

Selective Transfer Machine for Personalized Facial Expression Analysis

Wen-Sheng Chu, Fernando De la Torre and Jeffrey F. Cohn

Abstract—Automatic facial action unit (AU) and expression detection from videos is a long-standing problem. The problem is challenging in part because classifiers must generalize to previously unknown subjects that differ markedly in behavior and facial morphology (e.g., heavy versus delicate brows, smooth versus deeply etched wrinkles) from those on which the classifiers are trained. While some progress has been achieved through improvements in choices of features and classifiers, the challenge occasioned by individual differences among people remains. Person-specific classifiers would be a possible solution but for a paucity of training data. Sufficient training data for person-specific classifiers typically is unavailable. This paper addresses the problem of how to *personalize* a generic classifier without additional labels from the test subject. We propose a transductive learning method, which we refer as a Selective Transfer Machine (STM), to personalize a generic classifier by attenuating person-specific mismatches. STM achieves this effect by simultaneously learning a classifier and re-weighting the training samples that are most relevant to the test subject. We compared STM to both generic classifiers and cross-domain learning methods on four benchmarks: CK+ [44], GEMEP-FERA [67], RU-FACS [4] and GFT [57]. STM outperformed generic classifiers in all.

Index Terms—Facial expression analysis, personalization, domain adaptation, transfer learning, Support Vector Machine (SVM).

1 INTRODUCTION

AUTOMATIC facial AU detection confronts a number of challenges. These 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 [67].

To address these challenges, previous work has focused on identifying optimal feature representations and classifiers. Interested readers may refer to [20], [46], [49], [56] 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. 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. 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 propose that impaired generalizability occurs in part because of individual differences among subjects. Fig. 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. 2(a), they could be very difficult to separate without overfitting. When the data are interpreted as subjects in Fig. 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 [28].

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:

- Wen-Sheng Chu, Fernando De la Torre, and Jeffrey F. Cohn are with the Robotics Institute, Carnegie Mellon University. Jeffrey F. Cohn is also with the Department of Psychology, University of Pittsburgh.
- 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

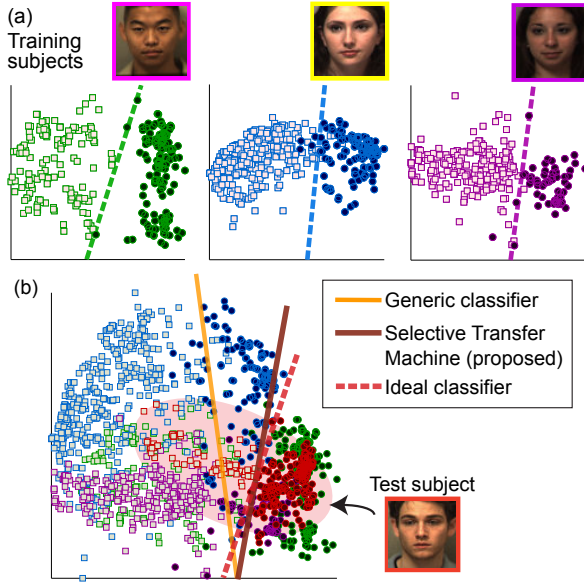


Fig. 1. An illustration of the proposed Selective Transfer Machine (STM): (a) 2D PCA projection of positive (squares) and negative (circles) samples for a given AU (in this case AU 12 or lip-corner raiser) for 3 subjects. An ideal classifier separates AU 12 nearly perfectly for each subject. (b) A generic 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.

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 paper is organized as follows. Sec. 2 reviews related work. Secs. 3–5 describes the STM model, optimization algorithm, and theoretical rationale. Sec. 6 introduces L-STM, an STM extension that utilizes labeled test data. Sec. 7 considers similarities and differences between STM and related methods. Sec. 8 evaluates STM and alternatives for AU and holistic expression detection. Sec. 9 concludes the paper with remarks and future work.

2 RELATED WORK

Our approach lies at the intersection between facial expression analysis and cross-domain adaptation. Below we briefly discuss each in turn.

2.1 Facial expression analysis

Automatic facial expression analysis entails at least three steps: Face tracking and registration, feature extraction, and learning classifiers. This section reviews recent advances in each.

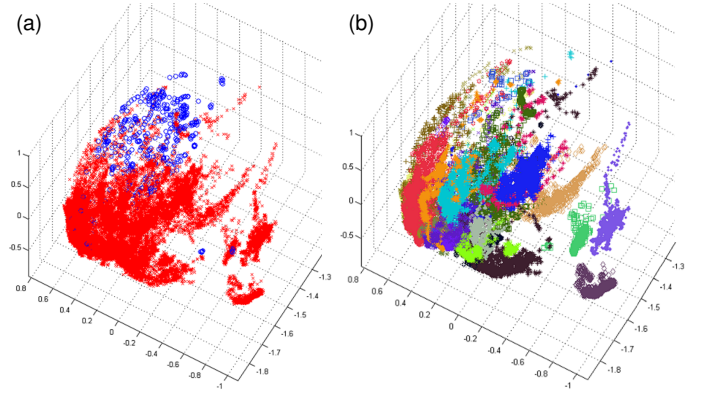


Fig. 2. Visualization of samples from the RU-FACS dataset [4] in 3D eigenspace: colors/markers indicate different (a) positive/negative classes, and (b) subjects (best viewed in color).

Tracking and registration: Tracking and registration of non-rigid facial features is a long-standing problem in computer vision. The goal of tracking is to detect facial landmarks (*e.g.*, eyes) in each frame. For facial landmark detections, Parametrized Appearance Models (PAM) are among the most popular methods. PAM include the Lucas-Kanade method [43], Active Appearance Models (AAM) [18], [47], Constrained Local Models (CLM) [15], and, more recently, Zface [34] and Supervised Descent Method [74]. Once facial landmarks are located, the registration step aims to align the face image to remove 3D rigid head motion, so features can be geometrically normalized. A similarity transformation [20], [61], [86] registers faces with respect to an averaged face. A Delaunay triangulation can be also applied with a backward piecewise affine warping to extract features in areas not explicitly tracked. This two-step registration proves to preserve better shape variation in appearance than by geometric normalization alone.

Feature extraction: With advances in tracking and registration, there has been renewed emphasis on biologically inspired features and temporal variation. As summarized in Table 1, current approaches to feature extraction may be broadly divided into four types: *geometric*, *appearance*, *dynamic*, and *fusion*. Geometric features contain information about shape and locations of permanent facial features, such as eyes or nose. Standard approaches rely on detecting fiducial facial points [45], a connected face [61], landmark coordinates [15], or face component shape parameterization [45]. Geometric features have performed well for many but not all AU detection tasks. They have difficulty detecting subtle expressions and are highly vulnerable to registration error [16].

Appearance features, which often are biologically inspired, afford increased robustness to tracking and registration error. Appearance features represent skin texture and its permutations and have been widely applied to facial expression analysis. Representative methods include SIFT [86], DAISY [86], Gabor jets [4], LBP [35], [84], Bag-of-Words model [60], [61], compositional [77] and others [72]. *Dynamic* features, a newly popular technique, encodes temporal information during the feature extraction stage. Examples include optical flow [32], bag of temporal words [62], volume LBP/LPQ [82], Gabor motion energy [73], and others. *Fusion* approaches incorporate multiple features, *e.g.*, Multiple Kernel Learning (MKL) [58], and have yet to prove superior to other approaches [67].

Classifiers: Two main trends have been pursued when designing classifiers for facial expression analysis, as summarized in Table 2. One trend, *static modeling*, typically tackles the

TABLE 1
Representative feature extraction methods

Type	Feature	Year	Reference
Geometric	Shape model parametrization	2012	[45]
	Geometry of facial components	2010	[85]
	Landmark locations	2006	[45]
Appearance	Active facial patches	2012	[84]
	SIFT/DAISY	2011	[86]
	Discrete Cosine Transform (DCT)	2011	[27]
	Local Phase Quantization (LPQ)	2011	[35]
	Local Binary Patterns (LBP)	2009	[59], [67]
	Hist. of Oriented Gradient (HOG)	2009	[48]
	Gabor	2006	[4], [41]
	Raw pixels	2000	[37]
Dynamic	Longitudinal expression atlases	2012	[33]
	Gabor motion energy	2010	[73]
	Bag of Temporal Words (BoTW)	2010	[61]
	Volume LBP (LBP-TOP)	2007	[82]
	Optical flow	2005	[32]
Fusion	Multiple feature kernels	2012	[58]

problem as discriminative classification and evaluates each frame independently. Representative approaches include Neural Network [38], Adaboost [4], SVMs [45], [61], [83], and Deep Networks [42]. Due to lack of temporal consistency, static models tend to produce non-smooth results. To address this issue, *temporal modeling*, the other trend, captures temporal transition between contiguous frames. For instance, Dynamic Bayesian Network (DBN) with appearance features [65] was proposed to model AU co-occurrence. Other variants of DBN include Hidden Markov Models [59] and Conditional Random Fields (CRF) [9], [68]. As an alternative, Simon *et al.* [61] proposed a structural-output SVM that detects AUs as temporal segments. To model relations between segments, Rudovic *et al.* [52] considered ordinal information in CRF. More recently, Ding *et al.* [21] proposed a hybrid approach that integrates frame-based, segment-based, and transition-based tasks in a sequential order. Interested readers are referred to [20], [46], [49], [56], [67] for more complete surveys.

Common to all these approaches is the assumption that training and test data are drawn from the same distribution. However, as Fig. 2 shows, they could suffer from individual differences, causing poor generalizability to an unseen subject. STM makes no such assumption. Instead, it seeks a personalized classifier by re-weighting training samples according to their distribution mismatch with test samples. Several studies merged into this direction could be found in [55], [78]–[80].

2.2 Cross-domain adaptation

Our approach is motivated by an increasing concern about dataset shift in the object detection literature. In real-world data, labels of interest often occur infrequently and features vary markedly between and within datasets. These factors contribute to significant biases in object categorization [66]. Saenko *et al.* [40], [54] proposed to reduce the discrepancy between features by learning metric transformation. Aytaar and Zisserman [2] regularized the training of a new object class by transferring pre-learned models. Chattopadhyay *et al.* [12] proposed to learn a combination of source classifiers that matches the target labels. Because these techniques use a supervised approach in which one or more labeled instances are required from the target domain, they are ill-suited to new domains or subjects for which no prior knowledge is

TABLE 2
Representative classifiers

Type	Classifier	Year	Reference
Static	Deep Networks	2013	[42]
	Support Vector Machine (SVM)	2007	[45]
	AdaBoost	2005	[4]
	Neural Network (NN)	2005	[38]
Temporal	Conditional Random Field (CRF)	2009	[9]
	Gaussian process	2009	[13]
	Dynamic Bayesian Network (DBN)	2007	[65], [70]
	Isomap embedding	2006	[10]
Hybrid	Cascade of Tasks (CoT)	2013	[21]

available. In contrast, our approach is unsupervised and thus better geared to the generalization to new domains or subjects.

Closer to our approach is a special case in unsupervised domain adaptation known as *covariate shift* [63]. In covariate shift, train and test domains follow different distributions but the label distributions remain the same. Given a source and a target domain, Domain Invariant Projection (DIP) [3] finds a domain-invariant space in which training and test data share a similar distribution. Similarly, Subspace Alignment (SA) [25] represents each domain as a subspace and then learns a mapping function that aligns the sources to the target one. However, learning the projection or mapping is unsupervised, and thus it is unclear how source labels can be incorporated.

On the other hand, Dudík *et al.* [24] infer the re-sampling weights through maximum entropy density estimation without target labels. Maximum Mean Discrepancy (MMD) [5] measures the discrepancy between two different distributions in terms of expectations of empirical samples. Without estimating densities, Transductive SVM (T-SVM) [36] simultaneously learns a decision boundary and maximizes the margin in the presence of unlabeled patterns. Domain adaptation SVM [6] extends T-SVM by progressively adjusting the discriminant function toward the target domain. SVM-KNN [81] 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. STM learns to re-weight training instances, which reduces the influence of irrelevant data.

Considering distribution mismatch, Kernel Mean Matching (KMM) [31] directly infers re-sampling weights by matching training and test distributions. Following this idea, Yamada *et al.* [75] estimated relative importance weights and learned from re-weighted training samples for 3D human pose estimation. See [50] for further review. These methods take a two-step approach that first estimates the sampling weights and then trains a re-weighted classifier or regressor. In contrast, STM jointly optimizes both the sampling weights and the classifier parameters and hence preserves the discriminant property of the new decision boundary.

3 SELECTIVE TRANSFER MACHINE (STM)

This section describes the proposed Selective Transfer Machine (STM) for personalizing a generic classifier. Unlike previous cross-domain methods [2], [22], [39], [76], STM requires no labels from a test subject. We will use Support Vector Machine (SVM) as classifier due to its popularity for AU detection [15], [35], [61].

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:

$$g(f, \mathbf{s}) = \min_{f, \mathbf{s}} R_f(\mathcal{D}^{\text{tr}}, \mathbf{s}) + \lambda \Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}), \quad (1)$$

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:

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

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 (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 [11], 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 penalized SVM:

$$R_{\beta}(\mathcal{D}^{\text{tr}}, \mathbf{s}) = \frac{1}{2} \beta^\top \mathbf{K} \beta + C \sum_{i=1}^{n_{\text{tr}}} s_i L^p(y_i, \mathbf{k}_i^\top \beta), \quad (3)$$

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 [11]. Details are given in Sec. 4.

Distribution mismatch: The second term in STM, $\Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}})$, 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.*, [64]). However, this strategy could be prone to error while taking the ratio of estimated densities [64].

Here we adopt the Kernel Mean Matching (KMM) [31] method to reduce the difference between the means of the training

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.

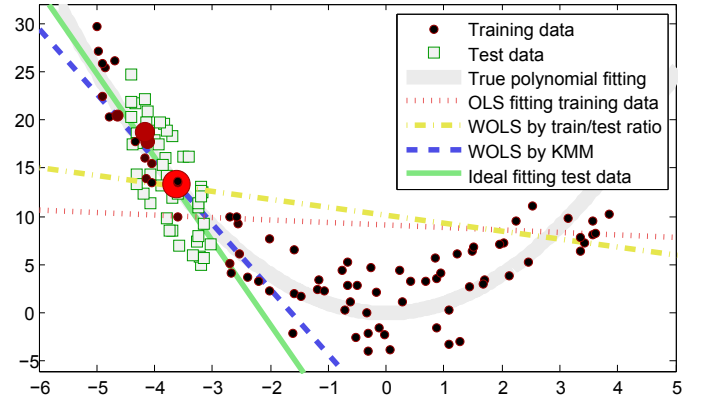


Fig. 3. Fitting a line to a quadratic function using KMM and other re-weighting methods. The larger size (more red) of training data, the more weight KMM adopted. As can be observed, KMM puts higher weights in the training samples closer to the test ones. Compared to standard OLS or WOLS, KMM allows to better approximation for the test data.

and the test distributions in the Reproducing Kernel Hilbert Space \mathcal{H} . KMM computes the instance re-weighting s_i that minimizes:

$$\Omega_{\mathbf{s}}(\mathbf{X}^{\text{tr}}, \mathbf{X}^{\text{te}}) = \left\| \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} s_i \varphi(\mathbf{x}_i^{\text{tr}}) - \frac{1}{n_{\text{te}}} \sum_{j=1}^{n_{\text{te}}} \varphi(\mathbf{x}_j^{\text{te}}) \right\|_{\mathcal{H}}^2. \quad (4)$$

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 (4) can be rewritten as a quadratic programming (QP):

$$\begin{aligned} \min_{\mathbf{s}} \quad & \frac{1}{2} \mathbf{s}^\top \mathbf{K} \mathbf{s} - \boldsymbol{\kappa}^\top \mathbf{s}, \\ \text{s. t.} \quad & s_i \in [0, B], \left| \sum_{i=1}^{n_{\text{tr}}} s_i - n_{\text{tr}} \right| \leq n_{\text{tr}} \epsilon, \end{aligned} \quad (5)$$

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 constraint ensures the weighted samples to be close to a probability distribution [31]. Observe in (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 [64]. Fig. 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.

4 OPTIMIZATION FOR STM

To solve Eq. (1), we adopt the Alternate Convex Search [26] that alternates between solving the decision function f and the selective coefficient \mathbf{s} . Note that the objective in (1) 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

Algorithm 1: Selective Transfer Machine

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 (6);
- 4 Update the decision function f and training loss ℓ^p by solving the penalized SVM in (2) or (3);
- 5 Infer test labels by $\mathbf{y}^* \leftarrow f(\mathbf{X}^{\text{te}})$

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:

$$\begin{aligned} \min_{\mathbf{s}} \quad & \frac{1}{2} \mathbf{s}^\top \mathbf{K} \mathbf{s} + \left(\frac{C}{\lambda} \ell^p - \kappa \right)^\top \mathbf{s} \\ \text{s. t.} \quad & 0 \leq s_i \leq B, n_{\text{tr}}(1 - \epsilon) \leq \sum_{i=1}^{n_{\text{tr}}} s_i \leq n_{\text{tr}}(1 + \epsilon). \end{aligned} \quad (6)$$

Since $\mathbf{K} \succeq 0$ by definition, (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 (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.

Fig. 5 illustrates the iterative effects on a synthetic example for learning a personalized classifier. In `it#1`, the hyperplane estimated by KMM is unreliable because it is completely unsupervised. On the other hand, STM simultaneously considers training loss and the weightings, and thus encourages the associated training samples to be well classified. As can be observed, as the iterations proceed, the STM separation hyperplane approaches toward the ideal one for the target data.

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 (2) can be written as:

$$\nabla_{\mathbf{w}} = \mathbf{w} + 2C\mathbf{X}\mathbf{S}\mathbf{I}^0(\mathbf{X}^\top \mathbf{w} - \mathbf{y}), \quad (7)$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + 2C\mathbf{X}\mathbf{S}\mathbf{I}^0\mathbf{X}^\top, \quad (8)$$

where $\mathbf{S} = \text{diag}(\mathbf{s}) \in \mathbb{R}^{n_{\text{tr}} \times n_{\text{tr}}}$ denotes the re-weighting matrix, $\mathbf{y} \in \mathbb{R}^{n_{\text{tr}}}$ 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

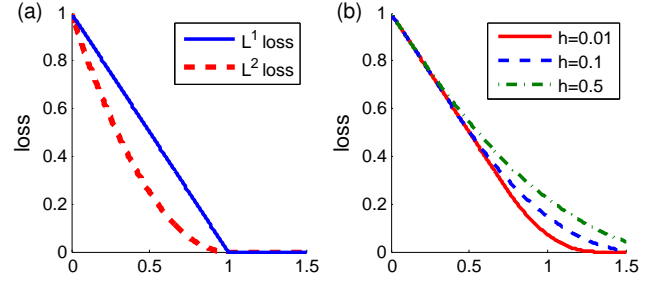


Fig. 4. Loss functions: (a) L^1 and L^2 loss, and (b) Huber loss.

being 0. Similarly, the gradient with respect to the expansion coefficient β in (3) can be derived as:

$$\nabla_{\beta} = \mathbf{K}\beta + 2C\mathbf{K}\mathbf{S}\mathbf{I}^0(\mathbf{K}\beta - \mathbf{y}), \quad (9)$$

$$\mathbf{H}_{\beta} = \mathbf{K} + 2C\mathbf{K}\mathbf{S}\mathbf{I}^0\mathbf{K}. \quad (10)$$

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 [11] 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:

$$L_H(a) = \begin{cases} 0 & \text{if } a > 1 + h, \\ \frac{(1+h-a)^2}{4h} & \text{if } |1-a| \leq h, \\ 1-a & \text{otherwise,} \end{cases} \quad (11)$$

where h is a parameter of choice. Fig. 4 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 [11], 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:

$$\nabla_{\mathbf{w}} = \mathbf{w} + \frac{C}{2h} \mathbf{X}\mathbf{S}\mathbf{I}^0 [\mathbf{X}^\top \mathbf{w} - (1+h)\mathbf{y}] - C\mathbf{X}\mathbf{S}\mathbf{I}^1 \mathbf{y}, \quad (12)$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + \frac{C}{2h} \mathbf{X}\mathbf{S}\mathbf{I}^0 \mathbf{X}^\top, \quad (13)$$

and for the penalized nonlinear SVM:

$$\nabla_{\beta} = \mathbf{K}\beta + \frac{C}{2h} \mathbf{K}\mathbf{S}\mathbf{I}^0 [\mathbf{K}\beta - (1+h)\mathbf{y}] - \mathbf{K}\mathbf{I}^1 \mathbf{y}, \quad (14)$$

$$\mathbf{H}_{\beta} = \mathbf{K} + \frac{C}{2h} \mathbf{K}\mathbf{S}\mathbf{I}^0 \mathbf{K}, \quad (15)$$

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} (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.

5 THEORETICAL RATIONALE

This section analyzes two important properties of STM, *bi-convexity* and *boundedness*, based on the techniques developed for biconvex optimization [30]. 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.

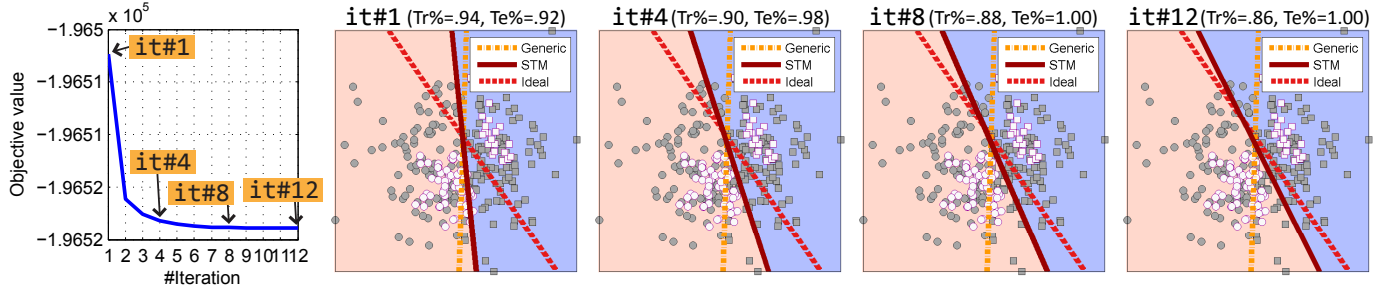


Fig. 5. Comparisons of a generic SVM, personalized STM, and ideal classifier for synthetic data. The left-most figure shows the convergence curve of the objective value where STM converges in 12 iterations. Figures *it#1*, *it#4*, *it#8*, *it#12* with training/test accuracy (Tr% and Te%) show the corresponding hyperplanes in at each iteration. Grey (shaded) dots denote training data, and white (unshaded) dots denote test data. Circles and squares denote positive and negative classes, respectively. Note that *it#1* indicates the result of KMM [31]. STM improves separation relative to generic SVM as early as the first iteration and converges toward the ideal hyperplane by the 12th iteration.

5.1 Properties of STM

We start by showing that STM is a biconvex problem.

Property 1. (Bi-convexity) *Selective Transfer Machine (STM) in (1) 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 $\mathbf{s} \in S \subseteq \mathbb{R}^{n_r}$, 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* [1]: $\min_{\mathbf{w}, \mathbf{s}} \{g(\mathbf{w}, \mathbf{s}) : (\mathbf{w}, \mathbf{s}) \in Z\}$. \square

Property 2. (Boundedness) *The STM optimization problem in Problem (1) is bounded from below.*

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) $\Omega_{\mathbf{s}}$ 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).

5.2 Algorithm

The following analysis mimics directly Sec. 4 in [30]. We present the key steps for proving the convergence and refer to more details on this style of proof in [30].

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 [71], 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. 4, 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. 8.5, 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}).

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
```

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 [30], the sequence $\{g(\mathbf{z}_t)\}_{t \in \mathbb{N}}$ converges to a limit real value. \square

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 [30] 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 [30]. \square

6 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.

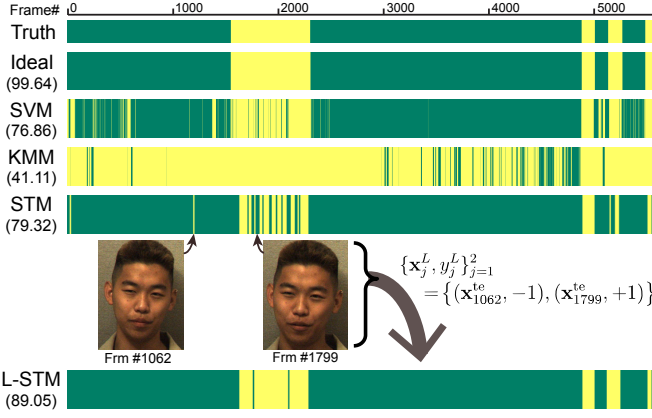


Fig. 6. Comparison of different methods on the RU-FACS dataset. Light yellow (dark green) indicates AU 12 presence (absence) of Subject 12. The numbers in the parentheses are F1 scores. Two misclassified frames of STM were chosen and fed into L-STIM with correct labels.

This section describes an inductive extension of STM, termed *L-STIM*, to adapt target labels for personalizing a classifier.

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

$$\min_{f, \mathbf{s}} R_f(\mathcal{D}^{tr}, \mathbf{s}) + \lambda \Omega_s(\mathbf{X}^{tr}, \mathbf{X}^{te}) + \lambda_L \Omega_L(\mathcal{D}^L), \quad (16)$$

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 [69], 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 (16) is still a biconvex optimization problem, and thus the ACS algorithm can be directly applied.

We show that solving problem (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, 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:

$$\nabla_{\mathbf{w}} = \mathbf{w} + \hat{\mathbf{X}} \hat{\mathbf{S}} (\hat{\mathbf{X}}^\top \mathbf{w} - \hat{\mathbf{y}}), \quad (17)$$

$$\mathbf{H}_{\mathbf{w}} = \mathbf{I}_d + \hat{\mathbf{X}} \hat{\mathbf{S}} \hat{\mathbf{X}}^\top, \quad (18)$$

where $\hat{\mathbf{X}} = [\mathbf{X}^{tr} | \mathbf{X}^L]$ is the augmented set with labeled target data, $\hat{\mathbf{S}} = \begin{bmatrix} 2C\mathbf{S}\mathbf{I}^0 & 0 \\ 0 & \lambda_L \mathbf{I}_{n_L} \end{bmatrix}$ is the augmented re-weighting matrix, and $\hat{\mathbf{y}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{y}^L \end{bmatrix}$ 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-STIM allows users to add just a few frames to alleviate false detections significantly. Fig. 6 illustrates the benefits of L-STIM 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. The

TABLE 3
Compare STM with related transductive transfer learning methods

Methods	Importance re-weight	Weight refine	Convexity	Labeled target data
SVM-KNN [81]	×	×	NA	×
T-SVM [17]	×	×	non-convex	×
KMM [31]	✓	×	convex	×
DA-SVM [6]	×	✓	non-convex	×
DT-MKL [22]	×	×	jointly convex	optional
DAM [23]	×	×	convex	optional
STM (proposed)	✓	✓	bi-convex	optional

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

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-STIM with two misclassified frames from STM with correct labels. Using the label information on the target domain, L-STIM 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-STIM approaches the ideal classifier.

7 DISCUSSION OF RELATED WORK

A few related efforts use *personalized modeling* for facial expression analysis, e.g., AU intensity estimation [53]. STM differs from them in how it accomplishes personalization. Chang and Huang [8] introduced an additional face recognition module and trained a neural network on the combination of face identities and facial features. Romera-Paredes *et al.* [51] 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. Motivated by covariate shift [63], Chen *et al.* [14] 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 [31], which with proper tuning, as implied in [64], yields better estimation.

The most related work is *transductive transfer learning*, which seeks to address domain shift problems without target labels. Table 3 summarizes the comparison. DT-MKL [22] simultaneously minimizes the MMD criterion [5] and a multi-kernel SVM. DAM [23] 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 [36] and SVM-KNN [81], these methods treat training data uniformly. By contrast, KMM [31] 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. 4). In addition, STM uses training loss to refine instance weights in successive steps, thus being able to correct sub-optimal weights. DA-SVM [6] 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.

TABLE 4
Detailed content of different datasets

Datasets	#Subjects	#Videos	#Frames/video	Content	AU annotation	Expression annotation
CK+ [44]	123	593	~20	Neutral→peak	Per video	Per video
GEMEP-FERA [67]	7	87	20~60	Acting	Frame-by-frame	Per video
RU-FACS [4]	34	34	5000~8000	Interview	Frame-by-frame	—
GFT [57]	720	720	~60,000	Multi-person social interaction	Frame-by-frame	—

8 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 [28]. 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.

8.1 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 4 and illustrated in Fig. 7.

(1) **The extended Cohn-Kanade (CK+) dataset** [44] contains brief (approximately 20 frames on average) videos of posed and un-posed facial expressions of men and women of various ethnic 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** [67] 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** [4] 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** [57] 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

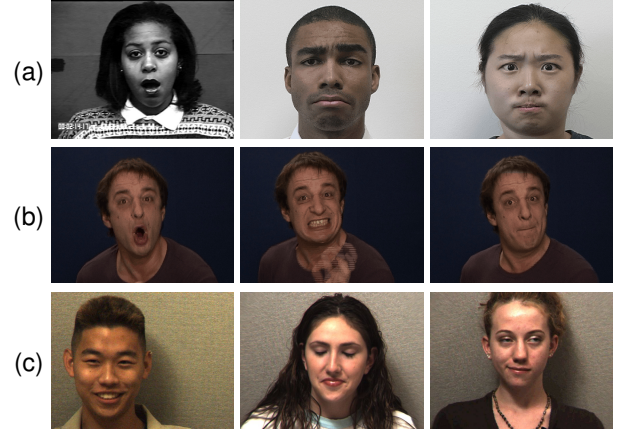


Fig. 7. Example images from (a) CK+ [44], (b) GEMEP-FERA [67], and (c) RU-FACS [4] datasets.

a total of approximately 302,000 frames. Head pose varies over a range of about plus/minus 15-20 degrees [28]. 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.

8.2 Settings

Face tracking & registration: For CK+, FERA, and GFT, 49 landmarks were detected and tracked using the Supervised Descent Method (SDM) [74]. For RU-FACS, we used available AAM detection and tracking of 68 landmarks. 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: A leave-one-subject-out protocol was used. For each AU, we iteratively chose one subject for test

TABLE 5
Within-subject AU detection with STM and PS classifiers

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

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.

8.3 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.

8.3.1 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 included in the training set. The GEMEP-FERA competition [67] defined PS in this way. An SVM trained with PS_2 (PS_2 -SVM) is sometimes considered to be a generic classifier (e.g., [45]). 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 5. 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. Table 6 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

TABLE 6
Selection percentage of STM for different subjects

	(a) Initialization								(b) Convergence						
	sub1	sub2	sub3	sub4	sub5	sub6	sub7		sub1	sub2	sub3	sub4	sub5	sub6	sub7
sub1	0.38	0.00	0.00	0.14	0.06	0.04	0.38	sub1	0.27	0.11	0.13	0.06	0.17	0.18	0.08
sub2	0.26	0.00	0.00	0.43	0.05	0.00	0.26	sub2	0.40	0.41	0.07	0.00	0.04	0.01	0.08
sub3	0.40	0.00	0.02	0.14	0.13	0.00	0.30	sub3	0.07	0.07	0.47	0.07	0.13	0.14	0.06
sub4	0.51	0.00	0.00	0.00	0.00	0.00	0.49	sub4	0.06	0.11	0.09	0.47	0.10	0.10	0.08
sub5	0.54	0.00	0.00	0.14	0.07	0.00	0.25	sub5	0.19	0.05	0.13	0.00	0.42	0.02	0.18
sub6	0.43	0.00	0.00	0.07	0.00	0.15	0.35	sub6	0.08	0.09	0.14	0.12	0.06	0.43	0.08
sub7	0.56	0.00	0.00	0.00	0.02	0.01	0.41	sub7	0.11	0.08	0.11	0.14	0.14	0.17	0.24

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.

8.3.2 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) [31], Domain Adaptation SVM (DA-SVM) [6], and Subspace Alignment (SA) [25]. 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 [23]. For baseline methods, we compared with linear SVMs and semi-supervised Transductive SVM (T-SVM) [17]. 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., [19], [40], [54]) 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 [31], 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 [17] since the original T-SVM [36] solves an integer programming and thus unsuitable 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 [7] as discussed in Sec. 4, $\tau = 0.5$ 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 [23], 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 [5]; 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.

TABLE 7
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

TABLE 8
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

Discussion: Tables 7 and 8 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 methods. 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. 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% [41], 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 [41] 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. 4).

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. 8.3.1). 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 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 7(a) 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. 8.3.1. When the number of subjects and training samples increase, as illustrated by the CK+ and the RU-FACS datasets in Tables 8 and 7, 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.

8.3.3 Cross-dataset AU detection

Detecting AUs across datasets is challenging because of differences in acquisition and participant characteristics and behavior. As shown in Fig. 7, 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. 8.3.1 and 8.3.2 have shown

TABLE 9

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

TABLE 10

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

(a) CK+	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
	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

(b) GEMEP-FERA	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

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 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 9 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 7 by 12.8% in AUC (76.5→86.3) and 33.7% in F1 (40.6→54.3), and in Table 9(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.

8.4 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+ [44] and FERA emotion subchallenge [67] for this experiment, and the same settings in Sec. 8.2, except for that the labels were replaced as holistic expressions. Similar to [67], 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. 8.3.2.

Table 10(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. 7). This effect becomes obvious when there is insufficient test data, such as this experiment. On the other hand, STM considers the labels for weight refinement and performed similarly as well as a generic SVM.

Table 10(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.

8.5 Analysis

8.5.1 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.*, [31]), can be interpreted as STM_w, as discussed in Sec. 4. To validate

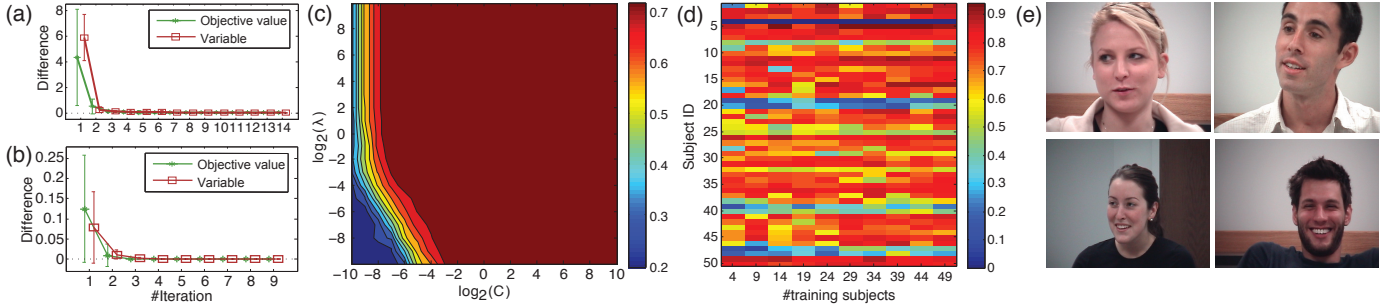


Fig. 8. Analysis experiments: (a)–(b) Objective and variable differences between iterations with initialization w_0 (STM_w) and s_0 (STM_s), respectively. (c) Performance versus parameter choices. (d) Per-subject F1 score v.s. # training subjects. (e) Exemplar images of the GFT dataset [57].

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 ($g(z_{t+1}) - g(z_t)$), and the absolute sum of variable difference ($\|z_{t+1} - z_t\|_1$). For the cases where STM took fewer iterations to converge, we set the difference of later iterations to 0.

Fig. 8(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. 5. 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.

8.5.2 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. 8.2, 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. 8(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 [64], 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 [64], with proper σ (kernel bandwidth) and standard CV, KMM consistently reaches lower error than the KL-divergence-based CV [64]. This serves as a justification for KMM’s ability to estimate importance weights.

8.5.3 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. 8.2. Fig. 9(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. 8.3.2, where a source domain with poor diversity was shown to 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 [57], 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. 8(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. 9(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. 8(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. 9(c), the F1 scores are markedly higher. The influence of the domain size becomes clear and replicates Fig. 9(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.

8.6 Discussion

In above experiments, we have evaluated STM against alternative methods in many scenarios: Within-subject (Sec. 8.3.1), across-subject (Sec. 8.3.2), across-dataset (Sec. 8.3.3), and holistic expression detection (Sec. 8.4). We also analyzed STM on its initialization order, and sensitivity to parameters and domain size (Sec. 8.5). 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

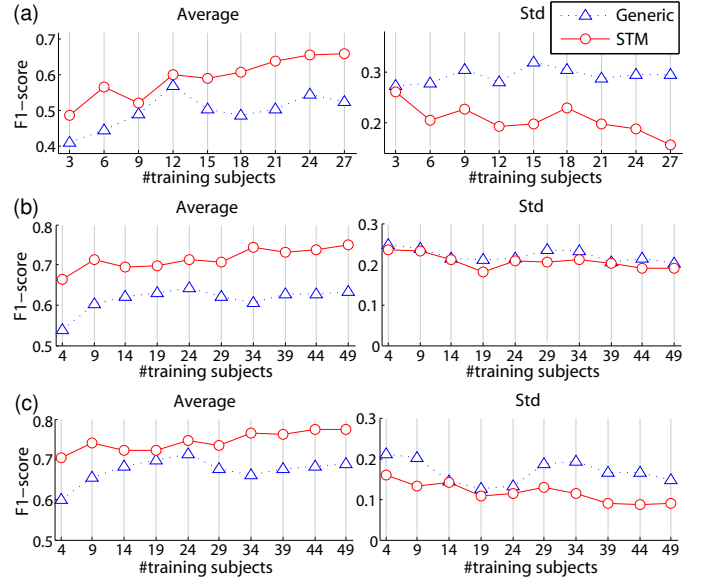


Fig. 9. Performance versus domain size: The averaged and standard deviation of F1 score on (a) RU-FACS. (b) and (c) show the F1 scores on the GFT dataset before and after removing the *outlier* subjects.

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 [29] 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).

9 CONCLUSION AND FUTURE WORK

Based on the observation of individuals differences, this paper proposed Selective Transfer Machine (STM) for personalized facial expression analysis. We showed that STM translates to a biconvex problem, and proposed an alternate 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 a test one. Experiments using within-subject, cross-subject, and cross-dataset scenarios revealed two insights: (1) Training data matter unevenly for test data, and (2) extending the variety of training subjects brings more value in improving performance.

It is worth noting that STM can be extended to other classifiers with convex decision functions and losses, such as logistic regression. This is a direct outcome of Property 1 in Sec. 5.1. 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. 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.

Acknowledgments: The authors would like thank many anonymous reviewers for constructive feedback. Research reported in this paper was supported in part by the National Institutes

of Health (NIH) under Award Number R01MH096951, the National Science Foundation (NSF) under the grant RI-1116583, and Army Research Laboratory Collaborative Technology Alliance Program under cooperative agreement W911NF-10-2-0016. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH or the NSF.

REFERENCES

- [1] R. Aumann and S. Hart, “Bi-convexity and bi-martingales,” *Israel Journal of Mathematics*, vol. 54, no. 2, pp. 159–180, 1986.
- [2] Y. Aytar and A. Zisserman, “Tabula rasa: Model transfer for object category detection,” in *ICCV*, 2011.
- [3] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann, “Unsupervised domain adaptation by domain invariant projection,” in *ICCV*, 2013.
- [4] M. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, “Automatic recognition of facial actions in spontaneous expressions,” *Journal of Multimedia*, vol. 1, no. 6, pp. 22–35, 2006.
- [5] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, B. Schölkopf, and A. J. Smola, “Integrating structured biological data by kernel maximum mean discrepancy,” *Bioinformatics*, vol. 22, no. 14, pp. 49–57, 2006.
- [6] L. Bruzzone and M. Marconcini, “Domain adaptation problems: A dasvm classification technique and a circular validation strategy,” *PAMI*, vol. 32, no. 5, pp. 770–787, 2010.
- [7] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [8] C.-Y. Chang and V.-C. Huang, “Personalized facial expression recognition in indoor environments,” in *International Joint Conference on Neural Networks*, 2010.
- [9] K.-Y. Chang, T.-L. Liu, and S.-H. Lai, “Learning partially-observed hidden conditional random fields for facial expression recognition,” in *CVPR*, 2009.
- [10] Y. Chang, C. Hu, R. Feris, and M. Turk, “Manifold based analysis of facial expression,” *Image and Vision Computing*, vol. 24, no. 6, pp. 605–614, 2006.
- [11] O. Chapelle, “Training a support vector machine in the primal,” *Neural Computation*, vol. 19, no. 5, pp. 1155–1178, 2007.
- [12] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye, “Multi-Source domain adaptation and its application to early detection of fatigue,” *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, p. 18, 2012.
- [13] J. Chen, M. Kim, Y. Wang, and Q. Ji, “Switching gaussian process dynamic models for simultaneous composite motion tracking and recognition,” in *CVPR*, 2009.
- [14] J. Chen, X. Liu, P. Tu, and A. Aragonés, “Learning person-specific models for facial expression and action unit recognition,” *Pattern Recognition Letters*, vol. 34, no. 15, pp. 1964–1970, 2013.
- [15] S. W. Chew, P. Lucey, S. Lucey, J. Saraghi, J. F. Cohn, and S. Sridharan, “Person-independent facial expression detection using constrained local models,” in *Automatic Face & Gesture Recognition*, 2011.
- [16] S. W. Chew, P. Lucey, S. Lucey, J. Saraghi, J. F. Cohn, I. Matthews, and S. Sridharan, “In the pursuit of effective affective computing: The relationship between features and registration,” *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 4, pp. 1006–1016, 2012.
- [17] R. Collobert, F. Sinz, J. Weston, and L. Bottou, “Large scale transductive svms,” *Journal of Machine Learning Research*, vol. 7, pp. 1687–1712, 2006.
- [18] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *PAMI*, vol. 23, no. 6, pp. 681–685, 2001.
- [19] H. Daumé III, “Frustratingly easy domain adaptation,” in *Conference of the Association for Computational Linguistics (ACL)*, 2007.
- [20] F. De la Torre and J. F. Cohn, “Facial expression analysis,” *Visual Analysis of Humans: Looking at People*, p. 377, 2011.
- [21] X. Ding, W.-S. Chu, F. De la Torre, and J. F. Cohn, “Facial action unit event detection by cascade of tasks,” in *ICCV*, 2013.
- [22] L. Duan, I. W. Tsang, and D. Xu, “Domain transfer multiple kernel learning,” *PAMI*, vol. 34, no. 3, pp. 465–479, 2012.
- [23] L. Duan, D. Xu, and I. W. Tsang, “Domain adaptation from multiple sources: A domain-dependent regularization approach,” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 23, no. 3, pp. 504–518, 2012.
- [24] M. Dudík, R. E. Schapire, and S. J. Phillips, “Correcting sample selection bias in maximum entropy density estimation,” in *NIPS*, 2005.
- [25] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” in *ICCV*, 2013.
- [26] C. Floudas and V. Visweswaran, “A global optimization algorithm (gop) for certain classes of nonconvex nlp. theory,” *Computers & chemical engineering*, vol. 14, no. 12, pp. 1397–1417, 1990.
- [27] T. Gehrig and H. K. Ekenel, “A common framework for real-time emotion recognition and facial action unit detection,” in *CVPR Workshop*, 2011.
- [28] J. M. Girard, J. F. Cohn, L. A. Jeni, M. A. Sayette, and F. De la Torre, “Spontaneous facial expression in unscripted social interactions can be measured automatically,” *Behavior research methods*, 2014.
- [29] D. A. Goldberg, M. B. Goldberg, M. D. Goldberg, and B. M. Goldberg, “Obtaining person-specific images in a public venue,” 2009, uS Patent 7,561,723.
- [30] J. Gorski, F. Pfeuffer, and K. Klamroth, “Biconvex sets and optimization with biconvex functions: a survey and extensions,” *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.
- [31] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf, “Covariate shift by kernel mean matching,” *Dataset shift in machine learning*, pp. 131–160, 2009.
- [32] H. Gunes and M. Piccardi, “Affect recognition from face and body: early fusion vs. late fusion,” in *International Conference on Systems, Man and Cybernetics*, vol. 4, 2005, pp. 3437–3443.
- [33] Y. Guo, G. Zhao, and M. Pietikäinen, “Dynamic facial expression recognition using longitudinal facial expression atlases,” in *ECCV*, 2012.
- [34] L. A. Jeni, J. F. Cohn, and T. Kanade, “Dense 3d face alignment from 2d videos in real-time,” in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2015.
- [35] B. Jiang, M. F. Valstar, and M. Pantic, “Action unit detection using sparse appearance descriptors in space-time video volumes,” in *Automatic Face & Gesture Recognition*, 2011.
- [36] T. Joachims, “Transductive inference for text classification using support vector machines,” in *ICML*, 1999.
- [37] T. Kanade, J. F. Cohn, and Y. Tian, “Comprehensive database for facial expression analysis,” in *Automatic Face & Gesture Recognition*, 2000.
- [38] A. Kapoor and R. W. Picard, “Multimodal affect recognition in learning environments,” in *ACMMM*, 2005.
- [39] A. Khosla, T. Zhou, T. Malisiewicz, A. Efros, and A. Torralba, “Undoing the damage of dataset bias,” in *ECCV*, 2012.
- [40] B. Kulis, K. Saenko, and T. Darrell, “What you saw is not what you get: Domain adaptation using asymmetric kernel transforms,” in *CVPR*, 2011.
- [41] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett, “The computer expression recognition toolbox (CERT),” in *Automatic Face & Gesture Recognition*, 2011.
- [42] M. Liu, S. Li, S. Shan, and X. Chen, “AU-aware deep networks for facial expression recognition,” in *IEEE Conference on Automatic Face and Gesture Recognition*, 2013.
- [43] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *International Joint Conference on Artificial Intelligence*, 1981.
- [44] P. Lucey, J. F. Cohn, T. Kanade, J. Saraghi, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression,” in *CVPR Workshops*, 2010.
- [45] S. Lucey, A. B. Ashraf, and J. Cohn, “Investigating spontaneous facial action recognition through aam representations of the face,” *Face recognition*, pp. 275–286, 2007.
- [46] A. Martinez and S. Du, “A model of the perception of facial expressions of emotion by humans: Research overview and perspectives,” *Journal of Machine Learning Research*, vol. 13, pp. 1589–1608, 2012.
- [47] I. Matthews and S. Baker, “Active appearance models revisited,” *IJCV*, vol. 60, no. 2, pp. 135–164, 2004.
- [48] C. Orrite, A. Gañán, and G. Rogez, “Hog-based decision tree for facial expression classification,” in *Pattern Recognition and Image Analysis*, 2009.
- [49] M. Pantic and M. S. Bartlett, “Machine analysis of facial expressions,” *Face recognition*, pp. 377–416, 2007.
- [50] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009.
- [51] B. Romera-Paredes, M. S. Aung, M. Pontil, N. Bianchi-Berthouze, A. de C. Williams, and P. Watson, “Transfer learning to account for idiosyncrasy in face and body expressions,” in *AFGR*, 2013.
- [52] O. Rudovic, V. Pavlovic, and M. Pantic, “Kernel conditional ordinal random fields for temporal segmentation of facial action units,” in *ECCV Workshop*, 2012.

- [53] —, “Context-sensitive dynamic ordinal regression for intensity estimation of facial action units,” *TPAMI*, vol. 37, no. 5, pp. 944–958, 2015.
- [54] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in *ECCV*, 2010.
- [55] E. Sangineto, G. Zen, E. Ricci, and N. Sebe, “We are not all equal: Personalizing models for facial expression analysis with transductive parameter transfer,” in *ACM International Conf. on Multimedia*, 2014.
- [56] E. Sariyanidi, H. Gunes, and A. Cavallaro, “Automatic analysis of facial affect: A survey of registration, representation, and recognition,” *TPAMI*, vol. 37, no. 6, pp. 1113–1133, 2015.
- [57] M. A. Sayette, K. G. Creswell, J. D. Dimoff, C. E. Fairbairn, J. F. Cohn, B. W. Heckman, T. R. Kirchner, J. M. Levine, and R. L. Moreland, “Alcohol and group formation a multimodal investigation of the effects of alcohol on emotion and social bonding,” *Psychological Science*, vol. 23, no. 8, pp. 869–878, 2012.
- [58] T. Senechal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost, “Facial action recognition combining heterogeneous features via multi-kernel learning,” *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 4, pp. 993–1005, 2012.
- [59] L. Shang and K. Chan, “Nonparametric discriminant HMM and application to facial expression recognition,” in *CVPR*, 2009.
- [60] K. Sikka, T. Wu, J. Susskind, and M. Bartlett, “Exploring bag of words architectures in the facial expression domain,” in *ECCV Workshops*, 2012.
- [61] T. Simon, M. H. Nguyen, F. De La Torre, and J. F. Cohn, “Action unit detection with segment-based svms,” in *CVPR*, 2010.
- [62] J. Sivic and A. Zisserman, “Video google: A text retrieval approach to object matching in videos,” in *ICCV*, 2003.
- [63] M. Sugiyama, M. Krauledat, and K. Müller, “Covariate shift adaptation by importance weighted cross validation,” *Journal of Machine Learning Research*, vol. 8, pp. 985–1005, 2007.
- [64] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Buenau, and M. Kawane, “Direct importance estimation with model selection and its application to covariate shift adaptation,” in *NIPS*, 2007.
- [65] Y. Tong, W. Liao, and Q. Ji, “Facial action unit recognition by exploiting their dynamic and semantic relationships,” *PAMI*, vol. 29, no. 10, pp. 1683–1699, 2007.
- [66] A. Torralba and A. Efros, “Unbiased look at dataset bias,” in *CVPR*, 2011.
- [67] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer, “Meta-analysis of the first facial expression recognition challenge,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 42, no. 4, pp. 966–979, 2012.
- [68] M. F. Valstar and M. Pantic, “Fully automatic recognition of the temporal phases of facial actions,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 1, pp. 28–43, 2012.
- [69] T. Van Gestel, J. A. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. De Moor, and J. Vandewalle, “Benchmarking least squares support vector machine classifiers,” *Machine Learning*, vol. 54, no. 1, pp. 5–32, 2004.
- [70] Z. Wang, S. Wang, and Q. Ji, “Capturing complex spatio-temporal relations among facial muscles for facial expression recognition,” in *CVPR*, 2013.
- [71] R. E. Wendell and A. P. Hurter, “Minimization of a non-separable objective function subject to disjoint constraints,” *Operations Research*, vol. 24, no. 4, pp. 643–657, 1976.
- [72] J. Whitehill, M. S. Bartlett, and J. R. Movellan, “Automatic facial expression recognition,” *Social Emotions in Nature and Artifact*, vol. 88, 2013.
- [73] T. Wu, M. S. Bartlett, and J. Movellan, “Facial expression recognition using gabor motion energy filters,” in *CVPR Workshop*, 2010.
- [74] X. Xiong and F. De la Torre, “Supervised descent method and its applications to face alignment,” in *CVPR*, 2013.
- [75] M. Yamada, L. Sigal, and M. Raptis, “No bias left behind: Covariate shift adaptation for discriminative 3d pose estimation,” in *ECCV*, 2012.
- [76] J. Yang, R. Yan, and A. Hauptmann, “Cross-domain video concept detection using adaptive svms,” in *International Conference on Multimedia*, 2007.
- [77] P. Yang, Q. Liu, and D. N. Metaxas, “Exploring facial expressions with compositional features,” in *CVPR*, 2010.
- [78] S. Yang, O. Rudovic, V. Pavlovic, and M. Pantic, “Personalized modeling of facial action unit intensity,” in *Advances in Visual Computing*, 2014, pp. 269–281.
- [79] G. Zen, E. Sangineto, E. Ricci, and N. Sebe, “Unsupervised domain adaptation for personalized facial emotion recognition,” in *International Conference on Multimodal Interaction*.
- [80] J. Zeng, W.-S. Chu, F. De la Torre, J. F. Cohn, and Z. Xiong, “Confidence preserving machine for facial action unit detection,” in *ICCV*, 2015.
- [81] H. Zhang, A. C. Berg, M. Maire, and J. Malik, “Svm-knn: Discriminative nearest neighbor classification for visual category recognition,” in *CVPR*, 2006.
- [82] G. Zhao and M. Pietikainen, “Dynamic texture recognition using local binary patterns with an application to facial expressions,” *PAMI*, vol. 29, no. 6, pp. 915–928, 2007.
- [83] K. Zhao, W.-S. Chu, F. De la Torre, J. F. Cohn, and H. Zhang, “Joint patch and multi-label learning for facial action unit detection,” in *CVPR*, 2015.
- [84] L. Zhong, Q. Liu, P. Yang, B. Liu, J. Huang, and D. Metaxas, “Learning active facial patches for expression analysis,” in *CVPR*, 2012.
- [85] F. Zhou, F. De la Torre, and J. F. Cohn, “Unsupervised discovery of facial events,” in *CVPR*, 2010.
- [86] Y. Zhu, F. De la Torre, J. F. Cohn, and Y. J. Zhang, “Dynamic cascades with bidirectional bootstrapping for action unit detection in spontaneous facial behavior,” *IEEE Transactions on Affective Computing*, vol. 2, no. 2, pp. 79–91, 2011.



Wen-Sheng Chu received his B.S. and M.S. degrees in computer science from National Cheng Kung University, Tainan, Taiwan. Currently, he is working toward the Ph.D. degree at the Robotics Institute, Carnegie Mellon University. His research interests lie in the development and use of machine learning techniques for computer vision problems. He is a student member of the IEEE and a member of the Phi Tau Phi Scholastic Honor Society.



Fernando De la Torre is an Associate Research Professor in the Robotics Institute at Carnegie Mellon University. He received his B.Sc. degree in Telecommunications, as well as his M.Sc. and Ph. D degrees in Electronic Engineering from La Salle School of Engineering at Ramon Llull University, Barcelona, Spain in 1994, 1996, and 2002, respectively. His research interests are in the fields of Computer Vision and Machine Learning. Currently, he is directing the Component Analysis Laboratory (<http://ca.cs.cmu.edu>)

and the Human Sensing Laboratory (<http://humansensing.cs.cmu.edu>) at Carnegie Mellon University. He has over 150 publications in referred journals and conferences and is Associate Editor at IEEE TPAMI. He has organized and co-organized several workshops and has given tutorials at international conferences on component analysis.



Jeffrey F. Cohn is Professor of Psychology and Psychiatry at the University of Pittsburgh and Adjunct Professor of Computer Science at the Robotics Institute at Carnegie Mellon University. He leads interdisciplinary and inter-institutional efforts to develop advanced methods of automatic analysis and synthesis of facial expression and prosody; and applies those tools to research in human emotion, social development, non-verbal communication, psychopathology, and biomedicine. He has served as Co-Chair of the 2008 IEEE International Conference on Automatic Face and Gesture Recognition (FG2008), the 2009 International Conference on Affective Computing and Intelligent Interaction (ACII2009), the Steering Committee for IEEE International Conference on Automatic Face and Gesture Recognition, and the 2014 International Conference on Multimodal Interfaces (ACM 2014). He has co-edited special issues of the Journal of Image and Vision Computing and is a Co-Editor of IEEE Transactions in Affective Computing (TAC).