

Automatic Analysis of Facial Actions: Learning from Transductive, Supervised and Unsupervised Frameworks

Wen-Sheng Chu
CMU-RI-TR-17-01

January 2017

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Fernando De la Torre, Co-chair
Jeffrey F. Cohn, Co-chair
Simon Lucey
Deva Ramanan
Vladimir Pavlovic, Rutgers University

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Robotics.*



Copyright © 2017 Wen-Sheng Chu

Keywords: Automated measurement, facial expression analysis, facial action unit (AU) detection, transfer learning, domain adaptation, importance re-weighting, Support Vector Machine (SVM), personalization, bi-convex optimization, deep learning, long short-term memory (LSTM), spatio-temporal fusion, multi-label sampling, global optimization, Branch-and-Bound (B&B) algorithm, common event discovery, event detection, video search, video analysis.

For my parents and Lisa

Abstract

Automatic analysis of facial actions (AFA) can reveal a person’s emotion, intention, and physical state, and make possible a wide range of applications. To enable reliable, valid, and efficient AFA, this thesis investigates automatic analysis of facial actions through *transductive*, *supervised* and *unsupervised* learning.

Supervised learning for AFA is challenging, in part, because of individual differences among persons in face shape and appearance and variation in video acquisition and context. To improve generalizability across persons, we propose a transductive framework, Selective Transfer Machine (STM), which personalizes generic classifiers through joint sample reweighting and classifier learning. By personalizing classifiers, STM offers improved generalization to unknown persons. As an extension, we develop a variant of STM for use when partially labeled data are available.

Additional challenges for supervised learning include learning an optimal representation for classification, variation in base rates of action units (AUs), correlation between AUs and temporal consistency. While these challenges could be partly accommodated with an SVM or STM, a more powerful alternative is afforded by an end-to-end supervised framework (*i.e.*, deep learning). We propose a convolutional network with long short-term memory (LSTM) and multi-label sampling strategies. We compared SVM, STM and deep learning approaches with respect to AU occurrence and intensity in and between BP4D+ [282] and GFT [93] databases, which consist of around 0.6 million annotated frames.

Annotated video is not always possible or desirable. We introduce an unsupervised Branch-and-Bound framework to discover correlated facial actions in un-annotated video. We term this approach Common Event Discovery (CED). We evaluate CED in video and motion capture data. CED achieved moderate convergence with supervised approaches and enabled discovery of novel patterns occult to supervised approaches.

Acknowledgments

This dissertation is in collaboration with my advisors, Fernando De la Torre and Jeffrey F. Cohn, to whom I give my first and foremost gratitude for their guidance and support throughout my doctoral study. They have been my mentor and my rock, teach, inspire, and give me the freedom to explore new ideas and the opportunity to mentor three visiting PhD students. I have enjoyed our time and work we accomplished, and I appreciate all you have done for me.

I would also like to thank Simon Lucey, Deva Ramanan and Vladimir Pavlovic for their interests in my work and serving as my committee members. I appreciate your time and invaluable advice and feedback regarding this work.

I want to thank Alyosha Efros and Leonid Sigal for giving me a wonderful book as a recognition of the best project in your Learning-based Computer Vision course, which lays me a strong foundation of critical thinking and confidence in being a computer vision researcher. I am grateful for Martial Hebert to have me as a TA in your popular Computer Vision course, from which I learned to provide intuitive explanations of complex concepts and design systematic, progressive teaching materials. I want to thank Srinivasa Narasimhan for the opportunities to help host ICCP'10 at CMU and to serve as a committee member at the MSCV program of Robotics, where I witnessed your leadership in decision making and invincible sense of humor. I also want to thank Yale Song for referring me to Yahoo! Research, and Josh Susskind for recruiting me to Apple. I am excited to start the next journey.

I have had a great time working with numerous talents, including the IntraFace team: Xuehan Xiong, Francisco Vicente, Marc Estruch, Ramon Sanabria, and formerly Ferran Altarriba, Santiago Ortega, Javier Lopez, Xavier Perex, Tomas Simon, and Zengyin Zhang; the Human Sensing Lab at CMU: Feng Zhou, Jayakorn Vongkulbhisal, Ada Zhang, Calvin Murdock, Minh Hoai, Dong Huang; the Affect Analysis Group at UPitt: Jeff Girard, Laszlo Jeni, Zakia Hammal and Nicki Siverling; my awesome mentees: Jiabei Zeng, Kaili Zhao, and Xiaoyu Ding. I have been also fortunate to work with Yale Song, Alejandro Jaimes, and Amanda Stent at Yahoo! Research, and Professor Daniel S. Messinger at University of Miami. Your creativity and dedication have been a constant inspiration through my time. Thank you for challenging my ideas and providing stimulating discussions along my research and my life.

Last but not least, I want to thank my parents and my wife for years of unconditional, endless love and support. You make this moment and my dream come true.

Contents

List of Publications	xviii
Notation	xx
1 Introduction	1
1.1 Motivation	2
1.2 Learning Frameworks	3
1.3 Contributions and Organization of This Dissertation	4
2 Advances and Challenges	7
2.1 Face Detection and Registration	9
2.2 Feature Extraction	9
2.3 Modeling	10
2.3.1 Static Modeling	11
2.3.2 Temporal Modeling	15
2.3.3 Ensemble Learning	16
2.3.4 End-to-end Learning	17
2.4 Source of Errors	17
2.5 Alternative Methods for Analyzing Human Behavior	18
2.5.1 Supervised Analysis of Behavior	18
2.5.2 Unsupervised Analysis of Behavior	18
2.6 Evaluation Metrics	19
3 A Transductive Framework for Personalized Facial Expression Analysis	21
3.1 Persona-Specific Biases for Facial Expression Analysis	22
3.2 Selective Transfer Machine (STM)	24
3.3 Optimization for STM	26
3.4 Theoretical Rationale	29
3.4.1 Properties of STM	29
3.4.2 Algorithm	30
3.5 STM with Labeled Target Data (L-STM)	31
3.6 Discussion on Related Work	32

3.7	Experiments	33
3.7.1	Dataset Description	33
3.7.2	Settings	35
3.7.3	Action Unit (AU) Detection	35
3.7.4	Holistic Expression Detection	40
3.7.5	Analysis	41
3.7.6	Discussion	44
3.8	Summary	44
4	An End-to-End Supervised Framework for Facial Action Unit Detection	47
4.1	More Aspects for Facial Action Unit (AU) Detection	49
4.2	The Hybrid Network for Multi-label Facial AU Detection	50
4.2.1	Learning spatial representation	50
4.2.2	Temporal modeling with stacked LSTMs	51
4.2.3	Frame-based spatiotemporal fusion	52
4.3	Evaluations	53
4.3.1	Datasets	53
4.3.2	Settings	53
4.3.3	Evaluation of learned representation	54
4.3.4	Evaluation of detection performance	56
4.3.5	Visualization of learned AU models	59
4.4	Multi-label sampling strategies to address class imbalance	60
4.4.1	Multi-label stratification	64
4.4.2	Multi-label minority oversampling majority undersampling (MOMU)	65
4.4.3	Evaluation of different multi-label sampling strategies	66
4.4.4	Comparisons among generic, personalized and deep models	71
4.5	Summary	74
5	An Unsupervised Framework for Common Event Discovery in Human Interaction	77
5.1	Common Events in Human Interaction	78
5.2	A Branch-and-Bound Framework for Common Event Discovery	79
5.2.1	Representation of time series	80
5.2.2	Problem formulation	81
5.2.3	Optimization by Branch-and-Bound (B&B)	81
5.3	Construction of bounding functions	82
5.4	Searching Scenarios	86
5.4.1	Synchrony discovery (SD)	87
5.4.2	Video search (VS)	88
5.4.3	Segment-based event detection (ED)	88
5.4.4	Comparisons with related work	89
5.5	Extensions to the B&B framework	91
5.6	Experiments	93
5.6.1	Common event discovery	93
5.6.2	Synchrony discovery	97

5.6.3	Event detection	100
5.7	Summary	103
6	Conclusion and Future Work	105
6.1	Summary of Contributions	106
6.2	Directions for Future Work	107
	Appendices	111
A	STM Derivation	113
A.1	Linear Penalized SVMs	113
A.1.1	Quadratic loss	113
A.1.2	Huber loss	114
A.2	Nonlinear Penalized SVMs	115
A.2.1	Quadratic loss	116
A.2.2	Huber loss	116
A.3	Adapt with Target Labels	117
A.4	Cross-subject experiment on GEMEP-FERA dataset	118
B	Evaluate CNNs on Baby-FACS	121
B.1	Miami Modeling Dataset (6-month)	121
B.2	CLOCK Dataset	122
C	CED Derivation	125
C.1	Bound derivations for symmetrized KL divergence	125
	Bibliography	127

List of Figures

1.1	Facial expression is one of the most powerful channel for non-verbal communication. Such actions of the face can convey emotions of an individual to observers, and serve as a primary means of exchanging social information between humans. .	1
1.2	Illustration of (a) 12 common action units (AUs), and (b) holistic facial expressions: (from left to right) happiness, sadness, anger, fear, surprise, disgust, contempt, and embarrassment. (figure credits to [49])	2
2.1	Illustration of contemporary challenges in automated analysis of facial actions (AFA): (a) Learning spatial feature presentation, (b) learning temporal consistency, (c) learning / employing AU correlation, and (d) learning active patches / regions. .	7
2.2	Conventional pipeline for automated analysis of facial actions (AFA).	8
2.3	Illustrations of three recent models that the author and colleagues pursued to address AFA: (a) Cascade of Tasks (CoT) [65, 66], (b) Confidence Preserving Machine (CPM) [277, 278], and (c) Joint Patch and Multi-label Learning (JPML) [285, 286]. Note that these studies are excluded from this thesis. We refer interested readers to references for more details.	11
3.1	An illustration of the proposed transductive approach, Selective Transfer Machine (STM) : (a) 2D PCA projection of positive (squares) and negative (circles) samples for a given AU (in this case AU12) for 3 subjects. An ideal classifier separates AU 12 nearly perfectly for each subject. (b) A generic classifier trained on all 3 subjects generalizes poorly to a new person (<i>i.e.</i> , test subject) due to individual differences between the 3-subject training set and the new person. STM personalizes a generic classifier and reliably separates an AU for a new subject. . .	21
3.2	Visualization of samples from the RU-FACS dataset [10] in 3D eigenspace: colors/markers indicate different (a) positive/negative classes, and (b) subjects. (c) shows an example that illustrates individual differences between Julia Roberts and Mona Lisa. (best viewed in color).	22

3.3	Fitting a line to a quadratic function using KMM and other re-weighting methods. The larger size (more red) of training data, the more weight KMM adopted. As can be observed, KMM puts higher weights in the training samples closer to the test ones. Compared to standard OLS or WOLS, KMM allows to better approximation for the test data.	25
3.4	Comparisons of a generic SVM, personalized STM, and an ideal classifier for synthetic data. The top-left figure shows the convergence curve of STM objective in 12 iterations. Iterations #1, #4, #8, #12 with training/test accuracy (Tr% and Te%) show corresponding hyperplanes at each iteration. Grey (shaded) dots denote training data, and white (unshaded) dots denote test data. Circles (squares) denote positive (negative) classes. Note that it #1 indicates the result of KMM [103]. STM improves generic SVM as early as the first iteration, and then converges toward the ideal hyperplane.	27
3.5	(Left to right) Convergence curve of a multi-class STM [279], separation hyperplane at iterations #1, #2, and #4, and the ideal classifier (SVM trained on the test samples). Circles (○) and squares (□) indicate training and test samples, respectively. Te% and Tr% indicate the accuracy on test and training data. As can be seen, as iteration proceeds, the STM hyperplane approaches the ideal hyperplane, which perfectly separates the test samples.	28
3.6	Loss functions used in this study: (a) L^1 and L^2 loss, and (b) Huber loss (a differentiable surrogate).	29
3.7	Comparison of different methods on the RU-FACS dataset. Light yellow (dark green) indicates AU 12 presense (absense) of Subject 12. The numbers in the parentheses are F1 scores. Two misclassified frames of STM were chosen and fed into L-STM with correct labels.	31
3.8	Example images from four datasets studied in this chapter: (a) CK+ [161], (b) GEMEP-FERA [248], (c) RU-FACS [10], and GFT[209] datasets.	34
3.9	Selection percentage of STM for different subjects on (a) initialization and (b) convergence step. Each row sums to one and stands for a test subject.	36
3.10	Analysis experiments: (a)–(b) Objective and variable differences between iterations with initialization w_0 (STM _w) and s_0 (STM _s), respectively. (c) Performance versus parameter choices. (d) Per-subject F1 score v.s. # training subjects.	41
3.11	Performance versus domain size: The averaged and standard deviation of F1 score on (a) RU-FACS. (b) and (c) show the F1 scores on the GFT dataset before and after removing the <i>outlier</i> subjects, respectively. (more descriptions in text)	43
4.1	An overview of the proposed hybrid deep learning framework. The proposed network first possesses strengths of CNNs and LSTMs to model and utilize both spatial and temporal cues. Then, we employ a fusion network to combine both cues to produce frame-based prediction for multiple AUs.	47

4.2	The structure of the proposed hybrid network: (a) Folded illustration of Fig. 4.1, showing 3 components of learning spatially representation, temporal modeling, and spatiotemporal fusion, (b) 8-layer CNN architecture for multi-label prediction, and (c) the schematic of an LSTM block. (d)-(e) conv1 kernel visualization on ImageNet [137] and GFT datasets, respectively. As can be seen, filters learned on faces contain less color blob detectors, suggesting color information is less useful for AU detection.	50
4.3	A visualization of t-SNE embedding using SIFT, VGG face descriptor [190] and fc7 features on the BP4D dataset by coloring samples in term of AU12 (top row) or subjects (bottom row). The clustering effect in SIFT features and VGG face descriptors reveal that face images encode not only information about facial AUs, but more on identities of subjects. The learned fc7 features are optimized for multi-label AU classification, and thus reduce such influence.	55
4.4	Analysis of subject-invariance on two datasets: BP4D (top row) and GFT (bottom row). Four representative features, shape, Gabor, SIFT and fc7, were compared (details in text). For display purpose, a computed divergence d is normalized by $\log(d) \times 1e6$	56
4.5	Synthetically generated images to maximally activate individual AU neurons in the fc8 layer of CNN, trained on GFT [50], showing what each AU model “wants to see”. The learned models show high agreement on attributes described in FACS [77]. (best view electronically)	59
4.6	Distributions of AU base rates in two of the largest spontaneous datasets used in this study: (a) GFT [50] and (b) BP4D+ [283]. (c) shows the exact base rate of individual AUs of each dataset. Base rate is defined as the frequency of a particular AU occurring in video frames of the entire dataset. Note that we only count the frames that can be validly face tracked and annotated completely with 12 AUs. . .	60
4.7	Distributions of AU classes in each mini-batch using different sampling strategies: (top) random sampling, (middle) multi-label stratification, (bottom) MOMU sampling. As can be seen in random sampling, the number of AU presence <i>between</i> and <i>within</i> batches are dramatically different. (see text for details)	63
4.8	An illustration of random cropping (<i>i.e.</i> , translation, rotation, scale) as standard data augmentation for training deep networks. (image credit from [42, 43])	66
4.9	Comparison of training performance on the GFT dataset in terms of F1-score (y-axis) vs the number of iterations (x-axis) over different sampling strategies: (Red) random sampling, (Green) multi-label stratification, (Blue) multi-label MOMU. As can be observed, for conventional random sampling and multi-label stratification, the performance of minority AUs, such as AUs 4 ($BR_4 = 3.4\%$) and 15 ($BR_{15} = 8.8\%$), remains rather low even after training phase with 8000 iterations. (the curve is higher better)	67
4.10	Improved points of MOMU over random sampling in both within-dataset and between-dataset scenarios for GFT [50] and BP4D+ [283] datasets. Results in AUC and F1 suggest that improvements are more consistent in BP4D+ than in GFT due to the more dramatic AU imbalance in the BP4D+ dataset (as illustrated in Fig. 4.6). . . .	71

5.1	An illustration of Common Event Discovery (CED) in human interaction: Given two videos, how can one efficiently discover common events in an unsupervised manner? This example illustrates the discovered common events, <i>Kissing</i> and <i>Handshaking</i> , shared between two videos. Note that the discovered events are of different lengths.	77
5.2	Humans are inherently social. Friends, romantic partners, families or coworkers tentatively make the same facial actions when they are engaged. We term these actions as <i>common events</i> , and propose an algorithm to discover them without the need of annotations. We note that the last column is not human, but just for illustration.	78
5.3	An example of CED on two 1-D time series: (a) An illustration of our notation (see Sec. 5.2.3). (b) Searching intervals at iterations (<i>it</i>) #1, #300 and #1181 over sequences \mathbf{S}^1 and \mathbf{S}^2 . Commonalities $\mathbf{S}^1[b_1, e_1]$ and $\mathbf{S}^2[b_2, e_2]$ are discovered at convergence (#1181). (c) Convergence curve w.r.t. bounding value and # <i>it</i> . (d) Histograms of the discovered commonalities. In this example, a naive sliding window approach needs more than 5 million evaluations, while the proposed B&B method converges at iteration 1181 using $\ell = 20$	80
5.4	Searching scenarios readily applicable to the proposed B&B framework: (a) Common event discovery (CED), (b) synchrony discovery (SD), (c) video search (VS), and (d) supervised segment-based event detection (ED). Green area indicates the search space; an orange box indicates a candidate solution \mathbf{r} . (see Sec. 5.4 for details)	86
5.5	An example of SD on two 1-D time series using $\ell = 13$ and $T = 5$: (a) Top 3 discovered synchronies at different iterations; exhaustive search takes 39151 iterations. (b) The convergence curve w.r.t. bounding value and #iter. (c)~(e) Discovered synchronies and their histograms, where blue and green bars indicate the segment features ϕ^{obs} and ϕ^{int} , respectively. ϕ^{int} is 10X magnified for display purpose. The ℓ_1 distances between the three histogram pairs are 6.3e-8, 1.5e-7, and 5.8e-2, respectively.	88
5.6	Illustration of extensions to SD: (a) pruning rules applied to multiple-commonality discovery, (b) SD with warm start, and (c) SD with parallelism.	92
5.7	Results on discovering common facial actions: (a) Facial features extracted from the tracked points. (b) An example of common discovered facial events (indicated by dashed-line rectangles). (c)(d) Accuracy evaluation on precision-recall and average precision (AP).	93
5.8	Efficiency evaluation between CED and alternative sliding window (SW) approach. Top: Parameter settings [142, 252]: size-ratio (SR), stepsize (SS), and aspect ratios (AR). Middle: Histogram of ratio of #evaluation: $\log \frac{n^{\text{CED}}}{n^{\text{SW}_i}}$. Red vertical lines indicate the average. Light green bars show CED performs less evaluations than SW; dark blue bars represent the opposite. Bottom: Histogram of difference between resulting commonality measure: $f_{\ell_1}(\mathbf{r}^{\text{SW}_i}) - f_{\ell_1}(\mathbf{r}^{\text{CED}})$	94
5.9	(a) Top six discovered common motions, indexed by numbers. Note that the shaded star (number 6) indicates an incorrect discovery that matched <i>walk</i> and <i>kick</i> . (b)(c) Precision-recall and average precision on ℓ_1 distance. (d) Precision-recall on χ^2 distance.	95

5.10	Analysis on top 10 discovered dyadic and triadic synchronies of the GFT dataset. SW denoted with \star indicates the optimal windows discovered, and without \star indicates the average and standard deviation over all visited windows.	96
5.11	Top 10 discovered synchronies from groups 128, 071, 094, 113 and 049 in the GFT dataset. Each column indicates a discovered synchrony and its frame number. The SD algorithm correctly matched the states of <i>smiling</i> , <i>talking</i> and <i>silent</i>	97
5.12	Discovered synchronies from 6 groups of parent-infant interaction. Each column indicates a discovery and its #frame.	98
5.13	Speedup of SD against sliding window (SW) on CMU-Mocap. All 7 pairs of sequences from subject 86 were evaluated. The speedup was computed as the relative number of evaluations $N^{\text{SW}}/N^{\text{USD}}$ using ℓ_1 , χ^2 and symmetrized KL divergence.	100
5.14	Discovered synchronies on 7 pairs of Subject 86 in CMU-Mocap dataset. Each pair is annotated with ground truth (colorful bars, each represents an action), and synchronies discovered by our method (shaded numbers). Synchronies with disagreed action labels are visualized.	101
5.15	Comparison between Dynamic Programming (DP) and ED in terms of (a) F1-even and (b) computation time (sec).	102
5.16	Comparison between ED and alternative approaches in terms of: (a) F1-event v.s. time, (b) F1 v.s. time, and (c) comparison between ground truth and detection results on 3 subjects. Light yellow and dark green indicate activation and deactivation of AU12, respectively.	103
6.1	An illustration that summarizes three learning frameworks studied in this thesis and related issues addressed in each framework.	105
B.1	Illustration of the interaction between a mother and her infant, as well as some face tracking results on the face area (red boxes) and detected head pose (green pyramids). (image credit from [109])	121

List of Tables

1.1	List of companies on automated analysis on facial actions	3
2.1	Conventional features for AFA	9
2.2	Conventional classification models for AFA	10
2.3	Summary of learning paradigms for attacking different challenges in AFA	11
3.1	Compare STM with related transductive transfer learning methods (in terms of their consideration of different learning factors)	32
3.2	Content of different datasets	33
3.3	Within-subject AU detection with STM and PS classifiers	35
3.4	Cross-subject AU detection on RU-FACS dataset. “SA (NN SVM)” indicates SA with NN and SVM, respectively.	37
3.5	Cross-subject AU detection on CK+ dataset	38
3.6	Cross-dataset AU detection: (a) RU-FACS→GEMEP-FERA, and (b) GFT→RU-FACS (“A→B” represents for training on dataset A and test on B).	39
3.7	Expression detection with AUC on (a) CK+ and (b) GEMEP-FERA	40
4.1	Limitations in standard AFA methods and related solutions presented in this chapter	48
4.2	F1-frame on GFT dataset [50]	57
4.3	F1-frame metrics on BP4D dataset [282]	58
4.4	Performance evaluation of different sampling strategies in terms of within-dataset (top) and between-dataset (bottom) scenarios in the GFT dataset [50]: random sampling, multi-label stratification, and multi-label MOMU sampling. The evaluation metrics are S: Kappa, AUC: Area Under the ROC Curve, PA: positive agreement or F1, NA: negative agreement.	68
4.5	Performance evaluation of different sampling strategies in terms of within-dataset (top) and between-dataset (bottom) scenarios in the BP4D+ dataset [283]: random sampling, multi-label stratification, and multi-label MOMU sampling. The evaluation metrics are S: Kappa, AUC: Area Under the ROC Curve, PA: positive agreement or F1, NA: negative agreement.	69
4.6	A summary of averaged performance of 12 AUs among alternative models for within-dataset (top) and between-dataset (bottom) scenarios using GFT [93] and BP4D+ [283] datasets.	72

4.7	Performance evaluation of SVM in terms of within-dataset (top) and between-dataset (bottom) scenarios in the GFT [50] and BP4D+ [283] datasets. For example, the between-dataset experiment on GFT means training on BP4D+ while testing on GFT.	75
4.8	Performance evaluation of STM (Chapter 3) in terms of within-dataset (top) and between-dataset (bottom) scenarios in the GFT [50] and BP4D+ [283] datasets. For example, the between-dataset experiment on GFT means training on BP4D+ while testing on GFT.	76
5.1	Distribution of event lengths in different datasets: min and max show the shortest and longest length of a common event. 25-, 50-, and 75-th indicate degrees of percentiles.	93
5.2	Distance and quality analysis on CMU Mocap dataset: (top) χ^2 distance using $1e-3$ as unit, (bottom) recurrent consistency. SW_s^* indicates the optimal window found by SW_s with step size $s = 5, 10$; SW_s^μ and SW_s^σ indicate average and standard deviation among all windows. Scores of the best discovery are marked in bold. . .	99
5.3	Comparison between ED and alternative methods in terms of running time, F1-event (F1E), and F1 on the supervised AU detection task.	102
A.1	Cross-subject AU detection on GEMEP-FERA dataset. “N S” denotes SA with either nearest-neighbor (N) or SVM (S).	119
B.1	The descriptive rates of the Miami Modeling dataset in terms of different sessions and faces of interest. #occ indicates the number of frame occurrence, #total indicates total number of frames in a particular session, and rate indicates the percentage of descriptive rates.	122
B.2	The performance evaluation on the Miami Modeling dataset [170] in terms of ACC, PA and NA for different sessions and faces of interest.	122
B.3	The performance evaluation on the CLOCK dataset in terms of PA, NA, S-score and AUC for individual AUs.	123

List of Publications

This thesis includes several materials that have appeared previously. Below is a full list of earlier publications in the context of this thesis.

Transductive Framework

Transductive learning is a class of semi-supervised learning that makes use of unlabeled data for training. In automatic analysis of facial actions, the most common scenario is the *leave-one-subject-out* (LOSO) protocol, *i.e.*, test on a subject that is excluded from the training set. Seeing that such test samples are from the same subject and given *for free* at prediction time, we investigated the use of unlabelled test data to improve generalizability for binary classification [38, 39, 277, 278] and multi-class classification [279].

- [279] Zhang, A., **Chu, W.S.**, De la Torre, F., Hodgins, J.K.: Multi-class selective transfer machine and its application to human activity recognition. In: In submission (2017)
- [278] Zeng, J., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Xiong, Z.: Confidence preserving machine for facial action unit detection. TIP **25**(10), 4753–4767 (2016)
- [39] **Chu, W.S.**, De la Torre, F., Cohn, J.F.: Selective transfer machine for personalized facial expression analysis. TPAMI (99) (2016)
- [277] Zeng, J., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Xiong, Z.: Confidence preserving machine for facial action unit detection. In: ICCV (2015)
- [38] **Chu, W.S.**, De la Torre, F., Cohn, J.F.: Selective transfer machine for personalized facial action unit detection. In: CVPR (2013)

Supervised Framework

Supervised learning is the most common framework in the context of facial action analysis. We have explored a number of supervised algorithms for detecting facial Action Units (AUs), ranging from multi-label learning with sparsity-induced groups [285, 286], ensemble learning with a cascade of tasks [65, 66], to deep learning with spatial and temporal cues [40, 287].

- [40] **Chu, W.S.**, De la Torre, F., Cohn, J.F.: Learning spatial and temporal cues for multi-label facial action unit detection. In: AFGR (2017)
- [286] Zhao, K., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Zhang, H.: Joint patch and multi-label learning for facial action unit and holistic expression recognition. TIP (2016)

- [287] Zhao, K., **Chu, W.S.**, Zhang, H.: Deep region and multi-label learning for facial action unit detection. In: CVPR (2016)
- [66] Ding, X., **Chu, W.S.**, la Torre, F.D., Cohn, J.F., Wang, Q.: Cascade of tasks for facial expression analysis. Image and Vision Computing (2016)
- [285] Zhao, K., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Zhang, H.: Joint patch and multi-label learning for facial action unit detection. In: CVPR (2015)
- [65] Ding, X., **Chu, W.S.**, Torre, F., Cohn, J.F., Wang, Q.: Facial action unit event detection by cascade of tasks. In: ICCV (2013)

Unsupervised Framework

Repetition in a social context plays a critical element in behavior research, and provides a valuable indication to important signals. Without a strict requirement of annotated data, we studied the problem of discovering such repetition for video summarization [45] and human interaction [41, 45, 46] (e.g., mother-infant interaction and young adults in a small group).

- [41] **Chu, W.S.**, De la Torre, F., Cohn, J.F., Messinger, D.S.: A branch-and-bound framework for common event discovery. IJCV (2017)
- [44] **Chu, W.S.**, Song, Y., Jaimes, A.: Video co-summarization: Video summarization by visual co-occurrence. In: CVPR (2015)
- [45] **Chu, W.S.**, Zeng, J., De la Torre, F., Cohn, J.F., Messinger, D.S.: Unsupervised synchrony discovery in human interaction. In: ICCV, pp. 3146–3154 (2015)
- [46] **Chu, W.S.**, Zhou, F., De la Torre, F.: Unsupervised temporal commonality discovery. In: ECCV (2012)

Dataset and Software

A part of this thesis has contributed to the Group Formation Task (GFT) dataset [93] and the IntraFace software¹ [60].

- [93] Girard, J., **Chu, W.S.**, Jeni, L., Cohn, J.F., De la Torre, F.: Sayette Group Formation Task (GFT): Spontaneous Facial Expression Database. In: AFGR (2017)
- [60] De la Torre, F., **Chu, W.S.**, Xiong, X., Ding, X., Cohn, J.F.: IntraFace. In: AFGR (2015)

¹An extension of the IntraFace software, *FacioMetrics*, was acquired by Facebook on Nov 2016.

Notation

\mathbf{X}	Matrix (bold upper-case letters)
\mathbf{X}_i	The i -th column of matrix \mathbf{X}
X_{ij}	The (i, j) -th element of \mathbf{X}
\mathbf{x}	Column vector (bold lower-case letters)
x_j	The j -th element of vector \mathbf{x}
a or α	Scalar (non-bold letters)
\mathbb{R}	Real numbers
$\mathbb{R}^{m \times n}$	An $m \times n$ real matrix
$\mathbf{I}_n \in \mathbb{R}^{n \times n}$	The identity matrix
$\mathbf{1}_{m \times n} \in \mathbb{R}^{m \times n}$	A matrix of ones
$\mathbf{0}_{m \times n} \in \mathbb{R}^{m \times n}$	A matrix of zeros (null matrix)
$\det(\mathbf{X})$ or $ \mathbf{X} $	Determinant of \mathbf{X}
$\text{vec}(\mathbf{X})$	Vectorized \mathbf{X} (column-major)
$\text{diag}(\mathbf{X})$	Diagonal matrix of \mathbf{X}
$\text{tr}(\mathbf{X})$	Trace of \mathbf{X}
\mathbf{X}^\top	Transposed matrix
\mathbf{X}^{-1}	Inversed matrix
$\ \mathbf{x}\ _p = (\sum_i x_i ^p)^{1/p}$	Vector norm
$\ \mathbf{X}\ _{p,q} = \left[\sum_j (\sum_i X_{ij} ^p)^{q/p} \right]^{1/q}$	$L_{p,q}$ norm
$\ \mathbf{X}\ _F^2 = \text{tr}(\mathbf{X}^\top \mathbf{X}) = \text{tr}(\mathbf{X} \mathbf{X}^\top)$	Frobenious norm
$\mathbf{X} \circ \mathbf{Y}$	Hadamard (elementwise) product
$\mathbf{X} \otimes \mathbf{Y}$	Kronecker product
\mathbf{S}	Sequence
$\mathbf{S}[a, b]$	Subsequence of \mathbf{S} starting from frame a to frame b

Chapter 1

Introduction

“Facial actions speak louder than words.”

This Thesis



Figure 1.1: Facial expression is one of the most powerful channel for non-verbal communication. Such actions of the face can convey emotions of an individual to observers, and serve as a primary means of exchanging social information between humans.

1.1 Motivation

Facial actions (or more ubiquitously known as facial expressions) are one of the most powerful channel of nonverbal communication, and has been a focus of research in human behavior for over a century [58]. It is central to several leading theories of emotion [75, 236] and has been a focus of heated debates about issues in emotion science. Activities on faces convey information that regulates social behavior, reveals brain function and pathology, indicates interpersonal attraction or repulsion, and verifies the physiological presence of emotion. These rich and irreplaceable cues on the face encourage research on coding facial actions, either manually or automatically, continue to thrive. Automatic analysis of facial actions has become an obligatory resource for researchers in psychology, psychiatric functioning, social and emotional development, behavioral science, pain assessment, and so forth.

The interest of using the face as a potential biometric begins from Prof. Kanade, one of the world's foremost researchers in computer vision and robotics. His Ph.D. thesis [128] is one of the pioneers that exploits faces for identity recognition. Since then, there has been an increasing interest in automatic analysis of facial actions (AFA) within the vision community. Early works focused facial actions on holistic facial expressions, which are mutually exclusive and mostly posed, *e.g.*, [147, 148]. Such expressions, including anger, contempt, disgust, fear, happy, sadness, and surprise, are universal. More recently, investigators have focused on automated systems based on the Facial Action Coding System (FACS) [76]. Being one of the most influential descriptions of facial actions, FACS defines the facial codes as the so-called *Action Units (AUs)*, which correspond to the contraction of one or more facial muscles. Fig. 1.2 illustrates the 12 common AUs and the 7 universal facial expressions.

Automatic analysis of facial action (AFA) enables various applications such as surveillance [62], marketing [217], drowsy driver detection [165], parent-infant interaction [89], social robotics [19], telenursing [56], expression transfer for video gaming [114], and subtle expression detection [101]. As can be seen in leading journals and conferences of computer vision and machine learning (*e.g.*, CVPR/ICCV/ECCV/NIPS/ICML/TPAMI/IJCV), advance has been dramatic toward face de-

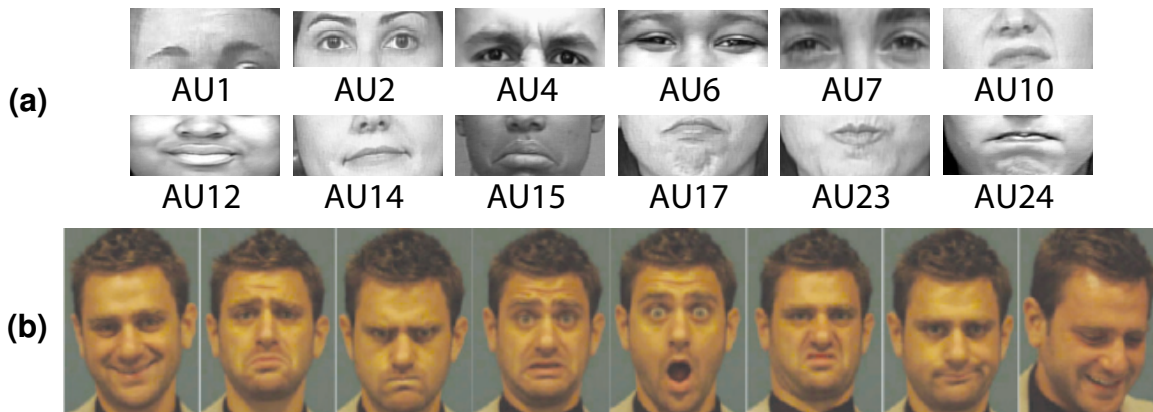


Figure 1.2: Illustration of (a) 12 common action units (AUs), and (b) holistic facial expressions: (from left to right) happiness, sadness, anger, fear, surprise, disgust, contempt, and embarrassment. (figure credits to [49])

tection, facial feature detection and tracking, face recognition, facial expression transfer, and facial attribute estimation. To meet the needs of diverse applications, numerous consumer packages for automatic face analysis have recently been introduced in recent five years. Table 1.1 summarizes some consumer software companies, among which Emotient and FacioMetrics were acquired by Apple and Facebook, respectively, in the year that the author prepared this dissertation. All these evidence shows that automated analysis of facial actions is an important problem both in academia and industry.

1.2 Learning Frameworks

Machine learning has become a dominant tool in achieving an AFA system, just as human learning process is prevailingly related to human visual system. A substantial portion of topics in major vision conferences is relevant to a learning problem, such as object recognition, visual tracking, pose estimation, face analysis, to name but a few. Analysis of facial actions (AFA) is no exception. We refer the reader to Chapter 2 for more detailed discussion on advances and challenges for AFA.

In this thesis, we are particularly interested in three learning frameworks, namely *transductive learning*, *supervised learning* and *unsupervised learning*. Denote $\mathcal{L} = \{\mathbf{x}_i, y_i\}_i$ as training data with labels $\{y_i\}_i$, and $\mathcal{U} = \{\mathbf{x}_j\}_j$ as the data without labels. We define more formally the three learning frameworks as follows:

- Supervised: Learn a classifier f using \mathcal{L} ; then use f to predict labels for \mathcal{U} .
- Transductive: The set \mathcal{U} is the test data and is available at training time; then use both \mathcal{L} and \mathcal{U} to transfer the information from labeled samples to unlabeled.
- Unsupervised: Without labels $\{y_i\}_i$, learn to find the structure or relationships among inputs.

We note that transductive learning is closely related to semi-supervised learning (SSL), which learns a function f with both \mathcal{L} and \mathcal{U} , and then uses f to predict labels for another previously

Table 1.1: List of companies on automated analysis on facial actions

Company (alphabetical order)	URL
Affectiva	http://www.affectiva.com/
Emotient	http://www.emotient.com/
Face++	http://www.faceplusplus.com/
FaceReader	http://www.noldus.com/
FacioMetrics	http://faciometrics.com/
megvii	https://megvii.com/about
NVSIO	http://www.nviso.ch/
PittPatt	http://www.pittpatt.com/
RealEyes	http://www.realeyes.me/
SenseTime	http://www.sensetime.com/
Visage	http://visagetech.com/

¹ PittPatt was acquired by Google on Jul 2011.

² Emotient is previously CERT [151] and acquired by Apple on Feb 2016.

³ FacioMetrics is an extension of IntraFace [60] and acquired by Facebook on Nov 2016.

unseen set of data. Since SSL learns a function to be applied on test data, it is also known as *inductive learning*. On the contrary, transductive learning does not require to learn an explicit prediction function f . Transductive learning thus offers benefits in situations when (1) labels are expensive to collect, (2) instances are cheap to collect, and (3) we know in advance the instances to be classified. Such situations strongly resemble realistic application scenarios for many AFA applications. For instance, the monetary and temporal costs of collecting and annotating a large dataset of facial actions are considerable. Annotating a single minute of video using the Facial Action Coding System (FACS) [76] can require well over an hour of annotators' time. In addition, most applications are tested with a fixed set, *e.g.*, test on a video containing a fixed number of subjects, and thus make the test instances known in advance.

1.3 Contributions and Organization of This Dissertation

This thesis describes the progression of automatic facial action analysis along multiple dimensions: *transductive learning* with unlabeled data from an unknown target domain, *supervised learning* with spatial and temporal cues, and *unsupervised learning* with completely no annotations. One important lesson of this work is the possibilities of automatically analyzing facial actions with various degrees of annotations. As the number of annotated samples and diversity in subjects increase over past years, we comprehensively investigate conventional approaches that treat separately the tasks of feature extract and classifier learning, and an end-to-end system that jointly optimize for both tasks. Furthermore, observing that facial actions are usually triggered through interaction, we provide algorithmic design for studying facial actions without annotations.

Below we outline the organization of this thesis, which naturally summarizes contributions this study brings to the community of automatic facial analysis (AFA):

1. **An in-depth review on the AFA literature (Chapter 2):** This chapter reviews a broad spectrum in the AFA literature, spanning from face detection and registration, feature extraction, numerous techniques in modeling, source of errors, and alternative methods for analyzing human behaviors. Despite existing surveys [49, 164, 186, 204, 242, 248], we also discuss relatively unexplored challenges from our very own perspective, in hope to motivate future studies. In our discussion, we cover multiple learning paradigms that have attracted increasing attention in AFA, including transfer learning, semi-supervised learning, region learning, multi-label learning, ensemble learning and continuous learning. We also discuss temporal models, and their integration with static models.
2. **A transductive framework for personalized facial expression analysis (Chapter 3):** In this chapter, we systematically identify the challenge of *individual differences*, which results in person-specific biases in the feature space and thus hinder the generalizability of contemporary classifiers. Person-specific classifiers would be a possible solution, but only for a paucity of training data. Instead, we introduce a transductive learning method, **Selective Transfer Machine (STM)**, which uses the unlabeled data from a test subject to personalize a standard generic classifier. Extensive experiments were conducted on four datasets using four scenarios: *within-subject*, *cross-subject*, *cross-dataset* AU detection, and *holistic expression recognition*. We found: (i) Some training samples are more instrumental than others, and we can identify those training samples using STM. (ii) The effectiveness of STM scales as the number of training subjects increases. To our best knowledge, this is one of the first studies

identifying the person-specific biases in the AFA community.

3. **A supervised framework with spatial and temporal cues for AU detection (Chapter 4):** This chapter identifies three more aspects that affect performance of AU detection: *spatial representation*, *temporal modeling*, and *AU correlation*. Unlike most studies that tackle these aspects separately, we propose a hybrid network architecture to jointly model them. Specifically, spatial representations are extracted by a Convolutional Neural Network (CNN), which, as analyzed in this paper, is able to reduce person-specific biases caused by hand-crafted features (*e.g.*, SIFT and Gabor). To model temporal dependencies, Long Short-Term Memory (LSTMs) are stacked on top of these representations, regardless of the lengths of input videos. The outputs of CNNs and LSTMs are further aggregated into a fusion network to produce per-frame prediction of 12 AUs. Our network naturally addresses the three issues together, and yields superior performance compared to existing methods that consider these issues independently. Furthermore, we provide visualization of the learned AU models, which, to our best knowledge, reveal how machines see AUs for the first time. Finally, we also proposed two multi-label sampling strategies to address the class imbalance issues in our AU data. Extensive experiments were conducted on two large spontaneous datasets, GFT and BP4D, with more than 400,000 frames coded with 12 AUs. On both datasets, we report improvements over a standard multi-label CNN and feature-based state-of-the-art.
4. **An unsupervised framework for common event discovery in human interaction (Chapter 5):** Without the requirement of annotated data, we investigate a relatively unexplored problem, Common Event Discovery (CED), which aims to find common *inter-personal* patterns among two or more videos. We develop an efficient branch-and-bound (B&B) framework that affords exhaustive search yet guarantees a global optimal solution. The proposed B&B framework is entirely general. It takes from two or more videos any signals that can be quantified into histograms. We show that a slight modification of the framework can be readily applied to discover events happening at different time (event commonality) or around the same time (synchrony), video indexing, and supervised event detection. The effectiveness was evaluated on human interaction tasks: group formation task, parent-infant interaction, and motion capture data.

■

Advances and Challenges

“It is paradoxical, yet true, to say, that the more we know, the more ignorant we become in the absolute sense, for it is only through enlightenment that we become conscious of our limitations. Precisely one of the most gratifying results of intellectual evolution is the continuous opening up of new and greater prospects.”

Nikola Tesla

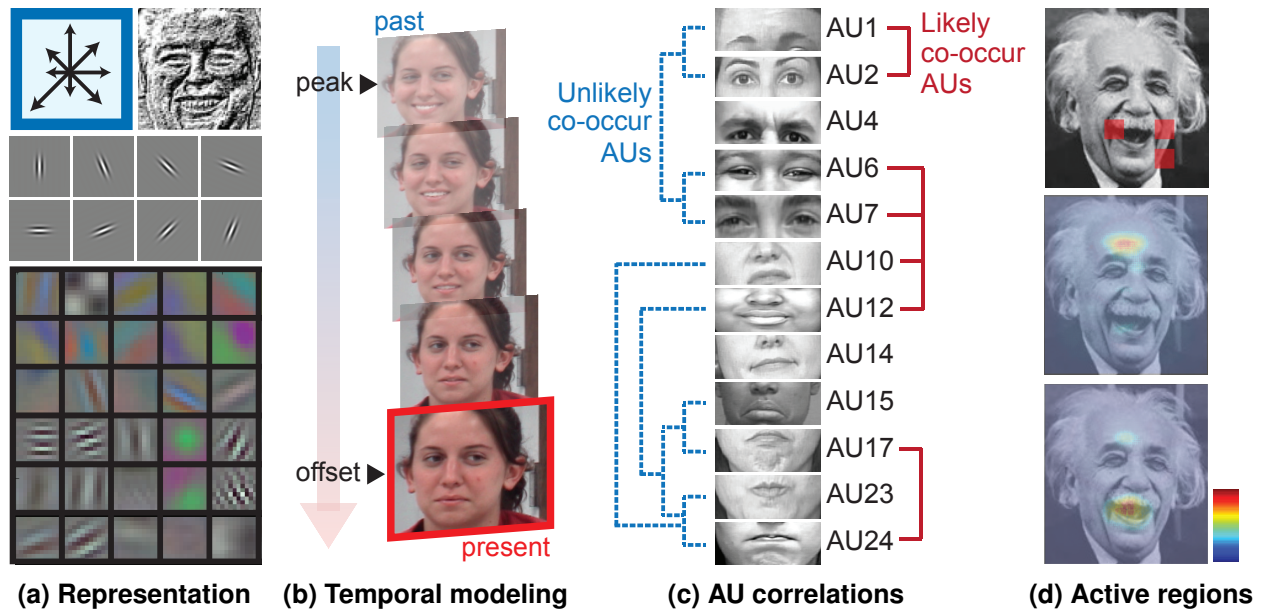


Figure 2.1: Illustration of **contemporary challenges** in automated analysis of facial actions (AFA): (a) Learning spatial feature presentation, (b) learning temporal consistency, (c) learning / employing AU correlation, and (d) learning active patches / regions.

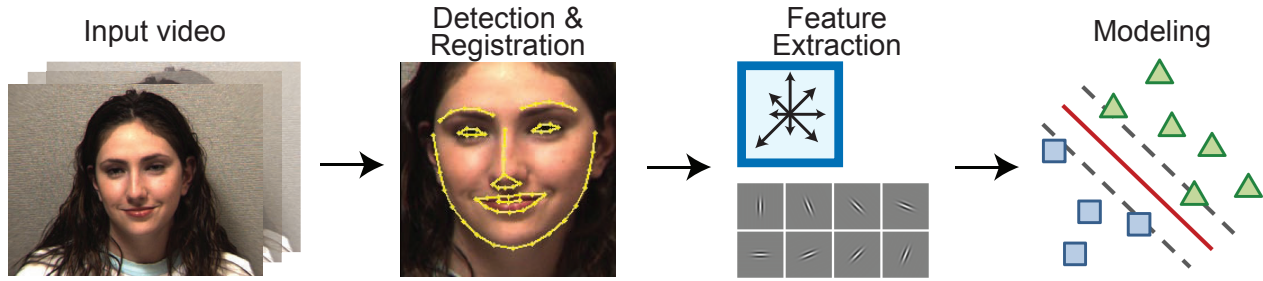


Figure 2.2: Conventional pipeline for automated analysis of facial actions (AFA).

Automated analysis of facial actions and facial expressions have been an important and popular topic in computer vision, and have been successfully applied in the domain of psychological studies, especially to detection of pain, emotion, depression, and distress. Various applications include digital marking, human-robot interaction, drowsy-driver detection, and other cognitive-emotional inferences such as in the classroom and in tutoring. As shown in Fig. 2.2, a conventional automated analysis of facial action (AFA) system entails at least three components: Face detection and registration, feature extraction, and modeling. Despite progress has been focused on different components in the system, a unified and generalizable approach has been mostly lacking. As depicted in Fig. 2.1, this document emphasizes at least four contemporary challenges remain for AFA systems in the context of machine learning:

1. **Spatial representation:** Engineered features, *e.g.*, SIFT, have shown to cause person-specific biases in estimating AUs, causing sophisticated learning methods such as personalization [38, 39, 205, 271]. A good representation for AUs must generalize to unseen subjects, regardless of the existence of individual differences caused by appearance, behaviors or facial morphology.
2. **Temporal consistency:** Temporal information is crucial for telling AUs like humans. Due to the richness, ambiguity, and dynamic nature of facial actions, it remains unclear how such temporary memory can be effectively encoded and recalled.
3. **Learning / employing AU correlations:** The presence of AUs influences each other. Fig 2.1(c) shows an illustration of likely and unlikely co-occurring AUs. For instance, the occurrence of AU12 suggests an occurrence of AU6, and reduces the likelihood of AU15. Such correlation helps an AU detector determine the occurrence of one AU given others.
4. **Active regions:** Most current approaches extract features across the entire face and concatenate them for discriminating facial actions. Knowing active facial regions gives valuable information of recognizing a specific facial action. Learning such regions not only improves interpretability and accuracy, but reduces computation cost by eliminating a large irrelevant portion of the face.

This section reviews recent advances in each building component (Sec. 2.1~Sec. 2.3), discusses existing attempts to address sources of errors (Sec. 2.4), and other relevant methods for analyzing human behaviors (Sec. 2.5). Interested readers may refer to more complete surveys in AFA, *e.g.*, [49, 164, 186, 204, 242, 248].

Table 2.1: Conventional features for AFA

Type	Feature	Year	Examples
Geometric	Shape model parametrization	2012	[162, 235]
	Geometry of facial components	2010	[46, 292]
	Landmark locations	2006	[36, 161, 162, 200]
Appearance	SIFT/DAISY	2011	[296]
	Discrete Cosine Transform (DCT)	2011	[90]
	Local Phase Quantization (LPQ)	2011	[55, 125]
	Local Binary Patterns (LBP)	2009	[174, 215, 216, 248, 290]
	Hist. of Oriented Gradient (HOG)	2009	[183]
	Gabor filters	2006	[10, 141, 151, 237]
	Raw pixels	2000	[129]
Dynamic	Longitudinal expression atlases	2012	[107]
	Gabor motion energy	2010	[263]
	Bag of Temporal Words (BoTW)	2010	[44, 222, 225]
	Dynamic Harr	2009	[269]
	Volume LBP (LBP-TOP)	2007	[284]
	Optical flow	2005	[105]
	Motion History Image (MHI)	2001	[14, 247]
Fusion	Multiple feature kernels	2012	[214]

2.1 Face Detection and Registration

Face detection and registration of non-rigid facial features is a long-standing problem in computer vision. The goal of detection is to first detect the face (typically as a bounding box), and localize facial landmarks (*e.g.*, eyes, nose and mouth). Using such landmarks, the registration step removes the effects of spatial variation in face position, rotation, proportions, and moderate head poses. For facial landmark detections, Parametrized Appearance Models (PAM) are among the most popular methods. PAM include the Lucas-Kanade method [160], Active Appearance Models (AAM) [54, 166], Constrained Local Models (CLM) [36], and, more recently, Zface [119] and Supervised Descent Method [266]. Similarity transformation [222, 242, 296] is a standard approach to register faces with respect to an averaged face. A Delaunay triangulation uses a backward piecewise affine warping to extract features in areas not explicitly tracked.

2.2 Feature Extraction

With advances in face detection and registration, there has been renewed emphasis on biologically inspired features and variations. As summarized in Table 2.1, we broadly categorize current approaches of feature extraction into four types: *geometric*, *appearance*, *dynamic*, and *fusion*. After a face is detected and registered, different feature extraction methods are applied to obtain a vector representation. Geometric features contain information about shape and locations of permanent facial features, such as eyes or nose. Standard approaches rely on landmark coordinates [36, 161, 162], geometry of facial components [46, 292], a connected face [222], or face compo-

Table 2.2: Conventional classification models for AFA

Type	Model	Year	Examples
Static	Convolutional Neural Networks	2012	[116, 155, 196]
	Support Vector Machine (SVM)	2007	[162]
	Boosting	2006	[10, 134, 156]
	Neural Network (NN)	2005	[130]
	Bayesian Network	2003	[47, 48]
Temporal	Long Short-Term Memory (LSTM)	2013	[260]
	Conditional Random Field (CRF)	2009	[25, 253]
	Gaussian process	2009	[32]
	Dynamic Bayesian Network (DBN)	2007	[239, 257]
	Isomap embedding	2006	[26]
	Rule-based	2006	[188, 243]
	Hidden Markov Model (HMM)	2000	[47, 57, 135, 148, 262]
Hybrid	Cascade of Tasks (CoT)	2013	[65, 66]

nent shape parameterization [162, 235]. Geometric features have performed well for facial actions with dramatic shape changes, such as mouth opening or smiles, but not all facial action recognition tasks. Geometric features alone are often insufficient due to individual differences (*e.g.*, facial morphology and behavior), head motions and registration errors [35].

Appearance features, which often are biologically inspired, represent skin texture changes and afford increased robustness to tracking and registration error. Because facial muscle activities produce momentary changes in facial appearance, appearance features and its permutations have been widely applied to facial expression analysis. Representative methods include SIFT [296], DAISY [296], Gabor filters [10, 150], LBP [125, 215, 290], Bag-of-Words model [221, 222], and compositional [270].

Motion is a critical cue to recognize facial actions. *Dynamic* features, a newer technique, encodes such temporal information into feature extraction. Examples include optical flow [105] to estimate motion in a subset of facial muscles, bag of temporal words [44, 222, 225] to quantify video features in analogy to text retrieval, Motion History Image (MHI) [14, 247] to compress temporal transition into one frame, volume LBP/LPQ [284] to capture local binary patterns in a temporal extension, Gabor motion energy [263] to explore spatio-temporal patterns using Gabor filters, and others. *Fusion* approaches incorporate multiple features, *e.g.*, Multiple Kernel Learning (MKL) [214], and have yet to prove superior to other approaches [248].

2.3 Modeling

Recent studies in facial action modeling have three major evolutionary trends: *static modeling*, *temporal modeling*, and *hybrids*, as summarized in Table 2.2. Below we review each in turn. Fig. 2.3 illustrates three recent examples to tackle different perspectives of modeling.

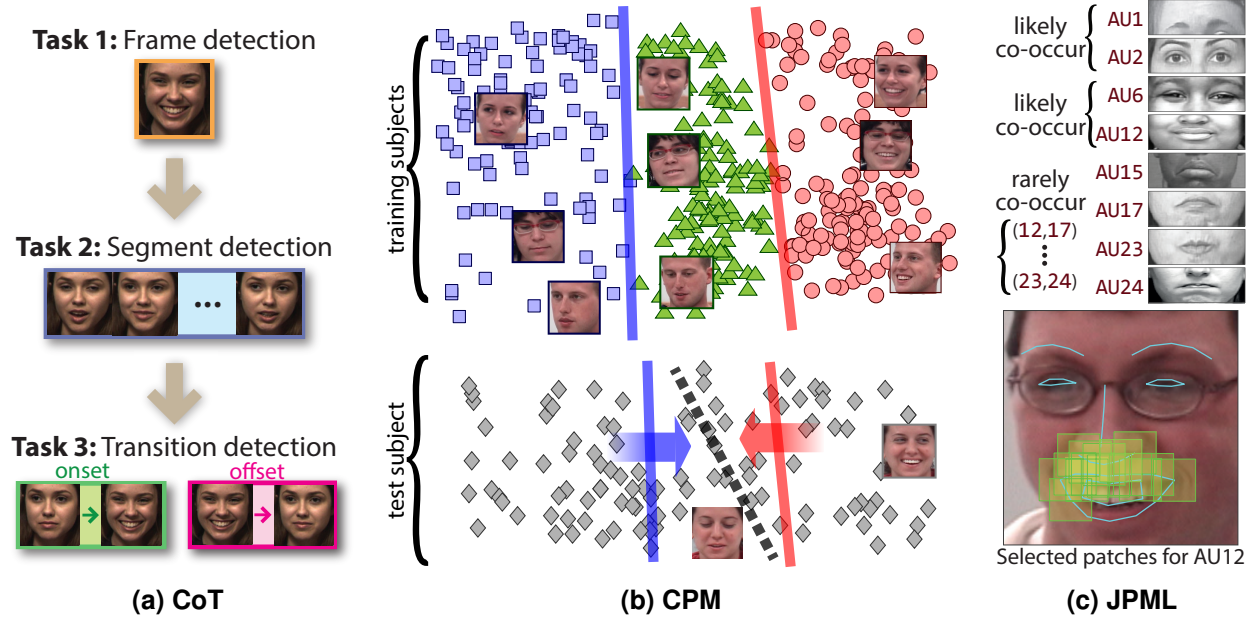


Figure 2.3: Illustrations of three recent models that the author and colleagues pursued to address AFA: (a) Cascade of Tasks (CoT) [65, 66], (b) Confidence Preserving Machine (CPM) [277, 278], and (c) Joint Patch and Multi-label Learning (JPML) [285, 286]. Note that these studies are excluded from this thesis. We refer interested readers to references for more details.

2.3.1 Static Modeling

Static modeling, or referred as *frame-based methods*, detects AU occurrence in individual frames. The first AU detection challenge (FERA) [248] indicates that most approaches, including the winning one, were frame-based. Static modeling is done by extracting geometric and/or appearance features to represent each frame, and then feeds these features into modeling (*i.e.*, classifiers or regressors). Representative approaches include Neural Network [130], Adaboost [10, 296], SVMs [36, 38, 39, 162, 222, 285], and Convolutional Neural Networks [84, 104, 155]. In general, frame-based models are shown to be able to detect subtle AU events because of their sensitivity to each frame. However, such models usually produce non-smooth predictions and are prone to noise due to the lack of temporal consistency. We broadly categorize static models into learning paradigms, as summarized in Table 2.3. Each learning paradigm shows an attempt to address at least one particular challenge in AFA. We denote “**L: C**” for a learning paradigm **L** to address a challenge **C**.

Table 2.3: Summary of learning paradigms for attacking different challenges in AFA

Learning paradigm	: Challenge
Transfer learning	: Remedy domain bias
Semi-supervised learning	: Employ unlabeled data
Region learning	: Identify decisive facial regions
Multi-label learning	: Leverage AU correlations
Continuous learning	: Estimate intensities of facial actions

Transfer learning: Remedy domain bias

A common assumption in AFA is that training and test samples are drawn from the same distribution. Recently, studies identified that facial actions exhibit strong domain biases due to individual or dataset differences caused by face appearance, behaviors, facial morphology or recording environment [33, 38, 39]. Causes to such bias trace back to the spatial representation. More details will be discussed in Chapter 3. Such bias is also known as *domain shift* in the object detection literature, where labels of interest could occur infrequently, or distributions of features vary dramatically between and within datasets. These factors were found to cause significant biases in object categorization [6, 30, 139, 203, 240, 268]. We refer interested readers to [194] for more comprehensive review. Most transfer learning methods used a supervised approach in which one or more labeled target instances are required to guide the transfer procedure. In AFA, typically no knowledge is provided for a target subject.

Close to our problem is a special case in transductive transfer learning known as *covariate shift* [227]. In covariate shift, train and test domains follow different distributions but the label distributions remain the same. Such differences produce shifted distributions in feature space, and thus hinder the generalizability of pre-trained classifiers. Given a source and a target domain, Domain Invariant Projection (DIP) [7] finds a domain-invariant space in which training and test data share a similar distribution. Similarly, Subspace Alignment (SA) [87] and geodesic distances on a Grassmann manifold [97, 98] represent each domain as a subspace/manifold, and then learns a mapping function that aligns the sources to the target one. For raw features, the discrepancy can be alleviated by directly learning a transformation [122, 175]. However, learning the above projection or transformation is unsupervised, and thus remains unclear how source labels can be incorporated.

On the other hand, Dudík *et al.* [74] infer the re-sampling weights through maximum entropy density estimation without target labels. Maximum Mean Discrepancy (MMD) [16] measures the discrepancy between two different distributions in terms of expectations of empirical samples. Without estimating densities, Transductive SVM (T-SVM) [126] simultaneously learns a decision boundary and maximizes the margin in the presence of unlabeled patterns. Domain adaptation SVM (DA-SVM) [20] extends T-SVM by progressively adjusting the discriminant function toward the target domain. SVM-KNN [280] labels a single query using an SVM trained on its k neighborhood of the training data. Each of these methods uses either all or a portion of the training data. Considering distribution mismatch, Kernel Mean Matching (KMM) [103] directly infers re-sampling weights by matching training and test distributions. Following this idea, Yamada *et al.* [267] estimated relative importance weights and learned from re-weighted training samples for 3D human pose estimation.

For AFA, more recently Chu *et al.* [38, 39] proposed Selective Transfer Machine (STM) to iteratively re-weight training instances such that the mismatch between training and test distributions and classification loss can be jointly minimized. Confident Preserving Machine (CPM) [277, 278] trained confident classifiers using “easy” training samples alone, and then propagate predictions of these classifiers from easy test samples to hard test samples. Along this direction, there have been several studies that describe a training domain as classifier parameters, and assume that an ideal classifier for the test domain can be represented as a combination of the learned classifiers [72, 205, 268]. Yang *et al.* [271] further extended personalization for estimating AU intensities by removing a person’s identity with a latent factor model. Rudovic *et al.* [201] interpreted the person-

specific variability as a context-modeling problem, and propose a conditional ordinal random field to address context effects. Other studies merged into this direction could be found for intensity estimation [271] and emotion recognition [276]. However, while progress has been made, these studies still resort to hand-crafted features. As will be discussed in Chapter 4, this thesis reveals that person-specific biases from such features can be instead reduced by learning them.

Semi-supervised learning: Employ unlabeled data

To capture the richness and ambiguity of facial actions, a sophisticated AFA system typically require large amounts of labeled training data. However, acquiring annotations of facial actions is laborious and time-consuming. One has to learn a basic system for describing anatomical movement of the face, and may need a 1-week FACS training to be qualified to FACS code. For an experience coder, coding a 1-minute video on one action unit (AU) could take 30 minutes. Semi-supervised learning (SSL), also known as *self-learning* or *decision-directed learning*, has emerged as a promising approach to incorporate unlabeled data for training. SSL starts by training only on labeled data, and uses the supervised model to annotate a part of the unlabeled data. The additional annotated data are then included to retrain the model. These two steps iterate until a stopping criterion is met.

Standard SSL methods make various assumptions on relationships between input and label space [28]. A *smoothness* assumption enforces neighboring samples in the feature space to share similar labels (*i.e.*, corresponding outputs should be close), and can be typically modeled by a graph-based method [158]. CPM [277], as an example, utilized a quasi-semi-supervised approach that preserves spatial-temporal smoothness on unlabeled test samples. A *cluster* assumption encourages samples within the same cluster to be assigned to the same class label [29]. This assumption has been shown to be equivalent to low-density separation, and can be extended to entropy minimization [100]. A *manifold* assumption considers that samples lie on a low-dimensional manifold. This assumption alleviates the *curse of dimensionality*¹, and thus facilitates the task for density estimation in the potentially high-dimensional input space. Laplacian SVM (Lap-SVM) [12, 168] incorporated this assumption as a regularization for learning an SVM in its primal form. Other work explored the combination of the three assumptions using a boosting framework [34]. Interested readers are referred to [28, 294] for a more extensive review. Similar to transfer learning, one may apply SSL to tackle with unlabeled target data. However, SSL could suffer from the fact that samples from different individuals or datasets could be drawn from different distributions.

Region learning: Identify decisive facial regions

Identifying decisive facial regions is critical in AFA systems, just as humans recognize a facial action. One option is to perform *feature learning* to select a representative subset of raw features, *e.g.*, AdaBoost [150], GentleBoost [117], and linear SVMs [161]. These methods typically quantify features across the entire facial image. However, features from different facial regions provide unique information to distinguish holistic expressions [218, 220]. For instance, mouth corners tell happiness from sadness, or eye brows tell surprise from anger. Cohn and Zlochower [51] found that facial regions differentially contribute to holistic expressions. Zafeiriou and Pitas [275] applied elastic graph matching to produce an expression-specific graph that identified the most discriminant

¹*Curse of dimensionality*: Volume grows exponentially with the number of dimensions, and an exponentially growing number of samples is required for statistical tasks such as reliable density estimation [28].

facial landmarks for specific expressions.

As described in FACS [76], facial regions are also decisive to AUs. For instance, to tell presence of AUs 12, 14 or 15, one needs to identify mouth corner beforehand; to tell AUs 1, 2, or 4, eyebrow is where to examine. In other words, for different facial actions, only a sparse facial region is meaningful, leading to a critical property of a learning algorithm to automatically pick up the important regions. Following this intuition, feature selection can be modeled within regions, or *patches*, of the face. Zhong *et al.* [290] divided a facial image into uniform patches, and then categorized these patches into common and specific ones for holistic expression recognition. Given uniformly divided facial patches, Liu *et al.* [157] proposed to select common and specific patches corresponding to an expression pair (*e.g.*, happiness-sadness). However, without a proper face registration, dividing a face image into uniform patches would easily fail on faces with modest or large pose. In addition, these models learn patch importance in an indirect way without considering regional importance for different facial actions. Recently, Taheri *et al.* [232] defined regions for different AUs, and proposed a two-layer group sparsity coding to recover facial expressions using the composition rule of AUs. Using predefined grids on faces, Kotsia *et al.* [136] designed an architecture to track grids for each AU and then adopted AU composition rules to predict holistic expressions. The relation between regions and AUs are pre-defined, and thus can not be learned. To explicitly model contributions of different facial regions to individual AUs, Zhao *et al.* [285, 286] proposed a joint framework that captures the dependencies between patches and between multiple AUs. This joint framework shows that an active region can be inferred directly from data and statistical prior.

Multi-label learning: Leverage AU correlations

The outputs of an AFA system (*e.g.*, AUs) are strongly correlated due to the biological structure of facial muscles. For instance, AUs 6 and 12 are known to co-occur in facial expressions of enjoyment, embarrassment, and pain, but not in distress or sadness. Due to this fact, analysis of AUs is more naturally a *multi-label* instead of a *multi-class* classification problem as in holistic expression recognition, *e.g.*, [69, 156]. Recent realization of the omnipresence of strong AU correlations [76, 257] has drawn more research attention to this domain. Multi-label learning (ML) aims to use such multi-label nature of AUs to improve AFA. ML was primarily motivated by the emerging need for automatic text-categorization and medical diagnosis. ML is close to *multi-task learning*, which has been empirically [81, 82, 241] and theoretically [4, 13] shown to often significantly improve performance relative to learning each task independently. To this end, generative Bayesian Networks (BN) [238, 239] and dynamic BN [257] have been used to exploit AU correlations with consideration of their temporal evolutions. Using generic domain knowledge, AU correlations can be modeled as a directional graph without training data [146]. Rather than learning, pairwise AU relations can be statistically inferred using annotations, and then injected into a multi-task framework to select important patches per AU [285]. Another framework, a multi-kernel learning approach [281] captured intrinsic AU relations by extending the regularized multi-task learning. Results showed that using the dependencies among AUs helped improve the AUs with smaller sample size, *e.g.*, AUs 1 and 2. Furthering this line, a multi-conditional latent variable model [78] was proposed for simultaneous facial feature fusion and detection of facial action units. In addition, a restricted Boltzmann machine (RBM) [258] was developed to directly capture the dependencies between image features and AU relationships. Along this direction, image features and AU outputs were fused

in a continuous latent space using a conditional latent variable model [79]. For the scenario with missing labels, a multi-label framework can be applied by enforcing the consistency between the prediction and the annotation and the smoothness of label assignment [262]. For a detailed review on ML see [226, 244].

Continuous learning: Estimate intensities of facial actions

While this thesis focuses more on AU detection, estimating AU intensity has attracted an increasing attention thanks to the availability of recent datasets [161, 167, 185, 282]. Reviewing the literature, there have been a few to address intensity estimation. Following [201], we broadly categorize relevant studies into classification-based or regression-based methods.

Classification-based approach aims to classify intensity levels into different classes. Among this type of approaches, Savran *et al.* [207] used both 2D and 3D data, along with, give that fact that distances to SVM hyperplanes for AU detection are correlated with intensity levels of AUs [10]. Mahoor *et al.* [163] applies SVM classifiers to categorize AU intensities. The original features were the concatenation of facial landmarks and grayscale pixel intensities, whose dimension were further reduced using locality preserving index. However, such methods assume independence between class labels, while AU intensities inherently preserve ordinality. For instance, if one knows intensity $C > B$ and $B > A$, one should be able to infer that $C > A$.

Due to the continuous and ordinal nature of AU intensities, regressors are amongst one of the most widely-applied approaches for intensity estimation. For instance, Savran *et al.* [207] applied Support Vector Regressor (SVR) on 2D features (*e.g.*, Gabor wavelets) and 3D features (*e.g.*, curvature, shape index, curvedness), and used AdaBoosting for feature selection. Similarly, Jeni *et al.* [120] utilized SVR on appearance features that were extracted from local facial patches. Another popular regressor, Relevance Vector Regression (RVR), was adapted in a late fusion framework for pain intensity estimation [127]. In specific, RVRs were trained on individual features (*e.g.*, DCT, LBP, or landmark points), and then fused by averaging or training an additional regressor on the output of each RVR. Another way to fuse features is through Multiple Kernel Learning (MKL). Ming *et al.* [171] employed an MKL-SVM that combines different appearance and geometry features. On the other hand, Rudovic *et al.* [201] jointly considered “W5+” (who, when, what, where, why and how) for intensity estimation. Variations such as person identity and temporal correlation were captured using two linear latent variables, each of which corresponds to changes with regards to identities or intensities. Ordinal regression was employed to infer the intermediate outputs, which were then modeled by a CRF for temporal consistency among AU intensity levels. Along this direction, Mohammadi *et al.* [173] decomposed facial image sequences into low-rank identity-related sequences and sparse expression-related sequences, and then exploited joint dictionary learning and regression to learn the intensities of multiple AUs.

2.3.2 Temporal Modeling

In AFA, modeling dynamics is crucial in recognizing facial actions like humans. To address this issue, *temporal modeling*, or referred as *segment-based methods*, captures temporal transition between contiguous frames. Pantic and Patras [187] proposed to use geometric features and temporal rule-based reasoning to recognize temporal phases of AU events. Later, Valstar and Pantic [249, 250] proposed a hybrid model that combines SVM and HMM to recognize AUs as temporal phases. This hybrid SVM-HMM approach combined the SVM’s discriminative power and HMM’s

ability to model time. Switching Gaussian process models [32] was built upon dynamic systems and Gaussian process to simultaneously track motions and recognize events. The Gaussian assumption unnecessarily holds in real-world scenarios where we do not know from which distribution video frames are sampled. Dynamic Bayesian Network (DBN) with appearance features [237, 239] was proposed to model semantic and temporal AU relationships. Non-parametric HMMs [216] were introduced to encode discrimination ability at class and state levels. A hidden CRF [25] classified over a sequence and established connections between the hidden AU states and underlying emotion. At each time step, inference was done incrementally by using previous inferences. However, these models made Markov assumption and thus lacked consideration of long-term dependencies. As an alternative, Simon *et al.* [222] proposed a structural-output SVM that detects AU segments using dynamic programming. To model relations between segments, Rudovic *et al.* [199] considered ordinal information in CRF. Recently, Walecki *et al.* [253] proposed a Variable-state Latent Conditional Random Field (VSL-CRF) model for segment expression analysis, which can automatically select optimal latent states (nominal or ordinal).

An important yet relatively unexplored task is to model AU transition (*i.e.* onsets and offsets) using temporal classifiers or models. Detecting AU transitions is arguably challenging even for humans, due to subtle changes between AU and non-AU frames. In previous approaches, accurate transition was usually detected with the help of additional information, such as an AU apex location [61]. For example, in the FAST-FACS system proposed by De la Torre *et al.* [61], the system automatically detects the boundaries of the event (*i.e.* onset and offset frames) with the help of AU apex location manually labelled by user.

In general, segment-based models make better AU detection in form of a set of contiguous frames, which is closer to human perception. However, compared with frame-level training data, segment-level training data are usually scarce. Moreover, AU segments can have complex temporal structure and are difficult to model. Consequently, segment-based models are often less discriminative and have difficulties in detecting subtle AU events.

2.3.3 Ensemble Learning

In AFA, despite many combinations of features and classifiers show better performance on different datasets, it remains unclear which combination is a clear winner over others. A recent notable yet relatively unexplored trend is, instead, to pursue ensemble learning over multiple features and/or classifiers. The intuition is that different features and classifiers bring unique information, and thus fusing diverse information helps generate more robust results than otherwise alone [208, 288]. As an example, Tariq *et al.* [234] used early-fusion by concatenating SIFT, Hierarchical Gaussianization and optical flow features, and feed such feature into an SVM classifier. Later, following a late-fusion strategy, Tariq *et al.* [233] used a log sum model to fuse the outputs of classifiers trained separately with different low-level image features. Wu *et al.* [264] studied multilayer architectures of texture-based image feature descriptors (filters). They proved that adding a second layer of nonlinear filters on top of the first layer brings consistent performance improvement. This approach can be viewed as a special way to fuse different feature descriptors. A temporal extension to the multilayer appearance features (LGBP-TOP) has been proposed by Almaev and Valstar [2]. More recently, Jiang *et al.* [124] proposed a decision-level fusion strategy to combine region-level classifiers using a weighted sum strategy. Ding *et al.* [65, 66] proposed a Cascade of Tasks (CoT) framework that carefully integrates frame-based, segment-based, and transition-based tasks in a

sequential order. Different from standard ensemble learning strategies that consider one set of features, CoT was designed to exploit both shape and appearance features and benefit subsequent task from its preceding task.

2.3.4 End-to-end Learning

Recent success of end-to-end learning, *a.k.a.* deep networks or convolutional networks, suggests that strategically composing nonlinear functions results in powerful models for visual perceptual problems. Such networks, with usually oversized parameters, enable a way of modeling the highly nonlinear visual world. Different from standard AFA systems that engineer features and classifiers separately, the deep networks are capable of jointly learning the feature representation and classifiers, and thus are referred as *end-to-end learning*. This success resorts to not only the massive available images and videos, but also the computational power especially GPUs. Closest to this thesis are the deep networks studied in AU detection and video classification.

Most deep networks for AU detection directly adapt CNNs. Gadi *et al.* [104] used a 7-layer CNN for estimating AU occurrence and intensity. Ghosh *et al.* [92] showed that a shared representation can be directly learned from input images using a multi-label CNN. To incorporate temporal modeling, Jaiswal *et al.* [118] trained CNNs and BLSTM on shape and landmark features to predict for individual AUs. Because input features were predefined masks and image regions, unlike this study, gradient cannot backprop to full face region to analyze per-pixel contributions to each AU. In addition, it ignored AU dependencies and temporal info that could improve performance in video prediction, *e.g.*, [224, 265]. In Chapter 4, we proposed a hybrid network that simultaneously models spatial-temporal context and AU dependencies, and thus serves as a more natural framework for AU detection.

The construction of our network is inspired by recent studies in video classification. Simonyan *et al.* [224] proposed a two-stream CNN that considers both static frames and motion optical flow between frames. A video class was predicted by fusing scores from both networks using either average pooling or an additional SVM. To incorporate “temporally deep” models, Donahue *et al.* [67] proposed a general recurrent convolutional network that combines both CNNs and LSTMs, which can be then specialized into tasks such as activity recognition, image description and video description. Similarly, Wu *et al.* [265] used both static frames and motion optical flow, combined with two CNNs and LSTMs, to perform video classification. Video-level features and LSTM outputs were fused to produce a per-video prediction.

Our approach fundamentally differs from the above networks in several aspects: (1) Video classification is a *multi-class* classification problem, yet AU detection is *multi-label*. (2) Motion optical flow is usually useful in video classification, but *not* in AU detection due to large head movements. (3) AU detection requires per-frame detection; video classification produces *video-based* prediction.

2.4 Source of Errors

There have been several efforts in facial expression analysis to address previously identified or suspected sources of error. To recognize subtle expressions, prior studies have investigated various combinations of features and classifiers, such as spatio-temporal directional features extracted by robust PCA [254], and a temporal interpolation {SVM,MKL,RF} classifiers [191]. Another

source of error involves head pose. For such cases, previous work sought to model head pose and expression simultaneously, *e.g.*, using a particle filter with multi-class dynamics [68] or a variable-intensity template [140]. Individual differences also cause errors, and can be approached using domain adaption methods [38, 39, 205, 271]. Other works seek to jointly recognize face identity and facial expression using a dictionary-based component separation algorithm [231]. However, other sources of error, such as human aging [106], are possible, and others may be unknown. Addressing specific sources of error individually may impair generalizability and fails to address unknown sources of error, which can further impair generalizability. Instead of dealing with specific factors, CPM [277] is a non-specific method that copes with sources of error both recognizable and not. Regardless of the type of error, CPM is able to automatically identify easy samples from hard ones, preserve confident knowledge using confident classifiers, and then transfer to a person-specific classifier.

2.5 Alternative Methods for Analyzing Human Behavior

Methods for analyzing human behavior are highly relevant to those for analyze facial actions. Below we summarize these methods in terms of supervised and unsupervised approaches.

2.5.1 Supervised Analysis of Behavior

Most literature for analyzing behaviors are supervised, including facial expression recognition [70, 150, 161, 246], surveillance system [86], activity recognition [73, 121, 195], and sign language interpretation [53]. Other works concern about the recognition of behaviors that involve more than one subject interacting in the scene. Brand *et al.* [17] introduced coupled hidden Markov models (CHMMs) to model dynamic interaction between multiple processes. Following up, Oliver and Pentland [182] proposed to recognize interaction between two people using HMMs and CHMMs, and concluded that CHMMs perform better in this task. Hongeng and Nevatia [113] proposed a hierarchical activity representation along with a temporal logic network for modeling and recognizing interaction. More recently, Liu *et al.* [152] proposed to recognize group behavior in AAL environment (nursing homes). A switch control module was performed to alternate between two HMM-based approaches according to the number of individual present in the scene. Messinger *et al.* [169] focused on specific annotated social signals, *i.e.*, smiling and gaze, and characterized the transition between behavior states by a maximum likelihood approach. Interested readers are referred to [21] for a review. These techniques, however, require adequate labeled training data, which can be time-consuming to collect and not applicable to our scenario.

2.5.2 Unsupervised Analysis of Behavior

Unsupervised methods for analyzing human behavior require no annotated data. Such methods rely on domain knowledge to discover patterns, which often preserve regularities, lying under the massive unlabeled data. For instance, Zheng *et al.* [289] presented a coordinated motion model to detect motion synchrony in a group of individuals such as fish schools and bird flocks. Zhou *et al.* [291] proposed Aligned Cluster Analysis that extends spectral clustering to cluster time series. [291] applied the technique to discover facial events in unsupervised manner. Chu *et al.* [46] proposed a B&B approach to find time boundaries of common events happening in two videos. On the other hand, time series motifs, defined as the closest pair of subsequences in one time series

stream, can be discovered with a tractable exact algorithm [177], or an approximated algorithm that is capable of tackling never-ending streams [11]. Some attempts on measuring interactional synchrony include using face tracking and expressions [273], and rater-coding and pixel changes between adjacent frames [211]. Nayak *et al.* [181] presented iterated conditional modes (ICM) to find most recurrent sign in all occurrences of sign language sentences. Recall that a synchrony is defined within a temporal window; it can contain subsequences from different videos that involve a temporal offset and sequence lengths different from each other. Given this structure, it remains unclear how a synchrony can be efficiently discovered using the above approaches.

Recently, there have been interest on temporal clustering algorithms for unsupervised discovery of human actions. Wang *et al.* [255] used deformable template matching of shape and context in static images to discover action classes. Si *et al.* [219] learned an event grammar by clustering event co-occurrence into a dictionary of atomic actions. Zhou *et al.* [292] combined spectral clustering and dynamic time warping to cluster time series, and applied it to learn taxonomies of facial expressions. Turaga *et al.* [245] used extensions of switching linear dynamical systems for clustering human actions in video sequences. However, if we cluster two sequences that only have one segment in common, previous methods for clustering time series will likely need many clusters to find the common segments. In our case, TCD discovers only similar segments and avoids the need for clustering all the video that is computationally expensive and prone to local minima. Another unsupervised technique related to TCD is motif detection [172, 176]. Time series motif algorithms find repeated patterns within a single sequence. Minnen *et al.* [172] discovered motifs as high-density regions in the space of all subsequences. Mueen and Keogh [176] further improved the motif discovery problem using an online technique, maintaining the exact motifs in real-time performance. Nevertheless, these work detects motifs within only one sequence, but TCD considers two (or more) sequences. Moreover, it is unclear how these technique can be robust to noise.

2.6 Evaluation Metrics

The performance of an AFA system is typically evaluated by various metrics to quantify the generalization ability of the trained models. Choices of one or another metric depend on a variety of factors, such as purposes of the task, preferences of individual investigators, the nature of the data, etc. For instance, behavioral scientists could prefer sensitivity for frame-wise recovery of subtle movements that are often difficult for humans to spot, while researchers in video-based recognition could find segment-based evaluation more attractive in describing temporal consistency [93]. In this thesis, we always report multiple metrics to reflect abilities carried in different models.

Denote tp and tn as the number of positive and negative instances that are correctly classified, and fp and fn as the number of misclassified negative and positive instances, respectively. Below we review a family of such evaluation metrics and their focuses.

- Accuracy (acc), computed as $\frac{tp+tn}{tp+fp+tn+fn}$, measures the percentage of correct predictions over total instances, and is commonly used for binary or multi-class classification problems.
- S-score, or “free-marginal kappa coefficient” [93], estimates chance agreement by assuming that each category is likely to be chosen at random. S-score is computed as a linear transformation of accuracy, *i.e.*, $S = 2acc - 1$ or $S = \frac{tp+tn-fp-fn}{tp+fp+tn+fn}$.
- F1-score, or positive agreement (PA), is computed as $\frac{2pr}{p+r}$ or $\frac{2tp}{2tp+fp+fn}$. F1-score measures the harmonic mean between recall (r) and precision (p) values. As in an AFA system, positive

samples are typically outnumbered by negative ones, F1-score is able to tell the performance on correct predictions on positive samples. The complement metric of F1-score is negative agreement (NA) [93], computed as $\frac{2tn}{2tn+fp+fn}$, which evaluates the produced solution by its “harmonic agreement” of the negative class.

- Event-based F1 [64] evaluate detection performance at event-level. An “event” is defined as a max continuous period with an action. In this sense, F1-event captures the “harmonic agreement”, *i.e.*, $\frac{2 \cdot ER \cdot EP}{ER + EP}$, by measuring the event-based recall ER and event-based precision EP . We refer more details to the original paper [64].
- Area under the ROC curve (AUC), as a type of ranking-based metrics, is computed as $AUC = \frac{S_p - n_p(n_n + 1)/2}{n_p n_n}$, where S_p is the sum of all positive samples ranked; n_p and n_n denote the number of positive and negative samples respectively. AUC was proven heoretically and empirically better than the accuracy metrics for evaluating classifier performance [115].

■

A Transductive Framework for Personalized Facial Expression Analysis

“We cannot solve our problems with the same thinking we used when we created them.”

Albert Einstein

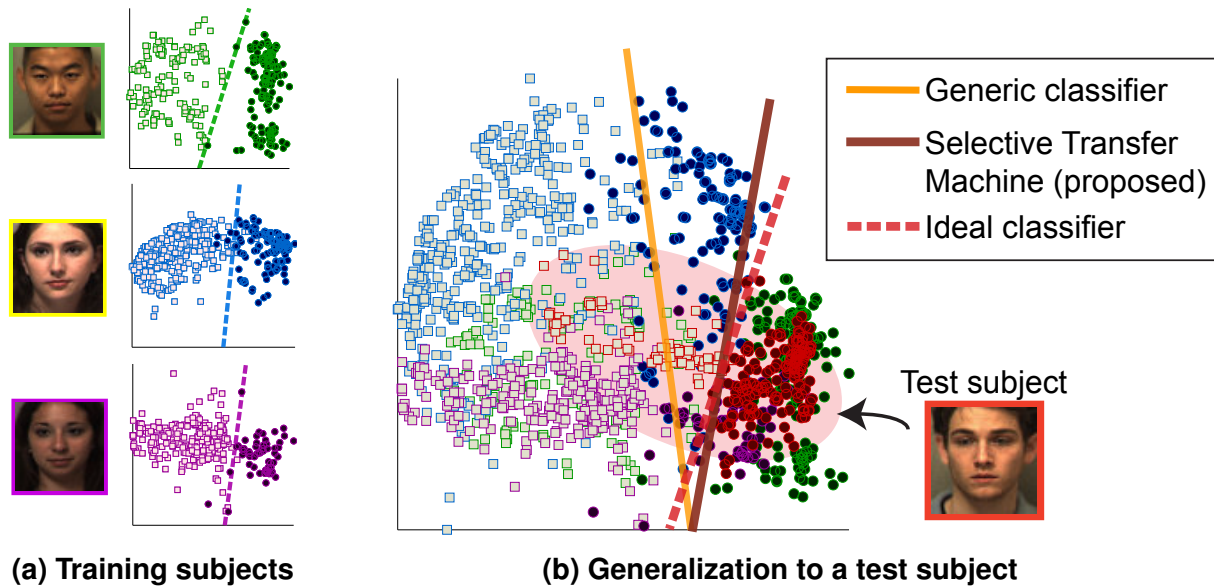


Figure 3.1: An illustration of the proposed transductive approach, **Selective Transfer Machine (STM)**: (a) 2D PCA projection of positive (squares) and negative (circles) samples for a given AU (in this case AU12) for 3 subjects. An ideal classifier separates AU 12 nearly perfectly for each subject. (b) A generic classifier trained on all 3 subjects generalizes poorly to a new person (*i.e.*, test subject) due to individual differences between the 3-subject training set and the new person. STM personalizes a generic classifier and reliably separates an AU for a new subject.

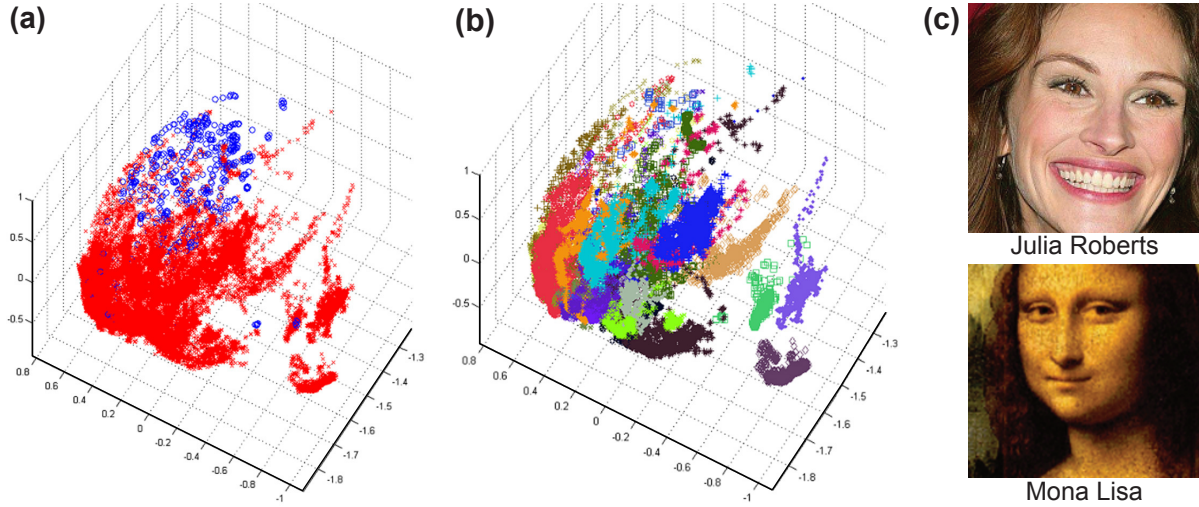


Figure 3.2: Visualization of samples from the RU-FACS dataset [10] in 3D eigenspace: colors/markers indicate different (a) positive/negative classes, and (b) subjects. (c) shows an example that illustrates individual differences between Julia Roberts and Mona Lisa. (best viewed in color).

As summarized in the previous chapter, automatic analysis of facial actions confronts a number of challenges. These, in particular, include changes in pose, scale, illumination, occlusion, and individual differences in face shape, texture, and behavior. Face shape and texture differ between and within sexes; they differ with ethnic and racial backgrounds, age or developmental level, exposure to the elements, and in the base rates with which they occur. For example, some people smile broadly and frequently; others rarely or only with smile controls, which counteract the upward pull of the zygomatic major on the lip corners. These and other sources of variation represent considerable challenges for computer vision. Then there is the challenge of automatically detecting facial actions that require significant training and expertise in humans [248].

In this chapter, we will address facial action analysis with the issues caused by individual differences. In particular, we identify that the sources of biases can be traced back to the person-specific representation described in the feature space. In other words, standard features extracted from a face can encode information about not only facial expressions but also person identity. Fig. 3.2(b) shows samples (frames in video) colored in terms of person identities, which exhibit clear clustering effects for samples of the same subject. One possible approach might be “neutralizing” these samples with neutral faces from each subject, yet such faces are not always available. Without further treatment on the features, we will borrow the idea from *transductive learning* to tackle distributional drift between individuals.

3.1 Persona-Specific Biases for Facial Expression Analysis

To address the challenges caused by individual differences, previous work has focused on identifying optimal feature representations and classifiers. Interested readers may refer to [164, 189, 206, 242] for comprehensive reviews. While improvements have been achieved, a persistent shortcoming of existing systems is that they fail to generalize well to previously unseen, or new, subjects. One way to cope with this problem is to train and test separate classifiers on each subject (*i.e.*,

person-specific classifier). Fig. 3.1(a) shows a real example of how a simple linear person-specific classifier can separate the positive samples of AU12 (lip corner puller, seen in smiling) from the negative ones. When ample training data are available, a *person-specific* classifier approaches an *ideal classifier*, one that best separates actions for the test subject.

A problem with person-specific classifiers is that sufficient quantity of training data is usually unavailable. In part for this reason, most approaches seek to use training data from multiple subjects in the hope to compensate for subject biases. However, as shown in Fig. 3.1(b), when a classifier is trained on all training subjects and tested on an unknown subject, its generalizability may disappoint. When a classifier is trained and tested in this manner, we refer it as to *generic classifier*. Because person-independent classifiers typically are not feasible, generic classifiers are most commonly used.

We identify that impaired generalizability occurs in part because of individual differences among subjects. Fig. 3.2 illustrates this phenomenon on real data in a 3-D eigenspace. One can observe that when the data are interpreted as positive and negative classes in Fig. 3.2(a), they could be very difficult to separate without overfitting. When the data are interpreted as subjects in Fig. 3.2(b), the grouping effect becomes clear and echoes with our conjecture on individual differences. In this example, these differences include sex, skin color and texture, illumination, and other ways in which people vary. Our guiding hypothesis is that such person-specific bias causes standard generic classifiers to perform worse on some subjects than others.

To mitigate the influence of individual biases, this paper explores the idea of *personalizing* a generic classifier for facial expression analysis. Given a common observation that test videos usually come from only a single subject, we assume the test distribution can be approximated by certain frames from training subjects. The problem of personalizing a generic classifier then is formulated as training a classifier on selected training samples, while reducing the discrepancy between distributions of selected training samples and test ones. In this way, generic classifiers can adapt to an unseen test subject without test labels. We term this transductive approach Selective Transfer Machine (STM). The major contributions of this work include:

- Based on both qualitative observations and empirical findings, individual differences attenuate AU detection. To address this problem, we introduce *Selective Transfer Machine (STM)*. STM is a personalization approach that reduces mismatch between feature distributions of training and test subjects. We propose an effective and robust procedure to optimize STM in its primal form.
- Considering that many applications afford labeled test data, we introduce a useful extension of STM, termed *L-STM*, to make use of labeled target data. This extension shows considerable performance improvement in situations for which some labeled test data exist.
- To evaluate STM, we conduct comprehensive experiments using *within-subject*, *cross-subject*, and *cross-dataset* scenarios on four benchmark datasets. We test STM for both AU detection and detection of holistic expressions.
- For test subjects, some training samples are more instrumental than others. We can identify those training samples using STM. The effectiveness of STM scales as the number of training subjects increases.

This chapter is organized as follows. Secs. 3.2–3.4 describes the STM model, optimization algorithm, and theoretical rationale. Sec. 3.5 introduces L-STM, an STM extension that utilizes labeled test data. Sec. 3.6 considers similarities and differences between STM and related meth-

ods. Sec. 5.6 evaluates STM and alternatives for AU and holistic expression detection. Sec. 3.8 concludes the paper with remarks and future work.

3.2 Selective Transfer Machine (STM)

This section describes the proposed Selective Transfer Machine (STM) for personalizing a generic classifier. Unlike previous cross-domain methods [6, 71, 133, 268], STM requires no labels from a test subject. We will use Support Vector Machine (SVM) as classifier due to its popularity for AU detection [36, 125, 222].

Problem formulation: Recent research and applications in automatic facial expression analysis involve video, which provides a wide sampling of facial appearance change. We assume the distribution of a subject’s appearance can be estimated by certain video frames. Based on this assumption, the main idea of STM is to re-weight training samples (*i.e.*, frames) to form a distribution closer to the test distribution. Classifiers trained on the re-weighted training samples are likely to generalize to the test subject.

Let us denote the training set as $\mathcal{D}^{\text{tr}} = \{\mathbf{x}_i, y_i\}_{i=1}^{n_{\text{tr}}}$, $y_i \in \{+1, -1\}$ (see notation¹). For notational simplicity, we stack 1 in each data vector \mathbf{x}_i to compensate for the offset, *i.e.*, $\mathbf{x}_i \in \mathbb{R}^{d+1}$. We formulate STM as minimizing the objective:

$$g(f, \mathbf{s}) = \min_{f, \mathbf{s}} R_f(\mathcal{D}^{\text{tr}}, \mathbf{s}) + \lambda \Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}), \quad (3.1)$$

where $R_f(\mathcal{D}^{\text{tr}}, \mathbf{s})$ is the SVM empirical risk defined on the decision function f , and training set \mathcal{D}^{tr} with each instance weighted by $\mathbf{s} \in \mathbb{R}^{n_{\text{tr}}}$. Each entry s_i corresponds to a positive weight for a training sample \mathbf{x}_i . $\Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}})$ measures training and test distribution mismatch as a function of \mathbf{s} . The lower the value of $\Omega_{\mathbf{s}}$, the more similar the training and the test distributions are. $\lambda > 0$ is a tradeoff between the risk and the distribution mismatch. The goal of the STM is to jointly optimize the decision function f as well as the selective coefficient \mathbf{s} , such that the resulting classifier can alleviate person-specific biases.

Penalized SVM: The first term in STM, $R_f(\mathcal{D}^{\text{tr}}, \mathbf{s})$, is the empirical risk of a penalized SVM, where each training instance is weighted by its relevance to the test data. In the following, we denote $\mathbf{X} \equiv \mathbf{X}^{\text{tr}}$ for notational simplicity unless further referred. The linear penalized SVM has the target decision function in the form $f(\mathbf{x}) = \mathbf{w}^{\top} \mathbf{x}$ and minimizes:

$$R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n_{\text{tr}}} s_i L^p(y_i, \mathbf{w}^{\top} \mathbf{x}_i), \quad (3.2)$$

where $L^p(y, \cdot) = \max(0, 1 - y \cdot)^p$ ($p = 1$ stands for hinge loss and $p = 2$ for quadratic loss). In general, L could be any loss function. The unconstrained linear SVM in (3.2) can be extended to a nonlinear version by introducing a kernel matrix $\mathbf{K}_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ corresponding to a kernel function k induced from some nonlinear feature mapping $\varphi(\cdot)$. Using the representer theorem [27], the nonlinear decision function can be represented $f(\mathbf{x}) = \sum_{i=1}^{n_{\text{tr}}} \beta_i k(\mathbf{x}_i, \mathbf{x})$, yielding the nonlinear

¹ Bold capital letters denote a matrix \mathbf{X} ; bold lower-case letters denote a column vector \mathbf{x} . \mathbf{x}_i represents the i th column of the matrix \mathbf{X} . All non-bold letters represent scalars. x_j denotes the scalar in the j th element of \mathbf{x} . $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is an identity matrix.

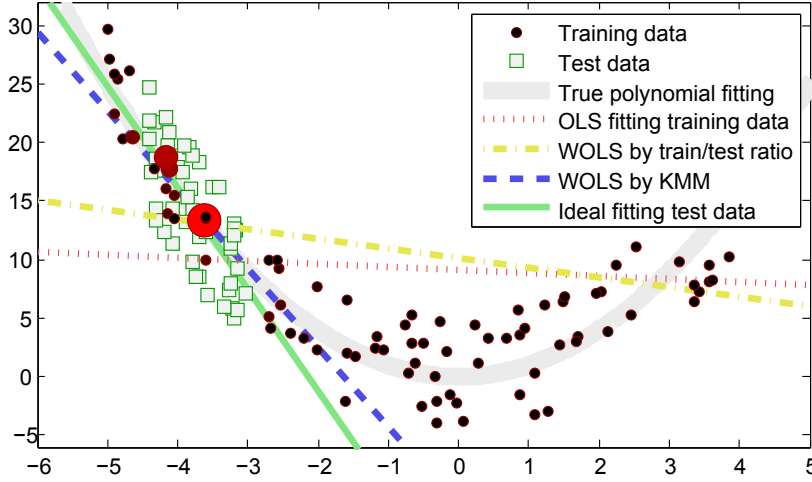


Figure 3.3: Fitting a line to a quadratic function using KMM and other re-weighting methods. The larger size (more red) of training data, the more weight KMM adopted. As can be observed, KMM puts higher weights in the training samples closer to the test ones. Compared to standard OLS or WOLS, KMM allows to better approximation for the test data.

penalized SVM:

$$R_{\beta}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \beta^{\top} \mathbf{K} \beta + C \sum_{i=1}^{n_{\text{tr}}} s_i L^p(y_i, \mathbf{k}_i^{\top} \beta), \quad (3.3)$$

where $\beta \in \mathbb{R}^{n_{\text{tr}}}$ is the expansion coefficient and \mathbf{k}_i is the i th column of \mathbf{K} . Unlike most standard solvers, we train the penalized SVM in the primal due to its simplicity and efficiency. Through the unconstrained primal problems, we applied Newton's method with quadratic convergence [27]. Details are given in Sec. 3.3.

Distribution mismatch: The second term in STM, $\Omega_s(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}})$, imitates domain mismatch and aims to find a re-weighting function that minimizes the discrepancy between the training and the test distributions. In previous cross-domain learning methods, the re-weighting function may be computed by separately estimating the densities and then the weights (*e.g.*, [228]). However, this strategy could be prone to error while taking the ratio of estimated densities [228].

Here we adopt the Kernel Mean Matching (KMM) [103] method to reduce the difference between the means of the training and the test distributions in the Reproducing Kernel Hilbert Space \mathcal{H} . KMM computes the instance re-weighting s_i that minimizes:

$$\Omega_s(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}) = \left\| \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} s_i \varphi(\mathbf{x}_i^{\text{tr}}) - \frac{1}{n_{\text{te}}} \sum_{j=1}^{n_{\text{te}}} \varphi(\mathbf{x}_j^{\text{te}}) \right\|_{\mathcal{H}}^2. \quad (3.4)$$

Introducing $\kappa_i := \frac{n_{\text{tr}}}{n_{\text{te}}} \sum_{j=1}^{n_{\text{te}}} k(\mathbf{x}_i^{\text{tr}}, \mathbf{x}_j^{\text{te}})$, $i = 1, \dots, n_{\text{tr}}$, that captures the closeness between training and each test sample, solving \mathbf{s} in (3.4) can be rewritten as a quadratic programming (QP):

$$\begin{aligned} \min_{\mathbf{s}} \quad & \frac{1}{2} \mathbf{s}^{\top} \mathbf{K} \mathbf{s} - \boldsymbol{\kappa}^{\top} \mathbf{s}, \\ \text{s. t.} \quad & s_i \in [0, B], \left| \sum_{i=1}^{n_{\text{tr}}} s_i - n_{\text{tr}} \right| \leq n_{\text{tr}} \epsilon, \end{aligned} \quad (3.5)$$

where B defines a scope bounding discrepancy between probability distributions P_{tr} and P_{te} ($B = 1000$ in our case). For $B \rightarrow 1$, one obtains an unweighted solution where all $s_i = 1$. The second

Algorithm 1: Selective Transfer Machine (STM)

Input : $\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}$, parameters C, λ
Output: Inferred test labels \mathbf{y}^* for test data
 1 Initialize training loss $\ell^p \leftarrow 0$;
 2 **while** *not converged* **do**
 3 Update the instance-wise re-weighting \mathbf{s} by solving the QP in (3.6);
 4 Update the decision function f and training loss ℓ^p by solving the penalized SVM in (3.2) or (3.3);
 5 Infer test labels by $\mathbf{y}^* \leftarrow f(\mathbf{X}^{\text{te}})$

constraint ensures the weighted samples to be close to a probability distribution [103]. Observe in (3.5) that larger κ_i leads to larger s_i when the objective is minimized. This matches our intuition to put higher selection weights on the training samples that are more likely to resemble the test distribution.

A major benefit from KMM is a direct importance estimation without estimating training and test densities. Compared to existing approaches, with proper tuning of kernel bandwidth, KMM shows the lowest importance estimation error and robustness to input dimension and the number of training samples, as suggested in [228]. Fig. 3.3 illustrates its effect on a synthetic data. As shown, KMM can estimate the ideal fitting well, while standard Ordinary Least Square (OLS) and Weighted OLS (WOLS) with training/test ratio lead to suboptimal prediction.

3.3 Optimization for STM

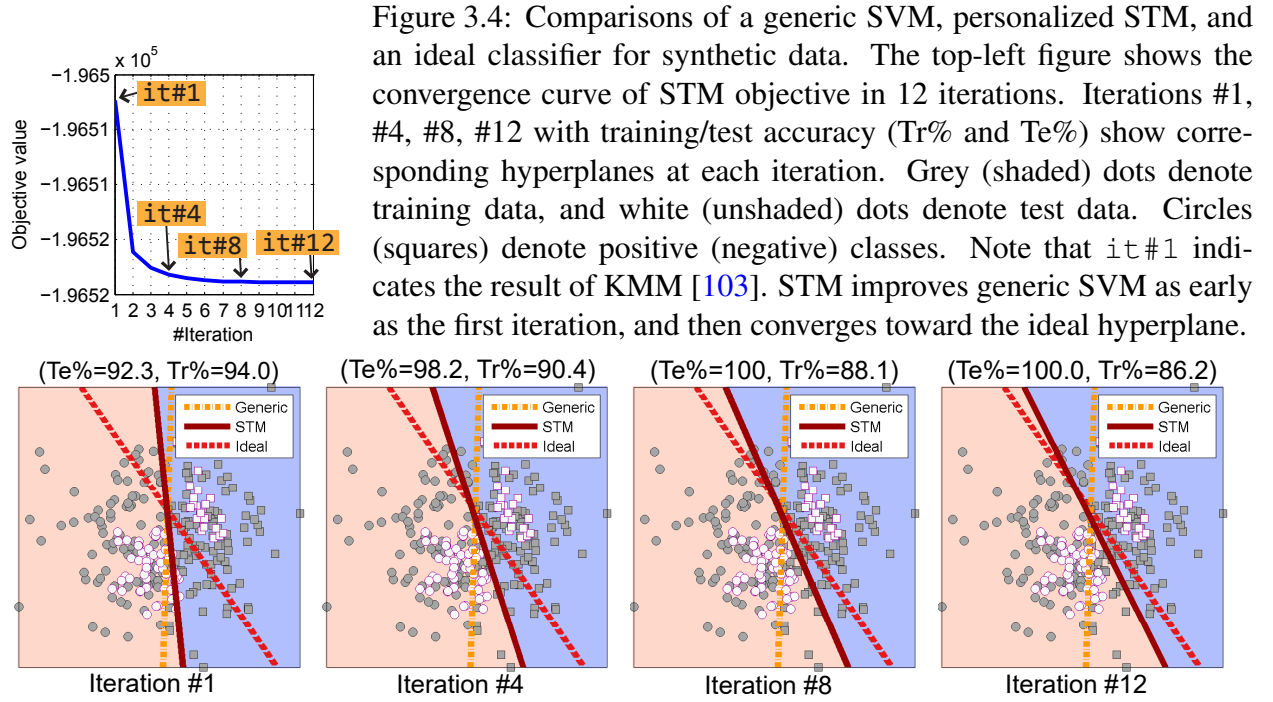
To solve Eq. (5.3), we adopt the Alternate Convex Search [88] that alternates between solving the decision function f and the selective coefficient \mathbf{s} . Note that the objective in (5.3) is biconvex: Convex in f when \mathbf{s} is fixed (f is quadratic and L^p is convex), and convex in \mathbf{s} when f is fixed (since $\mathbf{K} \succeq 0$). Under these conditions, the alternate optimization approach is guaranteed to monotonically decrease the objective function. Because the function is bounded below, it will converge to a critical point. Algorithm 1 summarizes the STM algorithm. Once the optimization is done, f is applied to perform the inference for test images. Below we detail the two steps in the alternate algorithm.

Minimizing over \mathbf{s} : Denote the training losses as $\ell_i^p := L^p(y_i, f(\mathbf{x}_i))$, $i = 1, \dots, n_{\text{tr}}$. The optimization over \mathbf{s} can be rewritten into the following QP:

$$\begin{aligned}
 \min_{\mathbf{s}} \quad & \frac{1}{2} \mathbf{s}^\top \mathbf{K} \mathbf{s} + \left(\frac{C}{\lambda} \ell^p - \boldsymbol{\kappa} \right)^\top \mathbf{s} \\
 \text{s. t.} \quad & 0 \leq s_i \leq B, n_{\text{tr}}(1 - \epsilon) \leq \sum_{i=1}^{n_{\text{tr}}} s_i \leq n_{\text{tr}}(1 + \epsilon).
 \end{aligned} \tag{3.6}$$

Since $\mathbf{K} \succeq 0$ by definition, (3.6) has only one global optimum. To make the algorithm numerically stable, we add a ridge σ on the diagonal so that $\mathbf{K} \succeq \sigma \mathbf{I}_{n_{\text{tr}}}$ ($\sigma = 10^{-8}$ in our case).

Note that the procedure here is different from the original KMM in terms of *weight refinement*: In each iteration \mathbf{s} will be refined through the training loss ℓ^p from the penalized SVM. This effect can be observed from minimizing the second term in (3.6): Larger ℓ^p leads to smaller \mathbf{s} to keep



the objective small. This effectively reduces the selection weights of incorrectly classified training samples. On the contrary, KMM uses no label information and thus is incapable of refining importance weights. Introducing training losses helps preserve the discriminant property of the new decision boundary and hence leads to a more robust personalized classifier. From this perspective, KMM can be treated as a special case as the first iteration in the STM framework.

Figs. 3.4 and 3.5 illustrate the iterative effects on synthetic data for personalizing a classifier for binary and multi-class classification, respectively. In `it#1`, the hyperplane estimated by KMM generalizes poorly partially due to its unsupervised nature. On the other hand, STM considers training loss and the weightings, and thus encourages relevant training samples to be well classified. As can be observed, as the iterations proceed, the separation hyperplane approaches toward the ideal one for the target data. Below we will focus discussion on binary classification and refer to the multi-class extension to [279] for more details.

Minimizing over f : Let sv indicate the index set of support vectors, and n_{sv} the number of support vectors. In the case of training loss ℓ^2 being quadratic, the gradient and Hessian of the linear penalized SVM in (3.2) can be written as:

$$\nabla_{\mathbf{w}} = \mathbf{w} + 2\mathbf{C}\mathbf{X}\mathbf{S}\mathbf{I}^0(\mathbf{X}^\top \mathbf{w} - \mathbf{y}), \quad (3.7)$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + 2\mathbf{C}\mathbf{X}\mathbf{S}\mathbf{I}^0\mathbf{X}^\top, \quad (3.8)$$

where $\mathbf{S} = \text{diag}(\mathbf{s}) \in \mathbb{R}^{n_{tr} \times n_{tr}}$ denotes the re-weighting matrix, $\mathbf{y} \in \mathbb{R}^{n_{tr}}$ the label vector, and $\mathbf{I}^0 \in \mathbb{R}^{n_{tr} \times n_{tr}}$ the proximity identity matrix with the first n_{sv} diagonal elements being 1 and the rest being 0. Similarly, the gradient with respect to the expansion coefficient β in (3.3) can be derived:

$$\nabla_{\beta} = \mathbf{K}\beta + 2\mathbf{C}\mathbf{K}\mathbf{S}\mathbf{I}^0(\mathbf{K}\beta - \mathbf{y}), \quad (3.9)$$

$$\mathbf{H}_{\beta} = \mathbf{K} + 2\mathbf{C}\mathbf{K}\mathbf{S}\mathbf{I}^0\mathbf{K}. \quad (3.10)$$

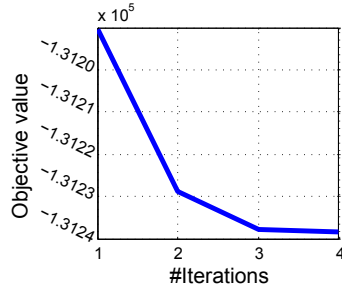
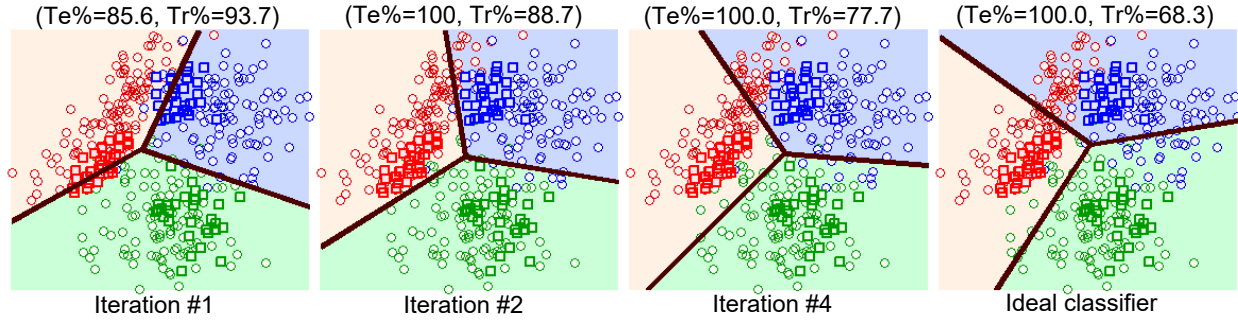


Figure 3.5: (Left to right) Convergence curve of a multi-class STM [279], separation hyperplane at iterations #1, #2, and #4, and the ideal classifier (SVM trained on the test samples). Circles (\circ) and squares (\square) indicate training and test samples, respectively. Te% and Tr% indicate the accuracy on test and training data. As can be seen, as iteration proceeds, the STM hyperplane approaches the ideal hyperplane, which perfectly separates the test samples.



Given the gradients and Hessians, the penalized SVM can be optimized by standard Newton's method or conjugate gradient.

Differentiable Huber loss: The L^1 (hinge) loss in standard SVMs are not differentiable, hampering its gradient and Hessian to be explicitly expressed and computed. Instead, we use the Huber loss [27] as a differentiable surrogate, *i.e.*, $L^1(y_i, f(\mathbf{x}_i)) \approx L_H(y_i \text{sign}(f(\mathbf{x}_i)))$. Note that any differential convex loss, *e.g.*, logistic loss and exponential loss, can be directly incorporated. The Huber loss can be defined as follows:

$$L_H(a) = \begin{cases} 0 & \text{if } a > 1 + h, \\ \frac{(1+h-a)^2}{4h} & \text{if } |1-a| \leq h, \\ 1-a & \text{otherwise,} \end{cases} \quad (3.11)$$

where h is a parameter of choice. Fig. 3.6 shows the influence of h in comparison to the L^1 and L^2 loss. As can be observed, L_H approaches the hinge loss when $h \rightarrow 0$. As indicated in [27], there is no clear reason to prefer the hinge loss because replacing the hinge loss with Huber loss does not influence much the results. With the differentiable Huber loss, the gradient and Hessian with Huber loss for the penalized linear SVM can be obtained:

$$\nabla_{\mathbf{w}} = \mathbf{w} + \frac{C}{2h} \mathbf{X} \mathbf{S} \mathbf{I}^0 [\mathbf{X}^\top \mathbf{w} - (1+h)\mathbf{y}] - C \mathbf{X} \mathbf{S} \mathbf{I}^1 \mathbf{y}, \quad (3.12)$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + \frac{C}{2h} \mathbf{X} \mathbf{S} \mathbf{I}^0 \mathbf{X}^\top, \quad (3.13)$$

and for the penalized nonlinear SVM:

$$\nabla_{\boldsymbol{\beta}} = \mathbf{K} \boldsymbol{\beta} + \frac{C}{2h} \mathbf{K} \mathbf{S} \mathbf{I}^0 [\mathbf{K} \boldsymbol{\beta} - (1+h)\mathbf{y}] - \mathbf{K} \mathbf{I}^1 \mathbf{y}, \quad (3.14)$$

$$\mathbf{H}_{\boldsymbol{\beta}} = \mathbf{K} + \frac{C}{2h} \mathbf{K} \mathbf{S} \mathbf{I}^0 \mathbf{K}, \quad (3.15)$$

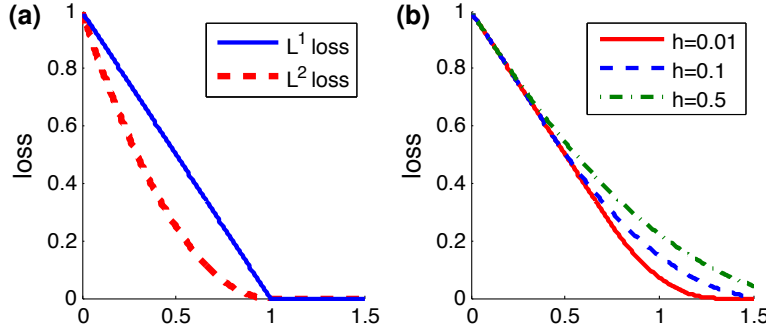


Figure 3.6: Loss functions used in this study: (a) L^1 and L^2 loss, and (b) Huber loss (a differentiable surrogate).

where $\mathbf{I}^1 \in \mathbb{R}^{n_{tr} \times n_{tr}}$ denotes the proximity identity matrix with the first n_{sv} diagonal elements being 0, followed by n_ℓ (the number of points in the linear part of the Huber loss) elements of ones, and the rest being 0. With the derived gradient and Hessian, we are able to optimize for f with quadratic convergence using standard Newton method. Please refer to Appendix A for the full derivation.

Avoid computing Hessian and its inversion: The main drawback in applying Newton method is on computing the Hessian, which has a computational cost $\mathcal{O}(d^2 n_{sv})$ and its inversion $\mathcal{O}(d^3)$. Although the mathematical expression of Newton's direction $-\mathbf{H}^{-1} \nabla$ is better understood, it does not imply that the inverse of Hessian should be computed. We avoid computing the Hessian and its inversion by solving the linear system $\mathbf{H}^{(k)} p^{(k)} = -\nabla^{(k)}$, and then iteratively update $\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} + p^{(k)}$. We setup the stopping criterion when the ratio of objective value decrease to the objective value of the first iteration is less than a certain small value (in our case 10^{-8}).

3.4 Theoretical Rationale

This section analyzes two important properties of STM, *bi-convexity* and *boundedness*, based on the techniques developed for biconvex optimization [99]. Then we justify the convergence of the Alternate Convex Search algorithm, which we used for solving STM, in terms of both objective value and optimization variables.

3.4.1 Properties of STM

We start by showing that STM is a biconvex problem.

Property 1. (Bi-convexity) *Selective Transfer Machine (STM) in (5.3) is a biconvex optimization problem.*

Proof. Denote the decision variable of f as $\mathbf{w} \in W \subseteq \mathbb{R}^d$ and the selection coefficient $\mathbf{s} \in S \subseteq \mathbb{R}^{n_{tr}}$, where W and S are two non-empty convex sets. Let $Z \subseteq W \times S$ be the solution set on $W \times S$; $Z_{\mathbf{w}}$ and $Z_{\mathbf{s}}$ be the subsets when \mathbf{w} and \mathbf{s} are given respectively. Because $Z_{\mathbf{s}}$ is convex for every $\mathbf{w} \in W$ (\mathbf{w} and L^p are convex; $s_i \in [0, B]$ are non-negative) and $Z_{\mathbf{w}}$ is convex for every $\mathbf{s} \in S$ ($\Omega_{\mathbf{s}}$ is QP and $\mathbf{K} \succeq 0$), the solution set Z is a *biconvex set*. Hence STM can be rewritten in the standard form of *biconvex optimization problem* [5]: $\min_{\mathbf{w}, \mathbf{s}} \{g(\mathbf{w}, \mathbf{s}) : (\mathbf{w}, \mathbf{s}) \in Z\}$. \square

Property 2. (Boundedness) *The STM optimization problem in Problem (5.3) is bounded from below.*

Algorithm 2: Alternate Convex Search Algorithm

```

1 Step 1: Choose a starting point  $\mathbf{z}_0 \leftarrow (\mathbf{w}_0, \mathbf{s}_0) \in Z$ ;
2 Set  $t \leftarrow 0$ ;
3 while not converged do
4   Step 2: Solve the convex optimization problem for fixed  $\mathbf{w}_t$  :
       $\mathbf{s}_{t+1} \leftarrow \min_{\mathbf{s}} \{g(\mathbf{w}_t, \mathbf{s}), \mathbf{s} \in Z_{\mathbf{w}_t}\}$ ;
5   Step 3: Solve the convex optimization problem for fixed  $\mathbf{s}_{t+1}$ :
       $\mathbf{w}_{t+1} \leftarrow \min_{\mathbf{w}} \{g(\mathbf{w}, \mathbf{s}_{t+1}), \mathbf{w} \in Z_{\mathbf{s}_{t+1}}\}$ ;
6   Step 4: Set  $\mathbf{z}_{t+1} \leftarrow (\mathbf{w}_{t+1}, \mathbf{s}_{t+1})$ ;
7   Set  $t \leftarrow t + 1$ ;
8 end
    
```

Proof. The boundedness can be observed from two aspects: (1) R_f is bounded due to the quadratic term in f and non-negative \mathbf{s} and L^p . (2) Ω_s is bounded since \mathbf{K} is positive semi-definite. \square

Following the same proof line, the above properties can be also shown for nonlinear STM defined with Eq. (3.3).

3.4.2 Algorithm

The following analysis mimics directly Sec. 4 in [99]. We present the key steps for proving the convergence and refer to more details on this style of proof in [99].

Alternate Convex Search: To solve the biconvex STM problem, a standard and popular approach is to exploit its convex substructure. We used the Alternate Convex Search (ACS) algorithm [259], a special case of *Block-Relaxation Methods*, by alternatively solving the convex subproblems. For explanation convenience, we recall the ACS algorithm in Algorithm 2.

Denote $\mathbf{z} = (\mathbf{w}, \mathbf{s})$ as the solution variable. As mentioned in Sec. 3.3, STM can be seen as initializing \mathbf{s}_0 using KMM, or simply as a vector of ones, and then solve the classifier \mathbf{w}_1 as an unweighed SVM. As will be discussed below and in Sec. 3.7.5, the permutation of order does not influence the convergence. For Step 4, there are several ways to determine the stopping criterion. Here, we used the relative decrease of \mathbf{z} compared to the last iteration. Below we discuss the convergence properties in terms of objective value (*i.e.*, the difference between $g(\mathbf{z}_t)$ and $g(\mathbf{z}_{t-1})$ of two consecutive iterations t and $t - 1$), and the variables (*i.e.*, the difference between \mathbf{z}_t and \mathbf{z}_{t-1}).

Convergence: Recall that W and S are two non-empty sets, and $Z \subseteq W \times S$ is a biconvex set on $W \times S$. We firstly show the convergence of the sequence of objective value $\{g(\mathbf{z}_t)\}_{t \in \mathbb{N}}$, and then convergence of the sequence of the variables $\{\mathbf{z}_t\}_{t \in \mathbb{N}}$.

Theorem 1. *Let the STM objective function be $g : Z \rightarrow \mathbb{R}$. Then the sequence of objective value $\{g(\mathbf{z}_t)\}_{t \in \mathbb{N}}$ generated by ACS converges monotonically.*

Proof. The sequence $\{g(\mathbf{z}_t)\}_{t \in \mathbb{N}}$ generated by Algorithm 2 decreases monotonically, since $g(\mathbf{w}^*, \mathbf{s}^*) \leq g(\mathbf{w}, \mathbf{s}^*), \forall \mathbf{w} \in Z_{\mathbf{s}^*}$ and $g(\mathbf{w}^*, \mathbf{s}^*) \leq g(\mathbf{w}^*, \mathbf{s}), \forall \mathbf{s} \in Z_{\mathbf{w}^*}$. In addition, Property 2 shows g is bounded from below. According to Theorem 4.5 in [99], the sequence $\{g(\mathbf{z}_t)\}_{t \in \mathbb{N}}$ converges to a limit real value. \square

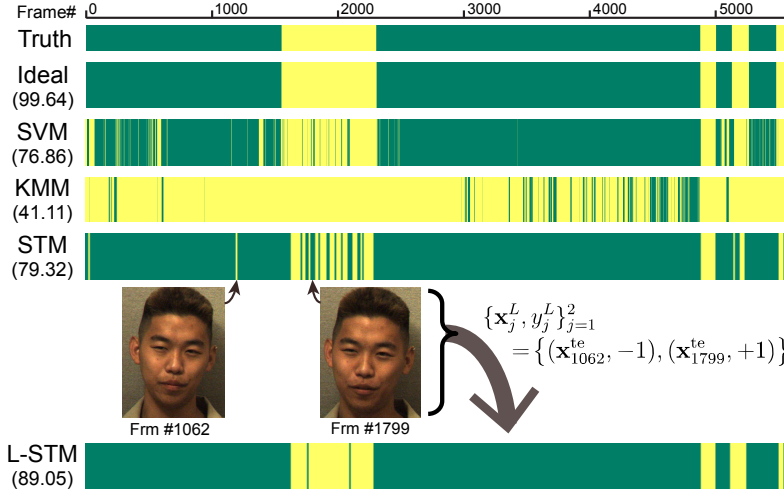


Figure 3.7: Comparison of different methods on the RU-FACS dataset. Light yellow (dark green) indicates AU 12 presense (absense) of Subject 12. The numbers in the parentheses are F1 scores. Two misclassified frames of STM were chosen and fed into L-STM with correct labels.

Theorem 1 only tells the convergence of the sequence $\{g(\mathbf{z}_t)\}_{t \in \mathbb{N}}$ but not of the sequence $\{\mathbf{z}_t\}_{t \in \mathbb{N}}$. See Example 4.3 in [99] where $\{g(\mathbf{z}_t)\}_{t \in \mathbb{N}}$ converge but $\{\mathbf{z}_t\}_{t \in \mathbb{N}}$ diverge. The following states the condition for convergence of $\{\mathbf{z}_t\}_{t \in \mathbb{N}}$.

Theorem 2. Let W and S be closed sets, and $\mathbf{z}_t = (\mathbf{w}_t, \mathbf{s}_t)_{t \in \mathbb{N}}$ where $\mathbf{w}_t \in W$ and $\mathbf{s}_t \in S$. The sequence of variables $\{\mathbf{z}_t\}_{t \in \mathbb{N}}$ generated by ACS converge to $\mathbf{z}^* \in W \times S$.

Proof. This can be proved using Theorem 4.7 in [99]. \square

3.5 STM with Labeled Target Data (L-STM)

As discussed above, STM requires no labels from the target subject to obtain the personalized classifier. Nevertheless, in many problems one might collect partially labeled data from the target domain, or acquire additional guidance with a few manual labels. Such labels can be considered as the only reference to the target subject and aid the determination of the personalized classifier. This section describes an inductive extension of STM, termed *L-STM*, to adapt target labels for personalizing a classifier.

Given the target data and their labels as $\mathcal{D}^L = \{\mathbf{x}_j^L, y_j^L\}_{j=1}^{n_L}$, $y_j^L \in \{+1, -1\}$, $0 \leq n_L \leq n_{\text{te}}$, we formulate L-STM by introducing an additional regularization term $\Omega_L(\mathcal{D}^L)$ to (5.3):

$$\min_{f, \mathbf{s}} R_f(\mathcal{D}^{\text{tr}}, \mathbf{s}) + \lambda \Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}) + \lambda_L \Omega_L(\mathcal{D}^L), \quad (3.16)$$

where $\lambda_L > 0$ is a tradeoff parameter. A choice of large λ_L makes sure the labeled target data are correctly classified. The goal of $\Omega_L(\mathcal{D}^L)$ is to regulate the classification quality on the labeled target data. In this paper, we define $\Omega_L(\mathcal{D}^L) = \sum_{j=1}^{n_L} L^p(y_j^L, f(\mathbf{x}_j^L))$. Note that an L^2 loss here is analogous to the regularization in Least Square SVM [251], which performs comparably with SVM using the hinge loss and has been shown to relate to a ridge regression approach for binary classification, such as our task at hand. Because $\Omega_L(\mathcal{D}^L)$ is convex in f , problem (3.16) is still a biconvex optimization problem, and thus the ACS algorithm can be directly applied.

We show that solving problem (3.16) is equivalent to solving the original STM using a training set augmented with weighted labeled target data. We demonstrate the use of L^2 loss on linear SVM,

Methods	Importance re-weight	Weight refine	Convexity	Labeled target data
SVM-KNN [280]	×	×	NA	×
T-SVM [52]	×	×	non-convex	×
KMM [103]	✓	×	convex	×
DA-SVM [20]	×	✓	non-convex	×
DT-MKL [71]	×	×	jointly convex	optional
DAM [72]	×	×	convex	optional
STM (proposed)	✓	✓	bi-convex	optional

✓: included, ×: omitted, NA: not applicable

Table 3.1: Compare STM with related transductive transfer learning methods (in terms of their consideration of different learning factors)

while different choices of loss functions (*e.g.*, L^1) and classifier types (*e.g.*, nonlinear SVM) can be applied. Specifically, updating for \mathbf{s} remains the same process. For updating \mathbf{w} , one can again use Newton’s method by associated gradient and Hessian:

$$\nabla_{\mathbf{w}} = \mathbf{w} + \widehat{\mathbf{X}}\widehat{\mathbf{S}}(\widehat{\mathbf{X}}^\top \mathbf{w} - \widehat{\mathbf{y}}), \quad (3.17)$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + \widehat{\mathbf{X}}\widehat{\mathbf{S}}\widehat{\mathbf{X}}^\top, \quad (3.18)$$

where $\widehat{\mathbf{X}} = [\mathbf{X}^{\text{tr}} | \mathbf{X}^L]$ is the augmented set with labeled target data, $\widehat{\mathbf{S}} = \left[\begin{array}{c|c} 2C\mathbf{S}\mathbf{I}^0 & 0 \\ \hline 0 & \lambda_L \mathbf{I}_{n_L} \end{array} \right]$ is the augmented re-weighting matrix, and $\widehat{\mathbf{y}} = \left[\begin{array}{c} \mathbf{y} \\ \hline \mathbf{y}^L \end{array} \right]$ is the augmented labels.

The above equivalence is useful particularly for the scenario of AU detection, where the unlabeled videos are usually abundant with limited laborious FACS coding. L-STM allows users to add just a few frames to alleviate false detections significantly. Fig. 3.7 illustrates the benefits of L-STM over different methods. Light yellow (dark green) indicates positive (negative) frames for AU 12 on Subject 12 of the RU-FACS dataset. Top two rows show the ground truth and the detection result of the ideal classifier, respectively. The numbers in the parentheses indicate the F1 score. The third and fourth rows illustrate the detection of generic SVM and KMM. Both approaches produced many false detections due to the person-specific biases and the lack of weight refinement. STM, on the fifth row, greatly reduced false positives and produced a better F1 score. The last row shows the detection using L-STM with two misclassified frames from STM with correct labels. Using the label information on the target domain, L-STM boosted $\sim 10\%$ F1 score by using labels from *only two* frames. As we observed empirically, the more the labeled target data are introduced, the better L-STM approaches the ideal classifier.

3.6 Discussion on Related Work

A few related efforts use *personalized modeling* for facial expression analysis, *e.g.*, AU intensity estimation [201]. STM differs from them in how it accomplishes personalization. Chang and Huang [24] introduced an additional face recognition module and trained a neural network on the combination of face identities and facial features. Romera-Paredes *et al.* [197] applied multi-task learning to learn a group of linear models and then calibrated the models toward the target subject using target labels. By contrast, STM requires neither a face recognition module nor target labels.

Table 3.2: Content of different datasets

Datasets	#Sub	#Vid	#Frm/vid	Content	AU label	Expression label
CK+ [161]	123	593	~20	Neutral→peak	Per-video	Per-video
GEMEP-FERA [248]	7	87	20~60	Acting	Per-frame	Per-video
RU-FACS [10]	34	34	5000~8000	Interview	Per-frame	–
GFT [209]	720	720	~60,000	Social interaction	Per-frame	–

Motivated by covariate shift [227], Chen *et al.* [33] proposed transductive and inductive transfer algorithms for learning person-specific models. In their transductive setting, KL-divergence was used to estimate sample importance. However, STM models the domain mismatch using KMM [103], which with proper tuning, as implied in [228], yields better estimation.

The most related work is *transductive transfer learning*, which seeks to address domain shift problems without target labels. Table 3.1 summarizes the comparison. DT-MKL [71] simultaneously minimizes the MMD criterion [16] and a multi-kernel SVM. DAM [72] leverages a set of pre-trained base classifiers and solves for a test classifier that shares similar predictions with the base classifiers on unlabeled data. However, similar to T-SVM [126] and SVM-KNN [280], these methods treat training data uniformly. By contrast, KMM [103] and STM consider importance re-weighting, properly adjusting the importance for each training instance to move the decision function toward test data. KMM performs re-weighting only once while STM does so in an iterative manner. From this perspective, KMM can be viewed as an initialization of STM (see Sec. 3.3). In addition, STM uses training loss to refine instance weights in successive steps, thus being able to correct sub-optimal weights. DA-SVM [20] refines instance weights as a quadratic function decaying with iterations. However, DA-SVM may fail to converge due to its non-convexity, while STM is formulated as a bi-convex problem and thus assures convergence. Moreover, STM can be extended to tackle labeled target data, which greatly improves the performance.

3.7 Experiments

STM was evaluated in datasets that afforded inclusion of both posed and unposed facial expression, frontal versus variable pose, complexity (*e.g.*, interview versus 3-person interaction), and differences in numbers of subjects, the amount of video per subject, and men and women of diverse ethnicity. These factors are among the individual differences that adversely affect classifier performance in previous work [94]. To evaluate STM with respect to alternative approaches and scenarios, it was compared with a generic classifier, person-specific classifiers, and cross-domain classifiers using within-subject, cross-subject, and cross-dataset scenarios. Operational parameters for STM included initialization order, parameter choice, and domain size.

3.7.1 Dataset Description

We tested the algorithms on four diverse datasets that involve posed, acted, or spontaneous expressions, and vary in video quality, length, annotation, the number of subjects, and context, as summarized in Table 3.2 and illustrated in Fig. 3.8.

(1) **The extended Cohn-Kanade (CK+) dataset** [161] contains brief (approximately 20 frames on average) videos of posed and un-posed facial expressions of men and women of various ethnic

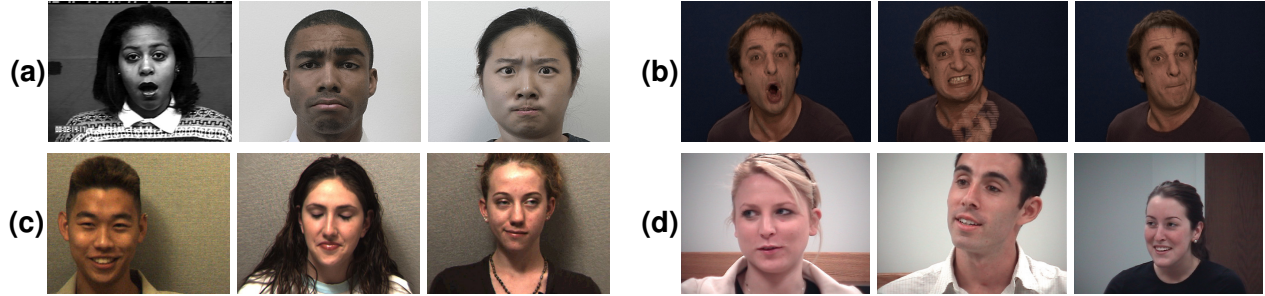


Figure 3.8: Example images from four datasets studied in this chapter: (a) CK+ [161], (b) GEMEP-FERA [248], (c) RU-FACS [10], and GFT[209] datasets.

backgrounds. Videos begin with a neutral expression and finish at the apex, or peak, which is annotated for AU and for holistic expression. Changes in pose and illumination are relatively small. Posed expressions from 123 subjects and 593 videos were used. Because STM requires some number of frames to estimate a test distribution, it is necessary to modify coding in CK+. In specific, we assume the last one-third frames share the same AU labels. We note that this may introduce some errors, compared to related methods that use only the peak frame for classification.

(2) **The GEMEP-FERA dataset** [248] consists of 7 portrayed emotion expressions by 10 trained actors. Actors were instructed to utter pseudo-linguistic phoneme sequences or a sustained vowel and display pre-selected facial expressions. Head pose is primarily frontal with some fast movements. Each video is annotated with AUs and holistic expressions. We used the GEMEP-FERA training set, which comprises 7 subjects (three of them men) and 87 videos.

(3) **RU-FACS dataset** [10] consists of video-recorded interviews of 100 young adults of varying ethnicity. Interviews are approximately 2.5 minutes in duration. Head pose is frontal with small to a moderate out-of-plane rotation. AU are coded if the intensity is greater than ‘A’, *i.e.*, lowest intensity on a 5-point scale. We had access to 34 of the interviews, of which video from 5 subjects could not be processed for technical reasons. Thus, the experiments reported here were conducted with data from 29 participants with more than 180,000 frames in total.

(4) **GFT** [209] consists of social interaction between 720 previously unacquainted young adults that were assembled into groups of three persons each and observed over the course of a 30-minute group formation task. Two minutes of AU-annotated video from 14 groups (*i.e.*, 42 subjects) was used in the experiments for a total of approximately 302,000 frames. Head pose varies over a range of about plus/minus 15-20 degrees [94]. For comparability with RU-FACS, we included AU 6, 9, 12, 14, 15, 20, 23 and 24.

Out of these datasets, CK+ is the most controlled, followed by GEMEP-FERA. Both include annotation for holistic expression and AU. GEMEP-FERA introduces variations in spontaneous expressions and large head movements but contains only 7 subjects. RU-FACS and GFT are both unposed and vary in complexity. RU-FACS is an interview context; GFT is a social interaction over a longer duration with greater variability. The first sets of experiments focus on CK+, GEMEP, and RU-FACS. GFT figures primarily in experiments on domain transfer between datasets and on the influence of numbers of subjects on performance.

AU	AUC			F1 Score		
	PS_1 -SVM	PS_2 -SVM	STM	PS_1 -SVM	PS_2 -SVM	STM
1	48.0	72.4	79.2	45.0	54.8	61.9
2	46.5	71.1	80.2	45.9	55.7	64.3
4	62.6	61.9	66.5	46.6	40.7	60.4
6	70.3	80.0	86.4	60.2	69.7	78.5
7	47.5	54.3	72.4	49.4	55.3	58.4
12	65.7	74.0	72.3	69.5	70.4	72.6
15	41.4	64.0	70.5	44.5	49.0	56.0
17	32.6	70.3	61.7	25.0	40.3	36.3
Av.	51.8	68.5	73.6	48.3	54.5	61.0

Table 3.3: Within-subject AU detection with STM and PS classifiers

3.7.2 Settings

Face tracking & registration: For CK+, FERA, and GFT, 49 landmarks were detected and tracked using the Supervised Descent Method (SDM) [266]. For RU-FACS, we used available AAM detection and 68 landmarks that were tracked in advance. Tracked landmarks were registered to a 200×200 template shape.

Feature extraction: Given a registered facial image, SIFT descriptors were extracted using 36×36 patches centered at selected landmarks (9 on the upper face and 7 on the lower face), because AUs occur only in local facial regions. The dimensionality of the descriptors was reduced by preserving 98% PCA energy.

AU selection & evaluation: Positive samples were taken as frames with an AU presence and negative samples as frames without an AU. We selected the 8 most commonly observed AUs across all datasets. To provide a comprehensive evaluation, we report both Area Under the ROC Curve (AUC) and F1 score. As AUC was originally designed for balanced binary classification tasks, F1 score, as the harmonic mean of precision and recall, could be more meaningful for imbalanced data, such as AUs.

Dataset split & validation: For a fair evaluation of training / test scenario, we used a leave-one-subject-out protocol. For each AU, we iteratively chose one subject for test and the remaining subjects for training and validation. For all iterations, we first identified the range of $\lambda \in \{2^{-10}, \dots, 2^{10}\}$ and $C \in \{2^{-10}, \dots, 2^{10}\}$ for which F1 score on the validation set was greatest. Then, we chose ones for which C was small. That is, we sought the parameters that maximize F1-score while preserving large margin of the decision boundary.

3.7.3 Action Unit (AU) Detection

We evaluated STM with generic and alternative approaches using three scenarios for AU detection: *within-subject*, *cross-subject*, and *cross-dataset*. We report results separately for each scenario.

Within-subject AU detection

A natural comparison with STM is a classifier trained on a single subject, also known as a *Person-Specific (PS)* classifier. A PS classifier can be defined in at least two ways. One, the more common definition, is a classifier trained and tested on the same subject. We refer to this usage as PS_1 . The other definition, referred to as PS_2 or *quasi-PS*, is a classifier that has been tested on a subject

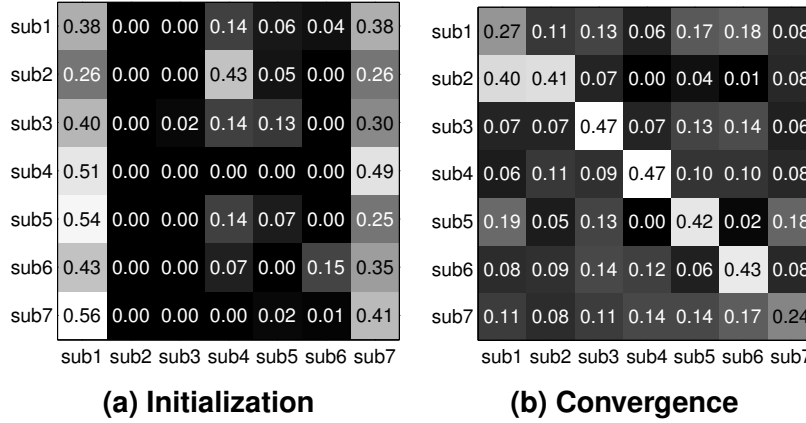


Figure 3.9: Selection percentage of STM for different subjects on (a) initialization and (b) convergence step. Each row sums to one and stands for a test subject.

included in the training set. The GEMEP-FERA competition [248] defined PS in this way. An SVM trained with PS_2 (PS_2 -SVM) is sometimes considered to be a generic classifier (*e.g.*, [162]). In our usage, we reserve the term “generic classifier” to the case in which training and test subjects are independent.

Here we compared STM with both PS_1 -SVM and PS_2 -SVM, and summarize the results in Table 3.3. In all, PS_1 -SVM shows the lowest AUC and F1. This outcome likely occurred because of the relatively small number of samples for individual subjects. Lack of sufficient training data for individual subjects is a common problem for person-specific classifiers. It is likely that PS_1 -SVM would have performed the best if the amount training data from the same subject is large enough. PS_2 -SVM achieved better AUC and F1 because it saw more training subjects. Overall, STM consistently outperformed both PS classifiers.

Selection ability of STM: Recall that PS_2 includes samples of the test subject in both training and test sets. Could STM improve PS_2 performance by selecting proper training samples? To answer this question, we employed PS_2 to investigate STM’s ability to select *relevant* training samples with respect to the test subject. Figure 3.9 shows the selection percentage of STM upon initialization and convergence. Each row sums to 1 and represents a test subject; each entry within one row denotes the percentage of selected samples from each training subject. For example, (a) shows the initialization phase that, when testing on Subject 2, 26% of training samples were selected from Subject 1. Upon convergence, as (b) shows, STM selected most training samples that belong to the target subject (higher diagonal value). Note that the selection percentages along the diagonal do not sum to 100% due to insufficient training samples for the target subject. However, STM was able to select relevant training samples, even from different subjects, to alleviate the mismatch between training and test distributions.

Cross-subject AU detection

Using a cross-subject scenario, *i.e.*, training and test subjects are independent in all iterations (a.k.a., *leave-one-subject-out*), we compared STM against various types of methods. Unsupervised domain adaptation methods are closest to STM. For comparisons we included Kernel Mean Matching (KMM) [103], Domain Adaptation SVM (DA-SVM) [20], and Subspace Alignment (SA) [87]. Multiple source domain adaptation methods serve as another natural comparison by treating each training subject as one source domains; we compared to the state-of-the-art DAM [72]. For baseline

Table 3.4: Cross-subject AU detection on RU-FACS dataset. “SA (NN|SVM)” indicates SA with NN and SVM, respectively.

AU	AUC								F1 Score							
	SVM	KMM	T-SVM	DA-SVM	SA (NN SVM)	DAM	STM		SVM	KMM	T-SVM	DA-SVM	SA (NN SVM)	DAM	STM	
1	72.0	74.0	72.0	77.0	41.2 82.0	82.6	83.9		40.8	37.7	37.4	35.5	20.9 24.2	11.3	55.3	
2	66.6	58.6	71.1	76.5	38.2 81.4	81.2	82.4		35.7	32.2	36.2	34.1	18.6 21.8	17.0	52.6	
4	74.8	62.2	50.0	76.4	24.5 71.1	51.3	82.4		25.2	14.5	11.2	35.3	5.7 5.8	2.9	30.4	
6	89.1	88.8	61.6	60.3	46.2 78.3	81.2	93.1		58.3	39.2	33.1	42.9	23.2 19.2	20.9	72.4	
12	86.7	87.0	86.7	84.4	55.9 86.1	93.1	92.3		61.9	63.0	62.6	71.4	37.5 38.6	36.6	72.3	
14	71.8	67.8	74.4	70.4	38.0 78.5	79.5	87.4		31.3	25.8	25.8	40.9	16.5 15.7	5.7	51.0	
15	72.5	68.8	73.5	58.1	37.7 79.2	71.8	86.1		32.3	29.5	32.3	34.9	10.1 8.8	3.2	45.4	
17	78.5	76.7	79.5	75.7	55.8 89.9	93.9	89.6		39.5	35.6	44.0	46.5	21.9 17.2	22.9	55.3	
Av.	76.5	72.3	71.1	72.3	42.2 80.8	79.3	86.3		40.6	37.3	40.6	42.7	19.3 18.9	15.1	54.3	

methods, we compared with linear SVMs and semi-supervised Transductive SVM (T-SVM) [52]. T-SVM, KMM, DAM and SA were implemented per the respective author’s webpage. Because STM requires no target labels, methods that use target labels for adaptation (*e.g.*, [59, 139, 203]) were not included.

All methods were compared in CK+ and RU-FACS with a few exceptions in CK+. In CK+, SA was ruled out because too few frames were available per subject to compute meaningful subspaces. DAM was also omitted in CK+ because it would be problematic to choose negative samples given the structure of the data (*i.e.*, pre-segmented positive examples). In training, a Gaussian kernel was used with bandwidth set as the median distance between pairwise samples. For KMM and STM we set $B = 1000$ so that none of s_i reached the upper bound, and $\epsilon = \frac{\sqrt{n_{tr}-1}}{\sqrt{n_{tr}}}$. As reported in [103], when B was reduced to the point where a small percentage of the s_i reached B , empirically performance either did not change, or worsened. For T-SVM we used [52] since the original T-SVM [126] solves an integer programming and thus unscalable to our problem that consists hundreds of thousands of frames. For fairness, we used linear SVMs in all cases. In DA-SVM, we used LibSVM [23] as discussed in Sec. 3.3, $\tau = 0.5$ and $\beta = 0.03$. For SA, we obtained the dimension of subspaces d_{max} using their theoretical bound with $\gamma = 10^6$ and $\delta = 0.1$; SA with both NN and SVM classifiers were reported. Following [72], we tuned DAM using $C = 1$, $\lambda_L = \lambda_{D_1} = \lambda_{D_2} = 1$; β was set as the median of computed MMD value [16]; the threshold for virtual labels were cross-validated in $\{0.01, 0.1, 0.5, 1\}$. Linear SVMs were used as base classifiers. Note that, because these alternative methods are not optimized for our task, their performance might be improved by searching over a wider range of parameters.

Discussion: Tables 3.4 and 3.5 show results on AUC and F1 scores. A linear SVM served as a generic classifier. For semi-supervised learning, T-SVM performed similarly to SVM in RU-FACS, but worse than SVM in CK+. An explanation is because in CK+ the negative (neutral) and positive (peak frames) samples are easier to separate than consecutive frames in RU-FACS. For transductive transfer learning, KMM performed worse than the generic classifier, because KMM estimates sample weights without label information. On the other hand, SA combined with both Nearest Neighbor (NN) and LibSVM led to unsatisfactory performance compared to above meth-

Table 3.5: Cross-subject AU detection on CK+ dataset

AU	AUC					F1 Score				
	SVM	KMM	T-SVM	DA-SVM	STM	SVM	KMM	T-SVM	DA-SVM	STM
1	79.8	68.9	69.9	72.6	88.9	61.1	44.9	56.8	57.7	62.2
2	90.8	73.5	69.3	71.0	87.5	73.5	50.8	59.8	64.3	76.2
4	74.8	62.2	63.4	69.9	81.1	62.7	52.3	51.9	57.7	69.1
6	89.7	87.7	60.5	94.7	94.0	75.5	70.1	47.8	68.2	79.6
7	82.1	68.2	55.7	61.4	91.6	59.6	47.0	43.8	53.1	79.1
12	88.1	89.5	76.0	95.5	92.8	76.7	74.5	59.6	59.0	77.2
15	93.5	66.8	49.9	94.1	98.2	75.3	44.4	40.4	76.9	84.8
17	90.3	66.6	73.1	94.7	96.0	76.0	53.2	61.7	81.4	84.3
Av.	86.1	72.9	64.7	81.7	91.3	70.0	54.7	52.7	64.8	76.6

ods. This is because SA obtained an optimal transformation through linear subspace representation, which could be improper due to the non-linearity of our data. In addition, SA weighted all training samples equally, and thus suffered from biases caused by individual differences (as illustrated in Fig. 3.2). Although SA+SVM performed better in AUC, its low F1 score tells a likely overfitting (low precision or recall). The proposed STM outperformed alternative approaches in general. For AUC in RU-FACS, STM had the highest averaged score about 6% higher over the 2nd highest, and the highest scores in all but 2 AUs. For F1, STM had the highest averaged score about 12 points higher than the nearest alternative, and the highest F1 score of all but AU4. For CK+, STM achieved 91% AUC on average, slightly better than the best-published result 90.5% [151], although the results may not be directly comparable due to different choices of features and registration. It is also noteworthy that we tested the last one-third of a video that could contain low intensities, while [151] tested only on peak frames with the highest intensity. On the other hand, STM may be benefited from additional frames due to more information.

Unlike STM that uses a penalized SVM, T-SVM and SA considered neither re-weighting for training instances nor weight refinement for irrelevant samples, such as noises or outliers. On the other hand, DA-SVM extends T-SVM by progressively labeling test patterns and removing labeled training patterns. Not surprisingly, DA-SVM showed better performance than KMM and T-SVM, because it selected relevant samples for training and thus obtained a better classifier. However, similar to T-SVM, DA-SVM did not update the re-weightings using label information. Moreover, it is not always guaranteed to converge to a correct solution. In our experiments, we faced the situation where DA-SVM failed to converge due to a large amount of samples lying within the margin bounds. In contrast, STM is a biconvex formulation, and therefore guaranteed to converge to a critical point and outperform existing approaches (details in Sec. 3.3).

As for multi-source domain adaptation, DAM overall performed comparably in AUC, but significantly worse than STM in F1. There are at least three explanations. First, AUs are by nature imbalanced: Simply predicting all samples as negative could yield high AUC for infrequent AUs (such as AUs 4), yet zero precision and recall for F1 score. Second, similar to person-specific classifiers, training samples for each subject are typically insufficient to estimate the true distribution (as discussed in Sec. 3.7.3). Using such limited training samples for each subject, therefore, limits the power of base classifiers and the final prediction in DAM. Finally, DAM uses MMD to estimate

Table 3.6: Cross-dataset AU detection: (a) RU-FACS→GEMEP-FERA, and (b) GFT→RU-FACS (“A→B” represents for training on dataset A and test on B).

(a)		AUC					F1 Score				
AU		SVM	KMM	T-SVM	DA-SVM	STM	SVM	KMM	T-SVM	DA-SVM	STM
1		44.7	48.8	43.7	56.9	63.2	46.3	46.4	41.8	46.1	50.4
2		52.8	70.5	52.1	52.3	74.0	47.4	54.2	38.6	45.4	54.6
4		52.7	55.4	54.2	52.7	58.6	57.1	57.1	40.2	42.9	57.4
6		73.5	55.2	77.1	79.9	83.4	60.7	55.2	52.8	56.3	72.7
12		56.8	60.1	70.9	76.1	78.1	67.7	67.7	63.5	62.6	71.5
15		55.1	52.1	59.3	60.2	58.6	31.5	32.8	29.7	26.4	41.1
17		44.3	41.1	39.1	46.2	52.7	27.3	27.1	24.3	24.6	31.4
Av.		54.3	54.8	56.6	60.6	66.9	48.3	48.6	41.6	43.5	54.2

(b)		AUC					F1 Score				
AU		SVM	KMM	T-SVM	DA-SVM	STM	SVM	KMM	T-SVM	DA-SVM	STM
1		45.8	63.6	70.3	71.2	73.7	23.7	29.8	26.6	31.8	38.6
2		46.4	62.8	68.5	68.2	71.7	21.3	25.4	19.4	32.1	30.2
4		56.9	60.1	59.1	47.2	61.7	18.3	24.5	20.7	19.4	28.5
6		65.5	73.9	81.5	74.1	93.3	42.2	46.8	30.4	38.7	61.4
12		65.3	72.1	76.3	80.9	90.3	43.2	47.6	45.8	56.8	62.2
14		57.2	54.8	53.7	70.2	72.2	25.8	23.8	25.9	29.7	36.2
15		56.9	61.8	64.2	65.5	80.4	23.7	30.3	28.2	29.9	37.8
17		52.4	54.5	64.8	72.6	72.6	30.8	31.5	32.3	38.9	39.5
Av.		55.8	62.9	67.3	68.7	77.0	28.6	32.5	28.7	34.7	41.8

inter-subject distance, which could be inaccurate due to insufficient samples or sampling bias (*e.g.*, some subjects have more expressions than others).

Although in Table 3.4 STM achieved slightly worse in AUC, STM showed a better improvement in F1 metric, which better suits our imbalanced detection task. A major reason that limits STM’s improvement is because GEMEP-FERA comprises limited subjects and training samples, and thus hinders STM from selecting and receiving proper supports from the training samples. This can be also explained by the findings of selection ability in Sec. 3.7.3. When the number of subjects and training samples increase, as illustrated by the CK+ and the RU-FACS datasets in Tables 3.5 and 3.4, STM is able to gain contributions from the selected data, and thus the improvement becomes clearer. Overall STM achieves the most competitive performance due to the properties of instance re-weighting, weight refinement, and convergence.

Cross-dataset AU detection

Detecting AUs across datasets is challenging because of differences in acquisition and participant characteristics and behavior. As shown in Fig. 3.8, participant characteristics, context, background, illumination, camera parameters, compression schemes are among the differences that may bias features. Generic SVMs fail to address such differences. Secs. 3.7.3 and 3.7.3 have shown the effectiveness of STM on *within-dataset* experiments involving within-subject and across-subject scenarios. This section aims to justify that STM can attain not only subject adaptation but can

Table 3.7: Expression detection with AUC on (a) CK+ and (b) GEMEP-FERA

Expression	SVM	KMM	T-SVM	DA-SVM	STM
Anger	95.1	85.3	76.1	–	96.4
Contempt	96.9	94.5	88.8	–	96.9
Disgust	94.5	81.6	84.2	–	96.0
(a) Fear	96.6	92.7	84.9	–	95.5
Happy	99.4	93.9	86.7	–	98.9
Sadness	94.5	76.0	78.7	–	93.3
Surprise	97.3	64.5	81.8	–	97.6
Av.	96.3	84.1	83.0	–	96.4

Expression	SVM	KMM	T-SVM	DA-SVM	STM
Anger	31.1	66.5	70.4	78.8	78.6
Fear	31.9	81.4	64.5	83.9	85.5
Joy	90.2	33.5	78.9	71.1	95.0
Relief	20.4	74.8	76.8	87.9	88.4
Sadness	73.4	80.2	77.1	74.7	84.8
Av.	49.4	67.3	73.5	79.3	86.5

be naturally extended for *cross-dataset* adaptation. Specifically, we performed two experiments, RU-FACS→GEMEP-FERA and GFT→RU-FACS, using the same settings described above.

Table 3.6 shows the results. One can observe that cross-domain approaches outperformed a generic SVM in most cases. It is not surprising because a generic SVM does not model the biases between datasets. That is, in the cross-dataset scenario, the training and test distributions are more likely different than in within-dataset scenario, causing an SVM to fail to transfer the knowledge from one dataset to another. Among the cross-domain methods, STM consistently outperforms the others. Observe STM gained improvement over SVM in Table 3.4 by 12.8% in AUC (76.5→86.3) and 33.7% in F1 (40.6→54.3), and in Table 3.6(b) by 37.9% in AUC (55.8→77.0) and 46.1% in F1 (28.6→41.8). The advantages of STM over SVM becomes clearer in the cross-dataset experiments.

3.7.4 Holistic Expression Detection

Taking into account of individual differences, STM showed improvement for AU detection. In this experiment, we ask whether the same could be found for holistic expression detection. We used the major benchmarks CK+ [161] and FERA emotion subchallenge [248] for this experiment, and the same settings in Sec. 3.7.2, except for that the labels were replaced as holistic expressions. Similar to [248], we utilized every frame of a video to train and test our algorithm. Because each video has only a single expression label instead of a frame-by-frame labeling, F1 score is meaningless in this experiment. For CK+, 327 out of the original 593 videos were given a nominal expression label based on the 7 basic and discrete expressions: *Anger*, *Contempt*, *Disgust*, *Fear*, *Happy*, *Sadness*, and *Surprise*. For GEMEP-FERA, 289 portrayals were retained one out of the five expression states: *Anger*, *Fear*, *Joy*, *Sadness*, and *Relief*. The training set included 7 actors with 3~5 instances of each expression per actor. We evaluated on the training set, which contains a total of 155 videos. STM was also compared to alternative approaches discussed in Sec. 3.7.3.

Table 3.7(a) shows the results from CK+. Note that DA-SVM is unavailable in this experiment because it failed to converge to a final classifier due to insufficient test data, recalling that we used the last one-third frames of each video for test. One can observe that a generic SVM performed fairly well because positive (peak expressions) and negative samples (neutral faces) are relatively easy to separate in CK+. KMM and T-SVM resulted in suboptimal results due to the lack of a weight-refinement step, and thus were unable to rectify badly estimated weights for learning the final classifier (see discussions in Sec. 3.6). This effect becomes obvious when there is insufficient test data, such as this experiment. On the other hand, STM considers the labels for weight

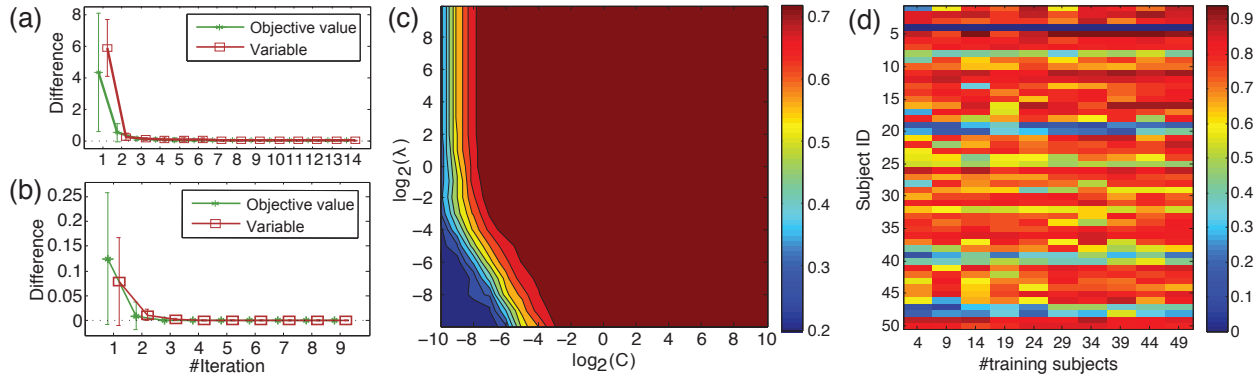


Figure 3.10: Analysis experiments: (a)–(b) Objective and variable differences between iterations with initialization \mathbf{w}_0 (STM_w) and \mathbf{s}_0 (STM_s), respectively. (c) Performance versus parameter choices. (d) Per-subject F1 score v.s. # training subjects.

refinement and performed similarly as well as a generic SVM.

Table 3.7(b) presents our results on GEMEP-FERA, which served as a larger and more challenging benchmark for evaluating the holistic expression detection performance. In this experiment, each test video consists of tens of frames, and thus enables DA-SVM to converge in most cases. The generic SVM performed poorly due to large variations in this dataset, such as head movements and spontaneous expressions. Without the ability to select meaningful training samples, the generic classifier suffered from the individual differences. Other cross-domain methods alleviated the person-specific biases and produced better results. Overall STM achieved the most satisfactory performance. This serves as evidence that when training data grow larger and more complex, the improvement of STM becomes clearer.

3.7.5 Analysis

Initialization order

A potential concern of STM is that the initialization order could affect the convergence property and performance. To evaluate this, we examined the initialization order with \mathbf{w}_0 (STM_w) and with \mathbf{s}_0 (STM_s). Standard two-stage approach, *i.e.*, solving the selection coefficients first and then the penalized SVM (*e.g.*, [103]), can be interpreted as STM_w, as discussed in Sec. 3.3. To validate convergence property of STM, we randomized 10 initialization sets for STM_w and STM_s respectively. Upon the convergence of STM, we computed their objective differences in consecutive iterations ($g(\mathbf{z}_{t+1}) - g(\mathbf{z}_t)$), and the absolute sum of variable difference ($\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_1$). For the cases where STM took fewer iterations to converge, we set the difference of later iterations to 0.

Fig. 3.10(a) shows the curve of mean and standard deviation of differences across the iterations of STM_w and STM_s. Note that the differences were scaled for visualization convenience. The random initial value was reflected in the first iteration and made a major difference with the value of the second iteration. One can observe that in STM_w and STM_s, both the objective value and difference between consecutive variables decreased at each step and toward convergence, as theoretically detailed in Sec. 3.4. Note that, although the resulting solution was slightly different due to different initialization, the performance remains the same as both converge to a critical point. We observed so by comparing the confusion matrices during the experiments.

Parameter choice

Recall that training STM involves two parameters: C for the tradeoff between maximal margin and training loss, and λ for the tradeoff between the SVM empirical risk and the domain mismatch. This section examines the sensitivity of performance with respect to different parameter choices. Specifically, we ran the experiment of detecting AU12 on the CK+ dataset with the parameters ranges $C \in \{2^{-10}, \dots, 2^{10}\}$ and $\lambda \in \{2^{-10}, \dots, 2^{10}\}$. Following the experiment settings in Sec. 3.7.2, we used the leave-one-subject-out protocol and computed an averaged F1 score for evaluating the performance. We used Gaussian kernel with a fixed bandwidth as the median distance between sample points.

Fig. 3.10(c) illustrates the contour plot of F1 score v.s. different parameter pairs in terms of $(\log_2(C), \log_2(\lambda))$. As can be observed, the performance scatters evenly in most region of the plot, showing that STM is robust to the parameter choices when their values are reasonable. The performance decayed when both (C, λ) become extremely small ($< 2^{-6}$), as shown in the bottom left of the plot. This is not surprising because smaller values of C and λ imply less emphasis on training loss and personalization. Note that with large enough λ , STM does not need large C to achieve comparable F1, providing an explanation that personalization helps avoid imposing large C and hence avoid overfitting. As a general guideline for choosing parameters, we suggest a small value of C with a reasonable λ (thus encouraging a large-margin decision boundary with reasonable distribution mismatch).

We note that cross validation (CV) for domain adaptation methods is difficult and remains an open research issue. As also mentioned in [228], this issue becomes vital in a conventional scenario where the number of training samples is much smaller than the number of test samples. However, in our case, we always have much more training samples than test samples, and thus, the CV process is less biased under covariate shift. In addition, as can be seen in Fig. 2 of [228], with proper σ (kernel bandwidth) and standard CV, KMM consistently reaches lower error than the KL-divergence-based CV [228]. This serves as a justification for KMM’s ability to estimate importance weights.

Domain size

The intuition for STM to work better in facial expression analysis is a judicious selection of training samples. The availability of richer diversity grants STM a broader knowledge to select better candidates that match the test distribution. This experiment examines performance changes w.r.t. diversities of the source domain, for which we evaluated by the domain size or the number of training subjects. Intuitively, the larger number of training subjects, the more diverse the training domain is, and thus the more likely STM could perform better. We compared STM to a generic SVM (with cross-validation) to contrast the performance.

This experiment was performed on AU 12 using the RU-FACS dataset. A subset from 3 to 27 training subjects was randomly picked as a shrunk domain. The leave-one-subject-out protocol and F1 score were used following Sec. 3.7.2. Fig. 3.11(a) illustrates the effects of #training subjects on averaged F1 scores. For each domain size, the mean and standard deviation were computed on F1 scores over all test subjects. Test subjects without true positives were ignored because their precision and F1 scores were not computable. One can observe that, as #training subjects grew, STM achieved higher F1 scores, and also performed more consistently with lower standard deviation. This observation imitates Sec. 3.7.3, where a source domain with poor diversity was shown to

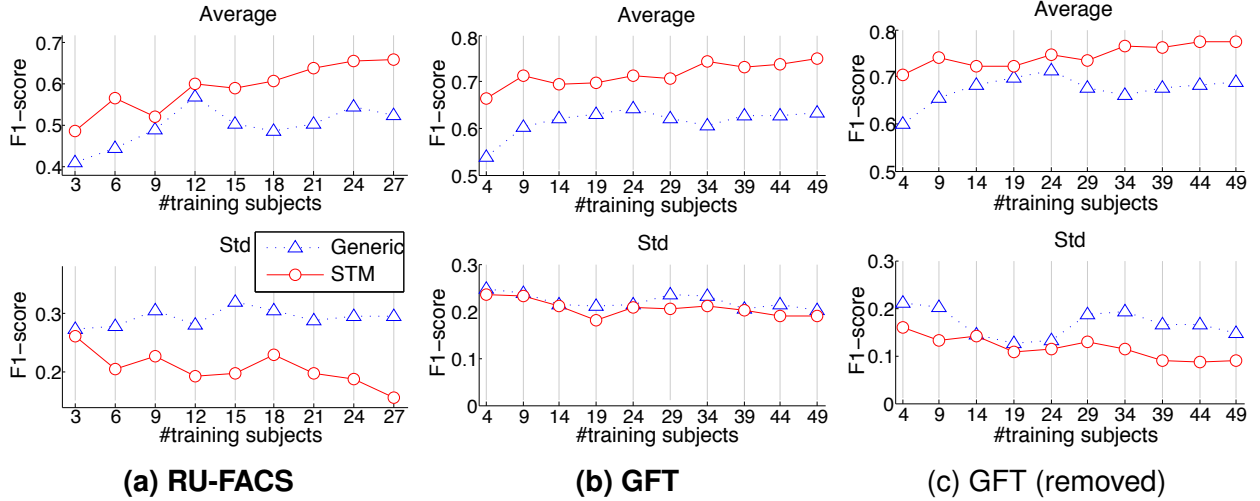


Figure 3.11: Performance versus domain size: The averaged and standard deviation of F1 score on (a) RU-FACS. (b) and (c) show the F1 scores on the GFT dataset before and after removing the *outlier* subjects, respectively. (more descriptions in text)

limit STM’s performance. On the other hand, generic classifier improved when #training subjects arose to 12. However, with more training subjects being introduced, its performance was slightly lowered due to the biases caused by individual differences. Note that, because the training subjects were downsampled in a randomized manner, it is possible that STM achieved better performance on a domain with less training subjects.

As another justification, we examined the effects of domain size on the GFT dataset [209], which contains a larger number of subjects and more intensive facial expressions than RU-FACS. The GFT dataset records videos of real-life social interactions among three-person groups in less constrained contexts. Videos were recorded using separate wall-mounted cameras facing each subject; Fig. 3.10(e) shows exemplar frames. The videos include moderate-to-large head rotations and frequent occlusions; facial movements are spontaneous and unscripted. We selected 50 videos with around 3 minutes each (5400 frames).

Following the same procedure, we randomly picked a subset of subjects varying from 4 to 49 as the shrunk domains. Fig. 3.11(b) shows the F1 scores with respect to the number of training subjects. One can observe the averaged F1 score increases with #training subjects, although the standard deviation fluctuates. To study the fluctuation, we broke down the averaged F1 into individual subjects corresponding to different training sizes, as shown in Fig. 3.10(d). Each row represents a test video; each column represents one number of training subjects (ranging from 4 to 49). Note that for subject 4 (the 4th row), there is no F1 score because AU 12 was absent. One can observe that for 6 *outlier* subjects (e.g., rows 19, 20, 39, 40, 47, 48), their F1 scores remained low even as the number of subjects was increased. This result suggests that these subjects share no or few instances in the feature space. Visual inspection of their data was consistent with this hypothesis. The outliers were ones with darker skin color, asymmetric smiles or relatively large head pose variations. Thus, for these subjects STM could offer no benefit. This finding suggests the need to include greater heterogeneity in training subjects. When these subjects were omitted, as shown in Fig. 3.11(c), the F1 scores are markedly higher. The influence of the domain size becomes clear

and replicates Fig. 3.11(a). It is interesting to note that, for generic classifiers, the performance increased until 24 training subjects and then drops abruptly. This observation serves as another evidence that individual differences (introduced by increasing number of training subjects) could bias generic classifiers.

Between these two experiments, generally the averaged F1 score in GFT is higher than in RU-FACS. At least two factors may have accounted for this difference. One is that participants in GFT may have been less inhibited and more expressive. In RU-FACS, subjects were motivated to convince an examiner of their veridicality. They knew that they would be penalized if they were not believed. In the three-person social interaction of GFT, there were no such negative contingencies. Subjects may have felt more relaxed and become more expressive. More intense AUs are more easily detected. The other factor is that inter-observer reliability of the ground truth FACS labels was likely much higher for GFT than for RU-FACS. Kappa coefficients for GFT were exceptionally good. While reliability for RU-FACS is not available, we know from past confirmation-coding that inter-observer agreement was not as high. Less error in the GFT ground truth would contribute to more accurate classifier performance.

3.7.6 Discussion


In above experiments, we have evaluated STM against alternative methods in many scenarios: Within-subject (Sec. 3.7.3), across-subject (Sec. 3.7.3), across-dataset (Sec. 3.7.3), and holistic expression detection (Sec. 3.7.4). We also analyzed STM on its initialization order, and sensitivity to parameters and domain size (Sec. 3.7.5). STM consistently outperformed a generic SVM and most transfer learning methods. The advantage of STM is clearest in GFT, where the variety of subjects are more extensive, and slightly so, in RU-FACS. The results indicate a more obvious improvement in F1 than in AUC, in large complex datasets than in posed datasets, in cross-dataset scenario than in within-dataset scenario, and with more training subjects than with fewer ones.

STM has some limitations. For example, it suffers from the lack of training subjects or crucial mismatch between training and test distributions, which are known as common drawbacks in unsupervised domain adaptation methods. For a theoretical analysis in terms of performance v.s. the number of samples, Corollary 1.9 in KMM [95] reaches a transductive bound for an estimated risk of a re-weighted task, given the assumptions of linear loss and data being iid. However, it remains unclear how to theoretically analyze STM's performance in terms the number of test samples, because STM involves nonlinear loss functions and the data are from real-world videos (non-iid).

3.8 Summary

Based on the observation on individuals differences, we have presented Selective Transfer Machine (STM) for personalized facial expression analysis. We showed that STM translates to a biconvex problem, and proposed an alternating algorithm with a primal solution. In addition, we introduced L-STM, an extension of STM that exhibited significant improvement when labeled test data are available. Our results on both AU and holistic expression detection suggested that STM is capable of improving test performance by selecting training samples that form a close distribution to test samples. Experiments using within-subject, cross-subject, and cross-dataset scenarios revealed two insights: (1) Some training data are more instrumental than others, and (2) the effectiveness of STM scales as the number of training subjects increases.

It is worth noting that STM can be extended to other classifiers with convex decision functions and losses, such as multi-class SVM (*e.g.*, [279]), regression (*e.g.*, [271]), or segment-based classifiers (*e.g.*, [222]). This is a direct outcome of Property 1 in Sec. 3.4.1. However, for non-convex cases, such as random forest, local minimum could cause worse performance. We leave extensions to non-convex classifiers as a focus of future work. Moreover, improving STM’s training speed could be another direction due to the QP for solving s . There have been a number of follow-up studies that address this scalability issue by learning a combination from pre-trained classifiers, *e.g.*, [205]. Finally, while this study focuses evaluations on facial expressions, STM could be applied to other fields where object-specific issues are involved, *e.g.*, object or activity recognition [279]. We also note that the results in this chapter were reported with hand-crafted features and relatively small datasets, which were the best possible data collection upon the beginning of this study. In the next chapter, we will introduce an end-to-end framework for jointly learning feature and classifier, and report results on much larger datasets.



An End-to-End Supervised Framework for Facial Action Unit Detection

“If you torture the data long enough, it will confess.”

Ronald Coase

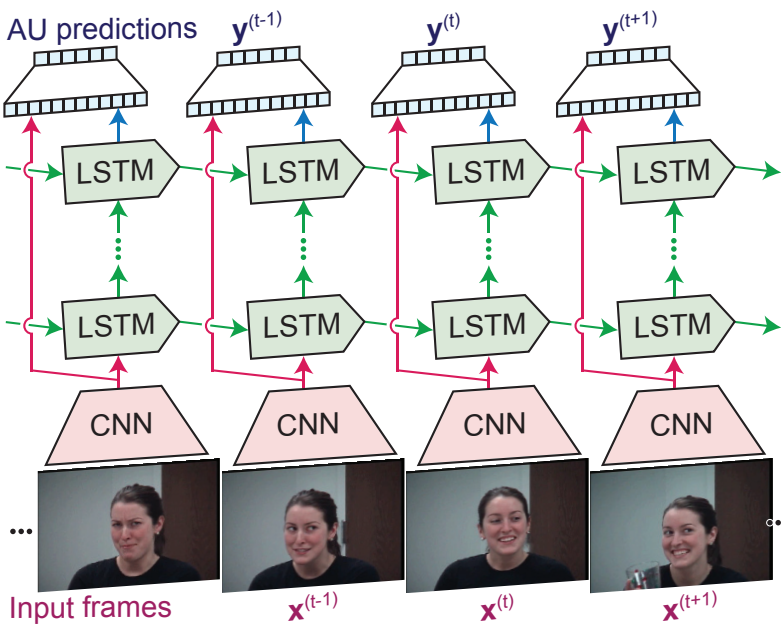


Figure 4.1: An overview of the proposed hybrid deep learning framework. The proposed network first possesses strengths of CNNs and LSTMs to model and utilize both spatial and temporal cues. Then, we employ a fusion network to combine both cues to produce frame-based prediction for multiple AUs.

In the previous chapter, we addressed distribution shifts between training and test subjects with Selective Transfer Machine (STM). The major breakthrough of STM is its use of transductive training strategy—guide the training procedure with unlabeled test samples that are freely available during prediction time. We showed that such “personalized” classifiers adapted with unlabeled or labeled test samples yield better performance for both tasks of facial expression recognition and AU detection in terms of within-subject, between-subject and between-dataset scenarios.

Although STM has shown superior results over the baseline generic classifiers and alternative transfer learning approaches, it endures several limitations. First, STM considers only hand-crafted features, which encode all information about the face, including both identity and facial expression. Such features, thus, are not optimized for classification purpose. Inspired by this observation, in this chapter, we conjecture that issues caused by individual differences can be further reduced as early as in the stage of learning a good feature representation. Second, STM treats each facial action independently, and neglects the relationships between AUs. For instance, due to the activation of the same group of facial muscles, AU 1 (inner-brow raise) increases the likelihood of AU 2 (outer-brow raise), yet decreases that of AU 6 (cheek raiser). Similarly, AUs 6 and 12 are likely to co-occur in expressions of enjoyment, embarrassment, and pain but not in expressions of distress or sadness. Knowing and utilizing such AU correlations could assist in predicting AUs given one another. Third, prediction carried out by STM is frame-based, and thus it misses the opportunity to take advantage of temporal transition to preserve better prediction consistency. Fourth, the inherent transductive nature of STM requires to carry all training samples and retrain for a new test domain (either a new dataset or a new subject). When the dataset is large, such nature can be impractical. Finally, distributions of spontaneous facial actions are dramatically imbalanced, yet neglected for most studies including STM. Table 4.1 summarizes these limitations.

Unlike most studies that tackle these limitation separately, this chapter proposes a hybrid network architecture to jointly model them. Specifically, spatial representations are extracted by a Convolutional Neural Network (CNN), which, as analyzed in this paper, is able to reduce person-specific biases caused by hand-crafted features (*e.g.*, SIFT and Gabor). To model temporal dependencies, Long Short-Term Memory (LSTMs) are stacked on top of these representations, regardless of the lengths of input videos. The outputs of CNNs and LSTMs are further aggregated into a fusion network to produce per-frame prediction of 12 AUs. Our network naturally addresses the three issues together, and yields superior performance compared to existing methods that consider these issues independently. Extensive experiments were conducted on two large spontaneous datasets, GFT and BP4D, with more than 400,000 frames coded with 12 AUs. Furthermore, we introduce two multi-label sampling strategies to address the inherent imbalance problem in AU classes. On both datasets, we report improvements over a standard multi-label CNN and feature-based state-of-

Table 4.1: Limitations in standard AFA methods and related solutions presented in this chapter

Limitations	Solutions
Hand-crafted features	Representation learning (Sec. 4.2.1)
Lack consideration of AU relations	Multi-label architecture (Sec. 4.2.1)
Lack of temporal consistency	Long short-term memory (Sec. 4.2.2)
Imbalanced AU classes	Multi-label sampling strategies (Sec. 4.4)

the-art. Finally, we provide visualization of the learned AU models, which, to our best knowledge, reveal how machines see AUs for the first time.

4.1 More Aspects for Facial Action Unit (AU) Detection

Facial actions convey information about a person’s emotion, intention, and physical state, and are vital for use in studying human cognition and related processes. To encode such facial actions, the Facial Action Coding System (FACS) [77] is the most comprehensive. FACS segments visual effects of facial activities into action units (AUs), providing an essential tool in affective computing, social signal processing and behavioral science. Such AUs have shown a powerful description in universal expressions and led discoveries to many areas such as marketing, mental health, and entertainment.

A conventional pipeline of automated facial AU detection compiles four major stages: face detection \mapsto alignment \mapsto representation \mapsto classification. With the progress made in face detection and alignment, most research nowadays focuses on features, classifiers, or their combinations. However, due to slow-growing rate in the amount of FACS-coded data, it remains unclear how to pick the best combination that generalizes across subjects and datasets. At least three aspects affect the performance of automated AU detection: (1) *Spatial representation*: Engineered features, *e.g.*, SIFT, induce person-specific biases in estimating AUs, and hence encourage sophisticated learning methods such as personalized classifiers [38, 205, 271]. A good representation must generalize to unseen subjects, regardless of the existence of individual differences caused by appearance, behaviors or facial morphology. (2) *Temporal modeling*: Temporal info is crucial for distinguishing AUs, due to the ambiguity and dynamic nature of facial actions. However, it remains unclear how temporary context can be effectively encoded and recalled. (3) *AU correlation*: The presence of AUs influences each other. For instance, the occurrence of AU12 suggests a co-occurrence of AU6, and reduces the likelihood of AU15. Such correlation helps a detector determine one AU given others. Despite the seemingly unrelated nature of the three aspects, this paper shows that it is possible and better consider them jointly. One observation is that a good representation would help learn temporal models and AU correlations, and knowing AU correlations could benefit representation learning and temporal modeling. Most existing studies, however, address these aspects separately, and thus are unable to fully capture their entangled nature.

To address the above issues, this paper proposes a hybrid network architecture that models both spatial and temporal relationships from multiple AUs. The proposed network is appealing for naturally modeling the three complementary aspects. Fig. 4.1 gives an overview of the proposed framework. To learn a generalizable representation, a CNN is trained to extract spatial features. As analyzed in this study, such features reduce the ubiquitous person-specific biases in hand-crafted features [38, 205, 271], and thus offer possibilities to reduce the burden of designing sophisticated classifiers. To capture temporal dependencies, LSTMs are stacked on top of the spatial features. We aggregate the output scores from both CNNs and LSTMs into a fusion network to predict 12 AUs for each frame. Extensive experiments were performed on two spontaneous AU datasets, GFT and BP4D, containing totally >400,000 frames. We report that the learned spatial features, further combined with temporal information, outperform a standard CNN and feature-based state-of-the-art methods. In addition, we visualize notions of each AU learned by the model, which, to our best knowledge, reveal how machines see facial AUs for the first time.

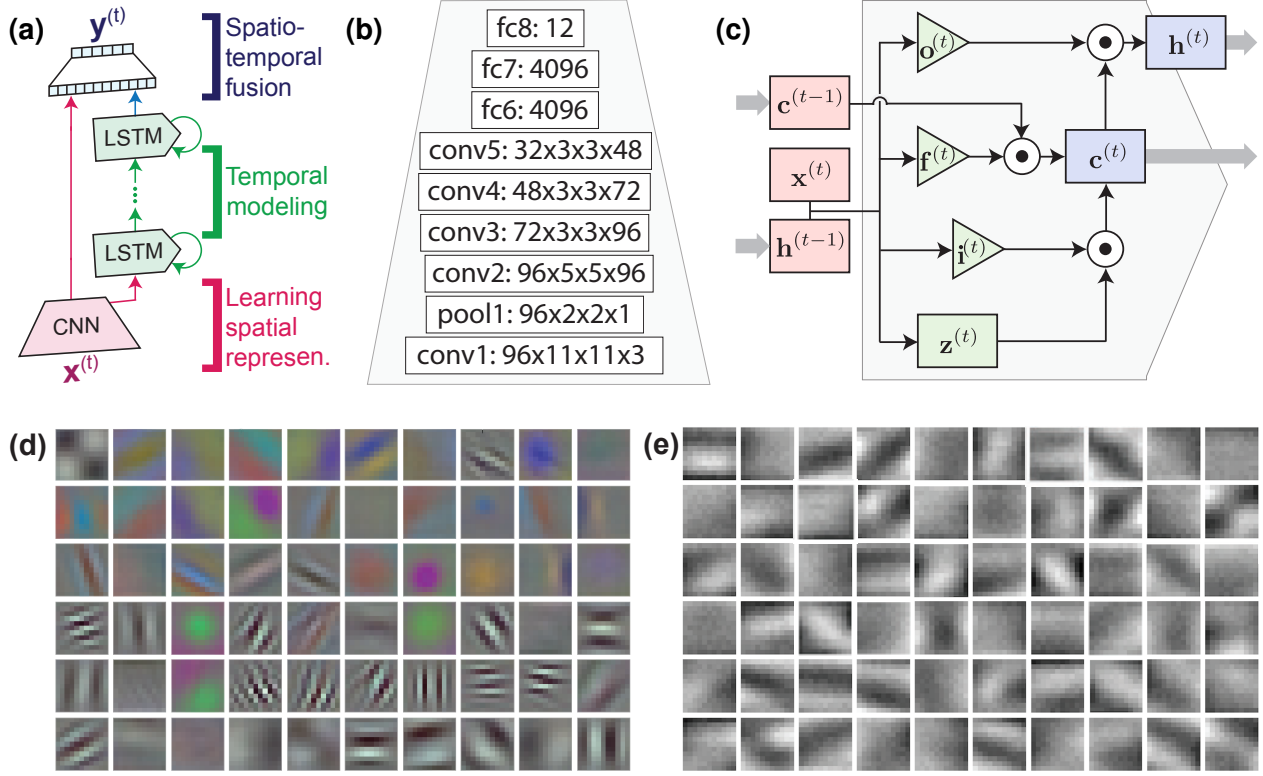


Figure 4.2: The structure of the proposed hybrid network: (a) Folded illustration of Fig. 4.1, showing 3 components of learning spatially representation, temporal modeling, and spatiotemporal fusion, (b) 8-layer CNN architecture for multi-label prediction, and (c) the schematic of an LSTM block. (d)-(e) conv1 kernel visualization on ImageNet [137] and GFT datasets, respectively. As can be seen, filters learned on faces contain less color blob detectors, suggesting color information is less useful for AU detection.

4.2 The Hybrid Network for Multi-label Facial AU Detection

This section describes the proposed hybrid network to jointly address multiple limitations encountered in the previous chapter. Fig. 4.2(a) shows a folded illustration of the network. Below we describe each component in turn.

4.2.1 Learning spatial representation

The literature has shown evidence that hand-crafted features impair generalization of AU detectors across subjects [38, 205, 271]. We argue that specialized representation could be learned to reduce the burden of designing a sophisticated classifier, and further improve detection performance. In addition, AUs are correlated: Some AUs co-occur frequently (*e.g.*, AUs 6+12 in a Duchenne smile), and some infrequently. Such relation is likely to lead to more reliable classifiers [79, 262, 285]. To this end, we train a multi-label convolutional neural network (CNN) to jointly a classification model with AU dependencies. Here we modified AlexNet [137] as shown in Fig. 4.2(b). Given a ground truth label $\mathbf{y} \in \{-1, 0, 1\}^L$ (-1/1 indicates absence/presence, and 0 missing label) and a prediction $\hat{\mathbf{y}} \in \mathbb{R}^L$ for L AU labels, this multi-label CNN aims to minimize the multi-label cross

entropy loss:

$$L_E(\mathbf{y}, \hat{\mathbf{y}}) = \frac{-1}{L} \sum_{\ell=1}^L [y_\ell > 0] \log \hat{y}_\ell + [y_\ell < 0] \log(1 - \hat{y}_\ell),$$

where $[x]$ is an indicator function returning 1 if x is true, and 0 otherwise. The outcome of the fc7 layer is L_2 normalized as the final representation, resulting in a 4096-D vector. We denote this representation “fc7” hereafter. Due to dropout and ReLu, fc7 feature contains $\sim 35\%$ zeros out of 4096 values, resulting in a significantly sparse vector. The proposed multi-label CNN is similar to [92] and AlexNet [137], with slightly different architecture and purpose. [92] takes a 40×40 image as input, which, in our experience, can be insufficient for recognizing subtle AUs on the face. AlexNet was designed for object classification, yet, for structured face images, the original design are less useful than for natural images such as objects and scenes. Instead, we train the entire network from scratch. Fig. 4.2(d) visualizes the learned kernels from the conv1 layer on the BP4D and the GFT datasets. As can be seen, the learned kernels contain less color blob detectors than the ones learned on ImageNet [137]. In Sec. 5.6, we will empirically show that fc7 is able to reduce identity factors compared to hand-crafted features such as SIFT or Gabor.

4.2.2 Temporal modeling with stacked LSTMs

It is usually hard to tell an “action” by looking at only a single frame. Having fc7 extracted, we use stacked LSTMs [102] for learning such temporal context. Fig. 4.2(c) shows the schematic of an LSTM block. We experimented various numbers of layers and memory cells, and chose 3 stacks of LSTMs with 256 memory cells each. One benefit of LSTM is its ability of encoding crucial information during the transition between two frames. Unlike learning spatial representation on fixed and cropped images, videos can be difficult to be modeled with a fixed-size architecture, *e.g.*, [25, 131]. LSTM serves as an ideal model for avoiding the well-known “vanishing gradient” effect in recurrent models, and makes it possible to model long-term dependencies.

Recurrent LSTMs: Denote a sequence of input frames as $(\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(T)})$, and their labels as $(\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(T)})$, where superscripts indicate time steps. A recurrent model is expressed by iterating the equations from $t = 1$ to T :

$$\mathbf{h}^{(t)} = \mathcal{H}(\mathbf{W}_{xh}\mathbf{x}^{(t)} + \mathbf{W}_{hh}\mathbf{h}^{(t-1)} + \mathbf{b}_h), \quad (4.1)$$

$$\mathbf{y}^{(t)} = (\mathbf{W}_{hy}\mathbf{h}^{(t)} + \mathbf{b}_y), \quad (4.2)$$

where \mathbf{W} denotes weight matrices, denotes bias vectors, \mathcal{H} is the hidden layer activation function (typically the logistic sigmoid function), and the subscripts $\{x, h, y\}$ denote the (input, hidden, output) layers respectively. LSTM replaces the hidden nodes in the recurrent model with a memory cell, which allows the recurrent network to remember long term context dependencies. Given an input vector $\mathbf{x}^{(t)}$ at each time t and the hidden state from previous time $\mathbf{h}^{(t-1)}$, we denote a linear mapping as:

$$\phi_\star^{(t)} = \mathbf{W}_\star \mathbf{x}^{(t)} + \mathbf{R}_\star \mathbf{h}^{(t-1)} + \mathbf{b}_\star, \quad (4.3)$$

where \mathbf{W} is the rectangular input weight matrices, \mathbf{R} is the square recurrent weight matrices, and \star denotes one of LSTM components $\{c, f, i, o\}$, *i.e.*, cell unit, forget gate, input gate, and output

gate. Element-wise activation functions are applied to introduce nonlinearity. Gate units often use a *logistic sigmoid* activation $\sigma(a) = \frac{1}{1+e^{-a}}$; cell units are transformed with *hyperbolic tangent* $\tanh(\cdot)$. Denote the point-wise multiplication of two vectors as \odot , LSTM applies the following update operations:

$$\begin{aligned} \text{Block input: } \mathbf{z}^{(t)} &= \tanh(\phi_c^{(t)}) \\ \text{Forget gate: } \mathbf{f}^{(t)} &= \sigma(\phi_f^{(t)}) \\ \text{Input gate: } \mathbf{i}^{(t)} &= \sigma(\phi_i^{(t)}) \\ \text{Output gate: } \mathbf{o}^{(t)} &= \sigma(\phi_o^{(t)}) \\ \text{Cell state: } \mathbf{c}^{(t)} &= \mathbf{i}^{(t)} \odot \mathbf{z}^{(t)} + \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} \\ \text{Block output: } \mathbf{h}^{(t)} &= \mathbf{o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}) \end{aligned}$$

As seen in the update of cell states, an LSTM cell involves *summation* over previous cell states. The gradients are distributed over sums, and propagated over a longer time before vanishing. Because AU detection is by nature a *multi-label classification* problem, we optimize LSTMs to jointly predict multiple AUs according to the maximal-margin loss:

$$L_M(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n_0} \sum_i \max(0, \lambda - y_i \hat{y}_i), \quad (4.4)$$

where λ is a pre-defined margin, and n_0 indicates the number of non-zero elements in ground truth \mathbf{y} . Although typically $\lambda=1$ (such as in regular SVMs), here we empirically choose $\lambda=0.5$ because the activation function has squeezed the outputs into $[-1, 1]$, making the prediction value never go beyond $\lambda=1$. During back propagation, we pass the gradient $\frac{\partial L}{\partial \hat{y}_i} = -\frac{y_i}{n_0}$ if $y_i \hat{y}_i < 1$, and $\frac{\partial L}{\partial \hat{y}_i} = 0$ otherwise. At each time step, LSTMs output a vector indicating potential AUs.

Practical issues: There has been evidence that a deep LSTM structure preserves better descriptive power than a single-layer LSTM [102]. However, because fc7 features are of high-dimension (4096-D), our design of LSTMs can lead to a large model with >1.3 million parameters. To ensure that the number of parameters and the size of our datasets maintain the same order of magnitude, we applied PCA to reduce the fc7 features to 1024-D (preserving 98% energy). We set dropout rate as 0.5 to the input and hidden layers, resulting in a final model of ~ 0.2 million parameters. More implementation details are in Sec. 5.6.

4.2.3 Frame-based spatiotemporal fusion

The spatial CNN performs AU detection from still video frames, while the temporal LSTM is trained to detect AUs from temporal transitions. Unlike video classification that produces video-based prediction, we model the correlations between spatial and temporal cues by adding an additional fusion network. We modify the late fusion model [131] to achieve this goal. Fig. 4.1(b) gives an illustration. For each frame, two separate fully connected layers with shared parameters are placed on top of both CNNs and LSTMs. The fusion network merges the stacked L_2 -normalized scores in the first fully connected layer. In experiments, we see this fusion approach consistently improves the performance compared to CNN-only results.

4.3 Evaluations

This section performs a number of evaluations on the proposed hybrid network. In particular, we provide evidence in hope to answer the following questions:

- (1) Can better features be learned for analyzing facial actions, and why are they better?
- (2) How can temporal information help predict facial actions, given the diverse, rich and complex actions that a face can exhibit?
- (3) Is it actually helpful to jointly consider all factors in one learning framework?
- (4) What exactly do machines learn about a facial action, or specifically, what do they “see”?

In the following, we will first describe the datasets used for our evaluation, and then the settings for experiments. Evaluations will be carried out for both the learned representation and detection performance. Finally, we will introduce an optimization-based visualization to explicitly show how machines see the AUs.

4.3.1 Datasets

We evaluated the proposed hybrid network on two of the largest spontaneous datasets BP4D [282] and GFT [50]. Each dataset was FACS-coded by certified coders. AUs occurring more than 5% base rate were included for analysis. In total, we selected 12 AUs to perform the experiments, resulting in >400,000 valid frames. Unlike previous studies that suffer from scalability issues and require downsampling of training data, the network is in favor of large dataset so we made use of all available data. Note that the CK+ benchmark [161] is not applicable because the AU annotations are given on single video; we aim at per-frame prediction. We refer interested readers on preliminary results of Baby-FACS detection using our CNN model in Appendix B.

BP4D [282] is a spontaneous facial expression dataset in both 2D and 3D videos. The dataset includes 41 participants associating with 8 interviews. Frame-level ground-truth for facial actions are obtained using the FACS. In our experiments, we used 328 2D videos from 41 participants, resulting in 146,847 available frames with AU coded. We selected positive samples as those with intensities equal or higher than A-level, and negative samples as the remaining.

GFT [50] contains 240 groups of three previously unacquainted young adults. Moderate out-of-plane head motion and occlusion are presented in the videos, making AU detection challenging. We used 50 participants with each containing one video of about 2 minutes (~ 5000 frames), resulting in 254,451 available frames with AU coded. Frames with intensities equal or greater than B-level are used as positive, otherwise, intensities less than B-level are negative.

4.3.2 Settings

Pre-processing: We pre-processed all videos by extracting facial landmarks using IntraFace [60]. Tracked faces were registered to a reference face using similarity transform, resulting in 200×200 face images, which were then randomly cropped into 176×176 and/or flipped for data augmentation. Each frame was labeled +1/-1 if an AU is present/absent, and 0 otherwise (*e.g.*, lost face tracks or occluded face).

Dataset splits: For both datasets, we adopted two protocols. First is a *3-fold protocol*: Each dataset was evenly partitioned into 3 folds with exclusive subjects. We iteratively trained a model using two folds and evaluated on the remaining one, until all subjects were tested. Validation was

assigned to $\sim 20\%$ of the training subjects. To maximize the limit of deep models, we adopted an additional train/validation/test splits as in the deep learning literature (e.g., [137, 224, 265]). Specifically, we used a *10-fold protocol*, where 9 folds were for training/validation and one fold for test. Different from the 3-fold protocol, here only the one out of 10 folds was tested. In addition, to measure the transferability of fc7 features, we performed a *between-dataset* protocol by training CNNs on one dataset and using it to extract spatial representations on another.

Evaluation metrics: To provide an evaluation in an objective manner, we reported performance using three metrics. Denote R and P as recall and precision. Frame-based F1-score ($F1_{\text{frame}} = \frac{2RP}{R+P}$) is used for its popularity in AU detection. It serves one gold standard to compare with results reported in the literature. To compensate the skewed nature of AUs, F1-norm computes a skew-normalized F1-frame by multiplying false negatives and true negatives by the factor of skewness, which is computed as the ratio of positive samples over negative ones. Because AUs occur as temporal signals, we also evaluated an event-based F1 ($F1_{\text{event}} = \frac{2ER \cdot EP}{ER+EP}$) to measure detection performance at segment-level, where ER and EP are event-based recall and precision as defined in [64]. Each metric captures different properties about the results, and thus is able to tell the prediction power in term of spatial and temporal consistency. For each method, we reported all metrics on each AU and their averages.

Network settings and training: We trained the CNNs with mini-batches of 196 samples, a momentum of 0.9 and weight decay of 0.0005. All models were initialized with learning rate of $1e-3$, which was further reduced manually whenever the validation loss stopped decreasing. The implementation was based on the Caffe toolbox [123] with modifications to support multi-label cross-entropy loss. For training LSTMs, we set an initial learning rate of $1e-3$, momentum of 0.9, weight decay 0.97, and RMSProp for stochastic gradient descent. All gradients were computed using back-propagation through time (BPTT) on 10 subsequences randomly sampled from training video. All sequences were 1300 frames long, and the first 10 frames were disregarded during the backward pass, as they carried insufficient temporal context. In the end, our network went through about 10 passes over the full training set. The matrix \mathbf{W} were randomly initialized within $[-0.08, 0.08]$. As AU data is heavily skewed, i.e., some AUs occur rarely and only a sparse subset of AU occur at a time, randomly sampled the sequences could cause LSTMs biased to negative predictions. As a result, we omitted training sequences with less than 1.5 active AUs per frame. All experiments were performed using one NVidia Tesla K40c GPU.

4.3.3 Evaluation of learned representation

To answer the question whether individual differences can be reduced by feature learning, we first evaluated the fc7 features with standard features in AU detection, including shape, Gabor, and SIFT features. Because such features for AU detection are unsupervised, for fairness, we used a pre-trained model of one dataset to test on another, i.e., fc7 features for BP4D were extracted using CNNs trained on GFT, and vice versa.

Fig. 4.3 shows the t-SNE embeddings of frames represented by SIFT, VGG face descriptor [190] and fc7 features, and visualize the effect of individual differences by coloring in terms of subjects. As can be seen in the first column, SIFT exhibits strong distributional biases, where the frames from the same subject tend to be closer in the feature space. Similarly, as shown in the second column, VGG network preserves more identity information because the network was originally trained for recognition purpose [190]. As can be seen in the second row where we colored

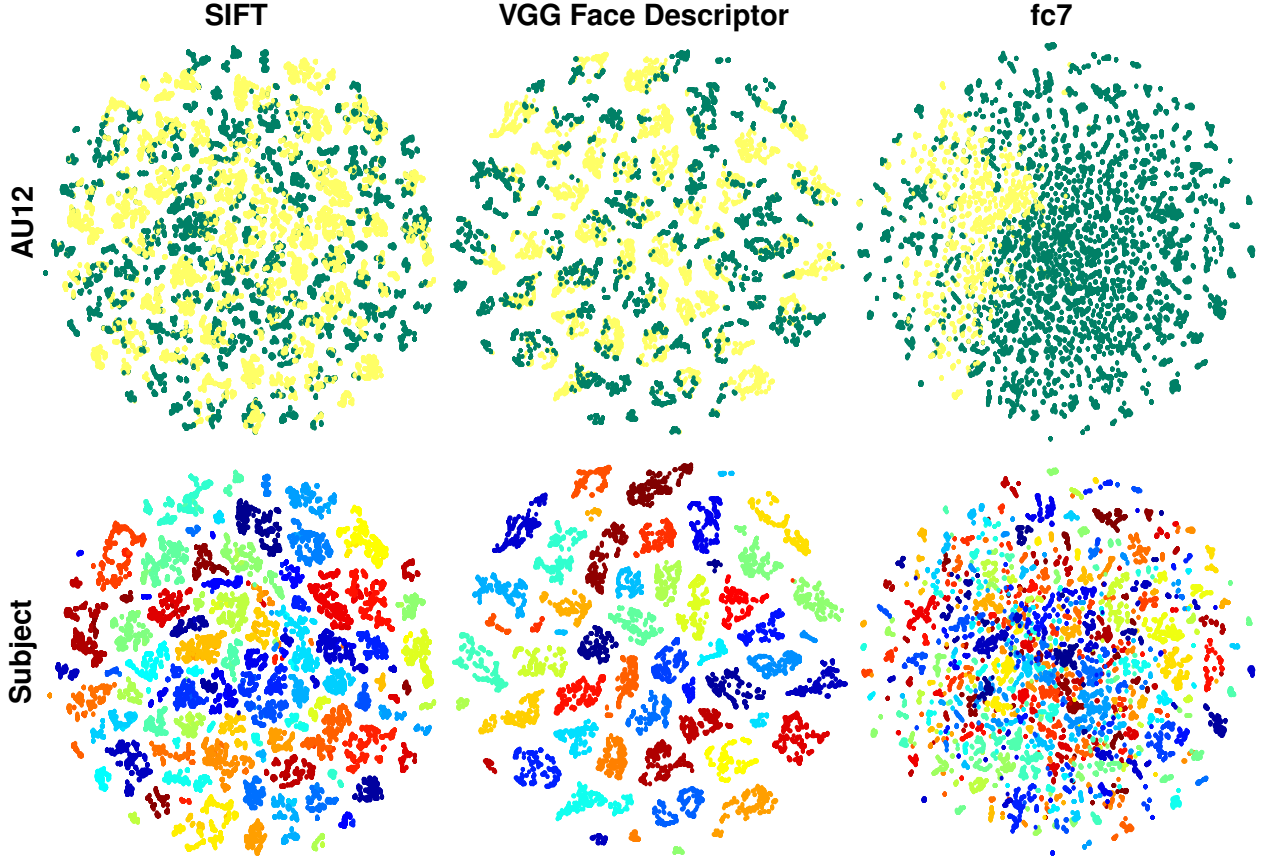


Figure 4.3: A visualization of t-SNE embedding using SIFT, VGG face descriptor [190] and fc7 features on the BP4D dataset by coloring samples in term of AU12 (**top row**) or subjects (**bottom row**). The clustering effect in SIFT features and VGG face descriptors reveal that face images encode not only information about facial AUs, but more on identities of subjects. The learned fc7 features are optimized for multi-label AU classification, and thus reduce such influence.

frames in terms of different subjects, the separation between subjects becomes more obvious than SIFT. On the other hand, as shown in third column, although our network is trained on using another exclusive dataset, fc7 features show great invariance to individual differences due to its supervised information for multi-label AU detection. Furthermore, on the second row, fc7 shows much lower sensitivity to subject identity, showing that the subject-wise differences were reduced. This serves as one evidence that the learned fc7 features can better preserve information for classification while reducing less useful information such as subject identity.

As a quantitative evaluation, we treated the frames from each subject as a distribution, and computed the distance between two subjects as Jensen-Shannon (JS) divergence [149]. Explicitly, we first computed a mean vector μ_s for each subject s in the feature space, and then squeezed μ_s using a logistic function $\sigma(a) = \frac{1}{1+e^{-a/m}}$ (m is median of μ_s as the median heuristic) and unity normalization, so that each mean vector can be interpreted as a discrete probability distribution, *i.e.*, $\mu_s \geq 0$, $\|\mu_s\|_1 = 1, \forall s$. Given two subjects p and q , we computed their JS divergence as:

$$D(\mu_p, \mu_q) = \frac{1}{2} D_{\text{KL}}(\mu_p \| \mathbf{m}) + \frac{1}{2} D_{\text{KL}}(\mu_q \| \mathbf{m}), \quad (4.5)$$

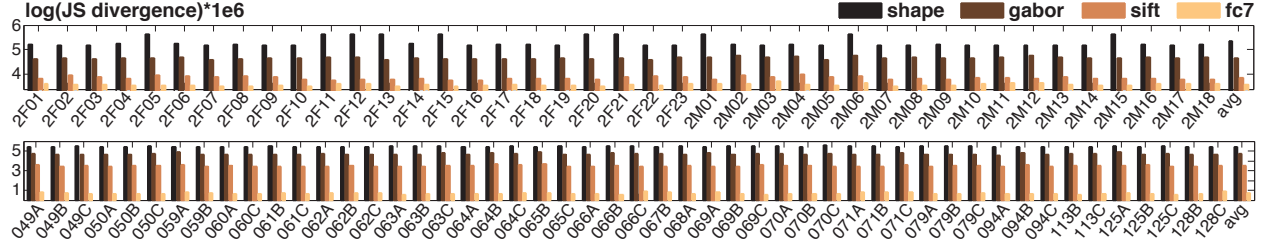


Figure 4.4: Analysis of subject-invariance on two datasets: BP4D (**top row**) and GFT (**bottom row**). Four representative features, shape, Gabor, SIFT and fc7, were compared (details in text). For display purpose, a computed divergence d is normalized by $\log(d) \times 1e6$.

where $\mathbf{m} = \frac{1}{2}(\boldsymbol{\mu}_p + \boldsymbol{\mu}_q)$ and $D_{\text{KL}}(\boldsymbol{\mu}_p, \mathbf{m})$ is the discrete KL divergence of $\boldsymbol{\mu}_p$ from \mathbf{m} . JS divergence is symmetric and smooth, and has been shown effective in measuring the dissimilarity between two distributions (e.g., [261]). Higher value of $D(\boldsymbol{\mu}_p, \boldsymbol{\mu}_q)$ tells larger mismatch given distributions for two subjects. Fig. 4.4 shows the distributional divergence for each subject p , which is computed by summing over $D(\boldsymbol{\mu}_p, \boldsymbol{\mu}_q), \forall q \neq p$. As can be seen, SIFT consistently reached a lower divergence than Gabor, providing an evidence that local descriptor (SIFT) is more robust to appearance changes compared to holistic ones (Gabor). This also serves as a possible explanation why SIFT consistently outperformed Gabor as found in [295]. Overall, fc7 yields much lower divergence compared to other popular engineered features, implying reduced individual differences.

4.3.4 Evaluation of detection performance

This section evaluates the performance of the proposed network on BP4D and GFT datasets. Below we summarize alternative methods, and then provide observations and discussion in hope to answer several fundamental questions.

Alternative methods: For evaluation, we compared a baseline SIFT method, a standard multi-label CNN, and feature-based state-of-the-arts. The first alternative approach is a baseline SVM with SIFT feature, which has been shown to outperform other appearance descriptors (i.e., Gabor/Daisy) [295]. Because SIFT is unsupervised, for fairness, we also evaluated a between-dataset protocol to train AlexNet on the other dataset, termed as ANet^T. fc7 features extracted by ANet^T were then used in comparison with SIFT descriptors. Linear SVMs were utilized as the base classifier, which also implicitly tells how separable different features are, i.e., higher classification rate suggests an easier linear separation, which supports the idea that a good representation could reduce the burden of designing a sophisticated classifier. We evaluated ANet^T on a 3-fold protocol, while we expect similar results could be obtained using 10-fold. Another alternative is our modified AlexNet (ANet), as mentioned in Sec. 4.2.1, with slightly different architecture and loss function (multi-label cross-entropy instead of multi-class softmax). ANet stood for a standard multi-label CNN, a representative of *feature learning* methods. On the other hand, CPM [277] and JPML [285] are feature-based state-of-the-art methods reported on the two datasets, while tackling the AU detection problem from different perspectives. Both CPM and JPML used SIFT features following [277, 285]. CPM is one candidate method of *personalization*, which addresses the distributional shift in the feature space by progressively adapting a classifier to best separate a test subject. On the other hand, JPML models *AU correlations*, and meanwhile considers patch learning to se-

Table 4.2: F1-frame on GFT dataset [50]

AU	3-fold protocol					cross	10-fold protocol				
	SIFT	CPM	JPML	ANet	Ours	ANet ^T	SIFT	CPM	JPML	ANet	Ours
1	12.1	30.7	17.5	31.2	29.9	9.9	30.3	29.9	28.5	57.5	63.0
2	13.7	30.5	20.9	29.2	25.7	10.8	25.6	25.7	25.5	61.4	74.6
4	5.5	–	3.2	71.9	68.9	45.4	–	–	–	75.9	68.5
6	30.6	61.3	70.5	64.5	67.3	46.2	66.2	67.3	73.1	61.6	66.3
7	26.4	70.3	65.5	67.1	72.5	51.5	70.9	72.5	70.2	80.1	74.5
10	38.4	65.9	67.9	42.6	67.0	23.5	65.5	67.0	67.1	54.5	70.3
12	35.2	74.0	74.2	73.1	75.1	55.2	74.2	75.1	78.3	79.8	78.2
14	55.8	81.1	52.4	69.1	80.7	62.8	79.6	80.7	61.4	84.2	80.4
15	9.5	25.5	20.3	27.9	43.5	14.2	34.1	43.5	28.0	40.3	50.5
17	31.3	44.1	48.3	50.4	49.1	34.2	49.2	49.1	42.4	61.6	61.9
23	19.5	19.9	31.8	34.8	35.0	21.8	28.3	35.0	29.6	47.0	58.2
24	12.9	27.2	28.5	39.0	31.9	18.9	31.9	31.6	28.0	56.3	50.8
Avg	24.2	48.2	41.8	50.0	53.9	32.9	50.5	52.4	48.4	63.4	66.4

lect important facial patches for specific AUs. All experiments followed protocols as described in Sec. 4.3.2.

Results and discussion: Tables 4.2 and 4.3 show F1 metrics reported on 12 AUs; “Avg” for the mean score of all AUs. The bar plots show the averaged F1-norm and F1-event across all AUs. For detailed F1-frame and F1-event of individual AUs, please see supplementary materials. According to the results, we discuss our findings in hope to answer three fundamental questions:

1) *Could we learn a representation that better generalizes across subjects or datasets for AU detection?* On both datasets, compared to SIFT, ANet^T trained with a cross-dataset protocol on average yielded higher scores with a few exceptions. In addition, for both 3-fold and 10-fold protocols where ANet was trained on exclusive subjects, ANet consistently outperformed SIFT over all AUs. These observations provide an encouraging evidence that the learned representation was transferable even when being tested across subjects and datasets, which also coincides with the findings in the image and video classification community [131, 224]. On the other hand, as can be seen, ANet trained within datasets leads to higher scores than ANet^T trained across datasets. This is because of the dataset biases (*e.g.*, recording environment, subject background, etc.) that could cause distributional shifts in the feature space. In addition, due to the complexity of deep models, the performance gain of ANet trained on more data (10-fold) became larger than ANet trained on 3-fold, showing the generalizability of deep models increases with the growing number of training samples. Surprisingly, compared to SIFT trained on 10-fold, ANet trained on 3-fold showed comparable scores, even with $\sim 30\%$ fewer data than what SIFT was used. All suggests that features less sensitive to the identity of subjects could improve AU detection performance.

2) *Could the learned temporal dependencies improve performance, and how?* The learned temporal dependencies was aggregated into the hybrid network denoted as “ours”. On both 3-fold and 10-fold protocols, our hybrid network consistently outperformed ANet in all metrics. This improvement can be better told by comparing their F1-event scores. The proposed network used

Table 4.3: F1-frame metrics on BP4D dataset [282]

AU	3-fold protocol					cross	10-fold protocol				
	SIFT	CPM	JPML	ANet	Ours	ANet ^T	SIFT	CPM	JPML	ANet	Ours
1	21.1	43.4	32.6	40.3	31.4	32.7	46.0	46.6	33.9	54.7	70.3
2	20.8	40.7	25.6	39.0	31.1	26.0	38.5	38.7	36.2	56.9	65.2
4	29.7	43.3	37.4	41.7	71.4	29.0	48.5	46.5	42.2	83.4	83.1
6	42.4	59.2	42.3	62.8	63.3	61.9	67.0	68.4	62.9	94.3	94.7
7	42.5	61.3	50.5	54.2	77.1	59.4	72.2	73.8	69.9	93.0	93.2
10	50.3	62.1	72.2	75.1	45.0	67.4	72.7	74.1	72.5	98.9	99.0
12	52.5	68.5	74.1	78.1	82.6	76.2	83.6	84.6	72.0	94.4	96.5
14	35.2	52.5	65.7	44.7	72.9	47.1	59.9	62.2	62.6	82.9	86.8
15	21.5	36.7	38.1	32.9	34.0	21.7	41.1	44.3	38.2	55.4	63.3
17	30.7	54.3	40.0	47.3	53.9	47.1	55.6	57.5	46.5	81.1	82.7
23	20.3	39.5	30.4	27.3	38.6	21.6	40.8	41.7	38.3	63.7	73.5
24	23.0	37.8	42.3	40.1	37.0	31.3	42.1	39.7	41.5	74.3	81.6
Avg	32.5	50.0	45.9	48.6	53.2	43.4	55.7	56.5	51.4	77.8	82.5

CNNs to extract spatial representations, stacked LSTMs to model temporal dependencies, and then performs a spatiotemporal fusion. From this view, predictions with fc7 features can be treated as a spacial case of ANet—a linear hyperplane with a portion of intermediate features. In general, adding temporal information helped predict AUs except for a few in GFT. A possible explanation is that in GFT, the head movement was more frequent and dramatic, and thus makes temporal modeling of AUs more difficult than moderate head movements in BP4D. In addition, adding temporal prediction into the fusion network attained an additional performance boost, leading to the highest F1 score on both datasets with either the 3-fold or the 10-fold protocols. This shows that the spatial and temporal cues are complementary, and thus is crucial to incorporate all of them into an AU detection system.

3) *Would jointly considering all issues in one framework improve AU detection?* This question aims to examine if the hybrid network would improve the performance of the methods that consider the aforementioned issues independently. To answer this question, we implemented CPM [277] as a personalization method that deals with representation issues, and JPML [285] as a multi-label learning method that deals with AU relations. Our modified ANet served as a feature learning method. All parameters settings were determined following the descriptions in the original papers. To draw a valid discussion, we fixed the exact subjects for all methods. Observing 3-fold on both datasets, the results are mixed. In GFT, ANet and JPML achieved 3 and 2 highest F1 scores; in BP4D, CPM and ANet reached 5 and 2 highest F1 scores. One potential explanation is, although CNNs possess highest degree of expressive power, the number training samples in 3-fold (33% left out for testing) were insufficient and might resulted in overfitting. In the 10-fold experiment, when the number of training samples was abundant, the improvements became clearer, as the parameters of the complex model can be better trained to fit the task. Overall, in most cases, our hybrid network outperformed alternative approaches by a significant margin, showing the benefits for considering all issues in one framework.

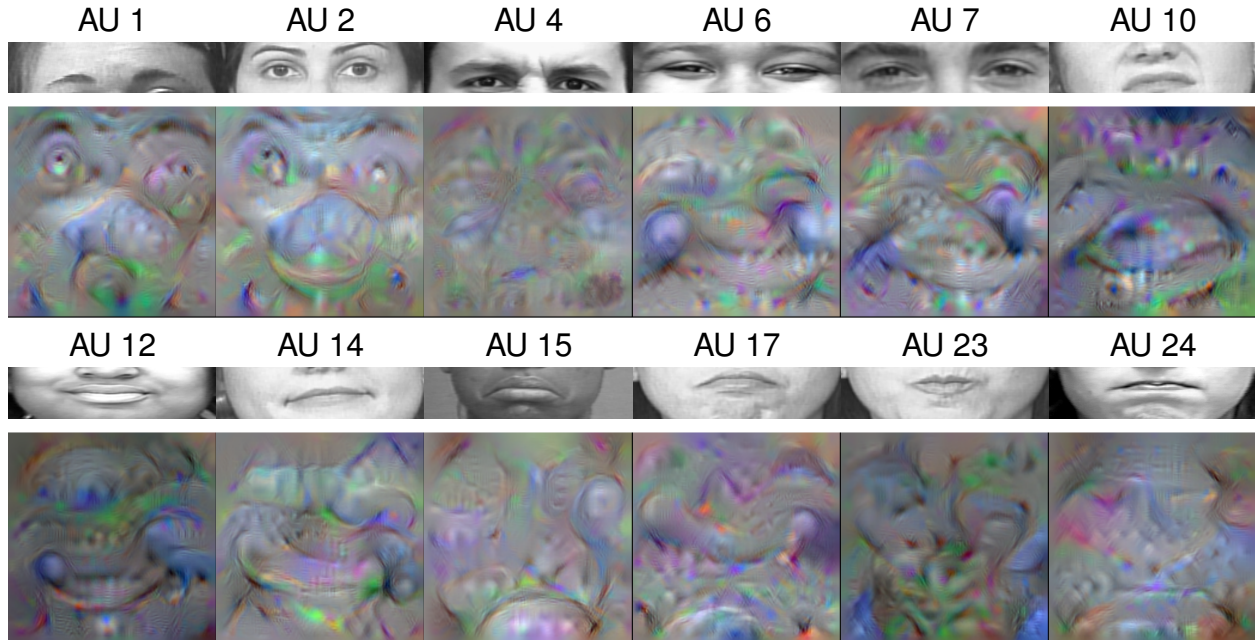


Figure 4.5: Synthetically generated images to maximally activate individual AU neurons in the fc8 layer of CNN, trained on GFT [50], showing what each AU model “wants to see”. The learned models show high agreement on attributes described in FACS [77]. (best view electronically)

4.3.5 Visualization of learned AU models

To better understand and interpret the proposed network, we implement a gradient ascent approach [223, 272] to visualize each AU model. More formally, we solve for such input image \mathcal{I}^* by solving the optimization problem:

$$\mathcal{I}^* = \arg \max_{\mathcal{I}} A_{\ell}(\mathcal{I}) - \Omega(\mathcal{I}), \quad (4.6)$$

where $A_{\ell}(\mathcal{I})$ is an activation function for the ℓ -th unit of the fc8 layer given an image \mathcal{I} , and $\Omega(\cdot)$ is a regularization function that penalizes \mathcal{I} to enforce a natural image prior. In particular, we implemented $\Omega(\cdot)$ as a sequential operation of L_2 decay, clipping pixels with small norm, and Gaussian blur [272]. The optimization was done by iteratively updating a randomized and zero-centered image with the backprop gradient of $A_{\ell}(\mathcal{I})$. In other words, each pixel of \mathcal{S} was renewed gradually to increase the activation of the ℓ -th AU. This process continued until 10,000 iterations.

Fig. 4.5 shows our visualizations of each AU model learned by the CNN architecture described in Sec. 4.2.1. As can be seen, most models match the attributes described in FACS [77]. For instance, model AU12 (lip corner puller) exhibits a strong “ \smile ” shape to the mouth, overlapped with some vertical “stripes”, implying the appearance of teeth is commonly seen in AU12. Model AU14 (dimpler) shows the dimple-like wrinkle beyond lip corners, which, compared to AU12, gives the lip corners a downward cast. Model AU15 (lip corner depressor) shows a clear “ \frown ” shape to the mouth, producing an angled-down shape at the corner. For upper face AUs, model AU6 (cheek raiser) captures deep texture of raised-up cheeks, narrowed eyes, as well as a slight “ \smile ” shape to the mouth, suggesting its frequent co-occurrence with AU12 in spontaneous smiles. Models AU1 and AU2 (inner/outer brow raiser) both capture the arched shapes to the eyebrows,

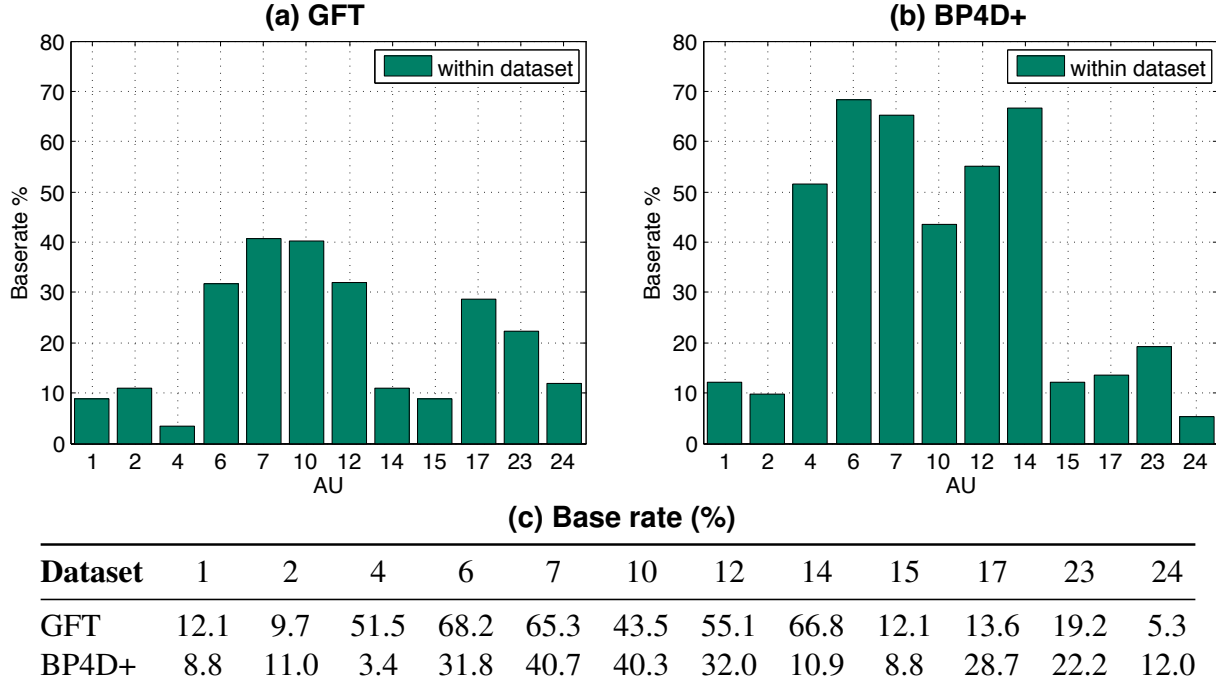


Figure 4.6: Distributions of AU base rates in two of the largest spontaneous datasets used in this study: (a) GFT [50] and (b) BP4D+ [283]. (c) shows the exact base rate of individual AUs of each dataset. Base rate is defined as the frequency of a particular AU occurring in video frames of the entire dataset. Note that we only count the frames that can be validly face tracked and annotated completely with 12 AUs.

horizontal wrinkles above eyebrows, as well as the widen eye cover that are stretched upwards. Model AU4 (brow lowerer) captures the vertical wrinkles between the eyebrows and narrowed eye cover that folds downwards.

Our visualizations suggest that the CNN was able to identify these important spatial cues to discriminate AUs, even though we did not ask the network to specifically learn these AU attributes. In addition, the global structure of a face was actually preserved throughout the network, despite that convolutional layers were designed for local abstraction (*e.g.*, corners and edges as shown in Fig. 4.2(d)). The widespread agreements between the synthetic images and FACS [77] confirm that the learned representation is able to describe, and thus reveal these attributes across multiple AUs. This was not shown possible in standard hand-crafted features in AU detection (*e.g.*, shape [118, 161], SIFT [277, 285], LBP [125, 258], or Gabor [258]). To the best of our knowledge, this is the first time to visualize how machines see facial AUs.

4.4 Multi-label sampling strategies to address class imbalance

In spontaneous datasets, the incidence of AU class labels varies greatly. As shown in Fig. 4.6, it is usual that certain AUs appear in a much higher base rate (*e.g.*, AUs 6, 7 and 12) while others are scarcely represented (*e.g.*, AUs 1, 2 and 15). Without any treatment on such class imbalance, classifiers trained on these imbalance distribution could cause predictions in favor of major classes (classes with higher base rate). As can be observed in the experimental section, every classifier

shows relatively worse performance in minor classes such as AUs 1, 2, and 15. This is because most classifiers are designed to minimize a global error measurement. When imbalanced class distribution is present, mis-classification of minor/rare classes does not contribute greatly in the global performance measure, resulting in a natural inclination to benefit the most frequent classes. However, in multi-label AU classification scenario, correct annotation should be valued equally for individual AUs, instead of only for the most common ones.

Class imbalance has been a well-studied problem in binary and multi-class classification. Classifiers could be improved if they had access to more samples of rare classes. We refer interested readers to comprehensive reviews (e.g., [111, 193, 230]). In these literature, the imbalance levels are often referred as to *imbalance ratio* or *skewness*, which is computed as the ratio of the number of samples in the majority class over the number of samples in the minority class. Standard approaches in learning from such imbalanced classes can be broadly categorized into the follows:

- **Resampling:** Resampling techniques aim at producing a new dataset from the original one. To balance the distributions between frequently and rarely occurring classes, oversampling or undersampling approaches are typically used. Another trend employs synthesis for the minority class, *i.e.*, growing the population of minority classes by synthesizing samples in the feature space (e.g., SMOTE [31]). Because the sampling is done at data-level, resampling can be seen as a classifier-independent approach that applies to most problems.
- **Classifier adaptation/cost-sensitive learning:** This type of methods is classifier-dependent. The goal here is to modify a classification algorithm to further emphasize the contributions from a minor class. The imbalanced nature of the data is addressed by either re-estimating sample distribution (such as STM discussed in Chapter 3), reinforcing the algorithm toward the minority class, or re-weighting training losses inversely proportional to each class size.

Although imbalance learning has been a well-known problem with rather comprehensive studies, most existing methods only consider sampling for only one majority class and one minority class. Because facial images contain several AU class labels per sample, the complexity of the sampling problem is higher, making standard sampling approaches not directly applicable.

As illustrated in Fig. 4.6, a clear imbalanced nature among AU classes exists in spontaneous datasets, such as GFT [93] and BP4D+ [283]. Note that the GFT dataset used in this section is a larger, renewed collection of 150 annotated subjects compared to the version of 50 subjects used in earlier experiments of Chapter 3 and Sec. 4.3.4. For example, in BP4D+, the most frequently occurring AU has more than 10 times more samples than the least occurring one. Recall that in an end-to-end supervised framework, “mini-batches” are randomly sampled from the training set for updating parameters in stochastic gradient descent. However, selecting images randomly causes two issues for properly training an end-to-end supervised model. First, as illustrated in the top row of Fig. 4.7, the number of AU presence *between* batches is imbalanced. This can potentially make the training procedure rather unstable for the end classification goal. Second, the number of AU presence *within* batches is also imbalance. As noted earlier, having an imbalanced AU distribution can cause the learned model to favor the majority class. Due to these differences between AU class distributions, a multi-label sampling strategy is of specific need.

In this section, we will introduce two multi-label sampling strategies to attack this specific imbalance in the multi-label space: multi-label stratification in Sec. 4.4.1, and multi-label minority oversampling majority undersampling (MOMU) in Sec. 4.4.2. Then, in Sec. 4.4.3, we will evaluate different multi-label sampling strategies in both training and test phases.

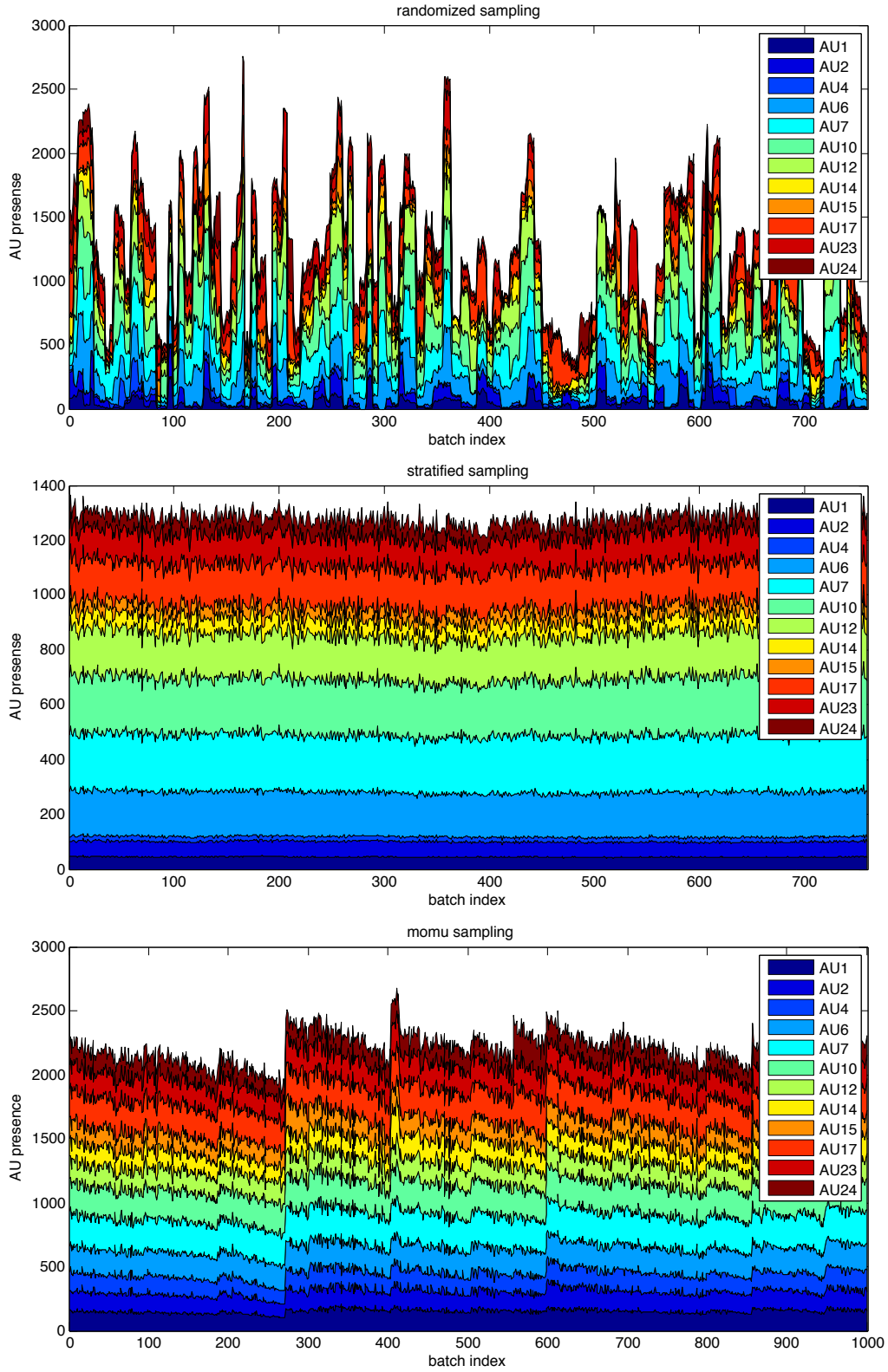


Figure 4.7: Distributions of AU classes in each mini-batch using different sampling strategies: **(top)** random sampling, **(middle)** multi-label stratification, **(bottom)** MOMU sampling. As can be seen in random sampling, the number of AU presence *between* and *within* batches are dramatically different. (see text for details)

Algorithm 3: Multi-label stratification

Input : Dataset \mathcal{D} annotated with L classes, the number of batches B
Output: Processed batches $\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^B$

- 1 Compute N_ℓ ($\ell = 1, \dots, L$) as the number of the ℓ -th AU in the dataset \mathcal{D} ;
- 2 **while** $|\mathcal{D}| > 0$ **do**
- 3 $\ell \leftarrow \arg \min_j N_j$; // Find the AU with fewest samples in \mathcal{D}
- 4 $\mathcal{D}_\ell \leftarrow \{(\mathbf{x}_i, Y_i) \in \mathcal{D} | Y_i^\ell = 1\}$ // Collect (image,label) of the AU with fewest samples
- 5 **if any of \mathcal{B}_i is not full then**
- 6 Distribute \mathcal{D}_ℓ evenly into all batches $\{\mathcal{B}_i\}_{i=1}^B$;
- 7 $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_\ell$;
- 8 Update N_ℓ ($\ell = 1, \dots, L$) as the number of the ℓ -th AU in the dataset \mathcal{D} ;
- 9 **return** \mathcal{B} ;

4.4.1 Multi-label stratification

We first propose an algorithm for balancing the distribution *between* batches. The idea was inspired by standard methods on *stratified sampling*, which utilizes independent sampling among each sub-population when sub-populations vary within an overall population. Algorithm 3 summarizes the proposed multi-label stratification approach. The input to the algorithm is a dataset $\mathcal{D} = \{\mathbf{x}_i, Y_i\}_{i=1}^{|\mathcal{D}|}$ annotated with L classes (*i.e.*, $Y_i \in \mathbb{R}^L$). Suppose $|\mathcal{D}|$ is the number of images in the dataset, and Y_i^ℓ is the ℓ -th AU annotation of the i -th image. The multi-label stratification starts by computing the total number of examples for each AU class, and then iteratively distributing images that contain the AU with the fewest samples. The distribution is performed evenly into each batch until the complete dataset is distributed ($|\mathcal{D}| = 0$) or the desired number of batches is collected. This normally terminates after $(L + 1)$ iterations (L iterations for distributing all AUs and 1 iteration for distributing samples with no AUs annotations), but could end up less if samples of certain AU class have been already distributed. Note that images without any AU annotations still carry information about being an opposite (negative) class for each AU, and thus we enforce the sampling to terminate until the dataset is empty.

This algorithm is performed in a greedy perspective. That is, we aim to have labels in every batch as diverse as possible. If images that contain minority class labels are not evenly distributed in priority, it is likely that some batches contain zero occurrence of rare labels, resulting in biased learning that is difficult to be repaired subsequently. On the other hand, due to the availability of more samples, distributing later the images with labels from the majority classes maintains to guide the model towards a desired parametric update.

The middle row of Fig. 4.7 illustrates the distribution of AU presence in each mini-batch. As can be seen, the number of AU presence are much more balanced between batches compared to the random sampling shown in the first row. However, each vertical slice (*i.e.*, AU distribution in one batch) still exhibits dramatic imbalanced AU distribution. For example, minor AUs (*e.g.*, 1, 2 and 4) are outnumbered by major AUs (*e.g.*, 7, 10, and 12). To balance the distribution of AU presence within batches, we are driven to the next sampling strategy.

Algorithm 4: Multi-label minority oversampling majority undersampling (MOMU)

Input : Dataset \mathcal{D} annotated with L labels, the size of a mini-batch N , the number of batches B , sampling step size S

Output: Processed batches $\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^B$

- 1 Compute N_ℓ ($\ell = 1, \dots, L$) as the number of the ℓ -th AU in the dataset \mathcal{D} ;
- 2 **for** $i = 1, \dots, B$ **do**
- 3 $\ell \leftarrow \arg \min_j N_j$;
- 4 **while** $|\mathcal{B}_i| < N$ **do**
- 5 **if** $N_\ell < S$ **then**
- 6 Restore all images that contain AU ℓ back to \mathcal{D} ;
- 7 **continue**;
- 8 $\mathcal{D}_\ell \leftarrow \{ \{(\mathbf{x}_i, Y_i)\}_{i=1}^S \in \mathcal{D} | Y_i^\ell = 1 \}$ // Sample S (image,label) pairs of the ℓ -th AU
- 9 $\mathcal{D} \leftarrow \mathcal{D} \setminus \mathcal{D}_\ell$;
- 10 Compute n_ℓ ($\ell = 1, \dots, L$) as the number of the ℓ -th AU in current batch distribution;
- 11 $\ell \leftarrow \arg \min_j n_j$; // Find the AU with fewest samples in current batch
- 12 Update N_ℓ ($\ell = 1, \dots, L$) as the number of the ℓ -th AU in the dataset \mathcal{D} ;
- 13 **return** \mathcal{B} ;

4.4.2 Multi-label minority oversampling majority undersampling (MOMU)

To the best of our knowledge, despite of numerous studies on multi-label classification and deep learning, there is limited discussion on how class imbalance of multi-label data can be systematically addressed between and within batches. As we have observed in the previous section, both random sampling and multi-label stratification suffer from dramatic imbalanced distributions *within* each mini-batch. This drives us to the next strategy termed multi-label minority oversampling majority undersampling (MOMU).

Algorithm 4 summarizes the proposed multi-label MOMU strategy. For each batch, MOMU proceeds by progressively filling the (image,label) pairs in a greedy manner. Similar to multi-label stratification, as discussed in the previous section, MOMU starts by picking S images that contain the AU with fewest samples in the population distribution (the AU distribution of an entire dataset). Because each image contains multiple labels, adding S images into the current batch can simultaneously increase the base rate for other AUs. These S samples are then removed from the dataset to ensure a maximal use of annotated data. In the next iteration, MOMU picks the AU with fewest samples in current batch, and then samples next S images (without replacement) that contain this particular AU. In this way, we ensure that the AU with the fewest samples can be always compensated through sampling. We repeat the procedure for the desired number of B batches until all batches are filled. Note that during sampling, it is likely that a particular minority class runs out of samples ($N_\ell < S$). In this case, we simply restore to the dataset with all images that contain AU ℓ , and then continue sampling images that contain this particular AU class. Because the images are added into each batch consecutively with guarantees to contain at least an active AU, the class distribution *between* batches will remain around a similar scale. More importantly, as we intentionally fill in images for the minority class, the class distribution *within* batches can be also



Figure 4.8: An illustration of random cropping (*i.e.*, translation, rotation, scale) as standard data augmentation for training deep networks. (image credit from [42, 43])

controlled within a balanced range.

The bottom row of Fig. 4.7 illustrates the AU distribution after the multi-label MOMU. As can be seen, the number of AU presence between batches remain in similar scale, while the AU distribution *within* batches becomes much more balanced. As we will show in the subsequent evaluation, such balanced distribution consistently improves training performance as well as test performance in both within-dataset and between-dataset scenarios. To our knowledge, this could serve as one of the first attempts that address multi-label sampling for imbalanced datasets in the context of stochastic training. Although we will illustrate only performance on deep learning models, we believe the same idea can be applied to more models such as multi-label stochastic SVMs [143].

Comparison with existing methods: Recall that most literature consider strategies that involve either resampling or classifier adaptation/cost-sensitive learning. One interpretation of MOMU is its behavior as a hybrid of both. As in standard deep learning, augmentation for training data is often done through random cropping of the input image (as illustrated in Fig. 4.8). From this perspective, MOMU takes the full advantage of both types of strategies by achieves resampling through sampling the minor classes in the image space, and cost-sensitive learning through balancing the contributions of different classes in the feature space.

4.4.3 Evaluation of different multi-label sampling strategies

In this section, we evaluate the effects of multi-label sampling strategies in terms of improvements in training and test performance. Following Sec. 4.3.2, we used a 10-fold data split protocol, *i.e.*, 80% of subjects for training, 10% for validation and the remaining 10% for test. We will report mainly in terms of F1-score and AUC due to their popularity in computer vision problems.

Evaluation of training performance

Fig. 4.9 reports the training performance on the GFT dataset in terms of F1-score (y-axis) and the number of iterations (x-axis). Three sampling strategies, *i.e.*, standard random sampling, multi-label stratification, and multi-label MOMU, were evaluated. The reason we picked F1-score as the evaluation metric is because of its sensitivity in true positives, which we believe can closer describe human perception compared to accuracy-based measures. In other words, given a distribution skewed toward negative samples in each AU class, we believe humans are more sensitive about a model classifying correctly on a positive sample than a negative one. If an accuracy-based metric (*e.g.*, S-score or kappa [93], AUC, or accuracy) is used over skewed classes, one may not be able to distinguish the classifier’s performance on top of the true positives (see also [85]). Having such metric is able to provide a more accurate description about performance of human’s interest.

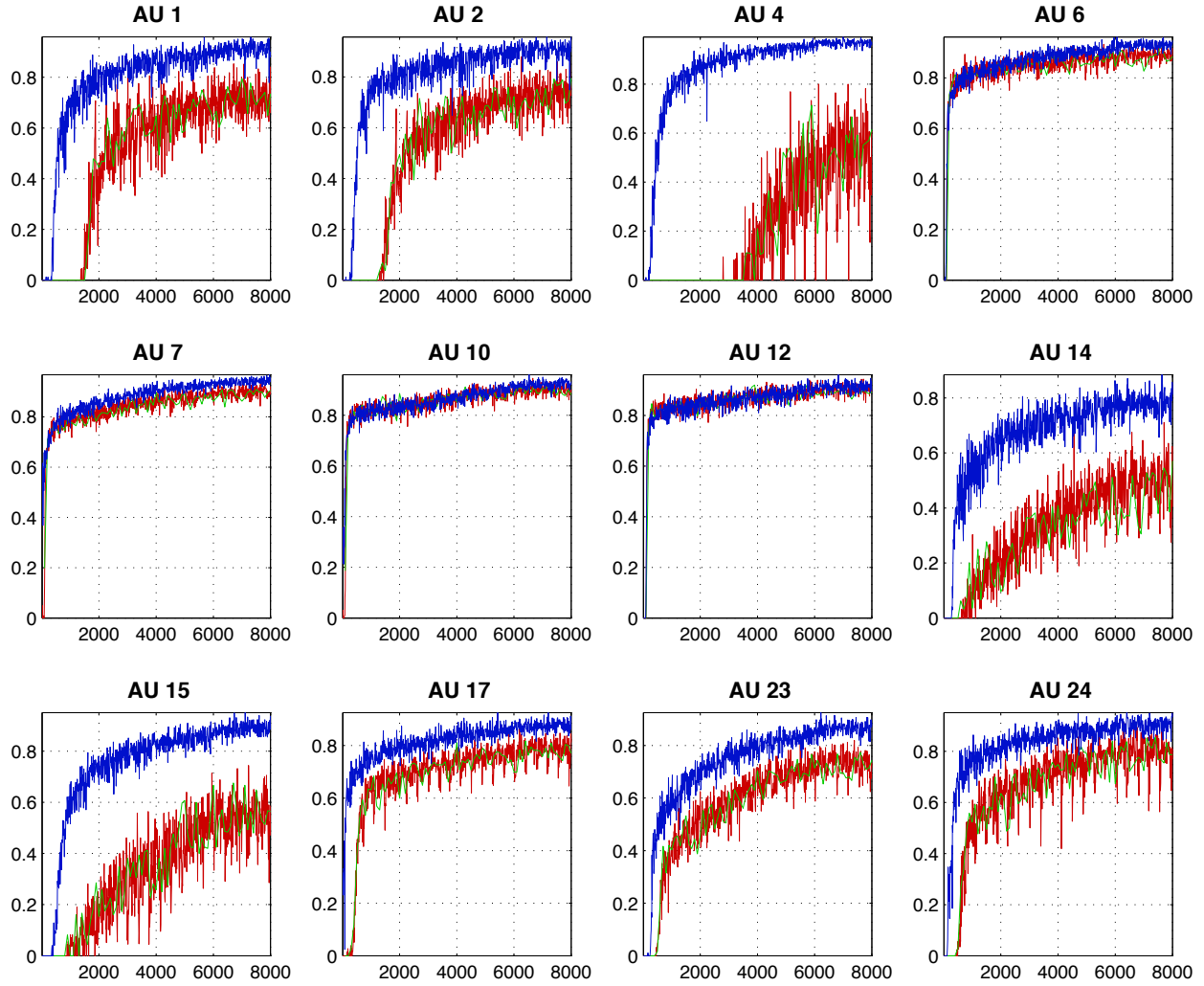


Figure 4.9: Comparison of training performance on the GFT dataset in terms of F1-score (y-axis) vs the number of iterations (x-axis) over different sampling strategies: (Red) random sampling, (Green) multi-label stratification, (Blue) multi-label MOMU. As can be observed, for conventional random sampling and multi-label stratification, the performance of minority AUs, such as AUs 4 ($BR_4 = 3.4\%$) and 15 ($BR_{15} = 8.8\%$), remains rather low even after training phase with 8000 iterations. (the curve is higher better)

Table 4.4: Performance evaluation of different sampling strategies in terms of within-dataset (**top**) and between-dataset (**bottom**) scenarios in the GFT dataset [50]: random sampling, multi-label stratification, and multi-label MOMU sampling. The evaluation metrics are S: Kappa, AUC: Area Under the ROC Curve, PA: positive agreement or F1, NA: negative agreement.

	Random sampling					Multi-label stratification					Multi-label MOMU				
	AU	S	AUC	PA	NA	AU	S	AUC	PA	NA	AU	S	AUC	PA	NA
Within-dataset	1	.73	.89	.44	.9	1	.65	.85	.38	.92	1	.63	.9	.47	.86
	2	.65	.83	.41	.87	2	.53	.79	.35	.88	2	.56	.82	.38	.84
	4	.88	.6	.	.97	4	.68	.62	.	.97	4	.77	.77	.29	.93
	6	.62	.92	.73	.76	6	.58	.91	.73	.78	6	.59	.91	.71	.71
	7	.59	.86	.72	.65	7	.59	.85	.72	.67	7	.57	.83	.73	.58
	10	.59	.89	.68	.73	10	.54	.89	.68	.73	10	.56	.89	.67	.67
	12	.69	.94	.72	.83	12	.64	.94	.75	.83	12	.68	.94	.75	.8
	14	.74	.82	.05	.93	14	.54	.83	.27	.92	14	.46	.82	.4	.78
	15	.75	.8	.17	.93	15	.41	.79	.14	.92	15	.44	.77	.29	.78
	17	.49	.77	.32	.83	17	.42	.8	.47	.76	17	.14	.79	.49	.57
	23	.49	.75	.39	.8	23	.37	.78	.38	.79	23	.24	.76	.48	.62
	24	.8	.89	.13	.95	24	.72	.89	.44	.93	24	.51	.9	.41	.81
	avg	.67	.83	.4	.84	avg	.56	.83	.44	.84	avg	.51	.84	.51	.75
Between-dataset	1	.75	.6	.03	.93	1	.76	.6	.	.93	1	.62	.67	.16	.86
	2	.66	.57	.	.9	2	.66	.59	.	.9	2	.5	.6	.13	.82
	4	-.12	.43	.07	.51	4	-.01	.42	.07	.58	4	.07	.47	.08	.65
	6	-.15	.74	.52	.05	6	-.08	.79	.52	.13	6	-.01	.73	.54	.31
	7	.31	.68	.63	.34	7	.4	.72	.61	.51	7	.17	.69	.62	.28
	10	.33	.77	.32	.72	10	.36	.81	.33	.74	10	.23	.76	.46	.64
	12	.47	.84	.58	.7	12	.56	.87	.61	.77	12	.25	.82	.55	.6
	14	-.52	.57	.23	.18	14	-.36	.57	.25	.31	14	-.72	.55	.22	.03
	15	.73	.66	.01	.92	15	.73	.57	.01	.92	15	.06	.63	.23	.61
	17	.42	.64	.22	.79	17	.29	.62	.3	.7	17	-.01	.59	.4	.43
	23	.29	.61	.34	.7	23	.15	.62	.43	.57	23	-.02	.54	.32	.47
	24	.79	.66	.	.94	24	.79	.67	.01	.94	24	.47	.69	.19	.8
	avg	.33	.65	.25	.64	avg	.35	.65	.26	.67	avg	.13	.64	.32	.54

Table 4.5: Performance evaluation of different sampling strategies in terms of within-dataset (**top**) and between-dataset (**bottom**) scenarios in the BP4D+ dataset [283]: random sampling, multi-label stratification, and multi-label MOMU sampling. The evaluation metrics are S: Kappa, AUC: Area Under the ROC Curve, PA: positive agreement or F1, NA: negative agreement.

	Random sampling					Multi-label stratification					Multi-label MOMU				
	AU	S	AUC	PA	NA	AU	S	AUC	PA	NA	AU	S	AUC	PA	NA
Within-dataset	1	.66	.74	.19	.89	1	.66	.74	.19	.89	1	.44	.84	.43	.74
	2	.67	.72	.15	.9	2	.67	.72	.15	.9	2	.44	.88	.46	.72
	4	.67	.94	.83	.83	4	.67	.94	.83	.83	4	.77	.97	.88	.88
	6	.53	.9	.82	.59	6	.53	.9	.82	.59	6	.76	.94	.9	.79
	7	.79	.97	.91	.86	7	.79	.97	.91	.86	7	.87	.98	.94	.9
	10	.67	.94	.78	.85	10	.67	.94	.78	.85	10	.74	.96	.82	.88
	12	.74	.96	.87	.86	12	.74	.96	.87	.86	12	.82	.98	.91	.9
	14	.53	.87	.8	.63	14	.53	.87	.8	.63	14	.59	.9	.83	.53
	15	.71	.83	.15	.91	15	.71	.83	.15	.91	15	.65	.88	.38	.87
	17	.63	.82	.3	.89	17	.63	.82	.3	.89	17	.67	.87	.54	.89
	23	.63	.86	.44	.87	23	.63	.86	.44	.87	23	.72	.88	.6	.91
	24	.83	.87	.02	.95	24	.83	.87	.02	.95	24	.61	.95	.4	.84
	avg	.67	.87	.52	.84	avg	.67	.87	.52	.84	avg	.67	.92	.67	.82
Between-dataset	1	.49	.7	.28	.81	1	-.24	.72	.31	.32	1	-.19	.66	.3	.41
	2	.46	.71	.3	.79	2	-.3	.67	.32	.29	2	-.08	.65	.31	.48
	4	.07	.46	.	.69	4	.07	.57	.	.69	4	.1	.66	.25	.64
	6	.36	.86	.66	.64	6	.49	.89	.78	.66	6	.39	.88	.71	.62
	7	.45	.86	.74	.64	7	.67	.94	.86	.75	7	.53	.89	.79	.7
	10	.3	.92	.68	.55	10	.51	.94	.74	.75	10	.52	.93	.74	.74
	12	.66	.92	.81	.82	12	.57	.96	.75	.79	12	.6	.95	.79	.79
	14	-.22	.53	.	.54	14	-.22	.72	.	.54	14	-.16	.63	.12	.53
	15	.76	.74	.02	.93	15	.75	.66	.05	.93	15	.5	.56	.15	.79
	17	.65	.73	.15	.9	17	.71	.6	.01	.92	17	.26	.7	.28	.71
	23	.62	.76	.14	.89	23	.56	.68	.17	.87	23	.26	.75	.39	.7
	24	.84	.8	.02	.95	24	.83	.64	.	.95	24	.78	.77	.09	.93
	avg	.45	.75	.25	.76	avg	.37	.75	.33	.7	avg	.29	.75	.41	.67

As can be seen the red curve in Fig. 4.9, standard random sampling (as used in most deep learning literature) suffers from imbalanced AU distribution. For notational convenience, we denote base rate for the ℓ -th AU as BR_ℓ . The performance of minority AUs, such as AUs 4 ($BR_4 = 3.4\%$) and 15 ($BR_{15} = 8.8\%$), remains rather low even during the training phase with 8000 iterations. Multi-label stratification, as indicated by the green curve, exhibit a relatively smoother training curve because each mini-batch contains similar amount of AU presence, which would help avoid the network favoring prediction on the negative samples. However, as can be seen, multi-label stratification only ends up with similar performance because the AU distribution *within* each mini-batch remains dramatically biases as discussed in the previous section. The MOMU strategy, as indicated by the blue curve, shows significant improvement for minority classes, including AUs 1 ($BR_1 = 8.8\%$), 2 ($BR_2 = 10.9\%$), 4 ($BR_4 = 3.4\%$), and 15 ($BR_{15} = 8.8\%$). Not surprisingly, the performance of major AUs did not decrease notably even though the samples in the majority classes were under-sampled. This is mainly due to the high redundancy of the video frames shown in spontaneous datasets. In all, as indicates by the F1 scores, the multi-label MOMU strategy effectively guides the network with reliable training for the multi-label AU data.

Evaluation of test performance

For performance evaluation during the test phase, we provide in depth evaluation by reporting in Tables 4.4 and 4.5 detailed performance for individual AUs using four metrics: accuracy-based S-score (or kappa [93], threshold-based AUC, F1-based PA and NA. As described in Sec. 2.6, different metrics capture different aspects of prediction power that researchers in different field might find useful. To further analyze the improvements of multi-label MOMU against random sampling strategies, we picked AUC and PA following the settings in the previous section. Fig. 4.10 shows the improvement on both GFT [50] and BP4D+ [283] datasets using within-dataset and between-dataset scenarios. Recall that the between-dataset scenario was performed in a way that the classifier was trained on one dataset while being tested on another.

As can be seen in the *within-dataset* scenario of Fig. 4.10, the improvements on GFT focus on the minor classes, such as AUs 1, 2, 15, 17 and 24. More precisely, the improvements are mostly obvious in the F1-score metric. As mentioned earlier, this is because F1-score maintains the sensitivity in true positives, and therefore including more samples from the minority classes can help improve detection of the true positives. AUC did not reflect much improvement or decrement because of its insensitivity to skewed class distributions, as also discussed in [85]. On the other hand, the improvements within BP4D+ are rather consistent. One possible explanation is because BP4D+ yields more dramatic skewness between AU distributions than GFT does, and our multi-label MOMU strategy is able to better balance the distribution between and within batches. More interestingly, the improvements on BP4D+ are roughly inverse-proportional to the underlying AU base rates as shown in Fig. 4.6(b). This provides an evidence that training with a more balanced distribution in multi-label data can help improve test time performance, and the improvement is even more obvious when the class distributions are significantly different.

For the *between-dataset* scenario, the improvements of minority classes can be still observed for both datasets. Because BP4D+ has much higher base rate in AUs than BP4D does, AU 4 was significantly improved in the between-GFT experiment for both AUC and F1. For some AUs such as 1, 2, 6, 7, 10 and 12, the improvements were much less obvious. On the other hand, for the between-BP4D+ experiments, the results were rather mixed. For AUs 14, 17, 23 and 24, we

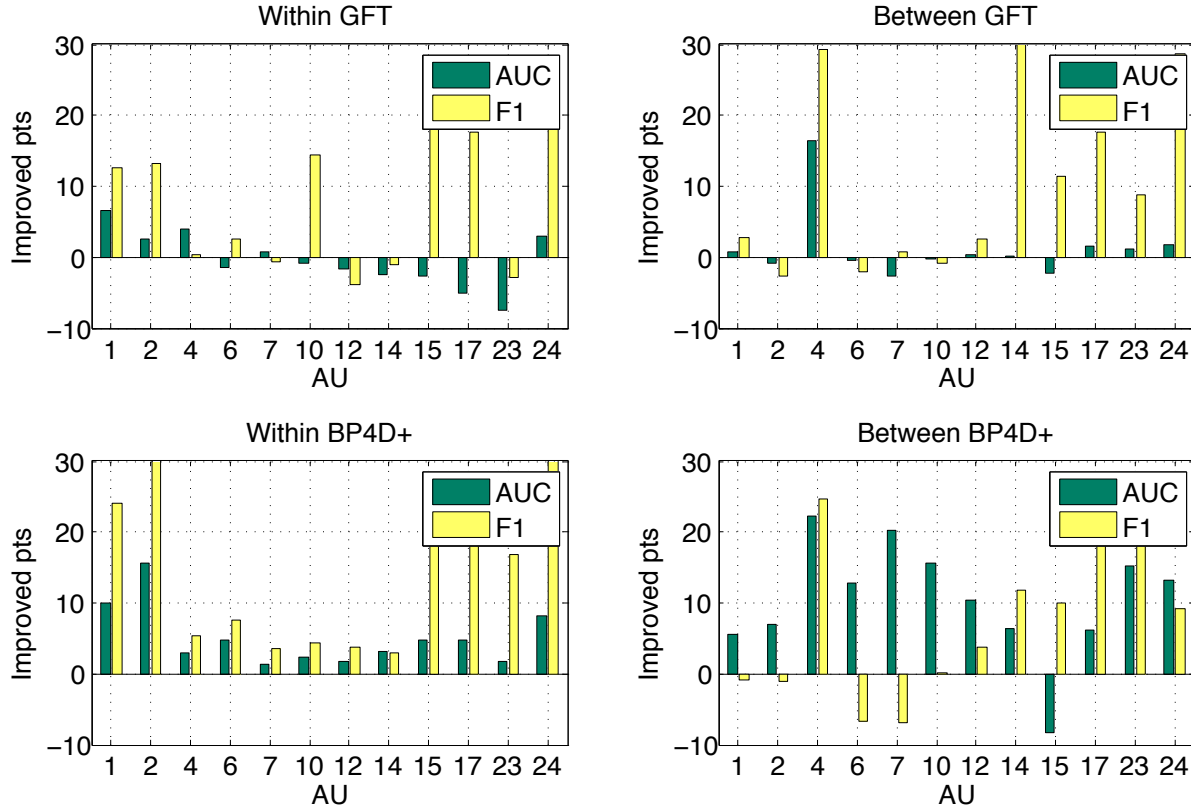


Figure 4.10: Improved points of MOMU over random sampling in both within-dataset and between-dataset scenarios for GFT [50] and BP4D+ [283] datasets. Results in AUC and F1 suggest that improvements are more consistent in BP4D+ than in GFT due to the more dramatic AU imbalance in the BP4D+ dataset (as illustrated in Fig. 4.6).

observed similar behaviors. However, for AUs 1, 2, 6, and 7, AUC was improved yet F1 behaved in the opposite. Similarly, for AUs 10 and 12, the improvements in terms of AUC were higher than the ones in F1-score. One potential reason is because GFT has more subjects and thus more number of frames to train the classifier. Although within each AU the distribution is biased toward negative samples, having more training data can potentially improve prediction on negative samples, and thus improves AUC better. Nevertheless, multiple variabilities between two dataset can account for such relatively unpredictable results. These variabilities include recording environments, interview context, skin color, head pose and so on, as also witnessed in Chapter 3. We believe this is still an open problem, and refer interested readers to the Conclusion Chapter for more of our thoughts and ideas to address these variabilities.

4.4.4 Comparisons among generic, personalized and deep models

Following the previous experimental settings, we evaluated a baseline linear SVM, a personalized STM and deep models using the 10-fold data split protocol with both within-dataset and between-dataset scenarios on GFT [93] and BP4D+ [283] datasets. For more descriptions about these metrics, we refer interested readers to Sec. 2.6 or [93].

Table 4.6: A summary of averaged performance of 12 AUs among alternative models for within-dataset (**top**) and between-dataset (**bottom**) scenarios using GFT [93] and BP4D+ [283] datasets.

	GFT					BP4D+				
	Model	S	AUC	PA	NA	Model	S	AUC	PA	NA
Within-dataset	SVM	.45	.75	.41	.77	SVM	.46	.78	.5	.74
	STM	.59	.79	.45	.82	STM	.5	.78	.52	.74
	DL-rand	.67	.83	.4	.84	DL-rand	.83	.87	.52	.95
	DL-strat	.56	.83	.44	.84	DL-strat	.67	.87	.52	.84
	DL-momu	.51	.84	.51	.75	DL-momu	.67	.92	.67	.82
Between-dataset	Model	S	AUC	PA	NA	Model	S	AUC	PA	NA
	SVM	.01	.59	.29	.52	SVM	.45	.75	.41	.77
	STM	.12	.61	.32	.57	STM	.47	.73	.43	.74
	DL-rand	.33	.65	.25	.64	DL-rand	.45	.75	.25	.76
	DL-strat	.35	.65	.26	.67	DL-strat	.37	.75	.33	.7
	DL-momu	.13	.64	.32	.54	DL-momu	.29	.75	.41	.67

Tables 4.4~4.8 show detailed AU-level results on each method. For both within- and between-dataset scenarios, the improvements of STM over SVM are more obvious in terms of PA than in other metrics. Recall that STM considers covariate shift between training and test distributions, and can thus help reduce domain mismatch caused by individual differences. However, we do notice that the improvement of STM over SVM was not as obvious as the results reported in Chapter 3. Two are possible reasons: (1) In the renewed GFT and BP4D+ datasets, the diversity of ~ 150 training subjects is larger than ≤ 50 subjects used in Chapter 3. Thus, the training distribution is likely to cover most test scenarios, making a generic classifier generalize better. (2) Given the large number of more than 500,000 training samples, STM suffers from scalability due to its quadratic complexity in solving the re-weighting parameters. This makes STM unable to take full advantage of all training samples as a scalable version of SVM [83]. We refer interested reader to the Conclusion Chapter for potential directions to resolve the scalability issue. Overall, we observed an interesting behavior of STM over SVM: The improvement margin is larger in GFT than in BP4D+ for both within- and between-dataset scenarios. This is likely because GFT contains more diverse racial background and large head motions due to spontaneous expressions, while subjects in BP4D+ are mostly frontal to the camera. Such variations can cause the distributional differences more obvious in GFT than in BP4D+ so that STM improves better. Directions to further error-analyze such between-dataset differences are summarized in the Chapter 6.

Table 4.6 summarizes the results averaged over all 12 AUs. As can be seen, the improvements of random sampling (DL-rand) and multi-label stratification (DL-strat) over SVM and STM were noticeable. In the *within-dataset* scenario, there were increased (S, AUC, PA, NA) scores for up to (8, 8, 3, 7) points for GFT, and up to (33, 9, 0, 21) points for BP4D+. Our multi-label MOMU (DL-momu) model achieved consistently the highest (AUC, PA) scores, yielding significantly improvement over STM and SVM of (5, 6) points for GFT and (14, 17) points for BP4D+. Note that

S and AUC are insensitive to skewness (*i.e.*, positive-negative ratio) in each AU, and thus can easily hit high scores with mostly negative predictions. For instance, predicting all samples as negative for AU4 in the GFT dataset gives $S=.93$ and $AUC=.98$. Instead, we believe F-measures on positive prediction (*e.g.*, PA) are better evaluation for AU detection performance. In all, the performance on BP4D+ was better than that on GFT by up to (16, 4, 7, 11) points mainly due to the more controlled recording environment in BP4D+. Interestingly, the improvement gap of DL-momu over other methods was larger in BP4D+ than in GFT, because the AU classes are more imbalanced in BP4D+ (as shown in Fig. 4.6). However, when the performance was boiled down into individual AUs, we observed that both DL-rand and DL-strat exhibited low performance on rare AUs (*e.g.*, AUs 1, 2, 4, 15, 24). This is again due to the AU imbalance during batch sampling, which potentially caused the training process bias toward frequent AUs. As shown in Tables 4.4 and 4.5, the improvements of DL-momu were mainly from such minor AUs. In specific, this can be inspected through the averaged improvements of DL-momu over SVM in terms of (S, AUC, PA, NA) were (9, 12, 14, 1) points for GFT minor classes (*i.e.*, AUs 1, 2, 4, 14, 15, 24 whose base rates are $\leq 12\%$), and (3, 6, 5, -6) points for the remaining major AUs. Similar improvements were (21, 18, 16, 7) points for BP4D+ minor classes (*i.e.*, AUs 1, 2, 15, 17, 24 with base rates $\leq 14\%$), and (21, 11, 14, 9) points for the remaining AUs. This shows that standard sampling approaches would make models inevitably biased toward the major classes.

In the *between-dataset* scenario, the results were relatively mixed. Not surprisingly, the between-dataset experiments resulted in lower performance than the within-dataset ones (~ 20 points lower in AUC and PA). Because STM tackles distribution mismatch while learning a classifier for the target domain, it achieved the top PA scores for both datasets. One reason is that distributional differences were particularly obvious between than within datasets. Our DL models improved almost all metrics in GFT, while only on par or sometimes lower in BP4D+. This implies that the models trained on GFT generalized worse than the ones trained on BP4D+. One explanation is because GFT contains more diverse variations in terms of skin colors, head poses, and partial occlusions. Hence, a model trained on GFT could potentially overfit the dataset and encode unnecessary information to generalize to another domain. In other words, using a simpler model (*e.g.*, SVM or STM) could avoid overfitting and thus generalize better for a complex dataset (as can be observed in BP4D+ between-dataset scenario of Table 4.6). Furthermore, we notice that metrics can often conflict. For instance, in the BP4D+ between-dataset scenario, DL-momu improved DL-rand by 16 points in PA, yet decreased 9 points in NA. We conjectured this is due to inherent difficulties in annotating AUs that are often confusing even for human coders, especially the frames during AU transition (onset \leftrightarrow offset). Ambiguities in AU annotations can likely mix the distributions between positive and negative samples, making an ideal separation rather unclear. This implies that forcing correct classification on positive samples could possibly sacrifice a number of negative ones, resulting in increase PA yet decreased NA. We note that addressing conflict for domain transfer remains an open problem, and point out potential directions in the Chapter 6.

Conclusive remarks

From the comparisons among generic, personalized and deep models, we reach two conclusive remarks in hope to provide insights for future research:

1. Individual differences matter. When a person-specific classifier (*i.e.*, a classifier trained with annotated samples of the test subject) is infeasible and large number of training subjects is

unavailable, we recommend to use a transductive framework (*e.g.*, STM in Chapter 3) rather than generic classifiers due to potential distributional mismatches.

2. When large number of training subjects and good-quality annotations are available, we recommend to use an end-to-end supervised framework with careful treatments in training (*e.g.*, sampling strategies in Sec. 4.4). The reasons that end-to-end deep models are preferred are because of their capabilities in:
 - Learning optimal features for classification (Sec. 4.2.1),
 - Scalable to almost infinite amount of training data thanks to stochastic optimization,
 - Easily extendable to address multiple issues in one framework, *e.g.*, multi-label prediction, or fusion with temporal information (Sec. 4.2.2).

4.5 Summary

We have presented a hybrid network that jointly learns three key factors in AU detection: *Spatial representation*, *temporal modeling*, and *AU correlation*. To the best of our knowledge, this is the first study that shows a possibility for exploring the three seemingly unrelated aspects within one framework. The hybrid network is motivated by existing progress on deep models, and takes advantage of spatial CNNs, temporal LSTMs, and their fusions to achieve multi-label AU detection. In particular, compared to popular hand-crafted features in AU detection, we empirically showed that a spatial representation can be learned, reduces sensitivity to the identity of subjects, and further improves performance even with a linear classifier. Experiments on two of the largest spontaneous AU datasets demonstrate that the proposed network outperformed a standard CNN and feature-based state-of-the-art methods. Furthermore, we utilized an optimization-based visualization to show the learned AU models. This is to our knowledge, for the first time, to see how machines sense facial AUs. Finally, we studied the sampling strategies for multi-label data, which is relatively neglected in the deep learning community. Future work include deeper investigation/analysis of this hybrid network, and incorporation of bi-directional LSTMs. We refer interested readers to Chapter 6 for detailed discussion.



Table 4.7: Performance evaluation of SVM in terms of within-dataset (**top**) and between-dataset (**bottom**) scenarios in the GFT [50] and BP4D+ [283] datasets. For example, the between-dataset experiment on GFT means training on BP4D+ while testing on GFT.

		GFT					BP4D+				
		AU	S	AUC	PA	NA	AU	S	AUC	PA	NA
Within-dataset	1	.42	.71	.24	.79		1	.23	.65	.33	.67
	2	.28	.68	.22	.73		2	.37	.63	.3	.76
	4	.7	.63	.13	.91		4	.56	.69	.17	.86
	6	.52	.85	.66	.75		6	.62	.91	.79	.8
	7	.39	.77	.66	.64		7	.54	.83	.82	.61
	10	.44	.8	.65	.68		10	.64	.92	.85	.74
	12	.6	.88	.72	.78		12	.6	.91	.8	.76
	14	.36	.7	.27	.78		14	.44	.81	.76	.62
	15	.45	.72	.21	.82		15	.38	.71	.25	.78
	17	.32	.72	.44	.73		17	.33	.72	.28	.75
	23	.33	.71	.37	.75		23	.39	.78	.44	.78
	24	.65	.82	.31	.9		24	.47	.8	.25	.78
	avg	.45	.75	.41	.77		avg	.46	.78	.5	.74
Between-dataset	1	-.23	.59	.15	.47		1	.42	.71	.24	.79
	2	.23	.58	.17	.7		2	.28	.68	.22	.73
	4	-.61	.53	.08	.28		4	.7	.63	.13	.91
	6	.12	.64	.48	.55		6	.52	.85	.66	.75
	7	.18	.61	.4	.61		7	.39	.77	.66	.64
	10	.01	.69	.59	.16		10	.44	.8	.65	.68
	12	.29	.73	.59	.55		12	.6	.88	.72	.78
	14	-.4	.49	.17	.37		14	.36	.7	.27	.78
	15	.39	.51	.11	.78		15	.45	.72	.21	.82
	17	-.29	.54	.39	.28		17	.32	.72	.44	.73
	23	.28	.5	.2	.75		23	.33	.71	.37	.75
	24	.16	.63	.16	.69		24	.65	.82	.31	.9
	avg	.01	.59	.29	.52		avg	.45	.75	.41	.77

Table 4.8: Performance evaluation of STM (Chapter 3) in terms of within-dataset (**top**) and between-dataset (**bottom**) scenarios in the GFT [50] and BP4D+ [283] datasets. For example, the between-dataset experiment on GFT means training on BP4D+ while testing on GFT.

	GFT					BP4D+				
	AU	S	AUC	PA	NA	AU	S	AUC	PA	NA
Within-dataset	1	.78	.89	.43	.89	1	.25	.66	.38	.69
	2	.63	.86	.46	.8	2	.28	.62	.31	.71
	4	.64	.59	.12	.89	4	.55	.68	.21	.85
	6	.71	.92	.7	.87	6	.57	.9	.82	.69
	7	.5	.74	.61	.6	7	.62	.85	.86	.68
	10	.66	.9	.62	.86	10	.68	.92	.81	.76
	12	.73	.94	.78	.87	12	.66	.91	.82	.74
	14	.72	.72	.11	.92	14	.47	.83	.75	.58
	15	.62	.72	.23	.87	15	.43	.7	.27	.81
	17	.33	.72	.47	.69	17	.4	.71	.31	.79
	23	.28	.69	.46	.7	23	.51	.8	.44	.81
	24	.51	.85	.45	.83	24	.58	.79	.2	.82
	avg	.59	.79	.45	.82	avg	.5	.78	.52	.74
Between-dataset	AU	S	AUC	PA	NA	AU	S	AUC	PA	NA
	1	-.13	.6	.17	.52	1	.43	.71	.27	.74
	2	.28	.6	.18	.81	2	.24	.7	.25	.7
	4	-.38	.59	.08	.4	4	.65	.67	.2	.68
	6	.1	.58	.5	.49	6	.49	.82	.61	.68
	7	.27	.65	.53	.57	7	.42	.81	.72	.65
	10	.2	.71	.55	.43	10	.45	.73	.62	.67
	12	.33	.75	.61	.65	12	.62	.83	.76	.77
	14	-.38	.51	.2	.35	14	.3	.61	.3	.73
	15	.44	.53	.18	.74	15	.51	.71	.24	.79
	17	.14	.55	.4	.45	17	.4	.7	.46	.75
	23	.28	.55	.3	.71	23	.36	.71	.38	.78
	24	.32	.65	.17	.76	24	.71	.79	.35	.89
	avg	.12	.61	.32	.57	avg	.47	.73	.43	.74

An Unsupervised Framework for Common Event Discovery in Human Interaction

“We build too many walls and not enough bridges.”

Isaac Newton

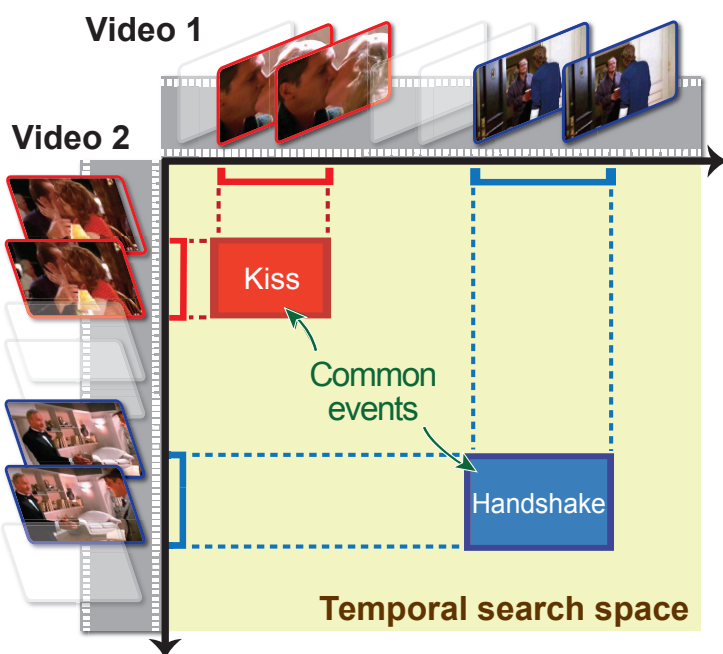


Figure 5.1: An illustration of **Common Event Discovery (CED)** in human interaction: Given two videos, how can one efficiently discover common events in an unsupervised manner? This example illustrates the discovered common events, *Kissing* and *Handshaking*, shared between two videos. Note that the discovered events are of different lengths.



Figure 5.2: Humans are inherently social. Friends, romantic partners, families or coworkers tentatively make the same facial actions when they are engaged. We term these actions as *common events*, and propose an algorithm to discover them without the need of annotations. We note that the last column is not human, but just for illustration.

Previous chapters have shown the capabilities of different learning frameworks that utilize human annotations, either partial supervision in a transductive framework or a full end-to-end supervised framework. Such annotations provide machines the direction to infer a function from labelled training data, which is later used for mapping new, unseen samples. These models trained for automatic analysis of facial actions are powerful, yet encounter their own limitations. First, AU annotations are usually unavailable for unseen domains, *e.g.*, age range, ethnicity, etc. Without information about a target domain, the generalizability is usually limited, as shown in the between-dataset experiment in Chapter 4. Second, collecting AU annotations requires knowledge from domain experts, and thus is time consuming and error-prone. Collecting annotations for 12 AUs for a single 30-minute video can easily consume more than 100 hours of human coders’ time. Finally, supervised models are limited to detection phenomena only described by FACS experts. That is, if there exist facial actions that were not defined or encoded by human coders, a trained machine will never be able to detect or discover such actions.

This chapter attempts to discover facial actions *without* human annotations. The key motivation to our attempt is the observation that human facial actions, or AUs, are usually driven from interactive partners. For example, all spontaneous datasets used in this study were collected either among an interview (*e.g.*, RU-FACS [10], BP4D [282], BP4D+ [283]) or during a particular type of interaction (*e.g.*, social interaction in GFT [50] or parent-infant interaction in Miami dataset [170]). Furthermore, as illustrated in Fig. 5.1, during human interaction, it is often observed a particular temporal window that involves common facial actions from multiple subjects. Given such common events among interactive members of two or more individuals, we propose a general unsupervised framework to efficiently discover these moments of comment events.

5.1 Common Events in Human Interaction

At present, taxonomies of facial actions are based on observer-based schemes, such as FACS. Consequently, approaches to automatic facial expression recognition are dependent on access to corpuses of well-labeled video. Most behavioral analysis methods of computer vision community are supervised in nature, where classification and event detection are common problems. An open question in facial analysis is whether facial actions can be learned directly from video in an unsupervised manner. Most existing studies focus on individuals alone rather than in social context. Exploring

the interaction between individual is critical to understand human social behavior.

In this chapter, we investigate a relatively unexplored problem termed Common Event Discovery (CED), which is able to find patterns that conventional behavioral analysis methods are not capable of. Discovering common patterns in images has been a long standing topic in computer vision, driven by applications of co-segmentation [37, 153, 178], learning grammars of images [293], irregularity detection [15] and automatic tagging [210]. Yet the discovery among time series remains relatively unexplored. Without prior knowledge, CED searches over all possible pairwise temporal segments, and selects the ones that retain maximum visual commonality. Fig. 5.1 illustrates the CED problem with two common events *Kissing* and *Handshake*.

A naive approach to CED would be to use a sliding window. That is, to exhaustively search all possible pairs of temporal segments and select pairs that have the highest similarities. Because the complexity of sliding window methods is quartic with the length of video, *i.e.*, $\mathcal{O}(m^2n^2)$ for two videos of lengths m and n , this cost would be computationally prohibitive in practice. Even in relatively short videos of 200 and 300 frames, there would be in excess of *three billion* possible matches to evaluate at different lengths and locations.

To meet the computational challenge, we propose to extend the Branch-and-Bound (B&B) method for CED. For supervised learning, B&B has proven an efficient technique to detect image patches [142] and video volumes [274]. Because previous bounding functions of B&B are designed for supervised detection or classification, which require pre-trained models, previous B&B methods could not be directly applied to CED. For this reason, we derive novel bounding functions for various commonality measures, including ℓ_1/ℓ_2 distance, intersection kernel, χ^2 distance, cosine similarity, symmeterized cross entropy, and symmeterized KL-divergence.

For evaluation, we apply the proposed B&B to application of discovering events at the same or different times (synchrony and event commonality, respectively), and variable-length segment-based event detection. We conduct the experiments on three datasets of increasing complexity: Posed motion capture and unposed, spontaneous video of mothers and their infants and of young adults in small groups. We report distance and similarity metrics and compare discovery with expert annotations. Our main contributions are:

1. **New CED problem:** While there exist studies that address supervised commonality discovery in images [37, 153, 178, 255], to the our best knowledge, this study is the first to tackle unsupervised discovery of common events in time series. The proposed CED can discover patterns that conventional supervised behavioral analysis methods are not capable of.
2. **New B&B framework:** The proposed B&B framework is entirely general. It takes from two or more videos any signals that can be quantified into histograms. We derive bounding functions for various commonality measures, and provide extensions including multiple commonalities discovery, and accelerated search using a warm-start strategy and parallelism. We show that a slight modification of the framework can be readily applied to discover events happening at different time (event commonality) or around the same time (synchrony), video search, and supervised event detection.

5.2 A Branch-and-Bound Framework for Common Event Discovery

This section describes our representation of time series, a formulation of CED, the proposed B&B framework, and the newly derived bounding functions that fit into the B&B framework.

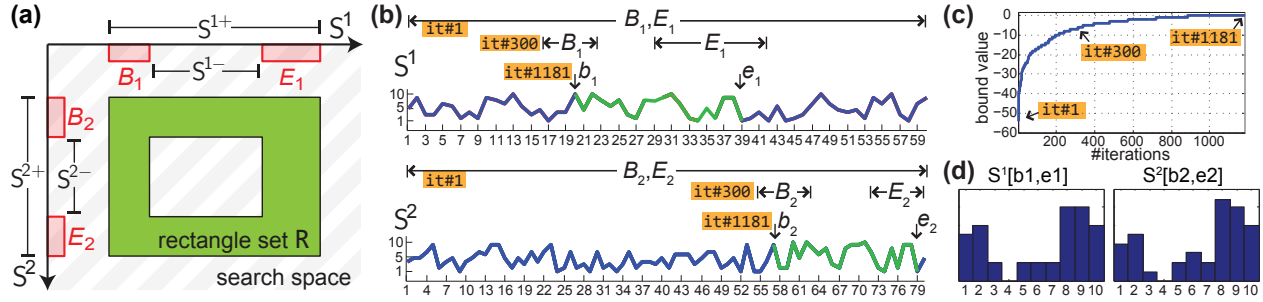


Figure 5.3: An example of CED on two 1-D time series: (a) An illustration of our notation (see Sec. 5.2.3). (b) Searching intervals at iterations (*it*) #1, #300 and #1181 over sequences S^1 and S^2 . Commonalities $S^1[b_1, e_1]$ and $S^2[b_2, e_2]$ are discovered at convergence (#1181). (c) Convergence curve w.r.t. bounding value and #*it*. (d) Histograms of the discovered commonalities. In this example, a naive sliding window approach needs more than 5 million evaluations, while the proposed B&B method converges at iteration 1181 using $\ell = 20$.

5.2.1 Representation of time series

Bag of Temporal Words (BoTW) model [44, 225, 274] has been shown effective in many video analysis problems, such as action recognition [18, 110, 144, 154, 202]. This section modifies the BoTW model to describe the static and dynamic information of a time series. Suppose a time series S can be described as a set of feature vectors $\{x_j\}$ for each frame j . For instance, a feature vector can be facial shape in face videos or joint angles in motion capture videos. Given such features, we extract two types of information: *observation info* from a single frame, and *interaction info* from two consecutive frames. Denote $S[b, e] = \{x_j\}_{j=b}^e$ as a temporal segment between the b -th and the e -th frames, we consider a segment-level feature mapping:

$$\varphi_{S[b,e]} = \sum_{j=b}^e \begin{bmatrix} \phi^{\text{obs}}(x_j) \\ \phi^{\text{int}}(x_j) \end{bmatrix}. \quad (5.1)$$

The observation info $\phi^{\text{obs}}(x_j)$ describes the pseudo-probability of x_j belonging to a latent state, and the interaction info $\phi^{\text{int}}(x_j)$ describes transition probability of states between two consecutive frames. To obtain $\phi^{\text{obs}}(x_j)$, we performed k -means to find K centroids $\{c_k\}_{k=1}^K$ as the hidden states. Then, we computed $\phi^{\text{obs}}(x_j) \in [0, 1]^K$ with the k -th element computed as $\exp(-\gamma \|x_j - c_k\|^2)$ and γ chosen as an inverse of the median distance of all samples to the centroids. An interaction info $\phi^{\text{int}}(x_j) \in [0, 1]^{K^2}$ is computed as:

$$\phi^{\text{int}}(x_j) = \text{vec}(\phi^{\text{obs}}(x_j) \otimes \phi^{\text{obs}}(x_{j+1})), \quad (5.2)$$

where \otimes denotes a Kronecker product of two observation vectors. As a result, each temporal segment is represented as an ℓ_2 -normalized feature vector of dimension $(K^2 + K)$.

Because this representation accepts almost arbitrary features, any signal, even with negative values, that can be quantified into histograms can be directly applied. One notable benefit of the histogram representation is that it allows for fast recursive computation using the concept of *integral image* [252]. That is, the segment-level representation for $S[b, e]$ can be computed as $\varphi_{S[b,e]} =$

$\varphi_{\mathbf{S}[1,e]} - \varphi_{\mathbf{S}[1,b-1]}$, which only costs $\mathcal{O}(1)$ per evaluation. Based on the time series representation, we develop our approach below.

5.2.2 Problem formulation

To establish notion, we begin with two time series \mathbf{S}^1 and \mathbf{S}^2 with m and n frames respectively. The goal of common event discovery (CED) is to find two temporal segments with intervals $[b_1, e_1] \subseteq [1, m]$ and $[b_2, e_2] \subseteq [1, n]$ such that their visual commonality is maximally preserved. We formulate CED:

$$\boxed{\text{CED}} \quad \begin{aligned} & \max_{\{b_1, e_1, b_2, e_2\}} f(\varphi_{\mathbf{S}^1[b_1, e_1]}, \varphi_{\mathbf{S}^2[b_2, e_2]}), \\ & \text{subject to } \ell \leq e_i - b_i, \forall i \in \{1, 2\}, \end{aligned} \quad (5.3)$$

where $f(\cdot, \cdot)$ is a commonality measure between two time series representations, and ℓ controls the minimal length for each temporal segment to avoid a trivial solution. More details about $f(\cdot, \cdot)$ are discussed in Sec. 5.3. Problem (5.3) is non-convex and non-differentiable, and thus standard convex optimization methods can not be directly applied. A naive solution is an exhaustive search over all possible locations for $\{b_1, e_1, b_2, e_2\}$. However, it leads to an algorithm with computational complexity $\mathcal{O}(m^2n^2)$, which is prohibitive for regular videos with hundreds or thousands of frames. To address this issue, we introduce a branch-and-bound (B&B) framework to efficiently and globally solve (5.3).

Note that, although ℓ controls the minimal length of discovered temporal segments, the optimal solution can be of length greater than ℓ . For instance, consider two 1-D time series $\mathbf{S}^1 = [1, 2, 2, 1]$ and $\mathbf{S}^2 = [1, 1, 3]$. Suppose we measure $f(\cdot, \cdot)$ by ℓ_1 distance, where smaller values indicate higher commonality. Let the minimal length $\ell = 3$, and represent their 3-bin histograms as $\varphi_{\mathbf{S}^1[1,4]} = [2, 2, 0]$, $\varphi_{\mathbf{S}^1[1,3]} = [1, 2, 0]$ and $\varphi_{\mathbf{S}^2} = [2, 0, 1]$. Showing the distance $f_{\ell_1}(\varphi_{\mathbf{S}^1[1,4]}, \varphi_{\mathbf{S}^2}) = 3 < 4 = f_{\ell_1}(\varphi_{\mathbf{S}^1[1,3]}, \varphi_{\mathbf{S}^2})$, we prove by contradiction.

5.2.3 Optimization by Branch-and-Bound (B&B)

With a proper bounding function, B&B has been shown empirically more efficient than straight enumeration. B&B can eliminate regions that provably do not contain an optimal solution. This can be witnessed in many computer vision problems, *e.g.*, object detection [142, 145], video search [274], pose estimation [229] and optimal landmark detection [3]. Inspired by previous success, this section describes the proposed B&B framework that globally solves (5.3).

Problem interpretation: As depicted in Fig. 5.1, we interpret Problem (5.3) as searching a rectangle in the 2-D space formed by two time series. A rectangle $\mathbf{r} \doteq [b_1, e_1, b_2, e_2]$ in the search space indicates one candidate solution corresponding to $\mathbf{S}^1[b_1, e_1]$ and $\mathbf{S}^2[b_2, e_2]$. To allow a more efficient representation for searching, we parameterize each step as searching over sets of candidate solutions. That is, we search over *intervals* instead of individual value for each parameter. Each parameter interval corresponds to a rectangle set $\mathbf{R} \doteq B_1 \times E_1 \times B_2 \times E_2$ in the search space, where $B_i = [b_i^{lo}, b_i^{hi}]$ and $E_i = [e_i^{lo}, e_i^{hi}]$ ($i \in \{1, 2\}$) indicate tuples of parameters ranging from frame lo to frame hi . Given the rectangle set \mathbf{R} , we denote the longest and the shortest possible segments as \mathbf{S}^{i+} and \mathbf{S}^{i-} respectively. We denote $|\mathbf{R}|$ as the number of rectangles in \mathbf{R} . Fig. 5.3(a) shows an illustration of the notation.

Algorithm 5: Common Event Discovery (CED)

Input : Collection of frame-based features for sequences $\mathbf{S}^1, \mathbf{S}^2$; minimal length ℓ

Output: The optimal rectangle \mathbf{r}^* in the search space

```

1  $Q \leftarrow$  empty priority queue;           // Initialize Q
2  $R \leftarrow [1, m] \times [1, m] \times [1, n] \times [1, n]$ ; // Initialize R
3  $\mathbf{r}^* \leftarrow \text{BnB}(Q, R)$ ; // Obtain the optimal  $\mathbf{r}$  using BnB
4 return  $\mathbf{r}^*$ ;
1 Procedure  $\text{BnB}(Q, R)$ 
2   while  $|R| \neq 1$  do
3      $R \rightarrow R' \cup R''$ ;           // Branch step
4      $Q.\text{push}(\text{bound}(R'), R')$ ; // Push  $R_1$  and bound
5      $Q.\text{push}(\text{bound}(R''), R'')$ ; // Push  $R_2$  and bound
6      $R \leftarrow Q.\text{pop}()$ ;       // Pop top state from Q
7   return  $R$ ;
```

The B&B framework: With the problem interpreted above, we describe here the proposed B&B framework. Algorithm 5 summarizes the procedure. To maintain the search process, we employ a priority queue denoted as Q . Each state in Q contains a rectangle set R , its upper bound $u(R)$ and lower bound $l(R)$. Each iteration starts by selecting a rectangle set R from the top state, which is defined as the state containing the minimal upper bound for $f(\cdot, \cdot)$. Given this structure, the algorithm repeats a *branch* step and a *bound* step until R contains a unique entry.

In the *branch* step, each rectangle set R is split by its largest interval into two disjoint subsets. For example, suppose E_2 is the largest interval, then $R \rightarrow R' \cup R''$ where $E'_2 = [e_2^{lo}, \lfloor \frac{e_2^{lo} + e_2^{hi}}{2} \rfloor]$ and $E''_2 = [\lfloor \frac{e_2^{lo} + e_2^{hi}}{2} \rfloor + 1, e_2^{hi}]$. In the *bound* step, we calculate the bounds for each rectangle set, and then update new rectangle sets and their bounds into Q . The computed bounds tell the worst possible values in $f(\cdot, \cdot)$, and therefore enable the algorithm to efficiently discard unlikely rectangle sets where their bounds are worse than the current best. The algorithm terminates when R contains a unique entry, i.e., $|R| = 1$. Fig. 5.3(b)-(d) show an example of CED for discovering commonality between two 1-D time series. Despite that in the worst case the complexity of B&B can be still $\mathcal{O}(m^2n^2)$, we will experimentally show that in general B&B is much more efficient than naive approaches.

5.3 Construction of bounding functions

One crucial aspect of the proposed B&B framework is the novel bounding functions for measuring commonality between two time series. The commonality measures can interchangeably be formed in terms of distance or similarity functions. Below we describe the conditions of bounding functions, and then construct the bounds.

Conditions of bounding functions: Recall that R represents a rectangle set and $\mathbf{r} \doteq [b_i, e_i, b_j, e_j]$ represents a rectangle corresponding to two subsequences $\mathbf{S}^i[b_i, e_i]$ and $\mathbf{S}^j[b_j, e_j]$. Without loss of generality, we denote $f(\mathbf{r}) = f(\varphi_{\mathbf{S}^i[b_i, e_i]}, \varphi_{\mathbf{S}^j[b_j, e_j]})$ as the commonality measure between $\mathbf{S}^i[b_i, e_i]$

and $\mathbf{S}^j[b_j, e_j]$. To harness the B&B framework, we need to find an upper bound $u(\mathbf{R})$ and a lower bound $l(\mathbf{R})$ that bounds the values of f over a set of rectangles. A proper bounding function has to satisfy the conditions:

$$\begin{aligned} a) \quad & u(\mathbf{R}) \geq \max_{\mathbf{r} \in \mathbf{R}} f(\mathbf{r}), \\ b) \quad & l(\mathbf{R}) \leq \min_{\mathbf{r} \in \mathbf{R}} f(\mathbf{r}), \\ c) \quad & u(\mathbf{R}) = f(\mathbf{r}) = l(\mathbf{R}), \text{ if } \mathbf{r} \text{ is the only element in } \mathbf{R}. \end{aligned} \quad \boxed{\text{Bounding conditions}}$$

Conditions *a*) and *b*) ensure that $u(\mathbf{R})$ and $l(\mathbf{R})$ appropriately bound all candidate solutions in \mathbf{R} from above and from below, whereas *c*) guarantees the algorithm to converge to the optimal solution. With both lower and upper bounds, one can further prune the priority queue for speeding the search, *i.e.*, eliminate rectangle sets \mathbf{R}' that satisfy $l(\mathbf{R}') > u(\mathbf{R})$ [8].

Bound histogram bins: Let \mathbf{S}^i denote the i -th time series and can be represented as an unnormalized histogram \mathbf{h}^i or a normalized histogram $\hat{\mathbf{h}}^i$ using the representation in Sec. 5.2.1. Denote h_k^i and \hat{h}_k^i as the k -th bin of \mathbf{h}^i and $\hat{\mathbf{h}}^i$, respectively. The normalized histogram is defined as $\hat{h}_k^i = h_k^i / |\mathbf{S}^i|$, where $|\mathbf{S}^i| = \sum_k h_k^i$. $\|\mathbf{S}^i\| = \sqrt{\sum_k (h_k^i)^2}$ is the Euclidean norm of histogram of \mathbf{S}^i . Considering histograms of \mathbf{S}^{i+} and \mathbf{S}^{i-} , we can bound their k -th histogram bin:

$$0 \leq h_k^{i-} \leq h_k^i \leq h_k^{i+}, \forall i. \quad (5.4)$$

Given a rectangle $\mathbf{r} = [b_1, e_1, b_2, e_2]$ and denote $\underline{h}_k^i = \frac{h_k^{i-}}{|\mathbf{S}^{i+}|}$ and $\overline{h}_k^i = \frac{h_k^{i+}}{|\mathbf{S}^{i-}|}$. For *normalized* histograms, we use the fact that $|\mathbf{S}^{i-}| \leq |\mathbf{S}^i[b_i, e_i]| \leq |\mathbf{S}^{i+}|$. Then we can rewrite (5.4) for bounding the normalized bins:

$$0 \leq \underline{h}_k^i \leq \hat{h}_k^i \leq \overline{h}_k^i, \forall i. \quad (5.5)$$

Given these bounds for individual histogram bins, below we construct bounds for various commonality measures with normalized histograms, whereas those with unnormalized histograms can be likewise obtained.

Bound commonality measures: Given two time series \mathbf{S}^i and \mathbf{S}^j represented as normalized histograms $\hat{\mathbf{h}}^i$ and $\hat{\mathbf{h}}^j$ respectively, we provide bounding functions for various commonality measures: ℓ_1/ℓ_2 distance, histogram intersection, χ^2 distance, cosine similarity, symmetrized KL divergence, and symmetrized cross entropy. These measures have been widely applied to many tasks such as objection recognition [80, 142] and action recognition [18, 110, 144, 154, 202].

1) ℓ_1/ℓ_2 distance: Applying the min/max operators on (5.4), we get

$$\begin{aligned} \min(h_k^{i-}, h_k^{j-}) &\leq \min(h_k^i, h_k^j) \leq \min(h_k^{i+}, h_k^{j+}), \\ \text{and } \max(h_k^{i-}, h_k^{j-}) &\leq \max(h_k^i, h_k^j) \leq \max(h_k^{i+}, h_k^{j+}). \end{aligned} \quad (5.6)$$

Reordering the inequalities, we obtain the upper bound u_k and lower bound l_k for the k -th histogram bin:

$$\begin{aligned} l_k &= \max(h_k^{i-}, h_k^{j-}) - \min(h_k^{i+}, h_k^{j+}) \\ &\leq \max(h_k^i, h_k^j) - \min(h_k^i, h_k^j) = |h_k^i - h_k^j| \\ &\leq \max(h_k^{i+}, h_k^{j+}) - \min(h_k^{i-}, h_k^{j-}) = u_k. \end{aligned} \quad (5.7)$$

Summing over all histogram bins, we obtain the bounds of the ℓ_1 distance for two unnormalized histograms $\mathbf{h}^i, \mathbf{h}^j$:

$$\sum_k l_k \leq \sum_k |h_k^i - h_k^j| = f_{\ell_1}(\mathbf{h}^i, \mathbf{h}^j) \leq \sum_k u_k. \quad (5.8)$$

For normalized histograms $\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j$, we obtain their bounds following same operations of (5.6) and (5.7):

$$l_{\ell_1}(\mathbf{R}) = \sum_k \hat{l}_k \leq f_{\ell_1}(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) \leq \sum_k \hat{u}_k = u_{\ell_1}(\mathbf{R}), \quad (5.9)$$

where

$$\begin{aligned} \hat{l}_k &= \max(\underline{h}_k^i, \underline{h}_k^j) - \min(\overline{h}_k^i, \overline{h}_k^j), \\ \text{and } \hat{u}_k &= \max(\overline{h}_k^i, \overline{h}_k^j) - \min(\underline{h}_k^i, \underline{h}_k^j). \end{aligned} \quad (5.10)$$

Deriving bounds for ℓ_2 -distance can be written as:

$$\begin{aligned} l_{\ell_2}(\mathbf{R}) &= \sum_k (\hat{l}_k)_+^2 \leq \sum_k (\hat{h}_k^i - \hat{h}_k^j)^2 \\ &= f_{\ell_2}(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) \leq \sum_k \hat{u}_k^2 = u_{\ell_2}(\mathbf{R}), \end{aligned} \quad (5.11)$$

where $(\cdot)_+ = \max(0, \cdot)$ is a non-negative operator.

2) Histogram intersection: Given two normalized histograms, we define their intersection distance by the Hilbert space representation [212]:

$$f_{\cap}(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) = - \sum_k \min(\hat{h}_k^i, \hat{h}_k^j). \quad (5.12)$$

Following (5.5) and (5.6), we obtain its lower bound and upper bound:

$$\begin{aligned} l_{\cap}(\mathbf{R}) &= - \sum_k \min(\overline{h}_k^i, \overline{h}_k^j) \\ &\leq f_{\cap}(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) \leq - \sum_k \min(\underline{h}_k^i, \underline{h}_k^j) = u_{\cap}(\mathbf{R}). \end{aligned} \quad (5.13)$$

3) χ^2 distance: The χ^2 distance has been proven to be effective to measure distance between histograms. The χ^2 distance is defined as:

$$f_{\chi^2}(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) = \sum_k \frac{(\hat{h}_k^i - \hat{h}_k^j)^2}{\hat{h}_k^i + \hat{h}_k^j}. \quad (5.14)$$

Incorporating the ℓ_1 -bounds \widehat{l}_k and \widehat{u}_k in (5.10) and the inequalities in (5.5), we obtain the lower bound and upper bound for f_{χ^2} as:

$$l_{\chi^2}(\mathbf{R}) = \sum_k \frac{(\widehat{l}_k)_+^2}{\widehat{h}_k^i + \widehat{h}_k^j}, \quad (5.15)$$

$$\text{and } u_{\chi^2}(\mathbf{R}) = \sum_k \frac{\widehat{u}_k^2}{\widehat{h}_k^i + \widehat{h}_k^j}. \quad (5.16)$$

4) Cosine similarity: Treating two normalized histograms $\widehat{\mathbf{h}}^i$ and $\widehat{\mathbf{h}}^j$ as two vectors in the inner product space, we can measure the similarity as their included cosine angle:

$$\begin{aligned} f_C(\widehat{\mathbf{h}}^i, \widehat{\mathbf{h}}^j) &= \frac{\widehat{\mathbf{h}}^i \cdot \widehat{\mathbf{h}}^j}{\|\widehat{\mathbf{h}}^i\| \|\widehat{\mathbf{h}}^j\|} = \frac{\sum_k \frac{h_k^i h_k^j}{|\mathbf{S}^i| |\mathbf{S}^j|}}{\sqrt{\sum_k (\frac{h_k^i}{|\mathbf{S}^i|})^2} \sqrt{\sum_k (\frac{h_k^j}{|\mathbf{S}^j|})^2}} \\ &= \frac{\sum_k h_k^i h_k^j}{\sqrt{\sum_k (h_k^i)^2} \sqrt{\sum_k (h_k^j)^2}} = \frac{\mathbf{h}^i \cdot \mathbf{h}^j}{\|\mathbf{h}^i\| \|\mathbf{h}^j\|}. \end{aligned} \quad (5.17)$$

Using (5.4) and the fact that $\|\mathbf{S}^{i-}\| \leq \|\mathbf{S}^i[b_i, e_i]\| \leq \|\mathbf{S}^{i+}\|$, we obtain the bounds:

$$l_C(\mathbf{R}) = \frac{\sum_k h_k^{i-} h_k^{j-}}{\|\mathbf{S}^{i+}\| \|\mathbf{S}^{j+}\|} \leq f_C(\mathbf{h}^i, \mathbf{h}^j) \leq \frac{\sum_k h_k^{i+} h_k^{j+}}{\|\mathbf{S}^{i-}\| \|\mathbf{S}^{j-}\|} = u_C(\mathbf{R}).$$

5) Symmetrized KL divergence: By definition, the normalized histograms $\widehat{\mathbf{h}}^i$ and $\widehat{\mathbf{h}}^j$ are non-negative and sum to one, and thus can be interpreted as two discrete probability distributions. Their similarity can be measured using the symmetrized KL divergence:

$$\begin{aligned} f_D(\widehat{\mathbf{h}}^i, \widehat{\mathbf{h}}^j) &= D_{KL}(\widehat{\mathbf{h}}^i \|\widehat{\mathbf{h}}^j) + D_{KL}(\widehat{\mathbf{h}}^j \|\widehat{\mathbf{h}}^i) \\ &= \sum_k (\widehat{h}_k^i - \widehat{h}_k^j) (\ln \widehat{h}_k^i - \ln \widehat{h}_k^j), \end{aligned} \quad (5.18)$$

where $D_{KL}(\widehat{\mathbf{h}}^i \|\widehat{\mathbf{h}}^j)$ is the KL divergence of $\widehat{\mathbf{h}}^j$ from $\widehat{\mathbf{h}}^i$. From (5.5) and that $\underline{h}_k^i - \overline{h}_k^j \leq \widehat{h}_k^i - \widehat{h}_k^j \leq \overline{h}_k^i - \underline{h}_k^j$, we have $\ln \underline{h}_k^i - \ln \overline{h}_k^j \leq \ln \widehat{h}_k^i - \ln \widehat{h}_k^j \leq \ln \overline{h}_k^i - \ln \underline{h}_k^j$. Then, we obtain the bounds for (5.18):

$$\begin{aligned} l_D(\mathbf{R}) &= \sum_k (\underline{h}_k^i - \overline{h}_k^j)_+ (\ln \underline{h}_k^i - \ln \overline{h}_k^j)_+ \\ &\leq f_D(\widehat{\mathbf{h}}^i, \widehat{\mathbf{h}}^j) \leq \sum_k (\overline{h}_k^i - \underline{h}_k^j) (\ln \overline{h}_k^i - \ln \underline{h}_k^j) = u_D(\mathbf{R}). \end{aligned} \quad (5.19)$$

6) Symmetrized cross entropy: The symmetrized cross entropy [179] measures the average number of bins needed to identify an event by treating each other as the true distribution. Similar to

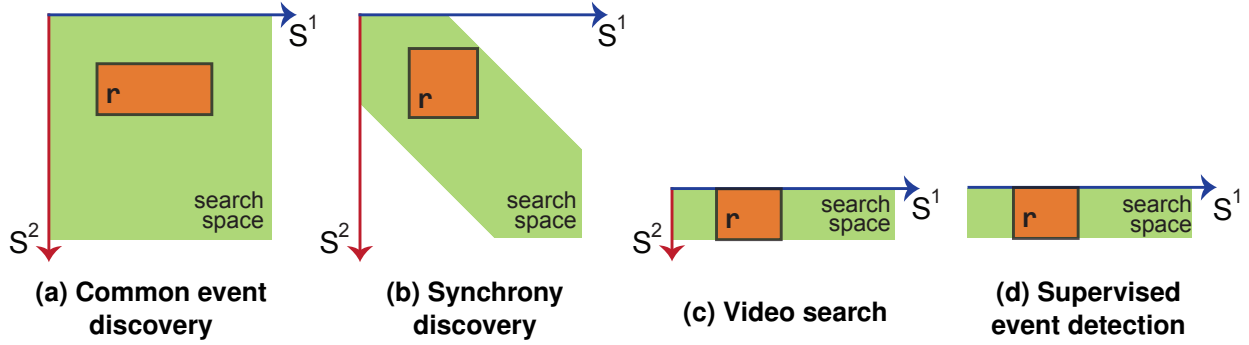


Figure 5.4: Searching scenarios readily applicable to the proposed B&B framework: (a) Common event discovery (CED), (b) synchrony discovery (SD), (c) video search (VS), and (d) supervised segment-based event detection (ED). Green area indicates the search space; an orange box indicates a candidate solution r . (see Sec. 5.4 for details)

KL divergence that treats $\hat{\mathbf{h}}^i$ and $\hat{\mathbf{h}}^j$ as two discrete probability distributions, the entropy function is written as:

$$f_E(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) = \sum_k \hat{h}_k^i \log \frac{1}{\hat{h}_k^j} + \sum_k \hat{h}_k^j \log \frac{1}{\hat{h}_k^i}. \quad (5.20)$$

Recall (5.5) and that $0 \leq \hat{h}_b^i \leq 1, 0 \leq \hat{h}_b^j \leq 1$, we obtain the bounds:

$$\begin{aligned} l_E(\mathbf{R}) &= \sum_b \left(-\underline{h}_k^i \log \overline{h}_k^j - \underline{h}_k^j \log \overline{h}_k^i \right) \\ &\leq f_E(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) \leq \sum_k \left(-\overline{h}_k^i \log \underline{h}_k^j - \overline{h}_k^j \log \underline{h}_k^i \right) = u_E(\mathbf{R}). \end{aligned} \quad (5.21)$$

We refer interested readers to the Appendix C for detailed derivation of the above bounds. Above we have reported derivations for six commonly used measures. However, choice of one or another is influenced by a variety of factors, such as the nature of the data, problem, preferences of individual investigators, etc. In experiments, we picked ℓ_1 , χ^2 , and KL-divergence because due to their popularity in computer vision applications. For instance, ℓ_1 -distance is popular in retrieval problems (e.g., [108, 198]), χ^2 -distance in object recognition (e.g., [80, 142]), and KL-divergence in measuring similarity between distributions (e.g., Gaussian mixtures for image segmentation [96]).

5.4 Searching Scenarios

With the B&B framework and various bounds derived in the previous section, this section discusses unsupervised and supervised searching scenarios that can be readily applied. Fig. 5.4 illustrates the searching scenarios in terms of different applications. The first application, Common Event Discovery (CED), as has been discussed in Sec. 5.6.1, has the most general form and the broadest search space. Below we discuss others in turn.

Algorithm 6: Synchrony Discovery (SD)

Input : A synchronized video pair $\mathbf{S}^1, \mathbf{S}^2$; minimal discovery length ℓ ; commonality period T
Output: Optimal intervals $\mathbf{r}^* = [b_1, e_1, b_2, e_2]$

- 1 $L \leftarrow T + \ell$; // The largest possible searching period
- 2 $Q \leftarrow$ empty priority queue; // Initialize Q
- 3 **for** $t \leftarrow 1$ **to** $(n - T - L + 1)$ **do**
- 4 $R \leftarrow [t, t + T] \times [t + \ell - 1, t + T + L - 1] \times [t - T, t + T] \times [t - T + \ell - 1, t + T + L - 1]$;
- 5 $Q.\text{push}(\text{bound}(R), R)$;
- 6 $\mathbf{r}^* \leftarrow \text{BnB}(Q, R)$; // BnB procedure in Algo. 5
- 7 **return** \mathbf{r}^* ;

5.4.1 Synchrony discovery (SD)

Social interaction plays an important and natural role in human behavior. In this section, we present that a slight modification of CED can result in a solution to discover *interpersonal synchrony*, which is referred as to two or more persons preforming common actions in overlapping video frames or segments. Fig. 5.4(b) illustrates the idea. Specifically, synchrony discovery searches for commonalities (or matched states) among two synchronized videos \mathbf{S}^1 and \mathbf{S}^2 with n frames each. Rewriting (5.3), we formulate SD as:

$$\begin{aligned}
 \boxed{\text{SD}} \quad & \max_{\{b_1, e_1, b_2, e_2\}} f(\varphi_{\mathbf{S}^1[b_1, e_1]}, \varphi_{\mathbf{S}^2[b_2, e_2]}), \\
 \text{subject to} \quad & \ell \leq e_i - b_i, \forall i \in \{1, 2\}, |b_1 - b_2| \leq T,
 \end{aligned} \tag{5.22}$$

where $f(\cdot, \cdot)$ is the commonality measure, and T is a *temporal offset* that allows SD to discover commonalities within a T -frame temporal window, *e.g.*, in mother-infant interaction, the infant could start smiling after the mother leads the smile for a few seconds. A naive solution has complexity $\mathcal{O}(n^4)$.

Algorithm: For an event to be considered as a synchrony, they have to occur within a temporal neighborhood between two videos. For this reason, we only need to search within neighboring regions in the temporal search space. Unlike CED or ESS [142] that exhaustively prunes the search space to a unique solution, we constrain the space before the search begins. In specific, we slightly modify Algorithm 5 to solve SD. Let $L = T + \ell$ be the largest possible period to search, we initialize a priority queue Q with rectangle sets $\{[t, t + T] \times [t + \ell - 1, t + T + L - 1] \times [t - T, t + T] \times [t - T + \ell - 1, t + T + L - 1]\}_{t=1}^{n-T-L+1}$ and their associated bounds (see details in Sec. 5.3). These rectangle sets lie sparsely along the diagonal in the 2-D search space, and thus prune a large portion during the search. Once all rectangle sets are settled, the CED algorithm can be employed find the exact optimum. Algorithm 6 summarizes the SD algorithm.

Fig. 5.5 shows a synthetic example of 1-D time series with two synchronies, denoted as red dots and green triangle, where one is a random permutation of another. USD discovered 3 dyads with the convergence curve in (b), and histograms of each dyad in (c)~(e). Note that the interaction feature distinguishes the temporal consistency for the first and second discovery, maintaining a much smaller distance than the third discovery.

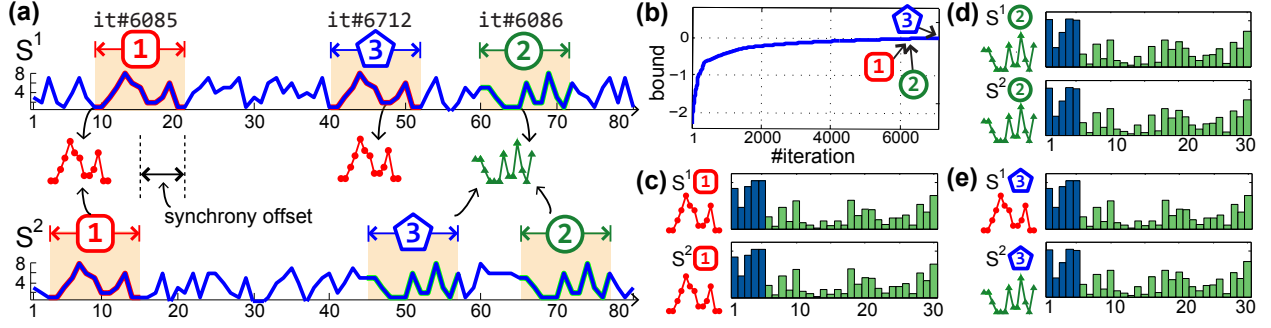


Figure 5.5: An example of SD on two 1-D time series using $\ell = 13$ and $T = 5$: (a) Top 3 discovered synchronies at different iterations; exhaustive search takes 39151 iterations. (b) The convergence curve w.r.t. bounding value and #iter. (c)~(e) Discovered synchronies and their histograms, where blue and green bars indicate the segment features ϕ^{obs} and ϕ^{int} , respectively. ϕ^{int} is 10X magnified for display purpose. The ℓ_1 distances between the three histogram pairs are $6.3e-8$, $1.5e-7$, and $5.8e-2$, respectively.

5.4.2 Video search (VS)

The CED algorithm can be also useful for efficient searching for a time series with similar content. That is, given a (relatively short) query time series, search for common temporal segments in a longer video in an efficient manner. Fig. 5.4(c) illustrates the idea. More formally, let Q be the query time series, we find in the target time series S by modifying (5.3) as:

$$\boxed{\text{VS}} \quad \max_{b,e} f(\varphi_{S[b,e]}, \varphi_Q), \quad (5.23)$$

subject to $\ell \leq e - b$.

The problem now becomes searching along one axis of the search space, but it is still non-convex and non-differentiable. Nevertheless, Algorithm 5 can be directly applied to find the optimal solution by fixing the beginning and ending frame of the query time series. Note that we do not claim that VS is state-of-the-art method for video search, but just illustrate the versatility of the B&B framework.

5.4.3 Segment-based event detection (ED)

Efficiently detecting events in time series arises in a wide spectrum of applications, ranging from diseases, financial decline, speech recognition to video security. While event detection has been studied extensively in the literature, little attention has been paid to efficient inference from a pre-trained classifier. Fig. 5.4(d) illustrates the idea. Here we demonstrate event detection using an SVM decision function, which has been shown effective in many event detection tasks [112, 144, 202, 213].

Given the BoTW representation discussed in Sec. 5.2.1, we represent time series by their histograms. These histograms are used to train an SVM classifier to tell whether a new time series contains an event of interest. To perform inference, temporal segmentation [144, 202, 213] or dynamic programming (DP) [112] is required. However, temporal segmentation for many real-world videos may not be trivial, and DP is computationally expensive to run it in large scale, especially

Algorithm 7: Video Search (VS)

Input : A query \mathbf{Q} with length ℓ ; a target series \mathbf{S} with length n ; a similarity threshold ϵ

Output: Detected events $\{\mathbf{S}[b_i, e_i]\}_i$

```

1  $\mathbf{Q} \leftarrow$  empty priority queue; // Initialize  $\mathbf{Q}$ 
2  $\mathbf{R} \leftarrow [1, n] \times [1, n] \times [1, 1] \times [\ell, \ell]$ ; // Initialize  $\mathbf{R}$ 
3 while true do
4      $\mathbf{r} \leftarrow \text{BnB}(\mathbf{Q}, \mathbf{R})$ ; // Obtain  $\mathbf{r}$  using BnB (Algo. 5)
5      $b \leftarrow \mathbf{r}[0], e \leftarrow \mathbf{r}[1]$ ;
6     if  $f(\varphi_{\mathbf{S}[b,e]}, \varphi_{\mathbf{Q}}) \leq \epsilon$  then
7         break;
8     Insert  $\mathbf{S}[b, e]$  into  $\{\mathbf{S}[b_i, e_i]\}_i$ ;
9      $\mathbf{Q} \leftarrow \text{prune}(\mathbf{Q}, \mathbf{r})$ ; // Prune space (Sec. 5.5)
10     $\mathbf{R} \leftarrow \mathbf{Q}.\text{pop}()$ ;
11 return  $\{\mathbf{S}[b_i, e_i]\}_i$ ;
```

when a time series is too long and relatively small portion of frames contain an interested event. Instead, we modify (5.3) for efficient inference of event detection:

$$\boxed{\text{ED}} \quad \max_{b,e} f_{\mathbf{w}}(\varphi_{\mathbf{S}[b,e]}), \quad (5.24)$$

subject to $\ell \leq e - b$

where \mathbf{w} is a pre-trained linear classifier with each element $w_j = \sum_i \alpha_i h_j^i$, and $f_{\mathbf{w}}(\cdot) = \sum_i \alpha_i \langle \cdot, \mathbf{h}^i \rangle$ is the commonality measure based on the classifier. α_i is the weight vector learned during SVM training.

Algorithm: The ED problem in (5.24) becomes supervised detection rather than unsupervised as mentioned in previous sections. The proposed bounds in Sec. 5.3 are thus inapplicable. Due to the summation property of BoTW in (5.1), we decompose the commonality measure into per-frame positive and negative contributions: $f_{\mathbf{w}}(\mathbf{S}[b, e]) = \sum_{i=b}^e (f_{\mathbf{w}}^+(\mathbf{S}[i, i]) + f_{\mathbf{w}}^-(\mathbf{S}[i, i]))$. Denote the longest and the shortest possible searching segments as \mathbf{S}^+ and \mathbf{S}^- respectively, with slight abuse of notation, we reach the bounds:

$$\begin{aligned} l_{\mathbf{w}}(\mathbf{R}) &= f_{\mathbf{w}}^+(\mathbf{S}^-) + f_{\mathbf{w}}^-(\mathbf{S}^+) \\ &\leq f_{\mathbf{w}}^+(\mathbf{S}) + f_{\mathbf{w}}^-(\mathbf{S}) = f_{\mathbf{w}}(\mathbf{S}) \\ &\leq f_{\mathbf{w}}^+(\mathbf{S}^+) + f_{\mathbf{w}}^-(\mathbf{S}^-) = u_{\mathbf{w}}(\mathbf{R}), \end{aligned} \quad (5.25)$$

where $\mathbf{R} = [b, e]$ corresponds to time series \mathbf{S} , instead of previous definition over two time series. With the derived bounds, the CED algorithm can be directly applied for efficient inference of an event of interest.

5.4.4 Comparisons with related work

The proposed CED bear similarities and differences with several related work. Below we discuss in terms of problem definition and technical details.

Algorithm 8: Segment-based Event Detection (ED)

Input : A video S of length n ; a pre-trained linear classifier w
Output: Detected events $\{S[b_i, e_i]\}_i$

```

1  $Q \leftarrow$  empty priority queue; // Initialize  $Q$ 
2  $R \leftarrow 1 \times n$ ; // Initialize  $R$ 
3 while true do
4    $r \leftarrow \text{BnB}(Q, R)$ ; // Obtain  $r$  using BnB (Algo. 5)
5    $b \leftarrow r[0], e \leftarrow r[1]$ ;
6   if  $f_w(S[b, e]) \leq 0$  then
7     break;
8   Insert  $S[b, e]$  into  $\{S[b_i, e_i]\}_i$ ;
9    $Q \leftarrow \text{prune}(Q, r)$ ; // Prune space (Sec. 5.5)
10   $R \leftarrow Q.\text{pop}()$ ;
11 return  $\{S[b_i, e_i]\}_i$ ;

```

Problem definition: Although CED achieves discovery via “matching” between subsequences, it has fundamental differences from standard matching problems. For instance, CED allows *many-to-many* mapping (e.g., Sec. 5.6.1), while standard matching algorithms assume *one-to-one* or *one-to-many* mapping. Moreover, a matching problem (e.g., graph matching or linear assignment) typically measures sample-wise similarity or distance to determine correspondence between one another, e.g., a feature vector on a node in a graph. CED uses bag-of-words representation that aggregates multiple samples (i.e., frames) into one vector, making the application of standard matching methods non-trivial.

CED is also different from time warping (e.g., dynamic time warping [132]) and temporal clustering (e.g., aligned cluster analysis [291]). Time warping aims to find the optimal match between two given sequences that allow for stretched and compressed sections of the sequences. Given this goal, time warping assumes the beginning and the ending frames of the sequences to be fixed, and performs matching on entire sequence. Similarly, temporal clustering considers entire sequence in its objective, and hence is likely to include irrelevant temporal segments in one cluster. On the contrary, CED does not assume fixed beginning and ending frames, instead directly targeting at subsequence-subsequence matching, and thus enables a large portion of irrelevant information to be ignored.

Technical details: Technically, the proposed B&B framework is closely related to Efficient Subwindow Search (ESS) [142] and Spatio-Temporal B&B (STBB) [274]. However, they have at least three differences. (1) *Learning framework*: ESS and STBB are supervised techniques that seek for a confident region according to a pre-trained classifier. CED is unsupervised, and thus requires no prior knowledge. (2) *Bounding functions*: We design new bounding functions for the unsupervised CED problem. Moreover, ESS and STBB consider only upper bounds, while CED can incorporate both upper and lower bounds. (3) *Search space*: ESS and STBB search over spatial coordinates of an image or a spatio-temporal volume in a video, while CED focuses on temporal positions over time series.

For segment-based event detection (ED), we acknowledge its similarity with the version of

STBB that omits spatial volume. Both address efficient search in a one-dimension time series, and differ in the following ways. (1) *Objective*: ED searches for segments with maximal, positive *segment-based* decision values. STBB uses a Kadane’s algorithm for *frame-based* max subvector search, which potentially lead to inferior detection performance because the max sum is usually found in an overly-large segment (as can be seen in Sec. 5.6.3). (2) *Searching strategy*: ED prunes the search space to avoid evaluating segments where an AU is unlikely to occur; STBB evaluates every frame. (3) *Inputs*: ED can take the minimal length and normalized histograms as input, yet it is unclear for STBB to accommodate such input because of the linear nature of Kadane’s algorithm.

5.5 Extensions to the B&B framework

Given the CED algorithm and variants described above, this section describes extensions to discovery among multiple time series and discover multiple commonalities. Due to the special diagonal nature of SD, we also introduce its acceleration using warm start and parallelism. Fig. 5.6 illustrates these extensions.

Discovery among multiple time series: We have described above how the B&B framework can discover temporal commonalities within a pair of time series. Here we show that the framework can be directly extended to capture commonality among multiple time series. Specifically, we formulate the discovery among N sequences $\{\mathbf{S}^i\}_{i=1}^N$ by rewriting (5.3) as:

$$\begin{aligned} \max_{\{b_i, e_i\}_{i=1}^N} \quad & F\left(\{\phi_{\mathbf{S}^i[b_i, e_i]}\}_{i=1}^N\right) \\ \text{subject to} \quad & \ell \leq e_i - b_i, \forall i \in \{1, \dots, N\}, \end{aligned} \quad (5.26)$$

where $F(\cdot)$ is a similarity measure for a set of sequences and defined as the sum of pairwise similarities:

$$F\left(\{\phi_{\mathbf{S}^i[b_i, e_i]}\}_{i=1}^N\right) = \sum_{i \neq j} f(\phi_{\mathbf{S}^i[b_i, e_i]}, \phi_{\mathbf{S}^j[b_j, e_j]}). \quad (5.27)$$

Given a rectangle set R and a time series pair $(\mathbf{S}^i, \mathbf{S}^j)$, we rewrite their pairwise bounds in Sec. 5.3 as $l_f^{ij}(R)$ and $u_f^{ij}(R)$. The bounds for $F(\cdot, \cdot)$ can be defined as:

$$\begin{aligned} l_F(R) &= \sum_{i \neq j} l_f^{ij}(R) \leq F\left(\{\phi_{\mathbf{S}^i[b_i, e_i]}\}_{i=1}^N\right) \\ &\leq \sum_{i \neq j} u_f^{ij}(R) = u_F(R). \end{aligned} \quad (5.28)$$

Given this bound, Algos. 5 and 6 can be directly applied to discover commonalities among multiple time series.

Discover multiple commonalities: Multiple commonalities occur frequently in real videos, while the B&B framework only outputs one commonality at a time. Here, we introduce a strategy that prunes the search space to accelerate multiple commonality discovery. Specifically, we repeat the searching algorithm by passing the priority queue Q from the previous search to the next, and continue the process until a desired number of solutions is reached, or the returned commonality

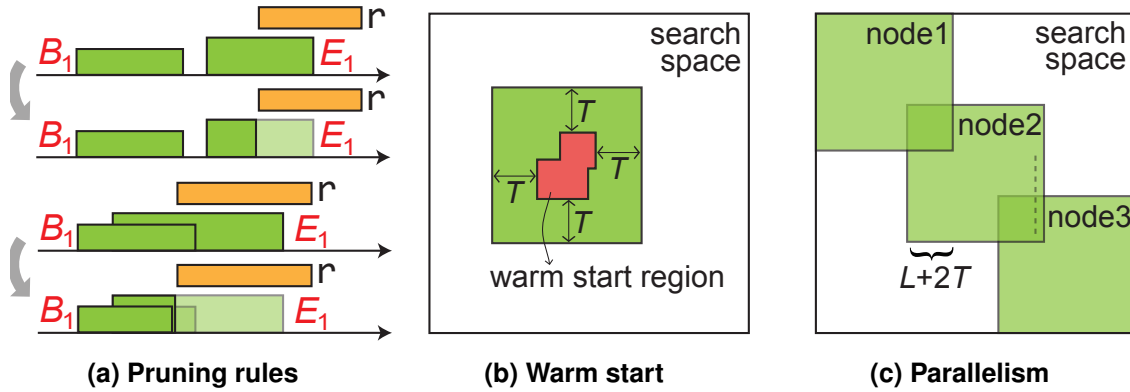


Figure 5.6: Illustration of extensions to SD: (a) pruning rules applied to multiple-commonality discovery, (b) SD with warm start, and (c) SD with parallelism.

measure $f(\cdot, \cdot)$ is less than some threshold. The threshold can be also used for excluding undesired discoveries for the scenario where two sequences have no events in common. That is, if the first discovery does not pass a pre-defined threshold, the algorithm returns empty because the subsequent discoveries perform no better than the first one. Fig. 5.6(a) illustrates an example of the pruning rule when E_1 overlaps with a previously discovered solution r . Because we want to exclude the same solution for the next discovery, the search region is updated by avoiding overlapping with previous solution. For axes of both S^1 and S^2 , all R overlapped with r is updated using the same rule, or discarded if the updated R is empty, *i.e.*, $|R| = 0$. The updated rectangle sets, along with their bounds, are then pushed back to Q before the next search.

This pruning strategy is simple yet very effective. Previously derived bounds remain valid because each updated set is a subset of R . In practice, it dramatically reduces $|Q|$ for searching the next commonality. For example, in synchrony discovery of Fig. 5.5, $|Q|$ is reduced 19% for the second search, and 25% for the third SD. Note that this pruning strategy differs from conventional detection tasks, *e.g.*, [142, 274], which remove the whole spatial or temporal region for the next search. In CED, temporal segments can be many-to-many matching, *i.e.*, $S^1[b_1, e_1]$ can match multiple segments in S^2 and vice versa. Thus, removing any segments from either time series would cause missing matches. This strategy allows us to maintain *many-to-many* matching.

SD with Warm start: Due to the B&B nature, SD exhibits poor worst-case behavior, leading to a complexity as high as an exhaustive search [180]. On the other hand, B&B can quickly identify the exact solution when a local neighborhood contains a clear optimum [142]. Given this motivation, we explore a “warm start” strategy that estimates an initial solution with high quality, and then initializes SD around the solution. Estimating an initial solution costs only few percentage of total iterations, and thus can effectively prune branches in the main SD algorithm. Fig. 5.6(b) illustrates the idea. Specifically, we run a sampled sliding window with stepsize=10, sort the visited windows according their distances, and then determine a warm start region around the windows within the top one percentile. Then the SD algorithm is performed only within an expanded neighborhood around the warm start region.

SD with Parallelism: The use of parallelism to speed up B&B algorithms has emerged as a way for large problems [91]. Based on the block-diagonal structure in the SD search space, this

Dataset	min	25-th	50-th	75-th	max	std
RU-FACS [10]	13	42	79	159	754	125.6
mocap [1]	41	142	175	218	483	67.5

Table 5.1: Distribution of event lengths in different datasets: min and max show the shortest and longest length of a common event. 25-, 50-, and 75-th indicate degrees of percentiles.

section describes an parallelized approach to scale up SD for longer time series. In specific, we divide SD into subproblems, and perform the SD algorithm solve each in parallel. Because each subproblem is smaller than the original one, the number of required iterations can be potentially reduced. As illustrated in Fig. 5.6(c), the original search space is divided into overlapping regions, where each can be solved using independent jobs on a cluster. The results are obtained as the top k rectangles collected from each subproblem. Due to the diagonal nature of SD in the search space, the final result is guaranteed to be a global solution. The proposed structure enables static overload distribution, leading to an easily programmable and efficient algorithm.

5.6 Experiments

We evaluated the B&B framework in three tasks: Temporal commonality discovery (5.6.1), synchrony discovery (5.6.2), and supervised event detection. We used face videos and motion capture data for illustration. However, any signals that can be quantifies into histograms can be directly applied to the B&B framework.

5.6.1 Common event discovery

In the first experiment, we evaluated CED on discovering common facial events, and discovering multiple common human actions. Table 5.1 shows the distribution of event lengths in respective

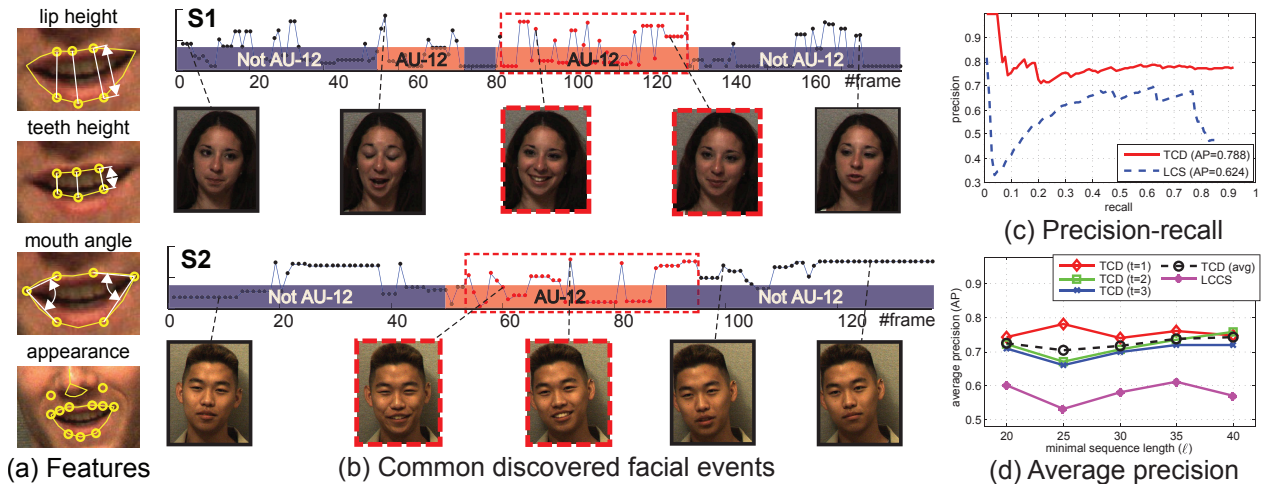


Figure 5.7: Results on discovering common facial actions: (a) Facial features extracted from the tracked points. (b) An example of common discovered facial events (indicated by dashed-line rectangles). (c)(d) Accuracy evaluation on precision-recall and average precision (AP).

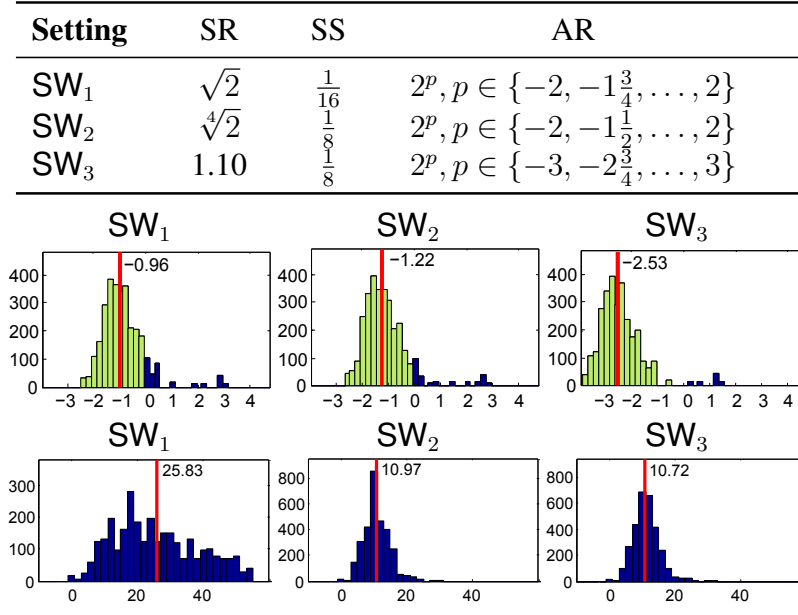


Figure 5.8: Efficiency evaluation between CED and alternative sliding window (SW) approach. Top: Parameter settings [142, 252]: size-ratio (SR), stepsize (SS), and aspect ratios (AR). Middle: Histogram of ratio of #evaluation: $\log \frac{n_{\text{CED}}}{n_{\text{SW}_i}}$. Red vertical lines indicate the average. Light green bars show CED performs less evaluations than SW; dark blue bars represent the opposite. Bottom: Histogram of difference between resulting commonality measure: $f_{\ell_1}(\mathbf{r}^{\text{SW}_i}) - f_{\ell_1}(\mathbf{r}^{\text{CED}})$.

experiments. The mixture of long and short events indicates a more realistic scenario of handling events with slow and fast motions. Specifically, for RU-FACS, we computed the distribution of AU12 events among the 4,950 sequence pairs. For mocap, the distribution was computed on a total of 25 actions from 45 sequence pairs (details below).

Discovering common facial events

This experiment evaluates the CED algorithm to find similar facial events in the RU-FACS dataset [10]. The RU-FACS dataset consists of digitized video of 34 young adults. They were recorded during an interview of approximately 2 minutes duration in which they lied or told the truth in response to an interviewer’s questions. Pose orientation was mostly frontal with moderate out-of-plane head motions. We selected the annotation of Action Unit (AU) 12 (*i.e.*, mouth corner puller) from 15 subjects that had the most AU occurrence. We collected 100 video segments containing one AU 12 and other AUs, resulting in 4,950 pairs of video clips from different subjects. For each video, we represented features as the distances between the height of lips and teeth, angles for the mouth corners and SIFT descriptors in the points tracked with Active Appearance Models (AAM) [166] (see Fig. 5.7(a) for an illustration).

Accuracy evaluation: Because the CED problem is relatively new in computer vision, to our knowledge there is no baseline we could directly compare to. Instead, we compared against the state-of-the-art sequence matching approach: Longest common consecutive subsequence matching (LCCS) [256]. Observe that when the per-frame feature was quantized into a temporal word, the unsupervised CED problem can be naturally interpreted as an LCCS. Following LCCS that uses a 0-1 distance, we chose ℓ_1 -distance for CED. Note that the segment-based BoTW representation is not helpful for LCCS [256], because LCCS computes matches only at frame-level. The minimal length ℓ was fixed as the smaller length of ground truth segments for both LCCS and CED. Given a discovered solution \mathbf{r} and a ground truth \mathbf{g} that indicates a correct matching, we measured their *overlap score* [80] as $\text{overlap}(\mathbf{r}, \mathbf{g}) = \frac{\text{area}(\mathbf{r} \cap \mathbf{g})}{\text{area}(\mathbf{r} \cup \mathbf{g})}$. The higher the overlap score, the better the algorithm

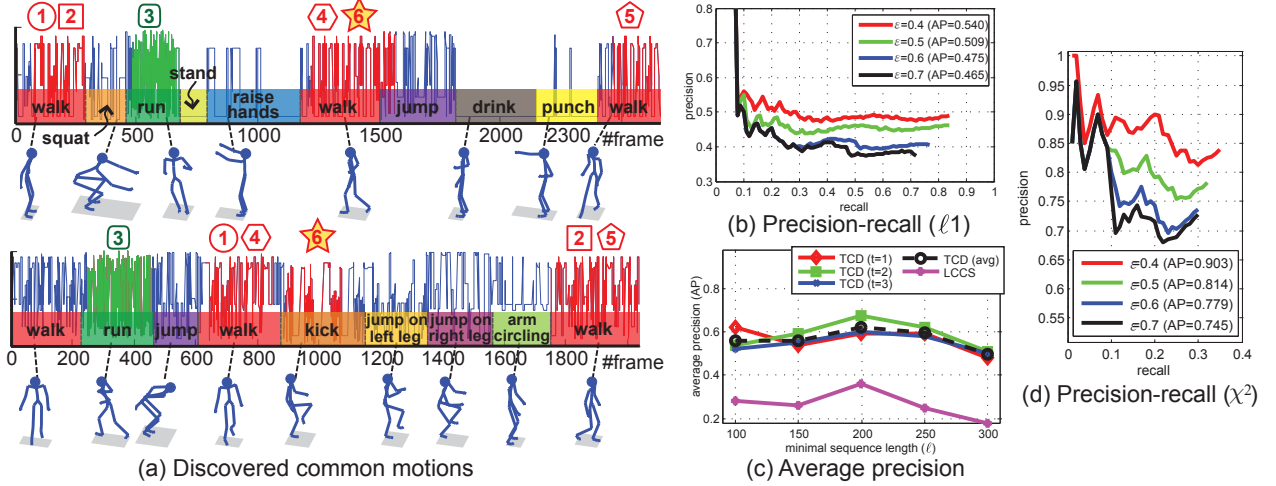


Figure 5.9: (a) Top six discovered common motions, indexed by numbers. Note that the shaded star (number 6) indicates an incorrect discovery that matched *walk* and *kick*. (b)(c) Precision-recall and average precision on ℓ_1 distance. (d) Precision-recall on χ^2 distance.

discovered the commonality. We considered r to be a correct discovery if the overlap score is greater than 0.5.

Fig. 5.7(b) shows an example of a correct discovery of AU 12. In this example, CED was able to correctly locate an AU 12 segment with >0.8 overlap score. Fig. 5.7(c) plots the precision-recall curves for the first discovery of CED and LCCS. We reported the average precision (AP) [80] and found CED outperformed LCCS by 15%. Unlike LCCS that sought for identical subsequences, CED considered a distribution of temporal words present in two videos, and thus was able to more reliably capture temporal commonality in real-world videos. Fig. 5.7(d) shows the average precision of our approach under different parameters. We varied the minimal sequence length ℓ in $\{20, 25, \dots, 40\}$, and examined the AP of the t -th result. As can be observed from the averaged AP (black dashed line), our B&B approach performed more stably across different combinations of ℓ and t . As a result, CED performed on average 16% higher AP than LCCS in discovering the common facial actions.

Efficiency evaluation: Using the above settings, we evaluated speedup of the CED algorithm against exhaustive sliding window (SW) approach, which was implemented following parameter settings in [142, 252]. Fig. 5.8(a) shows these settings denoted as SW_i ($i=1, 2, 3$). Denote lengths of two time series as m, n and the minimal length for each sequence is ℓ , we set the maximal and minimal rectangle size for SW to be $(m \times n)$ and $(\ell\sqrt{AR} \times \frac{\ell}{\sqrt{AR}})$, respectively. To be independent of implementation, we measured the *discovery speed* as the number of evaluation for the bounding functions, referred as n^{CED} and n^{SW_i} for CED and SW_i respectively. Fig. 5.8(b) shows the histograms of the log ratio for n^{CED}/n^{SW_i} . The smaller the value, the less times CED has to evaluate the distance function. As can be seen, although SW was parameterized to search only a subset of the search space, CED searched the entire space yet still performed on average 6.18 times less evaluations than SW. To evaluate the *discovery quality*, we computed the distance difference measured by CED and SW, i.e., $f_{\ell_1}(r^{SW_i}) - f_{\ell_1}(r^{CED})$. The larger the difference, the lower quality of discovery SW got. Fig. 5.8(c) shows the histograms of such differences. One can observe

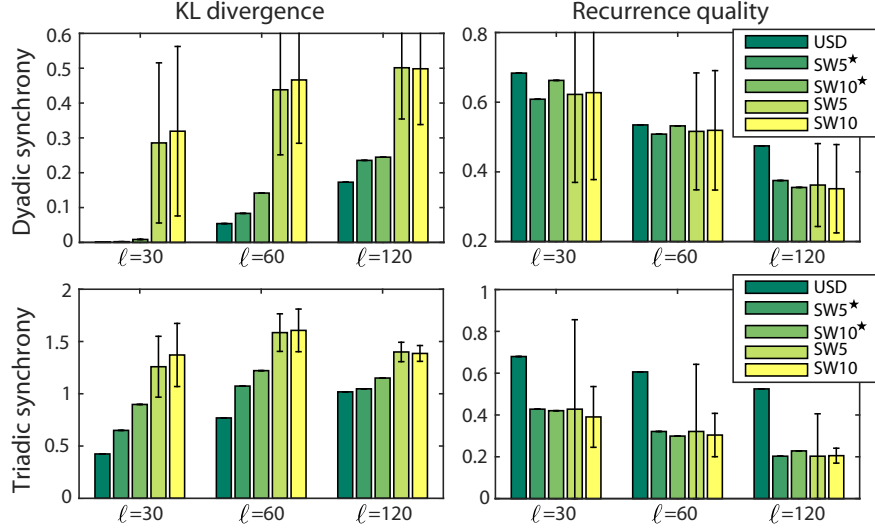


Figure 5.10: Analysis on top 10 discovered dyadic and triadic synchronies of the GFT dataset. SW denoted with \star indicates the optimal windows discovered, and without \star indicates the average and standard deviation over all visited windows.

that the differences are always greater than or equal to zero. This is because our method provably finds the global optimum. On the other hand, SW only performed a partial search according to its parameters, and thus was likely to reach larger distance than ours.

Discover multiple common human motions

This experiment attempts to discover *multiple* common actions using the CMU-Mocap dataset [1]. We used Subject 86 that contains 14 long sequences with 1,200~2,600 frames and human action annotation [9]. Each sequence contains up to 10 actions (out of a total of 25) such as *walk*, *jump*, *punch*, etc. See Fig. 5.9(a) for an example. Each action ranged from 100 to 300 frames. We randomly selected 45 pairs of sequences and discovered common actions among each pair. Each action was represented by root position, orientation and relative joint angles, resulting in a 30-D feature vector. Note that this experiment is much more challenging than the previous one due to the large number of frames and more complicated actions. In this case, we excluded SW for comparison because it needs 10^{12} evaluations that is impractical.

Fig. 5.9(a) illustrates the first six common motions discovered by CED. A failure discovery is shown in the shaded number 6, which matches *walk* to *kick*. An explanation is because these actions were visually similar, resulting in similar features of joint angles. Fig. 5.9(b) shows the precision-recall curve for different values of overlapping threshold ε . Using ℓ_1 distance, the curve decreases about 10% AP when the overlap score ε raises from 0.4 to 0.7, which implies that we can retain higher quality results without losing too much precision. Fig. 5.9(c) shows the average precision over various ℓ on the t -th discovered result. LCCS performed poorly to obtain long common subsequences because human motions have more variability than just one facial event (e.g., AU-12). On the contrary, CED used BoTW representation, and thus allowed more descriptive power for activity recognition. Fig. 5.9(d) shows the precision-recall curve evaluated with χ^2 distance. Although the Mocap dataset is very challenging in terms of various motions and diverse sequence



Figure 5.11: Top 10 discovered synchronies from groups 128, 071, 094, 113 and 049 in the GFT dataset. Each column indicates a discovered synchrony and its frame number. The SD algorithm correctly matched the states of *smiling*, *talking* and *silent*.

lengths, the CED algorithm with χ^2 performed 30% better than ℓ_1 and LCCS. It suggest χ^2 is a more powerful commonality measure for histograms than ℓ_1 . Overall, using the χ^2 measurement and $\varepsilon = 0.5$, CED achieved 81% precision.

5.6.2 Synchrony discovery

This section evaluates SD on three types of human interactions: *social group interaction*, *parent-infant interaction*, and *intra-person human actions*.

Social group interaction

This experiment investigates discovery of synchronies in social group interaction. We used the GFT dataset [209] that consists of 720 participants recorded during group-formation tasks. Previously unacquainted participants sat together in groups of 3 at a round table for 30 minutes while getting to know each other. We used 2 minutes of videos from 48 participants, containing 6 groups of two



Figure 5.12: Discovered synchronies from 6 groups of parent-infant interaction. Each column indicates a discovery and its #frame.

subjects and 12 groups of three subjects. SD was performed to discover *dyads* among groups of two, and *triads* among groups of three. Each video was tracked with 49 facial landmarks using IntraFace [60]. We represented each face by concatenating appearance features (SIFT) and shape features (49 landmarks). Denote the video index set as \mathcal{A} , we evaluated the discovery performance by the *recurrence consistency* [63]:

$$\mathcal{Q}(\mathbf{r}) = \frac{1}{C \prod_i n_i} \sum_c \sum_{(i,j) \in \mathcal{A}} \sum_{p,q} I(\mathbf{Y}_i^c[p] = \mathbf{Y}_j^c[q]), \quad (5.29)$$

where $I(X)$ is an indicator function returning 1 if the statement X is true and 0 otherwise, and $\mathbf{Y}_i^c[p]$ denote the c -th class annotation corresponding to the p -th frame in \mathbf{S}^i . In this dataset, we used annotations of AUs (10,12,14,15,17,23,24) that appear most frequently.

As the minimal length ℓ is an empirical parameter to determine, we examined SD with $\ell \in \{30, 60, 120\}$, resulting in synchronies that last at least 1, 2 and 4 seconds; we set the synchrony offset $T = 30$ (1 second). Baseline SW was performed using step sizes 5 and 10. Symmetrized KL divergence was used as the distance function. We evaluated the distance and quality among the optimal window discovered, as well as the average and standard deviation among all windows to tell a discovery by chance. Fig. 5.10 shows the averaged KL divergence and quality among top 10 discovered dyadic and triadic synchronies. As can be seen, SD always guarantees the lowest divergence because of its nature to find the exact optimum. The recurrence quality decreases while ℓ grows, showing that finding a synchrony with longer period while maintaining good quality is harder than finding one with shorter period. Note that, although the discover quality is not guaranteed in an unsupervised discovery, SD consistently maintained the best discovery quality across various lengths. This result illustrates the power of our unsupervised method that agrees with that of supervised labels.

Parent-infant interaction

Parent-infant interaction is critical for children in early development and social connections. This section attempts to characterize their affective engagement by exploring the moments where the behavior of both the parent and the infant are correlated. We performed this experiment on the

	Pair	(1,11)	(2,4)	(3,13)	(5,7)	(6,8)	(9,10)	(12,14)	Avg.
χ^2 -distance	USD	6.3	1.2	4.7	2.6	0.1	0.2	11.9	3.9
	SW ₅ [*]	6.5	1.3	6.7	5.4	0.1	0.4	12.0	4.6
	SW ₁₀ [*]	6.7	2.7	6.7	10.1	0.2	0.7	14.3	5.9
	SW ₅ ^{μ}	97.1	76.9	81.4	64.2	89.3	172.0	334.5	130.8
	SW ₅ ^{σ}	33.8	74.4	53.8	28.2	79.2	117.7	345.1	104.6
	SW ₁₀ ^{μ}	94.8	77.3	81.8	63.2	87.1	170.2	327.2	128.8
	SW ₁₀ ^{σ}	34.3	74.1	54.2	28.3	79.4	117.8	341.5	104.2
Rec. consistency	USD	0.89	0.85	0.46	0.90	1.00	0.64	0.76	0.79
	SW ₅ [*]	0.95	0.81	0.50	0.84	1.00	0.69	0.73	0.79
	SW ₁₀ [*]	0.95	0.75	0.50	0.64	1.00	0.55	0.00	0.63
	SW ₅ ^{μ}	0.07	0.32	0.09	0.07	0.08	0.13	0.12	0.12
	SW ₅ ^{σ}	0.16	0.33	0.25	0.20	0.21	0.29	0.22	0.24
	SW ₁₀ ^{μ}	0.08	0.31	0.09	0.07	0.09	0.13	0.12	0.13
	SW ₁₀ ^{σ}	0.19	0.33	0.26	0.21	0.22	0.29	0.23	0.25

Table 5.2: Distance and quality analysis on CMU Mocap dataset: **(top)** χ^2 distance using 1e-3 as unit, **(bottom)** recurrent consistency. SW_s^{*} indicates the optimal window found by SW_s with step size $s = 5, 10$; SW_s ^{μ} and SW_s ^{σ} indicate average and standard deviation among all windows. Scores of the best discovery are marked in bold.

mother-infant interaction dataset [170]. Participants were 6 ethnically diverse 6-month-old infants and their parents (5 mothers, 1 father). Infants were positioned in an infant-seat facing their parent who was seated in front of them. We used 3 minutes of normal interaction where the parent plays with the infant as they might do at home. Because this dataset does not provide ground truth annotations, we only evaluate the results quantitatively. After the faces were tracked, we used only the shape features because the appearance of adults and infants are different. Throughout this experiment, we set $\ell = 80$ and $T = 40$.

Fig. 5.12 illustrates three discovered synchronies among all parent-infant pairs. As can be seen, many synchronies were discovered as the moments when both infants and parents exhibit strong smiles, serving as a building block of early interaction [170]. Besides smiles, a few synchronies showed strong engagement in their mutual attention, such as the second synchrony of group ① where the infant cried after the mother showed a sad face, and the second synchrony of the second group where the mother stuck her tongue out after the infant did so. These interactive patterns offered solid evidence of a positive association between infants and their parents.

Human actions

This section provides an objective evaluation of discovering human actions on the CMU Mocap dataset [1], as used in Sec. 5.6.1 Mocap data provides high-degree reliability in measurement and serves as an ideal target for a clean-cut test of our method. To mimic a scenario for USD, we grouped the sequences into 7 pairs as the ones containing similar number of actions, and trimmed each action to up to 200 frames. USD was performed using $\ell = 120$ and $T = 50$.

Table 5.2 summarizes the USD results compared with the baseline sliding window (SW). Results are reported using χ^2 -distance and the recurrent consistency described in (5.29). A threshold of 0.012 was manually set to discard discovery with large distance. We ran SW with step sizes 5 and 10, and marked the windows with the minimal distance as SW₅^{*} and SW₁₀^{*}, respectively. Among all, USD discovers all results found by SW. To understand how well a prediction by chance can be, all windows were collected to report average μ and standard deviation σ . As can be seen, on average,

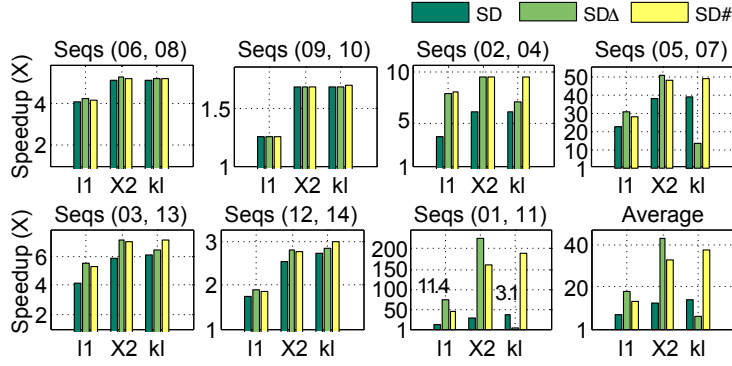


Figure 5.13: Speedup of SD against sliding window (SW) on CMU-Mocap. All 7 pairs of sequences from subject 86 were evaluated. The speedup was computed as the relative number of evaluations $N^{\text{SW}}/N^{\text{USD}}$ using ℓ_1 , χ^2 and symmetrized KL divergence.

a randomly selected synchrony can result in large distance over 100 and low quality below 0.3. USD maintained an exact minimal distance with good qualities as the ones found by exhaustive SW. Note that, because USD is totally unsupervised, the synchrony with minimal distance may not necessarily guarantee the highest quality.

Fig. 5.13 shows the speed up of USD against exhaustive SW. SD and its extensions demonstrated an improved efficiency over SW. In some cases, SD^Δ improved search speed by a large margin, *e.g.*, in (01,11) with χ^2 -distance reached a speed boost over 200 times. Across all metrics, the speed up of SD^Δ was less obvious with symmetrized KL divergence. $\text{USD}^\#$ was implemented on a 4-core machine; an extension to larger clusters is possible yet beyond the scope of this study. On average, $\text{SD}^\#$ consistently accelerated the original USD due to parallelism.

Fig. 5.14 shows the qualitative results on all 7 pairs, annotated with ground truth and the discovered synchronies. As can be seen, USD allows to discover multiple synchronies with varying lengths. Although some discovered synchronies contain disagreed action labels, one can observe that the discoveries share reasonable visual similarity, *e.g.*, in pair (9,10), the “look around” action in sequence 9 was performed when the subject was seated, sharing the similarity with the “sit” action in sequence 10.

5.6.3 Event detection

This experiment evaluates the computation time and performance of event detection on the GFT dataset [209], as used in Sec. 5.6.2. The task is to localize events of 12 AUs using a pre-trained segment-based linear SVM classifier. Specifically, we compared ED with a hybrid SVM-HMM [138] (denoted HMM hereafter for simplicity) and the state-of-the-art event detection algorithms, including a dynamic programming (DP) approach [112] and the Kadane’s algorithm used in STBB [274]. We trained a frame-based SVM for each AU, and used the same SVM for the detection task on different methods. For SVM-HMM, the HMM has two states, *i.e.*, activation or inactivation of an AU. The state transition probabilities and the a-priori probability were estimated by the frequency of an AU activation in the training data. The emission probabilities of HMM was computed based on normalized SVM output using Platt’s scaling [192]. During test, the most likely AU state path for each video was determined by a standard Viterbi algorithm, which has a complexity $\mathcal{O}(|s|^2 \times N)$, where $|s| = 2$ is the number of states and N is the number of frames of a test video. For both ED and DP, we set the minimal discovery length $\ell = 30$. Unlike previous work that require temporal segmentation [144, 202, 213], we focused on joint detection and segmentation of a temporal event. Specifically, we implemented a baseline HMM and the state-of-the-art dynamic programming (DP)

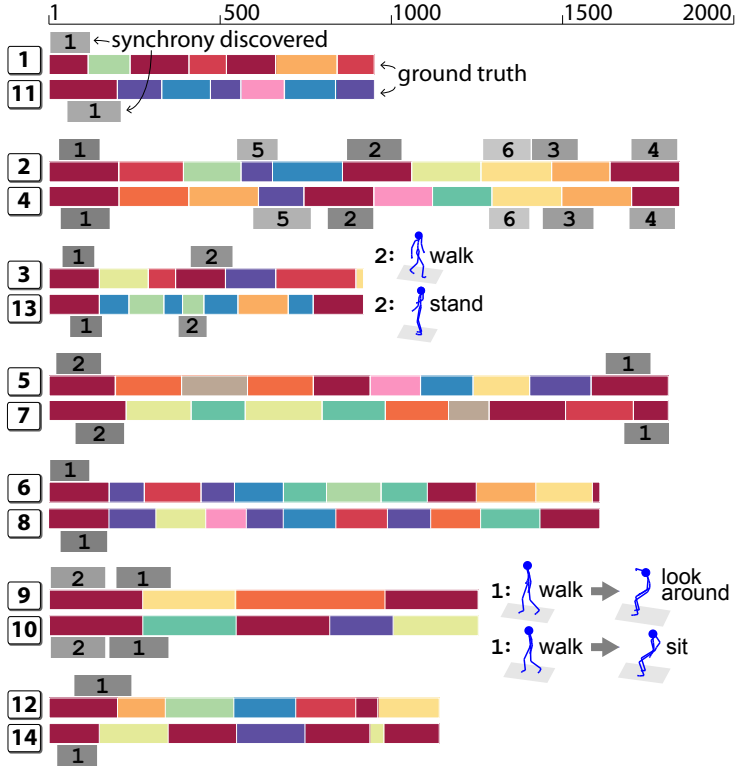


Figure 5.14: Discovered synchronies on 7 pairs of Subject 86 in CMU-Mocap dataset. Each pair is annotated with ground truth (colorful bars, each represents an action), and synchronies discovered by our method (shaded numbers). Synchronies with disagreed action labels are visualized.

approach [112]. In HMM, the most likely AU state path for each video was determined by a standard Viterbi algorithm. We set the maximal segment lengths of DP in $\{100, 150, 200\}$, denoted as DP_100, DP_150, and DP_200 in the figure. For evaluation, we used the F1-event metric [65] defined as $F1\text{-event} = \frac{2EP \cdot ER}{EP + ER}$, where EP and ER stand for event-based prevision and event-based recall. Unlike a standard F1 score, F1-event focuses on capturing the temporal consistency of prediction. An event-level agreement holds if the overlap of two temporal segments is above a certain threshold.

Fig. 5.15(a) shows the F1-event curve w.r.t. different event overlapping thresholds. Overall DP and ED performed better than the baseline HMM. The performance of DP dropped when threshold > 0.6 , which implies DP missed highly overlapped events during detection. This is because DP performed exhaustive search, and thus requested a maximal search length for computational feasibility. On the other hand, ED by construction excludes such limitation. Fig. 5.15(b) shows the running time on a 2.8GHz dual core CPU machine. Each detected AU event is plotted in terms of the running time and respective video length (#frame). As can be seen, the computation time for DP increased linearly with video length, while ED maintained invariance of video length. These results suggest that ED was able to perform comparably with significantly improved efficiency for event detection.

Figs. 5.16 shows the trend of running time v.s. F1-event and F1 score across ED and all alternative methods. Each marker indicates a detection result for a sequence. For visualization purpose, we randomly picked 120 sequences to include in this figure. The quantitative evaluation on the entire dataset is shown in Table 5.3. As can be seen in Figs. 5.16(a) and (b), STBB and HMM performed significantly faster than others due to their linear nature in computation. In general, for

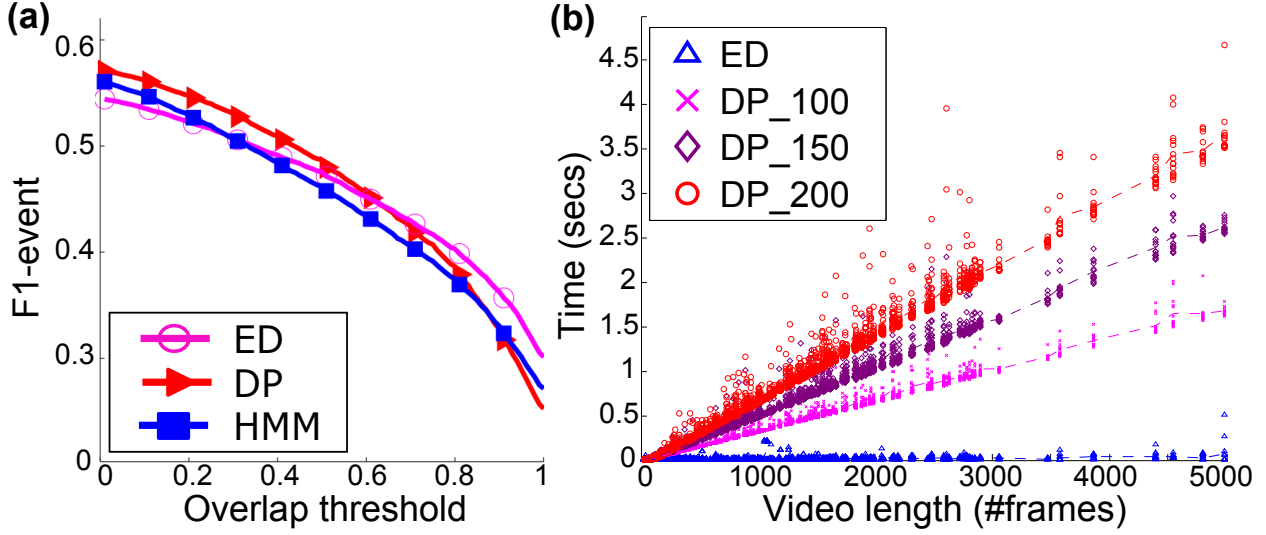


Figure 5.15: Comparison between Dynamic Programming (DP) and ED in terms of (a) F1-even and (b) computation time (sec).

F1-event and F1, STBB led to suboptimal performance because events with activation are usually found in over-length segments. Fig. 5.16(c) illustrates detection results of three subjects. In all cases, it reveals the over-length detection of STBB due to its consideration of max subvectors. As can be seen, STBB tends to include a large temporal window so that the sum of decision values is maximized. HMM took SVM outputs as emission probability, and thus performs similarly as a frame-based SVM. HMM tends to generate lower F1-event, as also suggested in Figs. 5.15(a). This is because of the memoryless property considered in the Markov chain, *i.e.*, the future state only depends upon the present state. On the contrary, ED and DP produced more visually smooth results due to their segment-based detection. Similar to Fig. 5.15(b), we observed that, with comparable performance, ED is consistently faster over DP with different parameters.

Table 5.3 summarizes the comparison between ED and alternative methods in terms of running time, F1-Event and F1 scores averaged over sequences in the entire dataset. As what we have observed in Fig. 5.16, STBB had the smallest running time yet with the worst performance. Among the top performing DP and ED, without losing much accuracy, ED improved the speed against DP from about 6x to 14x.

Method	Time (sec)	F1E	F1
STBB	0.003±0.002	0.297±0.256	0.420±0.270
HMM	0.090±0.049	0.405±0.209	0.698±0.182
DP100	3.987±2.184	0.586±0.188	0.756±0.179
DP150	6.907±3.720	0.586±0.188	0.756±0.179
DP200	9.332±5.268	0.586±0.188	0.756±0.179
ED (ours)	0.668±0.873	0.572±0.197	0.753±0.165

Table 5.3: Comparison between ED and alternative methods in terms of running time, F1-event (F1E), and F1 on the supervised AU detection task.

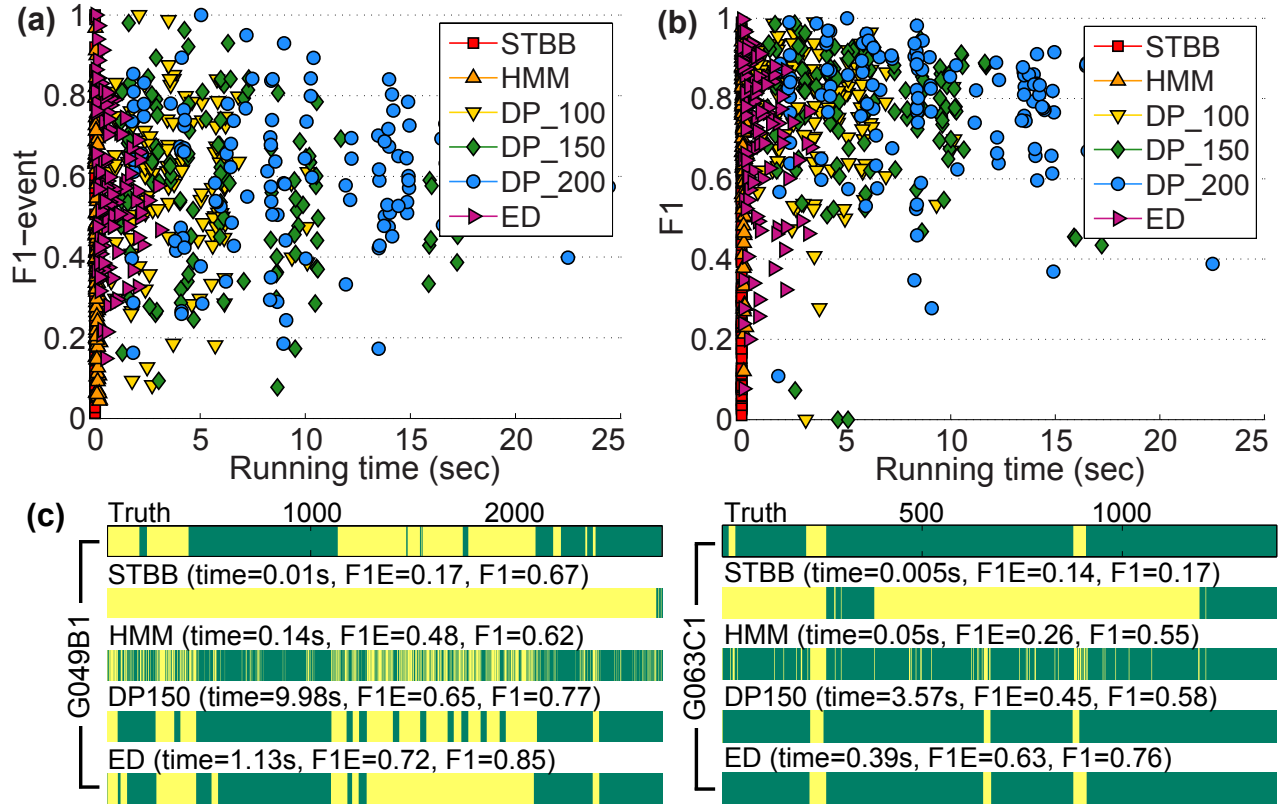


Figure 5.16: Comparison between ED and alternative approaches in terms of: (a) F1-event v.s. time, (b) F1 v.s. time, and (c) comparison between ground truth and detection results on 3 subjects. Light yellow and dark green indicate activation and deactivation of AU12, respectively.

5.7 Summary

This chapter introduced a novel Branch-and-Bound (B&B) framework to address unsupervised and supervised temporal commonality discovery. With numerous newly derived bounding functions, the B&B framework guarantees a global solution in an empirically efficient manner. We showed a slight modification of the B&B framework leads to four applications: Temporal commonality discovery, synchrony discovery, video search and supervised event detection. In addition, we demonstrated that the searching procedure can be extended to discovery among multiple time series, discover multiple commonalities and accelerated with warm start and parallelism. Results on discovering common facial actions, human motions and supervised detection showed the effectiveness and efficiency of the proposed method.

Future work includes promoting the scalability of the proposed algorithm. Given current pairwise design, the computational complexity grows quadratically with the number of input sequences. One direction is to pursue parallelism, *i.e.*, compute pairwise bounds independently using clusters or multi-threading, and then aggregate these bounds into a overall score.

■

Chapter 6

Conclusion and Future Work

“Simplicity is the ultimate sophistication.”

Leonardo da Vinci

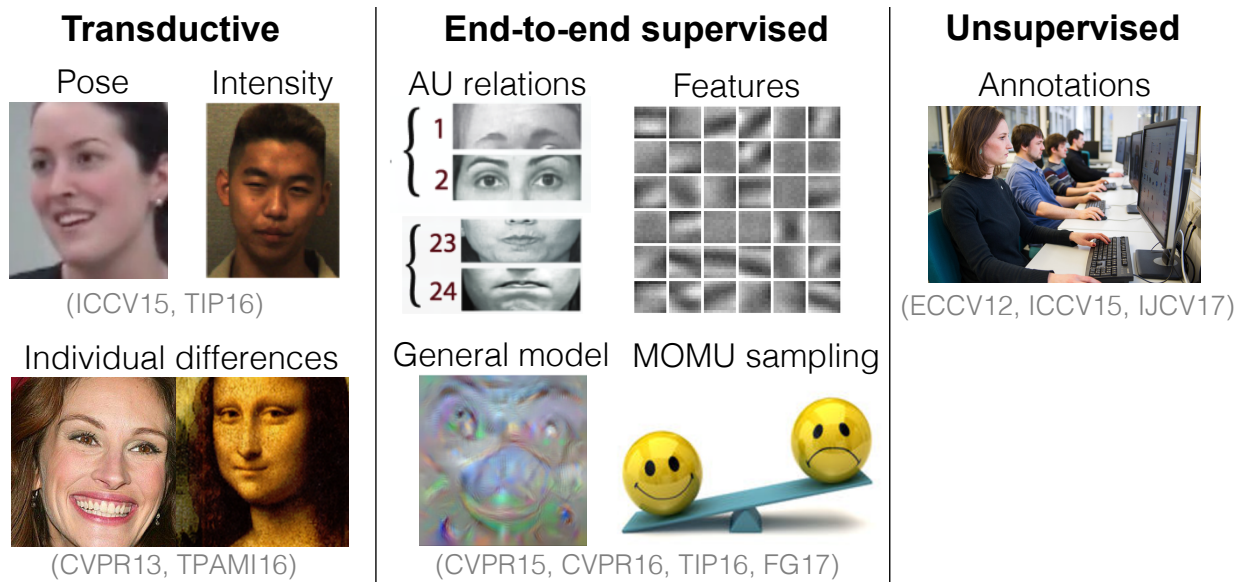


Figure 6.1: An illustration that summarizes three learning frameworks studied in this thesis and related issues addressed in each framework.

This thesis has focused a collection of work on automatic analysis of facial actions. In particular, we have developed three frameworks that involve transductive, end-to-end supervised, and unsupervised learning. As illustrated in Fig. 6.1, each framework focuses on addressing one or multiple challenges with regards to facial action analysis (AFA). While these methods have produced promising results, there remains several challenging problems in the AFA community. In the following, we conclude with a summary of our contributions and discuss potential directions to further this research.

6.1 Summary of Contributions

A transductive framework for personalized facial expression analysis

Chapter 3 identified that challenges remain in AFA systems are occasioned by individual differences such as behavior, facial morphology (face shape, texture, etc.), recording environments, ethnicity or racial background, and developmental level. A possible solution to address such challenge could be person-specific classifiers, which are available for a paucity of training data. Instead, observing that test samples are of the same subject and given for free at prediction time, we employed a transductive framework—Selective Transfer Machine (STM)—to *personalize* a generic classifier. Without additional labels from the test subject, STM is able to attenuate person-specific mismatches during classifier training. For both tasks of facial AU detection and expression recognition, we showed that STM yields consistent improvement over generic classifiers on four benchmark datasets.

An end-to-end supervised framework for multi-label AU detection

Chapter 4 presented a more powerful alternative afforded by an end-to-end supervised framework (*i.e.*, deep learning), which can be naturally extended to address multiple challenges in one joint framework. These challenges include learning an optimal representation for classification, variation in base rates of action units (AUs), correlation between AUs, temporal consistency and balancing AU distributions in a multi-label manner. In particular, we proposed a hybrid network that takes advantage of spatial CNNs, temporal LSTMs, and their fusions to achieve multi-label AU detection. To address the highly-skewed nature between AU classes, we proposed two multi-label sampling strategies: multi-label stratification, and multi-label minority oversampling majority undersampling (MOMU). Furthermore, for the first time, we showed how machines see facial AUs using a visualization approached based on gradient ascend. Experiments on two of the largest spontaneous AU datasets demonstrate that the proposed network outperformed a standard CNN and feature-based state-of-the-art methods with respect to AU occurrence and intensity in and between BP4D+ and GFT databases (size = 0.6 million annotated frames).

An unsupervised framework for common event discovery in human interaction

Chapter 5 introduced a relatively unexplored problem—Common Event Discovery (CED)—to address the limitations of acquiring expert annotations in AFA systems. Unlike an exhaustive approach that searches over all possible solutions, we propose an efficient branch-and-bound (B&B) framework that yields both efficient search and an optimal solution with theoretical guarantee. The B&B framework takes as input any multidimensional signal that can be quantified into histograms, and can be readily applied to discover events at the same or different times (synchrony and event

commonality, respectively). We also considered extensions to video search and supervised event detection. The effectiveness of the B&B framework is evaluated in motion capture of deliberate behavior and in video of spontaneous facial behavior in diverse interpersonal contexts: interviews [10], small groups of young adults [93], and parent-infant face-to-face interaction [170].

6.2 Directions for Future Work

Personalized models

Chapter 3 has demonstrated benefits of our transductive approach over generic models in two AFA scenarios, including holistic expression recognition and facial AU detection. Potential extensions of STM including its combination of other classifiers with convex decision functions and losses, such as logistic regression. For instance, Yang *et al.* [271] carried out a personalization approach for intensity estimation using Conditional Ordinal Random Fields. More recently, Zhang *et al.* [279] extended STM framework with “single-machine” multi-class classification for action recognition. Finally, while this study focuses evaluations on facial expressions, STM could be applied to other fields where subject- or object-specific biases are involved, *e.g.*, intensity estimation [271], object or activity recognition [279]. Directions to improve our personalized model include:

1. For classifiers with non-convex decision functions, such as random forest, local minimum could cause worse performance. Applying the transductive idea to non-convex models could be one challenge to explore.
2. Improving STM’s training speed could be another direction due to the complexity of solving s. Given large training datasets, finding s can become computationally prohibitive. Exploring sample-based approximation during the re-weighting step (*e.g.*, [22]) could be one direction to improve efficiency.
3. As analyzed in Chapter 4, the errors caused by identity factors can be traced back to feature representation. If an optimal feature can be learned for the classification task, we believe the burden of designing a sophisticated model could be largely reduced. In other words, one may need not to personalize the model if the features can precisely describe the human affect of interest.

Deep learning models

Chapter 4 showed advantages of learning a deep model for jointly addressing multiple factors, including learning an optimal representation for classification, modeling correlation between AUs, temporal consistency, and class imbalance. We believe such end-to-end framework is (and will likely continue to be) the best tool to embody the richness, ambiguity, and dynamic nature of facial actions. This is mainly due to the highly non-linearity nature and stochastic training inherent in deep models. The whole human brain contains 86 billion neurons (or 19/23 billion for female/male in the cerebral cortex). When more computational power and data are available, it is possible that a deep model can accommodate billions of parameters and trained like a human. There is still much work to be done to meet real-world settings. Future directions include:

1. We believe deeper investigation and analysis of the proposed hybrid network will continue to provide more insights about what the model is learning and enable better training strategies. For example, visualization of the temporal LSTM model can offer evidence to answer how facial actions evolve over time, or whether or not a temporal pattern of facial actions exists

- among human despite their identities. Using attention models can also help analyze which facial regions contribution to recognize a particular AU.
2. Bi-directional LSTMs can be used to extend learning of dynamics both forward and backward along time axis. Although this might sounds counter-intuitive at first glance (humans do not tell an action by reversing time axis), we do observe human coders benefit from this reverse information to locate an AU onset, and further confirm an AU occurrence [65, 66].
 3. While improvements were observed in a within-dataset scenario, deep models still suffer in the between-dataset experiments. This is because the model was only tuned to fit one particular datasets (domain). When covariate shift exists between datasets (sample distribution mismatches due to changes in appearance, recording environment, behavior, etc.), there is no way for the trained model to adjust for a new domain. One potential solution could be learning “transferable features” by jointly considering the distributions of samples from both domains, *e.g.*, Deep Adaptation Network [159].
 4. Collecting data and ensuring reliable annotations is error-prone and expensive, in terms of time and actual costs for human labor. In addition, the time could be spent at very low efficiency due to high redundancy in the annotation process, *e.g.*, annotating the same group of people with similar race, culture, recording environment, interview context, etc. We believe a potential direction could be weakly-supervised techniques that require only weak labels (*e.g.*, a temporal window of “smile” instead of per-frame AU annotation), or adaptation from pre-trained classifier (*e.g.*, L-STM in Chapter 3). Another direction is synthesis-based methods that generate photo-realistic facial images with controlled parameters, such as different facial action units, illumination, gender, age, head pose, etc.
 5. Intensity estimation is definitely another direction. Annotations on intensity are relatively rare compared to frame occurrence. Training a deep model solely on nowadays available datasets (*e.g.*, DISFA [167]) could be undesirable due to the limited number of training samples. The author believes that having a model that takes annotations of both frame occurrence and intensity could achieve more reliable results for intensity estimation.

Unsupervised models

Chapter 5 presented a new CED problem to discover facial actions using interactive information. This is inspired by the observation that human facial actions are mostly driven by interactive partners, such as an interview or social interaction scenarios used for dataset collection [10, 50, 282]. An extension to our B&B framework is to promote the scalability of the proposed CED algorithm. Given current pairwise design, the computational complexity grows quadratically with the number of input sequences. Parallelism can be one solution, *i.e.*, compute pairwise bounds independently using clusters or multi-threading, and then aggregate these bounds into a overall score. As CED opens a door to a relatively unexplored field, in fact the authors have received feedbacks regarding a number of interesting directions:

1. One extension can be studying the discovered common events to different interaction scenarios. For instance, parents with 6-month and older infants are often maximally engaged in their positive affects (*i.e.*, smiles) [170]. Or, for social interaction among young adults [50], alcoholic effects could improve group conversations with more frequent smiles. With our B&B as an automated tool, the statistics of such common events can be studied as an evidence for different interactive domains.

2. Another extension is with regards to video search. As visuals carry much richer information than plain texts, searching videos *using videos* can be more efficient than searching by texts. Say one is watching YouTube and interested in a particular video clip of an NBA game. If they can search other videos related to this particular clip by simply clicking on the playback bar to the start and end frames of the video clip, wouldn't that be cool?
3. Instead of finding "common events", we found another form of human interaction, causality-effect relationship, is also of significant interest. For example, a teacher asks a question in the class, then a student raises his hand to answer the question. This "teacher-student" pair of interaction makes the discovery problem even more challenging due to different features and larger search space.
4. We have complained about "hand-crafted features". However, most supervised approaches also suffer from "hand-crafted labels," which are may or may not well-defined for specific problems. For example, the AUs used in this study are only a subset of frequent ones that humans can express. If there exists AUs that were undefined in FACS or unannotated by the coders, the trained classifiers would never be able to detect such AUs. One possibility is to use the discovered common events to assist learning the AUs, or to *learn* such atomic units of facial expression using generic models.



Appendices

STM Derivation

A.1 Linear Penalized SVMs

Denote the training set as $\mathcal{D}^{\text{tr}} = \{\mathbf{x}_i, y_i\}_{i=1}^{n_{\text{tr}}}$, $y_i \in \{+1, -1\}$ (see notation¹). For notational simplicity, we denote $\mathbf{X} \equiv \mathbf{X}^{\text{tr}}$ in the following unless further referred. Recall that the objective for unconstrained penalized linear SVM in Sec. 3 of the manuscript:

$$R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n_{\text{tr}}} s_i L^p(y_i, \mathbf{w}^\top \mathbf{x}_i), \quad (\text{A.1})$$

where $L^p(y, \cdot) = \max(0, 1 - y \cdot)^p \in \mathbb{R}^{n_{\text{tr}}}$ is the training loss. $p = 2$ for quadratic loss and $p = 1$ stands for hinge loss.

A.1.1 Quadratic loss

Expanding Eq. (A.1) with $p = 2$, we have:

$$\begin{aligned} R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{n_{\text{tr}}} s_i \max(0, 1 - \mathbf{w}^\top \mathbf{x}_i)^2 \\ &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in sv} s_i (1 - 2y_i \mathbf{w}^\top \mathbf{x}_i + \mathbf{x}_i^\top \mathbf{w} \mathbf{w}^\top \mathbf{x}_i). \end{aligned}$$

We obtain the gradient by deriving $\frac{\partial R_{\mathbf{w}}}{\partial \mathbf{w}}$:

$$\begin{aligned} \nabla_{\mathbf{w}} &= \mathbf{w} + C \sum_{i \in sv} s_i (-2y_i \mathbf{x}_i + 2(\mathbf{w}^\top \mathbf{x}_i) \mathbf{x}_i) \\ &= \mathbf{w} + 2C \sum_{i \in sv} s_i (\mathbf{w}^\top \mathbf{x}_i - y_i) \mathbf{x}_i \\ &= \mathbf{w} + 2C \mathbf{X} \mathbf{S} \mathbf{I}^0 (\mathbf{X}^\top \mathbf{w} - \mathbf{y}) \end{aligned} \quad (\text{A.2})$$

¹ Bold capital letters denote a matrix \mathbf{X} ; bold lower-case letters denote a column vector \mathbf{x} . \mathbf{x}_i represents the i th column of the matrix \mathbf{X} . All non-bold letters represent scalars. x_j denotes the scalar in the j th element of \mathbf{x} . $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is an identity matrix.

where sv denotes the index set of support vectors. Computing $\frac{\partial \nabla_{\mathbf{w}}}{\partial \mathbf{w}}$, we drive the Hessian:

$$\begin{aligned}\mathbf{H}_{\mathbf{w}} &= \mathbf{I}_d + 2C \sum_{i \in sv} s_i \mathbf{x}_i \mathbf{x}_i^\top \\ &= \mathbf{I}_d + 2C \mathbf{X} \mathbf{S} \mathbf{I}^0 \mathbf{X}^\top,\end{aligned}\tag{A.3}$$

where \mathbf{I}_d denotes a $d \times d$ identity matrix.

A.1.2 Huber loss

Recall the Huber loss [27] is defined in three parts:

$$L_H(a) = \begin{cases} 0 & \text{if } a > 1 + h, \\ \frac{(1+h-a)^2}{4h} & \text{if } |1-a| \leq h, \\ 1-a & \text{otherwise,} \end{cases}\tag{A.4}$$

where h is a tradeoff parameter. When $h \rightarrow 0$, L_H approaches the hinge loss. We discuss the following by categorizing the points into 3 parts, *i.e.*, linear, quadratic, and zero parts, according to the Huber loss.

Part 1 (linear part): Let \mathcal{P}_ℓ be the index set containing n_ℓ points in the linear part of the Huber loss, *i.e.*, $y_i \mathbf{w}^\top \mathbf{x}_i < 1 - h, \forall i \in \mathcal{P}_\ell$. Replacing the L^1 loss with the linear part of Huber loss in Eq. (A.4), the objective becomes:

$$R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in \mathcal{P}_\ell} s_i (1 - y_i \mathbf{w}^\top \mathbf{x}_i).$$

The gradient and Hessian can be derived as:

$$\begin{aligned}\nabla_{\mathbf{w}} &= \mathbf{w} + C \sum_{i \in \mathcal{P}_\ell} s_i (-y_i \mathbf{x}_i) \\ &= \mathbf{w} - C \mathbf{X} \mathbf{S} \mathbf{I}^1 \mathbf{y},\end{aligned}\tag{A.5}$$

$$\mathbf{H}_\beta = \mathbf{I}_d,\tag{A.6}$$

where $\mathbf{I}^1 \in \mathbb{R}^{n_{\text{tr}} \times n_{\text{tr}}}$ denotes the proximity identity matrix with the first n_{sv} diagonal elements being 0, followed by n_ℓ 1 elements, and the rest being 0.

Part 2 (quadratic part): Let \mathcal{P}_q be the index set containing points in the quadratic part of the Huber loss, *i.e.*, $|y_j \mathbf{w}^\top \mathbf{x}_j - 1| \leq h, \forall j \in \mathcal{P}_q$. Replacing the L^1 loss with the quadratic part of Huber loss, the objective becomes:

$$\begin{aligned}R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in \mathcal{P}_q} s_i \frac{(1 + h - y_i \mathbf{w}^\top \mathbf{x}_i)^2}{4h} \\ &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in \mathcal{P}_q} \frac{s_i}{4h} [(1 + h)^2 \\ &\quad - 2(1 + h) y_i \mathbf{w}^\top \mathbf{x}_i + (y_i \mathbf{w}^\top \mathbf{x}_i)^\top (y_i \mathbf{w}^\top \mathbf{x}_i)].\end{aligned}$$

The gradient and Hessian can be derived as:

$$\begin{aligned}\nabla_{\mathbf{w}} &= \mathbf{w} + C \sum_{i \in \mathcal{P}_q} \frac{s_i}{4h} [-2(1+h)y_i \mathbf{x}_i + 2\mathbf{x}_i^\top \mathbf{x}_i \mathbf{w}] \\ &= \mathbf{w} + \frac{C}{2h} [\mathbf{XSI}^0 \mathbf{X}^\top \mathbf{w} - (1+h)\mathbf{XS}\mathbf{y}],\end{aligned}\tag{A.7}$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + \frac{C}{2h} (\mathbf{XSI}^0 \mathbf{X}^\top).\tag{A.8}$$

Part 3 (zero part): Let \mathcal{P}_z be the index set containing n_z points in the zero part of the Huber loss, i.e., $y_k \mathbf{w}^\top \mathbf{x}_i > 1 + h, \forall k \in \mathcal{P}_z$. Replacing the L^1 loss with the zero part of Huber loss, the objective becomes:

$$R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \|\mathbf{w}\|^2.$$

The gradient and Hessian are simply derived as:

$$\nabla_{\mathbf{w}} = \mathbf{w},\tag{A.9}$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d.\tag{A.10}$$

Summarizing Eqs. (A.5)~(A.10) from the above three parts, we obtain the gradient and Hessian for the linear SVM with Huber loss:

$$\nabla_{\mathbf{w}} = \mathbf{w} + \frac{C}{2h} \mathbf{XSI}^0 [\mathbf{X}^\top \mathbf{w} - (1+h)\mathbf{y}] - C\mathbf{XSI}^1 \mathbf{y},\tag{A.11}$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + \frac{C}{2h} \mathbf{XSI}^0 \mathbf{X}^\top.\tag{A.12}$$

A.2 Nonlinear Penalized SVMs

Using the representer theorem, we seek a solution in the form $f(\cdot) = \sum_{i=1}^{n_{\text{tr}}} \beta_i k(\mathbf{x}_i, \cdot)$, $\beta_i \in \mathbb{R}$. Using this expression in Eq. (3.2), we reach the objective for the unconstrained nonlinear SVM (Sec. 3 of the main paper):

$$\begin{aligned}R_{\beta}(\mathcal{D}^{\text{tr}}, \mathbf{s}) &= \sum_{i,j=1}^{n_{\text{tr}}} \beta_i \beta_j k(\mathbf{x}_i, \mathbf{x}_j) + C \sum_{i=1}^{n_{\text{tr}}} s_i L^p(y_i, \sum_{j=1}^{n_{\text{tr}}} \beta_j k(\mathbf{x}_i, \mathbf{x}_j)) \\ &= \frac{1}{2} \boldsymbol{\beta}^\top \mathbf{K} \boldsymbol{\beta} + C \sum_{i=1}^{n_{\text{tr}}} s_i L^p(y_i, \mathbf{K}_i^\top \boldsymbol{\beta}),\end{aligned}\tag{A.13}$$

where $\boldsymbol{\beta} = [\beta_1, \dots, \beta_{n_{\text{tr}}}]^\top \in \mathbb{R}^{n_{\text{tr}}}$.

A.2.1 Quadratic loss

Expanding Eq. (A.1) with $p = 2$, we have:

$$\begin{aligned} R_{\beta}(\mathcal{D}^{\text{tr}}, \mathbf{s}) &= \frac{1}{2} \beta^{\top} \mathbf{K} \beta + C \sum_{i=1}^{n_{\text{tr}}} s_i \max(0, 1 - \mathbf{K}_i^{\top} \beta)^2 \\ &= \frac{1}{2} \beta^{\top} \mathbf{K} \beta + C \sum_{i \in sv} s_i (1 - y_i \mathbf{K}_i^{\top} \beta)^2. \end{aligned} \quad (\text{A.14})$$

Computing $\frac{\partial R_{\beta}}{\partial \beta}$, we drive the gradient:

$$\begin{aligned} \nabla_{\beta} &= \mathbf{K} \beta + 2C \sum_{i \in sv} s_i (2y_i \mathbf{K}_i (y_i \mathbf{K}_i^{\top} \beta - 1)) \\ &= \mathbf{K} \beta + 2C \mathbf{K} \mathbf{S} \mathbf{I}^0 (\mathbf{K} \beta - \mathbf{y}), \end{aligned} \quad (\text{A.15})$$

where $\mathbf{S} = \text{diag}(s_1, \dots, s_{n_{\text{tr}}}) \in \mathbb{R}^{n_{\text{tr}} \times n_{\text{tr}}}$ the diagonal re-weighting matrix, $\mathbf{y} \in \mathbb{R}^n$ the label vector, and $\mathbf{I}^0 \in \mathbb{R}^{n_{\text{tr}} \times n_{\text{tr}}}$ the proximity identity matrix with the first n_{sv} diagonal elements being 1 and the rest being 0. Similarly, we have the Hessian $\frac{\partial \nabla_{\beta}}{\partial \beta}$:

$$\mathbf{H}_{\beta} = \mathbf{K} + 2C \mathbf{K} \mathbf{S} \mathbf{I}^0 \mathbf{K}. \quad (\text{A.16})$$

A.2.2 Huber loss

We discuss the following by categorizing the points into 3 parts, *i.e.*, linear, quadratic, and zero parts, according to the Huber loss:

Part 1 (linear part): Let \mathcal{P}_{ℓ} be the index set containing n_{ℓ} points in the linear part of the Huber loss, *i.e.*, $y_i \mathbf{K}_i^{\top} \beta < 1 - h, \forall i \in \mathcal{P}_{\ell}$. Replacing the L^1 loss with the linear part of Huber loss, the objective becomes:

$$R_{\beta}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \beta^{\top} \mathbf{K} \beta + C \sum_{i \in \mathcal{P}_{\ell}} s_i (1 - y_i \mathbf{K}_i^{\top} \beta).$$

The gradient and Hessian can be derived as:

$$\begin{aligned} \nabla_{\beta} &= \mathbf{K} \beta + C \sum_{i \in \mathcal{P}_{\ell}} s_i (-y_i \mathbf{K}_i) \\ &= \mathbf{K} \beta - C \mathbf{K} \mathbf{S} \mathbf{I}^1 \mathbf{y}, \end{aligned} \quad (\text{A.17})$$

$$\mathbf{H}_{\beta} = \mathbf{K}, \quad (\text{A.18})$$

where $\mathbf{I}^1 \in \mathbb{R}^{n_{\text{tr}} \times n_{\text{tr}}}$ denotes the proximity identity matrix with the first n_{sv} diagonal elements being 0, followed by n_{ℓ} 1 elements, and the rest being 0.

Part 2 (quadratic part): Let \mathcal{P}_q be the index set containing points in the quadratic part of the Huber loss, *i.e.*, $|y_j \mathbf{K}_j^{\top} \beta - 1| \leq h, \forall j \in \mathcal{P}_q$. Replacing the L^1 loss with the quadratic part of Huber

loss, the objective becomes:

$$\begin{aligned} R_{\beta}(\mathcal{D}^{\text{tr}}, \mathbf{s}) &= \frac{1}{2} \beta^{\top} \mathbf{K} \beta + C \sum_{i \in \mathcal{P}_q} s_i \frac{(1 + h - y_i \mathbf{K}_i^{\top} \beta)^2}{4h} \\ &= \frac{1}{2} \beta^{\top} \mathbf{K} \beta + C \sum_{i \in \mathcal{P}_q} \frac{s_i}{4h} [(1 + h)^2 \\ &\quad - 2(1 + h) y_i \mathbf{K}_i^{\top} \beta + (y_i \mathbf{K}_i^{\top} \beta)^{\top} (y_i \mathbf{K}_i^{\top} \beta)]. \end{aligned}$$

The gradient and Hessian can be derived as:

$$\begin{aligned} \nabla_{\beta} &= \mathbf{K} \beta + C \sum_{i \in \mathcal{P}_q} \frac{s_i}{4h} [-2(1 + h) y_i \mathbf{K}_i + 2 y_i \mathbf{K}_i^{\top} \beta y_i \mathbf{K}_i^{\top}] \\ &= \mathbf{K} \beta + \frac{C}{2h} [\mathbf{K} \mathbf{S} \mathbf{I}^0 \mathbf{K} \beta - (1 + h) \mathbf{K} \mathbf{S} \mathbf{y}], \end{aligned} \quad (\text{A.19})$$

$$\mathbf{H}_{\beta} = \mathbf{K} + \frac{C}{2h} (\mathbf{K} \mathbf{S} \mathbf{I}^0 \mathbf{K}). \quad (\text{A.20})$$

Part 3 (zero part): Let \mathcal{P}_z be the index set containing points in the zero part of the Huber loss, i.e., $y_k \mathbf{K}_k^{\top} \beta > 1 + h, \forall k \in \mathcal{P}_z$. Replacing the L^1 loss with the zero part of Huber loss, the objective becomes:

$$R_{\beta}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \beta^{\top} \mathbf{K} \beta.$$

The gradient and Hessian are simply derived as:

$$\nabla_{\beta} = \mathbf{K} \beta, \quad (\text{A.21})$$

$$\mathbf{H}_{\beta} = \mathbf{K}. \quad (\text{A.22})$$

Summarizing Eqs. (A.17)~(A.22) from the above three parts, we obtain the gradient and Hessian for the nonlinear SVM with Huber loss:

$$\nabla_{\beta} = \mathbf{K} \beta + \frac{C}{2h} \mathbf{K} \mathbf{S} \mathbf{I}^0 [\mathbf{K} \beta - (1 + h) \mathbf{y}] - C \mathbf{K} \mathbf{S} \mathbf{I}^1 \mathbf{y}, \quad (\text{A.23})$$

$$\mathbf{H}_{\beta} = \mathbf{K} + \frac{C}{2h} \mathbf{K} \mathbf{S} \mathbf{I}^0 \mathbf{K}. \quad (\text{A.24})$$

A.3 Adapt with Target Labels

Denote the labelled target data and their labels as $\mathcal{D}^L = \{\mathbf{x}_j^L, y_j^L\}_{j=1}^L, y_j^L \in \{+1, -1\}, 0 \leq n_L \leq n_{\text{te}}$. We introduce an additional regularization term $\Omega_L(\mathcal{D}^L) = \lambda_L \sum_{j=1}^{n_L} L^q(y_j^L, f(\mathbf{x}_j^L))$ to the original STM formulation:

$$\min_{f, \mathbf{s}} R_f(\mathcal{D}^{\text{tr}}, \mathbf{s}) + \lambda \Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}) + \lambda_L \Omega_L(\mathcal{D}^L), \quad (\text{A.25})$$

where $\lambda_L > 0$ is a tradeoff parameter. We show in the following the derivation of modelling $\Omega_L(\mathcal{D}^L)$ as L^2 ($q = 2$) on a linear SVM. Other loss functions and nonlinear SVMs could derive similarly as shown above.

Considering together the linear SVM risk function and the new regularization, we have:

$$\begin{aligned}
 R_{\mathbf{w}}(\mathcal{D}^{\text{tr}}, \mathbf{s}) + \lambda_L \Omega_L(\mathcal{D}^L) \\
 &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i \in sv} s_i (1 - 2y_i \mathbf{w}^\top \mathbf{x}_i + \mathbf{x}_i^\top \mathbf{w} \mathbf{w}^\top \mathbf{x}_i) \\
 &\quad + \lambda_L \sum_{i=1}^{n_L} (1 - 2y_i^L \mathbf{w}^\top \mathbf{x}_i^L + \mathbf{x}_i^{L\top} \mathbf{w} \mathbf{w}^\top \mathbf{x}_i^L).
 \end{aligned} \tag{A.26}$$

We obtain the gradient by deriving $\frac{\partial R_{\mathbf{w}}}{\partial \mathbf{w}}$:

$$\begin{aligned}
 \nabla_{\mathbf{w}} &= \mathbf{w} + 2C \sum_{i \in sv} s_i (\mathbf{w}^\top \mathbf{x}_i - y_i) \mathbf{x}_i + \lambda_L \sum_{i=1}^{n_L} (\mathbf{w}^\top \mathbf{x}_i^L - y_i^L) \mathbf{x}_i^L \\
 &= \mathbf{w} + 2C \mathbf{X} \mathbf{S} \mathbf{I}^0 (\mathbf{X}^\top \mathbf{w} - \mathbf{y}) + \lambda_L \mathbf{X}^L (\mathbf{X}^{L\top} \mathbf{w} - \mathbf{y}^L) \\
 &= \mathbf{w} + \underbrace{[\mathbf{X} | \mathbf{X}^L]}_{\hat{\mathbf{X}}} \underbrace{\begin{bmatrix} 2C \mathbf{S} \mathbf{I}^0 & 0 \\ 0 & \lambda_L \mathbf{I}_{n_L} \end{bmatrix}}_{\hat{\mathbf{S}}} \begin{bmatrix} (\mathbf{X}^\top \mathbf{w} - \mathbf{y}) \\ (\mathbf{X}^{L\top} \mathbf{w} - \mathbf{y}^L) \end{bmatrix} \\
 &= \mathbf{w} + \hat{\mathbf{X}} \hat{\mathbf{S}} (\hat{\mathbf{X}}^\top \mathbf{w} - \hat{\mathbf{y}})
 \end{aligned} \tag{A.27}$$

where $\hat{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{y}^L \end{bmatrix}$. Computing $\frac{\partial \nabla_{\mathbf{w}}}{\partial \mathbf{w}}$, we drive the Hessian:

$$\begin{aligned}
 \mathbf{H}_{\mathbf{w}} &= \mathbf{I}_d + 2C \sum_{i \in sv} s_i \mathbf{x}_i \mathbf{x}_i^\top + \lambda_L \sum_{i=1}^{n_L} \mathbf{x}_i^L \mathbf{x}_i^{L\top} \\
 &= \mathbf{I}_d + 2C \mathbf{X} \mathbf{S} \mathbf{I}^0 \mathbf{X}^\top + \lambda \mathbf{X}^L \mathbf{X}^{L\top} \\
 &= \mathbf{I}_d + \underbrace{[\mathbf{X} | \mathbf{X}^L]}_{\hat{\mathbf{X}}} \underbrace{\begin{bmatrix} 2C \mathbf{S} \mathbf{I}^0 & 0 \\ 0 & \lambda_L \mathbf{I}_{n_L} \end{bmatrix}}_{\hat{\mathbf{S}}} \begin{bmatrix} \mathbf{X}^\top \\ \mathbf{X}^{L\top} \end{bmatrix} \\
 &= \mathbf{I}_d + \hat{\mathbf{X}} \hat{\mathbf{S}} \hat{\mathbf{X}}^\top,
 \end{aligned} \tag{A.28}$$

where \mathbf{I}_d denotes a $d \times d$ identity matrix. Using the derived gradient and Hessian, one can consider that solving problem (A.25) is equivalent to solving the original STM using augmented training set with weighted labeled target data \mathbf{X}^L .

A.4 Cross-subject experiment on GEMEP-FERA dataset

Table A.1 shows the results of STM compared to alternative methods, including baseline linear SVM, Kernel Mean Matching (KMM), Transductive SVM (T-SVM), Domain-Adaptation SVM (DA-SVM), Subspace Alignment, Domain Adaptation Machine (DAM), and STM (our method). Please refer to detailed settings in Chapter 3.

■

Table A.1: Cross-subject AU detection on GEMEP-FERA dataset. “N|S” denotes SA with either nearest-neighbor (N) or SVM (S).

AU	AUC								F1 Score							
	SVM	KMM	T-SVM	DA-SVM	SA (N S)	DAM	STM		SVM	KMM	T-SVM	DA-SVM	SA (N S)	DAM	STM	
1	71.5	43.3	72.2	83.3	21.5 53.1	80.6	84.3		56.5	48.5	60.3	59.1	30.3 0.0	36.2	68.1	
2	73.9	51.0	74.3	76.8	20.2 57.3	78.1	73.3		56.9	50.2	58.5	57.1	27.9 0.6	31.7	65.5	
4	58.5	53.5	42.8	66.6	12.3 52.2	58.6	60.0		43.5	39.8	36.9	46.3	16.1 0.1	5.8	43.3	
6	80.4	60.2	81.1	91.1	14.7 52.6	83.2	87.7		63.7	58.7	63.8	72.7	20.5 0.8	49.7	71.6	
7	66.9	59.4	70.8	76.9	17.8 48.5	77.2	75.4		63.1	63.5	63.7	68.3	27.8 0.0	34.0	66.2	
12	77.7	58.8	74.8	74.5	25.3 53.4	85.8	84.7		79.1	68.4	77.6	75.5	49.7 25.1	74.5	82.1	
15	55.5	58.7	67.2	67.5	12.6 52.4	75.2	67.8		33.4	35.2	35.2	41.3	9.4 0.1	6.5	39.3	
17	59.8	51.8	63.8	66.5	7.4 43.6	70.3	63.3		32.0	27.8	36.2	42.0	9.1 0.2	19.8	35.9	
Av.	68.0	54.6	68.4	75.4	16.5 51.7	76.1	74.5		53.5	49.0	54.0	57.8	23.9 3.4	32.3	59.0	

Evaluate CNNs on Baby-FACS

We performed preliminary experiments of evaluating the CNN models on Baby-FACS (facial action coding system for infants and young children [184]). We followed the exact parameter settings as described in Chapter 4. Below we briefly summarize the results and some discoveries.

B.1 Miami Modeling Dataset (6-month)

Thanks to the Miami Modeling group [170], we used 6-month infants for training and test purpose. Fig. B.1 shows some illustration of an interaction between a mother and her infant. The copy of dataset that we used contains 61 ethnically diverse infants, each of which were recorded with 3 consecutive sessions, including face-to-face (FF), still face (SF), and reunion (RE). FF involves the parents and infants playing normally for 2 minutes, and then followed by an SF session which the parents remain unresponsive. RE comes last as another 2 minutes for which the parents and infants resume normal behavior. There were 448,692 validly tracked frames in our experiments. As usual, we followed a 10-fold subject-exclusive data partition for the experiments: 8 folds for training, 1 fold for validation and 1 for test. For evaluating our multi-label network, we manually picked three most frequently occurring faces on infants: neutral face (AU 0), cry face (AUs 4+20), and smile

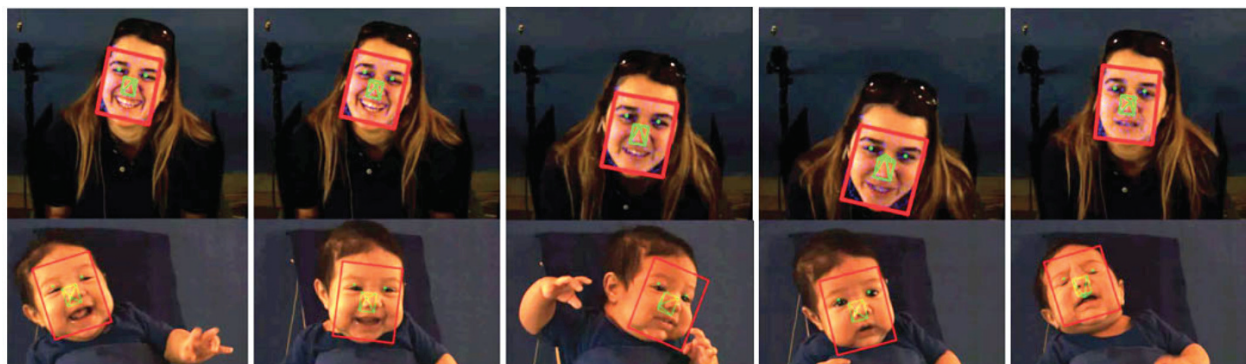


Figure B.1: Illustration of the interaction between a mother and her infant, as well as some face tracking results on the face area (red boxes) and detected head pose (green pyramids). (image credit from [109])

Table B.1: The descriptive rates of the Miami Modeling dataset in terms of different sessions and faces of interest. #occ indicates the number of frame occurrence, #total indicates total number of frames in a particular session, and rate indicates the percentage of descriptive rates.

	Neutral face			Cry face			Smile face		
Session	#occ	#total	rate	#occ	#total	rate	#occ	#total	rate
FF	77741	130025	59.8%	2842	130025	2.19%	50847	130025	39.11%
SF	48975	63622	77.0%	5524	63622	8.68%	8632	63622	13.57%
RE	67487	110804	60.9%	7336	110804	6.62%	37648	110804	33.98%

(AU 6+12). Table B.1 shows the descriptive rates of different sessions and faces of interest. As can be observed, the descriptive rates of neutral face and cry face are higher in SF than in FF and RE when the parents remain irresponsive. On the contrary, the rate of smile face is significantly higher in FF and RE, when the infants and parents played in a normal routine.

Table B.2 shows the performance in terms of ACC, PA and NA (see detailed descriptions about metrics in Sec. 2.6). Overall, the performance of detecting both neutral face and smile face were quite consistent across three sessions, resulting in accuracy of over 75% for the neutral face and over 73% for the smile face. We noticed a coherence between the PA metric and the descriptive rate, where lower rate could possibly reduce the PA metric due to the imbalance distribution between positive and negative samples. The cry face was relatively difficult to detect due to the large head motion and less texture on infant faces. For the FF session, due to a very low $\sim 2\%$ rate, the accuracy was only 34%, which was about half the accuracy of another two sessions. In general, PA of the cry face was around .4, which was lower than smile and neutral faces. For cry and smile faces where the distributions were biased toward negative samples, the NA metric remained rather higher: over .95 for cry face and over .82 for smile face. This is because of the global loss function that the CNN was trained on. Contributions to the loss function can be easily dominated by the larger population of negative samples. We believe a further balancing on the positive/negative ratio would help improve the PA metric, but not necessarily improve ACC.

B.2 CLOCK Dataset

The CLOCK dataset is a not yet publicly available collection of infants. We used a total of 86 infants and a 80-20 subject-exclusive partition: 61 infants ($\sim 70\%$) for training, 7 infants ($\sim 10\%$)

Table B.2: The performance evaluation on the Miami Modeling dataset [170] in terms of ACC, PA and NA for different sessions and faces of interest.

	Neutral face			Cry face			Smile face		
Session	ACC	PA	NA	ACC	PA	NA	ACC	PA	NA
FF	0.82	0.81	0.71	0.34	0.18	0.97	0.79	0.66	0.82
SF	0.75	0.88	0.55	0.62	0.44	0.96	0.73	0.44	0.92
RE	0.78	0.79	0.67	0.74	0.41	0.95	0.73	0.62	0.83

Table B.3: The performance evaluation on the CLOCK dataset in terms of PA, NA, S-score and AUC for individual AUs.

AU (base rate %)	PA	NA	S	AUC
1 (27.2%)	0.48	0.94	0.77	0.78
2 (22.1%)	0.33	0.94	0.77	0.73
3 (23.0%)	0.5	0.91	0.69	0.78
4 (11.7%)	0.19	0.96	0.84	0.74
6 (30.9%)	0.76	0.91	0.74	0.92
9 (7.0%)	0.26	0.98	0.93	0.77
12 (20.2%)	0.64	0.93	0.77	0.92
20 (18.4%)	0.48	0.92	0.72	0.82
28 (7.7%)	0.25	0.95	0.83	0.72
avg	0.43	0.94	0.78	0.8

for validation, and 18 infants ($\sim 20\%$) for test. For evaluating our multi-label architecture, we manually selected 9 AUs: 1, 2, 3, 4, 6, 9, 12, 20 and 28. In total, there were 158,869 frames that was validly tracked and FACS-coded. This amounts to about $\sim 40\%$ of the total number of frames in the entire dataset.

Table B.3 shows the performance on the CLOCK dataset in terms of four metrics. We would also like to highlight that the reliability between human coders in terms of S-score is: AU1=0.61, AU2=0.67, AU4=0.68, AU6=0.79, AU12=0.84, AU20=0.74, and AU28=0.86. This reliability could potentially serve as a “gold standard” on how the best visual system (*i.e.*, humans) would agree on each other. Compared to the S-scores reported in Table B.3, we found some AUs achieved higher S-scores than the reliability between humans, showing a potential higher agreement between the trained classifiers and the FACS codes for training than between the human coders. More specifically, such high agreement involves AUs 1, 2 and 4, which can be difficult to detect due to the lack of textures on infant eyebrows. As indicated in human reliability measures, these AUs often suffer from agreements even between human coders. The authors are preparing a more in-depth analysis on such factors, which will be likely to appear in a future publication. Please stay tuned!

■

CED Derivation

C.1 Bound derivations for symmetrized KL divergence

We start by resembling the ℓ_1 bound in [46], and then bound derivations of symmetrized KL divergence. Recall the definition $\underline{h}_k^i = \frac{h_k^{i-}}{|\mathbf{S}^{i+}|}$ and $\overline{h}_k^i = \frac{h_k^{i+}}{|\mathbf{S}^{i-}|}$ and property (c) $0 \leq \underline{h}_k^i \leq \widehat{h}_k^i \leq \overline{h}_k^i$ in Sec. 3.2 of the main paper. Given two sequences $(\mathbf{S}^i, \mathbf{S}^j)$, we denote the ℓ_1 bound in [46] as:

$$\begin{aligned} l_{\ell_1}(\mathbf{R}) &= \sum_k l_k \\ &\leq \ell_1(\widehat{\mathbf{h}}_k^i, \widehat{\mathbf{h}}_k^j) = \sum_k |\widehat{h}_k^i - \widehat{h}_k^j| \\ &\leq \sum_k u_k = u_{\ell_1}(\mathbf{R}), \end{aligned} \tag{C.1}$$

where

$$\begin{aligned} l_k &= \max(\underline{h}_k^i, \underline{h}_k^j) - \min(\overline{h}_k^i, \overline{h}_k^j), \\ u_k &= \max(\overline{h}_k^i, \overline{h}_k^j) - \min(\underline{h}_k^i, \underline{h}_k^j). \end{aligned} \tag{C.2}$$

Following the same procedure, we obtain the bound for log ℓ_1 -distance:

$$\begin{aligned} l_{\ln \ell_1}(\mathbf{R}) &= \sum_k l_k^{\ln} \\ &\leq \sum_k |\ln \widehat{h}_k^i - \ln \widehat{h}_k^j| \\ &\leq \sum_k u_k^{\ln} = u_{\ln \ell_1}(\mathbf{R}), \end{aligned} \tag{C.3}$$

where

$$\begin{aligned} l_k^{\ln} &= \max(\ln \underline{h}_k^i, \ln \underline{h}_k^j) - \min(\ln \overline{h}_k^i, \ln \overline{h}_k^j) \\ u_k^{\ln} &= \max(\ln \overline{h}_k^i, \ln \overline{h}_k^j) - \min(\ln \underline{h}_k^i, \ln \underline{h}_k^j). \end{aligned} \tag{C.4}$$

Recall the definition of symmetrized KL divergence in Eq. (3) of the main paper:

$$D(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) = \sum_k (\hat{h}_k^i - \hat{h}_k^j)(\ln \hat{h}_k^i - \ln \hat{h}_k^j). \quad (\text{C.5})$$

Observe that $(\hat{h}_k^i - \hat{h}_k^j)$ and $(\ln \hat{h}_k^i - \ln \hat{h}_k^j)$ always share the same sign, we can rewrite the definition into:

$$D(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) = \sum_k |\hat{h}_k^i - \hat{h}_k^j| |\ln \hat{h}_k^i - \ln \hat{h}_k^j|. \quad (\text{C.6})$$

Using Eqs. (C.1), (C.3) and the fact that $0 \leq D(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j)$, we obtain the bounds as:

$$\begin{aligned} l_D(\mathbf{R}) &= \sum_k (l_k)_+ (l_k^{\text{ln}})_+ \\ &\leq D(\hat{\mathbf{h}}^i, \hat{\mathbf{h}}^j) \\ &\leq \sum_k u_k u_k^{\text{ln}} = u_D(\mathbf{R}). \end{aligned} \quad (\text{C.7})$$

■

Bibliography

- [1] CMU Motion Capture Dataset. <http://mocap.cs.cmu.edu/> 93, 96, 99
- [2] Almaev, T.R., Valstar, M.F.: Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In: ACII, pp. 356–361 (2013) 16
- [3] Amberg, B., Vetter, T.: Optimal landmark detection using shape models and branch and bound. In: ICCV (2011) 81
- [4] Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. JMLR 6, 1817–1853 (2005) 14
- [5] Aumann, R., Hart, S.: Bi-convexity and bi-martingales. Israel Journal of Mathematics 54(2), 159–180 (1986) 29
- [6] Aytar, Y., Zisserman, A.: Tabula rasa: Model transfer for object category detection. In: ICCV (2011) 12, 24
- [7] Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised domain adaptation by domain invariant projection. In: ICCV (2013) 12
- [8] Balakrishnan, V., Boyd, S., Balemi, S.: Branch and bound algorithm for computing the minimum stability degree of parameter-dependent linear systems. International Journal of Robust and Nonlinear Control 1(4), 295–317 (1991) 83
- [9] Barbič, J., Safonova, A., Pan, J.Y., Faloutsos, C., Hodgins, J.K., Pollard, N.S.: Segmenting motion capture data into distinct behaviors. In: Proceedings of Graphics Interface 2004, pp. 185–194. Canadian Human-Computer Communications Society (2004) 96
- [10] Bartlett, M.S., Littlewort, G.C., Frank, M.G., Lainscsek, C., Fasel, I.R., Movellan, J.R.: Automatic recognition of facial actions in spontaneous expressions. Journal of Multimedia 1(6), 22–35 (2006) xi, xii, 9, 10, 11, 15, 22, 33, 34, 78, 93, 94, 107, 108
- [11] Begum, N., Keogh, E.: Rare time series motif discovery from unbounded streams. In: VLDB (2015) 19
- [12] Belkin, M., Niyogi, P., Sindhvani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7, 2399–2434 (2006) 13
- [13] Ben-David, S., Schuller, R.: Exploiting task relatedness for multiple task learning. In: COLT (2003) 14
- [14] Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. TPAMI 23(3), 257–267 (2001) 9, 10
- [15] Boiman, O., Irani, M.: Detecting irregularities in images and in video. In: ICCV (2005) 79

- [16] Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H.P., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* **22**(14), 49–57 (2006) [12](#), [33](#), [37](#)
- [17] Brand, M., Oliver, N., Pentland, A.: Coupled HMMs for complex action recognition. In: *CVPR* (1997) [18](#)
- [18] Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: *ICCV* (2011) [80](#), [83](#)
- [19] Bruce, V.: What the human face tells the human mind: Some challenges for the robot-human interface. In: *IEEE Int. Workshop on Robot and Human Communication* (1992) [2](#)
- [20] Bruzzone, L., Marconcini, M.: Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *TPAMI* **32**(5), 770–787 (2010) [12](#), [32](#), [33](#), [36](#)
- [21] Chaaraoui, A.A., Climent-Pérez, P., Flórez-Revuelta, F.: A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Systems with Applications* **39**(12), 10,873–10,888 (2012) [18](#)
- [22] Chandra, S., Haque, A., Khan, L., Aggarwal, C.: Efficient sampling-based kernel mean matching. In: *International Conference on Data Mining* (2016) [107](#)
- [23] Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2**, 1–27 (2011). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm> [37](#)
- [24] Chang, C.Y., Huang, V.C.: Personalized facial expression recognition in indoor environments. In: *IJCNN* (2010) [32](#)
- [25] Chang, K.Y., Liu, T.L., Lai, S.H.: Learning partially-observed hidden conditional random fields for facial expression recognition. In: *CVPR* (2009) [10](#), [16](#), [51](#)
- [26] Chang, Y., Hu, C., Feris, R., Turk, M.: Manifold based analysis of facial expression. *Image and Vision Computing* **24**(6), 605–614 (2006) [10](#)
- [27] Chapelle, O.: Training a support vector machine in the primal. *Neural Computation* **19**(5), 1155–1178 (2007) [24](#), [25](#), [28](#), [114](#)
- [28] Chapelle, O., Schölkopf, B., Zien, A., et al.: *Semi-supervised learning*, vol. 2. MIT press Cambridge (2006) [13](#)
- [29] Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. *International Conference on Artificial Intelligence and Statistics* (2005) [13](#)
- [30] Chattopadhyay, R., Sun, Q., Fan, W., Davidson, I., Panchanathan, S., Ye, J.: Multi-Source domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data* **6**(4), 18 (2012) [12](#)
- [31] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16**, 321–357 (2002) [61](#)
- [32] Chen, J., Kim, M., Wang, Y., Ji, Q.: Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition. In: *CVPR* (2009) [10](#), [16](#)
- [33] Chen, J., Liu, X., Tu, P., Aragonés, A.: Learning person-specific models for facial expression and action unit recognition. *Pattern Recognition Letters* **34**(15), 1964–1970 (2013) [12](#), [33](#)
- [34] Chen, K., Wang, S.: Semi-supervised learning via regularized boosting working on multiple semi-

-
- supervised assumptions. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **33**(1), 129–143 (2011) [13](#)
- [35] Chew, S.W., Lucey, P., Lucey, S., Saragih, J., Cohn, J.F., Matthews, I., Sridharan, S.: In the pursuit of effective affective computing: The relationship between features and registration. *TSMC, Part B: Cybernetics* **42**(4), 1006–1016 (2012) [10](#)
 - [36] Chew, S.W., Lucey, P., Lucey, S., Saragih, J., Cohn, J.F., Sridharan, S.: Person-independent facial expression detection using constrained local models. In: *AFGR* (2011) [9](#), [11](#), [24](#)
 - [37] **Chu, W.S.**, Chen, C.P., Chen, C.S.: Momi-cosegmentation: Simultaneous segmentation of multiple objects among multiple images. In: *ACCV* (2010) [79](#)
 - [38] **Chu, W.S.**, De la Torre, F., Cohn, J.F.: Selective transfer machine for personalized facial action unit detection. In: *CVPR* (2013) [xviii](#), [8](#), [11](#), [12](#), [18](#), [49](#), [50](#)
 - [39] **Chu, W.S.**, De la Torre, F., Cohn, J.F.: Selective transfer machine for personalized facial expression analysis. *TPAMI* (99) (2016) [xviii](#), [8](#), [11](#), [12](#), [18](#)
 - [40] **Chu, W.S.**, De la Torre, F., Cohn, J.F.: Learning spatial and temporal cues for multi-label facial action unit detection. In: *AFGR* (2017) [xviii](#)
 - [41] **Chu, W.S.**, De la Torre, F., Cohn, J.F., Messinger, D.S.: A branch-and-bound framework for common event discovery. *IJCV* (2017) [xix](#)
 - [42] **Chu, W.S.**, Huang, C.R., Chen, C.S.: Identifying gender from unaligned facial images by set classification. In: *ICPR* (2010) [xiii](#), [66](#)
 - [43] **Chu, W.S.**, Huang, C.R., Chen, C.S.: Gender classification from unaligned facial images using support subspaces. *Information Sciences* **221**, 98–109 (2013) [xiii](#), [66](#)
 - [44] **Chu, W.S.**, Song, Y., Jaimes, A.: Video co-summarization: Video summarization by visual co-occurrence. In: *CVPR* (2015) [xix](#), [9](#), [10](#), [80](#)
 - [45] **Chu, W.S.**, Zeng, J., De la Torre, F., Cohn, J.F., Messinger, D.S.: Unsupervised synchrony discovery in human interaction. In: *ICCV*, pp. 3146–3154 (2015) [xix](#)
 - [46] **Chu, W.S.**, Zhou, F., De la Torre, F.: Unsupervised temporal commonality discovery. In: *ECCV* (2012) [xix](#), [9](#), [18](#), [125](#)
 - [47] Cohen, I., Sebe, N., Garg, A., Chen, L.S., Huang, T.S.: Facial expression recognition from video sequences: temporal and static modeling. *CVIU* **91**(1), 160–187 (2003) [10](#)
 - [48] Cohen, I., Sebe, N., Gozman, F., Cirelo, M.C., Huang, T.S.: Learning bayesian network classifiers for facial expression recognition both labeled and unlabeled data. In: *CVPR* (2003) [10](#)
 - [49] Cohn, J.F., De la Torre, F.: *The Oxford Handbook of Affective Computing*, chap. Automated Face Analysis for Affective Computing (2014) [xi](#), [2](#), [4](#), [8](#)
 - [50] Cohn, J.F., Sayette, M.A.: Spontaneous facial expression in a small group can be automatically measured: An initial demonstration. *Behavior Research Methods* **42**(4), 1079–1086 (2010) [xiii](#), [xvi](#), [xvii](#), [53](#), [57](#), [59](#), [60](#), [68](#), [70](#), [71](#), [75](#), [76](#), [78](#), [108](#)
 - [51] Cohn, J.F., Zlochower, A.: A computerized analysis of facial expression: Feasibility of automated discrimination. In: *American Psychological Society* (1995) [13](#)
 - [52] Collobert, R., Sinz, F., Weston, J., Bottou, L.: Large scale transductive svms. *JMLR* **7**, 1687–1712 (2006) [32](#), [37](#)
 - [53] Cooper, H., Bowden, R.: Learning signs from subtitles: A weakly supervised approach to sign lan-

- guage recognition. In: CVPR (2009) 18
- [54] Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. TPAMI (6), 681–685 (2001) 9
- [55] Cruz, A., Bhanu, B., Yang, S.: A psychologically-inspired match-score fusion model for video-based facial expression recognition. In: ACII (2011) 9
- [56] Dai, Y., Shibata, Y., Ishii, T., Hashimoto, K., Katamachi, K., Noguchi, K., Kakizaki, N., Ca, D.: An associate memory model of facial expressions and its application in facial expression recognition of patients on bed. In: ICME (2001) 2
- [57] Dapogny, A., Bailly, K., Dubuisson, S.: Dynamic facial expression recognition by joint static and multi-time gap transition classification. In: AFGR (2015) 10
- [58] Darwin, C.: The expression of the emotions in man and animals. New York: Oxford University. (1872/1998) 2
- [59] Daumé III, H.: Frustratingly easy domain adaptation. In: Conference of the Association for Computational Linguistics (2007) 37
- [60] De la Torre, F., **Chu, W.S.**, Xiong, X., Ding, X., Cohn, J.F.: IntraFace. In: AFGR (2015) xix, 3, 53, 98
- [61] De la Torre, F., Simon, T., Ambadar, Z., Cohn, J.F.: FAST-FACS: A computer-assisted system to increase speed and reliability of manual FACS coding. In: ACII (2011) 16
- [62] De la Torre, M., Granger, E., Radtke, P.V., Sabourin, R., Gorodnichy, D.O.: Partially-supervised learning from facial trajectories for face recognition in video surveillance. Information Fusion (2014) 2
- [63] Delaherche, E., Chetouani, M., Mahdhaoui, A., Saint-Georges, C., Viaux, S., Cohen, D.: Interpersonal synchrony: A survey of evaluation methods across disciplines. TAFPC 3(3), 349–365 (2012) 98
- [64] Ding, X., Chu, W.S., De la Torre, F., Cohn, J.F., Wang, Q.: Facial action unit event detection by cascade of tasks. In: IEEE Conference on International Conference on Computer Vision (2013) 20, 54
- [65] Ding, X., **Chu, W.S.**, Torre, F., Cohn, J.F., Wang, Q.: Facial action unit event detection by cascade of tasks. In: ICCV (2013) xi, xviii, xix, 10, 11, 16, 101, 108
- [66] Ding, X., **Chu, W.S.**, la Torre, F.D., Cohn, J.F., Wang, Q.: Cascade of tasks for facial expression analysis. Image and Vision Computing (2016) xi, xviii, xix, 10, 11, 16, 108
- [67] Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: CVPR (2015) 17
- [68] Dornaika, F., Davoine, F.: Simultaneous facial action tracking and expression recognition in the presence of head motion. IJCV 76(3), 257–281 (2008) 18
- [69] Du, S., Martinez, A.M.: Compound facial expressions of emotion: from basic research to clinical applications. Dialogues in clinical neuroscience 17(4), 443 (2015) 14
- [70] Du, S., Tao, Y., Martinez, A.M.: Compound facial expressions of emotion. Proceedings of the National Academy of Sciences 111(15), 1454–1462 (2014) 18
- [71] Duan, L., Tsang, I.W., Xu, D.: Domain transfer multiple kernel learning. TPAMI 34(3), 465–479 (2012) 24, 32, 33
- [72] Duan, L., Xu, D., Tsang, I.W.: Domain adaptation from multiple sources: A domain-dependent regu-

-
- larization approach. *IEEE Trans. on Neural Networks and Learning Systems* **23**(3), 504–518 (2012) [12](#), [32](#), [33](#), [36](#), [37](#)
- [73] Duchenne, O., Laptev, I., Sivic, J., Bach, F., Ponce, J.: Automatic annotation of human actions in video. In: *ICCV* (2009) [18](#)
 - [74] Dudík, M., Schapire, R.E., Phillips, S.J.: Correcting sample selection bias in maximum entropy density estimation. In: *NIPS* (2005) [12](#)
 - [75] Ekman, P.: An argument for basic emotions. *Cognition & Emotion* **6**(3-4), 169–200 (1992) [2](#)
 - [76] Ekman, P., Friesen, W., Hager, J.C.: Facial action coding system. *A Human Face* (2002) [2](#), [4](#), [14](#)
 - [77] Ekman, P., Rosenberg, E.L.: What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS). Oxford University Press, USA (1997) [xiii](#), [49](#), [59](#), [60](#)
 - [78] Eleftheriadis, S., Rudovic, O., Pantic, M.: Multi-conditional latent variable model for joint facial action unit detection. In: *ICCV* (2015) [14](#)
 - [79] Eleftheriadis, S., Rudovic, O., Pantic, M.: Multi-conditional latent variable model for joint facial action unit detection. In: *ICCV* (2015) [15](#), [50](#)
 - [80] Everingham, M., Zisserman, A., Williams, C.I., Van Gool, L.: The PASCAL visual object classes challenge 2006 results. In: *2th PASCAL Challenge* (2006) [83](#), [86](#), [94](#), [95](#)
 - [81] Evgeniou, A., Pontil, M.: Multi-task feature learning. In: *NIPS* (2007) [14](#)
 - [82] Evgeniou, T., Micchelli, C.A., Pontil, M.: Learning multiple tasks with kernel methods. In: *JLMR*, pp. 615–637 (2005) [14](#)
 - [83] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. *JMLR* **9**, 1871–1874 (2008) [72](#)
 - [84] Fasel, B.: Head-pose invariant facial expression recognition using convolutional neural networks. In: *ICMI* (2002) [11](#)
 - [85] Fawcett, T.: An introduction to roc analysis. *Pattern recognition letters* **27**(8), 861–874 (2006) [66](#), [70](#)
 - [86] Feris, R., Bobbitt, R., Brown, L., Pankanti, S.: Attribute-based people search: Lessons learnt from a practical surveillance system. In: *ICMR* (2014) [18](#)
 - [87] Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: *ICCV* (2013) [12](#), [36](#)
 - [88] Floudas, C., Visweswaran, V.: A global optimization algorithm (gop) for certain classes of nonconvex nlpssi. theory. *Computers & chemical engineering* **14**(12), 1397–1417 (1990) [26](#)
 - [89] Forbes, E.E., Cohn, J.F., Allen, N.B., Lewinsohn, P.M.: Infant affect during parent-infant interaction at 3 and 6 months: Differences between mothers and fathers and influence of parent history of depression. *Infancy* **5**, 61–84 (2004) [2](#)
 - [90] Gehrig, T., Ekenel, H.K.: A common framework for real-time emotion recognition and facial action unit detection. In: *CVPR Workshops* (2011) [9](#)
 - [91] Gendron, B., Crainic, T.G.: Parallel branch-and-branch algorithms: Survey and synthesis. *Operations research* **42**(6), 1042–1066 (1994) [92](#)
 - [92] Ghosh, S., Laksana, E., Scherer, S., Morency, L.P.: A multi-label convolutional neural network approach to cross-domain action unit detection. In: *ACII* (2015) [17](#), [51](#)

- [93] Girard, J., **Chu, W.S.**, Jeni, L., Cohn, J.F., De la Torre, F.: Sayette Group Formation Task (GFT): Spontaneous Facial Expression Database. In: AFGR (2017) **v, xvi, xix, 19, 20, 61, 66, 70, 71, 72, 107**
- [94] Girard, J.M., Cohn, J.F., Jeni, L.A., Sayette, M.A., De la Torre, F.: Spontaneous facial expression in unscripted social interactions can be measured automatically. *Behavior research methods* (2014) **33, 34**
- [95] Goldberg, D.A., Goldberg, M.B., Goldberg, M.D., Goldberg, B.M.: Obtaining person-specific images in a public venue (2009). US Patent 7,561,723 **44**
- [96] Goldberger, J., Gordon, S., Greenspan, H.: An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In: ICCV (2003) **86**
- [97] Gong, B., Shi, Y., Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: CVPR (2012) **12**
- [98] Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: ICCV (2011) **12**
- [99] Gorski, J., Pfeuffer, F., Klamroth, K.: Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research* **66**(3), 373–407 (2007) **29, 30, 31**
- [100] Grandvalet, Y., Bengio, Y.: Semi-supervised learning by entropy minimization. In: NIPS (2005) **13**
- [101] Gratch, J., Cheng, L., Marsella, S., Boberg, J.: Felt emotion and social context determine the intensity of smiles in a competitive video game. In: AFGR (2013) **2**
- [102] Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: ICASSP (2013) **51, 52**
- [103] Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: Covariate shift by kernel mean matching. *Dataset shift in machine learning* pp. 131–160 (2009) **xii, 12, 25, 26, 27, 32, 33, 36, 37, 41**
- [104] Gudi, A., Tasli, H.E., den Uyl, T.M., Maroulis, A.: Deep learning based facs action unit occurrence and intensity estimation. In: AFGR (2015) **11, 17**
- [105] Gunes, H., Piccardi, M.: Affect recognition from face and body: early fusion vs. late fusion. In: *International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3437–3443 (2005) **9, 10**
- [106] Guo, G., Guo, R., Li, X.: Facial Expression Recognition Influenced by Human Aging. *TAFFC* **4**(3), 291–298 (2013) **18**
- [107] Guo, Y., Zhao, G., Pietikäinen, M.: Dynamic facial expression recognition using longitudinal facial expression atlases. In: ECCV (2012) **9**
- [108] Gusfield, D.: *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge Univ Press (1997) **86**
- [109] Hammal, Z., Cohn, J.F., Messinger, D.S.: Head movement dynamics during play and perturbed mother-infant interaction. *IEEE transactions on affective computing* **6**(4), 361–370 (2015) **xv, 121**
- [110] Han, D., Bo, L., Sminchisescu, C.: Selection and Context for Action Recognition. In: ICCV (2009) **80, 83**
- [111] He, H., Garcia, E.A.: Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering* **21**(9), 1263–1284 (2009) **61**

-
- [112] Hoai, M., Lan, Z.Z., De la Torre, F.: Joint segmentation and classification of human actions in video. In: CVPR (2011) **88**, **100**, **101**
- [113] Hongeng, S., Nevatia, R.: Multi-agent event recognition. In: ICCV (2001) **18**
- [114] Huang, D., De la Torre, F.: Bilinear kernel reduced rank regression for facial expression synthesis. In: ECCV (2010) **2**
- [115] Huang, J., Ling, C.X.: Using auc and accuracy in evaluating learning algorithms. IEEE Transactions on knowledge and Data Engineering **17**(3), 299–310 (2005) **20**
- [116] Ijjina, E.P., Mohan, C.K.: Facial expression recognition using kinect depth sensor and convolutional neural networks. In: ICMLA (2014) **10**
- [117] Jaiswal, S., Martinez, B., Valstar, M.F.: Learning to combine local models for facial action unit detection. In: AFGR (2015) **13**
- [118] Jaiswal, S., Valstar, M.F.: Deep learning the dynamic appearance and shape of facial action units. In: WACV (2016) **17**, **60**
- [119] Jeni, L.A., Cohn, J.F., Kanade, T.: Dense 3d face alignment from 2d videos in real-time. In: AFGR (2015) **9**
- [120] Jeni, L.A., Girard, J.M., Cohn, J.F., De La Torre, F.: Continuous au intensity estimation using localized, sparse facial feature space. In: AFGR (2013) **15**
- [121] Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A biologically inspired system for action recognition. In: ICCV (2007) **18**
- [122] Jhuo, I.H., Liu, D., Lee, D., Chang, S.F.: Robust visual domain adaptation with low-rank reconstruction. In: CVPR (2012) **12**
- [123] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014) **54**
- [124] Jiang, B., Martinez, B., Valstar, M.F., Pantic, M.: Decision Level Fusion of Domain Specific Regions for Facial Action Recognition. In: ICPR (2014) **16**
- [125] Jiang, B., Valstar, M.F., Pantic, M.: Action unit detection using sparse appearance descriptors in space-time video volumes. In: AFGR (2011) **9**, **10**, **24**, **60**
- [126] Joachims, T.: Transductive inference for text classification using support vector machines. In: ICML (1999) **12**, **33**, **37**
- [127] Kaltwang, S., Rudovic, O., Pantic, M.: Continuous pain intensity estimation from facial expressions. In: International Symposium on Visual Computing, pp. 368–377 (2012) **15**
- [128] Kanade, T.: Picture processing system by computer complex and recognition of human faces. Doctoral dissertation, Kyoto University **3952**, 83–97 (1973) **2**
- [129] Kanade, T., Cohn, J.F., Tian, Y.: Comprehensive database for facial expression analysis. In: AFGR (2000) **9**
- [130] Kapoor, A., Picard, R.W.: Multimodal affect recognition in learning environments. In: ACM MM (2005) **10**, **11**
- [131] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014) **51**, **52**, **57**
- [132] Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and infor-

- mation systems **7**(3), 358–386 (2005) [90](#)
- [133] Khosla, A., Zhou, T., Malisiewicz, T., Efros, A., Torralba, A.: Undoing the damage of dataset bias. In: ECCV (2012) [24](#)
 - [134] Koelstra, S., Pantic, M.: Non-rigid registration using free-form deformations for recognition of facial actions and their temporal dynamics. In: AFGR (2008) [10](#)
 - [135] Koelstra, S., Pantic, M., Patras, I.: A dynamic texture-based approach to recognition of facial actions and their temporal models. TPAMI **32**(11), 1940–1954 (2010) [10](#)
 - [136] Kotsia, I., Pitas, I.: Facial expression recognition in image sequences using geometric deformation features and support vector machines. TIP **16**(1), 172–187 (2007) [14](#)
 - [137] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS (2012) [xiii](#), [50](#), [51](#), [54](#)
 - [138] Krüger, S.E., Schafföner, M., Katz, M., Andelic, E., Wendemuth, A.: Speech recognition with support vector machines in a hybrid system. In: Interspeech (2005) [100](#)
 - [139] Kulis, B., Saenko, K., Darrell, T.: What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: CVPR (2011) [12](#), [37](#)
 - [140] Kumano, S., Otsuka, K., Yamato, J., Maeda, E., Sato, Y.: Pose-invariant facial expression recognition using variable-intensity templates. IJCV **83**(2), 178–194 (2009) [18](#)
 - [141] Kyperountas, M., Tefas, A., Pitas, I.: Salient feature and reliable classifier selection for facial expression classification. Pattern Recognition **43**(3), 972–986 (2010) [9](#)
 - [142] Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. TPAMI **31**(12), 2129–2142 (2009) [xiv](#), [79](#), [81](#), [83](#), [86](#), [87](#), [90](#), [92](#), [94](#), [95](#)
 - [143] Lapin, M., Hein, M., Schiele, B.: Loss functions for top-k error: Analysis and insights. In: CVPR (2016) [66](#)
 - [144] Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008) [80](#), [83](#), [88](#), [100](#)
 - [145] Lehmann, A., Leibe, B., Van Gool, L.: Fast prism: Branch and bound hough transform for object class detection. IJCV **94**(2), 175–197 (2011) [81](#)
 - [146] Li, Y., Chen, J., Zhao, Y., Ji, Q.: Data-free prior model for facial action unit recognition. TAFRC **4**(2), 127–141 (2013) [14](#)
 - [147] Lien, J.J.J., Cohn, J.F., Kanade, T., Li, C.C.: Subtly different facial expression recognition and expression intensity estimation. In: CVPR (1998) [2](#)
 - [148] Lien, J.J.J., Kanade, T., Cohn, J.F., Li, C.C.: Detection, tracking, and classification of action units in facial expression. Robotics and Autonomous Systems **31**(3), 131–146 (2000) [2](#), [10](#)
 - [149] Lin, J.: Divergence measures based on the Shannon entropy. IEEE Trans. on Information Theory **37**(1), 145–151 (1991) [55](#)
 - [150] Littlewort, G., Bartlett, M.S., Fasel, I., Susskind, J., Movellan, J.: Dynamics of facial expression extracted automatically from video. Image and Vision Computing **24**(6), 615–625 (2006) [10](#), [13](#), [18](#)
 - [151] Littlewort, G., Whitehill, J., Wu, T., Fasel, I., Frank, M., Movellan, J., Bartlett, M.: The computer expression recognition toolbox (CERT). In: AFGR (2011) [3](#), [9](#), [38](#)
 - [152] Liu, C.D., Chung, Y.N., Chung, P.C.: An interaction-embedded HMM framework for human behavior

-
- understanding: with nursing environments as examples. *IEEE Trans. on Information Technology in Biomedicine* **14**(5), 1236–1246 (2010) [18](#)
- [153] Liu, H., Yan, S.: Common visual pattern discovery via spatially coherent correspondences. In: *CVPR* (2010) [79](#)
- [154] Liu, J., Shah, M., Kuipers, B., Savarese, S.: Cross-view action recognition via view knowledge transfer. In: *CVPR* (2011) [80](#), [83](#)
- [155] Liu, M., Li, S., Shan, S., Chen, X.: AU-aware deep networks for facial expression recognition. In: *AFGR* (2013) [10](#), [11](#)
- [156] Liu, P., Han, S., Meng, Z., Tong, Y.: Facial expression recognition via a boosted deep belief network. In: *CVPR* (2014) [10](#), [14](#)
- [157] Liu, P., Zhou, J.T., Tsang, I.W.H., Meng, Z., Han, S., Tong, Y.: Feature disentangling machine—a novel approach of feature selection and disentangling in facial expression analysis. In: *ECCV* (2014) [14](#)
- [158] Liu, W., Wang, J., Chang, S.F.: Robust and scalable graph-based semi-supervised learning. *Proceedings of the IEEE* **100**(9), 2624–2638 (2012) [13](#)
- [159] Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: *ICML* (2015) [108](#)
- [160] Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: *IJCAI* (1981) [9](#)
- [161] Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression. In: *CVPR Workshops* (2010) [xii](#), [9](#), [13](#), [15](#), [18](#), [33](#), [34](#), [40](#), [53](#), [60](#)
- [162] Lucey, S., Ashraf, A.B., Cohn, J.: Investigating spontaneous facial action recognition through AAM representations of the face. *Face recognition pp.* 275–286 (2007) [9](#), [10](#), [11](#), [36](#)
- [163] Mahoor, M.H., Cadavid, S., Messinger, D.S., Cohn, J.F.: A framework for automated measurement of the intensity of non-posed facial action units. In: *CVPRW* (2009) [15](#)
- [164] Martinez, A., Du, S.: A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *JMLR* **13**, 1589–1608 (2012) [4](#), [8](#), [22](#)
- [165] Matsuo, H., Khiat, A.: Prediction of drowsy driving by monitoring driver’s behavior. In: *ICPR* (2012) [2](#)
- [166] Matthews, I., Baker, S.: Active appearance models revisited. *IJCV* **60**(2), 135–164 (2004) [9](#), [94](#)
- [167] Mavadati, S.M., Mahoor, M.H., Bartlett, K., Trinh, P., Cohn, J.F.: Disfa: A spontaneous facial action intensity database. *IEEE Trans. on Affective Computing* **4**(2), 151–160 (2013) [15](#), [108](#)
- [168] Melacci, S., Belkin, M.: Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research* **12**, 1149–1184 (2011) [13](#)
- [169] Messinger, D.M., Ruvolo, P., Ekas, N.V., Fogel, A.: Applying machine learning to infant interaction: The development is in the details. *Neural Networks* **23**(8), 1004–1016 (2010) [18](#)
- [170] Messinger, D.S., Mahoor, M.H., Chow, S.M., Cohn, J.F.: Automated measurement of facial expression in infant–mother interaction: A pilot study. *Infancy* **14**(3), 285–305 (2009) [xvii](#), [78](#), [99](#), [107](#), [108](#), [121](#), [122](#)
- [171] Ming, Z., Bugeau, A., Rouas, J.L., Shochi, T.: Facial action units intensity estimation by the fusion of features with multi-kernel support vector machine. In: *AFGR* [15](#)

- [172] Minnen, D., Isbell, C., Essa, I., Starner, T.: Discovering multivariate motifs using subsequence density estimation. In: AAAI (2007) 19
- [173] Mohammadi, M.R., Fatemizadeh, E., Mahoor, M.H.: Intensity estimation of spontaneous facial action units based on their sparsity properties. *IEEE Transactions on Cybernetics* **46**(3), 817–826 (2016) 15
- [174] Moore, S., Bowden, R.: Local binary patterns for multi-view facial expression recognition. *CVIU* **115**(4), 541–558 (2011) 9
- [175] Muandet, K., Balduzzi, D., Schölkopf, B.: Domain generalization via invariant feature representation. In: ICML (2013) 12
- [176] Mueen, A., Keogh, E.: Online discovery and maintenance of time series motifs. In: KDD (2010) 19
- [177] Mueen, A., Keogh, E.J., Zhu, Q., Cash, S., Westover, M.B.: Exact discovery of time series motifs. In: SDM (2009) 19
- [178] Mukherjee, L., Singh, V., Peng, J.: Scale invariant cosegmentation for image groups. In: CVPR (2011) 79
- [179] Murphy, K.P.: Machine learning: a probabilistic perspective. MIT press (2012) 85
- [180] Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. *IEEE Trans. on Computers* **100**(9), 917–922 (1977) 92
- [181] Nayak, S., Duncan, K., Sarkar, S., Loeding, B.: Finding recurrent patterns from continuous sign language sentences for automated extraction of signs. *JMLR* **13**(1), 2589–2615 (2012) 19
- [182] Oliver, N.M., Rosario, B., Pentland, A.P.: A bayesian computer vision system for modeling human interactions. *TPAMI* **22**(8), 831–843 (2000) 18
- [183] Orrite, C., Gañán, A., Rogez, G.: Hog-based decision tree for facial expression classification. In: Pattern Recognition and Image Analysis (2009) 9
- [184] Oster, H., Ekman, P.: Facial behavior in child development. In: Minnesota symposia on child psychology, vol. 11, pp. 231–276 (1978) 121
- [185] P., L., F., C.J., M., P.K., E., S.P., I., M.: Painful data: The unbc-mcmaster shoulder pain expression archive database. In: AFGR (2011) 15
- [186] Pantic, M., Bartlett, M.S.: Machine analysis of facial expressions. *Face Recognition* **2**(8), 377–416 (2007) 4, 8
- [187] Pantic, M., Patras, I.: Dynamics of Facial Expression : Recognition of Facial Actions and Their Temporal Segments. *TSMC, Part B: Cybernetics* **36**(2), 433–449 (2006) 15
- [188] Pantic, M., Patras, I.: Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. *TSMC, Part B: Cybernetics* **36**(2), 433–449 (2006) 10
- [189] Pantic, M., Pentland, A., Nijholt, A., Huang, T.S.: Human computing and machine understanding of human behavior: a survey. In: Artificial Intelligence for Human Computing, pp. 47–71. Springer (2007) 22
- [190] Parkhi, O.M., Vedaldi, A., Zisserman, A.: Deep face recognition. In: British Machine Vision Conference (2015) xiii, 54, 55
- [191] Pfister, T., Li, X., Zhao, G., Pietikainen, M.: Recognising spontaneous facial micro-expressions. In: ICCV (2011) 17
- [192] Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers* **10**(3), 61–74 (1999) 100

-
- [193] Prati, R.C., Batista, G., Silva, D.F.: Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowledge and Information Systems* **45**(1), 247–270 (2015) 61
- [194] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.: *Dataset shift in machine learning*. The MIT Press (2009) 12
- [195] Reddy, K.K., Shah, M.: Recognizing 50 human action categories of web videos. *Machine Vision and Applications* **24**(5), 971–981 (2013) 18
- [196] Rifai, S., Bengio, Y., Courville, A., Vincent, P., Mirza, M.: Disentangling factors of variation for facial expression recognition. In: *ECCV* (2012) 10
- [197] Romera-Paredes, B., Aung, M.S., Pontil, M., Bianchi-Berthouze, N., de C Williams, A., Watson, P.: Transfer learning to account for idiosyncrasy in face and body expressions. In: *AFGR* (2013) 32
- [198] Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover’s distance as a metric for image retrieval. *IJCV* **40**(2), 99–121 (2000) 86
- [199] Rudovic, O., Pavlovic, V., Pantic, M.: Kernel conditional ordinal random fields for temporal segmentation of facial action units. In: *ECCV Workshops* (2012) 16
- [200] Rudovic, O., Pavlovic, V., Pantic, M.: Multi-output Laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In: *CVPR* (2012) 9
- [201] Rudovic, O., Pavlovic, V., Pantic, M.: Context-sensitive dynamic ordinal regression for intensity estimation of facial action units. *TPAMI* **37**(5), 944–958 (2015) 12, 15, 32
- [202] Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: *CVPR* (2012) 80, 83, 88, 100
- [203] Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: *ECCV* (2010) 12, 37
- [204] Sandbach, G., Zafeiriou, S., Pantic, M., Yin, L.: Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing* (2012) 4, 8
- [205] Sangineto, E., Zen, G., Ricci, E., Sebe, N.: We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer. In: *ACM MM* (2014) 8, 12, 18, 45, 49, 50
- [206] Sariyanidi, E., Gunes, H., Cavallaro, A.: Automatic analysis of facial affect: A survey of registration, representation, and recognition. *TPAMI* **37**(6), 1113–1133 (2015) 22
- [207] Savran, A., Sankur, B., Bilge, M.T.: Regression-based intensity estimation of facial action units. *IVC* **30**(10), 774–784 (2012) 15
- [208] Savran, A., Sankur, B., Taha Bilge, M.: Regression-based intensity estimation of facial action units. *Image and Vision Computing* **30**(10), 774–784 (2012) 16
- [209] Sayette, M.A., Creswell, K.G., Dimoff, J.D., Fairbairn, C.E., Cohn, J.F., Heckman, B.W., Kirchner, T.R., Levine, J.M., Moreland, R.L.: Alcohol and group formation a multimodal investigation of the effects of alcohol on emotion and social bonding. *Psychological science* (2012) xii, 33, 34, 43, 97, 100
- [210] Schindler, G., Krishnamurthy, P., Lubliner, R., Liu, Y., Dellaert, F.: Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In: *CVPR* (2008) 79
- [211] Schmidt, R.C., Morr, S., Fitzpatrick, P., Richardson, M.J.: Measuring the dynamics of interactional synchrony. *Journal of Nonverbal Behavior* **36**(4), 263–279 (2012) 19
- [212] Scholkopf, B.: The kernel trick for distances. In: *NIPS* (2001) 84

- [213] Schuller, B., Rigoll, G.: Timing levels in segment-based speech emotion recognition. In: Interspeech (2006) **88**, **100**
- [214] Senechal, T., Rapp, V., Salam, H., Segulier, R., Bailly, K., Prevost, L.: Facial action recognition combining heterogeneous features via multikernel learning. *TSMC, Part B: Cybernetics* **42**(4), 993–1005 (2012) **9**, **10**
- [215] Shan, C., Gong, S., Mcowan, P.W.: Facial expression recognition based on Local Binary Patterns: A comprehensive study. *Image and Vision Computing* **27**(6), 803–816 (2009) **9**, **10**
- [216] Shang, L., Chan, K.P.: Nonparametric discriminant HMM and application to facial expression recognition. In: *CVPR* (2009) **9**, **16**
- [217] Shergill, G.H., Sarrafzadeh, H., Diegel, O., Shekar, A.: Computerized sales assistants: The application of computer technology to measure consumer interest. *Journal of Electronic Commerce Research* **9**(2), 176–191 (2008) **2**
- [218] Shojaeilangari, S., Yau, W.Y., Nandakumar, K., Li, J., Teoh, E.K.: Robust representation and recognition of facial emotions using extreme sparse learning. *TIP* **24**(7), 2140–2152 (2015) **13**
- [219] Si, Z., Pei, M., Yao, B., Zhu, S.: Unsupervised learning of event and-or grammar and semantics from video. In: *ICCV* (2011) **19**
- [220] Siddiqi, M.H., Ali, R., Khan, A.M., Park, Y.T., Lee, S.: Human facial expression recognition using stepwise linear discriminant analysis and hidden conditional random fields. *TIP* **24**(4), 1386–1398 (2015) **13**
- [221] Sikka, K., Wu, T., Susskind, J., Bartlett, M.: Exploring bag of words architectures in the facial expression domain. In: *ECCV Workshops* (2012) **10**
- [222] Simon, T., Nguyen, M.H., De La Torre, F., Cohn, J.F.: Action unit detection with segment-based SVMs. In: *CVPR* (2010) **9**, **10**, **11**, **16**, **24**, **45**
- [223] Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013) **59**
- [224] Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: *NIPS* (2014) **17**, **54**, **57**
- [225] Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: *ICCV* (2003) **9**, **10**, **80**
- [226] Sorower, M.S.: A literature survey on algorithms for multi-label learning. Oregon State University, Corvallis (2010) **15**
- [227] Sugiyama, M., Krauledat, M., Müller, K.: Covariate shift adaptation by importance weighted cross validation. *JMLR* **8**, 985–1005 (2007) **12**, **33**
- [228] Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P.V., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: *NIPS* (2007) **25**, **26**, **33**, **42**
- [229] Sun, M., Telaprolu, M., Lee, H., Savarese, S.: An efficient branch-and-bound algorithm for optimal human pose estimation. In: *CVPR* (2012) **81**
- [230] Sun, Y., Wong, A., Kamel, M.S.: Classification of imbalanced data: A review. *IJPRAI* **23**(04), 687–719 (2009) **61**
- [231] Taheri, S., Patel, V.M., Chellappa, R.: Component-Based Recognition of Faces and Facial Express-

- sions. *TAFFC* **4**(4), 360–371 (2013) [18](#)
- [232] Taheri, S., Qiu, Q., Chellappa, R.: Structure-preserving sparse decomposition for facial expression analysis. *TIP* **23**(8), 3590–3603 (2014) [14](#)
- [233] Tariq, U., Huang, T.S.: Features and fusion for expression recognitionA comparative analysis. In: *CVPR* (2012) [16](#)
- [234] Tariq, U., Lin, K.H., Li, Z., Zhou, X., Wang, Z., Le, V., Huang, T.S., Lv, X., Han, T.X.: Emotion recognition from an ensemble of features. In: *AFGR* (2011) [16](#)
- [235] Tian, Y.L., Kanade, T., Cohn, J.F.: Facial expression analysis. *Handbook of face recognition* pp. 247–275 (2005) [9](#), [10](#)
- [236] Tomkins, S.S.: *Affect, imagery, consciousness*. New York: Springer (1962) [2](#)
- [237] Tong, Y., Chen, J., Ji, Q.: A unified probabilistic framework for spontaneous facial action modeling and understanding. *TPAMI* **32**(2), 258–273 (2010) [9](#), [16](#)
- [238] Tong, Y., Ji, Q.: Learning bayesian networks with qualitative constraints. In: *CVPR* (2008) [14](#)
- [239] Tong, Y., Liao, W., Ji, Q.: Facial action unit recognition by exploiting their dynamic and semantic relationships. *TPAMI* **29**(10), 1683–1699 (2007) [10](#), [14](#), [16](#)
- [240] Torralba, A., Efros, A.: Unbiased look at dataset bias. In: *CVPR* (2011) [12](#)
- [241] Torralba, A., Murphy, K.P., Freeman, W.T.: Sharing features: efficient boosting procedures for multi-class object detection. In: *CVPR* (2004) [14](#)
- [242] De la Torre, F., Cohn, J.F.: Facial expression analysis. *Visual Analysis of Humans: Looking at People* p. 377 (2011) [4](#), [8](#), [9](#), [22](#)
- [243] Tsalakanidou, F., Malassiotis, S.: Real-time 2D+3D facial action and expression recognition. *Pattern Recognition* **43**(5), 1763–1775 (2010) [10](#)
- [244] Tsoumakas, G., Katakis, I.: *Multi-label classification: An overview*. Dept. of Informatics, Aristotle University of Thessaloniki, Greece (2006) [15](#)
- [245] Turaga, P., Veeraraghavan, A., Chellappa, R.: Unsupervised view and rate invariant clustering of video sequences. *CVIU* **113**(3), 353–371 (2009) [19](#)
- [246] Valstar, M., Pantic, M.: Fully automatic facial action unit detection and temporal analysis. In: *CVPR Workshops* (2006) [18](#)
- [247] Valstar, M., Pantic, M., Patras, I.: Motion history for facial action detection in video. In: *International Conference on Systems, Man and Cybernetics* (2004) [9](#), [10](#)
- [248] Valstar, M.F., Mehu, M., Jiang, B., Pantic, M., Scherer, K.: Meta-analysis of the first facial expression recognition challenge. *TSMC, Part B: Cybernetics* **42**(4), 966–979 (2012) [xii](#), [4](#), [8](#), [9](#), [10](#), [11](#), [22](#), [33](#), [34](#), [36](#), [40](#)
- [249] Valstar, M.F., Pantic, M.: Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In: *International Conference on Human-Computer Interaction* (2007) [15](#)
- [250] Valstar, M.F., Pantic, M.: Fully automatic recognition of the temporal phases of facial actions. *TSMC, Part B: Cybernetics* **42**(1), 28–43 (2012) [15](#)
- [251] Van Gestel, T., Suykens, J., Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., De Moor, B., Vandewalle, J.: Benchmarking least squares support vector machine classifiers. *Machine Learning* **54**(1), 5–32 (2004) [31](#)

- [252] Viola, P., Jones, M.J.: Robust real-time face detection. *IJCV* **57**(2), 137–154 (2004) [xiv](#), [80](#), [94](#), [95](#)
- [253] Walecki, R., Rudovic, O., Pavlovic, V., Pantic, M.: Variable-state Latent Conditional Random Fields for Facial Expression Recognition and Action Unit Detection. In: *AFGR* (2015) [10](#), [16](#)
- [254] Wang, S.J., Yan, W.J., Zhao, G., Fu, X., Zhou, C.G.: Micro-expression recognition using robust principal component analysis and local spatiotemporal directional features. In: *ECCV Workshops* (2014) [17](#)
- [255] Wang, Y., Jiang, H., Drew, M.S., Li, Z., Mori, G.: Unsupervised discovery of action classes. In: *CVPR* (2006) [19](#), [79](#)
- [256] Wang, Y., Velipasalar, S.: Frame-level temporal calibration of unsynchronized cameras by using Longest Consecutive Common Subsequence. In: *ICASSP* (2009) [94](#)
- [257] Wang, Z., Li, Y., Wang, S., Ji, Q.: Capturing global semantic relationships for facial action unit recognition. In: *ICCV* (2013) [10](#), [14](#)
- [258] Wang, Z., Li, Y., Wang, S., Ji, Q.: Capturing global semantic relationships for facial action unit recognition. In: *ICCV* (2013) [14](#), [60](#)
- [259] Wendell, R.E., Hurter, A.P.: Minimization of a non-separable objective function subject to disjoint constraints. *Operations Research* **24**(4), 643–657 (1976) [30](#)
- [260] Wöllmer, M., Kaiser, M., Eyben, F., Schuller, B., Rigoll, G.: LSTM-Modeling of continuous emotions in an audiovisual affect recognition framework. *IVC* **31**(2), 153–163 (2013) [10](#)
- [261] Wong, A., You, M.: Entropy and distance of random graphs with application to structural pattern recognition. *TPAMI* (5), 599–609 (1985) [56](#)
- [262] Wu, C., Wang, S., Ji, Q.: Multi-instance hidden markov model for facial expression recognition. In: *AFGR* (2015) [10](#), [15](#), [50](#)
- [263] Wu, T., Bartlett, M.S., Movellan, J.: Facial expression recognition using gabor motion energy filters. In: *CVPR Workshops* (2010) [9](#), [10](#)
- [264] Wu, T., Butko, N.J., Ruvolo, P., Whitehill, J., Bartlett, M.S., Movellan, J.R.: Multilayer architectures for facial action unit recognition. *TSMC, Part B: Cybernetics* **42**(4), 1027–1038 (2012) [16](#)
- [265] Wu, Z., Wang, X., Jiang, Y.G., Ye, H., Xue, X.: Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In: *ACM MM* (2015) [17](#), [54](#)
- [266] Xiong, X., De la Torre, F.: Supervised descent method and its applications to face alignment. In: *CVPR* (2013) [9](#), [35](#)
- [267] Yamada, M., Sigal, L., Raptis, M.: No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation. In: *ECCV* (2012) [12](#)
- [268] Yang, J., Yan, R., Hauptmann, A.: Cross-domain video concept detection using adaptive SVMs. In: *International Conference on Multimedia* (2007) [12](#), [24](#)
- [269] Yang, P., Liu, Q., Metaxas, D.: Boosting encoded dynamic features for facial expression recognition. *Pattern Recognition Letters* **30**(2), 132–139 (2009) [9](#)
- [270] Yang, P., Liu, Q., Metaxas, D.N.: Exploring facial expressions with compositional features. In: *CVPR* (2010) [10](#)
- [271] Yang, S., Rudovic, O., Pavlovic, V., Pantic, M.: Personalized modeling of facial action unit intensity. In: *Advances in Visual Computing*, pp. 269–281 (2014) [8](#), [12](#), [13](#), [18](#), [45](#), [49](#), [50](#), [107](#)
- [272] Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through

-
- deep visualization (2015) [59](#)
- [273] Yu, X., Zhang, S., Yu, Y., Dunbar, N., Jensen, M., Burgoon, J.K., Metaxas, D.N.: Automated analysis of interactional synchrony using robust facial tracking and expression recognition. In: AFGR (2013) [19](#)
- [274] Yuan, J., Liu, Z., Wu, Y.: Discriminative video pattern search for efficient action detection. TPAMI **33**(9), 1728–1743 (2011) [79](#), [80](#), [81](#), [90](#), [92](#), [100](#)
- [275] Zafeiriou, S., Pitas, I.: Discriminant graph structures for facial expression recognition. TMM **10**(8), 1528–1540 (2008) [13](#)
- [276] Zen, G., Sangineto, E., Ricci, E., Sebe, N.: Unsupervised domain adaptation for personalized facial emotion recognition. In: ICMI (2014) [13](#)
- [277] Zeng, J., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Xiong, Z.: Confidence preserving machine for facial action unit detection. In: ICCV (2015) [xi](#), [xviii](#), [11](#), [12](#), [13](#), [18](#), [56](#), [57](#), [58](#), [60](#)
- [278] Zeng, J., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Xiong, Z.: Confidence preserving machine for facial action unit detection. TIP **25**(10), 4753–4767 (2016) [xi](#), [xviii](#), [11](#), [12](#)
- [279] Zhang, A., **Chu, W.S.**, De la Torre, F., Hodgins, J.K.: Multi-class selective transfer machine and its application to human activity recognition. In: In submission (2017) [xii](#), [xviii](#), [27](#), [28](#), [45](#), [107](#)
- [280] Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: CVPR (2006) [12](#), [32](#), [33](#)
- [281] Zhang, X., Mahoor, M.H., Mavadati, S.M., Cohn, J.F.: A lp-norm MTMKL framework for simultaneous detection of multiple facial action units. In: WACV (2014) [14](#)
- [282] Zhang, X., Yin, L., Cohn, J.F., Canavan, S., Reale, M., Horowitz, A., Liu, P.: A high-resolution spontaneous 3d dynamic facial expression database. In: AFGR (2013) [v](#), [xvi](#), [15](#), [53](#), [58](#), [78](#), [108](#)
- [283] Zhang, X., Yin, L., Cohn, J.F., Canavan, S., Reale, M., Horowitz, A., Liu, P., Girard, J.M.: Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. Image and Vision Computing **32**(10), 692–706 (2014) [xiii](#), [xvi](#), [xvii](#), [60](#), [61](#), [69](#), [70](#), [71](#), [72](#), [75](#), [76](#), [78](#)
- [284] Zhao, G., Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. TPAMI **29**(6), 915–928 (2007) [9](#), [10](#)
- [285] Zhao, K., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Zhang, H.: Joint patch and multi-label learning for facial action unit detection. In: CVPR (2015) [xi](#), [xviii](#), [xix](#), [11](#), [14](#), [50](#), [56](#), [57](#), [58](#), [60](#)
- [286] Zhao, K., **Chu, W.S.**, De la Torre, F., Cohn, J.F., Zhang, H.: Joint patch and multi-label learning for facial action unit and holistic expression recognition. TIP (2016) [xi](#), [xviii](#), [11](#), [14](#)
- [287] Zhao, K., **Chu, W.S.**, Zhang, H.: Deep region and multi-label learning for facial action unit detection. In: CVPR (2016) [xviii](#), [xix](#)
- [288] Zhao, X., Dellandréa, E., Zou, J., Chen, L.: A unified probabilistic framework for automatic 3D facial expression analysis based on a Bayesian belief inference and statistical feature models. Image and Vision Computing **31**(3), 231–245 (2013) [16](#)
- [289] Zheng, Y., Gu, S., Tomasi, C.: Detecting motion synchrony by video tubes. In: ACM MM (2011) [18](#)
- [290] Zhong, L., Liu, Q., Yang, P., Huang, J., Metaxas, D.N.: Learning multiscale active facial patches for expression analysis. IEEE Transactions on Cybernetics (99) (2014) [9](#), [10](#), [14](#)
- [291] Zhou, F., De la Torre, F., Hodgins, J.K.: Hierarchical aligned cluster analysis for temporal clustering of human motion. TPAMI **35**(3), 582–596 (2013) [18](#), [90](#)

- [292] Zhou, F., De la Torre, F., Cohn, J.F.: Unsupervised discovery of facial events. In: CVPR (2010) **9**, **19**
- [293] Zhu, S., Mumford, D.: A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision* **2**(4), 259–362 (2006) **79**
- [294] Zhu, X.: Semi-supervised learning. In: C. Sammut, G. Webb (eds.) *Encyclopedia of Machine Learning*, pp. 892–897 (2010) **13**
- [295] Zhu, Y., De la Torre, F., Cohn, J.F., Zhang, Y.J.: Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection. *IEEE Trans. on Affective Computing* **2**, 79–91 (2011) **56**
- [296] Zhu, Y., De la Torre, F., F. Cohn, J., Zhan, Y.J.: Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior. *TAFFC* **2**(2), 79–91 (2011) **9**, **10**, **11**