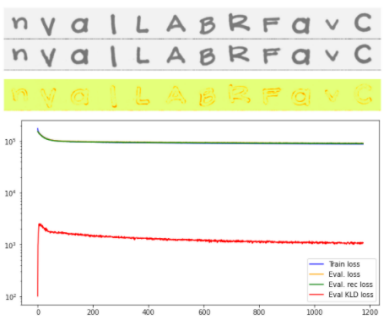
Overall process

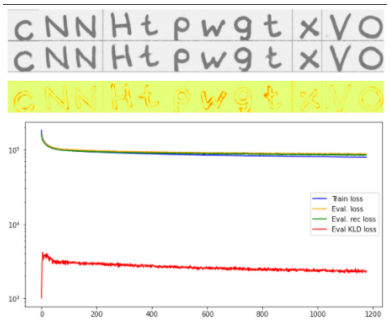
Overall, I learned a ton from this process, it was really interesting to work through all the ML pipeline, from the data collection to touching up the results. Something I found really funny, was during the data pre-processing where I didn't changed the row and col steps based on my sheets width and height, so when I ran that notebook, I got this beautiful image.

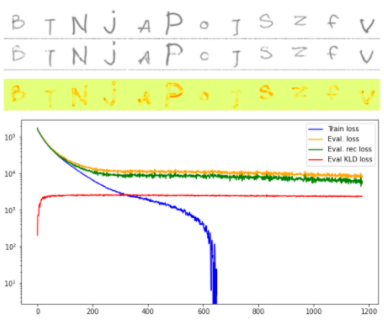
It looks like something that needs to be deciphered, I loved it, and found it really funny.

Furthermore, I was really amazed by the results of the other students during their presentation, at some point I thought why did they go for less than the default 500 training epochs. I mean, most of them were working with <400, which made me think that maybe I was doing it wrong but after their results, I just realized that wasn't it. So I showed my work, which apparently everyone enjoyed.

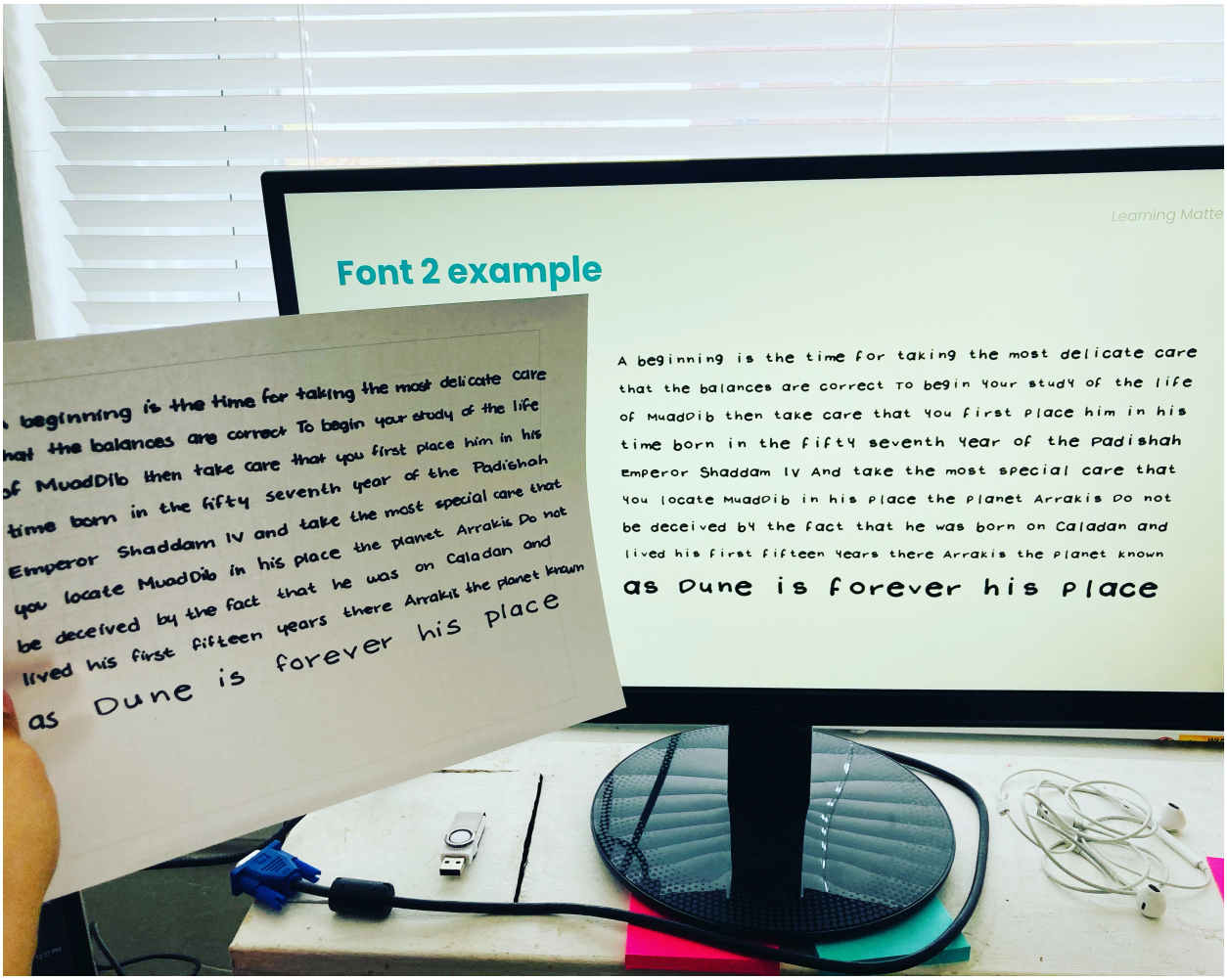
Something interesting was how the training loss of the third font dropped (as shown in the image below) after the 600 epoch, almost like it was not converging, but I realized it wasn't only my problem, because many of the other students shared the same issue.







In the end, I really enjoyed this assignment and I was amazed by the results and accuracy of the created font which does looks a lot like mine. So thanks for this.



Participant #7

ASSIGNMENT 2 | THE SECOND HAND

OVERVIEW

This assignment focuses on practicing the basics of situated machine learning and interactive machine learning. It uses a conditional variational autoencoder (CVAE) to generate a handwriting generator, based on the handwriting data created by users. The main idea is to create a data pipeline to manipulate the input data to train the model.

DATA COLLECTION

The provided worksheets were to be populated with our one's own handwriting samples. The handwriting data was generated using an iPad Pro 3rd Generation and Apple Pencil 1st Generation in the Goodnotes App. To process this data further, Adobe Photoshop was used for cropping and resizing. This creates the necessary raw data required for the model.

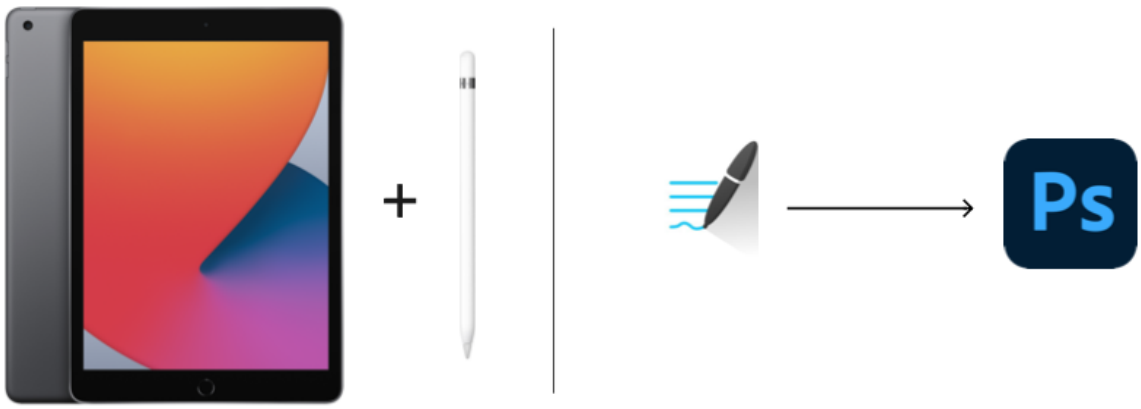


Figure 1. Left: Hardware used - Ipad, Apple Pencil; Right: Applications used - GoodNotes, Photoshop

Data preprocessing involves cropping the entire alphabet into labeled data wherein every letter is cropped and labeled with its corresponding textual alphabet. In this entire project - three kinds of datasets are used: 01 Created by me, 02 Created by me, 03 Created by Entire class, 04 Combination of datasets 01 and 02.

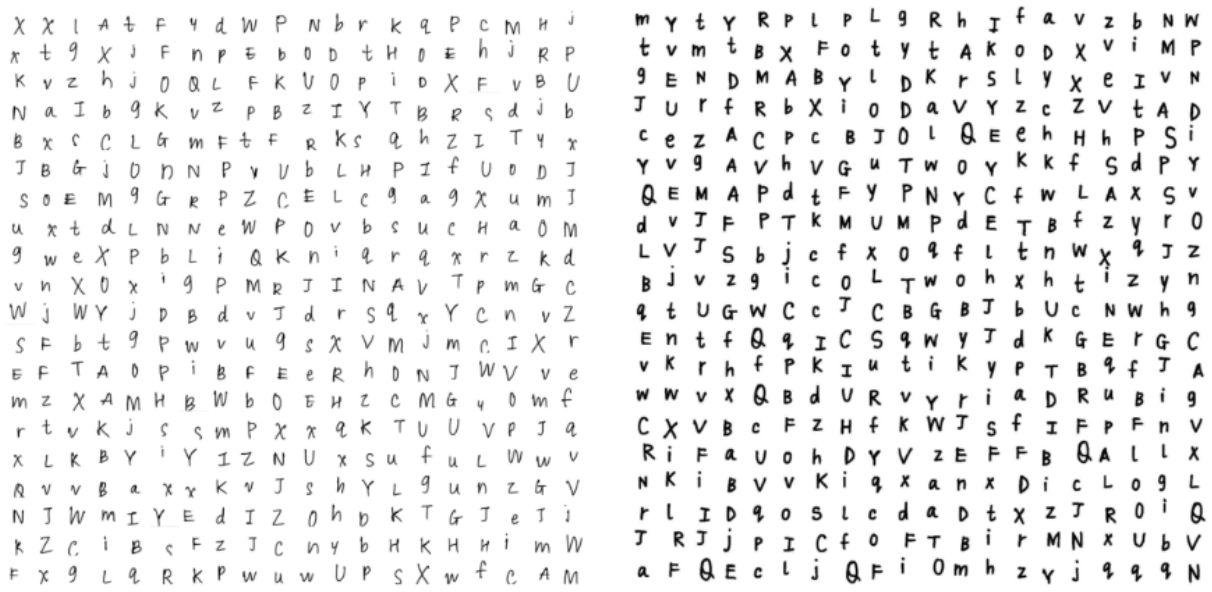


Figure 2. Left: Handwriting dataset 01; Right: Handwriting dataset 02

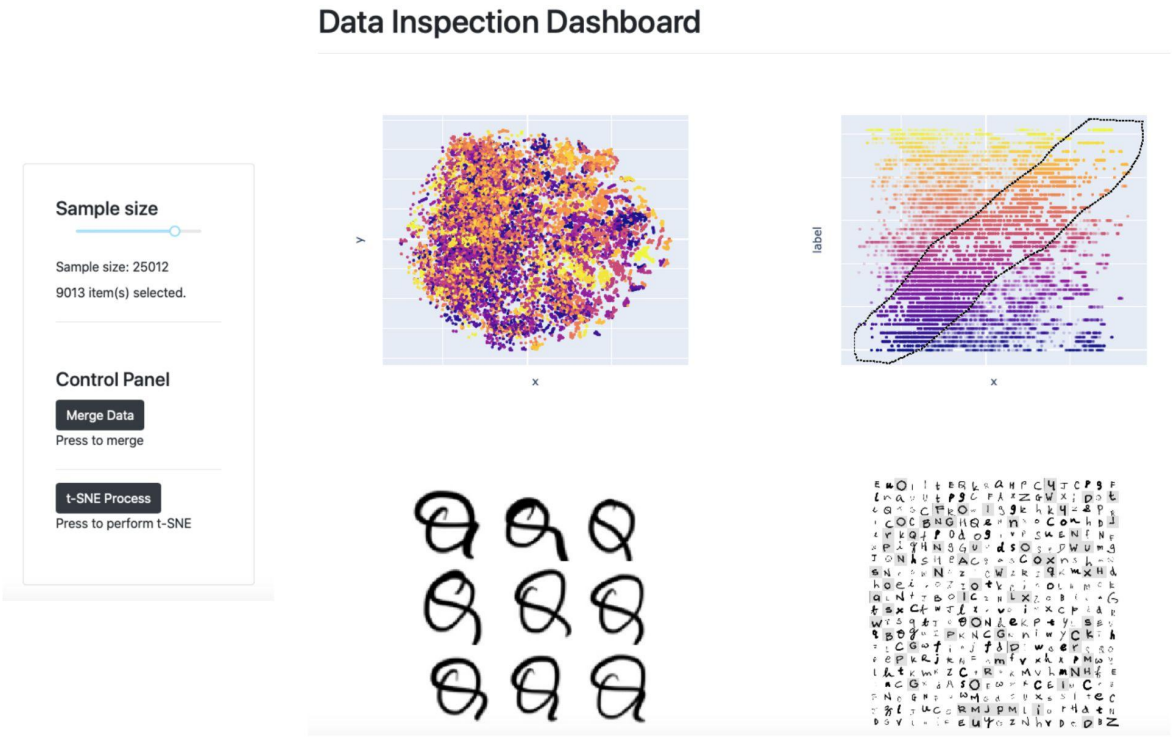


Figure 3. Left: Handwriting dataset 01; Right: Handwriting dataset 02

I always believed that my handwriting is pretty uniform, but this assignment proved me wrong. I have a great deviation in a few characters and that is where the generated data was inefficient. The first dataset I created did not recognize letters such as ‘c’ or

'q' or 'Q' that well. These observations were noted and the new dataset was made bearing in mind these changes. The letters in the first dataset were getting cropped in the data preprocessing stage. Centering all letters and being mindful of the sizes of letters was also an additional note in the second dataset. This off the shelf data collection technique proves to be helpful in this scenario as the user somewhat has direct control over the data,and has the ability to manipulate it based on desired requirements.
(View: Creating the data, testing it, making changes in the data as feedback and then testing again is exciting. It is a good feeling to have (somewhat) control over the results of the model through my inputs.)

The project files also include this interactive dashboard app that lets the user interactively select data from the entire dataset. I believe this kind of visual communication with data is really helpful for one to visualise the features of data and create a dataset of their desire. Figure 03 shows data selected using the lasso tool from the entire dataset which was used as my final training data.

TRAINING PROCESS-RESULTS

The training process is pretty straightforward. While training with the first dataset keeping all settings at default at 250 epochs, the results were not quite desirable - a few letters had been cropped more than required thus generating unclear letters. Letters that had a higher value of standard deviation in the samples naturally generated blurred samples.This was a prompt for having better data samples for training. A thicker brush stroke, clearer letters with minimum deviation within samples, and centering the letters in the worksheet provided were the points noted while making the new handwriting samples. These new samples were trained for about 270 epochs, keeping all other settings at default and the results were significantly better.

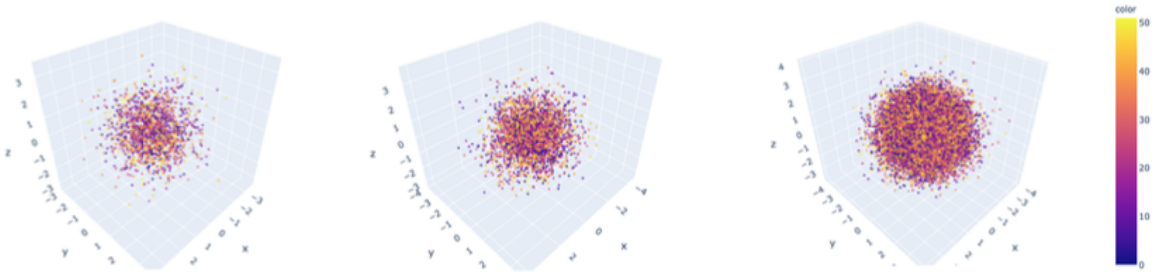


Figure 4. Left: Handwriting dataset 01; Right: Handwriting dataset 02

The above plots indicate the data samples and it is observed that the data is quite diverse. Seen below are training plots from all three runs. The first try with 250 epochs seemed a good amount of training as 270 was overfitting the model. The second image is from the improved handwriting samples trained for 270 epochs and finally the third dataset that was run for 250 epochs as well.The model starts overfitting at about 40 epochs.

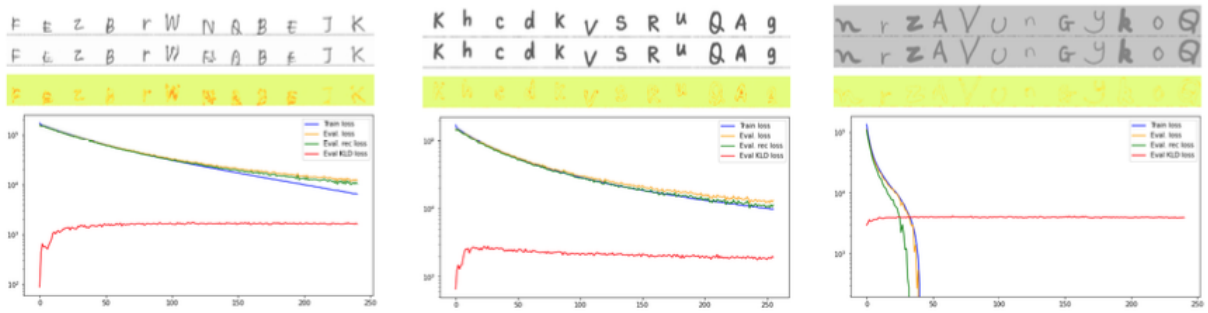


Figure 5. From left to right: Training plots for first to third datasets

For testing the model, “fake” text was generated using the GPT-02. The results were quite amusing! The images below show the rendered images for the following text:

“Does Architecture need software developers?

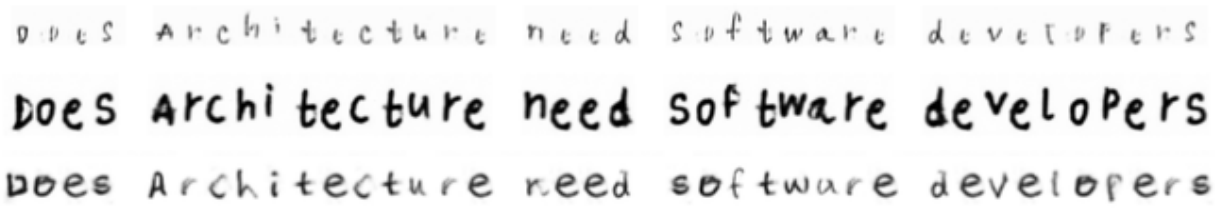


Figure 6. Top to bottom: Handwriting 01 Generated, Handwriting 02 Generated, Handwriting 03 Generated

Capital letters seemed to have a greater latency for some reason in my datasamples and they were tougher to optimize. Curvy letters such as “s”, “c”,”o”, “q”, etc also had a greater standard deviation. A thicker stroke automatically generates very clear characters.

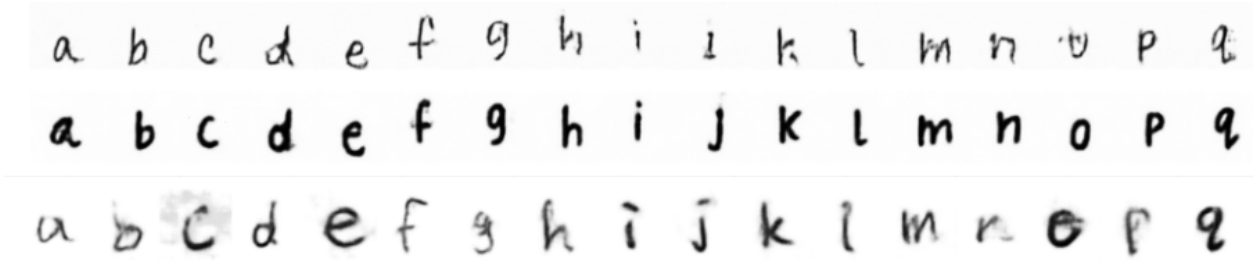


Figure 7. Alphabet Catalogue

CHALLENGES | BOTTLENECKS

Creating the data samples is an extremely important step in this workflow but can get a little monotonous. Using the first file for training the model was quite inefficient. The tuning of individual characters is really frustrating and seemed to be ineffective to me in the first place. The results of the entire data set was initially quite disappointing - the characters were extremely blurry. With the new notebook and being able to choose data within the code, the data processing pipeline was simplified a lot. The character tuning also had prompts in this one and was way more efficient. The dashboard is a great visualisation tool but is not super intuitive in deciding what kind of data to select.

DISCUSSION

This assignment was exciting to me because I was a direct stakeholder in the machine learning process. I wish I had access to this tool when I was in undergrad school - would save hours spent in note taking. It is interesting to think how this assignment would have an extended use case within the field of architecture and what sort of labelled images could be used to generate new ones.

Participant #8

IDENTIFIABLE DATA REMOVED

Learning Matter Homework 2 - Reflection

March 6, 2021

I used an XP-Pen StarG640 digitizer and both Adobe Acrobat Reader and Adobe Acrobat Pro to create the data. The line thickness was set at 2 pt. The software created vector representations of the characters. Therefore, it sometimes created unwanted artifacts at sharp corners of the character outline. In general, I deleted those characters and recreated them. Sometimes, the eraser tool allowed me to trim off the unwanted area of the character, but the tool was imprecise and often did not produce satisfactory results. I don't have much experience with a digitizer so the feel of the pen in my hand and the feel of the pen on the tablet negatively impacted my penmanship quality. While I think the system I used was a bit frustrating and did not capture my script well, it was still a worthwhile experience. In fact the frustration highlights an interesting paradox of capturing data in this way. From the presentations today it was evident that the best results came from very uniform, almost robotic character creation. For most people this is an unnatural and very self-conscious way to work. It seems to me that to really capture the individual style of a writer, woodworker, etc., one would need to capture their movements and results when they are in a flow and operating without self-conscious manipulation of the tool.

Images showing how the vector graphics processing modified the shape of the characters. Note the elongated corners and squared line caps.

The results from my first data set did not provide me with any clues on what to do differently when creating the second set. Perhaps more specifically, I was not able to recognize any of the clues that were probably in the results. I imagine that an intuition could be built if this tool or workflow was used frequently. The chance to build skill and judgement with the tool could make it a potentially rich way of working. So overall, the idea is very intriguing. However, as I mentioned above, the process of creating the data is unnatural and time consuming. That dissuaded me from exploring by creating more and different data. Since I did not know what specifically would improve the results, for the second set I focused on making the characters more uniform in size, shape and position in the frame. I was also more strict about recreating misshapen characters or those with the artifacts mentioned above. For this set I used Adobe Acrobat Pro which seemed to perform slightly better. Using a large collection of data created by others could be one way to lessen the task of data creation. But, my sense is that it then becomes a different pipeline and loses some of the

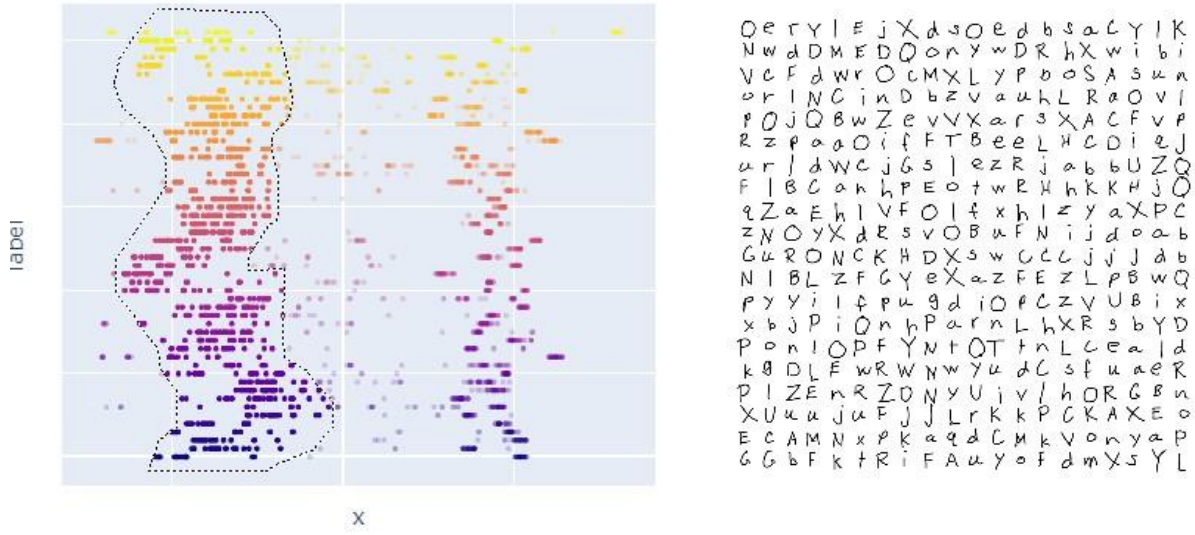
qualities that make self-generated data so unique. That is not a comment on the quality of the results, just that I feel it becomes a different tool.

My second set of characters did produce better results than the first. As will be explained below, I did not use the class data to create a third font. The results of those experiments were on par with the quality of the results of the second data set alone.

Results from data sets 1 (left) and 2 (right)

I was not able to markedly improve the character shape using the interactive widgets. At most I was able to improve the contrast or remove a white or black blob from the character. If the character was poorly shaped at the start, adjusting the sliders did not recover it. The concept of navigating the latent space sounds promising and I guess I wanted it to be like the videos of faces and shapes transforming the latent space is transversed. However, it was difficult to predict what effect the sliders would have on the character. It almost seemed random because moving the slider in a single direction did not seem to change the character in a progressive way. For the combined data set, I attempted to adjust each letter but gave up at 'f' because it felt like a lot of effort with little result.

Although I seemed to always find a way for it to not work, when I did get it working I found it to be helpful and a potentially powerful part of the pipeline. I did not try to use the t-SNE representation for selection. It seemed too likely that I would miss a whole letter or two. It was interesting to play with and see how characters were grouped. The graph with the characters in rows was more useful to me. Instead of using the whole class dataset, I combined my own two data sets and only visualized those. Then I selected a column of characters to train the model. Based on the results presented in class by others today, I felt that this might be a way to avoid the problems of a too varied data set. By selecting only a column I thought I might avoid some outlying characters and thereby improve the model training. My results were not as good as I hoped but they were better than the results of a combined data set without selection.



Images showing the column of characters selected from the combination of the two data sets I created.

Results from combined data set (left) and selection from combined data set (right)

I am not sure that I was actually able to control the process with the data. It was definitely fun to try though. It felt more engaging than tweaking arbitrary parameters. Like I said above, I sense that one could get into a conversation with the tool through the data. That could be enjoyable. A less tedious data collection process would definitely help. I can see the design of the data collection being a creative process in itself. Of course, writing code is more immediate and direct and that has its value too.

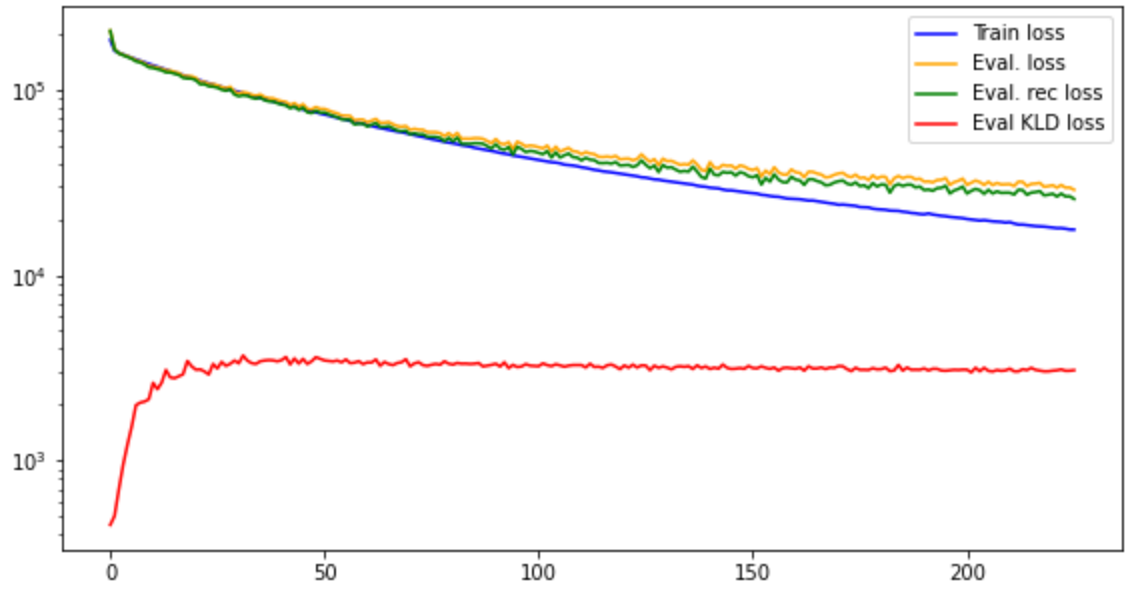
My results were disappointing, but seeing [retracted]'s amazing results proves that legible results are possible. However, it was mentioned in class that exploration of her latent space did not actually give her the

ability to tweak her characters and fonts. Since that is the promise of the tool, it needs to be addressed somehow. Does the latent space actually allow for search and exploration of different font types? Is there a better way to navigate the space?

Work notes:

I changed the padding in the image processing code to 10 because the characters were being cut off.

All models were trained to 250 epochs and not other parameters were changed. Training on the combined data set took significantly longer than training on a single set of 52 characters. Below is the training plot for selected data, but all of the training plots looked similar.



Before I saved each character in the font array, I removed all gray values. I used a threshold of 0.8. Pixel values below the threshold became 0. Pixel values equal or above became 1. I think this improved the clarity of the font. I also adjusted the 'squeeze' value to give each character more white space between them. I believe this improved the legibility. Finally, I wrote a script that loaded the numpy font array and converted it to a dictionary so that the provided render methods could be used to print examples using the saved fonts. The provided render methods had to be modified to ignore punctuation and any character not in the 52 character font.

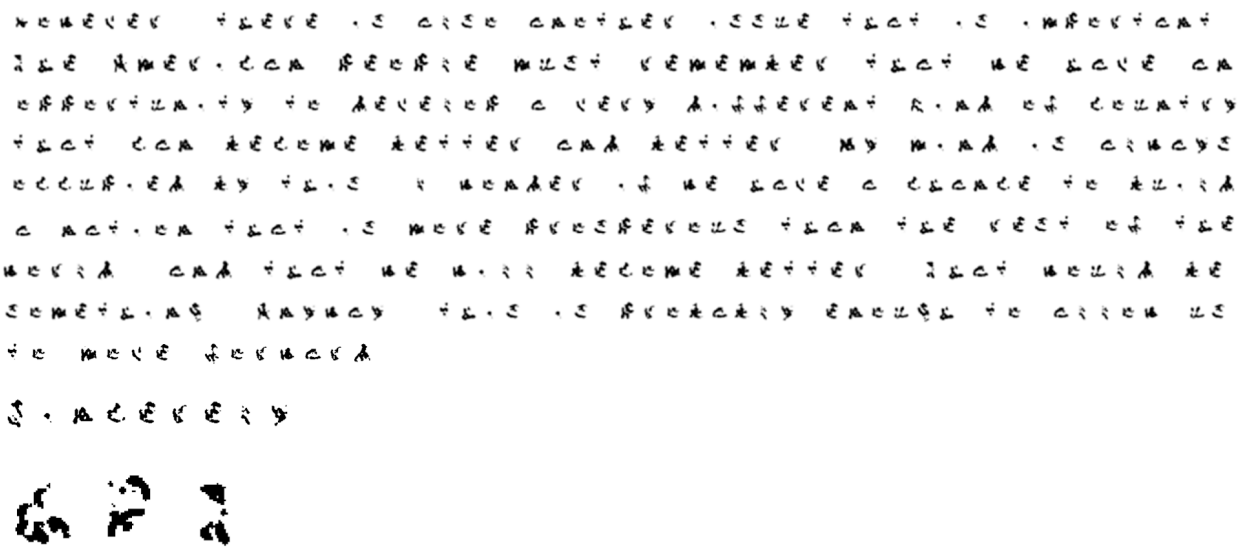
Font 2 before and after gray removal

Paragraph created with GPT-2 website:

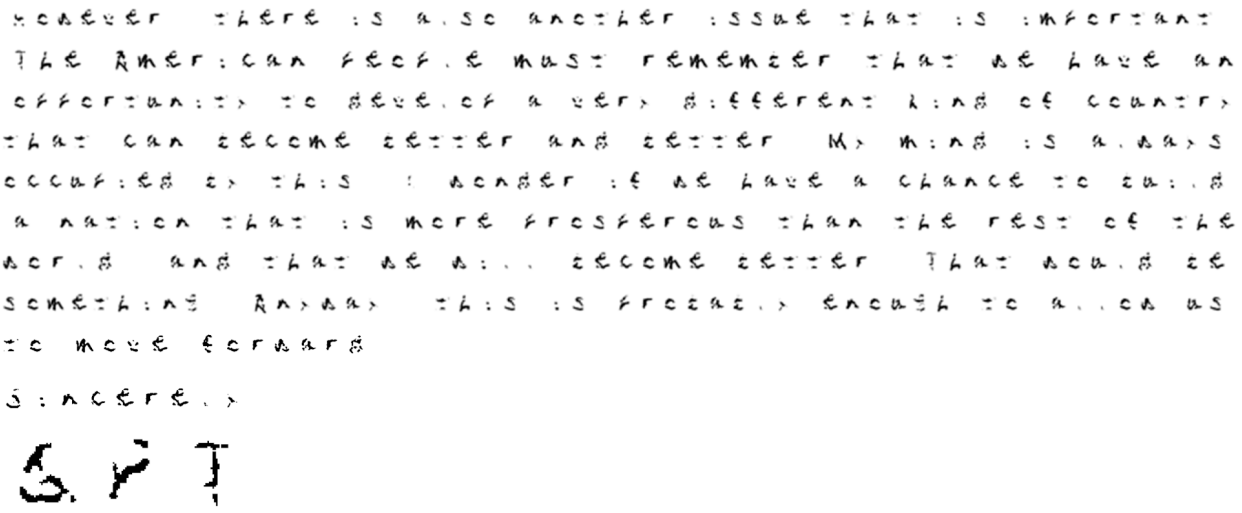
However, there is also another issue that is important. The American people must remember that we have an opportunity to develop a very different kind of country that can become better and better. My mind is always occupied by this. I wonder if we have a chance to build a nation that is more prosperous than the rest of the world, and that we will become better. That would be something. Anyway, this is probably enough to allow us to move forward.

Sincerely,

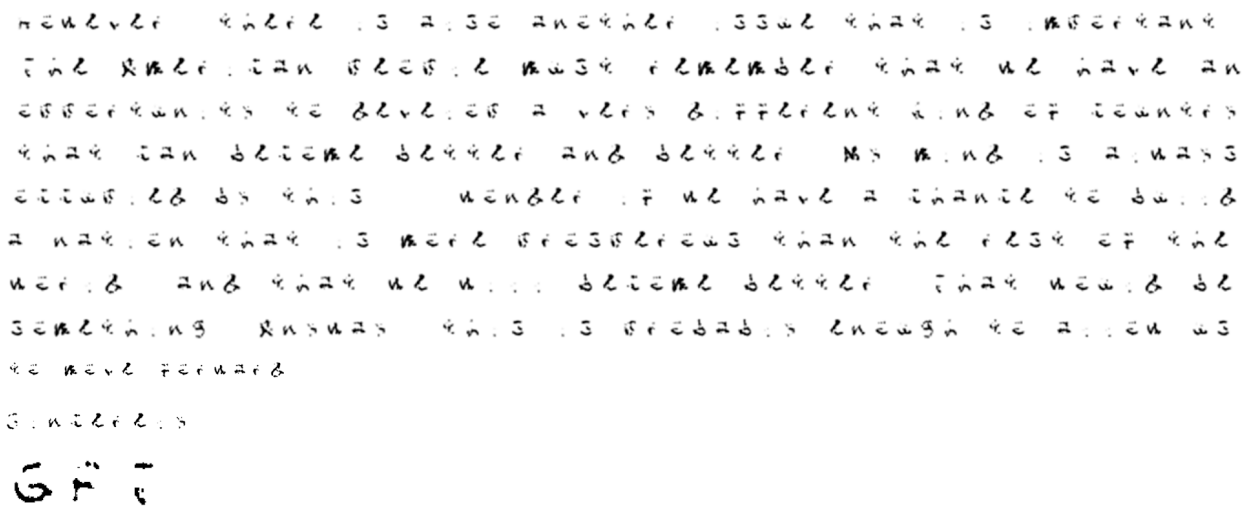
GPT-2



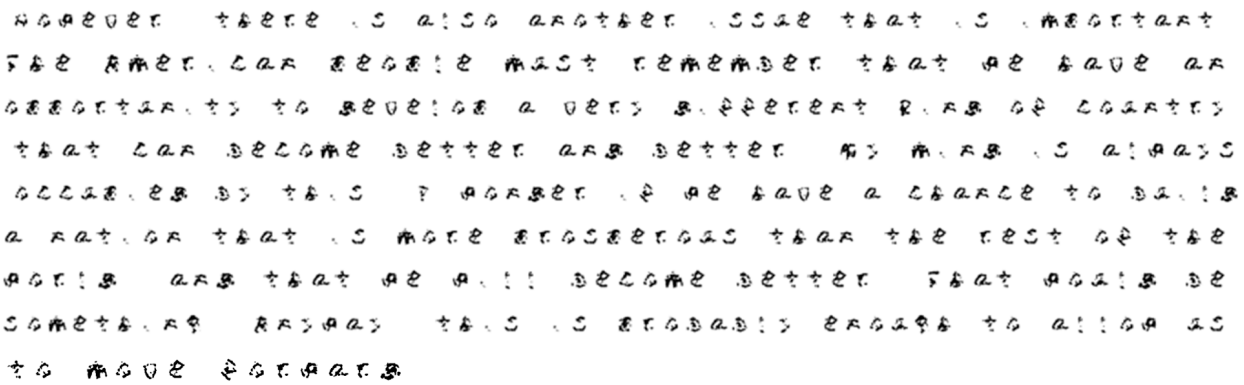
Paragraph rendered with Font 1 (data set 1)



Paragraph rendered with Font 2 (data set 2)



Paragraph rendered with Font 3 (combined data set)



Paragraph rendered with Font 4 (selection from combined data set)

Participant #9

HM2 Reflection IDENTIFIABLE DATA REMOVED

Process:

In the first iteration, I just used a different number of epochs to train the model so as to get familiar with the pipeline.

In the second iteration, I use four different datasets and 3 different numbers of epochs to figure out the best number of epochs.

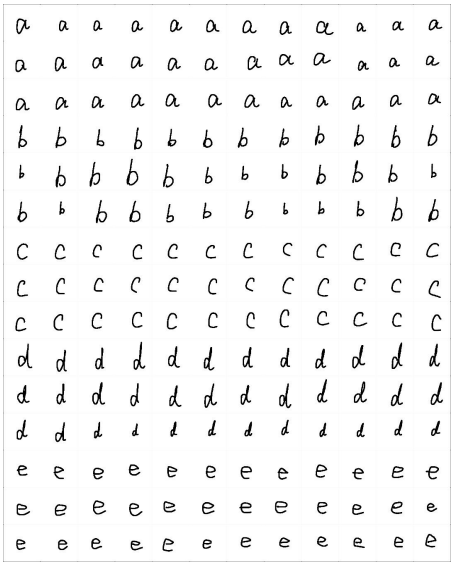
In the final interaction, I deliberately use 3 different datasets to train the model. The number of epochs is chosen based on the previous experience. The first dataset uses only the second dataset I created. The second dataset combines the two datasets I created. The third dataset uses all the ten datasets.

Reflection:

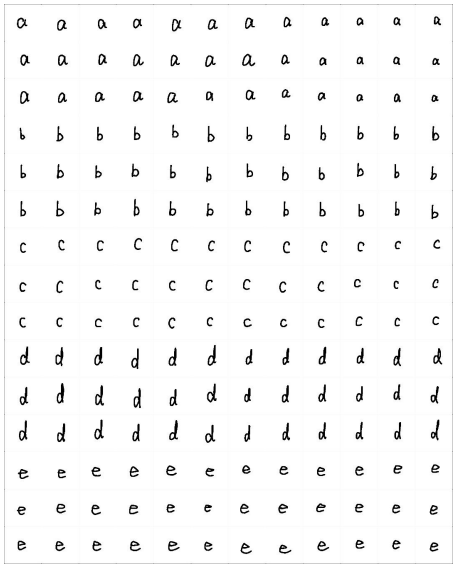
1. The data collection process:

   Compared to using an off-the-shelf approach, the data collection process really helps in understanding how the data collection would influence the training result, especially after knowing how other people collect and process the dataset differently.

   At the very beginning, I am not aware of this kind of difference. I kind of overestimated the generalization learning ability of the model. Thus, I deliberately write the letter in different sizes and different ways, hoping the model could generalize and learn from them. However, the result is not promising. In the next stage of the design, I try to control the size and shape. The result is way better.



First Dataset



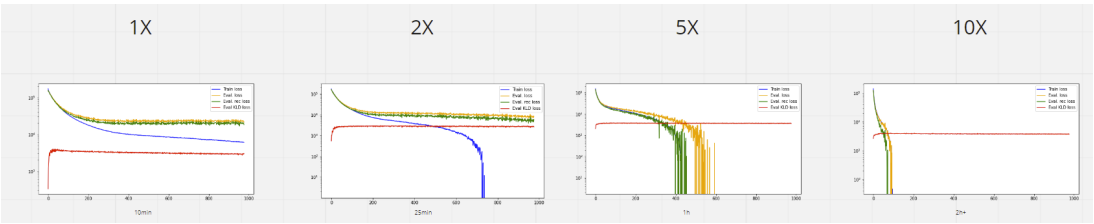
Second Dataset

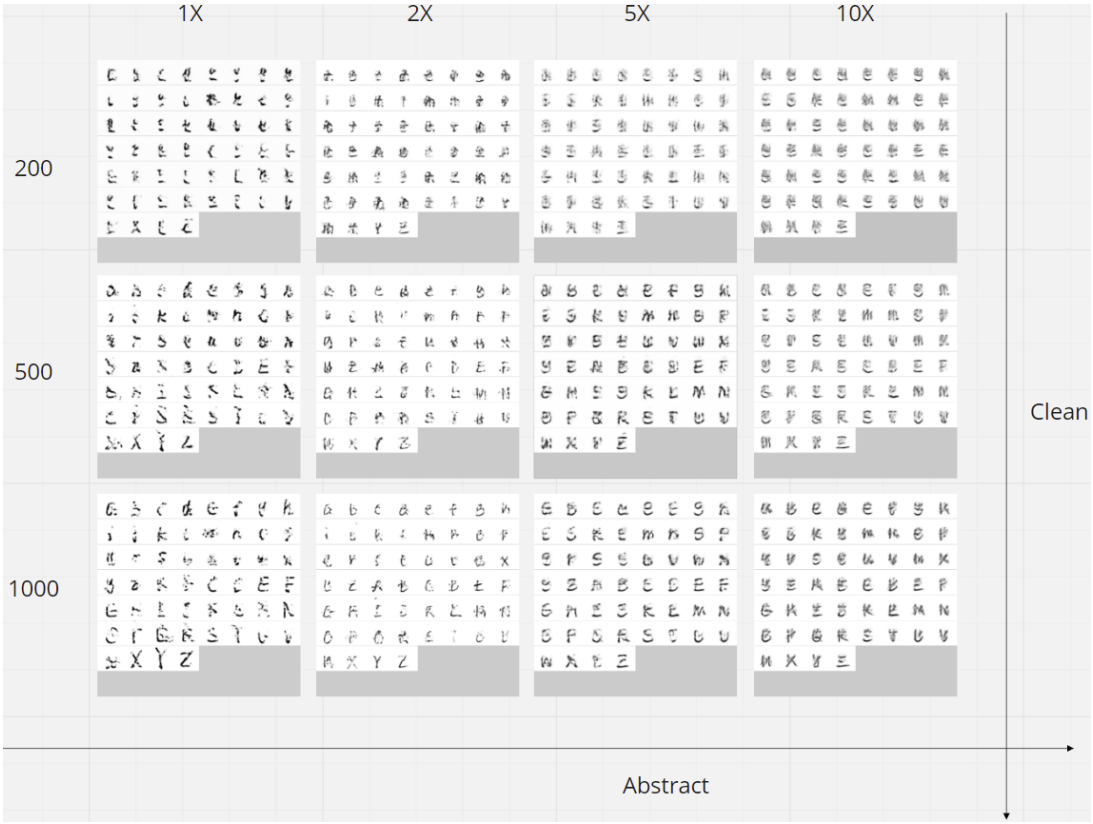
Besides, I use the IPAD + Goodnotes as the working platform for data collection, which save a lot of post processing time.

2. Your experience with the training process through providing various sets of data instead of only changing the architecture and hyperparameters.

In addition to deliberately controlling the size of the letter in the second set of data in order to improve the training result. I have also used some image preprocessing to improve the training result. Although the background is clean enough from human perspective, the training result is clearly improved using the pre-processing dataset.

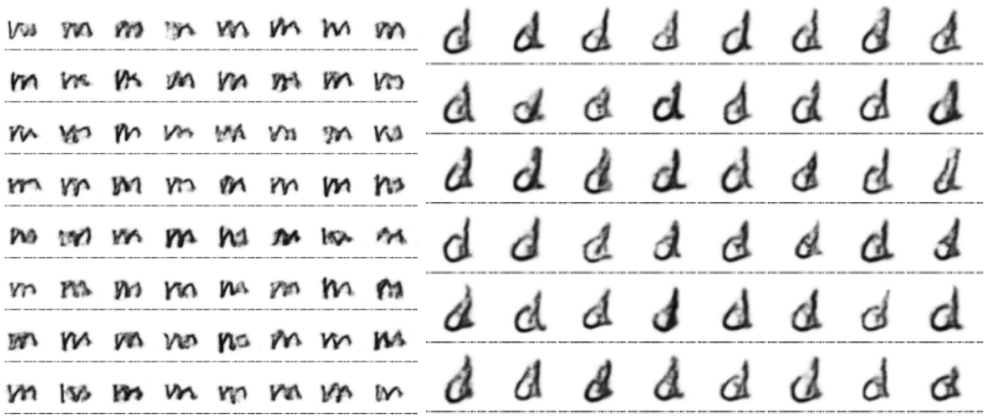
In the presentation, I am so surprised to see my classmates deliberately use the bold letter to improve the training result. I also show my process of using a different number of dataset to influence the training. I use one set, two sets, five sets and ten sets separately with 200,500,1000 epochs to train the model. For two sets of data, it seems that 1000 epochs are definitely enough. While for 10 sets of data, the generated letter still has some features from other letters and is not clear enough.





3. Your experience with the navigation of the latent space using interactive widgets in the CoLab notebook.

   Navigating the latent space using interactive widgets is so interesting, especially while you are using a big dataset. Although we can't see the actual latent space, changing the mean and standard deviation really gives me understanding of how the letters cluster in the latent space. And while using a large dataset to train the model, I could really see how different types of handwriting emerge with comparatively large standard deviation input.



4. Your experience with the data viewer/selection dashboard.

   It is good to have the data viewer/selection dashboard. The program is easy to use for merging data and label data selection. Moreover, while the size of data is relatively small, the t-SNE method could correctly visualize the data in 2d space, which makes the selection of similar dataset easier. However, while the size of data goes bigger, the 2d space is not enough for visualizing and it is hard to visually explain why part of the dataset is concentrated in the same place.

5. How was your experience with using data as a method of controlling an ML model compared with your experience of using code to modify an ML model?

   Using data to control the ML model is more intuitive that you can clearly sense the relationship between the dataset and the result. It is so amazing that while using the large dataset, not only the generated letters themselves, but also their position tends to be better.

   However, it is also a little boring since a lot of time is spent on making the dataset and personally speaking, I enjoy the process of finding logical relationships between different parameters and the training result.

6. Some suggestion to the model and pipeline

   a. Some code could be added to load the saved font to generate the paragraph.
   b. The domain of the mean at the interaction part could be smaller for more precisely controlling the number.
   c. Code for implementing the training and saving could be separated into two parts since the plot will be cleaned when you just load an existing trained model.

Participant #10

IDENTIFIABLE DATA REMOVED
Ardavan Bigdoli
Inquiry into Machine Learning and Design
10/22/21

The Second Hand: Reflection

Data Collection

Providing the data for our model required fifteen pages of handwritten words that were selected to fulfill an ample diversity of letters. Given the previous semester's process of filling out a specified number of the same letter in a grid, this semester's approach differed in that as a reaction to the individually handwritten letters, in an attempt to capture the essence and organic characteristics of handwritten letters in a word, extra effort was taken not only to fully write out words but to also have to go into the CVAT online tool and manually identify, in sequential order, the bounding box of the individual letters. This approach was tedious, though highly warranted, as letters written in a word provide more natural inconsistencies than just printed letters in a box.

With the option to utilize a digital tablet or manually write with pen and paper, this was already a significant factor in determining the final outcome of my dataset. I chose to do paper and pen almost without choice, whereas my classmates had the option to use a tablet due to owning one or borrowing one. Though I could have used a Wacom tablet, I found the process unnatural given initial attempts and concluded to create my dataset by hand for practical purposes and to capture my handwriting in the truest sense.

Comparing the digital written set of my classmates to my own handwritten scans, there are some key visual differences that affected the data set. Firstly, on a tablet, the writing software naturally smooths your written gestures as well as tries rendering your writing to look a certain style. In the case of using pen and paper, I found my writing to be relatively messier in appearance, and the thinner pen not blurring any of my linework. Too, my scans were noticeable of varying contrast to that of the black and white nature of writing digitally. Lastly, another noticeable differentiation between a digital set vs my handwritten set was that my words often did not fit and had to be written on other parts of the paper, whereas digitally, one can zoom into the page and write with correct and consistent spacing.

The above steps naturally give a lot of agency in decision-making for the one making the data set. I believe that this offers the most realistic example of trying to capture one's own handwriting in the case of training a machine, though with that being said, the many variables that can create variety in the data mean that more data would need to be collected than what we alone were able to produce. In that case, off-the-shelf data provides that much more convenience and access to that large amount of data. That being said, there is a lot one might not know about the data being collected and where it comes from, given all the potential for biases.

The last point touch on is the CVAT online tool itself, which was the most demanding task in our data collection process. I do speculate if there are more automated approaches to this step, but naturally, to be precise the human eye is best in this scenario. But the CVAT tool itself did export the labels in illogical ordering at times and that did tamper with the training process. Though a script that ordered the labels by x,y coordinates would have been a solution, I myself fit words wherever I can on the page in order for my handwriting to fit, which meant such a script would not work; classmates that used digital writing tools would not have this problem. Perhaps this small consideration of how one designs sheet we write on can go to lengths to make this manual process more efficient.

Training Process

The three sets of data that I have trained were pages 0-10, pages 0-15, and the thinner lettering of the collective class data. I naturally embraced that I was looking at results with thematic similarities rather than creating something very different between fonts and allowed this to be a greater point of comparison.

Comparing the results of the first font to the second was intended to identify the visual difference when providing more training data. The training loss charts had similar tendencies, which made logical sense. The final results also had a pretty expected outcome, where the fidelity of the font trained with 15 pages, was noticeably crisper. This is seen most in some of the letters I provide more inconsistent results for, reflecting in a blurrier type for the final font. This is in particular for my letter ‘g’, where ‘g’ in the second font looks noticeably more defined than the first.

The collective data from the class results, generated using the selection dashboard, had noticeably more inconsistencies in results. This is likely due to the great variety of styles in handwriting data. It is clear to me that when more variables are in place for training, more data is needed to supplement that. Interestingly, the font made from the collective class data of thin letters seemed to have a closer resemblance to my own handwriting than expected.

Notebook Interaction

The interactive widgets in the CoLab notebook helped me understand the way the model and its latent space operate more intuitively. By being a responsive graphical user interface, it was easier to create and observe the results of the reproduced letters. Generating fixing the font with sliders representing the mean value for the sampling and the standard deviation was intriguing to see visually effect-wise, though honestly I still have lengths to go to understand the mathematics behind these values.

My biggest takeaway from the interactive widgets is how it makes me reflect on machine learning being a more visual and tangible process that can be integrated with people like my current self who understand the process at a higher level. There is a design agency in generating the font based on changing values of the latent space and I find that that is not something a graphic designer would typically consider but now can.

Selection Dashboard

It being my first time using an Anaconda environment, I certainly saw the benefits of how a package manager can be useful in developing tools that can also be shared with the class, like this dashboard. As a user interface, I think it builds even further on the ideas discussed in the interactive widgets on CoLabs, though my critique is that it is more interactive than it is responsive and informative (though that does not lessen the potential for what it can be). I enjoyed seeing it as an all-encompassing tool for data selection and directly data processing and learning to that of exporting. Though I did have some personal issues, like when resetting the model it did not always work for retraining, and I had to restart the app. That being said, the app felt generally very intuitive and with more visual cues could see itself realized as a great app for training data.

Data vs Code

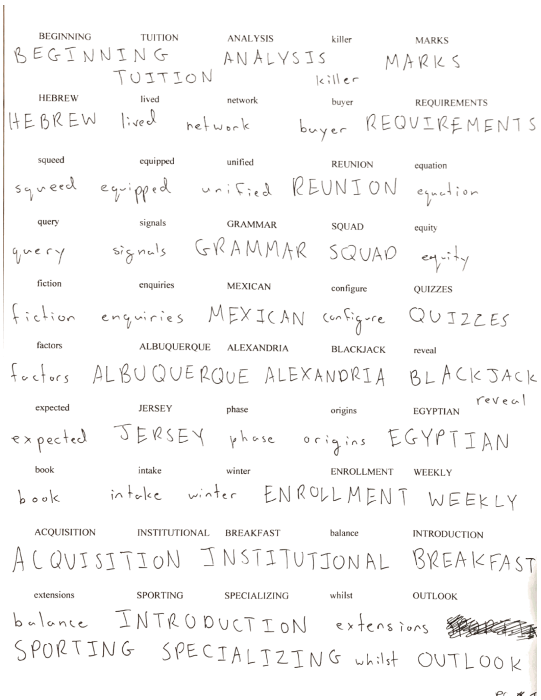
The data itself and its creation, the way it is labeled and collected, all play a large determining factor to the ML model, I'd argue, more than the code itself, at least to my current knowledge. My reasoning is that there is a seemingly infinite amount of possibilities to the training data itself, and a rather creative effort is required to compose a means to get useful data that can be trained by the machine.

In our previous assignment with the paintbrush strokes where we had to adjust the layers and code of the neural network, I find the process was a matter of experimentation and analyzing results. Of course, experience and knowledge are important too, but these things become learned and prescribed as a process.

In contrast, data collection itself seems to be a unique problem to be addressed given the desired objective of the model. There is a far greater amount of concern and possibility related to creating a data set. What are the biases in the data? How varied should the data be and is that something we striving for? Naturally, made clear by the dashboard, data in its vastness and diversity is what determines the code, rather than the other way around.

Overall Process

Data collection:



11.txt - Notepad

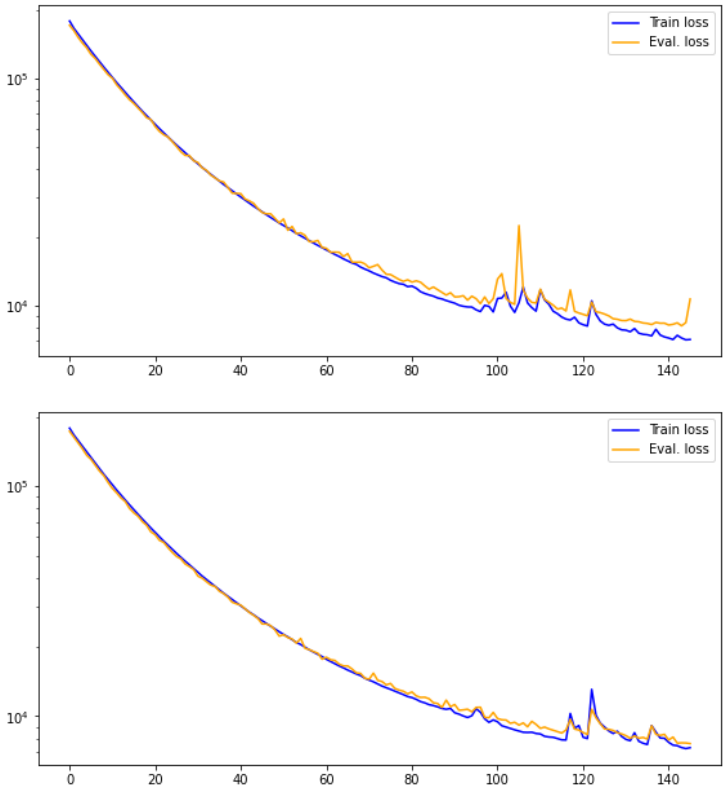
File Edit Format View Help

```
0 0.019810 0.069451 0.031862 0.031203
0 0.062647 0.071093 0.031153 0.030107
0 0.098050 0.071367 0.025488 0.022992
0 0.128850 0.073556 0.023365 0.024084
0 0.160006 0.074377 0.026200 0.023538
0 0.189390 0.074104 0.021244 0.025180
0 0.219128 0.073008 0.028321 0.026276
0 0.242849 0.107220 0.021950 0.030107
0 0.268693 0.105031 0.019826 0.024632
0 0.290997 0.104483 0.020535 0.025728
0 0.316841 0.102568 0.025488 0.024086
0 0.342331 0.105305 0.029738 0.030655
0 0.387293 0.080947 0.016285 0.016969
0 0.404285 0.079304 0.013453 0.014779
0 0.421279 0.080946 0.014871 0.011496
0 0.437210 0.074651 0.012744 0.022990
0 0.451372 0.078483 0.014162 0.008759
0 0.466243 0.080125 0.014162 0.014234
0 0.480049 0.074103 0.012038 0.024086
0 0.492087 0.079851 0.010621 0.009307
0 0.508371 0.072734 0.019118 0.021348
0 0.559353 0.105031 0.030447 0.027915
0 0.585196 0.103389 0.018409 0.022444
0 0.605729 0.103114 0.021241 0.027369
0 0.626618 0.103389 0.020535 0.022444
0 0.654940 0.101473 0.026197 0.025180
0 0.685032 0.101473 0.028324 0.028463
0 0.716541 0.100379 0.031865 0.030653
0 0.741324 0.101473 0.019118 0.025180
0 0.763272 0.097093 0.021950 0.027369
0 0.786640 0.097094 0.020532 0.029559
0 0.848949 0.099279 0.033279 0.030664
```

Above is the photo of one of my hand written sheets. Things to observe is that I used a scanning app on my phone, which loses contrast and is not necessarily fixed as a dimension. There are also some discrepancies within shadows and greyscaling. Also errors and awkward fitting remains in the data set.

Using CVAT was difficult - despite outlining letters in the correct order, it would export as a text file in weird ordering. I manually had to rerun the CoLabs script and calculate mentally how to edit the text file, represented on the left, to correspond with the correct order of the words.

Training Evaluation:

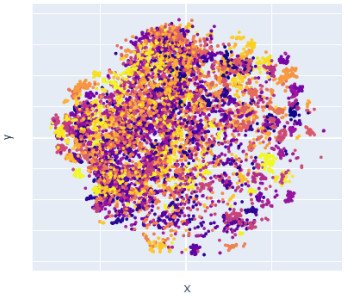


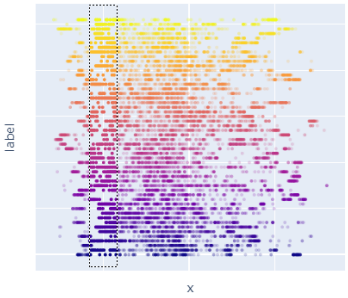
The above two charts represent the training and evaluation loss throughout the 150 epochs. The top chart is from Font 1, pages 0-10, the second chart from Font 2, pages 0-15. Interestingly about the model and dataset is the large spikes that occur between epochs 100 - 120. I wonder if this has to do with some of the more varied examples of letters (like ‘g’), when the model begins to overfit and get more specific.

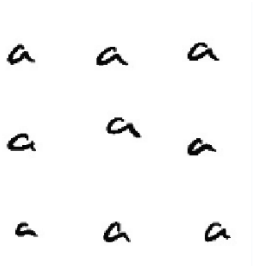
Here is the raw font sets, the above being Font 1, below Font 2. Font 2 is slightly crisper and less fuzzy in letters like ‘g’ or ‘Z’, but having the progression is interesting to

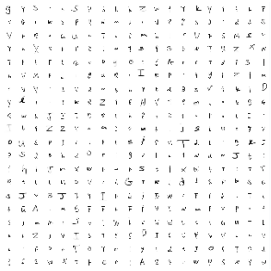
observe when using greater amount of data that is relatively similar and cohesive, as I did not change how I wrote for pages 11-15, contrary to my peers.

Dashboard:









Utilizing the dashboard to create a third font that utilized the greater class dataset was fairly intuitive, and more straightforward than the notebook process. During the notebook process, my own misunderstanding of the size of the data as intended meant I had to change the code to suit my size of data (0-15 pages was not the intention). The dashboard took all that possibility away and streamlined that process.

For my own selection of fonts, as illustrated above, I stuck to the collective thinner font to observe against my own writing style.

Post-Processing:

Using the scripts from the CoLab rendered results at high fidelity. As a result, I did not need to edit those results in Photoshop. The dahsbaord instead, outputted the paragraph texts at very low fidelity, and it was hard to discern the nature of the handwriting unless one was to individually zoom into the text and stitch together screenshots.

To the left is the post-processing underwent for the paragraph output of the dashboard font. The dashboard font had a grey background around each word and the dimension of the image makes it difficult to discern the actually font.

Participant #12

Reflection of *The Second Hand*

The Second-Hand assignment provides me with a chance to experience the complete process of a simple Machine Learning project, from data collection to training, then generating some rendered texts as project outcomes. The valuable part for me in this assignment is not only understanding how to create auto-generated texts from training my hand scripts but also how each step is interrelated with others. For example, a minor change, or defect, in the dataset, model, or parameter causes impacts on the training result. Therefore, I organized this reflection paper according to the steps of the assignment, narrated my methods of reaching goals and my experiences throughout the whole process.

Data Collection

1. Handwritings:

The first step, and the most fundamental yet painstaking one, is data collection. In this part, I finished two rounds of English words handwritings. In the first round, I wrote ten pages of words with 50 words on each, 500 words in total. I exported the blank word sheets into ten PNG files with a dimension of 4400 by 3400 and sent them to my iPad, enabling me to write with Apple pencil on the app Sketchbook. The reason for choosing Sketchbook is its clean and handy interface and better file managing method, which could read files from google drive and share the completed writings to my Mac by Airdrop *(Figure 1)*.

Figure 1: Preparing Handwriting sample

I experimented with different stroke weights and letter sizes to avoid over-similar samples, left enough space between each of the letters because I knew I should annotate them afterward. It took me around three hours to finish the first ten pages.

However, annotating those letters is even more time-consuming. I spent around two hours drawing rectangles over each of the letters on the website CVAT before realizing the disordered indices problem. When I was trying to process the first-round handwriting with the CoLab notebook, I found most of the handwriting letters didn't match the labels from the 1000 words TXT file, because the indices exported from that website sometimes do not follow the order of my annotations.

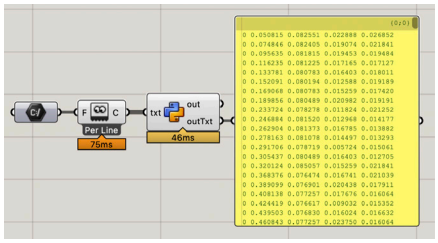


Figure 2: GH program

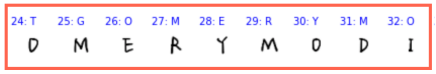


Figure 3: Disordered letters

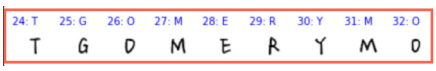


Figure 4: Fixed letters

The initial solution for this problem is exporting the annotations right after I finish it. Unfortunately, this method has only an 80% success rate, which means I'll still get two pages of incorrect annotations in ten. It is also time-consuming because I should annotate the wrong page again and again. So, for the productivity of my second round of handwriting data collection, I wrote a sorting program in Grasshopper in Rhinoceros 7 by taking advantage of the high interactivity of the software's Panel node.*(Figure 2)* The sorting logic is: first to sort the rectangles according to their Y-coordinate, put them into different rows, then in each row, sort rectangles by X-coordinate. This program functions flawlessly to correct the disordered indices. *(Figure 3, 4)*

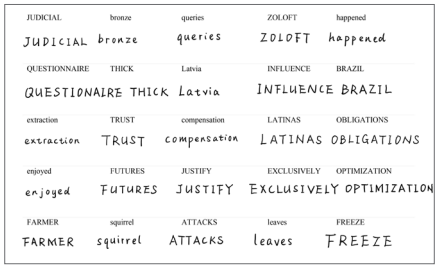


Figure 5: Samples of the first round writing

Based on the experience from the first ten sheets of handwriting, the

second round went faster and fluently. I only wrote five pages, 250 words in total, and I tried to make the stroke thicker and write each letter as large as possible, hoping to provide clear samples for the machine to train. *(Figure 5, 6)*

Figure 6: Samples of the second round writing

2. Samples:

Because of English's language feature, the total of 750 words containing more samples of the letter "e(E)," "a(A)," and "i(I)," while only including insufficient samples of "b(B)," "j(J)," and "m(M)." With the help of the CoLab notebook provided by the class's instructor Ardavan Bidgoli, I standardized the sample size to 64 for each letter; the identical sample size ensures a less biased training result.

3. Thoughts:

It is the best way to get handwriting samples by writing 750 words for now, and I think this method reflects my writing habit because I wrote in the context of words instead of single letters. From the training and paragraph rendering result, this method reflects my way of wiring every single letter but does not really show my way of writing words and texts; I prefer to connect letters in each word in regular writings, but I purposely separate letters for a better annotating outcome during data collection.

Another limit of getting natural generative hand scripts comes from my method of collecting samples. Using an iPad to do the writing makes my data collection easier, but my handwriting samples became overly refined because I can rotate, zoom the sheet, and redo each stroke. The samples would look wilder but natural if I wrote on actual papers with actual pens.

Training:

1. My Datasets:

The first two training rounds were based on my samples; I used the first ten sheets of handwriting in the first round and the other five ones in the second round. Unfortunately, the unmatched data structure between my sample and label files prevented me from entering the training process. The handwriting sample's structure is (52, 36, 64, 64), where 52 means total English letters (lowercase and uppercase), 36 means the sample number for each letter, and 64 * 64 indicates sample dimensions. On the other hand, the labels' structure is (1872, 52), which is very different from the samples' structure. *(Figure 7)*

```
Loading your own dataset
/content/drive/MyDrive/ML Temp/HW02/datab
X shape: (52, 36, 64, 64)
Y shape: (1872, 52)
```
Figure 7: Mismatched datastructure

```
print (x_path)
X = np.load(x_path)
###Reshaped here
X = np.reshape(X, (1872, 64, 64))
Y = np.load(y_path)
```
Figure 8: Reshape code

According to the knowledge from the previous classes, I got to know that the label.npy file is one-hot-vectors, which means in (1872, 52), 1872 indicates the total sample size, and 52 represents the labels in a one-hot way. So, visiting back to the (52, 36, 64, 64) structure, if I multiply 52 with 36, the result is 1872, identical to the label's sample size. So, I added one line of code in the file import function, X = np.reshape(X, (1872, 64, 64)), to match both numpy files' structures. *(Figure 8)* After this simple modification of the code, the machine started training without encountering any issues.

To find an efficient number of epochs, I did some test training ahead of formal training. The default epoch is 150, which is enough for the machine to yield a clear result. But when I observed the training loss and evaluation loss graphs of the 150 epochs, I figured out that the evaluation curve became jaggy around and after 110 epochs; this feature might indicate the machine started remembering the existing data instead of kept training. So, I reduced the epochs to 120 and trained the machine again. The result was a bit surprised that the

fewer training epochs generated a clear result than the more epochs. *(Figure 9, 10)* Of course, this surprising phenomenon could merely be a coincidence; it still indicated that 120 epochs were enough for training my handwriting samples, so I trained both rounds of my writings with the epoch of 120.

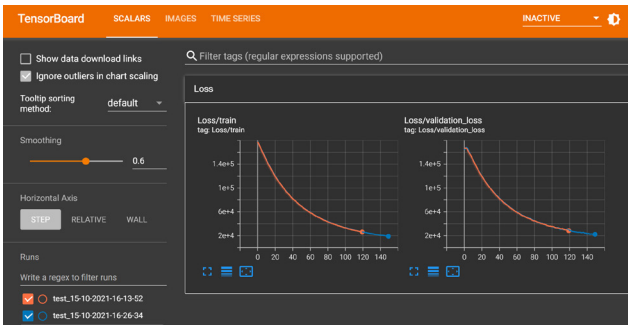


Figure 9: Training data

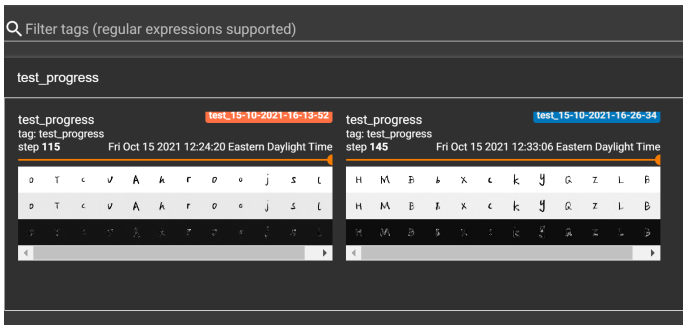


Figure 10: Training Samples

2. Latent Space:

The mean value changed the letter's appearance; the far the value away from zero, the less the letter looked like what it was. So, for instance, as I'm designing my 'm,' if I change the mean value to 0.6, it no longer looks like an 'm,' but something deformed and similar to the combination of two different letters. The standard deviation controls the variety of my font set. When I played with the letter 'f,' I set the item number as 6, which generated 36 'f's. If I set the STD as 0, all the 36 letters look the same; the higher the STD, the more varied the 36 letters. *(Figure 11)*

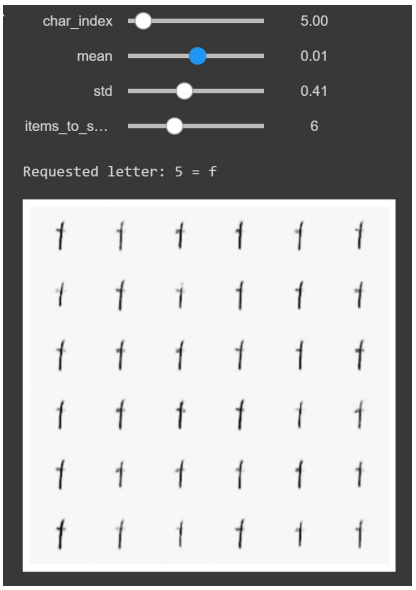


Figure 11: Playing with latent space

3. Combined Samples:

In the third round of training, I tried to get rid of a single handwriting source and was willing to train the machine with a combination of my and other classmates' hand script samples. I picked [retracted] and [retracted]'s first 10-page samples, the only two available datasets then.

It was a pity that I wasn't very confident in reforming the label data to match the new combination; I tried to keep the labels unchanged and matched the combination's data structure as the standard one, (52, 36, 64, 64). To do this, I randomly picked 12 samples from each letter from each person's dataset, and the filtered data's shape was (52, 12, 64, 64). Thus, when combining my, [retracted]'s, and [retracted]'s filtered data, I got a new sample set structured as (52, 36, 64, 64), identical to the previous training rounds.

As the new dataset has more varied samples, I decided to increase the training epoch to get a better and reliable result. I didn't do some test training but just arbitrarily set the epoch to 180. From training experiences, I believe this number is enough for the machine to yield a highly recognizable result. And it works.

4. Thoughts:

Three pieces of paragraph renderings from three training rounds look clear; that might be due to well-formatted samples and enough training epochs. But the product from the latter two training rounds appeared fuzzier than the one from the first round. *(Figure 12, 13, 14)* I attributed such phenomenon to insufficient samples; I only wrote five pages in the second data collection, and the combined dataset only included one-third of each person's hand scripts.

A noticeable defect of the renderings is the letters' size and positions: upper-case letters are in the same size as

lower-case ones, and there's no position shift for the letter "j," "y." Because during the annotation process, I tried to wrap each letter as tight as possible, hoping to make each sample appear as large as possible to increase the training precision. The result looked cumbersome, not natural enough.

I think there might be two ways to address the size and position issues. The first one is to refine the annotating process: require the annotate person to leave more headspace for shifted letters and white space surrounding the lower-case letters. However, implementing these rules undoubtedly adds to the data collector's workload. Another solution is to write some functions in the program to tell the computer to change the size or lower-case letters and change the position for special ones.

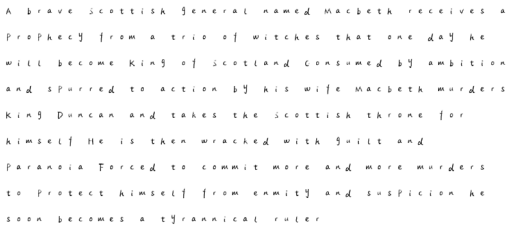


Figure 12: Paragraph from first training

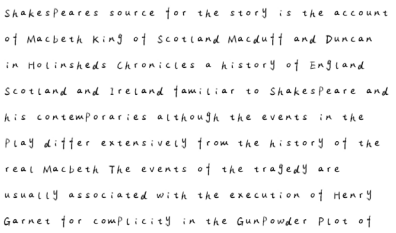


Figure 13: Paragraph from second training

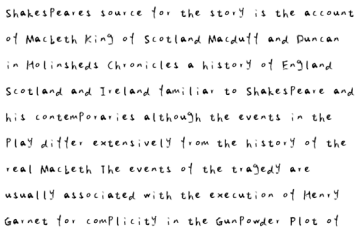


Figure 14: Paragraph from third training

Experiences from Different Interfaces

The dashboard provides me with a completely different training experience. When working on the dashboard, every step is very intuitive and has less visual burden than the notebook one. On the CoLab file, the traveling distance between each step is long and requires more attention; I should go through many many lines of functions to locate the executing code of each step. But in the dashboard, I find the buttons, labels, sliders without any effort; the interface on the dashboard is so clean. *(Figure 15, 16, 17, 18)*

I think the dashboard and the CoLab notebook are created for different purposes. In the CoLab notebook, everything is changeable. It's a perfect place for developers to refine the program and models. The dashboard provides an unobstructed interface for training; users can entirely focus on selecting data, training models, designing fonts, and rendering paragraphs. As a freshman in the field of Machine Learning, I prefer working with the dashboard, because the developer already did everything for me in the background; if I grew as a developer, I would like to work on the notebook files to keep refining each step of the whole project. Yeah, at that time, I should learn to create a dashboard for others

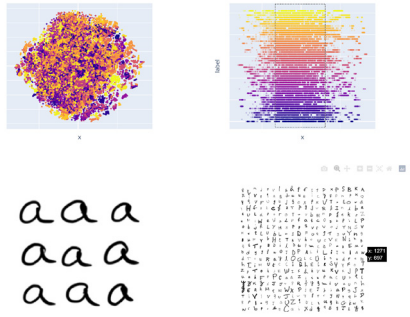


Figure 15: Dashboard interface 1

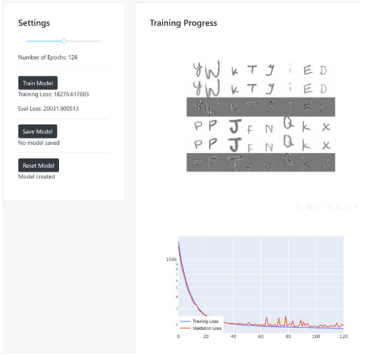


Figure 16: Dashboard interface 2

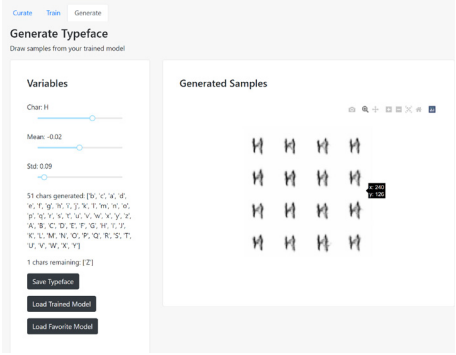


Figure 17: Dashboard interface 3

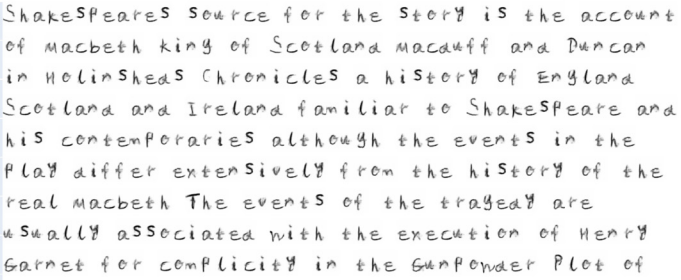


Figure 18: Paragraph from dashboard

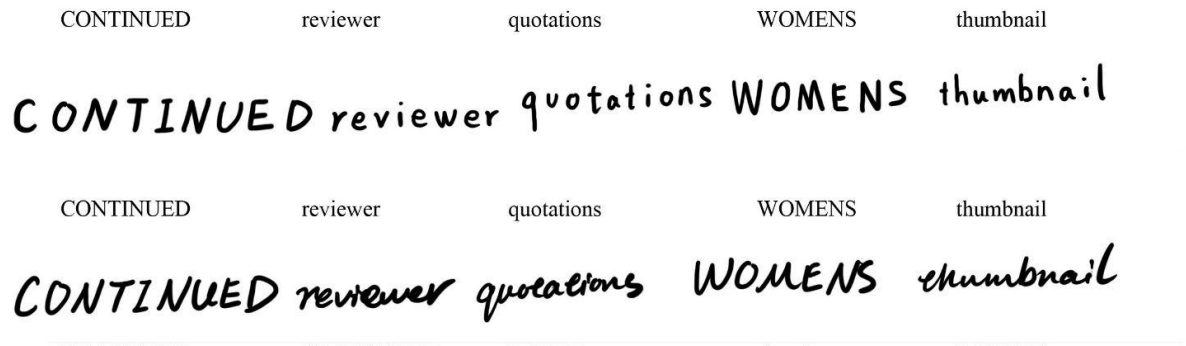
Participant #13

Reflection upon data collection and CVAE training

The secondhand data collection exercise is a thorough project to understand the process of data collection, training and generation. Particularly it helps identify the relationship between each step, the influencing factors that potentially affect the results, and how latent spaces play an important role in the post training process.

Data collections

Through my experience and presentation by other students, I've noticed that writing style, the media used, and the size of letters are the 3 key factors throughout this stage. My daily handwriting habits differs greatly from what would be desired by the clarity of data collection: 'f' would share the same stroke if it is repeated twice, 'a' and 't' would be connected if they are neighboring to each other, and 'g', 'y' would have their tails under the previous letters. In order for better resolution by square annotation, all letters are totally separated, while tails are trimmed. Below are illustrations of how data is different from daily handwriting.

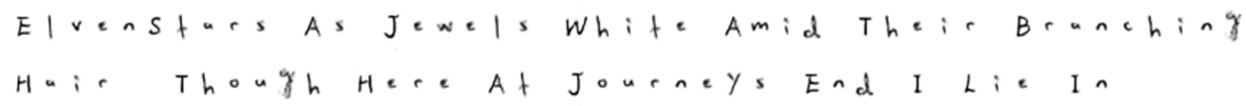


During the annotation phase, the most common problem occurred is the missing letters in a long word or shuffled index of square annotations. 'l' in a word that contains a lot of letters are often missing and hard to spot. Therefore using editable platforms like Photoshop give a strong support to workflow when corrections are needed, whereas handwriting would take significantly longer to adjust.

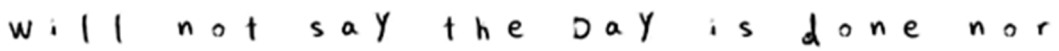
The Training phase

The CVAE model focuses on the accuracy and clarity of generative results. This is the part where the amount of data and its characteristics affects the training process heavily. The first set that contains 500 words was able to contribute abundant samples even in uncommon letters. Therefore the edge of the model it produced is shaper and hardly contains any stroke inconsistencies. However 250 words displayed many recognition errors during the equal distribution phase of post processing, thus its

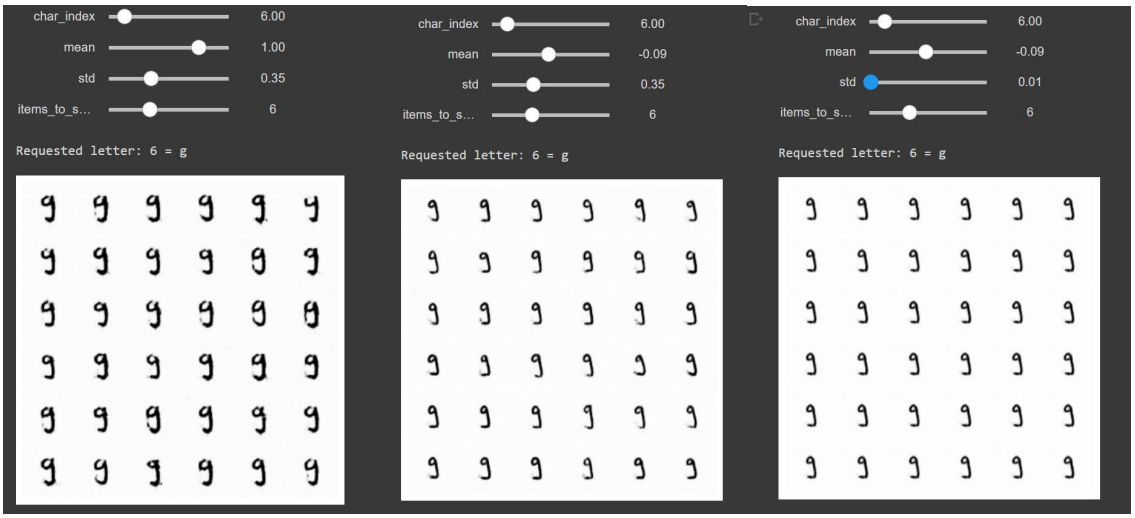
generative model has a more grainy and coarse texture. In the third set of data I've sliced Mitchell's lowercase letters and concatenate it with Terrence's uppercase letters. After training I've found that different datasets exhibit strong personality depending on style and tools that can be easily identified while typing.



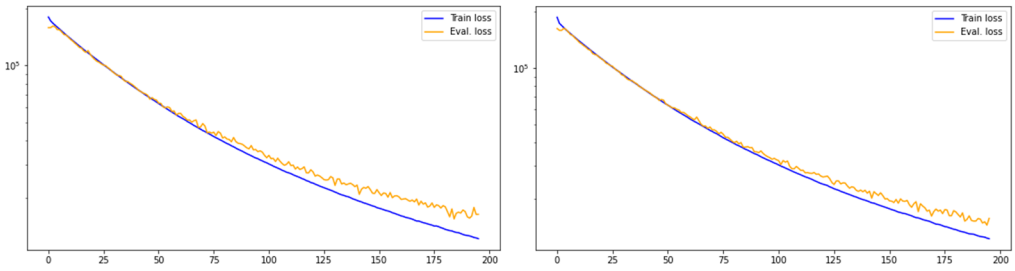
Viewing retrospectively at the optimized sample letters, I found stretching letters like 'y', 'g', 'Q' have less sharp edges than compact letters like 'a', 'w' ,'x', etc. In addition, by comparing the model trained with 10 pages of word and 5 pages, stroke discontinuity is more significant in enclosed letters like 'D' and 'B'.



The latent space is an important step in the post training phase for me to create a desirable font. With a dataset that contains deficient samples, The change of means is important for the typeface output: by randomly sliding back and forth between (-0.3,0.3), desirable renderings can show up spontaneously for every letter.



In terms of data adjustment, the most commonly encountered problem is about matching labels with samples, and reorganizing the sample with correct shape. I've gained valuable experience with understanding the shape of data and the ways to compile data through code tracing. Laterly after the training of all three fonts, I reflect that the performance of training can be improved by reducing the number of Epochs (from 150 to 120 instead of 150 to 200) so that the model would not be overtrained based on the graphics of evaluation and training loss.



The Dashboard

The Interface is a more versatile tool for data selection than notebook, it provides immediate identification upon selecting letters with similarities and stroke thickness. To do these manually through coding might take significantly longer and come up with a less desirable set of selection. It is a very useful tangible window to test various combinations. I was trying to come up with oblique selection with lasso that contains thinner strokes in lowercase letters and increases the thickness based on ascii orders. However there is one quality difference about training with a mix of groups of data rather than a single set: speculatively, due the nature of different individual's writing style, letters with similar thickness may still exhibit contrasting organization. Therefore compared to a regular set training, training on dashboard selection with similar numbers of data (eg, 1810 vs 1872) and with the same epochs(eg, 200) will still result in lower resolution of generative clarity. In this case, latent space becomes an important manual adjustment tool. Unfortunately my dashboard was not responding on the sample texting part, below is the sample lasso selection from the dashboard.

Data Control vs Coding Control

In general I think the data is decisive upon the detailed quality of learning output and the way of constructing neural structures. While coding focuses more on reducing the loss during training and increasing efficiency, they are mutually supporting factors. Since most machine learning models have no control over the dataset itself, one needs to be careful about data selection and filtering to ensure that the dataset is balanced, cohesive in quality at its best capability.

9.2 SecondHand Meta-Tool Handbook

Data Pipeline

Dashboard Folder:

- Download this whole folder and run it on your desktop machine.

Notebooks:

- Initial data processing Notebook: use this notebook to process the annotations
- Data post-processing Notebook: use this notebook to export your data in the correct size and format
- Training notebook: use this to train your model!

Data Collection

In this step, you provide a relatively small data set of your handwriting samples. You should follow these steps:

1. Download the text sample file
2. Print them on paper or import them into a digital device and write pages 0 to 10. **DO NOT** fill pages 11-20.
3. Scan the pages or export them in high-quality (with 3000px width)

Labeling

After scanning the pages, you need to manually label the data. To do this, you should use CVAT online tool. This animation guides you through the steps.
At the end of this process, you will have a .txt file that contains a set of numbers as below:

```
0 0.035064 0.159735 0.039169 0.099119
0 0.071784 0.166474 0.027745 0.085641
...
0 0.339410 0.157750 0.024521 0.071366
0 0.370616 0.158941 0.032284 0.076911
```

The first number shows the label, we ignore this. The next four digits are representing the bounding box that you have drawn. The first two digits are x and y coordination of the center of the box, the last two digits are the width and height. These numbers should be multiplied by the

size of the original image to get converted into integer numbers. The provided notebook will take care of this process.

- **Note**: I strongly suggest you start with one page, then go through all the steps and then come back and repeat the process for the rest of the pages.

Pre/Post Processing data

Using the provided notebook, you can upload your txt file and the scanned image. This notebook reads the text file and crops the image around the boundaries that you have drawn. Finally, it will format and save the image as a NumPy file next to the label files.
You can use these two files and share them with your friends.

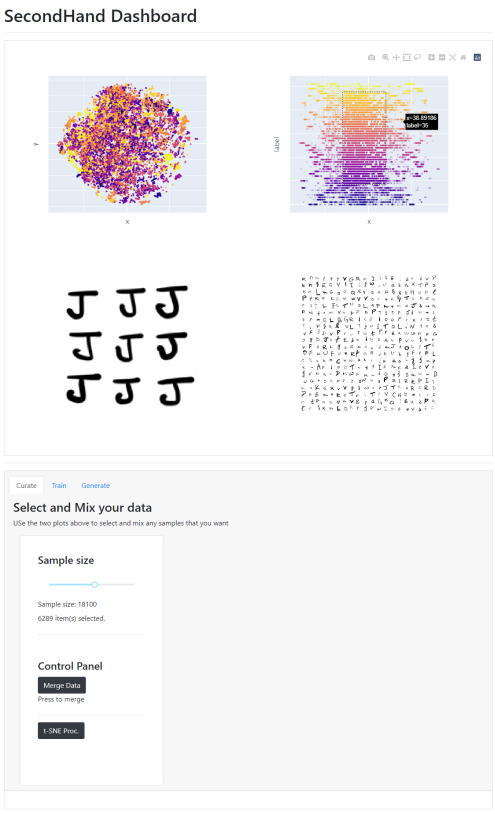
- **Note**: It is very important to double-check your data. It is your responsibility to make sure you share a reliable dataset with your friends.
- Notebooks:
    - Initial data processing Notebook: use this notebook to process the annotations
    - Data post-processing Notebook: use this notebook to export your data in the correct size and format

Shared Database

- Once made your data in the right format, upload it in this folder: Fall_2021_dataset with this naming format: `alphabet_handwriting_64_n_yourInitials.npy`
- Don't upload the label files, it is the same for all.
- I will check this folder three times a day (10:00 am, 6:00 pm, and 11:59 pm).
- Use this notebook to train your model based on your first set of data.

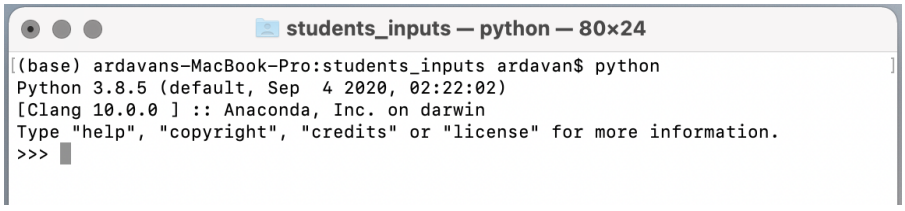
Dashboard

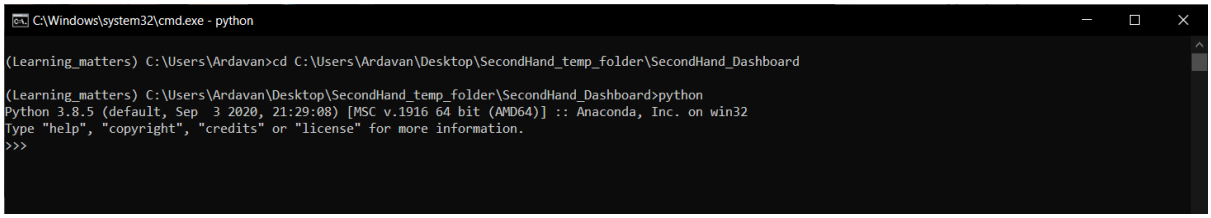
Data viewer is a Dash app that helps you view your big data and make a selection out of it. (please watch the video for more details)



You need to run this app on your local machine, as Colab is not completely compatible with this service. To do so:

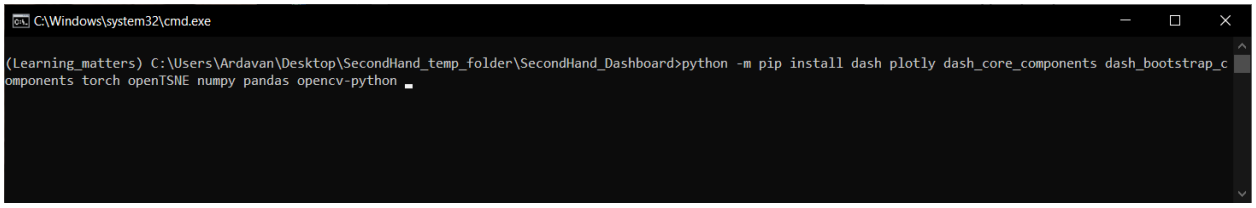
- Open terminal in Mac or Command Prompt in your Windows machines,
- Check and see if you have Python 3 installed (just type python and hit enter, it should run python with a note about its version, here we have version 3.8.5 on Mac and Windows:





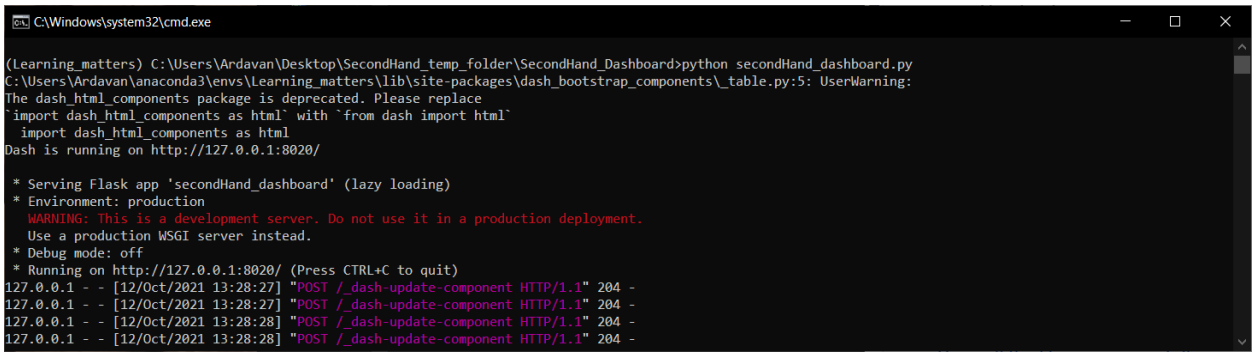
- Now navigate to the folder that you have the zip file unzipped:
    - `cd address/to/folder`

- Then install all the dependencies using this line:
    - `python -m pip install dash plotly dash_core_components dash_bootstrap_components torch openTSNE numpy pandas opencv-python kaleido`



- The script above installs the original PyTorch library with CPU support. If you are planning to use an **Nvidia GPU**, then install PyTorch using this script after you have all other libraries installed:
    - `python -m pip install torch==1.9.1+cu102 torchvision==0.10.1+cu102 torchaudio===0.9.1 -f https://download.pytorch.org/whl/torch_stable.html`

- Once all the libraries are installed, you can run the app:
    - `python secondHand_dashboard.py`



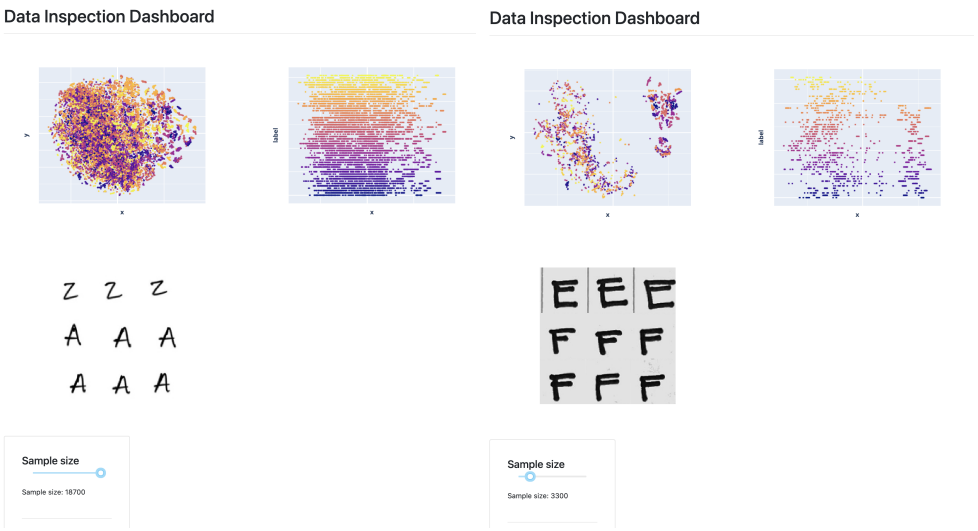
- The app starts running in the background and it is accessible through your browser in this address:
    - `http://127.0.0.1:8020/`

Data Curation Tab:

The goal of this tab is to let you observe the datasets, select a set of desired samples, and train your model based on that.
Note: After each round of training, you can modify your training samples and continue the training process with the new samples.

Interaction models:

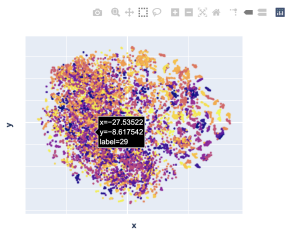
You can select how many samples will be shown in the main plot, the slider at the bottom determines the number of points in the plots. Be advised that this slider cuts the dataset in order. Thus, showing 1000 samples means that it will only pick the first 1000 samples and omit all the other couple of thousand samples in your dataset.



The plot on the left side distributes the samples in 2 dimensions based on their visual appearances. The plot on the right distributes them on x based on their visual properties and on the y-axis based on their labels (lower samples are a,b,c,... and upper samples are …, X, Y, and Z.

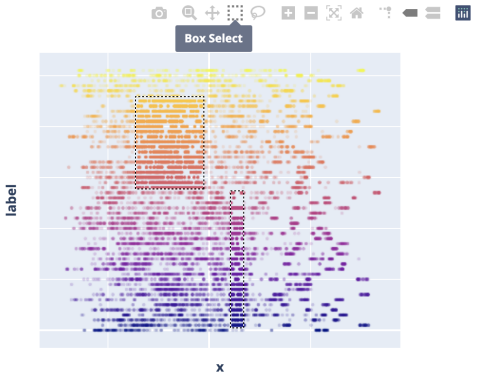
Hovering

The samples are distributed based on the t-SNE algorithm, similar-looking characters are located close to each other. Hovering your mouse over each dot will show that sample and 8 other samples close to it in the dataset (not in the plot).

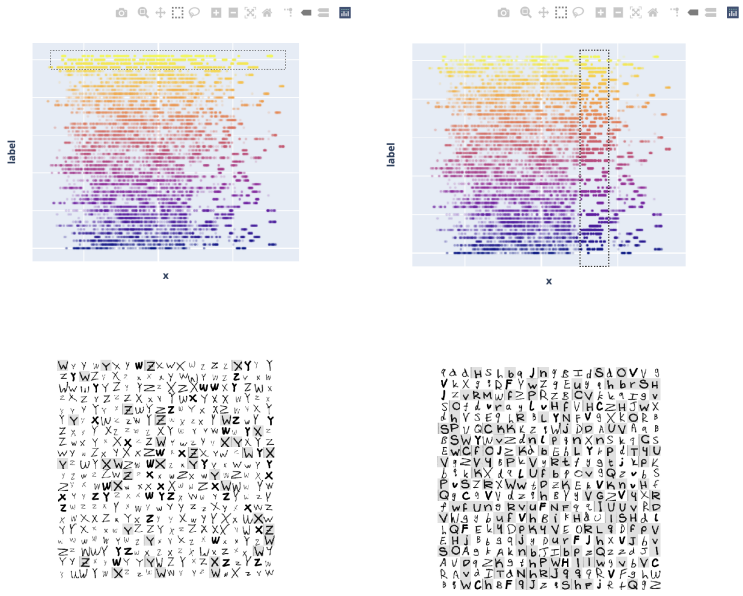


Selection

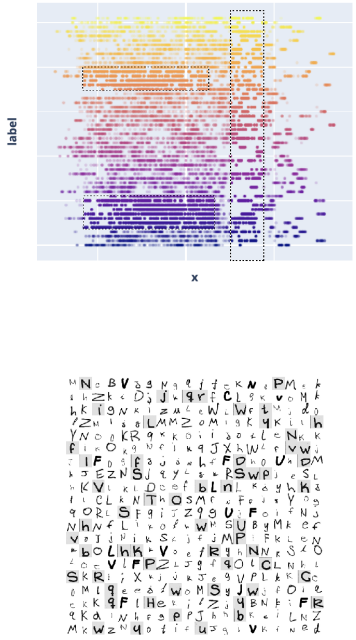
You can use the Box selection and Lasso Select tool to choose what samples you want to use. Play with selection tools on both plots and make a data set to train your model.
As soon as you make a selection, the selection will be saved in a NumPy file `/data/selected_data.npy` and `/data/selected_data_labels.npy` and you can use them to train your model.



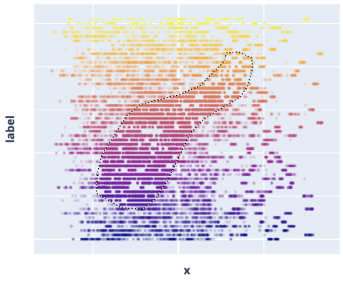
You can make vertical and horizontal selection boxes on the right-side plot to focus on one specific letter or all the letters with similar visual characteristics.



You can also draw multiple boxes by holding shift while selecting boxes:

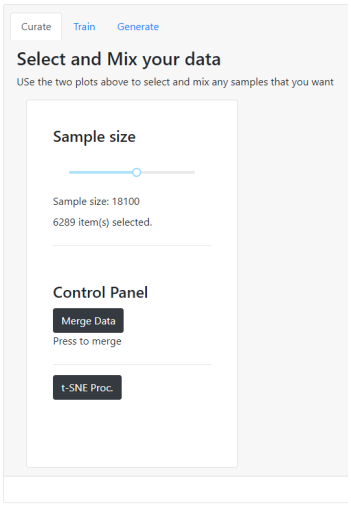


You can also use Lasso select tool to make wild selection too:



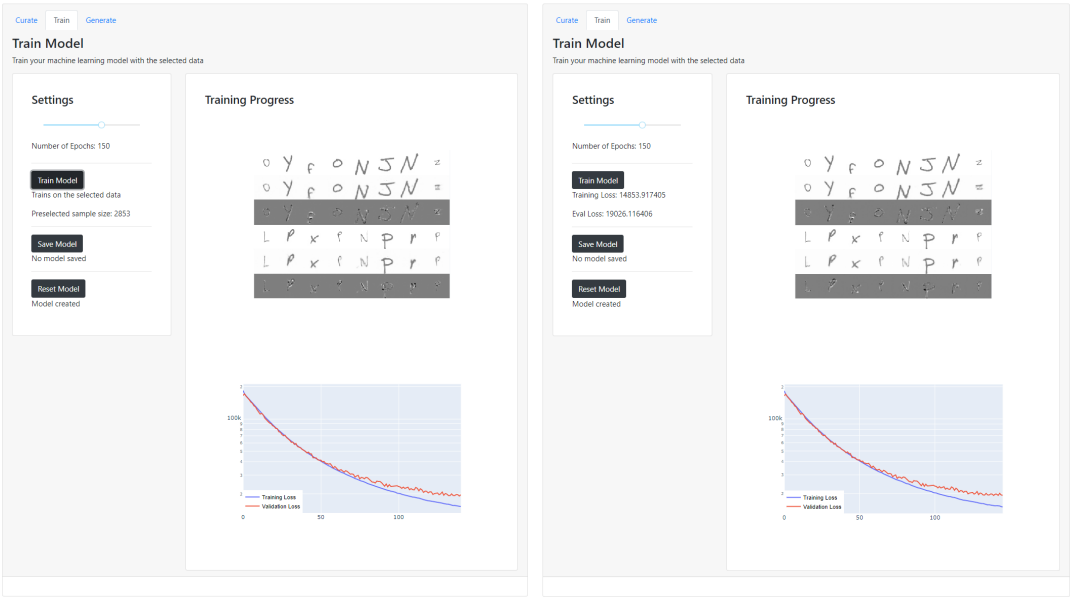
When making a selection, make sure you have samples from all the letters included in your selection, otherwise, your typeface generation will be denied.

Merge/Visualization



The other two buttons can merge the NumPy files in the folder, which should take under one second, and run the t-SNE algorithm on your dataset, which may take a few minutes. Only use these two buttons if you have added some new NumPy files to the data folder.

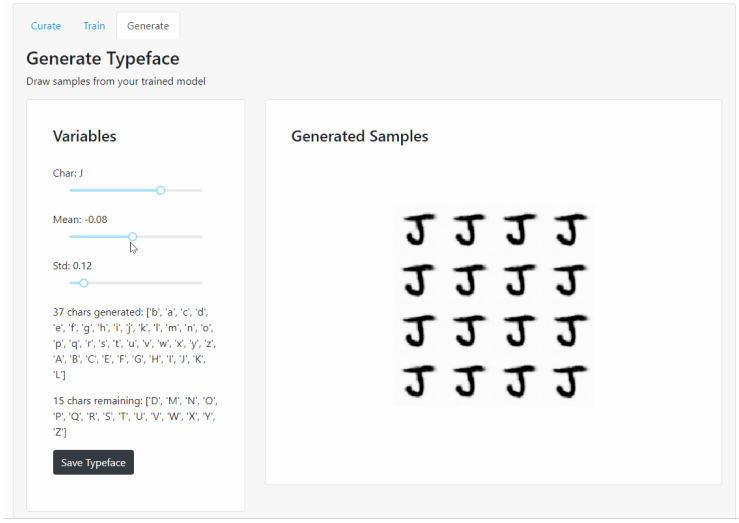
Training Tab



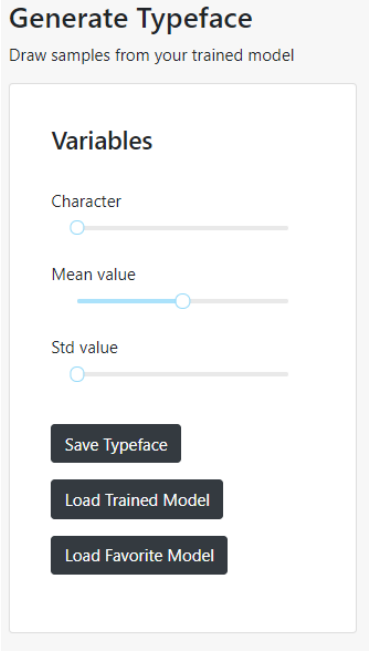
Once you have your data selected, you can start training. Set your desired number of epochs and hit Train Model. The plots provide you with enough clue to monitor the training process (left image). Once the training is over, you will see the loss values (right image) and you can save your model. If you do so, it will overwrite your latest model that has been saved as `./model/trained_model`.

You can repeat this part as much as you want. Each time, you can edit your data selection and continue training with the newly added or removed samples. You can save the model, or reset the model and begin training from the scratch.

Generation Tab



You can use the sliders to generate all the letters and save them as a numpy file. If you just want to practice to a model that you have trained previously, you can hit Load Model and it will load `./model/trained_model`. You can use the Load Favorite Model button to load a model you manually saved as: `./model/favorite_model`. It can be a model that you have trained previously. But I do not recommend loading models that you have trained with a totally different dataset.



9.3 Recruiting Email

Recruiting email for the SecondHand study

Greetings.

My name is [identifiable data retracted] and I am the teaching assistant for “ARC48770 Learning Matters, exploring artificial intelligence in architecture and design,” offered at the School of Architecture at Carnegie Mellon University (CMU).

The Learning Matters team conducting a research study that explores the use of interactive machine learning to address the gap in the current state of creative computing toolmaking that sets apart end-users and from the toolmaking process. The study will be executed in parallel with one of the modules of 48-770, a course which you are a part of.

If you would like to participate in this study, you must be at least 18 years of age, and a student of “ARC48770 Learning Matters, exploring artificial intelligence in architecture and design” course. Your participation is entirely voluntary. If you decide to participate, you will be contributing to a preliminary work in a PhD research project focusing on the issues of situatedness and machine learning, as well as the resulting academic knowledge production on the subject.

There is no cost to you to participate and your participation will be limited to class time and the time you should normally spend on your class assignments. The study will document your assignment for the interactive machine learning module, including handwriting samples, information about the trained models, output samples, and the final class discussions on your experience with interactive machine learning. All these data will be collected as a part of the class routine workflow, and we would like to have your permission to use them for this study.

The risks and discomfort associated are no greater than those ordinarily encountered in daily life or during the regular course activities of ARC48770. You will NOT be asked to provide any personal information other than your name on the consent form, commit any personal time other than what you already spend in ARC48770 class and its regular assignments, perform any special tasks other than what you do in ARC48770 class and its assignments, or travel to a destination other than the school of Architecture, only if you need to pick up a piece of hardware.

The study is not intended to assess or evaluate your performance or quality of your work in ARC48770. Your participation will NOT be shared with the instructors until the grades and evaluations for ARC48770 are finalized and will NOT have any effect on the grading or any other evaluation within the course. Your privacy and data confidentiality will be respected and protected at all times. More information is provided in the attached consent form.

Before deciding whether or not to participate, please read the consent form attached and ask questions about anything you do not understand. If you volunteer to participate, please let me know and I will provide you a pdf copy of the form to sign.

Thank you very much.
Best,
[identifiable data retracted]

9.4 Consent Forms

Consent forms for the SecondHand and the ThirdHand studies

Consent Form for Participation in Research

Study Title: Situated/Interactive Machine Learning for Creative Computing

Principal Investigator: Ardavan Bidgoli, Ph.D. Candidate, Department of Architecture, 5000 Forbes Avenue, College of Fine Arts 201, Pittsburgh, PA 15213, 412.268.2354, abidgoli@andrew.cmu.edu

Faculty Advisor: Daniel Cardoso Llach, associate professor, dcardoso@andrew.cmu.edu

Purpose of this Study

The purpose of this study is to investigate the way creative users can interface with generative machine learning models through interactive data curation to make creative computing tools. The study is intended to explore how this approach can address the gap in the current state of creative computing toolmaking that sets apart the end-users from the toolmaking process.
The study will be organized along with one of the class assignments for "ARC48770 Learning Matters, exploring artificial intelligence in architecture and design," offered at the School of Architecture at Carnegie Mellon University (CMU). The assignment is focused on interactive and situated machine learning to create a handwriting generator tool.

Summary

Through this study, the participants will collaboratively develop a dataset of handwriting samples to train a generative machine learning model. The participants will train their own unique machine learning model, using the data they individually provided in combination with the samples that other participants have shared.

Procedures

The study will follow the procedure listed below:

1. Onboarding the participants:
   In this phase, the class TA, which is an independent colleague and is not part of the study team, will introduce the participant to the research, goals, and the process. Participants will have time to ask their questions.
   The research will be a study of an already-scheduled module of the course that will be conducted for all class members whether they choose to allow their data be used for research or not.

2. Data collection and t raining the machine learning model:
   This phase will be focused on collecting data and interactively training the model by each participant. It will consist of 5-6 sessions, each between 60-90 minutes over a ten-day period (March 23rd, April 1st).
   - During each session, the participants will:
     i. Provide handwriting samples, using physical pen and paper or a touch-enabled device with a stylus,
     ii. Digitalize the samples (if written with physical pen and paper),
     iii. Analyze the samples and feed them to train the machine learning model,

Consent Form for Participation in Research

iv. Inspect the samples created by other users and integrate them in their training process,
v. Analyze the learning trajectory and steering it with manipulating the data set, until a desired handwriting style is achieved.

3. Creation:

After these sessions, the participants will use their handwriting generator tool to create a series of scripts rendered with a desired handwriting style.

None of the activities require the physical presence of participants. For all the above-mentioned activities Zoom platform will be used.

Participant Requirements
Participants must be at least 18 years of age and a student of course ARC48770.

Risks
The risks and discomfort associated with participation in this study are no greater than those ordinarily encountered in daily life or regular remote class activities.

Benefits
There may be no personal benefit from your participation in the study, but the knowledge received may be of value to humanity. Both participating and non-participating students will have the opportunity to learn concepts of interactive and situated machine learning. Both groups will also gain hands-on experience with data collection methods, working collaboratively on making machine learning tools.

Compensation & Costs
There is no compensation for participation in this study.
There will be no cost to you if you participate in this study.

Future Use of Information
The future use of the collected data, with all identifiable information removed, and as anonymized output, will be limited to academic publications or presentations for scientific and educational purposes. We would do this without getting additional informed consent from you (or your legally authorized representative). Sharing of data with other researchers will only be done in such a manner that you will not be identified.

Confidentiality
Your decision to accept or decline participation **will not be shared with the instructors** and the PIs until the assessments, evaluations, and grading for course ARC48770 are finalized.

By participating in the study, you understand and agree that Carnegie Mellon may be required to disclose your consent form, data, and other personally identifiable information as required by law, regulation, subpoena. or court order. Otherwise, your confidentiality will be maintained in the following manner:

Consent Form for Participation in Research

Your data and consent form will be kept separate. Your research data will be stored in a secure location on Carnegie Mellon property. By participating, you understand and agree that the data and information gathered during this study may be used by Carnegie Mellon and published and/or disclosed by Carnegie Mellon to others outside of Carnegie Mellon. However, your name, address, contact information and other direct personal identifiers will not be mentioned in any such publication or dissemination of the research data and/or results by Carnegie Mellon. Note that per regulation all research data must be kept for a minimum of 3 years.

Effects on your class experience and evaluation:
This study is by no means affecting the quality of your class experience. It has no effect on the grading or any other evaluation within the course.

Optional Permission
I understand that the researchers may want to use any of the video, audio, handwriting samples, and final image outputs for illustrative reasons in presentations of this work, as printed or digital publication, publishing still images, slide shows, video clips, or raw image-based data on online platforms for scientific or educational purposes. I give my permission to do so.

Please initial here: _____YES _____NO

Rights
Your participation is voluntary. You are free to stop your participation at any point. Refusal to participate or withdrawal of your consent or discontinued participation in the study **will not result in any penalty or loss** of benefits or rights to which you might otherwise be entitled. The Principal Investigator may at his/her discretion remove you from the study for any a number of reasons. In such an event, you will not suffer any penalty or loss of benefits or rights which you might otherwise be entitled.

Right to Ask Questions & Contact Information
If you have any questions about this study, you should feel free to ask them now. If you have questions later, desire additional information, or wish to withdraw your participation please contact the Principal Investigator by mail, phone, or e-mail in accordance with the contact information listed on the first page of this consent.

If you have questions pertaining to your rights as a research participant; or to report concerns to this study, you should contact the Office of Research Integrity and Compliance at Carnegie Mellon University. Email: irb-review@andrew.cmu.edu . Phone: 412-268-1901 or 412-268-5460.

Voluntary Consent
By signing below, you agree that the above information has been explained to you and all your current questions have been answered. You are encouraged ask questions about any aspect of this research

Consent Form for Participation in Research

study during the course of the study and in the future. By signing this form, you agree to participate in this research study. A copy of the consent form will be given to you.

PRINT PARTICIPANT'S NAME

PARTICIPANT SIGNATURE DATE

I certify that I have explained the nature and purpose of this research study to the above individual and I have discussed the potential benefits and possible risks of participation in the study. Any questions the individual has about this study have been answered and any future questions will be answered as they arise.

SIGNATURE OF PERSON OBTAINING CONSENT DATE

Carnegie Mellon University

Consent Form for Participation in Research

Study Title: Situated/Interactive Machine Learning for Creative Computing

Principal Investigator: Ardavan Bidgoli, Ph.D. Candidate, Department of Architecture, 5000 Forbes Avenue, College of Fine Arts 201, Pittsburgh, PA 15213, 412.268.2354, abidgoli@andrew.cmu.edu

Faculty Advisor: Daniel Cardoso Llach, associate professor, dcardoso@andrew.cmu.edu

Purpose of this Study

The purpose of this study is to investigate how creative users can interface with generative machine learning models through interactive data curation to make creative computing tools. The study is intended to explore how this approach can address the gap in the current state of creative computing toolmaking that sets apart the end-users from the toolmaking process.

The study will be organized as a collaboration between the PI and an expert Santur—a traditional Persian instrument—player. The study is focused on interactive and situated machine learning to create a tool to facilitate playing Santur.

Summary

Through this study, the expert musician and the PI will collaboratively develop a dataset of Santur playing samples to train a generative machine learning model. The expert musician will train its own unique machine learning model, using the data she individually provided and demonstrate its performance in a demo session.

Procedures

The study will follow the procedure listed below:

- Onboarding
  - The artist will be introduced to the study, scope, procedure, and goals
- Adaptation
  - Through a series of unstructured interviews, the artist and PI will decide on the process of adapting the research for the specific music instrument, method of playing the instrument, the personal preferences of the artist, and the scope of the demo/test session
- Data collection sessions
  - The artist provides samples of Santur performance, the performances will be recorded through:
    - motion capture system
    - sound recording
    - video shoots for documentations and calibrations
  - Unstructured interview

Consent Form for Participation in Research

- Training the machine learning models
  - The artist, with proper guidance by the PI, will use the data collected in the previous sessions to train a machine learning model,
    - Machine learning model's performance data
    - Time spent on the process
    - Misc. performance data
  - Unstructured interview

- Test/demo sessions
  - The artist will use the trained model in a test/demo session
    - Based on the adaptation phase, this demo may be conducted digitally or executed on a robotic arm.
    - In the latter case, the robot will be operated solely by the PI, observing the safety regulations. The artist will be provided with a proper safety briefing

Participant Requirements
The participant must be at least 18 years of age.

Risks
The risks and discomfort associated with participation in this study are no greater than those ordinarily encountered in daily life or regular musical performance activities.

Benefits
There may be no personal benefit from your participation in the study, but the knowledge received may be of value to humanity. The artist will have the opportunity to learn concepts of interactive and situated machine learning. She will also gain hands-on experience with data collection methods and work on making machine learning-based toolmaking processes.

Compensation & Costs
There is no compensation for participation in this study.
There will be no cost to you if you participate in this study.

Future Use of Information
The future use of the collected data, as anonymized output, will be limited to academic publications or presentations for scientific and educational purposes. We would do this without getting additional informed consent from you (or your legally authorized representative). Sharing of data with other researchers will only be done in such a manner that you will not be identified.

Confidentiality
By participating in the study, you understand and agree that Carnegie Mellon may be required to disclose your consent form, data, and other personally identifiable information as required by law,

Consent Form for Participation in Research

regulation, subpoena, or court order. Otherwise, your confidentiality will be maintained in the following manner:

Your data and consent form will be kept separate. Your research data will be stored in a secure location on Carnegie Mellon's property and Carnegie Mellon-approved cloud storage system. By participating, you understand and agree that the data and information gathered during this study may be used by Carnegie Mellon and published and/or disclosed by Carnegie Mellon to others outside of Carnegie Mellon. However, your name, address, contact information, and other direct personal identifiers will not be mentioned in any such publication or dissemination of the research data and/or results by Carnegie Mellon. Note that per regulation, all research data must be kept for a minimum of 3 years. Note that you can provide Carnegie Mellon the option to mention your name through the optional permissions.

Optional Permissions

1. I understand that the researchers may mention my name in publications or dissemination of the research data and/or results by Carnegie Mellon. I give my permission to do so.

   Please initial here: MN YES _____NO

2. I understand that the researchers may want to use any of the video footage, audio samples, and final image outputs for illustrative reasons in presentations of this work, as printed or digital publication, publishing still images, slide shows, video clips, sound clips, or raw image-based data on online platforms for scientific or educational purposes. I give my permission to do so.

   Please initial here: MN YES _____NO

3. I understand that the researchers may want to play audio and video recordings of my performances for illustrative reasons in presentations of this work. In such cases, I would like to be credited as the performer.

   Please initial here: MN YES _____NO

Rights

Your participation is voluntary. You are free to stop your participation at any point. Refusal to participate or withdrawal of your consent or discontinue participation in the study **will not result in any penalty or loss** of benefits or rights to which you might otherwise be entitled. The Principal Investigator

Consent Form for Participation in Research

may at his/her discretion remove you from the study for any a number of reasons. In such an event, you will not suffer any penalty or loss of benefits or rights which you might otherwise be entitled.

Right to Ask Questions & Contact Information
If you have any questions about this study, you should feel free to ask them now. If you have questions later, desire additional information, or wish to withdraw your participation please contact the Principal Investigator by mail, phone, or e-mail in accordance with the contact information listed on the first page of this consent.

If you have questions pertaining to your rights as a research participant; or to report concerns to this study, you should contact the Office of Research Integrity and Compliance at Carnegie Mellon University. Email: irb-review@andrew.cmu.edu . Phone: 412-268-1901 or 412-268-5460.

Voluntary Consent
By signing below, you agree that the above information has been explained to you and all your current questions have been answered. You are encouraged ask questions about any aspect of this research study during the course of the study and in the future. By signing this form, you agree to participate in this research study. A copy of the consent form will be given to you.

Mahtab Nadalian
PRINT PARTICIPANT'S NAME

PARTICIPANT SIGNATURE

8/18/2021
DATE

I certify that I have explained the nature and purpose of this research study to the above individual and I have discussed the potential benefits and possible risks of participation in the study. Any questions the individual has about this study have been answered and any future questions will be answered as they arise

SIGN TAINING CONSENT

Aug. 18th, 2021
DATE