1-1-2015

Facial Expression Analysis via Transfer Learning

Xiao Zhang
University of Denver

Facial Expression Analysis via Transfer Learning

A Dissertation

Presented to

the Faculty of the Daniel Felix Ritchie School of Engineering and Computer Science

University of Denver

in Partial Fulfillment

of the Requirements for the Degree

of Doctor of Philosophy

by

Xiao Zhang

March 2015

Advisor: Dr. Mohammad H. Mahoor

Author: Xiao Zhang
Title: Facial Expression Analysis via Transfer Learning
Advisor: Dr. Mohammad H. Mahoor
Degree Date: March 2015

Abstract

Automated analysis of facial expressions has remained an interesting and challenging research topic in the field of computer vision and pattern recognition due to vast applications such as human-machine interface design, social robotics, and developmental psychology. This dissertation focuses on developing and applying transfer learning algorithms – multiple kernel learning (MKL) and multi-task learning (MTL) – to resolve the problems of facial feature fusion and the exploitation of multiple facial action units (AUs) relations in designing robust facial expression recognition systems. MKL algorithms are employed to fuse multiple facial features with different kernel functions and tackle the domain adaption problem at the kernel level within support vector machines (SVM). l_p-norm is adopted to enforce both sparse and non-sparse kernel combination in our methods. We further develop and apply MTL algorithms for simultaneous detection of multiple related AUs by exploiting their inter-relationships. Three variants of task structure models are designed and investigated to obtain fine depiction of AU relations. l_p-norm MTMKL and TD-MTMKL (Task-Dependent MTMKL) are group-sensitive MTL methods that model the co-occurrence relations among AUs. On the other hand, our proposed hierarchical multi-task structural learning (HMTSL) includes a latent layer to learn a hierarchical structure to exploit all possible AU inter-relations for AU detection. Extensive experiments on public face databases show that our proposed transfer learning methods have produced encouraging results compared to several state-of-the-art methods for facial expression recognition and AU detection.

Acknowledgements

First and foremost, I thank my research advisor, Dr. Mohammad H. Mahoor. Without his assistance and dedicated involvement in my research works, this dissertation would have never been accomplished. I would like to thank you very much for your support and supervision over the past four and half years.

I would also like to show gratitude to my committee members, Dr. Kimon Valavanis, Dr. Nathan Sturtevant and Dr. Michael Kinyon. Thanks for your comments and suggestions on my comprehensive proposal and this dissertation.

Getting through my dissertation required more than the support from university faculty, and I have many, many people to thank for listening to and, at times, having to tolerate me over the past years. I cannot begin to express my gratitude and appreciation for their kindness and friendship. The colleagues in my lab, Ms. Saba Bakhshi, Dr. S. Mohammad Mavadati, Mr. Ali Mollahosseini and Mr. Howard Finn have been unwavering in their personal and professional support during the time I spent at the University of Denver.

Most importantly, none of these could have happened without my family. My parents offered their encouragement through phone calls every week. To my wife, it would be an understatement to say that, as a couple, we have experienced some ups and downs together in the past years. Every time I was ready to quit, she did not let me give up. This dissertation stands as a testament to your unconditional love and encouragement.

Table of Contents

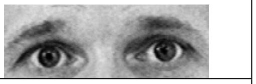
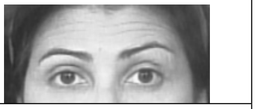
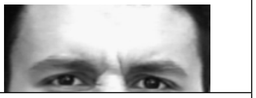
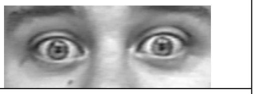
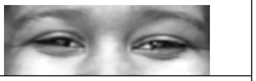
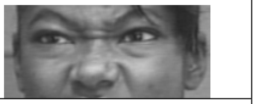
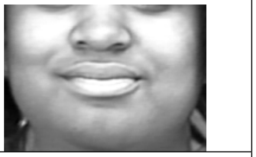
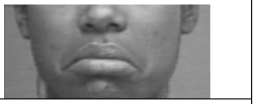
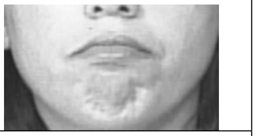
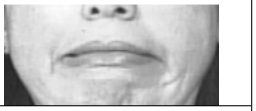
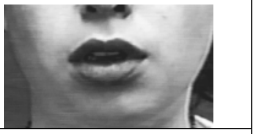
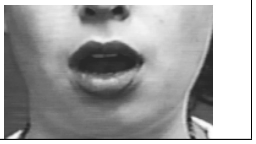
Chapter 1

Introduction

Facial expressions are the most important non-verbal visual channel used by humans in face-to-face communication [1]. Psychologists believe that facial expressions complete and reinforce verbal messages, and agree upon six basic facial emotions: joy, anger, fear, sadness, disgust, and surprise [2, 3] that are considered to be fundamental and common among different cultures. Moreover, in order to describe and quantify facial expressions, Paul Ekman and Wallace Friesen proposed the Facial Action Coding System (FACS) [2], which defines all possible and visually detectable facial muscle variations in terms of 44 action units (AUs). Table 1.1 lists the description of 12 AUs involved in our work.

Automated analysis of facial expressions in visual data is an interesting topic in the field of computer vision and pattern recognition. It has received great attention in recent years due to the vast number of applications including human machine interface design, robotics and developmental psychology. Although much progress has been made [4, 5], recognizing basic facial expressions and action units with a high accuracy still remained challenging due to the complexity, subtlety, and variations of human facial behaviors.

Table 1.1: AU description

AU	Description	Figure
AU1	Inner Brow Raiser	
AU2	Outer Brow Raiser	
AU4	Brow Lowerer	
AU5	Upper Lid Raiser	
AU6	Cheek Raiser	
AU9	Nose Wrinkler	
AU12	Lip Corner Puller	
AU15	Lip Corner Depressor	
AU17	Chin Raiser	
AU20	Lip stretcher	
AU25	Lips part	
AU26	Jaw Drop	

This dissertation focuses on applying transfer learning algorithms to resolve two major problems existed in facial expression analysis. One is the facial feature fusion problem. In the recognition of basic facial expressions, which is a multiclass classification problem, one type of facial features may not be distinguishable for all expressions whereas using another type of features may produce better results in several expressions. Since different features have different distributions, it is necessary to fuse multiple facial feature representations to increase the discriminative power of classifiers. However, it is usually difficult to represent a combination of features within the widely used single-kernel-based support vector machines (SVM) when considering the compatibility of different features domains. For the sake of this problem, we propose to apply multiple kernel learning (MKL) in the transfer learning methodology to fuse multiple facial features at the kernel level with SVM. The "transferring skill" of MKL is to optimally combine different kernel matrices calculated base on multiple features with multiple kernels. Within this framework, the problem of feature data representation through single type of features with a single kernel function in the canonical SVM is transferred to set the optimal value of kernel combination weights for multiple kernel matrices.

The other challenge is how to properly exploit the relations among facial AUs and basic expressions during facial expression analysis. According to the description in the FACS manual [6], there are some relationships among different AUs such as simultaneous presence in basic facial expressions. For examples, AU4 (brow lowerer) is usually co-occurred with AU1 (inner brow raiser) and AU2 (outer brow raiser) to generate negative expressions such as fear and sadness. AU6 (cheek raiser) is usually co-occurred with AU12 (lip corner puller) in the case of Duchenne smile [7]. These relationships are by their nature good resources for AU detection. However, almost all the existing AU detection approaches, based on either static [8, 9, 10] or dynamic modeling [11, 12, 13, 14], turn to recognize AUs or certain AU combinations separately without considering their inter-relations. In this dissertation, viewing the detection of each AU as a task, we apply multi-task learning (MTL) algorithms to simultaneously detection multiple AUs by properly modeling the AU inter-relations. As surveyed in [15], MTL encodes the idea of inductive transfer learning, and aims to learn one problem with the help of other related problems by properly modeling their related structures.

In summary the main contributions of our works are two-folded.

- We present a novel facial expression recognizer via MKL by extending the l_p-norm MKL algorithm into multiclass classification problem. Different types of facial features with multiple kernel functions are fused by adopting l_p-norm ($p \geq 1$) to obtain both sparse and non-sparse kernel combinations. For solving the optimization problem of our proposed method, we learn one kernel weight vector for each binary classifier in the multiclass-SVM. Compared to the SimpleMKL-based multiclass-SVM [16], which jointly learns the same kernel weight vector for all binary classifiers, our method has a better flexibility of selecting different kernel combinations and also reflects the contribution of each binary classifier to the whole objective function of MKL-based multiclass-SVM. We also comprehensively studied the impact of "p" on controlling the sparsity of the kernel combinations, and provide insight explanation of why our proposed method outperforms the state-of-the-art methods based on the discussion of the experimental results.

- We cast the AU detection problem into MTL frameworks, where given a specific facial image, multiple AUs are detected simultaneously by exploiting the relations of their discriminative hyperplanes in SVM. Moreover, we take the advantage of MKL to increase the discriminant power of the MTL classifiers by fusing different types of facial feature representations with multiple kernels. Three task structures are designed in our proposed methods to achieve proper modeling of AU inter-relations. l_p-norm multi-task multiple kernel learning (MTMKL) and task-dependent MTMKL (TD-MTMKL) are group-sensitive MTL methods for modeling AU inter-relations, where AUs are packaged into different groups via our pre-knowledge of AU co-occurrence relations. Whereas, hierarchical multi-task structure learning (HMTSL) is proposed to avoid of such pre-knowledge and utilize all possible AU inter-relations via a hierarchical model in HMTSL.

The remainder of this dissertation is organized as follows. Chapter 2 reviews the related works on facial expression analysis including feature representation and classifier design for basic expressions and AUs as well as the existing transfer learning algorithms including MKL and MTL. Chapter 3 introduces the public databases utilized in this dissertation for both basic expression recognition and AU detection. Chapter 4 presents the formulation of our proposed transfer learning framework, l_p-norm MKL-based Multiclass-SVM, for basic facial expression recognition. Experiments on three public face databases are presented and discussed based on the comparison with several state-of-the-art methods. Chapter 5 presents our proposed group-sensitive MTL methods for AU detection on four AU packages. Chapter 6 describes the designed hierarchical model and the optimization formulation for HMTSL. Experimental results on two face databases with posed and spontaneous AUs are shown and discussed in Chapter 5 and Chapter 6 based on the comparison with several state-of-the-art methods. Finally, Chapter 7 concludes the paper and envisions the future work.

Chapter 2

Related work

Automatic facial expression analysis has made good progresses in the last decade. For a detailed survey on the existing and state-of-the-art methods in this topic, we refer our reader to [4, 5]. Most recently, the Facial Expression Recognition and Analysis Challenge (FERA 2011) [17] outlined the evaluation protocol, the data used, and the results of a baseline method for facial action unit (AU) detection and expression recognition. Here, we briefly review some previous works on two main aspects of this challenge, facial feature extraction and classifier design. The transfer learning algorithms including MKL and MTL methods are also introduced.

2.1 Facial feature representation

Facial images are represented by extracting a set of features from registered images, where procrustes analysis is usually applied for image registration using several annotated facial landmark points such as in [18, 19]. Good features are those with small inner-class variations of facial expressions and large intra-class variations. Three categories of features are commonly seen in the literatures: geometric features, appearance features, and combination of geometric and appearance features.

Geometric representations usually utilize shapes and locations of facial components to model the face geometry. Chang et al. [20] learned a specific active shape model (ASM) [21] defined by 58 fiducial points to avoid incorrect matching due to non-linear image variations. Pantic et al. [8, 22] tracked a set of facial characteristic points around the mouth, eyes, eyebrows, nose, and chin. Some approaches combined geometric and appearance features (i.e., active appearance models (AAM) [23]) and applied them for facial expression recognition [24, 25]. AAM are statistical models that can provide good spatial information of key facial landmark points for valid examples. However, they are highly dependent of an accurate matching of the model to facial images, and usually need manual labor for their construction.

Appearance features are often used for representing facial textures such as wrinkles, bulges and furrows exhibited in facial expressions. Gabor wavelet analysis [26, 27] is one of the first appearance-based features used to represent the facial appearance variations. These features are usually applied to either the entire face or specific face regions. The computation of Gabor-wavelet representation is both time and memory intensive [18]. LBP operator was introduced as an effective appearance feature for facial image analysis [28, 29]. Shan et al. [18] achieved better results for facial expression recognition using LBP features compared to Gabor features. The most important properties of LBP features are their tolerance against illumination changes and computational simplicity. HOG features were firstly described in [30] for pedestrian detection, which count occurrences of gradient orientations in localized portions of an image. It has further been determined in [31] that when HOG features are combined with LBP descriptors, detection performances are improved. Recently, HOG features were used and revised to extract facial appearance and shape variations for facial expression recognition in [32, 33]. The invariance to geometric transformations and good description of edge orientations are the key advantages of HOG descriptor over other methods. In our work, we fuse the LBPH features with HOG features at kernel level within SVM classifiers and study its impact on facial expression recognition and AU detection.

2.2 Facial expression recognition

Several studies have evaluated different classifiers for facial expression recognition. Bartlett et al. [10] and Shan et al. [18] respectively performed systematic comparison of different techniques including AdaBoost, SVM and Linear Discriminant Analysis (LDA), and the best results were obtained by selecting a subset of facial features using AdaBoost and then sent to SVM for automatic expression labeling. Sebe et al. [34] evaluated 14 different classifiers like SVM, Bayesian Nets and Decision Trees and achieved the best classification results using k-nearest neighbor (kNN) algorithm. They also used voting algorithms such as bagging and boosting to improve the results of facial expression recognition.

In order to exploit the temporal information of facial behaviors, different methods have been presented for facial expression recognition in image sequences. Several early works [35, 36] attempted to track and recognize facial expressions over time based on optical flow methods. Hidden Markov Models (HMM) are widely used to model the temporal relations between consecutive facial behaviors [37, 38]. In [37], a multi-level HMM classifier was proposed to combine temporal information and automatically segment long video sequences. In their work, a Bayesian classifier was used for still images while a HMM classifier was applied to deal with emotion recognition in video sequences. However, as HMM can not model the dependencies among observed facial features, some other research groups applied Dynamic Bayesian networks (DBN) for facial expression classification. Zhang and Ji [39] exploited DBN with a multisensory information fusion strategy while in [40] a novel Bayesian temporal model was formulated to capture the dynamic facial expression transition in a manifold.

Recently, MKL algorithms were proposed to combine multiple kernels instead of using a single one in training kernel-based classifiers such as SVM. These methods, as surveyed in [41], have been applied to affective analysis and achieved better recognition results compared to single kernel SVM equipped with a single type of facial feature [42, 43]. The authors of [43] combined two types of facial features with two kernel functions, local Gabor binary pattern histograms (LGBPH) [44] with histogram intersection kernel and AAM coefficients with RBF kernel, and tuned the parameters of kernel functions during experiments on facial action unit detection. The SimpleMKL algorithm [16] was applied for solving the optimization problem of MKL in their work, which is a binary classification task.

Since different types of features can represent different information in facial images, by combining multiple features with different kernel functions in MKL framework, plenty of useful facial representations and kernel functions can be utilized simultaneously during classification. In this work, a novel MKL framework is presented for multiclass classification using SVM, and comprehensive study is conducted to evaluate the effect of our method on the application in facial expression recognition.

2.3 Facial action unit detection

AU detection is a binary classification problem. There are mainly two approaches in the literatures: one is the static modeling as presented in [8, 9, 10], where each face image is recognized separately by solving a binary discriminative classification problem; the other one is the dynamic modeling such as in [45, 12, 13], where video frames are segmented into subsequences to exploit the temporal behaviors based on a variant of dynamic models such as Hidden Markov Models (HMMs), and are usually described in terms of onset, apex and offset. However, almost all these methods turn to recognize AUs or certain AU combinations independently without considering the inter-relations among different AUs.

There are only a few studies in the literatures that exploit the relations among AUs in detecting them. The research group of Qiang Ji proposed to use Dynamic Bayesian Network (DBN) to model the AU inner-relations. In [46, 47, 48], a two-step processing technique was engaged, where the learned DBN model from training data was employed to infer the AU labels of video frames based on the output of adaboost SVM. In this framework, the DBN can be viewed as a reasoning module that post-processes the predicted AU labels from the previous module with adaboost classifiers. Thus, necessary theoretical discussions on how the two modules can cooperatively increase the overall detection performance and the impact of overfitting are needed. Further, unified frameworks [49, 50] were presented to learn more complete graphical models to combine the classification and inference steps together using probabilistic classifiers instead of SVM. However, the DBN-based classifiers and the Restricted Boltzmann Machines (RBMs) in [50] need some prior assumptions of samples' probability distribution models such as the Gaussian distribution function, which may not be accurate for real applications. In comparison, our proposed method is based on SVM classifiers, which are more robust since the convex optimization problems are defined to learn the maximum-margin hyperplanes between samples from different classes in the feature space. In contrast to the works in [46, 47, 48, 49], which model the AU relations based on probabilistic dependencies among the presence and absence of multiple AUs, our work aims to exploit the relations among the SVM classifiers that detect different AUs. That is, in our approach the detection of multiple AUs and the utilization of their intrinsic relations are conducted simultaneously.

2.4 Transfer learning algorithms

In this section, we introduce the general idea of two topics in the transfer learning framework – MKL and MTL, and review some famous methods proposed in the literatures.

MKL is proposed to deal with the domain adaption problem, which minimizes the data distribution mismatch between feature domains. For a detailed survey of MKL algorithms, we refer our readers to [41]. Usually, the parameters of kernel functions in canonical SVM classifiers are tuned during the training-validation experiments, and the parameters that result in the best classification rate on validation samples are applied to recognize the test samples. However, it is known that different kernels with different parameters correspond to different representation of features. Instead of trying to find which works the best, MKL-based SVM use a combination of them and define automatic learning methods to pick the optimal parameters. It is defined to learn both the decision boundaries between data from different classes and the kernel combination weights in a single optimization problem [51]. Therefore, features in different domains are transferred and fused at the kernel level in SVM.

Lanckriet et al. [51] considered a linear combination of basis kernels. By restricting the kernel combination weights to have nonnegative values, the authors formulated their algorithm to a Quadratically-Constrained Quadratic Program (QCQP), where the support vector coefficients and the kernel combination weights were jointly learned. Sonnenburg et al. [52] proposed a formulation of linearly combining kernels in the primal form of SVM. In their method, a l_1-norm restriction was used on the regularizer of the SVM objective function to enforce sparse kernel combinations. Later on, Rakotomamonjy et al. [16] proposed a modified regularizer with explicit kernel combination weights in their primal formulation. The authors exploited the weighted l_1-norm across different kernel spaces and the l_2-norm within each kernel space. This formulation named as SimpleMKL was then proved to be equivalent to the optimization problem of [52].

Instead of constraining kernel weights via l_1-norm regularization, Cortes et al. [53] studied the performance of l_2-norm for MKL, and found that the l_2-norm outperformed l_1-norm when larger sets of basis kernels were utilized. Sun et al. [54] proposed a new kernel evaluation technology to utilize both l_1 and l_2 norms in their Selective Multiple Kernel Learning (SMKL) method. SMKL obtained a sparse solution by a pre-selection procedure, and meanwhile preserved a subset of kernels with complementary information out of the entire set of basis kernels. Kloft et al. [55, 56] generalized these MKL algorithms and formulated l_p-norm MKL with arbitrary p ($p \geq 1$) to regularize over kernel combination coefficients in a binary classification problem. In our paper, we proposed l_p-norm MKL multiclass-SVM by extending the original method to a multi-class classification problems. Recently, Yan et al. [57] compared the performance of l_1-norm and l_2-norm MKL in the applications of image and video classification tasks, and concluded that the l_2-norm should be used as it carries complementary information resources. Following this idea, we apply our proposed method for facial expression recognition, and study the performance of nonsparse kernel combinations ($p > 1$) versus sparse ones ($p = 1$).

Multi-task learning is a transfer learning approach that learns multiple related problems simultaneously using a common representation. It seeks to improve the performance of a task with the help of other related tasks by properly leveraging information across the involved tasks in the learning process. The multi-class classification problem can be viewed as a special case in the MTL framework. Koby et al. [58] proposed an MTL method to solve the one-vs-the-rest multi-class classification problem. Several works have extended the kernel-based classifier SVM to the case of MTL. One set of algorithms, such as [59, 60], propose to present the task relations via the classification hyperplanes of SVM and can make closer the parameters of the hyperplanes for similar tasks. Whereas, other methods aim to learn the common structure among data from all the tasks. For example, the authors of [61] model the task relations by assuming that there only exists a small set of features shared across multiple tasks. The approach in [62] learns a shared feature map across all the tasks that can projects the input features of classifiers into a low-dimensional space with higher discriminative power. Our work presents a unified framework of the above two categories of MTL approaches, where task relations are encoded based on both SVM classification functions and a set of shared kernels for feature representation across all the tasks.

Recent works also explored the utilization of multiple kernel learning (MKL) for classification with multiple tasks. The authors of [63] model the task relations by defining meta-tasks. Each meta-task corresponds to a subset of all tasks, representing the common properties of the tasks within this subset. Without utilizing multiple kernel functions as usually done to fuse different types of features in MKL algorithms, a l_p-norm MKL solver [55] is employed for solving the proposed MTL problem. In [64, 65], multiple kernels are fused to enhance the discriminative power of MTL-based classifiers. However, these methods assume uniform kernel combinations and weight too much on the commonalities among multiple tasks without considering their diversities. Sharing exactly the same kernel combination weights might be too restrictive for weakly correlated tasks. Tang et al. [66] propose to simultaneously learn multiple kernels for multiple tasks. Similar to our methods, a specific kernel combination is assigned for each task, but the relations of SVM classification functions are not utilized for task structure modeling. We will compare this work with our method in the experiment section, and confirm the importance of modeling the relations of SVM hyperplanes in the application of facial AU detection.

Chapter 3

Face databases for facial expression analysis

In this chapter, we introduce the face databases utilized in this dissertation for basic expression recognition and action unit detection.

Facial expressions can be categorized into spontaneous expressions and posed expressions. Spontaneous expressions are those that occur in real life. Whereas, posed facial expressions are assumed to be artificially behaved and can differ markedly in configuration, intensity, and timing from spontaneous expressions. Especially, some facial actions that usually co-occur or are highly correlated in posed facial behaviors may rarely be seen in spontaneous ones. Therefore, in order to investigate the robustness of our proposed methods for facial expression analysis, we used four databases in our experimental works: the extended Cohn-Kanade (CK+) [67] database, the MMI database [68, 69] and the GEMEP-FERA data [17, 70] with posed expressions and the Denver Intensity of Spontaneous Facial Action (DISFA) database [71] with spontaneous expressions.

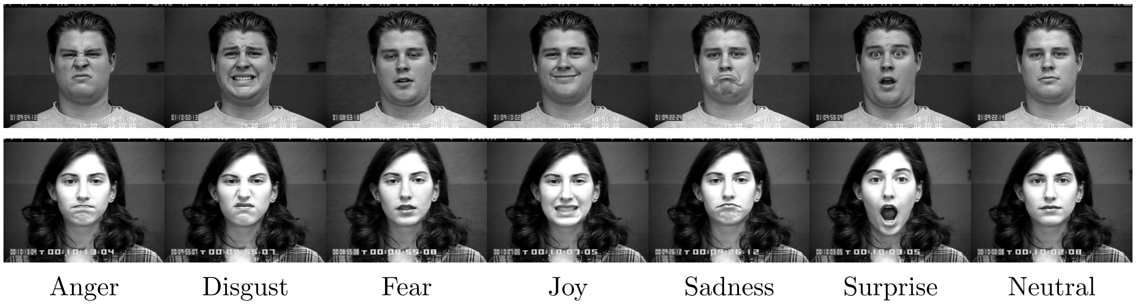


Figure 3.1: Sample images in the CK+ database

The three posed facial expression databases – CK+, MMI and GEMEP-FERA – were utilized for the experiments on basic facial expression recognition. Both within-database and cross-database tests were conducted to evaluate our proposed l_p-norm MKL-based multiclass-SVM. For AU detection, we used the CK+ database and the DISFA database to verify the performance of our proposed MTL-based methods including l_p-norm MTMKL, TD-MTMKL and HMTSL, which simultaneously detect multiple AUs by exploiting their inter-relations.

3.1 The CK+ database

The CK+ database is one of the most comprehensive face databases available in the research community. It consists of 593 image sequences from 123 subjects. The image sequences vary in duration from 10 to 60 frames and incorporate the neutral frame to peak formation including seven facial expressions: Anger, Disgust, Fear, Joy, Sadness, Surprise and Contempt as well as 30 AUs. All the images were digitized into 640 × 480 pixel arrays, and the X–Y coordinates of 68 landmark points were given for every image in the database. Figure 3.1 shows the sample images of the CK+ database with seven expressions.

In our work on AU detection, we used the first frame (neutral face) and the last three frames (peak frames) in each of the 593 image sequences, resulting in 2372 images. Since some of the image sequences do not necessarily represent six basic expressions and may just be a combination of various AUs, for basic facial expression recognition, we only used the images that are labeled as one of the six basic emotions including Anger, Disgust, Fear, Joy, Sadness and Surprise, resulting in 1236 images (135 Anger, 177 Disgust, 75 Fear, 207 Joy, 84 Sadness, 249 Surprise and 309 neutral faces). These images are selected from 309 image sequences with 106 subjects.

After converting the selected images to 8-bit gray-scale ones, we calculated the average X–Y coordinates of the located 68 landmark points among them. Then, each image was registered using a similarity transformation [72]. The transformation matrix was calculated between the X–Y coordinates of the 68 landmark points in that image and the average X–Y coordinates. Afterwards, we cropped the face region from each registered image based on the boundary described by its 68 landmark points, and resized them to 128×128 pixels. Histogram of oriented gradient (HOG) [30] and local binary pattern histogram (LBPH) [73] features with 8×8 windows and 59 bins in each window were separately extracted from each cropped facial image, and the size of each window is 16×16 pixels without overlap between windows. Further, for each feature category (LBPH and HOG) the PCA algorithm [74] was used for data dimensionality reduction to preserve 95% of the energy. The block diagram of our manipulation on image registration and feature extraction are shown in Figure 3.2.

3.2 The MMI database

The MMI facial expression database includes subjects from students and research staff members of both sexes aged 19-62 years old. It is a continually growing resource

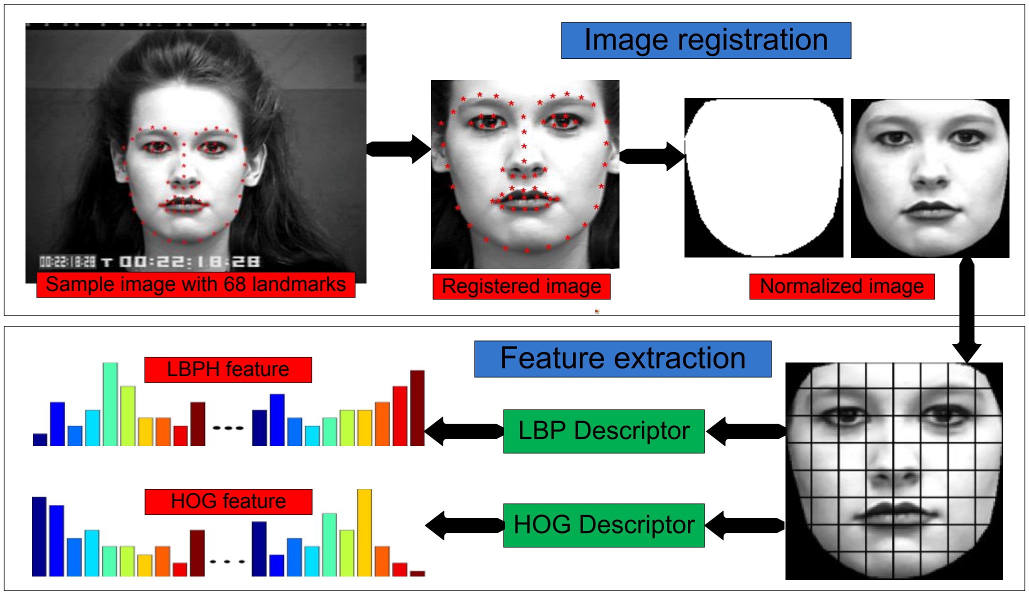


Figure 3.2: Facial feature extraction on the CK+ database

for AU and basic emotion recognition from face videos, of which the online search engine will facilitate researchers' selection of samples by setting different criteria. Figure 3.3 shows the sample images of the MMI database with seven expressions.

In our work, 209 sessions were chosen from the database. Our selection criteria were that for each selected session it could be labeled as one of the six basic emotions and contains frontal or near-frontal view of the participant's faces. The selected sessions were from 30 subjects. Facial images in each session were digitized into 720 × 576 pixels with 24-bit color values. Similar to our experiments on the CK+ database, for each selected session, the first frame and three peak frames were used for prototypic expression recognition resulting in 836 images (99 Anger, 96 Disgust, 87 Fear, 126 Joy, 96 Sadness, 123 Surprise and 209 neutral faces). Different from the CK+ databae, the locations of 68 landmark points on facial images are not provided in the MMI database. Therefore we apply the recently proposed facial feature tracking method in [75] to extract their geometric information (X–Y coordinates). Then,

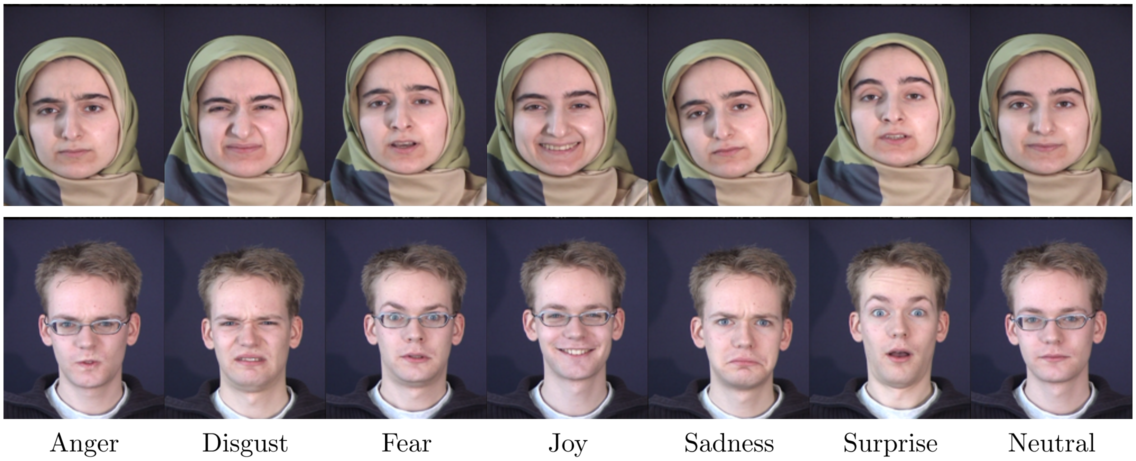


Figure 3.3: Sample images in the MMI database

the image registration, LBPH and HOG feature extraction and data dimensionality reduction were conducted the same as on the CK+ database.

3.3 The GEMEP-FERA database

The GEMEP-FERA database is provided by the 2011 Facial Expression Recognition and Analysis Challenge (FERA2011). The emotion challenge of the GEMEP-FERA database [17, 70] contains 289 portrayals of five emotions: anger, fear, joy, sadness, and relief. Figure 3.4 shows the sample images of GEMEP-FERA with five expressions.

Of all the portrayals, 155 sessions are for training including seven subjects (three men) with three to five instances of each emotion for each subject. The remaining 134 sessions are provided for testing including six subjects (three men), where half of the subjects are not present in the training set. Each actor contributed three to ten instances per emotion in the test set. For this database, the facial feature extraction process were kept the same as on the MMI database including the face landmark

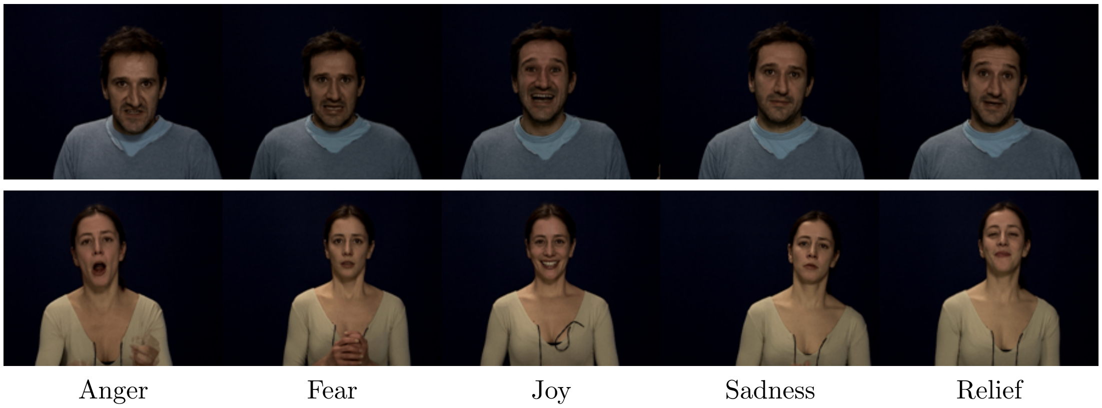


Figure 3.4: Sample images in the GEMEP-FERA database

tracking, image registration, and HOG and LBPH feature extraction from all video frames.

3.4 The DISFA database

The DISFA database contains videos with facial activities from 27 adult subjects. Each subject was video-recorded using a high resolution camera (1024 × 768) at 20 frames per second (fps) while the subject was viewing a 4-minute stimulus video clip. The video clip was designed with the intent of eliciting subjects' spontaneous facial expressions. For each video frame, the intensity of 12 AUs was manually annotated at a six-point ordinal scale (i.e, from 0 to 5). This database also provides 66 facial landmark points for all video frames of each subject. Figure 3.5 shows the sample images of the DISFA database.

In our experiments, to be consistent with the study on DISFA reported in [76], for each AU the video frames with intensity 0–1 were labeled as the absence of that AU while the frames with intensity 2–5 were labeled as the presence of that AU. Afterwards, the image registration, LBPH and HOG feature extraction and data

Figure 3.5: Sample images in the DISFA database

dimensionality reduction were conducted by following the same settings as on the CK+ database.

Chapter 4

l_p-norm MKL-based multiclass-SVM for basic facial expression recognition

This chapter presents the formulation of our proposed transfer learning method to fuse different types of facial features with multiple kernel functions for facial expression recognition. We first introduce the optimization problem of the MKL with sparse kernel combinations (l_1-norm MKL) and the l_p-norm MKL algorithm for binary classification problems, and then formulate our multiclass extension, l_p-norm MKL-based multiclass-SVM, via one-against-one and one-against-all techniques [77]. Experimental works on CK+, MMI and GEMEP-FERA databases are shown and discussed based the comparison with several state-of-the-art methods.

4.1 l_1-norm MKL-based binary SVM

Usually, the parameters of kernel functions in canonical SVM classifiers are tuned during the training-validation experiments, and the parameters that result in the best classification rate on validation samples are applied to recognize the test samples. However, it is known that different kernels with different parameters correspond to different representation of features. Instead of trying to find which works the best, MKL-based SVM use a combination of them and define automatic learning methods to pick the optimal parameters.

In this section, we present the formulation of the MKL optimization problem with sparse constraints and review some algorithms for solving it. Given a set of N training samples $\{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the i^{th} feature vector of the training set $\mathcal{X}$ with dimension D, and $y_i \in \{-1, +1\}$ is its corresponding class label, the MKL optimization problem in [52] is formulated either by Equation 4.1.1 or in its equivalent form by Equation 4.1.2 as proposed in [16]:

$$\begin{aligned} \min_{w,w_0,\xi} \quad & J(w, w_0, \xi) = \frac{1}{2}(\sum_{m=1}^{M} \|w_m\|_2)^2 + C\sum_{i=1}^{N} \xi_i \\ s.t. \quad & y_i(\sum_{m=1}^{M} w_m^T \phi_m(x_i) + w_0) \geq 1 - \xi_i, \;\; i = 1, 2, ..., N \\ & \xi_i \geq 0, \;\; i = 1, 2, ..., N \end{aligned} \tag{4.1.1}$$

Here $\xi = (\xi_1, \xi_2, ..., \xi_N)^T$ is known as the vector of slack variables in canonical SVM for nonseparable classification problems, and C is a positive constant preset to control the relative influence of nonseparable samples. $\phi_m(\cdot)$ is a map that maps the feature domain $\mathcal{X}$ into the m^{th} reproducing kernel Hilbert space (RKHS) $\mathcal{H}_m$, based on which kernel function $k_m(\cdot, \cdot)$ is defined as $k_m(\cdot, \cdot) =< \phi_m(\cdot), \phi_m(\cdot) >$. M is the number of kernels in use. w_m is the direction of hyperplane in $\mathcal{H}_m$, w denotes the set $\{w_m\}$ and w_0 is the exact in-space position of hyperplanes.

$$
\begin{aligned}
\min_{w,w_0,\xi,\theta} \quad & J(w, w_0, \xi, \theta) = \frac{1}{2}\sum_{m=1}^{M} \frac{1}{\theta_m}\|w_m\|_2^2 + C\sum_{i=1}^{N} \xi_i \\
s.t. \quad & y_i(\sum_{m=1}^{M} w_m^T \phi_m(x_i) + w_0) \geq 1 - \xi_i, \;\; i = 1, 2, ..., N \\
& \xi_i \geq 0, \;\; i = 1, 2, ..., N \\
& \sum_{m=1}^{M} \theta_m = 1, \quad \theta_m \geq 0, \;\; m = 1, 2, ..., M
\end{aligned}
\tag{4.1.2}
$$

$\theta = (\theta_1, \theta_2, ..., \theta_M)^T$ is the kernel combination vector that controls the weight of the squared form of w_m in the objective function. When $\theta_m = 0$, $\|w_m\|_2$ should also be equal to zero to yield a finite objective value.

Several MKL algorithms are proposed in [52, 16, 78] to solve the above two equivalent optimization problems. In [52], Equation 4.1.1 was transformed to be a semi-infinite linear program (SILP), and a chunking algorithm was proposed to solve the SILP by simultaneous optimization of SVM and kernel combination weights. This algorithm can be applied to large scale learning tasks. The SimpleMKL and HessianMKL algorithms are two other optimization techniques proposed to solve Equation 4.1.2. These techniques utilize two nested loops that iteratively learn the decision hyperplanes in C-SVM and the kernel combination vector. In the inner iteration, both algorithms solve the canonical binary SVM by fixing the kernel combination vector. In the outer iteration, the SimpleMKL utilizes a reduced gradient descent algorithm [79] with a 1D search algorithm – the golden section search method [80] – to update the combination weights, whereas the HessianMKL expands the weight updating problem to be a standard quadratic programming problem.

All these methods solve the same convex optimization problem and give the same optimum, though. The HessianMKL algorithm turns out to be the most efficient one as justified in [78, 16]. However, as presented in [81], the objective function in Equation 4.1.1 contains a $l_{2,1}$-norm penalizer $\frac{1}{2}(\sum_{m=1}^{M}\|w_m\|_2)^2$, which calculates the l_1-norm of the squared hyperplane directions over multiple kernel spaces and will promote a sparse solution for the usage of kernel functions. Similarly, the l_1-norm constraint on the vector θ in Equation 4.1.2 is a sparsity constraint that encourages sparse basis kernel combinations. Thereafter, very limited number of basis kernel functions ($k_m(\cdot,\cdot)$, $m = 1, 2, ..., M$) are used to represent the test samples during classification tasks, which may reduce the discriminative power of MKL-based SVM.

4.2 l_p-norm MKL-based binary SVM

To allow for non-sparse kernel mixtures, the authors of [55, 56] extended MKL to arbitrary norms, that is l_p-norm MKL with $p \geq 1$. In this part, we present the formulation and the solution of this generalized MKL optimization problem for binary classification tasks.

The l_p-norm MKL is named from the novel regularizer of SVM as follows:

$$\Omega(w) = \frac{1}{2}\|w\|_{2,p}^2, \quad p \geq 1,$$

where the $l_{2,p}$-norm is defined as $\|w\|_{2,p} = (\sum_{m=1}^{M}\|w_m\|_2^p)^{\frac{1}{p}}$, and w_m is the direction of discriminative hyperplane to be learned in each RKHS. Together with the hinge loss function of SVM, the primal form of l_p-norm MKL-based binary SVM is obtained as:

$$\begin{aligned}
&\min_{w,\xi} \quad J(w,\xi) = \frac{1}{2}\|w\|_{2,p}^2 + C\sum_{i=1}^{N}\xi_i,\ \ p \geq 1 \\
&s.t. \quad y_i\left(\sum_{m=1}^{M} w_m^T\phi_m(x_i)\right) \geq 1-\xi_i,\ \ i=1,2,...,N \\
&\qquad\ \ \xi_i \geq 0,\ \ i=1,2,...,N
\end{aligned} \tag{4.2.1}$$

which is a convex optimization problem as proved in [55, 56].

Note that when $p = 1$ the formulation is the same as the one defined in Equation 4.1.1, which enforces sparse kernel combinations. Equation 4.2.1 is solved based on its dual form shown as follows:

$$\begin{aligned}
&\min_{\theta}\max_{\alpha} \ L(\theta,\alpha) = \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T Y K_\theta Y \alpha,\ p \in [1,2) \\
&\max_{\theta}\max_{\alpha} \ L(\theta,\alpha) = \mathbf{1}^T\alpha - \frac{1}{2}\alpha^T Y K_\theta Y \alpha,\ p \in (2,+\infty) \\
&\qquad s.t. \ \ K_\theta = \sum_{m=1}^{M}\theta_m K^{(m)}, \\
&\qquad\qquad \sum_{i=1}^{N}\alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \\
&\qquad\qquad \sum_{m=1}^{M}\theta_m^{p/(2-p)} \leq 1, \quad \theta_m \geq 0
\end{aligned} \tag{4.2.2}$$

where $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_N)^T$ is the vector of Lagrangian dual variables corresponding to each training sample, and $Y = diag(y_1, y_2, \cdots, y_N)$ is an $N \times N$ diagonal matrix. $K^{(m)}$ is the kernel matrix corresponding to the m^{th} kernel function, and $K_{i,j}^{(m)} = k_m(x_i, x_j)$.

In [56], the authors proposed a simple macro-wrapper algorithm for solving Equation 4.2.2, and proved its convergence in the case of $p > 1$. The macro-wrapper solver contains two nested steps for parameter updating. In the outer one, the kernel combination weights are updated by fixing the variables of SVM. Whereas in the inner iteration, with fixed kernel combination weight vector the optimization problem is transformed to the canonical C-SVM problem, and can be solved by any SVM solver. Detailed steps are shown in Algorithm 1.

Algorithm 1 The simple macro-wrapper algorithm

```
Require: $p \in (1, +\infty)\backslash\{2\}$, $C$ and $Y$
  for all $m \in \{1, 2, \cdots, M\}$ do
    initialize $\theta_m := (1/M)^{(2-p)/p}$
    compute $K^{(m)}$
  end for
  while optimality conditions are not satisfied do
    compute $K_\theta$ based on the constraint in Equation 4.2.2
    update $\alpha$ based on a canonical SVM solver
    for all $m \in \{1, 2, \cdots, M\}$ do
      if $p < 2$ then
        update $\theta_m := \frac{\theta_m(\alpha^T Y K^{(m)} Y \alpha)^{\frac{2-p}{2}}}{\left[\sum_{m'=1}^{M} \theta_m(\alpha^T Y K^{(m')} Y \alpha)^{\frac{p}{2}}\right]^{\frac{2-p}{p}}}$
      else
        update $\theta_m := \frac{(\alpha^T Y K^{(m)} Y \alpha)^{\frac{2-p}{2p-2}}}{\left[\sum_{m'=1}^{M} (\alpha^T Y K^{(m')} Y \alpha)^{\frac{p}{2p-2}}\right]^{\frac{2-p}{p}}}$
      end if
    end for
  end while
Ensure: $\alpha$ and $\theta$
```

We implement Algorithm 1 for the case $p > 1$. The optimality conditions are set based on number of total iterations and the variations of updated θ between consecutive iterations. For the case of $p = 1$, we apply the HessianMKL algorithm due to its higher computational efficiency.

In the test phase, given a test sample $x_0 \in \mathbb{R}^D$, the label of x_0 (denoted by y_0) can be calculated as follows.

$$y_0 = \text{sgn}[\sum_{m=1}^{M} \theta_m \underbrace{\left(\sum_{i=1}^{N} \alpha_i y_i k_m(x_i, x_0)\right)}_{\text{single kernel with single feature}}] \tag{4.2.3}$$

The formulation within the under bracket is the discriminant function used for classifying new samples in canonical binary SVM. In other words, by using MKL-based SVM the label of a sample is determined based on weighted summation of the results obtained from each RKHS, which enhances the discriminant power for classification. In Appendix A, we justify the superiority of MKL-based SVM to the canonical single kernel SVM by showing that the minimized the objective function in Equation 4.2.1 preserves the lower boundary of the one in canonical SVM.

4.3 l_p-norm MKL-based multiclass-SVM

In this section, we present our proposed MKL-based multiclass-SVM framework by extending the binary l_p-norm MKL classifier described by Equation 4.2.2 for multiclass classification. Suppose we want to classify U classes using binary classifiers. Two techniques are commonly used in the literature: one-against-one and one-against-rest. In the one-against-one technique, $U(U-1)/2$ binary classifiers are built for all pairs of distinct classes, whereas in the one-against-rest technique U binary classifiers are built for each class of data.

The authors of [16] presented a structure of MKL-based multiclass-SVM using the SimpleMKL algorithm. In their structure, a single kernel combination weight vector is jointly learned for all binary classifiers in the multiclass-SVM as:

$$\min_{\theta} \quad L(\theta) = \sum_{u \in \Phi} L_u(\theta) \tag{4.3.1}$$

where Φ is the set of all pairs of distinct classes considered in the multiclass-SVM, and $L_u(\theta)$ denotes the objective function for optimizing kernel combination vector θ with fixed SVM parameters in Equation 4.2.2 ($p = 1$). By this definition, the inner loop of the SimpleMKL-based multiclass-SVM is to solve the multiclass-SVM while in the outer loop a single kernel weight vector is learned to minimize the summation of the objective functions from all binary classifiers. Therefore, the learned optimal kernel weight vector can be used for all binary classifiers, which generally increases the recognition result of multiclass-SVM.

However, only one kernel weight vector may not be good enough to reflect the contribution of each binary classifier in the whole objective function. Since features are projected into different spaces using different kernel functions, for a binary classifier the values of kernel combination weights reflect the choice of optimal kernel functions and features used for distinguishing between two classes. Therefore, in the MKL-based multiclass-SVM it is most likely that different binary classifiers may have different optimal kernel weight vectors for classification. However, the structure of the SimpleMKL-based multiclass-SVM, which uses the same kernel weight vector for all binary classifiers, does not have a good resolution of selecting kernel combinations for different pairwise classifiers.

Based on the discussion above, we are looking for one combination weight vector θ_u to optimize each pairwise SVM's objective function L_u as in Equation 4.3.2, and the proof of the superiority of our method over SimpleMKL-based multiclass-SVM is shown in B.

$$\begin{aligned} \min_{\theta_u} \quad & \hat{L} = \sum_{u \in \Phi} L_u(\theta_u), \text{ if } p \in [1, 2) \\ \max_{\theta_u} \quad & \hat{L} = \sum_{u \in \Phi} L_u(\theta_u), \text{ if } p \in (2, +\infty) \end{aligned} \tag{4.3.2}$$

We implemented the structure of our MKL-based multiclass-SVM proposed in Equation 4.3.2 using the l_p-norm MKL algorithm for the $p > 1$ case and the Hessian-MKL algorithm for the $p = 1$ case. Similar to the SimpleMKL-based multiclass-SVM, in the inner loop the common multiclass-SVM is solved by a C-SVM solver – SVM-KM [82] with either the one-against-one rule or the one-against-rest rule while in the outer loop the two MKL algorithms for different cases are implemented to learn one kernel weight vector for each binary classifier. In our application for facial expression recognition, the one-against-rest rule is used and the classification of novel samples is done by a max-wins voting strategy. The pseudo code of our framework is shown in Algorithm 2.

Algorithm 2 The l_p-norm MKL-Based Multiclass-SVM

```
Ensure: $p(\geq 1)$, $K^{(m)}(m = 1, \cdots, M)$, $y_u$ and feasible $\theta_u(u \in \Phi)$
Require:
  if $p = 1$ then
    for all $u \in \Phi$ do
      run the HessianMKL for $\theta_u^\star$ and $\alpha_u^\star$
    end for
  else
    for all $u \in \Phi$ do
      run the Algorithm 1 for $\theta_u^\star$ and $\alpha_u^\star$
    end for
  end if
```

4.4 Facial expression recognition experiments

This section illustrates the settings of the l_p-norm MKL-based multiclass-SVM for facial expression recognition. The experimental results on the CK+ and MMI databases are shown and discussed to evaluate the performance of our proposed method.

4.4.1 Classifier Settings

We used the following configuration for fusing LBPH and HOG features at the kernel level with different kernel function parameters based on our proposed MKL framework. In the experiments, we used HtRBF and polynomial functions as defined in Equation 4.4.1.

$$\begin{aligned} &k_{HtRBF}(x,y) = e^{-\rho \sum_i |x_i^a - y_i^a|^b}, \rho > 0, 0 \leq a \leq 1, 0 \leq b \leq 2 \\ &k_{poly}(x,y) = \langle x,y \rangle^r, r \in \mathbb{N} \end{aligned} \tag{4.4.1}$$

where a, b and ρ are the kernel parameters of the HtRBF kernel, x_i is the i^{th} element of feature vector x, and r is the order of the polynomial function. The HtRBF was first defined in [83], where the commonly used Gaussian function or RBF [84] is a special case when $a = 1, b = 2$. As stated in [83], it achieves better classification results than polynomial function and Gaussian function for image classification within SVM classifiers.

In our experiments, we set different values for parameters of the above two kernel functions with the criterion that they fill a proper range of the defined domain. For the HtRBF, we set $a \in \{0.1, 0.3, 0.7, 1\}$, $b \in \{0.1, 0.5, 1, 1.5, 2\}$ and $\rho \in \{0.01, 0.1, 0.5, 1, 10, 50, 100\}$; for the polynomial function, we set $r \in \{1, 2, 3\}$. Thereafter, we obtained 143 parameterized kernels from the two defined kernel functions (i.e., $4 \times 5 \times 7 + 3 = 143$). Hence given any pair of samples (e.g., the i^{th} and j^{th} registered images), the fusion of extracted LBPH ($\{x_i, x_j\}$) and HOG features ($\{z_i, z_j\}$) at the kernel level within our framework is handled as follows:

$$K_{i,j} = \sum_{m=1}^{143} [\theta_m k_m(x_i, x_j) + \theta_{m+143} k_m(z_i, z_j)] \tag{4.4.2}$$

where $k_m(\cdot,\cdot)$ is one of the 143 parameterized kernels, and $\theta = (\theta_1, \theta_2, ..., \theta_{286})^T$ ($\|\theta\|_{2/(2-p)} = 1, p \geq 1$) is the kernel combination vector to be optimized during MKL.

4.4.2 Experimental results on CK+

We designed eight independent SVM classifiers in our experiments based on the CK+ database. A standard 10-fold cross-validation scheme was adopted to find the best values of the parameters for the classifiers while conducting person-independent experiments. We randomly separated subjects into 10 folds including training, validation and test sets. In each round of our cross-validation, one fold was left out as test set. Among the rest 9 folds, we use one as validation set and repeat 9 times to find the best classifier parameters for the test set. Hence, the samples in the test sets were never used in training or validating the algorithm. By comparing the experimental results of these classifiers, we empirically studied the advantage of our framework for the application of facial expression recognition. The detailed information of the designed classifiers is listed in Table 4.1.

Table 4.1: Information of designed eight SVM classifiers

#	Classifier	Feature	# of Kernels	Kernel Parameters
C1	Canonical SVM	LBPH	1	C, a, b, ρ, r
C2	Canonical SVM	HOG	1	C, a, b, ρ, r
C3	SimpleMKL-based multiclass-SVM	LBPH HOG	143	C
C4	l_p-norm MKL-based multiclass-SVM($p = 1$)	LBPH HOG	143	C
C5	l_p-norm MKL-based multiclass-SVM($p > 1$)	LBPH HOG	143	C, p
C6	l_p-norm MKL-based multiclass-SVM($p \geq 1$)	LBPH HOG	1	C, p, a, b, ρ, r
C7	Canonical SVM with averaging kernels	LBPH HOG	143	C
C8	Canonical SVM with product kernels	LBPH HOG	143	C

In classifiers C1, C2 and C6, we tuned single kernel from the 143 parameterized kernels among the selected two kernel functions listed in Section 4.4.1 as well as the parameter C ($C \in \{0.01, 0.1, 1, 10, 100, 500, 800, 1000\}$) in C-SVM during the training and validation steps. However, by fusing the HtRBF and polynomial kernels in Equation 4.4.2, both the SimpleMKL-based multiclass-SVM (C3) and our proposed l_p-norm MKL-based multiclass-SVM (C4, C5) can automatically learn the optimal kernel combination weights, leaving only C or p to be tuned during cross-validation. In our work, we set $p \in \{1, 1.05, 1.2, 1.35, 1.5, 1.65, 1.8, 1.95, 2.1, 4, 8, 16\}$. In order to further show the benefit of using MKL-based SVM, we evaluate two other kernel combination methods within C-SVM, averaging kernels (C7) and product kernels (C8), as other baseline classifiers. The kernel combination matrices in these two methods can be precalculated without learning, then their optimization problems for learning the discriminant hyperplanes are equivalent to only solving the canonical multiclass-SVM. Finally, for all the eight classifiers, the parameters that correspond to the highest overall recognition rates on the validation data were applied to predict the facial expressions of the test samples.

Figure 4.1 lists the overall recognition rate of the designed classifiers. Figure 4.2 shows the performance of the eight classifiers on each of the 7 expression classes (six basic facial expressions and the neutral faces), and Table 4.2 shows the confusion matrix of classifier C5, where we achieved the highest overall recognition rate using our MKL framework for the case $p > 1$.

In the following, we focus on comparing the performance of different SVM classifiers and empirically study the effect of MKL-based SVM classifiers on facial expression recognition. Our results using eight designed classifiers are compared from five aspects as follows.

a) Canonical SVM with single kernel vs. MKL-based SVM (C1, C2 vs. C3, C4, C5): Comparing the results shown in Figure 4.1, we can see that both the SimpleMKL-based multiclass-SVM (C3) and our proposed l_p-norm based multiclass-

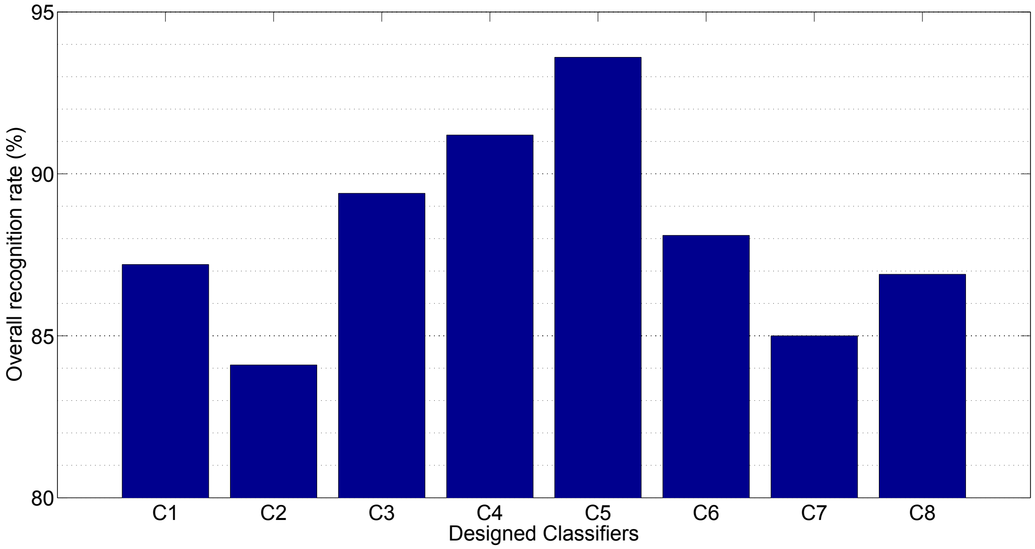


Figure 4.1: Overall recognition rate of designed classifiers

Table 4.2: Confusion matrix of classifier C_5 using l_p-norm MKL-based multiclass-SVM with multiple kernels and features on CK+ ($p > 1$, overall recognition rate: 93.6%)

%	Ag	Sp	Dg	Fr	Jy	Sd	Nt
Anger	**97.8**	0	1.5	0	0	0.7	0
Surprise	2.0	**91.2**	1.6	1.2	0.8	2.8	0.4
Disgust	1.7	0	**96.0**	0.6	0	0.6	1.1
Fear	0	0	2.7	**93.3**	0	1.3	2.7
Joy	0.5	1.0	1.9	1.4	**92.3**	0.5	2.4
Sadness	0	0	1.2	1.2	0	**96.4**	1.2
Neutral	0.3	1.3	0.6	1.0	2.3	1.9	**92.6**

SVM (C4, C5) can generally boost the accuracy of facial expression recognition by fusing different features with multiple kernels. Specifically, in Figure 4.2, the recognition rate of Anger has been increased from 82.2% (C1) and 71.9% (C2) to 88.2% (C3), 91.1% (C4) and 97.8% (C5), respectively. Moreover, both C3 and C4 increased the recognition rates of Surprise, Fear, and Sadness expressions from C1 and C2, and their recognition rates of Disgust, Joy, and neutral faces are comparable to the first two classifiers. Compared to C1 and C2, classifier C5 achieved higher recognition results of all classes except the classes of Surprise and Joy, which are kept comparable.

b) SimpleMKL vs. l_p-norm MKL ($p = 1$) in multiclass-SVM (C3 vs. C4): The similarity of these two MKL methods lies in the fact that they share the same objective function and force the optimized kernel combination vectors to be sparse with l_1-norm. The only difference between them is that for multiclass classification tasks the SimpleMKL-based multiclass-SVM keeps the kernel weight vectors for all binary classifiers to be the same while our proposed multiclassi-SVM structure learns one kernel weight vector for each binary classifier. The advantage of our method is that it gives the system more flexibility in selecting optimal kernel weights for each binary classifier. Comparing the experimental results of these two methods shown in Figures 4.1 and 4.2, we can see that the overall recognition rate was increased from 89.4% (C3) to 91.3% (C4) using our proposed MKL-based multiclass-SVM framework. Especially, the recognition results of all classes are boosted by 1.5% $\sim$ 3.9% expect the Fear and Sadness expressions, which are kept comparable.

c) Sparse case vs. Non-sparse case in l_p-norm MKL (C4 vs. C5): This comparison is between the $p = 1$ case and the $p > 1$ case within our proposed framework. In the case of $p = 1$, the MKL algorithm in our framework forces the kernel weight vectors to be sparse for all binary classifiers. Thereafter, only the activated parameterized kernels (corresponding to non-zero kernel combination weights) and features

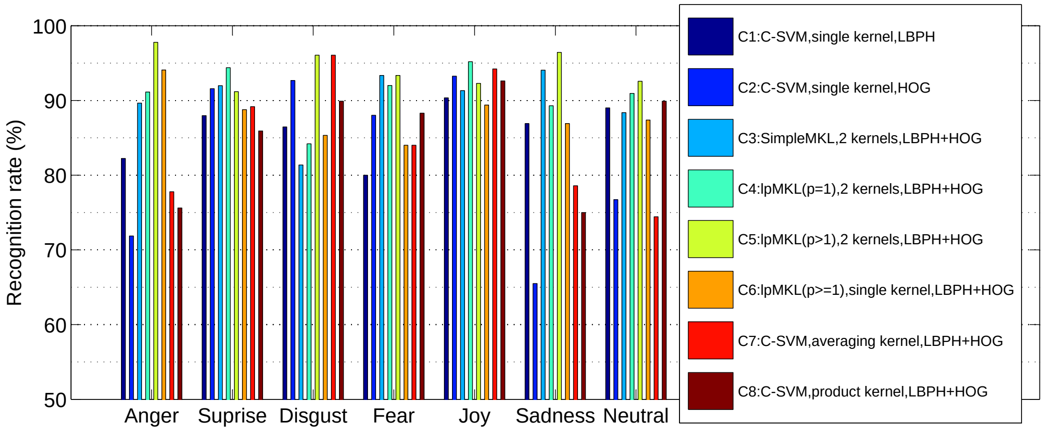


Figure 4.2: The performance of designed classifiers on each of the 7 facial expressions

are used in the testing step. However, for the non-sparse case ($p > 1$), usually all fused features and parameterized kernels are activated. Table 4.3 reports the average number of activated kernels associated with feature types and kernel functions after training-validation steps in our experiments. As we can see that, in the sparse case, for each type of feature, at most 3 out of 140 parameterized HtRBF kernels are activated whereas in the non-sparse case, almost all are utilized. Reviewing the experimental results in Figure 4.1 and Figure 4.2 show that the non-sparse classifier C5 in our framework generally achieved higher recognition rates than the sparse one C4 (increased by 2.5%). In addition, we also backtracked the values of the tuned parameter p in each round of the 10-fold cross-validation for testing samples, and found that $p \in [1.05, 1.65]$. Therefore, we conclude that the non-sparse MKL method is more suitable for facial expression recognition application.

Table 4.3: Number of activated parameterized kernels associated with features and kernel functions in l_p-norm MKL

(a) Sparse case (C4)

#	LBPH	HOG
HtRBF	3/140	2/140
poly	1/3	1/3

(b) Nonsparse case (C5)

#	LBPH	HOG
HtRBF	133/140	137/140
poly	3/3	3/3

d) Single kernel vs. Multiple kernels for fusing multiple features in MKL-based SVM (C6 vs. C4, C5): In our experiments, classifiers C4 and C5 used multiple kernels to fuse LBPH and HOG features, whereas C6 only applied single kernel function for both features. The overall recognition result of C6 was improved from 88.2% to 91.3% (C4) and to 93.6% (C5). Furthermore, C5 achieved higher recognition results than C6 for all 7 classes.

e) l_p-norm based kernel combination vs. Other kernel combination methods (C4, C5 vs. C7, C8): In order to justify the benefit of using l_p-norm MKL, we provided two extra kernel combination strategies proposed in [85] as baseline methods. One is the averaging kernels, a linear kernel combination method, which forces all kernel combination weight to be equal with their summation to be 1. The other is the product kernel, where kernel functions are non-linearly combined based on dot product. Viewing the experimental results from Figure 4.1 and Figure 4.2, we conclude that the l_p-norm MKL based kernel combination generally outperform the kernel combination methods with averaging kernels and product kernels for facial expression recognition.

We further recognized the six basic facial expressions excluding neutral faces (i.e., 6-class recognition) using our proposed MKL framework. The detailed results are shown in Table 4.4.

Table 4.4: Confusion matrix using l_p-norm MKL-based multiclass-SVM for six basic expressions on CK+ (overall recognition rate: 95.5%)

%	Ag	Sp	Dg	Fr	Jy	Sd
Anger	**95.6**	1.5	0.7	0.7	0	1.5
Surprise	1.2	**94.4**	0.4	2.0	0.4	1.6
Disgust	1.1	0.6	**95.5**	1.7	0	1.1
Fear	1.3	4.0	0	**92.0**	0	2.7
Joy	0	1.4	0.5	0	**97.6**	0.5
Sadness	1.2	0	1.2	1.2	0	**96.4**

Compared with several state-of-the-art facial expression recognition methods listed in Table 4.5, we can see that our method achieves the best recognition rate for seven expressions and the second best accuracy for six basic expressions. On a personal computer with Intel i5 CPU (2.66 GHz) and 8 GB memory, the average computation time of our proposed l_p-norm MKL multiclass-SVM with Matlab implementation is 13.5 minutes for classifier training and parameter tuning in each round of the 10-fold cross-validation scheme. It takes about 0.03 seconds in the recognition step given extracted LBPH and HOG features of each test sample.

Table 4.5: Recognition rate (%) of state-of-the-art methods on CK/CK+

References	Database	# of classes	%
[86]	CK+	7	92.7
[87]	CK+	7	85.8
	CK+	6	95.8
[88]	CK+	7	89.3
[18]	CK	7	91.4
	CK	6	95.1
[89]	CK	7	91.5
Our l_p-norm MKL framework	CK+	7	93.6
	CK+	6	95.5

4.4.3 Experimental results on MMI

Similar to the works on CK+, 10-fold cross-validation was applied for pruning C and p on MMI's 30 subjects. Tables 4.6 and 4.7 show the confusion matrices for 6-class and 7-class recognition by l_p-norm MKL-based multiclass-SVM.

It can be observed from Table 4.6 and Table 4.7 that besides the CK+ database, our proposed method can achieve promising recognition accuracies for each facial expression on the MMI database (i.e., all above or close to 90%), which implies its effectiveness. We also backtracked the value of p tuned in each round of the cross-

Table 4.6: Confusion matrix using l_p-norm MKL based multiclass-SVM with multiple kernels and features for 6 expressions on the MMI database (overall recognition rate: 93.6%)

%	Ag	Sp	Dg	Fr	Jy	Sd
Anger	**94.9**	0	2.0	0	0	3.0
Surprise	1.6	**95.1**	0	0.8	2.4	0
Disgust	0	2.1	**90.6**	2.1	0	5.2
Fear	0	4.6	1.1	**92.0**	2.3	0
Joy	0.8	0.8	2.4	0.8	**94.4**	0
Sadness	4.2	0	0	2.1	0	**93.8**

Table 4.7: Confusion matrix using l_p-norm MKL-based multiclass-SVM with multiple kernels and features for 7 expressions on the MMI database (overall recognition rate: 92.8%)

%	Ag	Sp	Dg	Fr	Jy	Sd	Nt
Anger	**93.0**	2.0	1.0	1.0	1.0	2.0	0
Surprise	1.6	**92.6**	2.4	0.8	0.8	0.8	0.8
Disgust	0	3.1	**91.7**	0	1.0	2.1	2.1
Fear	0	2.3	0	**95.4**	0	2.3	0
Joy	0.8	0.8	3.1	0.8	**89.7**	2.4	2.4
Sadness	2.1	2.1	0	3.1	0	**90.6**	2.1
Neutral	0.5	0	1.4	1.9	0	1.0	**95.2**

validation process, and the tuned best p for test samples are within the range $[1.2, 1.8]$. Therefore, similar to the results on the CK+ database, the non-sparse kernel weight vectors in our framework outperformed the sparse ones.

Table 4.8 compares our proposed l_p-norm MKL framework with several state-of-the-art methods on the MMI database. As listed below, our method achieved favorable experimental results. Especially we obtained a significant improvement in the recognition of seven facial expression compared to [18]. The results of six basic expressions are quite comparable with the best one among state-of-the-art methods. These confirm the effectiveness of our method. Nevertheless, as the techniques used for image registration, facial feature representation and experimental setup such as image sequence selection across these methods are not exactly the same, it is hard to hold a completely fair comparison with the CK+ and the MMI databases. Thus, this comparison could only be regarded as a reference to demonstrate that fusing feature with non-sparse MKL will help enhance the classification performance.

Table 4.8: Recognition rate (%) of state-of-the-art methods on the MMI database

References	# of sequences	# of classes	%
[18]	99	7	86.9
[88]	238	6	95.8
[90]	175	6	94.1
[91]	96	6	82.7
Our l_p-norm MKL framework	209	7	92.8
	209	6	93.6

We further performed cross-database evaluation of our proposed method. To be specific, we trained our l_p-norm MKL-based multiclass-SVM on one database and then tested the classifier on the other one. During the classifier training phase, samples from each facial expression were randomly selected across subjects, and the number of training samples for each class (expression) was kept the same to conduct uniformly weighted classifiers. Table 4.9 shows detailed information of classifier settings and the recognition results. The values of C and p for training classifiers were set based on their best tuned values obtained in the within-database experiments.

Table 4.9: Cross-database evaluation performance of our proposed l_p-norm MKL-based multiclass-SVM

Experiment Settings	C	p	Overall	[18]
Train: CK+; Test: MMI	500	1.35	66.9%	51.1%
Train: MMI; Test: CK+	800	1.65	61.2%	–

Compared to the result reported in [18] (51.1%), our method achieved better recognition performance (66.9%) when we trained on the CK+ database and tested on the MMI database. We further observed that the overall recognition results of cross-database experiments was much lower than those of within-database experiments. As the image registration, feature extraction and dimension reduction on two databases were conducted in the same way, one reason of such disparity may be due to different controlled environments during database collection. The paper [92] suggested that in order to obtain good cross-database evaluation environments, large training databases should be collected to cover variations of image and subject conditions. This statement can be reinforced by our experimental results. That is, when training classifiers on CK+ with more subjects and testing on MMI with less subjects, we achieved better results than training on MMI and testing on CK+.

4.4.4 Experimental results on GEMEP-FERA

The objective of the emotion recognition challenge in GEMEP-FERA is to classify each of the entire video session into one of the five emotion classes including anger, fear, joy, sadness, and relief. We apply our proposed l_p-norm MKL multiclass-SVM framework to the GEMEP-FERA emotion database. 7-fold cross-validation (one fold per subject in the training set) was adopted in the training phase for finding the best values of parameters C and p in our framework. Each of the seven subjects in the training set was associated with one fold. In our test phase, every frame in each test session was classified, and similar to [43] the emotion that was labeled in the largest number of the frames in one session was assigned to the class of that session.

The confusion matrices of our experimental results are shown in Tables 4.10, 4.11 and 4.12 with person-independent, person-specific and overall partitions respectively. Unlike CK+ and MMI databases that lack common protocols for experimental settings, the splitting of the training and test sets provided in this GEMEP-FERA database gives a benchmark setup for users to hold a fair comparison with others' works in the literature. In Table 4.13, we compare our results with several state-of-the-art methods. Especially, the UCRiverside, UIUC and KIT are the best three groups among all the participants in the competition of emotion recognition challenge as reported in [70]. We can notice that our method obtained the best result on the person-independent experiment (1.1% better than UCRiverside) and the second best result on the entire test set (0.2% lower than UCRiverside); the performance of the person-specific partition is kept comparable with the state-of-the-art methods. These confirm the effectiveness of our proposed framework.

Table 4.10: Confusion matrix for person-independent emotion recognition on GEMEP-FERA (overall recognition rate: 76.3%)

%	Anger	Fear	Joy	Relief	Sadness
Anger	**85.7**	0	0	0	14.3
Fear	13.3	**66.7**	6.7	13.3	0
Joy	0	5.0	**85.0**	10.0	0
Relief	12.5	6.3	0	**75.0**	6.3
Sadness	6.7	6.7	0	20.0	**66.7**

GEMEP-FERA is a very challenging database. Different from the CK+ and the MMI databases, its video sessions are neither initialized from neutral faces nor ended with an apex emotive state, and the subjects' expressions are less posed and carica-

Table 4.11: Confusion matrix for person-specific emotion recognition on GEMEP-FERA (overall recognition rate: 94.4%)

%	Anger	Fear	Joy	Relief	Sadness
Anger	**92.3**	0	0	0	7.7
Fear	0	**90.0**	10.0	0	0
Joy	0	0	**100.0**	0	0
Relief	0	0	0	**100.0**	0
Sadness	0	0	0	10.0	**90.0**

Table 4.12: Confusion matrix for all test sessions on GEMEP-FERA (overall recognition rate: 83.6%)

%	Anger	Fear	Joy	Relief	Sadness
Anger	**88.9**	0	0	0	11.1
Fear	8.0	**76.0**	8.0	8.0	0
Joy	0	3.2	**90.3**	6.5	0
Relief	7.7	3.9	0	**84.6**	3.9
Sadness	4.0	4.0	0	16.0	**76.0**

Table 4.13: Emotion recognition results (%) of the state-of-the-art methods on GEMEP-FERA

References	Person Independent	Person Specific	Entire Test Set
KIT	65.8	94.4	77.3
UIUC [93]	65.5	**100.0**	79.8
[43]	73.9	98.0	83.5
UCRiverside [94]	75.2	96.2	**83.8**
Our work	**76.3**	94.4	83.6

tured. Moreover, this database also includes speech activities, which usually cause strong variability in the appearances of lower face expressions and some significant non-frontal head poses of subjects. All these challenges can be viewed as the complexity of facial expression recognition in dynamic schemes. In this case, dynamic relations among sequential frames are worth considering since the task is to label an entire video session.

In our future work, we will exploit our framework to fuse the facial features with rich dynamic information. The LGBP-TOP [95] and the LBP-TOP [96] features proposed in [95] can be good examples in this case, which designed temporal extensions of classical LBPH feature across consecutive video frames. This feature can represent the dynamic appearance information between consecutive video frames. In addition, proper image registration can also help improve the performance. This idea can be strengthened by the work of the Riverside group [94], which achieved the best result on the entire test set of the GEMEP-FERA database. The authors proposed a facial image registration framework to perform a global alignment of the faces and meanwhile preserve the facial dynamic motions across each expression event. Similar work can be found in [90], where the authors modeled facial feature changes during expression events by a diffeomorphic image registration framework. Moreover, post-processing techniques can also be applied to infer the final emotive labels of videos based on the output of our l_p-norm MKL multiclass-SVM. For instance, in [43] an average filter was used on the SVM outputs to exploit the temporal component of facial image sequences, and the authors achieved 1.7–3.5 % improvement of their facial expression recognition performance.

Chapter 5

Group-sensitive MTL for facial action unit detection

In this chapter, we focus on formulating our proposed group-sensitive MTL algorithms for AU detection including l_p-norm MTMKL and TD-MTMKL. In these two methods, AUs are packaged into several groups via our pre-knowledge of their co-occurrence relations, from which "group-sensitive" are named. Then such occurrence relations are modeled at both feature level and labeling level via different task structures. At the labeling level, we encode the AU relations via discriminative hyperplanes. MKL was incorporated in our methods for fusing multiple facial features and conduct the AU relation modeling at feature level. In our experiments, AUs in the same group are jointly detected by exploiting their co-existent relations. We compare the proposed group-sensitive MTL methods with several state-of-the-art methods on CK+ with posed AUs and DISFA with spontaneous AUs. The experimental results confirm the superiority of our MTL-based methods.

5.1 MTL for AU inter-relation modeling

In this section, we demonstrate the general idea of our designed MTL-based frameworks and analyze their feasibility for AU detection. In our work, the detection of each AU is viewed as a task, and we propose to simultaneously detect a set of AUs by exploiting their co-occurrence relations.

Our view is upon the fact that there exist commonalities among the classification tasks for multiple AUs. One instance of these commonalities as shown in Figure 5.1 can be that the same set of training data is usually shared and commonly used to learn the SVM hyperplanes for detecting different AUs. Another instance can be that there exists a main task among multiple AU detection tasks, which is to distinguish between the neutral faces and the occurrences of AUs.

Following these perspectives, we extend the Regularized MTL algorithm [59] to the l_p-norm MKL framework introduced in Section 4.2, and refer to it as l_p-norm MTMKL. The l_p-norm MTMKL learns the same shared kernel combinations from a given set of base kernels among all the tasks. In this case, the shared kernel weight vector can also be viewed as one of the commonalities across the tasks.

Actually, in humans' social interactions, similar emotions can be exhibited differently by subjects either via a single AU or a combination of AUs. Even an individual may show various combination of AUs for demonstrating the same emotions, such as the difference between Duchenne and polite smile. These imply that the AU relations defined in FACS are not always fixed, or at least the degree of the relations among AUs are not uniform. Thus, when detecting a set of AUs that are usually co-occurred in specific emotions, it is essential for the system to not only determine the commonalities across multiple AU detection tasks but also adapt to the task differences, or say, diversities.

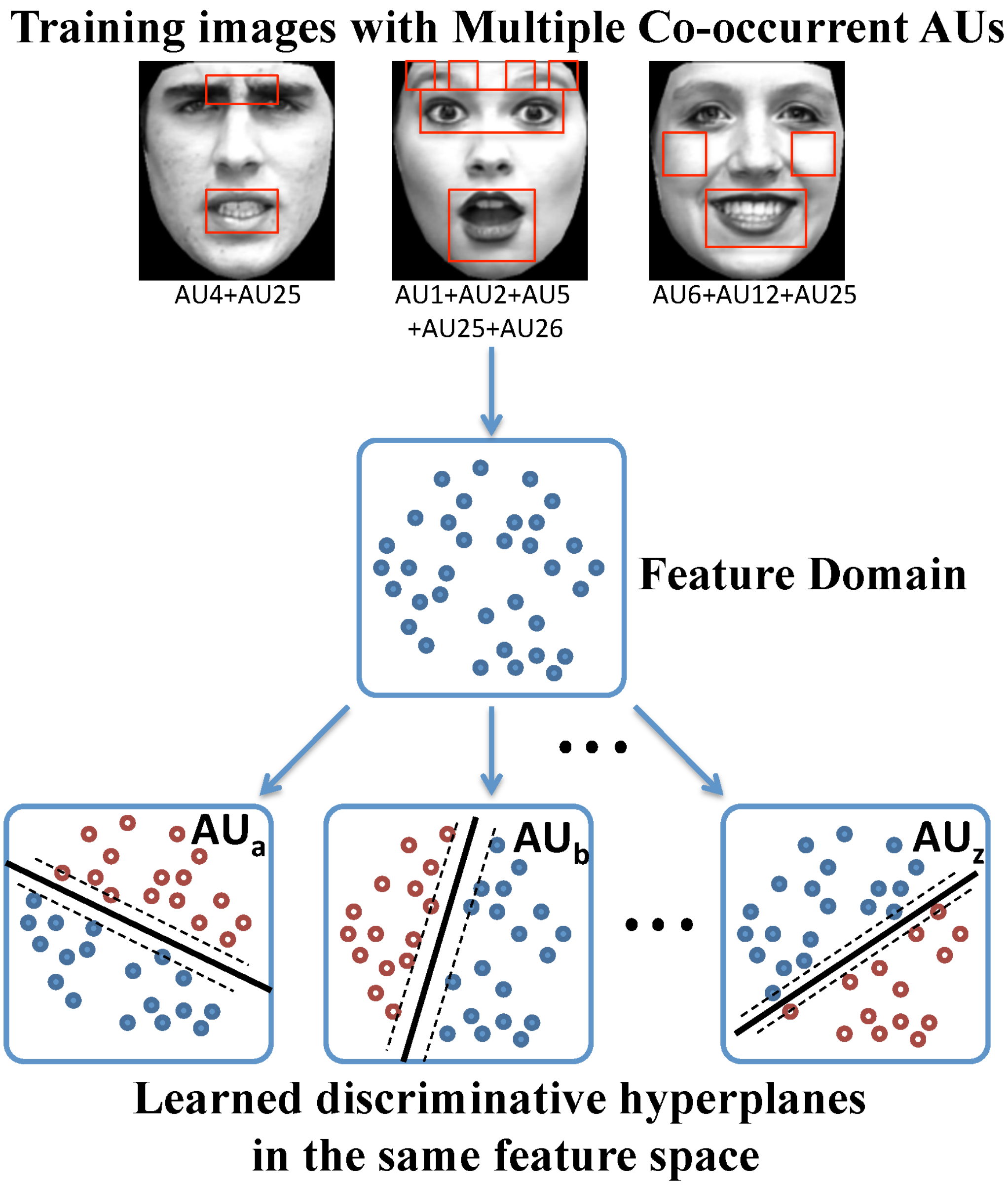

Training images with Multiple Co-occurrent AUs
AU4+AU25
AU1+AU2+AU5 +AU25+AU26
AU6+AU12+AU25
Feature Domain
• • •
AU$_a$
AU$_b$
AU$_z$
• • •
Learned discriminative hyperplanes in the same feature space

Figure 5.1: SVM classifier training in common AU detection systems

We present the TD-MTMKL method that learns an optimal kernel combination from a given set of basis kernels for each involved task and obtain a finer depiction of task relations through kernel combination weights. In this method, samples within a specific task share the same kernel weights while samples from different tasks may employ distinct sets of kernels. This “task-dependent” characteristic seeks to capture the AU commonalities through MTL meanwhile adapt to AU variations via kernel learning. By doing this, our proposed method can incorporate the benefits of both MTL and MKL, and identify local distributions in the training data from all AU detection tasks. Compared to l_p-norm MTMKL, TD-MTMKL captures both the commonalities and the variations among tasks at feature level via MKL, as the former enforces the same kernel combination weight vector across all tasks.

5.2 l_p-norm MTMKL

Let’s introduce the following notations for MTL and keep the symbols related with MKL the same as in Chapter 4: $\{(x_i, y_{it})\}_{i=1, t=1}^{N,\ T}$ denotes N training samples for simultaneously detecting T AUs, where $x_i \in \mathbb{R}^D$ is the feature vector of the i^{th} sample shared across all the tasks, and $y_{it} \in \{-1, +1\}$ is its corresponding class label for the t^{th} task (i.e., the detection of the t^{th} AU, and “+1” denotes presence while “-1” is for absence).

We write the direction of hyperplanes for every task $t \in \{1, \cdots, T\}$ in each $\mathcal{H}_m$ as

$$w_t^{(m)} = w_0^{(m)} + v_t^{(m)} \qquad (5.2.1)$$

where $w_0^{(m)}$ indicates the direction of the main task among all tasks in $\mathcal{H}_m$, and the vector $v_t^{(m)}$ represents the variation of each task to the main task. Equation 5.2.1 is defined based on the fact that 1) distinguishing between the absence and presence of AUs can be viewed as the main task; 2) by utilizing the same set of kernels and training samples, the discriminative hyperplanes of all tasks are in the same kernel space. That is, given $m \in \{1, 2, ..., M\}$, $\forall t \in \{1, 2, ..., T\}$, $w_t^{(m)} \in \mathcal{H}_m$.

The optimization problem of our proposed l_p-norm MTMKL is formulated as

$$\begin{aligned} \min_{w_t^{(m)}, \xi_{it}} \quad & C \sum_{i=1}^{N} \sum_{t=1}^{T} \xi_{it} + \frac{1}{2} \left[\sum_{m=1}^{M} \left(\lambda_m \| w_0^{(m)} \|_2^2 + \sum_{t=1}^{T} \| v_t^{(m)} \|_2^2 \right)^{\frac{p}{2}} \right]^{\frac{2}{p}} \\ s.t. \quad & y_{it} \left(\sum_{m=1}^{M} w_t^{(m)} \phi_m(x) \right) \geq 1 - \xi_{it}, \;\; \xi_{it} \geq 0 \\ & p \geq 1 \end{aligned} \tag{5.2.2}$$

where λ_m is a positive hyperparameter for controlling the difference among all tasks. To be specific, for any given m, a large value of λ_m, e.g., $\lambda_m > 100$, will make the solution of the mean function $w_0^{(m)}$ close to 0. In this case, all of the tasks tend to be unrelated as the commonality they share is tiny. Whereas, a small value of λ_m, e.g., $\lambda_m < 0.01$, will force all the tasks to be the same as the main task, as the solution of $v_t^{(m)}$ is much insignificant compared to $w_0^{(m)}$.

In essence, based on this definition the commonalities among AU detection tasks are encoded via the hyperplanes of their shared main task $\{w_0^{(m)}\}_{m=1}^{M}$ and the commonly utilized kernel set $\{\phi_m\}_{m=1}^{M}$. Moreover, $\{\lambda_m\}_{m=1}^{M}$ are defined to capture the task relations and should be tuned in experiments for an accurate estimation of the task relatedness. The optimization problem defined in Equation 5.2.2 can be reformulated as

$$\begin{aligned} \min_{\mathcal{W}_m, \xi_{it}} \quad & C \sum_{i=1}^{N} \sum_{t=1}^{T} \xi_{it} + \frac{1}{2} \left(\sum_{m=1}^{M} \| \mathcal{W}_m \|_2^p \right)^{\frac{2}{p}}, \;\; p \geq 1 \\ s.t. \quad & y_{it} \left(\sum_{m=1}^{M} \mathcal{W}_m \cdot \Phi_m(x_i, t) \right) \geq 1 - \xi_{it}, \;\; \xi_{it} \geq 0 \end{aligned} \tag{5.2.3}$$

where we define $\mathcal{W}_m$ and $\Phi_m(\cdot,\cdot)$ as

$$\begin{aligned} &\mathcal{W}_m = (\sqrt{\lambda_m} w_0^{(m)}, v_1^{(m)}, \cdots, v_T^{(m)}) \\ &\Phi(x,t) = (\frac{\phi_m(x)}{\sqrt{\lambda_m}}, \underbrace{\mathbf{0}, \cdots, \mathbf{0}}_{t-1}, \phi_m(x), \underbrace{\mathbf{0}, \cdots, \mathbf{0}}_{T-t}) \end{aligned} \tag{5.2.4}$$

Based on the above definition, our l_p-norm MTMKL formulation is transformed to be a single-task problem, and can be solved via the canonical l_p-norm MKL solver introduced in Algorithm 1 of Section 4.2. In our implementation, the optimality conditions are set based on number of total iterations and the variations of updated θ between consecutive iterations. In addition, the values of p, $\{\lambda_m\}_{m=1}^M$ and C are tuned based on cross-validation during experiments.

The learned discriminant hyperplanes of multiple tasks ($t = 1, ..., T$) in our proposed l_p-norm MTMKL is formulated as

$$f_t(\mathbf{x}) = \sum_{m=1}^{M} \theta_m^\star \underbrace{(\sum_{i=1}^{N}\sum_{s=1}^{T} \alpha_{is}^\star y_{is} k_{st}^{(m)}(\mathbf{x}_{is}, \mathbf{x}))}_{\text{hyperplane in each } \mathcal{H}_m} \tag{5.2.5}$$

where $\alpha_{is}^\star$ and $\theta_m^\star$ are learned optimum from Algorithm 1, and $k_{st}^{(m)}(\mathbf{x}_{is}, \mathbf{x}_{jt}) = (\frac{1}{\lambda_m} + \delta_{st})k_m(x_i, x_j)$. Here, δ_{st} is 1 if $s = t$ and 0 otherwise.

5.3 TD-MTMKL

The l_p-norm MTMKL method forces all the tasks share the same kernel weight vector, which may be too restrict, as kernels utilized in different tasks may employ distinct sets of kernels. In our work, we seeks a trade-off between capturing commonalities and adapting to variations in modeling AU relations, and learn one kernel weight vector for each task. The primal optimization problem of our TD-MTMKL can be formulated as:

$$\begin{aligned} \min_{w_0^{(m)}, v_t^{(m)}} \quad & C \sum_{i=1}^{N} \sum_{t=1}^{T} \xi_{it} + \frac{1}{2} \sum_{m=1}^{M} \left(\sum_{t=1}^{T} \|v_t^{(m)}\|_2^2 + \lambda_m \|w_0^{(m)}\|_2^2 \right) \\ s.t. \quad & y_{it} \left(\sum_{m=1}^{M} \theta_m^t \langle w_t^{(m)}, \phi_m(x_i) \rangle \right) \geq 1 - \xi_{it}, \;\; \theta_m^t, \xi_{it} \geq 0 \end{aligned} \tag{5.3.1}$$

Here, we refer to $\theta^t = (\theta_1^t, ..., \theta_m^t, ..., \theta_M^t)$ as the task-dependent kernel combination vector corresponding to the t^{th} task. Besides $\{\lambda_m\}_{m=1}^M$, another measurement of task relatedness is defined based on the angles (radians) between learned kernel combination vectors for different tasks as

$$\eta_{st} = arccos(\frac{|\langle \theta^s, \theta^t \rangle|}{\|\theta^s\|_2 \cdot \|\theta^t\|_2}), (s \neq t) \tag{5.3.2}$$

Therefore, $\eta_{st} \in [0, \frac{\pi}{2}]$. Based on these measurements, we investigate the performance of our method for the adaption to the AU detection task diversities.

The optimization problem in Equation 5.3.1 is solved based on its dual form, which is a min-max problem:

$$\begin{aligned} \min_{\theta^t} \max_{\boldsymbol{\alpha}} \;\; & J := \sum_{i=1}^{N} \sum_{t=1}^{T} \alpha_{it} - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{s=1}^{T} \sum_{t=1}^{T} \alpha_{is} \alpha_{jt} y_{is} y_{jt} K_\theta^{st}(x_i, x_j) \\ s.t. \;\; & \sum_{i=1}^{N} \sum_{t=1}^{T} \alpha_{it} y_{it} = 0, \;\; 0 \leq \alpha_{it} \leq C \end{aligned} \tag{5.3.3}$$

where $K_\theta^{st}(x_i, x_j) = \sum_{m=1}^{M} (\frac{1}{\lambda_m} + \delta_{st}) \theta_m^s \theta_m^t k_m(x_i, x_j)$ and $\boldsymbol{\alpha} = \{\alpha_{it}\}_{i=1, t=1}^{N, \ T}$ are Lagrange multipliers corresponding to the inequality constraints in the primal form of TD-MTMKL. In addition, we refer to $\{\theta^t\}_{t=1}^T$ as $\boldsymbol{\theta}$. Then, an alternating optimization approach is adopted:

Step 1: Fix $\boldsymbol{\theta}$, and optimize Equation 5.3.3 with respect to $\boldsymbol{\alpha}$;

Step 2: Fix $\boldsymbol{\alpha}$, and optimize Equation 5.3.3 with respect to $\boldsymbol{\theta}$;

Step 3: Iterate until convergence.

Notice that step 1 is equivalent to solving a standard SVM problem with $T \times N$ training data. In the following part, we focus on the second step, in which the optimization problem is defined as:

$$\min_{\boldsymbol{\theta}} \ J(\boldsymbol{\theta}) := \sum_{i=1}^{N}\sum_{t=1}^{T}\alpha_{it} - \sum_{s=1}^{T}\sum_{t=1}^{T}\sum_{m=1}^{M}(\frac{1}{\lambda_m} + \delta_{st})\theta_m^s\theta_m^t G_m^{st}(\boldsymbol{\alpha}) \tag{5.3.4}$$

where $G_m^{st}(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{is}\alpha_{jt}y_{is}y_{jt}k_m(x_i, x_j)$. Note that the product between θ_m^s and θ_m^t makes the optimization problem in Equation 5.3.4 non-convex. Inspired by [97], instead of solving $\boldsymbol{\theta}$ directly, we use a softmax gating function to guarantee the non-negativity of $\boldsymbol{\theta}$ and approximately approach the optimal solution. In our implementation, this function is defined as:

$$\theta_m^t = \frac{exp(p_m^t A_m^t + q_m^t)}{\sum_{m'=1}^{M} exp(p_{m'}^t A_{m'}^t + q_{m'}^t)} \tag{5.3.5}$$

where p_m^t and q_m^t are the parameters of the gating function, and A_m^t corresponds to a specific statistical property of training samples for the t^{th} task over the m^{th} kernel. In our work, A_m^t is defined as:

$$A_m^t = \frac{\sum_{i\in\Omega_t}\sum_{j\in\Omega_t} k_m(x_i, x_j)/n_{\Omega_t}^2}{\sum_{i'\in\Gamma_t}\sum_{j'\in\Gamma_t} k_m(x_{i'}, x_{j'})/n_{\Gamma_t}^2} \tag{5.3.6}$$

Here, the set $\Omega_t = \{a|\exists l \neq t, y_{al} \neq y_{at}\}$ while $\Gamma_t = \{b|\forall l \neq t, y_{bl} = y_{bt}\}$, and $n_{\Omega_t}, n_{\Gamma_t}$ denote the number of elements in the corresponding sets. In this definition, A_m^t characterizes the non-uniform task relations over the labels of training samples. Especially for our application of AU detection, it captures the relative relationships between the task diversities (inconsistence with other AUs co-occurrences) and the uniform co-occurrence of AUs across the training samples. In this case, by minimizing J over $\boldsymbol{\theta}$ we learn a finer depiction of AU task relation model through kernel combinations, which aims to capture the task commonalities and meanwhile adapt to task variations.

We take the derivatives of $J(\boldsymbol{\theta})$ with respect to p_m^t and q_m^t, and then is employ gradient-descent method to learn the gating model in Equation 5.3.5 by searching in the opposite direction of the derivatives. Note that $J(\boldsymbol{\theta})$ is differentiable due to the fact that all kernel matrices are strictly positive definite:

$$\begin{aligned} \frac{\partial J(\boldsymbol{\theta})}{\partial p_m^t} &= -2\sum_{m_0=1}^{M}(\delta_{mm_0} - \theta_{m_0}^t)\theta_m^t A_m^t \triangle_{m_0}^t(\boldsymbol{\alpha}) \\ \frac{\partial J(\boldsymbol{\theta})}{\partial q_m^t} &= -2\sum_{m_0=1}^{M}(\delta_{mm_0} - \theta_{m_0}^t)\theta_m^t \triangle_{m_0}^t(\boldsymbol{\alpha}) \end{aligned} \tag{5.3.7}$$

where $\triangle_{m_0}^t(\boldsymbol{\alpha}) = \sum_{t_0=1}^{T}(\theta_{m_0}^{t_0} G_{m_0}^{t_0 t}(\boldsymbol{\alpha}))$. After updating the gating model, we obtain a new $\boldsymbol{\theta}$ and send it to step 1 for the next iteration.

The optimization algorithm of TD-MTMKL with the designed gating function is summarized and shown in Algorithm 3.

The convergence criteria are set based on the consistency of $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ as well as the maximum number of iterations. $\mu^{(n)}$ and $\gamma^{(n)}$ control the step sizes of each iteration (the n^{th} iteration) and can be assigned as constants or determined with a 1D search method.

Once the final $\boldsymbol{\alpha}^\star$ and $\boldsymbol{\theta}^\star$ are determined, given a test sample $x \in \mathbb{R}^D$, the learned discriminant function of each task is:

$$f_t(x) = \sum_{m=1}^{M}\theta_m^{t\star}\sum_{i=1}^{N}\sum_{s=1}^{T}(\frac{1}{\lambda_m} + \delta_{st})\alpha_{is}^\star y_{is} k_m(x_i, x) \tag{5.3.8}$$

Algorithm 3 The TD-MTMKL Optimization Algorithm

```
Require: $\lambda_m, m \in \{1, 2, ..., M\}$
for all $m \in \{1, 2, \cdots, M\}, t \in \{1, 2, ..., T\}$ do
    initialize $p_m^t, q_m^t$ to small random numbers
end for
while convergence criteria are not satisfied do
    compute $\boldsymbol{\theta}$ based on Equation 5.3.5
    compute $K_{\theta}^{st}(x_i, x_j)$
    solve the canonical SVM with respect to $\boldsymbol{\alpha}$
    update: $p_m^t \Leftarrow p_m^t - \mu^{(n)} \frac{\partial J(\boldsymbol{\theta})}{\partial p_m^t}$
    update: $q_m^t \Leftarrow q_m^t - \gamma^{(n)} \frac{\partial J(\boldsymbol{\theta})}{\partial q_m^t}$
end while
Ensure: $\boldsymbol{\alpha}^{\star}$ and $\boldsymbol{\theta}^{\star}$
```

5.4 Facial action unit detection experiments

This section shows and discusses the experimental results of our proposed group-sensitive MTL-based AU detection frameworks on the CK+ and the DISFA databases. The comparison with several state-of-the-art methods are also given.

5.4.1 Classifier settings

In order to empirically study the advantage of our proposed l_p-norm MTMKL and TD-MTMKL for AU detection, we implement several benchmark classifier for comparison including canonical SVM, RMTL, l_p-norm MKL, l_1-norm MTMKL and Multiple Kernel Learning with Multiple Labels (MLMKL) [66]. Here, SVM and l_p-norm MKL are single task learning problem, which detect AUs separately. Whereas, the rest classifiers are MTL-based methods. Different from RMTL which utilizes single kernel, l_p-norm MTMKL, l_1-norm MTMKL, MLMKL and TD-MTMKL can fuse multiple types of facial features with different kernel functions. Furthermore, l_p-norm MTMKL and l_1-norm MTMKL employ uniform kernel weights across all the tasks while MLMKL and TD-MTMKL are designed to capture the non-uniform task relations. Compared to TD-MTMKL, MLMKL models the task relations only through kernel combinations without considering the relations among SVM hyperplanes.

For MKL-based classifiers, radial basis function (RBF) and polynomial function (poly) as defined in Equation 5.4.1 with LBPH and HOG features were utilized. Whereas for single kernel based classifiers, features and kernels that corresponded to the best recognition results on the validation data were applied to the test samples. We set different values for parameterizing kernel functions with the criterion that the parameters fill a proper range in their defined domain. For RBF, we set $\rho \in \{0.01, 0.1, 0.5, 1, 10, 50, 100\}$; for poly, we set $r \in \{1, 2, 3\}$.

$$\begin{aligned} k_{RBF}(x, y) &= e^{-\rho\|x-y\|_2^2}, \rho > 0 \\ k_{poly}(x, y) &= \langle x, y\rangle^r, r \in \mathbb{N} \end{aligned} \tag{5.4.1}$$

In our experiments, 10-fold cross-validation scheme was used for tuning the parameters of designed classifiers. The detailed information is listed in Table 5.1, where $C \in \{0.01, 0.1, 10, 100, 1000\}$, $\lambda \in \{0.05, 0.1, 0.5, 1, 25, 50\}$, $\beta \in \{0, \frac{T}{10}, \frac{T}{8}, \frac{T}{6}, \frac{T}{4}, \frac{T}{2}\}$ and $p \in \{1, 1.05, 1.2, 1.35, 1.5, 1.65, 1.8, 1.95, 2.1, 4, 8, 16\}$. For l_p-norm MTMKL, l_1-norm MTMKL and TD-MTMKL, we set the hyperparameters $\{\lambda_m\}_{m=1}^M$ as $\lambda_m = \lambda, \forall m \in \{1, 2, ..., M\}$. Hence, the SVM hyperplane in each $\mathcal{H}_m$ was equally weighted so that no one would dominate the others. In MLMKL, the hyperparameter $\beta \in [0, \frac{T}{2}]$ controls the degree of the kernel weight differences among multiple tasks. To be specific, $\beta = 0$ enforces uniform kernel weights across all tasks while $\beta = \frac{T}{2}$ implies that 0% kernels are commonly shared among the tasks. We report the value $\overline{\beta}_{MLMKL} = (1 - \frac{2\beta}{T}) \times 100\%$ to show the percentage of shared kernels over the entire set of basis kernels in MLMKL. This value measures the overall similarities among packaged AUs. The higher the percentage, the closer relations the tasks have.

Table 5.1: Information of our designed seven classifiers

Classifiers	Feature	Kernel	Parameter
SVM	LBPH or HOG	RBF or poly	C, ρ, r
RMTL	LBPH or HOG	RBF or poly	C, λ, ρ, r
l_p-norm MKL	LBPH and HOG	RBF and poly	C, p
MLMKL	LBPH and HOG	RBF and poly	C, β
l_1-norm MTMKL	LBPH and HOG	RBF and poly	C, λ
l_p-norm MTMKL	LBPH and HOG	RBF and poly	C, λ, p
TD-MTMKL	LBPH and HOG	RBF and poly	C, λ

5.4.2 AU packaging for MTL-based SVM

We apply MTL-based classifiers for simultaneous detection of multiple related AUs. Four AU groups including 10 AUs are designed for RMTL, l_p-norm MTMKL, l_1-norm MTMKL, MLMKL and TD-MTMKL as listed in Table 5.2. Our criterion for AU packaging is based on AUs that are usually co-occurred in facial emotions regardless of their locations on the face. AU1, AU2 and AU4 in G1 are upper face AUs that usually behave simultaneously to show negative expressions (e.g. fear and sadness). G2 contains both upper face AUs (AU6, AU12) and lower face AU (AU25), which are usually co-occurred in the joy expression with some variations between Duchenne smile and non-Duchenne smile. Moreover, AU15, AU17 and AU20 in G3 are lower face AUs that are often associated with negative expressions. Further, in order to obtain non-uniform degree of relatedness among packaged AUs and study its effect to our proposed methods, we add AU26 to G2 and refer the generated package as G4. Note that the criteria for packaging G4 are consistent to G2.

Table 5.2: The designed AU packages for MTL-based classifiers

G1	G2	G3	G4
AU1,2,4	AU6,12,25	AU15,17,20	AU6,12,25,26

5.4.3 Reliability measurement

The performance of our designed classifiers was evaluated using $F1$ score defined as

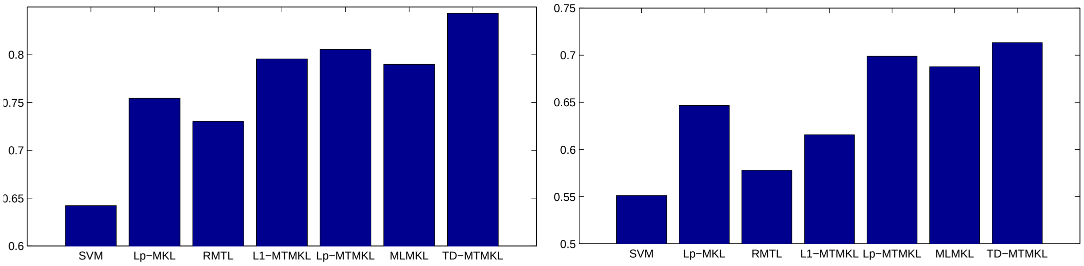
$$F1 = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision} \tag{5.4.2}$$

This reliability measurement considers and balances the recall and the precision rates. It is a better measurement than commonly used recognition rate in our case, due to the fact that it reflects the effect of the proportion of positive to negative samples among imbalanced test data [98].

5.4.4 Experimental results and discussions

In this section, the discussions of our experimental results are divided into two parts. The first part focuses on the comparison among the group-sensitive MTL-based classifiers and the single task classifiers. The experimental results of the involved classifiers are shown in Table 5.3 on the CK+ database and in Table 5.4 on the DISFA database. For single task classifiers (i.e. SVM and l_p-norm MKL), the detection results of AU6, AU12, and AU25 in G4 are reported based on those in G2. Table 5.5 reports the average value of best tuned hyperparameters during cross-validation for MTL-based classifiers. Figure 5.4 shows the learned AU similarities in G4 on both databases. The experimental results are discussed as follows.

a) General performance of MTL-based classifiers: From Table 5.3, Table 5.4 and Figure 5.2, we can see that compared to the canonical SVM, MTL-based classifiers can boost the average $F1$ of AUs in the first three packages on both databases. This confirms that exploiting AU co-occurrence relationships through MTL can generally increase the AU detection performance. Moreover, comparing MKL-based MTL methods (i.e. MLMKL, l_1-norm MTMKL, l_p-norm MTMKL and TD-MTMKL) with



(a) CK+ database

(b) DISFA database

Figure 5.2: Average F1 score of the first three packages

RMTL, we conclude that the fusion of multiple features with different kernels can enhance the performance of classifiers. This can also be observed based on the comparison between the results of canonical SVM and l_p-norm MKL.

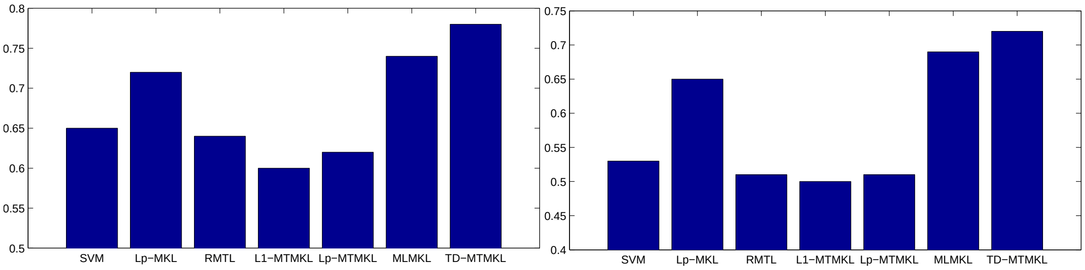
Table 5.3: Person-independent AU detection results ($F1$ score) on CK+

AUs	SVM	RMTL	l_p-MKL	MLMKL	l_1-MTMKL	l_p-MTMKL	TD-MTMKL
AU1	.63	.78	.83	.79	.83	.86	.88
AU2	.88	.80	.88	.87	.90	.90	.92
AU4	.64	.80	.86	.81	.86	.88	.89
Avg.G1	.72	.79	.86	.82	.86	.88	**.90**
AU6	.82	.84	.88	.89	.92	.93	.93
AU12	.72	.81	.86	.84	.90	.89	.90
AU25	.72	.76	.72	.75	.70	.73	.78
Avg.G2	.75	.80	.82	.83	.84	.85	**.87**
AU15	.43	.57	.47	.71	.66	.63	.75
AU17	.38	.61	.62	.73	.70	.74	.78
AU20	.56	.60	.67	.72	.69	.69	.76
Avg.G3	.46	.59	.59	.72	.68	.69	**.76**
AU6	.82	.75	.88	.84	.70	.72	.90
AU12	.72	.68	.86	.83	.62	.63	.89
AU25	.72	.66	.72	.77	.61	.65	.79
AU26	.32	.45	.43	.51	.48	.47	.55
Avg.G4	.65	.64	.72	.74	.60	.62	**.78**

b) Posed vs. Spontaneous AUs with MTL-based classifiers: From Table 5.5 we find that for both CK+ and DISFA databases, the values of the best tuned λ in RMTL, l_1-norm MTMKL, l_p-norm MTMKL and TD-MTMKL vary a lot among different AU

Table 5.4: Person-independent AU detection results ($F1$ score) on DISFA

AUs	SVM	RMTL	l_p-MKL	MLMKL	l_1-MTMKL	l_p-MTMKL	TD-MTMKL
AU1	.60	.62	.69	.70	.67	.72	.76
AU2	.52	.54	.56	.65	.59	.63	.67
AU4	.61	.61	.65	.66	.66	.69	.69
Avg.G1	.58	.59	.63	.67	.64	.68	**.71**
AU6	.54	.57	.69	.72	.64	.71	.71
AU12	.60	.63	.69	.71	.68	.76	.77
AU25	.47	.53	.71	.73	.56	.74	.75
Avg.G2	.54	.58	.70	.72	.63	**.74**	**.74**
AU15	.61	.60	.70	.68	.61	.72	.74
AU17	.53	.57	.55	.65	.59	.63	.63
AU20	.48	.53	.58	.69	.54	.69	.70
Avg.G3	.54	.61	.61	.67	.58	.68	**.69**
AU6	.54	.50	.69	.70	.53	.56	.76
AU12	.60	.57	.71	.74	.50	.52	.74
AU25	.47	.48	.66	.73	.49	.48	.78
AU26	.49	.47	.55	.58	.46	.46	.61
Avg.G4	.53	.51	.65	.69	.50	.51	**.72**



(a) CK+ database

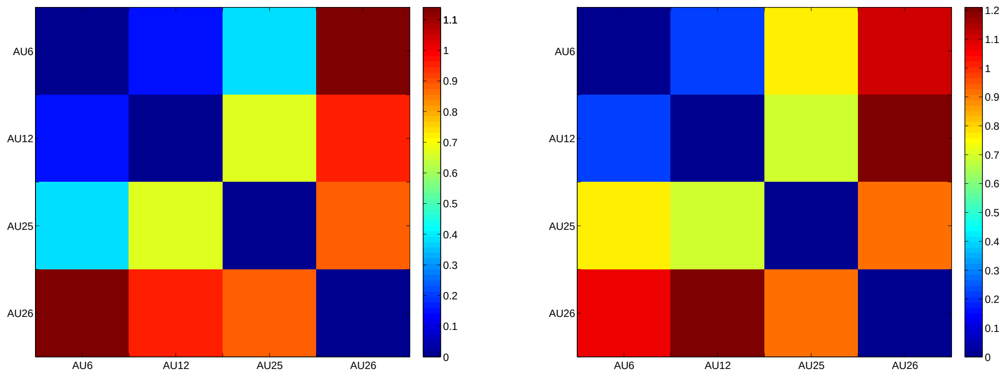
(b) DISFA database

Figure 5.3: Average F1 score of AUs in P4

packages. These results confirm the functionality of λ, which is to measure the general similarities between multiple tasks and their shared main task. Therefore, different AU packages may have different value of λ. Furthermore, for each AU package, the value of λ on the CK+ database is always less than that on the DISFA database. Since smaller λ makes the tasks close to their shared main task, our experimental results reinforced the fact that compared to posed AUs, spontaneous AUs contain more variations. This phenomenon can also be implied from the hyperparameter β in the MLMKL method, as for each AU package the value of $\overline{\beta}_{MLMKL}$ on the DISFA database is always less than that on the CK+ database.

c) Sparse vs. Non-sparse kernel combinations: As shown in Figures 5.2 and 5.3, on both posed and spontaneous face databases, the l_p-norm MTMKL outperformed the l_1-norm MTMKL in all four packages. This confirms the power of utilizing non-sparse kernel combinations in MKL-based classifiers for facial expression analysis. However, different from the first three packages, where we obtained good augmentation of detection accuracies by using arbitrary norms, the l_p-norm MTMKL just slightly boost the average $F1$ score of G4 from l_1-norm MTMKL on both databases, and both of these two methods did not perform well compared to other MTMKL methods. This phenomenon may be due to the non-uniform relatedness among AUs in G4.

d) Uniform vs. Non-uniform kernel combinations: As shown in Table 5.3 and Table 5.4, the l_1-norm MTMKL and the l_p-norm MTMKL with uniform kernel combinations did not perform well for the AUs in G4 on both databases, although it enhanced the detection results of the AUs in G2 from the canonical SVM. This problem may lie in the fact that by adding AU26 into G2, the task relations in G4 are quite non-uniform, since less kernels were commonly shared among tasks in G4 than in G2 (see $\overline{\beta}_{MLMKL}$ in Table 5.5). Compared to Table 5.5, Figure 5.4 gives a more visualized capture on the degree of relatedness between different AU detection tasks



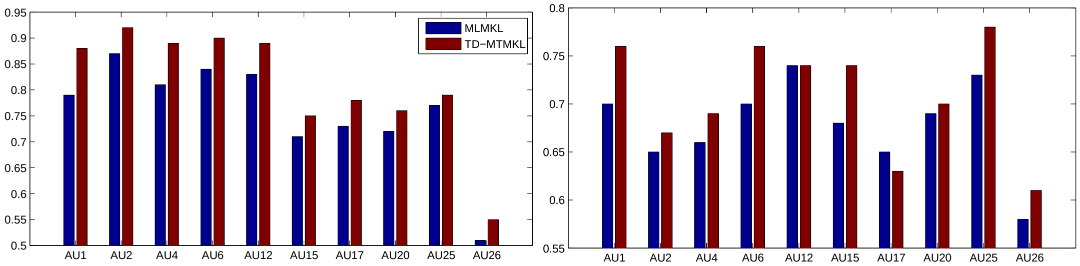
(a) CK+ database

(b) DISFA database

Figure 5.4: The calculated angles (radians) between kernel combination vectors for AUs in G4

(i.e. η_{st} in Equation 5.3.2) in G4. It is indicated that large variations were existed between the detection task of AU26 and the other three AUs in G4 on both CK+ and DISFA databases, as the angles between the kernel combination vector of AU26 and the other AUs are larger than the other pairwise vectors. Therefore, we can conclude that MTL-based method with uniform kernel combinations will not achieve good results when jointly detecting AUs with high relation diversities. Nevertheless, in this case MLMKL and TD-MTMKL can perform well, since they can adapt to task variations by learning different kernel combinations across tasks.

e) TD-MTMKL vs. MLMKL: As shown in Figure 5.5, our proposed TD-MTMKL method outperformed the MLMKL approach for all AU packages on both posed and spontaneous databases. Thus, we say that compared to MLMKL, the task structure in TD-MTMKL is more suitable for modeling the relations among multiple AU detection tasks, which are encoded based on both SVM hyperplanes and shared kernels. Moreover, our proposed TD-MTMKL can capture the relatedness of every pairwise AUs in each AU package via the angles between kernel combination vectors (i.e., η_{st} in Equation 5.3.2). In contrast, the MLMKL approach can only control the general



(a) CK+ database

(b) DISFA database

Figure 5.5: The comparison between TD-MTMKL and MLMKL

relatedness among all packaged AUs via the hyperparameter β. Therefore, we say that compared to MLMKL our proposed TD-MTMKL can obtain a finer depiction of AU relations.

Table 5.5: Average of best tuned hyperparameters in MTL-based SVM

Hyperparameters	G1		G2		G3		G4	
	CK+	DISFA	CK+	DISFA	CK+	DISFA	CK+	DISFA
λ_{RMTL}	0.72	1.96	1.57	3.88	2.60	3.36	3.16	7.91
$\lambda_{l_1-MTMKL}$	0.64	2.34	2.06	4.96	3.34	6.84	2.56	5.74
$\lambda_{l_p-MTMKL}$	0.47	2.04	2.88	6.94	3.34	2.84	2.86	4.57
$\lambda_{TD-MTMKL}$	0.49	1.37	1.66	4.33	2.37	5.20	4.03	7.21
$\overline{\beta}_{MLMKL}$	75%	66%	88%	72%	81%	58%	64%	41%

Based on the above discussion, we conclude that AU co-occurrence relations within the packaged set were properly modeled via discriminative hyperplanes in our proposed l_p-norm MTMKL and TD-MTMKL methods as we obtained higher AU detection accuracy than other group-sensitive MTL methods and single task classifiers. The MKL capability of l_p-norm MTMKL which fuse multiple facial features with different kernel functions can increase classification performance when AUs are almost uniformly related in the set. Whereas, it reduced the discriminative power for non-uniformly related AUs as it forces unique kernel combination weights across all the involved tasks. Typically, TD-MTMKL achieved a good trade-off between AU commonalities and diversities via its "task-dependent" character as it learns one kernel weight vectors for each task.

Table 5.6: Computational time of the classifiers' training phase in seconds

Classifier	Hyperparameter	Feature	CK+	DISFA
SVM	C, ρ, r	LBPH or HOG	45	67
RMTL	C, λ, ρ, r	LBPH or HOG	942	1627
l_p-MKL	C, p	LBPH and HOG	26	34
MLMKL	C, β	LBPH and HOG	143	252
l_1-MTMKL	C, λ	LBPH and HOG	483	697
l_p-MTMKL	C, λ, p	LBPH and HOG	862	1314
TD-MTMKL	C, λ	LBPH and HOG	732	1026

The time complexity of training RMTL is $O(T^3N^3)$ compared to the canonical SVM $O(TN^3)$. As MKL-based classifiers are solved based on alternative iterations, given the convergence termination criteria, the number of iterations depends on the training data and the searching step sizes. In our experiments it also takes time to tune several hyperparameters of the designed classifiers as well as the best facial features during cross-validation steps. Table 5.6 summarizes the average running time of our designed classifiers in each round of the 10-fold cross-validation step on a personal computer with Intel i5 CPU (2.66 GHz) and 8 GB memory.

Chapter 6

Hierarchial multi-task structure learning for facial action unit detection

In l_p-norm MTMKL and TD-MTMKL, in order to utilize the co-occurrence inter-relations among AUs, we have to preset several packages based on the relations between AUs and basic facial expressions, and simultaneously detect AUs in the same package. However, within these methods, we only employed the relations among AUs within the same package without considering the relationship across different packages. It is possible that different packages share several number of AUs but having non-identical AU relationships. Moreover, our AU packaging criterion is limited to the co-occurrence relations among AUs in the same basic facial expression, which turns to avoid of other possible AU inter-relations such as mutual exclusion and geometry locations. There might be other AU combinations that represent these relations among AUs, which can also help us increase the AU detection performance via MTL. To this end, we propose to design hierarchical task structures to introduce the relations across AU sets and learn the important AU combinations instead of pre-defining them in our MTL-based AU detection framework.

6.1 HMTSL

This section focuses on formulating the optimization problem of our proposed HMTSL. In our framework, the detection tasks of T AUs are jointly considered, and the task relations are modeled based on a hierarchical structure.

As shown in Figure 6.1, each leaf (marked in dark gray) of the hierarchical model denotes the detection task of a specific AU. The latent layer of the model is defined based on the father nodes of the leaves (marked in light gray). Here, each node in the latent layer corresponds to a subset of all tasks (leaves). In order to utilize the pre-knowledge of AU inter-relations, we can package all the AUs into several subsets based on some criteria, such as AUs' co-occurrence in basic expression as we did for group-sensitive MTL-based AU detection. We can even include all combinations of involved AUs in the lower layer to account for all possible AU relation variations.

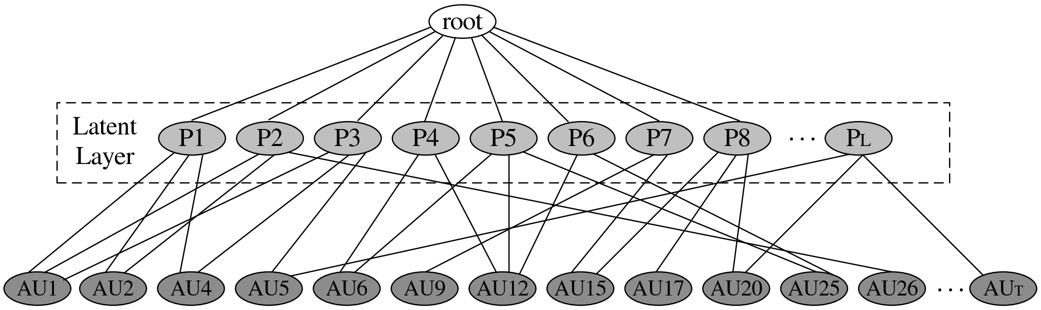


Figure 6.1: Our designed hierarchical structure

Suppose given L subsets out of T AUs denoted as $\{P_l\}_{l=1}^{L}$, we define a new kernel $K_{P_l}^{st}$ on each subset of tasks as follows:

$$K_{P_l}^{st}(x_i, x_j) = \begin{cases} k(x_i, x_j),\ if\ \{s, t\} \subseteq P_l \\ \quad 0, \qquad else \end{cases} \tag{6.1.1}$$

where $k(\cdot, \cdot)$ represents one of the standard kernels. The kernel representation across all nodes in the latent layer of our hierarchical model is formulated as

$$K_L^{st}(x_i, x_j) = \sum_{l=1}^{L} \theta_l K_{P_l}^{st}(x_i, x_j) \tag{6.1.2}$$

which is a weighted summation of kernels on all subsets of tasks. We define $\theta = (\theta_1, \theta_2, ..., \theta_L)$ as the combination weight vector of nodes in the latent layer.

Based on this task structure, the model of each AU detection task can be represented by a weighted combination of their corresponding father nodes in the latent layer, and the common information (feature representations of training samples) across leaves is also shared and utilized among their father nodes. The relatedness between two AUs (t and t') can be determined based on the number of subsets that include both t and t' as well as the importance of each of these subsets (nodes in the latent layer) captured by their corresponding combination weights.

In order to jointly learn multiple AU detection classifiers with linearly combined kernel representations across the nodes in the latent layer of our designed hierarchical model, we follow the line of research in [63], and cast our framework into the following

MTMKL problem:

$$\begin{aligned}\min_{\theta}\max_{\alpha}\ & \sum_{i=1}^{N}\sum_{t=1}^{T}\alpha_{it}-\frac{1}{2}\sum_{i,j=1}^{N}\sum_{s,t=1}^{T}\alpha_{is}y_{is}\alpha_{jt}y_{jt}K_L^{st}(x_i,x_j)\\ s.t.\ & \sum_{i=1}^{N}\sum_{t=1}^{T}\alpha_{it}y_{it}=0,\ 0\leq\alpha_{it}\leq C\\ & \|\theta\|_q\leq 1,\ \theta_l\geq 0\end{aligned} \tag{6.1.3}$$

where C is a positive constant preset to control the relative influence of non-separable samples as in canonical C-SVM, and the l_q-norm constrains the sparsity of θ. α denotes the set $\{\alpha_{it}\}_{i=1,t=1}^{N,\ T}$.

We set $q=\frac{p}{(2-p)}, p\in[1,2)$, then the optimization problem defined in Equation 6.1.3 is equivalent to the one defined in Equation 4.2.2 in the case of $p\in[1,2)$, and can be solved via standard l_p-norm MKL solver introduced in Algorithm 1 of Section 4.2. In our implementation, the convergence criteria are set based on the consistency of α and θ as well as the maximum number of iterations, and the values of q and C are tuned via cross-validation during experiments. Once the optimal $\alpha^\star$ and $\theta^\star$ are determined, the learned discriminant function of each task ($t\in\{1,2,...,T\}$) is:

$$f_t(x)=\sum_{i=1}^{N}\sum_{s=1}^{N}\sum_{\{s,t\}\subseteq P_l}\alpha_{is}^\star y_{is}\theta_l^\star K_{P_l}^{st}(x_i,x) \tag{6.1.4}$$

where $x\in\mathbb{R}^D$ is a given test sample.

Since the learned combination weight $\theta_l^\star$ reflects the importance of its corresponding subset P_l over all AU subsets. The similarity between two tasks s and t ($s\neq t$) is defined as follows:

$$\eta_{st}=2\cdot\frac{\sum_{P_l\supseteq\{s,t\}}\theta_l^\star}{\sum_{s',t'}\eta_{s't'}}, s\neq t \tag{6.1.5}$$

Here, $\eta_{st} = \eta_{ts}$. According to this definition, if two tasks are often jointly present in several AU subsets with high combination weights, the calculated value of their similarity is also high. Therefore, this AU similarity measurement provides insight information of the task relations, as a high value of η_{st} reflects a close resemblance between the tasks s and t. Based on this measurement, we are able to investigate the performance of our method on the adaption to non-uniform AU relations.

6.2 Hierarchical model in HMTSL

In this work, we implement our HMTSL via two hierarchical model. One is designed based on our pre-knowledge of AUs' co-occurrence relations while the other considers all possible AU combinations. We refer the former one as knowledge-based HMTSL (KB-HMTSL) and the latter one as knowledge-free HMTSL (KF-HMTSL).

In KB-HMTSL, eight packages including 12 AUs are designed as shown in Figure 6.2. Of all the eight nodes in the latent layer, AU1, AU2 and AU4 corresponding to P_1 are upper face AUs that usually behave simultaneously to show negative expressions (e.g. "Fear" and "Sadness"). P_2 contains both upper face AUs (AU1, AU2) and lower face AU (AU26), which are usually co-occurred in the surprise expression. P_3 is with AU1, AU4 and AU5 which are often jointly behaved across the facial expressions "Sadness", "Fear" and "Anger". P_5 includes AU6, AU12 and AU25, which are usually co-occurred in the joy expression with some variations between the Duchenne smile (P_4 with AU6 and AU12) and the non-Duchenne smile (P_6 with AU12 and AU25). Moreover, P_7 with AU9 and AU15 defines the disgust expression. Finally, AU15, AU17 and AU20 associated with P_8 are lower face AUs that are often associated with negative expressions.

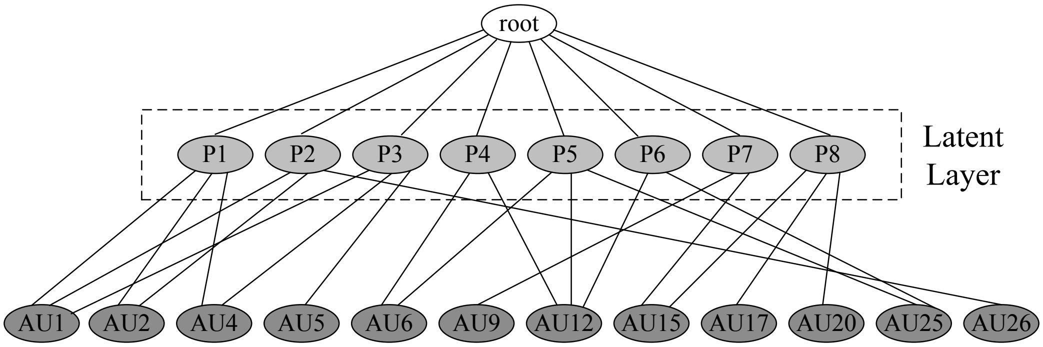


Figure 6.2: Our designed hierarchical model with eight subsets out of 12 AUs

For KF-HMTSL, we divided the involved 12 AUs into all possible subsets, and the number of subsets L is

$$L = C_T^1 + C_T^2 + \cdots + C_T^T = 2^T - 1 \ \ (T = 12) \tag{6.2.1}$$

In our experiments, we compared these two HMTSL implementations in order to achieve a deep understanding of the commonalities and variations of AU inter-relations in posed and spontaneous expressions and also find the learned and exploited important AU combinations with high combination weights.

6.3 AU detection experiments and discussions

In this section, we compare KB-HMTSL and KF-HMTSL as well as TD-MTMKL, which achieved the best AU detection results in the proposed group-sensitive MTL. We follow the same experimental settings as in Section 5.4.1 including kernel functions (RBF and poly) and the 10-fold cross-validation schema for pruning C of SVM, ρ, r of kernels and $q \in \{1, 1.05, 1.2, 1.35, 1.5, 1.65, 1.8, 1.95, 2.1, 4, 8, 16\}$ of MKL.

The comparison of experimental results on the CK+ database and the DISFA database are shown in Figure 6.3 and Figure 6.4 as well as Figure 6.5. For TD-

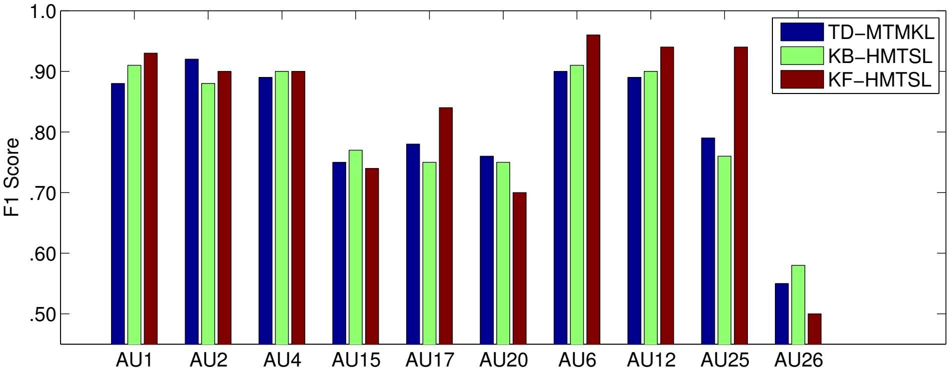


Figure 6.3: Comparison among TD-MTMKL (best), KB-HMTSL and KF-HMTSL on CK+

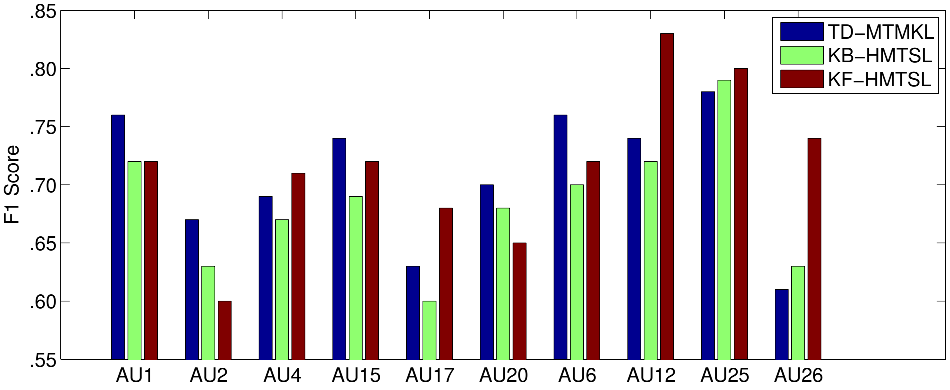
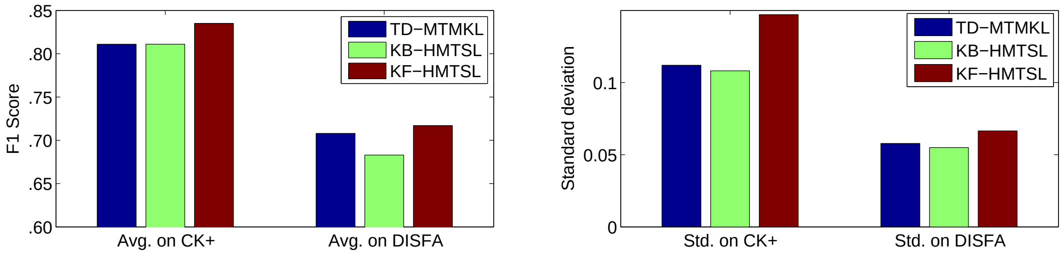


Figure 6.4: Comparison among TD-MTMKL (best), KB-HMTSL and KF-HMTSL on DISFA

MTMKL (best), the best AU detection results in G2 and G4 are chosen. Figure 6.6 and Figure 6.8 illustrate the captured AU relatedness for both posed and spontaneous AUs. Figure 6.7 and Figure 6.9 show the eight highly weighted AU subsets and their corresponding weights in the latent layer of the hierarchical model in both KB-HMTSL and KF-HMTSL. Table 6.2 list the corresponding AU subsets in Figure 6.9. The experimental results are discussed as follows.

TD-MTMKL models the non-uniform relations among AUs in the same package, and outperformed the other group-sensitive classifier – l_p-norm MTMKL. Whereas,



(a) Average F1 score

(b) Standard deviation

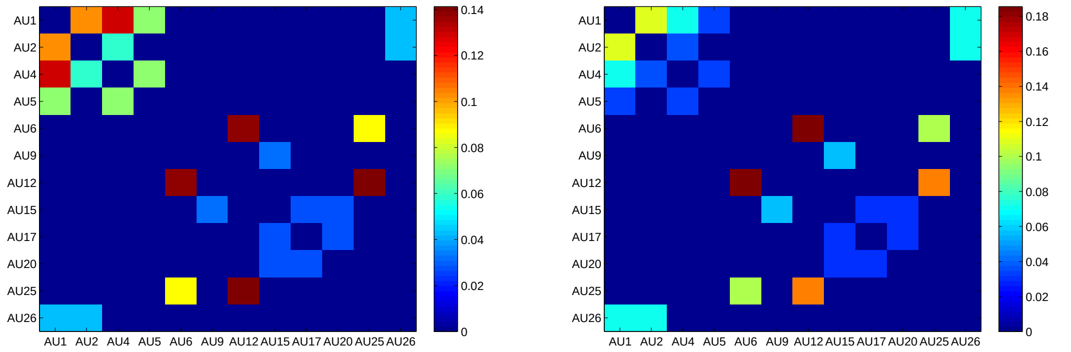
Figure 6.5: Average and standard deviation of F1 scores on CK+ and DISFA

HMTSL methods utilize AU relations within each group and across different groups via hierarchical structure learning. Compared with TD-MTMKL and KB-HMTSL, KF-HMTSL achieved the best average AU detection results on both databases, of which 6 out of 10 F1 scores are the highest on CK+ and 5 out of 10 on DISFA. This confirms the superiority of KF-HMTSL, which exploited various AU inter-relations besides the co-occurrence one in the other methods. KB-HMTSL learns the hierarchical model with pre-defined AU groups in its latent layer, and obtained identical performance on CK+ while 5.6% less on DISFA compared to TD-MTMKL. Nevertheless, as shown in Table 6.1 compared to KF-HMTSL and TD-MTMKL, KB-HMTSL achieved comparable AU detection performance with much less computational time (about 27% of TD-MTMKL and 40% of KF-HMTSL).

Table 6.1: Computational time of TD-MTMKL, KB-HMTSL and KF-HMTSL during training phase in seconds

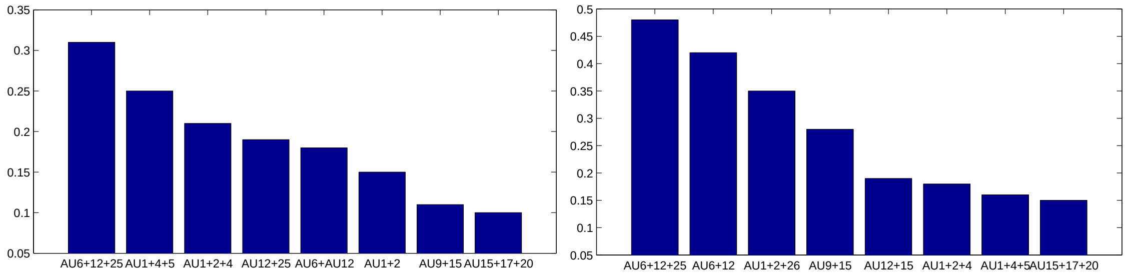
Classifier	Hyperparameter	Feature	CK+	DISFA
TD-MTMKL	C, λ	LBPH and HOG	732	1026
KB-HMTSL	C, q, ρ, r	LBPH or HOG	193	286
KF-HMTSL	C, q, ρ, r	LBPH or HOG	452	764

There are several key outcomes in this comparison. Firstly, TD-MTMKL benefited from the fusion of multiple features with different kernels while HMTSL methods utilized only single feature and single kernel. Secondly, KB-HMTSL only exploited



(a) CK+ database ($q = 1.25$) (b) DISFA database ($q = 1.70$)

Figure 6.6: AU similarities in KB-HMTSL on CK+ and DISFA

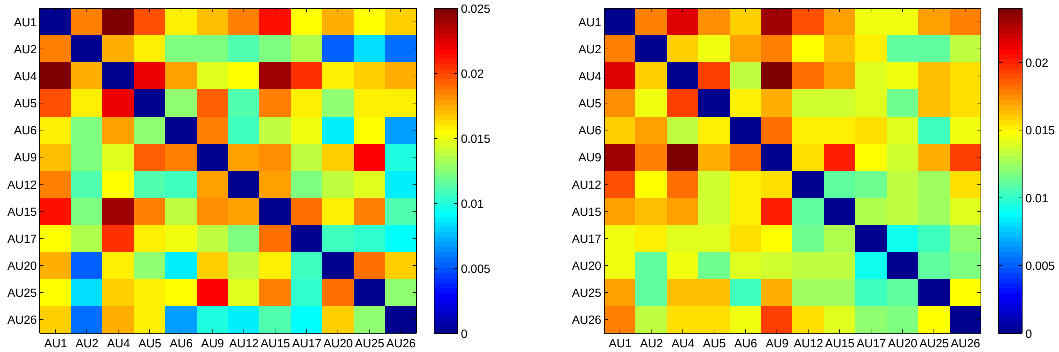


(a) CK+ database ($q = 1.25$) (b) DISFA database ($q = 1.70$)

Figure 6.7: AU subset weights in KB-HMTSL on CK+ and DISFA

the AU relations from the pre-defined AU subsets based on our pre-knowledge of AU co-occurrence relations whereas KF-HMTSL considered all possible AU combinations and thus obtain more information from data. Actually, from Table 6.6, we can see that the AU inter-relations are more diverse than what we defined in the latent layer of KB-HMTSL.

Moreover, the KF-HMTSL classifier is essentially a data-driven method which captures the importance of AU subsets via the learned combination weights from training data. As shown in Table 6.2, the most salient AU combinations learned in KF-HMTSL are different from what we defined in KB-HMTSL. This is mainly due to the variations of human facial activities. Thus, KB-HMTSL capturing very



(a) CK+ database ($q = 1.05$)

(b) DISFA database ($q = 1.17$)

Figure 6.8: AU similarities in KF-HMTSL on CK+ and DISFA

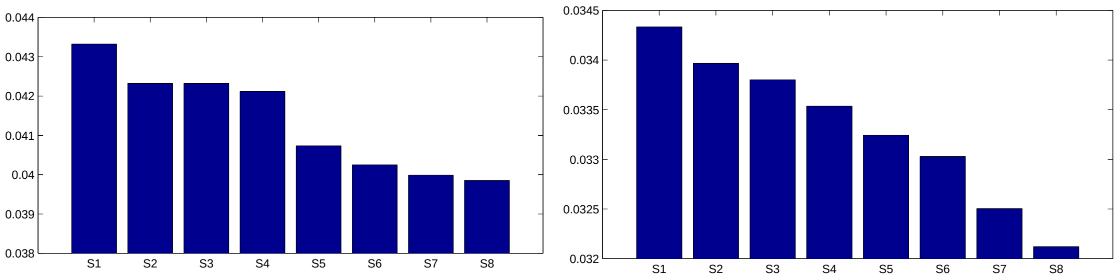
limited information of AU inter-relations did not achieve better performance than TD-MTMKL and KF-HMTSL.

In addition to the comparison of classification performance, we can also obtain some valuable findings about AU inter-relations from our experimental results. For one thing, in both HMTSL methods, the values of best tuned q on the CK+ database are lower than those on the DISFA database, which means that the optimized combination weights of AU subsets for posed AUs are more sparse than spontaneous AUs. Especially, the best $q = 1.05$ in KF-HMTSL on DISFA. This indicates that the number of important AU combinations learned from posed AUs is relatively less than that of spontaneous ones. This result reinforces our common understanding of spontaneous expressions' characteristics – various AU relationship.

For the other thing, as shown in Table 6.2 the AU inter-relations reflected from the learned important AU subsets in HMTSL are not limited to the co-occurrence relationship. The mutually exclusive relationship between AUs is also encoded in some of the AU subsets. For examples, AU12 (Lip Corner Puller) and AU15 (Lip Corner Depressor) in S1 and S5 on CK+ as well as AU5 (Upper Lid Raiser) and AU6 (Cheek Raiser) in S4 on DISFA. Since salient AU subsets have strong influence

Table 6.2: Eight highly weighted AU subsets on CK+ and DISFA

AU subset	CK+	DISFA
S1	AU1,2,4,5,12,15,17	AU4,9,12,20,25,26
S2	AU1,20,25,26	AU1,2,9,15,17,26
S3	AU5,9,15,20,25	AU1,2,4,5,12,17,20,25
S4	AU1,2,4,5,26	AU1,2,5,6,9,15,17,25,26
S5	AU1,2,4,6,9,12,15,25	AU1,2,4,5,12,25,26
S6	AU1,4,5,15,20,25,26	AU1,2,6,9,12,17,26
S7	AU1,2,5,9,12	AU1,6,12,20,26
S8	AU4,5,17,20,26	AU2,4,15,20



(a) CK+ database ($q = 1.05$)

(b) DISFA database ($q = 1.17$)

Figure 6.9: Eight highest weights in KF-HMTSL on CK+ and DISFA

on the classification outputs, the AU inter-relations embedded in these sets can also contribute to the AU labeling results of samples. Thus, we say that KF-HMTSL outperformed other benchmark classifiers in Table 5.1 by exploiting various AU relationships including AU co-occurrence and mutually exclusive relations.

Table 6.3 and Table 6.4 compare the performance of HMTSL methods with several state-of-the-art methods reported in the literatures. Papers [12] and [13] used Gentleboost SVM and HMM as their classifiers, and different AUs were separately detected. The classifiers of [47] and [48] are Adaboost SVM with DBN which considered the inner-relations among multiple AUs. In [99], the authors packaged the AU sets based on their geometry locations on the face (i.e. eye, mouth and chin, cheek and nose). In their work, MTFL [61] was applied for feature learning against AUs in the same group. Afterwards, BN was applied to revise the AU labels from MTFL via AU occurrence dependencies. As we can see, our KF-HMTSL method achieved the best average F1 score on both databases, which confirms its effectiveness. Especially, on the CK+ database, 10 out of 12 AUs were detected with higher accuracy by the proposed HMTSL methods than the state-of-the-art methods. On the DISFA database, the standard deviation of our KF-HMTSL classifier is much lower than the one with MTFL and BN. This comparison confirms the strong robustness and reliability of our method across different AUs.

In this dissertation, the proposed MTL-based classifiers are all static machine learning models as time relations of sample labels or classifier outputs are not formulated in their optimization problems. This is mainly due to the fact that our methods are MTMKL extensions of canonical SVM which is also a static classifier. In Section 4.4.4, we discussed the necessity of capturing temporal information across video frames in facial expression analysis, and proposed to fuse dynamic features in our frameworks towards this challenge. Besides this feature-level proposal, we can

Table 6.3: The comparison ($F1$ score) with the state-of-the-art methods on CK/CK+ (reported in literatures)

AU	KB-HMTSL	KF-HMTSL	[12]	[13]	[47]	[48]
AU1	.91	**.93**	.87	.83	.66	.78
AU2	.88	**.90**	.90	.83	.57	.80
AU4	.90	**.90**	.73	.63	.71	.77
AU5	.74	**.81**	.80	.60	–	.64
AU6	.91	**.96**	.80	.80	.94	.77
AU9	**.82**	.78	.77	.57	–	.79
AU12	.90	**.94**	.84	.84	.88	.89
AU15	.77	.74	.70	.36	.84	.70
AU17	.75	**.84**	.76	–	.79	.81
AU20	.75	.70	.79	.52	–	–
AU25	.76	.94	.96	.75	–	.88
AU26	**.58**	.50	–	.36	–	–
Avg.	.81	.83	.81	.64	.77	.78

Table 6.4: The comparison ($F1$ score) with the state-of-the-art method on DISFA (reported in the literature)

AU	KB-HMTSL	KF-HMTSL	[99]
AU1	.72	**.72**	.61
AU2	.63	.60	.79
AU4	.67	.71	.80
AU5	.55	**.60**	.39
AU6	.70	**.72**	.71
AU9	.63	**.75**	.65
AU12	.72	.83	.96
AU15	.69	.72	.77
AU17	.60	.68	.81
AU20	**.68**	.65	.35
AU25	.79	.80	.90
AU26	.63	.74	.78
Avg.	.67	.71	.71
Std.	.064	.068	.185

also model the temporal relations of facial expressions and AUs at the labeling level. This idea as reviewed in Section 2.3 was implemented based on the use of HMM and DBN. Recently, kSeg-SVM [11] was proposed to cast the AU detection as a problem of recognizing temporal events from video frames. In their work, each video sequence was represented as a time series of facial feature segments, and then structured output SVM [100] was applied to learn the temporal model between the segments. Inspired by kSeg-SVM, we can incorporate structured SVM into MTMKL problems for facial expression analysis in dynamic modes.

Chapter 7

Conclusion and future work

In this dissertation, we present transfer learning algorithms, MKL and MTL, for facial expression analysis including basic facial expression recognition and AU detection. Our methods achieve the promising performance compared with the state-of-the-art methods on four public face databases with posed and spontaneous facial expressions. The key points and contributions of this dissertation are summarized in this chapter. We also present the limitations of the proposed facial expression analysis frameworks and give recommendations for future work.

7.1 MKL for basic expression recognition

For basic facial expression recognition, we employ the idea of feature domain adaption in the transfer learning framework and fuse multiple types of facial features (LBPH and HOG) with different kernel functions (HtRBF and polynomial function) via MKL to increase the discriminative power of SVM over canonical SVM. MKL learns the kernel combination weights within SVM classifiers and obtains the optimal feature representation for classification. The learned kernel combination weights indicate their importance to the classification output. The l_p norm is utilized to constrain the kernel weight vector and obtain both sparse and non-sparse kernel combinations.

Moreover, the proposed l_p-norm MKL multiclass-SVM learns one kernel weight vector for each binary classifier in the multiclass-SVM. In contrast, the SimpleMKL-based multiclass-SVM jointly learns the same kernel weight vector for all binary classifiers. Thus, our method achieves more flexibility in utilizing different kernel combinations for distinguishing between different expressions. We prove that our method preserves the lower boundary of SimpleMKL's objective function, which is to be minimized during learning process.

In general, our method fuses different facial features at the kernel level of SVM, which turns to bridge the gaps between the feature selection/learning and the classification steps in facial expression recognition frameworks. That is, instead of empirically finding which feature or kernel worked the best for an expression, we jointly utilize a set of features and kernels and learn the optimal combination of them within the expression classifiers.

In our experiments, we compare our l_p-norm MKL multiclass-SVM with several state-of-the-art single-kernel-based classifiers and the SimpleMKL-based multiclass-SVM with one kernel weight vector for all binary classifiers. Experimental results on three face databases, CK+, MMI and GEMEP-FERA, confirm the superiority of our method over the others. We also comprehensively study the effect of p on the recognition performance, and concluded that non-sparse kernel combinations outperformed the sparse ones by utilizing more discriminative information from fused features and kernels.

7.2 MTL for AU detection

For facial action unit detection, we propose three MTL-based transfer learning methods, l_p-norm MTMKL, TD-MTMKL and HMTSL, for simultaneous detection of multiple facial AUs by exploiting their inter-relations. In our approaches, the detection of each AU is viewed as a task, and the relations among multiple tasks are modeled based on their commonalities and variations. One instance of the task commonalities is their shared main task to distinguish between neutral faces and presences of AUs. The other commonality is the commonly shared feature representations among the tasks.

We propose the group-sensitive MTL including l_p-norm MTMKL and TD-MTMKL to model the AU co-occurrence relations at both feature utilization level and AU labeling level. At labeling level, these methods represent the discriminative hyperplane for each task via the main task and its variation to the main task. At feature level, the l_p-norm MTMKL extends the regularized MTL algorithm to an MKL problem for fusing multiple facial features and enforces all tasks to share the same kernel combinations in MKL. The task-dependent property of our TD-MTMKL method is designed to adapt to the non-uniform degree of AU relatedness, and is conducted via learning nonidentical kernel combination weights across the AU detection tasks. The group-sensitive MTL are limited to packaging AUs based on our pre-knowledge of their co-existent relations, and do not consider the AU relations from different packages.

We propose HMTSL to exploit the relationship across different AU sets via hierarchical structures and utilize all possible AU inter-relations besides the co-occurrence one. Variants of AU combinations were linearly combined in the latent layer of the hierarchical model in HMTSL, and each encodes a specific AU inter-relation. The combination weights are learned during the optimization of HMTSL, which represent the importance of their corresponding AU combinations to the AU detection results. In this structure, the relations between pairwise AUs are captured based on the number of their shared subsets as well as the importance of these subsets.

We comprehensively study the effectiveness of our methods on both posed and spontaneous AUs, and obtain deep understanding of AU relation commonalities and variations via AU similarity measurements. Extensive experiments confirm the superiority of our methods over several state-of-the-art single-task-based and MTL-based classifiers for AU detection. Especially, our proposed HMTSL method with hierarchical task structures outperforms the proposed group-sensitive MTL methods and other state-of-the-art MTL-based methods, which imply that exploiting various AU relations instead of just co-occurrence ones helps increase the discriminative power of MTL-based AU detector.

7.3 Future recommendations

At the current stage, facial features utilized in our work, LBPH and HOG, are extracted only from static images and do not capture the dynamic facial information across consecutive video frames. Moreover, our proposed classifiers, either for multi-class classification in basic expression recognition or multi-task learning in simultaneous detection of multiple AUs, are all static machine learning problems without modeling the temporal relations of the discriminative functions or labels of samples from time series. Since facial expressions are dynamic facial activity events, it is worth considering the challenges of facial expression analysis in a dynamic schema. In the future, it is recommended that

- at feature level: facial features with good dynamic information of consecutive video frames, such as LGBP-TOP and LBP-TOP, are utilized in the proposed expression analysis frameworks.

- at classification level: dynamic MKL, MTL and MTMKL classifiers are proposed by extending the structured SVM instead of the canonical SVM in this work, as structured SVM models the output of SVM via variants of dynamic structures and can use the temporal information from consecutive samples.

Both of these approaches will build upon the contributions of this dissertation to extend the current works, and further improve the state-of-the-art in facial expression analysis.

Appendix A

Proof of the superiority of MKL-based SVM over canonical binary SVM with single kernel and single type of features

Without loss of generality, our proof is pursed in the case of $1 < p < 2$. We transform the object function of Equation 4.2.1 based on the Lemma 26 in [101] as:

$$\min_{\theta, \|\theta\|_r \leq 1} \min_{w, w_0, \xi} \quad J(\theta, w, w_0, \xi) = \frac{1}{2} \sum_{m=1}^{M} \frac{\|w_m\|_2^2}{\theta_m} + C \sum_{i=1}^{N} \xi_i$$

where $r = p/(2 - p)$.

As described in Section 4.2, this convex optimization problem is solved by the Two-Step method, where two nested iterations are equipped in each loop of the method. In the outer iteration the kernel combination weights are updated by fixing the parameters of SVM. Whereas, in the inner iteration the optimization problem of canonical SVM is solved by fixing the updated kernel combination weights. Let N_f be the number of features extracted from each sample and N_k be the number of kernel functions used in the l_p-norm MKL-based SVM. We denote the updated kernel combination vector in the t^{th} loop of the Two-Step method as follows.

$$\theta^{(t)} = [\underbrace{\theta_1^{(t)}, \cdots, \theta_{N_k}^{(t)}}_{\text{the } 1^{st} \text{ feature}}, \cdots, \underbrace{\theta_{(i-1)N_k+1}^{(t)}, \cdots, \theta_{i \cdot N_k}^{(t)}}_{\text{the } i^{th} \text{ feature}}, \cdots, \underbrace{\theta_{(N_f-1)N_k+1}^{(t)}, \cdots, \theta_{N_f N_k}^{(t)}}_{\text{the } {N_f}^{th} feature}]^T$$

$$\theta^{(t)} \in \mathbb{R}_+^{\star N_f N_K}, \|\theta^{(t)}\|_r = 1$$

In addition, the SVM discriminant hyperplane obtained in the outer iteration of the t^{th} loop is denoted based on $w^{(t)}$ and $w_0^{(t)}$.

For the canonical binary SVM, we suppose that the i^{th} feature with the j^{th} kernel function is utilized. Then the canonical SVM becomes a special case in the framework of MKL-based SVM, and its corresponding kernel combination vector can be defined as follows.

$$\hat{\theta} = [\underbrace{0, 0, \cdots, 0, \cdots, 0}_{\theta_1 \sim \theta_{(i-1)N_k+j-1}}, 1, \underbrace{0, 0, \cdots, 0, \cdots, 0}_{\theta_{(i-1)N_k+j+1} \sim \theta_{N_f N_k}}]^T$$

Further, the learned discriminant hyperplane of canonical SVM is defined based on $\hat{w}^\star$ and $\hat{w}_0^\star$.

By assuming that in the first loop of the Two-Step method $\theta^{(1)}$ is initialized as $\hat{\theta}$ in the outer iteration, we obtain that in the inner iteration of the first loop the learned $w^{(1)} = \hat{w}^\star$ and $w_0^{(1)} = \hat{w}_0^\star$. Thereafter, our proof is formulated as follows,

$$\begin{aligned} \hat{J}^\star = J(\hat{\theta}, \hat{w}^\star, \hat{w}_0^\star) &= J(\theta^{(1)}, w^{(1)}, w_0^{(1)}) \\ &\geq J(\theta^{(2)}, w^{(1)}, w_0^{(1)}) \geq J(\theta^{(2)}, w^{(2)}, w_0^{(2)}) \\ &\geq \quad \cdots \quad \geq J(\theta^\star, w^\star, w_0^\star) = J^\star \end{aligned}$$

where $J^\star$ is the learned minimum of the objective function in l_p-norm MKL-based SVM with its corresponding optimum $\theta^\star, w^\star, w_0^\star$, and $\hat{J}^\star$ with $\hat{w}^\star, \hat{w}_0^\star$ is for canonical binary SVM.

Based on the above justification, we can naturally extend the conclusion to a more general case. That is:

Suppose $\exists$ a set of basis kernel functions S ($S \neq \emptyset$) and a set of features F ($F \neq \emptyset$). Then $\forall S' \subseteq S$ ($S' \neq \emptyset$) and $F' \subseteq F$ ($F' \neq \emptyset$), we obtain that $J^\star_{S\times F} \leq J^\star_{S'\times F'}$, since $d^\star_{S'\times F'}$ can be seen as a special case of $\theta_{S\times F}$. The subscripts $S \times F$ and $S' \times F'$ denote the kernels and features in use.

To be more specific, we conclude that MKL-based SVM with multiple kernels and features perform better or at least equally than those with multiple kernels and single feature or with single kernel and multiple features.

Appendix B

Proof of the superiority of our proposed MKL-based multiclass-SVM over the SimpleMKL-based multiclass-SVM

To be consistent with the SimpleMKL-based multiclass-SVM, we set $p = 1$ in our framework. Then the only difference between the two methods are the ways of updating the kernel combination vectors for multi-class classification tasks as mentioned in Equation 4.3.1 and Equation 4.3.2. The superiority of our proposed MKL framework for multiclass-SVM lies in the fact that its minimized objective function preserves the lower boundary of the one obtained using SimpleMKL-based multiclass-SVM. That is, the derived hyperplanes from our method performs better or at least equally among the training data.

Suppose that $\hat{L}^\star$ is the optimal value of the objective function in Equation 4.3.2, and $\theta_u^\star$ is the learned optimum for each binary classifier in our framework. $L^\star$ and $\theta^\star$

are the corresponding notations for the SimpleMKL-based multiclass-SVM in Equation 4.3.1. Our proof is as follows,

$$\hat{L}^\star = \sum_{u \in \Phi} L_u(\theta_u^\star) \leq \sum_{u \in \Phi} L_u(\theta^\star) = L^\star$$

since $L_u(\theta_u^\star) \leq L_u(\theta^\star), \forall u \in \Phi$.

Bibliography

[1] A. Mehrabian and M. Wiener, "Decoding of inconsistent communications." *Journal of Personality and Social Psychology*, vol. 6, no. 1, pp. 109–114, 1967.

[2] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Palo Alto: Consulting Psychologists Press, 1978.

[3] R. Cornelius, "Theoretical approaches to emotion," in *ISCA Tutorial and Research Workshop (ITRW) on Speech and Emotion*, 2000.

[4] B. Fasel and J. Luettin, "Automatic facial expression analysis: a survey," *Pattern Recognition*, vol. 36, no. 1, pp. 259–275, 2003.

[5] Z. Zeng, M. Pantic, G. Roisman, and T. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 1, pp. 39–58, 2009.

[6] P. Ekman, W. Friesen, and J. Hager, *Facial Action Coding System (FACS): Manual.* Salt Lake City (USA): A Human Face, 2002.

[7] P. Ekman, R. J. Davidson, W. V. Friesen *et al.*, "The duchenne smile: Emotional expression and brain physiology ii," *Journal of personality and social psychology*, vol. 58, no. 2, pp. 342–353, 1990.

[8] M. Pantic and L. J. Rothkrantz, "Facial action recognition for facial expression analysis from static face images," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 34, no. 3, pp. 1449–1461, 2004.

[9] J. J. Bazzo and M. V. Lamar, "Recognizing facial actions using gabor wavelets with neutral face average difference," in *FG*, 2004, pp. 505–510.

[10] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan, "Recognizing facial expression: machine learning and application to spontaneous behavior," in *CVPR*, 2005, pp. 568–573.

[11] T. Simon, M. H. Nguyen, F. De La Torre, and J. F. Cohn, "Action unit detection with segment-based svms," in *CVPR*, 2010, pp. 2737–2744.

[12] S. Koelstra, M. Pantic, and I. Patras, "A dynamic texture-based approach to recognition of facial actions and their temporal models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 11, pp. 1940–1954, 2010.

[13] M. F. Valstar and M. Pantic, "Fully automatic recognition of the temporal phases of facial actions," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 1, pp. 28–43, 2012.

[14] X. Ding, W.-S. Chu, F. D. L. Torre, J. F. Cohn, and Q. Wang, "Facial action unit event detection by cascade of tasks," in *ICCV*, 2013, pp. 2400–2407.

[15] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[16] A. Rakotomamonjy, F. Bach, S. Canu, Y. Grandvalet *et al.*, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[17] M. Valstar, B. Jiang, M. Mehu, M. Pantic, and K. Scherer, “The first facial expression recognition and analysis challenge,” in *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, march 2011, pp. 921 –926.

[18] C. Shan, S. Gong, and P. McOwan, “Facial expression recognition based on local binary patterns: A comprehensive study,” *Image and Vision Computing*, vol. 27, no. 6, pp. 803–816, 2009.

[19] Y. Zhu, F. De la Torre, J. Cohn, and Y. Zhang, “Dynamic cascades with bidirectional bootstrapping for spontaneous facial action unit detection,” in *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, 2009, pp. 1–8.

[20] 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.

[21] T. Cootes, C. Taylor, D. Cooper, J. Graham *et al.*, “Active shape models-their training and application,” *Computer vision and image understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[22] M. Pantic and I. Patras, “Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 2, pp. 433–449, 2006.

[23] T. Cootes, G. Edwards, and C. Taylor, “Active appearance models,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 6, pp. 681–685, 2001.

[24] J. Sung and D. Kim, “Pose-robust facial expression recognition using view-based 2d + 3d aam,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 4, pp. 852–866, 2008.

[25] Y. Cheon and D. Kim, “Natural facial expression recognition using differential-aam and manifold learning,” *Pattern Recognition*, vol. 42, no. 7, pp. 1340–1350, 2009.

[26] M. Lyons, J. Budynek, and S. Akamatsu, “Automatic classification of single facial images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 12, pp. 1357–1362, 1999.

[27] T. Wu, M. Bartlett, and J. Movellan, “Facial expression recognition using gabor motion energy filters,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, pp. 42–47.

[28] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” *Computer Vision-ECCV 2004*, pp. 469–481, 2004.

[29] S. Liao, W. Fan, A. Chung, and D. Yeung, “Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features,” in *Image Processing, 2006 IEEE International Conference on*, 2006, pp. 665–668.

[30] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, 2005, pp. 886–893.

[31] X. Wang, T. Han, and S. Yan, “An hog-lbp human detector with partial occlusion handling,” in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 32–39.

[32] Z. Li, J. Imai, and M. Kaneko, "Facial-component-based bag of words and phog descriptor for facial expression recognition," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 1353–1358.

[33] M. Dahmane and J. Meunier, "Emotion recognition using dynamic grid-based hog features," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, 2011, pp. 884–888.

[34] N. Sebe, M. Lew, Y. Sun, I. Cohen, T. Gevers, and T. Huang, "Authentic facial expression analysis," *Image and Vision Computing*, vol. 25, no. 12, pp. 1856–1863, 2007.

[35] Y. Yacoob and L. Davis, "Recognizing human facial expressions from long image sequences using optical flow," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 6, pp. 636–642, 1996.

[36] I. Essa and A. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 757–763, 1997.

[37] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang, "Facial expression recognition from video sequences: temporal and static modeling," *Computer Vision and Image Understanding*, vol. 91, no. 1, pp. 160–187, 2003.

[38] M. Yeasin, B. Bullot, and R. Sharma, "Recognition of facial expressions and measurement of levels of interest from video," *Multimedia, IEEE Transactions on*, vol. 8, no. 3, pp. 500–508, 2006.

[39] Y. Zhang and Q. Ji, "Active and dynamic information fusion for facial expression understanding from image sequences," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 5, pp. 699–714, 2005.

[40] C. Shan, S. Gong, and P. McOwan, "Dynamic facial expression recognition using a bayesian temporal manifold model," in *Proc. BMVC*, vol. 1, 2006, pp. 297–306.

[41] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[42] S. Fu, X. Kuai, and G. Yang, "Multiple kernel active learning for facial expression analysis," *Advances in Neural Networks–ISNN 2011*, pp. 381–387, 2011.

[43] T. Sénéchal, V. Rapp, H. Salam, R. Seguier, K. Bailly, and L. Prevost, "Facial action recognition combining heterogeneous features via multikernel learning," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 993–1005, 2012.

[44] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, 2005, pp. 786–791.

[45] M. F. Valstar and M. Pantic, "Combined support vector machines and hidden markov models for modeling facial action temporal dynamics," in *ICHCI*, 2007, pp. 118–127.

[46] Y. Tong, W. Liao, and Q. Ji, "Facial action unit recognition by exploiting their dynamic and semantic relationships," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 10, pp. 1683–1699, 2007.

[47] Y. Li, J. Chen, Y. Zhao, and Q. Ji, "Data-free prior model for facial action unit recognition," *Affective Computing, IEEE Transactions on*, vol. 4, no. 2, pp. 127–141, 2013.

[48] Y. Li, S. Wang, Y. Zhao, and Q. Ji, “Simultaneous facial feature tracking and facial expression recognition,” *Image Processing, IEEE Transactions on*, vol. 22, no. 7, pp. 2559–2573, 2013.

[49] Y. Tong, J. Chen, and Q. Ji, “A unified probabilistic framework for spontaneous facial action modeling and understanding,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 2, pp. 258–273, 2010.

[50] Z. Wang, Y. Li, S. Wang, and Q. Ji, “Capturing global semantic relationships for facial action unit recognition,” in *ICCV*, 2013, pp. 3304–3311.

[51] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan, “Learning the kernel matrix with semidefinite programming,” *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[52] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, “Large scale multiple kernel learning,” *The Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[53] C. Cortes, M. Mohri, and A. Rostamizadeh, “l_2 regularization for learning kernels,” in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 109–116.

[54] T. Sun, L. Jiao, F. Liu, S. Wang, and J. Feng, “Selective multiple kernel learning for classification with ensemble strategy,” *Pattern Recognition*, vol. 46, no. 11, pp. 3081–3090, 2013.

[55] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, “Lp-norm multiple kernel learning,” *Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.

[56] M. Kloft, “Lp-norm multiple kernel learning,” Ph.D. dissertation, Berlin Institute of Technology, 2011.

[57] F. Yan, K. Mikolajczyk, J. Kittler, and M. Tahir, “A comparison of l_1 norm and l_2 norm multiple kernel svms in image and video classification,” in *Content-Based Multimedia Indexing, 2009. CBMI’09. Seventh International Workshop on*, 2009, pp. 7–12.

[58] K. Crammer and Y. Singer, “On the algorithmic implementation of multiclass kernel-based vector machines,” *The Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2002.

[59] T. Evgeniou and M. Pontil, “Regularized multi–task learning,” in *KDD*, 2004, pp. 109–117.

[60] T. Evgeniou, C. A. Micchelli, and M. Pontil, “Learning multiple tasks with kernel methods,” *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[61] A. Argyriou, T. Evgeniou, and M. Pontil, “Convex multi-task feature learning,” *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[62] R. K. Ando and T. Zhang, “A framework for learning predictive structures from multiple tasks and unlabeled data,” *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[63] C. Widmer, N. Toussaint, Y. Altun, and G. Rätsch, “Inferring latent task structure for multitask learning by multiple kernel learning,” *BMC bioinformatics*, vol. 11, no. Suppl 8, p. S5, 2010.

[64] P. Jawanpuria and J. S. Nath, “Multi-task multiple kernel learning,” in *SDM*, 2011, pp. 828–838.

[65] S. Bucak, R. Jin, and A. K. Jain, “Multi-label multiple kernel learning by stochastic approximation: Application to visual object recognition,” in *NIPS*, 2010, pp. 325–333.

[66] L. Tang, J. Chen, and J. Ye, “On multiple kernel learning with multiple labels,” in *IJCAI*, 2009, pp. 1255–1260.

[67] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, 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, pp. 94–101.

[68] M. Pantic, M. Valstar, R. Rademaker, and L. Maat, “Web-based database for facial expression analysis,” in *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 2005, pp. 5–8.

[69] M. Valstar and M. Pantic, “Induced disgust, happiness and surprise: an addition to the mmi facial expression database,” in *The Workshop Programme*, 2010, pp. 65–70.

[70] M. F. Valstar, M. Mehu, B. Jiang, M. Pantic, and K. Scherer, “Meta-analysis of the first facial expression recognition challenge,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 966–979, 2012.

[71] S. Mavadati, M. Mahoor, K. Bartlett, P. Trinh, and J. Cohn, “Disfa: A spontaneous facial action intensity database,” *Affective Computing, IEEE Transactions on*, vol. 4, no. 2, pp. 151–160, 2013.

[72] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[73] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[74] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[75] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 532–539.

[76] S. Mavadati, M. Mahoor, K. Bartlett, and P. Trinh, "Automatic detection of non-posed facial action units," in *ICIP*, 2012, pp. 1817–1820.

[77] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.

[78] O. Chapelle and A. Rakotomamonjy, "Second order optimization of kernel parameters," in *Proc. of the NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels*, 2008.

[79] D. Luenberger and Y. Ye, *Linear and Nonlinear Programming.* Springer, 2008.

[80] J. Nocedal and S. J. Wright, *Numerical Optimization.* Springer, 2000.

[81] F. R. Bach, "Consistency of the group lasso and multiple kernel learning," *The Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[82] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy, “Svm and kernel methods matlab toolbox,” Perception Systémes et Information, INSA de Rouen, Rouen, France, 2005.

[83] O. Chapelle, P. Haffner, and V. Vapnik, “Support vector machines for histogram-based image classification,” *Neural Networks, IEEE Transactions on*, vol. 10, no. 5, pp. 1055–1064, sep 1999.

[84] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* Cambridge, MA, USA: MIT Press, 2001.

[85] P. Gehler and S. Nowozin, “On feature combination for multiclass object classification,” in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 221–228.

[86] K. Roy and M. Kamel, “Facial expression recognition using game theory,” *Artificial Neural Networks in Pattern Recognition*, pp. 139–150, 2012.

[87] S. Jain, C. Hu, and J. Aggarwal, “Facial expression recognition with temporal modeling of shapes,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, 2011, pp. 1642–1649.

[88] A. Ramirez Rivera, J. Rojas Castillo, and O. Chae, “Local directional number pattern for face analysis: Face and expression recognition,” *Image Processing, IEEE Transactions on*, vol. 22, no. 5, pp. 1740–1752, May 2013.

[89] W. Gu, C. Xiang, Y. Venkatesh, D. Huang, and H. Lin, “Facial expression recognition using radial encoding of local gabor features and classifier synthesis,” *Pattern Recognition*, vol. 45, no. 1, pp. 80–91, 2012.

[90] Y. Guo, G. Zhao, and M. Pietikäinen, "Dynamic facial expression recognition using longitudinal facial expression atlases," *Computer Vision–ECCV 2012*, pp. 631–644, 2012.

[91] A. Sánchez, J. V. Ruiz, A. B. Moreno, A. S. Montemayor, J. Hernández, and J. J. Pantrigo, "Differential optical flow applied to automatic facial expression recognition," *Neurocomputing*, vol. 74, no. 8, pp. 1272–1282, 2011.

[92] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind, and J. Movellan, "Dynamics of facial expression extracted automatically from video," *Image and Vision Computing*, vol. 24, no. 6, pp. 615–625, 2006.

[93] U. Tariq, K.-H. Lin, Z. Li, X. Zhou, Z. Wang, V. Le, T. S. Huang, X. Lv, and T. X. Han, "Emotion recognition from an ensemble of features," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, 2011, pp. 872–877.

[94] S. Yang and B. Bhanu, "Facial expression recognition using emotion avatar image," in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, 2011, pp. 866–871.

[95] T. R. Almaev and M. F. Valstar, "Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition," in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, 2013, pp. 356–361.

[96] G. Zhao and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 915–928, 2007.

[97] M. Gönen and E. Alpaydin, “Localized multiple kernel learning,” in *ICML*, 2008, pp. 352–359.

[98] M. Sokolova, N. Japkowicz, and S. Szpakowicz, “Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation,” in *AI*, 2006, pp. 1015–1021.

[99] Y. Zhu, S. Wang, L. Yue, and Q. Ji, “Multiple-facial action unit recognition by shared feature learning and semantic relation modeling,” in *Pattern Recognition (ICPR), 2014 22nd International Conference on.* IEEE, 2014, pp. 1663–1668.

[100] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, “Large margin methods for structured and interdependent output variables,” in *Journal of Machine Learning Research*, 2005, pp. 1453–1484.

[101] C. A. Micchelli and M. Pontil, “Learning the kernel function via regularization,” *The Journal of Machine Learning Research*, vol. 6, pp. 1099–1125, 2005.