# Learning Active Facial Patches for Expression Analysis

Lin Zhong[†], Qingshan Liu[‡], Peng Yang[†], Bo Liu[†], Junzhou Huang[§], Dimitris N. Metaxas[†]
[†]Department of Computer Science, Rutgers University, Piscataway, NJ, 08854
[‡]Nanjing University of Information Science and Technology, Nanjing, 210044, China
[§]Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, 76019
{linzhong,qsliu,peyang,lb507,dnm}@cs.rutgers.edu, Jzhuang@uta.edu

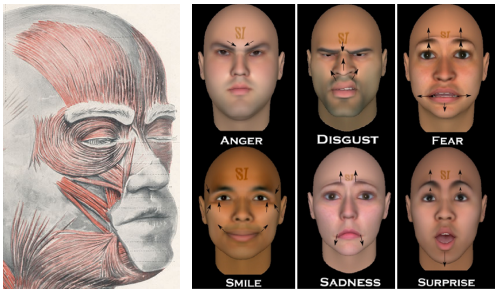
## Abstract

*In this paper, we present a new idea to analyze facial expression by exploring some common and specific information among different expressions. Inspired by the observation that only a few facial parts are active in expression disclosure (e.g. around mouth, eye), we try to discover the common and specific patches which are important to discriminate all the expressions and only a particular expression, respectively. A two-stage multi-task sparse learning (MTSL) framework is proposed to efficiently locate those discriminative patches. In the first stage MTSL, expression recognition tasks, each of which aims to find dominant patches for each expression, are combined to located common patches. Second, two related tasks, facial expression recognition and face verification tasks, are coupled to learn specific facial patches for individual expression. Extensive experiments validate the existence and significance of common and specific patches. Utilizing these learned patches, we achieve superior performances on expression recognition compared to the state-of-the-arts.*

## 1. Introduction

Facial expressions play significant roles in our daily communication. Recognizing these expressions has extensive applications, such as human-computer interface, multimedia, and security [21, 15, 23]. However, as the basis of expression recognition, the exploration of the underline functional facial features is still an open problem.

Studies in psychology show that facial features of expressions are located around *mouth, nose*, and *eyes*, and their locations are essential for explaining and categorizing facial expressions. Through electrical muscle stimulation, Duchenne [7, 1] found that most expressions are invoked by a small number of facial muscles around the mouth, nose and eyes (See Figure 1(a)). This indicates that most of the descriptive regions for each expression are located around certain face parts. Moreover, expressions can be forcedly categorized into six popular *"basic expressions"* [10]: anger, disgust, fear, happiness, sadness and surprise. As shown in Figure 1(b), each of these basic expressions can be further decomposed into a set of several related action units (AUs) [8], e.g. happiness can be decomposed to cheek raiser and lip corner puller. However, non-existing methods statistically utilize these prior knowledge about facial muscle and AUs to aid facial expression analysis in computer vision.




(a) (b)

Figure 1. (a) Illustration of facial muscles distribution [7]. (b) Major AUs for six expressions. The arrows represent for AUs.

Previous expression recognition methods can be generally categorized into two groups: *AU-based* methods and *appearance-based* methods. **AU-based** methods [18, 19] recognize expressions by detecting AUs, but all of them suffer from the difficulties of AU detection. **Appearance-based** methods[13, 25, 16] reveal the differences among expressions by facial appearance variations, which has been proved to be more reliable on single images. However, these methods assign weights to different face parts empirically, thus it lacks statistical support for the weight settings. This motivates us to fully make use of the prior knowledge from facial muscles and AU studies to extract the most discriminative regions, which can further assist expression analysis.

Inspired by the locations of AUs, we divide human face into non-overlapping patches and then conceptually group these patches into three categories: *common facial patches*,

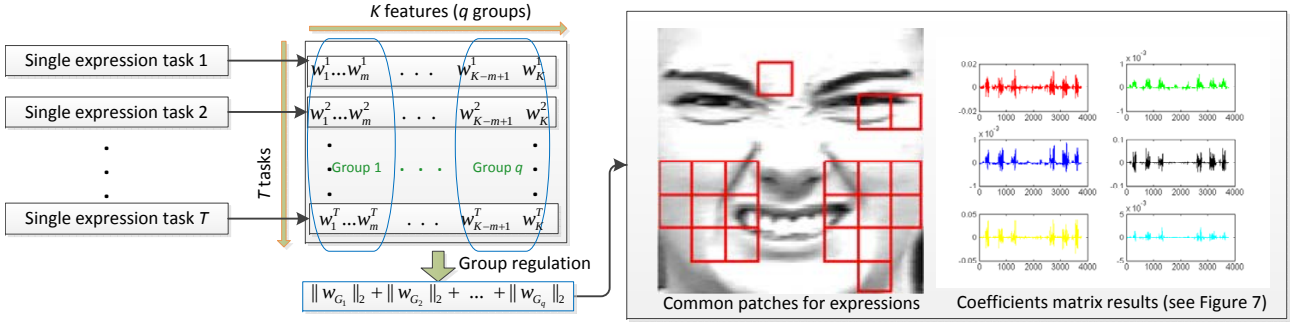



Figure 2. Discovering the common patches across six expressions using multi-task sparse learning (MTSL). Each single expression task is the binary classification task for one expression (See Figure 4). Expression tasks are combined in a MTSL model to select out the common patches under the group sparsity constraint.

*specific facial patches*, and the *rest*. *Common facial patches* are active ones for all expressions. *Specific facial patches* are only active for one particular expression. Therefore, the most important facial patches are the common ones shared by all expressions; specific patches are only a few and only useful to discriminate a particular expression; the rest of the patches are of less help to expression recognition.

A two-stage multi-task sparse learning framework is proposed to explore common and specific patches statistically. In the first stage, the binary classification problem for each expression are treated as an individual task (see Figure 4), then a multi-task sparse learning (MTSL) model is built based on these related tasks to extract the common facial patches. In the second stage, the face verification task (see Figure 5) is designed to be coupled with the previous classification task for one expression. In this way, another MTSL model can be constructed to find out the specific patches for this particular expression. Similarly, the specific patches for all the expressions can be figured out separately.

The common and specific patches, found by extensive experiments on the Cohn-Kanade database [11] and the MMI database [20], not only confirm the psychology discoveries of the facial muscles and AUs, but also provide more accurate appearance locations. Moreover, these patches can be used to boost the performance of expression recognition. Only using extremely small number of patches ($\sim 1/3$ of the face), our method still outperforms other methods in expression recognition.

**Our contributions are**: 1) As far as we know, we are the first to provide a solid validation for an important psychology discovery, that only a few facial muscles (areas) are discriminative for expression recognition. 2) A two-stage multi-task sparse learning framework is proposed to formulate the commonalities among expressions, and find out the locations of common and specific patches for expressions. 3) Extensive experiments on two public databases demonstrate that these active patches are effective in recognizing expressions, and can be utilized to further improve the performances of state-of-the-arts.

## 2. Related work

### 2.1. Facial expression analysis

Most appearance-based facial expression analysis methods follow the two main steps: facial representation and expression recognition.

**Facial representation** derives a set of features from original facial images to effectively represent all faces. Different features have been applied to either the whole-face or specific face regions to extract the facial appearance changes, such as Gabor [13, 4], haar-like features [25], local binary patterns (LBP) [14]. In Shan [16], facial images are equally divided into small regions, and then LBP features are extracted from these empirically weighted sub-regions to represent the facial appearance. The LBP features are shown to be effective in expression recognition, so our paper will also utilize the LBP features with the same sub-region division strategy. Different from their work, we will focus on learning the effective sub-regions statistically.

**Expression recognition** aims to correctly categorize different facial representations. Support Vector Machine (SVM) [4, 16, 27] is the most popular and effective learning method in facial expression recognition. Shan's work [16] is the most similar work with ours, so it will be considered as the baseline. For fair comparison, our paper will also employ SVM as the the classification algorithm.

### 2.2. Multi-task sparse learning

Multi-task learning is an inductive transfer machine learning approach. It aims to learn a problem together with some related problems for better performance [2, 5]. Multi-task sparse learning was then designed in [3] for feature selection, through encouraging multiple predictors from different tasks to share similar parameter sparsity patterns. Multi-task sparse learning also obtained a rewarding performance on handwritten character recognition in [9]. Yuan [28] developed a visual classification algorithm by learning the shared parts among different representation tasks. Recently, Chen [6] provided a faster solution to

multi-task sparse learning problems.

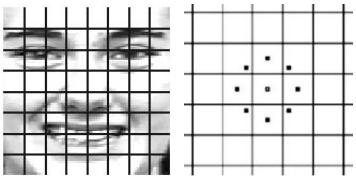
Suppose there are $T$ related tasks, and $(x_i^t, y_i^t), i = 1, 2, ..., N_t$ is the training set of task $t$, where each sample is represented by $K$-dimensional features, $x_i^t \in R^K$, and $y_i^t \in \{-1, 1\}$ indexes $x_i^t$ is positive or not. $w^t$ is a $K$-dimensional vector of representation coefficients for task $t$. All the $w^t$s are the rows of the matrix $W = [w_k^t]_{t,k}$, while every column of the matrix $W$ is a $T$-dimensional vector that means the representation coefficients from the $k$-th feature across different tasks, $w_k = [w_k^1, w_k^2, ..., w_k^T]'$. Multi-task sparse learning aims to learn the shared sparse information among all the tasks. The formulation with $L_1/L_2$ mixed-norm regularization is as follows:

$$\arg\min_W \sum_{t=1}^{T} \frac{1}{N_t} \sum_{i=1}^{N_t} J^t(w^t, x_i^t, y_i^t) + \lambda \sum_{k=1}^{K} \|w_k\|_2 \quad (1)$$

where $J^t(w^t, x_i^t, y_i^t)$ is the cost function of the $t$th task, $\lambda$ is a constant to balance the sparsity, and $\sum$ is the mathematic format for $L_1$ norm. The regularization term encourages most columns of matrix $W$ to be zero, and the remaining non-zero columns indicate the corresponding features are shared features across all the tasks.

## 3. Proposed Work

Facial expressions are usually manifested by local facial appearance variations. However, it is not easy to automatically localize these local active areas on a facial image. A facial image is divided into $p$ local patches, and then local binary pattern (LBP) features are used to represent the local appearance of the patch. These features have been proven to be a powerful descriptor in expression recognition [16] and face verification [24]. We set $p = 64$ in the experiments with the image size of $96 \times 96$, as shown in Figure 3(a). For each patch, the uniform LBP features are extracted with the LBP operator $LBP_{8,1}$, as shown in Figure 3(b), and mapped to a $m$-dimensional histogram ($m = 59$ in our paper).



(a) (b)

Figure 3. (a)A cropped facial image is divided into 64 patches. (b) LBP feature ($P = 8, R = 1$)

Based on these local patches, the common patches across all expressions are learned for expression recognition. Then, some specific patches for each expression are explored to further enhance the performance.

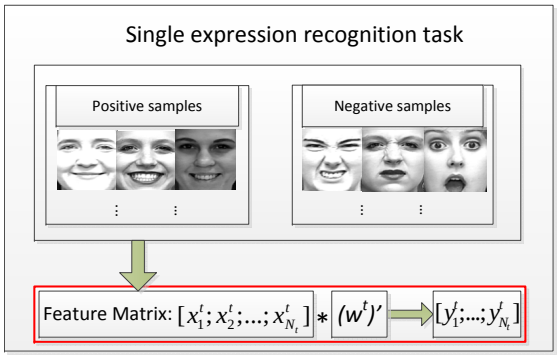



Figure 4. Illustration of one single expression task. Each task is a binary expression classification problem. Take Expression task of happiness for example here.

### 3.1. Learning Common Patches Across Expressions

Discovering the common patches across all the expressions is actually equivalent to learning the shared discriminative patches for all the expressions. Since Multi-task sparse learning (MTSL) can learn common representations among multiple related tasks [3], our problem can be transfered into a MTSL problem. $T$ related tasks are defined as $T$ discriminative patch learners for $T$ facial expressions respectively (we set $T = 6$ for six basic expressions). Supposing each image has $p$ patches, it can be represented by $(p \times m)$-dimensional LBP-based histogram features. Let $K = p \times m$. However, equation (1) cannot directly model our problem, because different from the MTSL model described in Equation(1), we focus on the selection of common patches instead of individual features. Since a group of consecutive features stand for one patch, and the number of common patches are not large, group sparsity prior can be assumed [29, 30]. Our problem is modeled as the following MTSL problem, in which the regularization term of Equation(1) is modified to a patch level sparse constrain:

$$\arg\min_W \sum_{t=1}^{T} \frac{1}{N_t} \sum_{i=1}^{N_t} J^t(w^t, x_i^t, y_i^t) + \lambda \sum_{j=1}^{p} \|w_{G_j}\|_2 \quad (2)$$

Here, $w_{G_j}$ is a sub-matrix of matrix $W$, where $G_j$ denotes the $j$-th patch, as shown in Figure 2. Figure 4 illustrate how to set up each task. In each task, images of one particular expression are considered as positive samples, while others are negative samples. This regularization term encourages the representation coefficients of the features in most patches to be zero, and then the remaining non-zero patches indicate the shared important representation for all the expressions. The cost function of $J^t$ is defined as a logistic lost function:

$$J^t(w^t, x_i^t, y_i^t) = ln(1 + exp(-y_i^t x_i^t \cdot w^t)). \quad (3)$$

To solve this patch-based multi-task sparse learning, based on the accelerated gradient method proposed in [26], we

heuristically make the representation coefficients of the features in one patch to be zeros or non-zeros simultaneously by step 5-9 in Algorithm 1. The detailed problem solving procedure are summarized in Algorithm 1.

**Algorithm 1** *Algorithm for learning common patches*

**Input** : Training data $\{(x_i^t, y_i^t), i = 1, .., N_t\}$, define $X^t = [x_1^t; ...; x_{N_t}^t], Y^t = [y_1^t; ...; y_{N_t}^t], V = [v^1; ...; v^T]$. $t$ indicates the task index, and $t = 1, ..., T$. $j$ is the group index, and $j = 1, ..., p$.

**Initialize** : $W_0$ takes equal weights, $V_0 = W_0$ and $a_0 = 1$. Tuning parameter $\lambda$ and step size $\eta$.

**for** $s = 0...S$ **do**

  $w_{s+1}^t = v_s^t - \eta[\frac{1}{1+exp(-(Y^t)^{'} X^t v_s^t)} exp(-(Y^t)^{'} X^t v_s^t)(-(X^t)^{'} Y^t)]$

  **if** $\|w_{G_j,s+1}\|_2 \geq \lambda\eta$ **then**

    Set $w_{G_j,s+1} = (1 - \frac{\lambda\eta}{\|w_{G_j,s+1}\|_2}) w_{G_j,s+1}$

  **else**

    Set $w_{G_j,s+1} = 0$

  **end if**

  $a_{s+1} = \frac{2}{s+3}, \delta_{s+1} = W_{s+1} - W_s$

  $V_{s+1} = W_{s+1} + \frac{1-a_s}{a_s} a_{s+1} \delta_{s+1}$

  **if** $\|\delta_{s+1}\|_2 \leq \epsilon$ **then**

    break

  **end if**

**end for**

**Normalization** : $w^t = \frac{w^t}{\|w^t\|_2}$

$w_{G_j} = \sum w_k^t$, where $w_{G_j}$ is the weight for patch $j$, and $w_k^t \in G_j$

**Output** : order $w_{G_j}$ decreasingly, and output the top patches as the common patches for all expressions.

### 3.2. Learning Specific Patches For Individual Expression

Although learned common patches can discriminate all facial expressions, the performance could not be the best, because each expression also has its special properties besides the common properties. Here, we aim to explore some specific facial patches for each expression with the help of face verification, and then they are used to further boost the performance of common facial patches. The motivation to employ the face verification task is that those special facial patches are important face regions, which are not only useful for recognizing this expression, but also very significant for identifying the subjects. Take an expression $e$ for example. If recognizing $e$ is a task and face verification is another related task, a multi-task sparse learning model can be used to learn the shared patches between these two tasks. The learned patches should embed some specific signatures of the face identity. This multi-task sparse learning model is the same as the model of learning the common patches except the different task design. The individual expression analysis task is organized in the same way as in Figure 4. Figure 5 illustrates how to organize the task of face verification. For face verification, we need to compare two images and label them as the same person or not, so we organize the training data of this task by the feature difference between two images. Assuming $(x_i^2, y_i^2)_{i=1}^{N_2}$ is the training set, $x_i^2$ is the feature difference between two images in $i$-th image pair. $y_i^2 \in \{-1, 1\}$ indicates whether the two images in $i$-th pair come from one subject or not. $N_2$ is the number of image pairs. The superscript 2 means this task is the second task in the multi-task sparse learning model. The procedure for solving this problem is the same with Algorithm 1. Because there are six expressions, six multi-task sparse learning models are needed to built to learn their specific patches respectively.

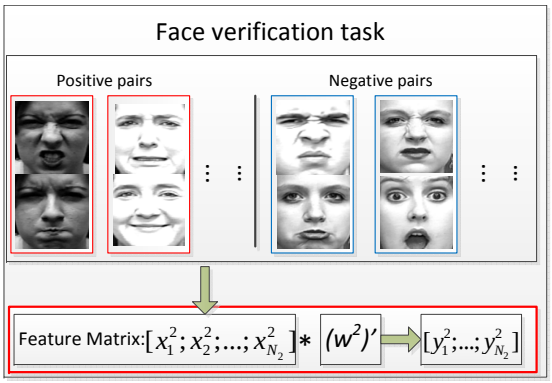



Figure 5. The design of Face Verification task. Image pairs from the same subject are considered as positive samples. Otherwise, negative samples.

The specific patches have overlap with the learned common patches. Since the common patches will be used for all expressions, the overlapped patches are removed from the specific patches. The rest patches are considered as the final specific patches.

### 3.3. Classifier Design

With the extracted common and specific patch features based on the training data, classifiers are then built based on these features for testing data. Multi-task sparse learning model can directly give out classification results [28]. However, to fairly compare with previous work [12, 16], SVM is adopted to learn the expression classifiers and the one-against-all strategy is employed to decompose the six class problem into multiple binary classes problem. The performances of common patches and the combination of common and specific patches are evaluated respectively. For expression $e$, denotes the common patches as $P_c$, and the specific patches as $\{P_s^e\}_{e=1}^6$. When both common and specific patches are investigated, the features from $P_c$ and $P_s^e$

are concatenated to represent facial images, and train the SVM classifiers; While only use the features of $P_c$ when common patches are tested.

## 4. Experiments

We evaluate the learned common and specific patches for facial expression recognition. All methods are compared on two datasets, the Cohn-Kanade database [11] and the MMI database [20], which are widely used for facial expression recognition algorithms. Our methods are denoted as CPL and CSPL respectively (see Table 1). To efficiently evaluate the performance of our proposed methods, they are compared with [16], which is the most recent comprehensive study on expression recognition with remarkable results. In [16], two methods are evaluated, denoted as ADL and AFL respectively. ADL uses Adaboost to select important patches and then perform SVM on the extracted LBP features of these patches. AFL uses all the patches to train the classifier without feature selection. For fair comparison, all the methods are based on the same patch(sub-region) division strategy, same feature representation, and the same classification method (SVM). The only difference for the methods are the patches they use. All method abbreviations are listed in table 1. 10 folds cross-validation is employed for all methods.

Table 1. Method abbreviations.

| | |
|---|---|
| CPL | only use **C**ommon **P**atches. (our method) |
| CSPL | use **C**ommon and **S**pecific Patches. (our method) |
| ADL | only use patches selected by **AD**aboost are used. |
| AFL | **A**ll patches of the whole **F**ace are used. |

### 4.1. Experiments On the Cohn-Kanade Database

The Cohn-Kanade database consists of 100 university students aged from 18 to 30 years old, of which 65% were female, 15% were African-American and 3% were Asian or Latino. Subjects were instructed to perform a series of 23 facial displays, six of which were based on description of prototypic emotions. For our experiments, image sequences are selected out from 96 subjects, whose sequences could be labeled as one of the six basic emotions. For each sequence, we only use the three peak frames with the most expressions. The faces are detected automatically by Viola's face detector [22], and then they are normalized to $96 \times 96$ as in Tian [17] based on the location of the eyes. Figure 6 shows some normalized samples with all expressions.

Figure 6. Example of six basic expressions from the Cohn-Kanade database.(Anger, Disgust, Fear, Happiness, Sadness and Surprise).

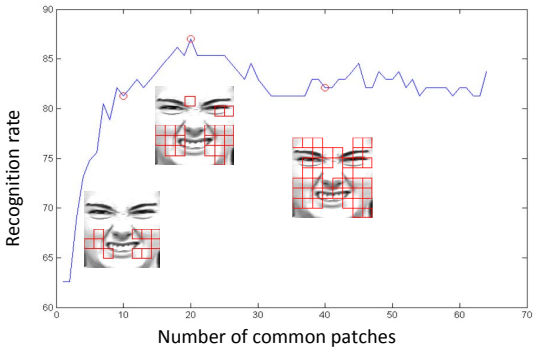



Figure 8. The expression recognition rate with different number of common patches. The patch number for the three faces images marked with selected common patches are 10, 20, 40 respectively.

#### 4.1.1 Analysis of Common Patches

As described in section 2.1, the proposed multi-task sparse learning aims to select the shared patches instead of the shared features, so we apply the $L_1/L_2$ norm regularization on the patch level to obtain patch-based group sparsity. Figure 7 reports the representation coefficient results for six expression tasks. We can see that the representation coefficients of features are not only sparse, but also show the property of patch-based group sparsity. It is also clear to see the index correspondences for non-zero values across six expressions, which indicates the commonalities among them. So, this result demonstrates the effectiveness of our proposed algorithm in learning the shared common patches for expressions.

Before evaluating the recognition performance of the common patches, we want to inspect the performance when a different number of common patches is selected. Figure 8 reports the results with different number of the common patches. We can see that the recognition rate increases quickly with the first leading common patches, and when the number of the selected patches reaches around 20, it will get a recognition rate of 88.42%. If too many common patches are selected, the performance goes down slightly and fluctuates. It means that only some common patches are discriminative for all the expressions. When some patches with little importance are selected as the common patches, they will introduce some noises and influence the discriminative power of the common patches. We set the number of the common patches to be 20 in the following experiments. Figure 9 shows the superimposing effect of the selected common patches over the 10 fold experiments. There are great overlaps between different fold experiments. It indicates that our algorithm is robust to the selection of the training set. The selected common patches are basically around the areas of mouth, eye, and eyebrows, which are consistent with AU-based analysis in FACS [8].

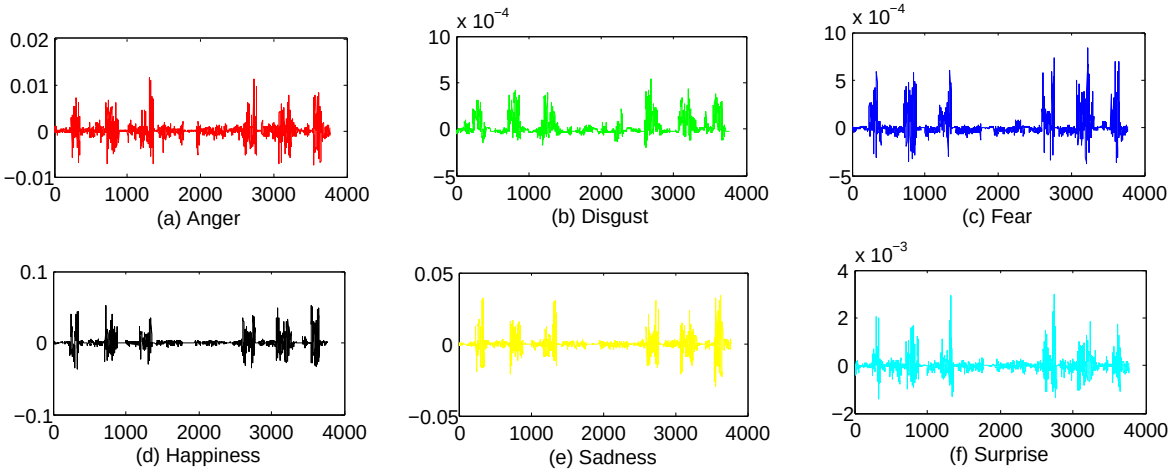



Figure 7. Results for six expressions in the coefficient matrix after multi-task sparse learning for learning the common patches. X-axis corresponds to the feature index in the coefficient matrix, where features index are ordered consecutively as group by patches. Y-axis is the weight value for features in each task after multi-task sparse learning. The non-zeros parts are grouped, and matches across all tasks.

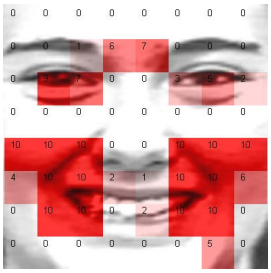


Figure 9. The distribution of selected common patches on faces. The darker the red color is, the more times (shown as numbers) the patch has been selected as common patches in 10-fold experiments.

Table 2 reports the detail recognition performance of the common patches on each expression, where the expressions of anger, disgust, fear, happiness, sadness, surprise are denoted as ag, dg, fa, hp, sd, and sp for simplicity. Promising recognition rates are obtained on all the expressions except anger. Anger is often misclassified as sadness. This is because these two expressions have similar appearance variations on the common patches. This problem can be alleviated by adding some specific patches, which will be discussed next.

#### 4.1.2 Analysis of Specific Patches

Although a rewarding recognition result can be obtained by only using the common patches, the performance can be further improved by integrating some specific patches of each expression. Figure 10 shows the top three learned specific patches for each expression based on the proposed multi-task learning. We can see the locations of these patches are highly related to expression types. Taking surprise for example. The selected specific patches show the characteristics of surprise expression, in which special appearance changes are distributed in opened mouth, on stared eye, and raised eyebrow. In CSPL, the common patches and the specific patches are integrated together, and the experimental results are reported in Table. 3. Compared to the results of CPL (Table. 2), we can see that adding specific patches can further improve the performances of the common patches.

Table 2. The confusion matrix of CPL on the Cohn-Kanade database.(Measured by recognition rate: %)

| | ag | dg | fa | hp | sd | sp |
|---|---|---|---|---|---|---|
| ag | **65.56** | 8.33 | 0 | 0 | 25.28 | 0.83 |
| dg | 2.67 | **92.67** | 0.67 | 2 | 2 | 0 |
| fa | 0 | 1.98 | **78.97** | 13.25 | 5.79 | 0 |
| hp | 0.33 | 0.67 | 4.24 | **94.76** | 0 | 0 |
| sd | 6.20 | 1.67 | 3.33 | 0 | **87.69** | 1.11 |
| sp | 0 | 0 | 1.25 | 0 | 0.48 | **98.27** |

Table 3. The confusion matrix of CSPL on the Cohn-Kanade database.(Measured by recognition rate: %)

| | ag | dg | fa | hp | sd | sp |
|---|---|---|---|---|---|---|
| ag | **71.3889** | 7.5 | 0 | 0.83 | 19.44 | 0.83 |
| dg | 2.67 | **95.33** | 0 | 0 | 2 | 0 |
| fa | 0 | 2.46 | **81.11** | 10 | 6.43 | 0 |
| hp | 0.33 | 0.33 | 3.58 | **95.42** | 0.33 | 0 |
| sd | 7.45 | 1.25 | 2.92 | 0 | **88.01** | 0.37 |
| sp | 0 | 0 | 1.25 | 0 | 0.48 | **98.27** |

#### 4.1.3 Experimental Comparisons

To further evaluate the proposed CPL and CSPL, we compare them to ADL and AFL developed in [16]. Table 4 lists

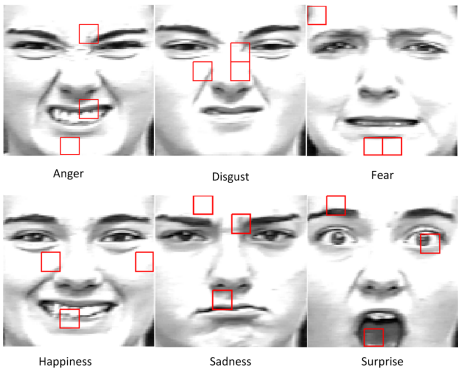



Figure 10. The top 3 specific patches for six expressions after eliminating the shared patches on the Cohn-Kanade database.

the recognition rates of these four methods. AFL gets the recognition rate of 86.94%, which is much worse than our methods. It shows the importance of selecting discriminative patches. Although ADL also uses Adaboost to select the patches, it does not take the commonalities among all the expressions into account. ADL gets a recognition rate of 82.26% with the selected patches (highest rate with $20\pm3$ patches), while the recognition rates of CPL and CSPL are 88.42% and 89.89% respectively. It demonstrates that the learned common and specific patches by our proposed two-stage multi-task sparse learning can really improve the performance of expression recognition. As far as we know, this is also the first time to validate the facial muscles and AUs on large real data.

Table 4. Recognition performances for CPL, CSPL, AFL, ADL on the Cohn-Kanade database.

| Methods | CPL | CSPL | AFL | ADL |
|---|---|---|---|---|
| Recognition Rate% | **88.42** | **89.89** | 86.94 | 82.26 |

## 4.2. Results on the MMI database

The MMI database includes 30 students and research staff members aged from 19 to 62, of whom 44% are female, having either a European, Asian, or South American ethnic background. In this database, 213 sequences have been labeled with six basic expressions, in which 205 sequences are with frontal face. Different from [16], in which only the experimental data are collected from 99 selected sequences, we conduct our experiments on the data from all the 205 sequences. As in [16], the apex images are extracted from the sequences as the experimental data. Facial image are corpped based on locations of eyes, and resize it to $96 \times 96$ too, same as on Cohn-Kanada database.

MMI is a more challenging database than the Cohn-Kanade database. First, the subjects make expressions non-uniformly. Different people make the same expression in different ways. Second, some subjects wear accessories, such as glasses, headcloth, or moustache. Additionally, in some sequences, the apex frames are not with high expression intensity. All these factors will greatly degrade the recognition performance.

Table 5. Recognition performances for CPL, CSPL, AFL, ADL on the MMI database.

| Methods | CPL | CSPL | AFL | ADL |
|---|---|---|---|---|
| Recognition rate % | **49.36** | **73.53** | 47.74 | 47.78 |

Table 6. The confusion matrix of CSPL on the MMI database.(Measured by recognition rate: %)

| | ag | dg | fa | hp | sd | sp |
|---|---|---|---|---|---|---|
| ag | **50.28** | 10.56 | 5.56 | 2.50 | 28.61 | 2.50 |
| dg | 5.50 | **79.83** | 3.50 | 2.17 | 9.00 | 0 |
| fa | 1.67 | 4.13 | **67.14** | 15.56 | 8.97 | 2.54 |
| hp | 2.63 | 0.67 | 12.82 | **82.91** | 0.67 | 0.30 |
| sd | 16.34 | 2.87 | 13.98 | 4.54 | **60.28** | 1.99 |
| sp | 0.42 | 0 | 4.94 | 0.83 | 5.30 | **88.51** |

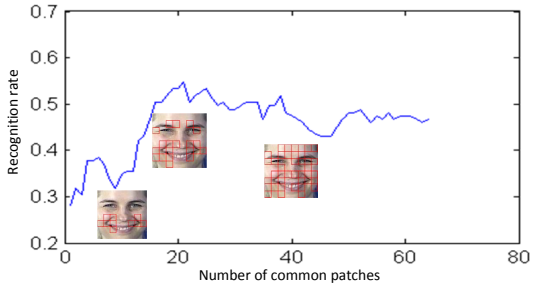



Figure 11. Recognition rate with different common patch number. Result of one fold experiment is shown.

We first investigate the performance of the common patches with different patch number, and Figure 11 shows the results. It can be seen that the results are similar to the results on the Cohn-Kanade database. About 20 common patches are discriminative for all the expressions, so we set the number of the common patches as 20 on this database too. Table 5 lists the recognition rates of CPL, CSPL, AFL, and ADL respectively. Same as on the Cohn-Kanade data, CPL and CSPL are superior to AFL and ADL. However, the performances of all four methods are much lower than the Cohn-Kanade database, because this database have several challenging factors as mentioned above. CSPL obtains much better performance than CPL. This is because each expression has a very big variance due to the diversity of the subjects in this database, but the common patches cannot describe these specific variations. Although a much better result of 86.7% is reported in [16], their experimental data are carefully chosen 99 sequences, while we perform the experiments on all the 205 sequences. Besides, they adopt sliding and multi-scale windows to extract much more patches. We only divide the facial image into 64 patches, and we also obtain a recognition rate of 73.53% on more than double size of the data than [16]. Table 5 lists the

confusion matrix of CSPL.

The experimental results indicate the location of learned common and specific patches, which confirms the previous knowledge about active facial parts in psychology. The rewarding performances of these patches in facial expression recognition provide a solid basis for patches selection and weight setting in similar applications. Our work opens the road for the researches of utilizing the prior knowledge of facial muscles in psychology, and further improve the performances of existing methods in computer vision.

## 5. Conclusions

In this paper, a new method to analyze facial expressions is proposed. Different from previous work, we aimed at exploring the commonalities among the expressions by discovering the common and specific patches. A two-stage sparse learning model is proposed to learn the locations of these patches based on the prior knowledge of facial muscles and AUs. The effectiveness of these patches are evaluated by facial expression recognition. Extensive experiments show that common patches can generally discriminate all the expressions, and the recognition performance can be further improved by integrating specific patches. The learned location information of these patches also confirms the discovery in psychology.

## References


[1] http://en.wikipedia.org/wiki/Facial_expression. 1

[2] Special issue on inductive transfer. *Machine Learning*, 28(1), 1997. 2

[3] A. Argyriou and T. Evgeniou. Multi-task feature learning. *NIPS*, 2007. 2, 3

[4] M. S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. *CVPR*, 2, 2005. 2

[5] R. Caruana. Multi-task learning. *Machine Learning*, 28:41–75, 1997. 2

[6] J. Chen, J. Liu, and J. Ye. Learning incoherent sparse and low-rank patterns from multiple tasks. *SIGKDD*, 2010. 2

[7] G. Duchenne. *Mecanisme de la Physionomie Humaine*. 1862. 1

[8] P. Ekman and W. V. Friesen. Facial action coding system. *Consulting Psychologists Press*, 1, 1978. 1, 5

[9] G.Obozinski. Multi-task feature selection. *Technical Report. Department of Statistics, UC Berkeley*, 2006. 2

[10] C. E. Izard. The face of emotion. *New York: Appleton-Century-Crofts*, 1, 1971. 1

[11] T. Kanade, J. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. *FG*, 2000. 2, 5

[12] G. Littlewort, M. S. Bartlett, J. S. I. Fasel, and J. Movellan. Dynamics of facial expression extracted automatically from video. *CVPR*, 2004. 4

[13] M. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *TPAMI*, 21(12):1357 –1362, 1999. 1, 2

[14] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971 –987, 2002. 2

[15] A. Ryan, J. F. Cohn, S. Lucey, J. Saragih, P. Lucey, F. D. la Torre, and A. Rossi. Automated facial expression recognition system. *International Carnahan Conference on Security Technology.*, pages 172 –177, 2009. 1

[16] C. Shan, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27:803–816, 2009. 1, 2, 3, 4, 5, 6, 7

[17] Y. Tian. Evaluation of face resolution for expression analysis. *CVPR*, jun. 2004. 5

[18] Y. Tian, T. Kanade, and J. F. Cohn. Recognizing upper face action units for facial expression analysis. *CVPR*, 2000. 1

[19] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *TPAMI*, 29(10):1683 –1699, 2007. 1

[20] M. F. Valstar and M. Pantic. Induced disgust, happiness and surprise: an addition to the mmi facial expression database. *LREC*, 2010. 2, 5

[21] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 31(1):1743 –1759, 2009. 1

[22] P. Viola and M. Jones. Robust real-time object detection. *IJCV*, 2001. 5

[23] E. Vural, M. Cetin, A. Ercil, G. Littlewort, M. Bartlett, and J. Movellan. Automated drowsiness detection for improved driver safety comprehensive databases for facial expression analysis. *Int. Conf. on Automative Technologies*, 1, 2008. 1

[24] X. Wang, C. Zhang, and Z. Zhang. Boosted multi-task learning for face verification with applications to web image and video search. *CVPR*, 2009. 3

[25] J. Whitehill and C. W. Omlin. Haar features for facs au recognition. *FG*, 2006. 1, 2

[26] C. Xi, P. Weike, J. T. Kwok, and J. G. Carbonell. Accelerated gradient method for multi-task sparse learning problem. *ICDM*, 2009. 3

[27] P. Yang, Q. Liu, and D. N. Metaxas. Exploring facial expressions with compositional features. *CVPR*, 2010. 2

[28] X. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. *CVPR*, 2010. 2, 4

[29] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. Metaxas. Automatic image annotation using group sparsity. *CVPR*, 2010. 3

[30] S. Zhang, J. Huang, H. Li, and D. Metaxas. Automatic image annotation and retrieval using group sparsity. *TSMC*, 2012. 3