# Exploring Facial Expressions with Compositional Features

Peng Yang          Qingshan Liu          Dimitris N. Metaxas

Computer Science Department, Rutgers University
110 Frelinghuysen Road, Piscataway, NJ 08854, USA
peyang@cs.rutgers.edu, qsliu@cs.rutgers.edu, dnm@cs.rutgers.edu

## Abstract

*Most previous work focuses on how to learn discriminating appearance features over all the face without considering the fact that each facial expression is physically composed of some relative action units (AU). However, the definition of AU is an ambiguous semantic description in* Facial Action Coding System *(FACS), so it makes accurate AU detection very difficult. In this paper, we adopt a scheme of compromise to avoid AU detection, and try to interpret facial expression by learning some compositional appearance features around AU areas. We first divided face image into local patches according to the locations of AUs, and then we extract local appearance features from each patch. A minimum error based optimization strategy is adopted to build compositional features based on local appearance features, and this process embedded into Boosting learning structure. Experiments on the Cohn-Kanada database show that the proposed method has a promising performance and the built compositional features are basically consistent to FACS.*

## 1. Introduction

As early as 1970s, facial expression analysis has attracted some attention in the community of psychology. Two typical pioneering works are: categorizing facial expression into six basic expressions (happiness, sadness, disgust, surprise, anger, and fear) proposed by Izard [7] and *Facial Action Coding System* (FACS) designed by Ekman and Friesen [5]. FACS is composed of comprehensive standards that decompose each expression into several relative action units (AUs). Although AU-based facial expression analysis is much more precise compared to the six emotions based analysis, the definitions of AUs are actually ambiguous semantic descriptions. Therefore, it is not easy to do accurate AU detection automatically. Thus, in the communities of computer vision and pattern recognition, most of automatic facial expression analysis work only focused on

identifying an input facial image or sequence as one of six basic emotions [13][24][8][3] [6][22][16][21][2][27]. Generally, the first step is to extract appearance features, such as, Gabor features[15][10], haar-like features[23], and local binary patterns(LBP)[14], and then some learning methods are adopted to select discriminant features over the whole face to build the classifiers, such as SVM [10][14], Adaboost [23][10], and Adaboost + SVM [10]. Although they obtained good performance in some cases, the features they considered or selected are short of physical interpretation. According to FACS [5], each expression has explicit and intuitive local appearance variations, which are corresponding to AUs. In other words, each expression should be physically represented by some features with special spatial information.

Moreover, it is rare that a single AU appears alone in an expression. Usually several AUs appear simultaneously to show a meaningful facial expression. For example, the happiness expression may involve AU6+AU12+AU25 (lips part); the surprise expression may involve AU1 (inner brow raiser)+AU2 (outer brow raiser)+AU5 (upper lid raiser) + AU25 + AU27 (mouth stretch); and the sadness expression may involve AU1+AU4 (brow lowerer)+ AU15+AU17 [5]. Driven by this observation, Yan[18] developed the method to do AUs recognition based on UAs' dynamic and semantic relationships. In [18], the relation among AU's, which comes from the phycologist scientists' observation, is used as the prior to build the structure of a Bayesian network, and then the learning strategy is applied to update the structure. Due to considering the co-occurrence of AUs, Yan got very good recognition results. However, as we know, FACS only gives semantic descriptions of AUs. For example, AU1 means inner brow raiser, and AU4 means eye brow lower. The definition of the level of the AU is much ambiguous, and it makes accurate AU detection much difficult in practices. Therefore, it is hard to do low level expression recognition through analyzing AUs in the practical system.

In order to handle the above issues, we propose a compromise scheme in this paper. Although FACS does not
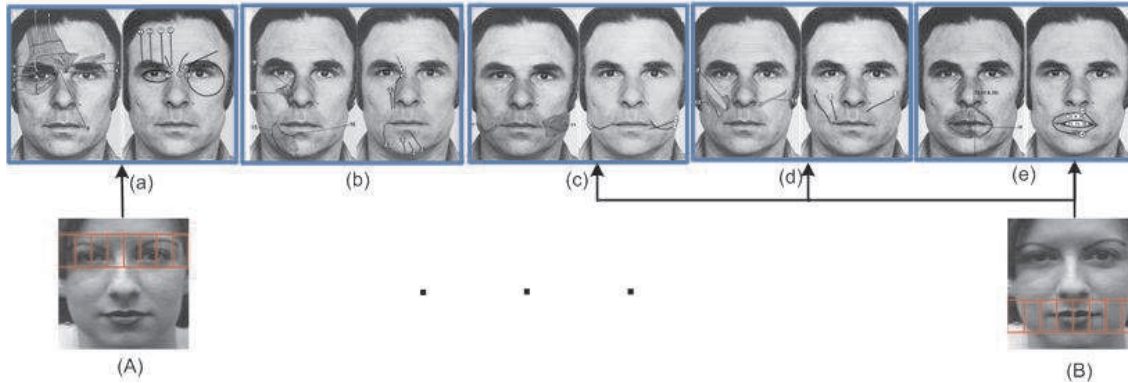
Figure 1. The distribution of AUs on the face is shown in first row, and some superimposed sub-windows on the face image. (a)Upper face action units;(b)Lower face action units(up/down Actions);(c)lower face action units(horizontal actions);(d)lower face action units( oblique actions);(e)lower face action units(orbital actions). Bottom:(A)The row of superimposed sub-windows along the eyes area;(B)one row of superimposed sub-windows at face bottom.

make a clear definition for AU's level, it points out where each AU occurs in the face. Based on the locations of AUs, we first divide the face into the overlap blocks to cover the location of all the possible AUs, and we extract local appearance features from each patch with haar-like descriptors [19]. Inspired by the observation that each expression is composed of several relative AUs, we try to build compositional features with constrains to simulate the combination of AUs. We develop an optimization method to build compositional features through minimizing the error. The procedure of compositional feature searching is integrated into Boosting framework to construct the final classifiers. The proposed method is tested on the Cohn-Kanada database, and experimental results demonstrate its efficiency and its consistency to FACS.

## 2. Local Patches and Feature Description

FACS defines 32 AUs, and each expression is composed of several AUs. Although it is hard to obtain accurate AU detection according its definition, the location it occurs is clear, so we divide the face image into blocks to cover almost all the AUs' location. Assuming the image with the size of $m, m$, we superimpose the sub-window (local patch) with the size of $(m/4, m/4)$, and use the step of $m/8$ in both $x$ and $y$ direction to obtain the local patches. In the experiments, the image size is $64 \times 64$, so we totally obtain 49 local patches. Figure1 shows the distribution of active units on the facial image, we can see the red rectangles in (A) almost covers upper AUs and the red ones in(B) covers the orbital AUs. Therefore, we can use the features within the patch to describe the potential information of AUs inside.

Feature representation plays an important role in facial expression recognition. In previous works, there have two main categories of features: geometrical feature and appear-ance feature. The popular geometrical feature is the key points extracted by active shape model (ASM)[4]. Because ASM is sensitive to illumination, pose, and exaggerated expression, we use the appearance features. The popular local appearance descriptors contain Gabor filter, Harr-like features, and LBP. In [14], Shan, et al, reported that LBP is as powerful as Gabor features. The experiments in [20] demonstrated that haar-like features are comparable with Gabor feature and even better for expression recognition, and similar conclusions were obtained in [23]. Due to the simplicity of the haar-like features and the effectiveness in face detection [17] and facial expression[23], we adopt the haar-like features to represent face appearance.

As described above, we extract 49 patches(sub-windows) from one facial image, and we first extract the haar-like features from each patch. For convenience, we denote the patches as $\{P_1, ..., P_{49}\}$, and the haar-like feature set $\{f_{p,i}\}$ for each patch $P$, where $p$ is the index of patch, $i$ is the feature index within the corresponding patch. The number of the haar-like features are $M$ features in each patch. Therefore, for one facial image, we totally have a feature pool of $\{\{f_{1,i}\}, ...\{f_{49,i}\}\}$. Based on these raw features $\{\{f_{1,i}\}, ...\{f_{49,i}\}\}$, we want to explore the compositional features which combine some possible subset features together, and we expect such combination consistent to the interpretation of FACS. Figure 2 shows an example of a compositional feature which is composed of three raw features from three different patches. The details of compositional feature pursuit are addressed in the next section.

## 3. Building Compositional Features

Inspired by the observation that usually several AUs appear simultaneously in an expression, we build compositional features from different patches to analyze facial ex-
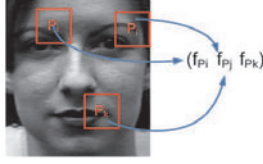
Figure 2. The sample of a compositional feature.

pression. Some works on face detection have demonstrated that using compositional features to build weak learner can further improve the performance of Boosting classifier [12][25][11]. For example, Mita, et.al, [12] used jointed haar-like features to improve the performance of face detector; Liu, et. al, [11] used linear combined features through Kullback-Leibler(KL) boosting to learn face detector. From the view of composition strategy, there are two popular strategies for boosting: 1) the compositional feature pool is built first, and then boosting is directly performed on the compositional feature pool and to select some compositional features to construct the final strong classifiers[26][12]; 2) the procedure of feature combination is integrated into boosting iterations [11][28]. For the former, feature combination is based on some rules or random selection, and it does not need any optimization procedure. The improvement of performance depends on the power of the compositional features in the feature pool. For the latter, during each iteration of boosting, an optimization procedure is used to combine some features to fit the data well. However, such optimization procedure often takes much time because of the large search space. To alleviate this issue, [11] and [26] choose the promising feature set as the candidates, and linear feature combination is done by 1D sequential optimization.

In this paper, we adopt the second strategy, but it is different from [11] and [26] in that we use encoding technique instead of linear combination to compose features. Its main advantage is to reduce the computational cost. It is convenient to build the weak learner of the compositional feature [12], and does not need to optimize the weights of linear combination. We also select a candidate feature set for combination to avoid the exhausted search as in [11] and [26]. The selection of candidate set is as follows: For each patch $P$, we calculate the training error of each feature with the following weak classifier:

$$s(x) = \begin{cases} 1 & if \quad p \cdot f_{p,i}(x) > p \cdot \theta_{p,i} \\ 0 & otherwise \end{cases} \quad (1)$$

where $\theta_{p,i}$ minimizes the weighted error, and it changes with the updated weights in each iteration of boosting. We pick up the top $l$ features with minimum errors from each patch as the candidate features. Therefore, the total size of

the candidate set $\{\mathscr{F}_i\}$ is $49 * l$. We rank all the features in the $\{\mathscr{F}_i\}$ according to the error. Based on $\{\mathscr{F}_i\}$, the procedure of feature combination is as follows: First, the most discriminative feature $\mathscr{F}_1$ is selected from $\{\mathscr{F}_i\}$ and the training error $\varepsilon$ is set as $\varepsilon_1$. Then the $\mathscr{F}_i$ is to be added to the combined feature if the training error is reduced after combination. Equation 2 shows this update rule. The weak classifier (1) is based on the binary value of the compositional feature, and the weighted training error is updated to the reduced one. Recursively, the combined feature continually grows up to the maximum length $L$ or stops when Equation (2) is not satisfied.

$$\mathscr{F}_i \rightarrow [\mathscr{F}_1, \ldots, \mathscr{F}_m],$$
$$if \quad \varepsilon_{[\mathscr{F}_1, \ldots, \mathscr{F}_m, \mathscr{F}_i]} < \varepsilon_{[\mathscr{F}_1, \ldots, \mathscr{F}_m]} \quad (2)$$

The combined feature is encoded as a binary vector. Here we give an example to show how to do encoding. Assume the size of the compositional feature is 3, and the selected promising features are $(\mathscr{F}_1, \mathscr{F}_j, \mathscr{F}_k)$. On the sample $x_i$, $s(\mathscr{F}_1(x_i)) = 0$, $s(\mathscr{F}_j(x_i)) = 1$ and $s(\mathscr{F}_k(x_i)) = 1$. The encoded vector of this compositional feature is: $V_{[\mathscr{F}_1, \mathscr{F}_j, \mathscr{F}_k]}(x_i) = (011)_2 = 3$.

The procedure of building compositional feature is summarized in Algorithm 1.

---

**Algorithm 1** *Feature combination procedure*

---

1: Give image Samples and the corresponding weights $(x_i, y_i, w_i), \ldots, (x_n, y_n, w_i)$, $y_i \in \{1, 0\}$.
2: Calculate the weighted training errors $\varepsilon_{p,i}$ on each feature $f_{p,j}$.
3: Pick up the top $l$ features from each $\{f_{p,i}\}$ base on $\varepsilon_{p,i}$ , and rank them to build $\{\mathscr{F}\}$.
4: Encode $\{\mathscr{F}\}$ on each sample for feature combination.
5: **while** The length of combined feature is less than $L$ or Equation 2 is not satisfied when the whole promising feature set is explored. **do**
6:     Expend the combined feature by greedy search based on minimizing the training error.
7:     Update the training error
8: **end while**
9: Record the current combined feature.

---

## 4. Boosting Learning

We integrate the procedure of building compositional features into boosting, so the total boosting learning can be summarized as in Algorithm 2. Different from the original the original Adaboost, which selects the best feature in each round, we try to find a compositional feature, and the weak classifier is built based on the combined feature to update the weight of samples. Because Adaboost is a typical binary classifier, while we need to classify six basic expressions, so we use the one-against-all strategy to decompose

the six-class problem into multiple two-class problems. For each expression, we set its samples as the positive samples, and the samples of other expressions as the negative samples.

---

**Algorithm 2** *Boosting Learning procedure*

---

1: Give example images $x_1,...,x_n$, and labels $y_1,...,y_n$, where $y_i \in \pm1$
2: Initialize weight $w_t(i) = 1/N$.
3: **for** $t = 0 \,.... \, T$ **do**
4:    Run **Algorithm1** to build one combined feature.
5:    Train a weak classifier $h_t$ based on the combined feature, and calculate the training error $\varepsilon_t$
6:    Choose $\alpha_t = \frac{1}{2}\frac{1-\varepsilon_t}{\varepsilon_t}$, and
   Update:
   $D_{t+1}(i) = \frac{D_t(i)\exp(-y_i\alpha_t h_t(i))}{Z_t}$
   where $Z_t$ is a normalization factor.
7: **end for**
8: Output the final classifier
$$H(x) = \sum_{t}^{T} \alpha_t h_t(x).$$

---

## 5. Experiment

We conducted our experiments on the Cohn-Kanade facial expression database [9], which is widely used to evaluate the facial expression recognition algorithms. This database consists of 100 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 prototypic emotions mentioned above. For our experiments, we selected 352 image sequences from 96 subjects. The selection criterion was that a sequence could be labeled as one of the six basic emotions: anger, disgust, fear, happiness, sadness and surprise. Figure 3 shows the samples of six expressions.

In the previous work[15][10][6][14], the researchers focus on the recognizing the expressions at the apex level, and missed the frames at the low intensity level. However, in recent years, psychological researches have demonstrated that besides the categories of expression, facial expression dynamics is important when attempting to decipher its meaning[1]. Therefore, recognizing the expression with low intensity is also necessary. Driven by this purpose, we take two strategies to organize experiment data set: 1) Apex frames only; 2) The most frames which cover the status from onset to apex (We denote it as the extended data for simplicity). For the apex data, we pick up the last three frames from each sequence following to [15][10][6]. For the extended data, we pick up more frames from each sequence, which cover the status from onset to apex. Figure 3

display the apex data and the the data at low intensity. We can see that facial expression recognition at low intensity level is a challenge task, even for humane beings.
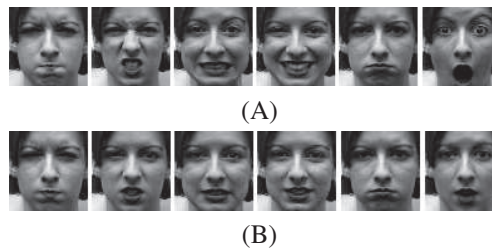


(A)

(B)

Figure 3. The example of six expressions: Anger, Disgust, Fear, Happiness, Sadness and Surprise. (A)Samples of the apex data which come from the last frame of the sequence; (B)Samples of the extended data which are at low intensity level.

In this paper, we normalize the images to $64 \times 64$ based on the location of eyes as in [23]. The patch size is set to $16 \times 16$, and 774 haar-like features are extracted from each patch. So totally we have 37926 haar-like features for each image. To better evaluate the proposed method, we compare the proposed method to two methods. One is similar to [10], where we replace the Gabor feature by Haar feature. Due to no feature scale limitation in the framework of[10], totally 195552 haar features are extracted for Adaboost to select. We also perform Adaboost on the feature pool extracted from the local patches exactly as the proposed method. For convenience, we denote the proposed method and these two methods as Combined feature+Adaboost, Haar+Adaboost, and Haar(constrained)+Adaboost respectively. In our experiment, we randomly select 66% subjects as the training set, and the rest subjects as the testing set. There are two key parameters in the proposed method: 1) the number of top feature picked $l$ ; 2) the maximum length of compositional feature $L$; We set both $l$ to 5 for all the experiments empirically, and $L$ is up to 15.

### 5.1. Results on Apex Data

Before we report the results on the testing set, we would like to analyze the performances of three methods on the training set. Figure 4 shows the performance of the training error vs. the number of the features. We can see that the converge rates of Combine feature+Adaboost is a little bitter faster than Haar+Adaboost and Haar(constrained)+Adaboost. The experimental results on the testing set are listed in Table 1. We can see that the proposed method outperforms the other two. The improvement of the proposed method is slight compared to Haar+Adaboost. Our method obtains the recognition accuracy of 92.3%, while that of Haar+Adaboost is 92.1%. This is reasonable that the apex is easy to discriminate. However, for the low expression intensity data, the advan-
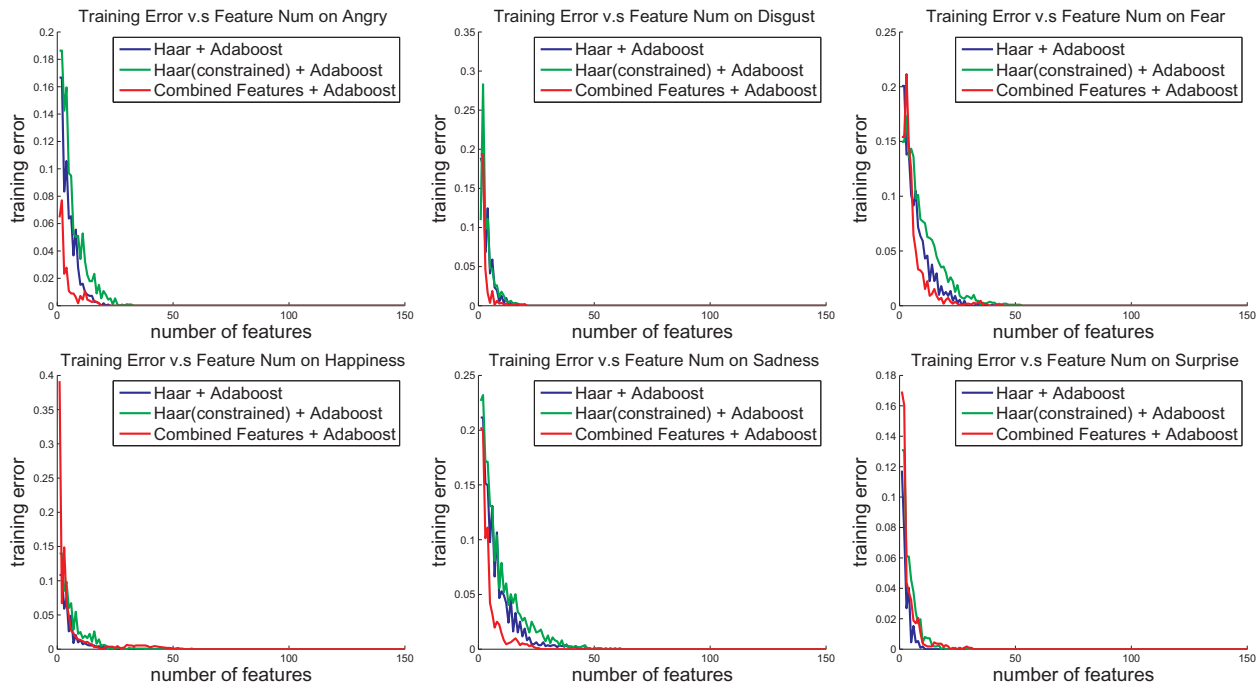
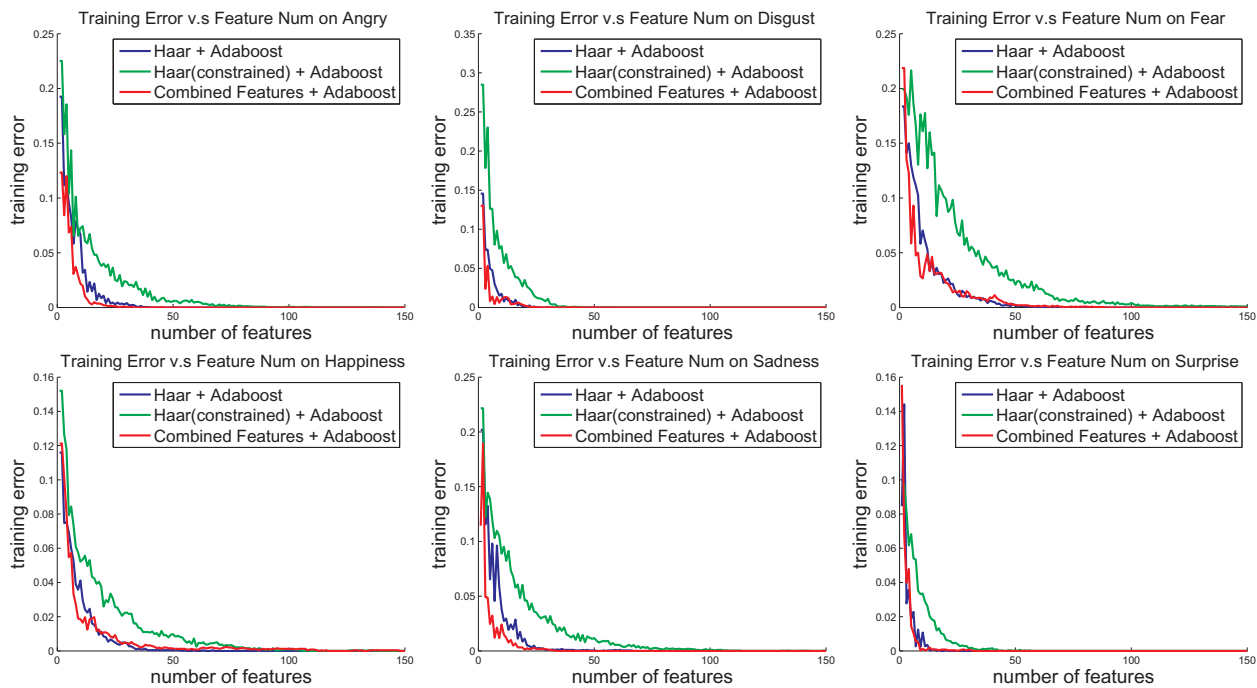Figure 4. Training error VS. Number of features on the apex data.



Figure 5. Training error VS. Number of features on the extended data.

tage of the proposed method will become obvious, and we will report this group experiments in the next subsection. Haar(constrained)+Adaboost obtains a worse performance, because its feature pool is much smaller than that of

Haar+Adaboost. Based on the results reported in the state-of-arts methods [15][10][6][14] on the same apex data, the proposed method is comparable to them. For example, in [10], they used the Gabor features and Adaboost classifier,

Table 1. Performances on the testing set

| Methods | Accuracy (%) |
|---|---|
| Haar + Adaboost | 92.1 |
| Haar(constrained) + Adaboost | 88.43 |
| Combined Features + Adaboost | 92.3 |

Table 3. Performances on the extended testing set

| Methods | Accuracy (%) |
|---|---|
| Haar + Adaboost | 78.4 |
| Haar(constrained) + Adaboost | 74.1 |
| Combined Features + Adaboost | 80.0 |

and they obtained the recognition accuracy of 90.1% under the leave-one-out testing protocol. Table 2 lists the confusion matrix of the proposed method.

To further evaluate the proposed method, we want to investigate the relationship between the selected compositional features and AUs. Figure 6 shows the top features selected by the three methods. We can see that the selected top compositional features are consistent to the interpretation of FACS. For example, sadness expression may involve AU1+AU4+AU15+AU17, the first compositional feature contains the corresponding features which cover all these AUs.
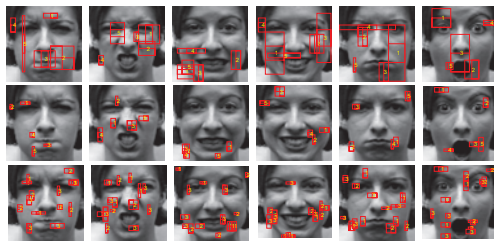


Figure 6. The distribution of the selected features on the Apex Data. Top row: the top 5 features selected in Haar+boosting; Middle row: the top 5 features selected in Haar(constrained)+boosting; Bottom row: the first 3 compositional features in our method.

### 5.2. Results on Extended Data

The apex data is too limited to evaluate the real performance of methods, so we extend the data set to cover more frames at low intensity level to evaluate the three methods. Figure 5 shows the performance of the training error vs. the number of features. Table 3 shows the results on the testing set. Similar to the results on the apex data, combined feature+Adaboost outperforms the other two, and Haar+Adaboost is better than Haar(constrained)+Adaboost. However, Combine feature+Adaboost achieves much higher recognition accuracy (80.0%) than Haar+Adaboost (78.41%), because the proposed method can efficiently capture several AUs simultaneously to enhance the discrimination ability, especially at the low expression level. Figure7 shows the distribution of the top features selected by the three methods. We can see that the top selected compositional features are similar to the results on the apex data, so it shows that the proposed method has physical significance. Table 4 shows the confusion matrix of the proposed method.



Figure 7. The distribution of the selected features on the extended Data. Top row: the top 5 features selected in Haar+boosting; Middle row: the top 5 features selected in Haar(constrained)+boosting; Bottom row: the first 3 compositional features in our method.

## 6. Conclusion

In this paper, we proposed a new facial expression recognition method with compositional features. We first divide face image into blocks to cover the AUs' location, and then we extract local appearance features from each patch. A minimum error based optimization strategy is adopted to build compositional features based on local appearance features, and this process embedded into Boosting learning structure. Experiments on the Cohn-Kanada database demonstrated that the proposed method has a promising performance. Especially the proposed method showed a good performance on the low expression intensity data. Moreover, the experimental results illustrated that and the selected compositional features are basically consistent to the interpretation of FACS.

## References

[1] Z. Ambadar, J. Schooler, and J. F. Cohn. Deciphering the enigmatic face The importance of facial dynamics in interpreting subtle facial expression. *Psychological Science*, 2005. 4

[2] 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*, 91(1-2), 2003. 1

[3] J. Cohn. Automated analysis of the configuration and timing of facial expression. *What the face reveals (2nd edition): Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*, 2005. 1

[4] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models: their training and application. *Comput. Vis. Image Underst.*, 61(1), 1995. 2

Table 2. The confusion matrix of our proposed method based on the Apex Data

| Recognition rate | Angry | Disgust | Fear | Happiness | Sadness | Surprise |
|---|---|---|---|---|---|---|
| Angry | 84.38 | 0 | 3.13 | 0 | 12.50 | 0 |
| Disgust | 2.78 | 91.67 | 0 | 0 | 5.56 | 0 |
| Fear | 0 | 0 | 93.18 | 2.27 | 4.55 | 0 |
| Happiness | 1.47 | 0 | 0 | 95.59 | 0 | 2.94 |
| Sadness | 0 | 0 | 9.62 | 0 | 90.38 | 0 |
| Surprise | 0 | 0 | 0 | 0 | 1.67 | 98.33 |

Table 4. The confusion matrix of our proposed method based on the extended Data

| Recognition rate | Angry | Disgust | Fear | Happiness | Sadness | Surprise |
|---|---|---|---|---|---|---|
| Angry | 81.18 | 11.82 | 3.76 | 1.07 | 2.15 | 0 |
| Disgust | 13.52 | 55.29 | 22.94 | 2.94 | 2.94 | 2.35 |
| Fear | 0.46 | 0.93 | 81.39 | 10.69 | 6.04 | 0.46 |
| Happiness | 0 | 0 | 5.09 | 94.90 | 0 | 0 |
| Sadness | 1.26 | 0.84 | 11.34 | 8.40 | 74.36 | 3.78 |
| Surprise | 0 | 1.93 | 1.16 | 0.38 | 5.42 | 91.08 |

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

[6] G.Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6), 2007. 1, 4, 5

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

[8] J.Hoey. Hierarchical unsupervised learning of facial expression categories. *IEEE Workshop on Detection and Recognition of Events in Video.*, 2001. 1

[9] T. Kanade, J. Cohn, and Y.-L. Tian. Comprehensive database for facial expression analysis. *Proceedings of the 4th IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG'00)*, 2000. 4

[10] G. Littlewort, M. S. Bartlett, I. Fasel, J. Susskind, and J. Movellan. Dynamics of facial expression extracted automatically from video. *J. Image and Vision Computing*, 2006. 1, 4, 5

[11] C. Liu and H.-Y. Shum. Kullback-leibler boosting. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2003. 3

[12] T. Mita, T. Kaneko, and O. Hori. Joint haar-like features for face detection. *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, 2005. 3

[13] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000. 1

[14] C. Shan, S. Gong, and P. W.McOwan. Robust facial expression recognition using local binary patterns. *IEEE Int. Conf. on Image Processing*, 2005. 1, 2, 4, 5

[15] Y. Tian. Evaluation of face resolution for expression analysis. *Computer Vision and Pattern Recognition Workshop on Face Processing in Video*, 2004. 1, 4, 5

[16] Y. Tian, T. Kanade, and J. Cohn. Evaluation of gaborwavelet -based facial action unit recognition in image sequences of increasing complexity. *(FG'02)*, 2002. 1

[17] K. Tieu and P. Viola. Boosting image retrieval. *IEEE Computer Vision and Pattern Recognition*, 2000. 2

[18] Y. Tong, W. Liao, and Q. Ji. Facial action unit recognition by exploiting their dynamic and semantic relationships. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10), 2007. 1

[19] P. Viola and M. Jones. Robust real-time object detection. *Int. J. Computer Vision*, 57(2), 2002. 2

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

[21] Y. Yacoob and L. Davis. Computing spatio-temporal representations of human faces. *Computer Vision and Pattern Recognition*, 1994. 1

[22] P. Yang, Q. Liu, and D. N. Metaxas. Facial expression recognition based on dynamic binary patterns. *Computer Vision and Pattern Recognition*, 2008. 1

[23] P. Yang, Q. Liu, and D. N. Metaxas. Boosting encoded dynamic features for facial expression recognition. *Pattern Recogn. Lett.*, 30(2), 2009. 1, 2, 4

[24] M. Yeasin, B. Bullot, and R. Sharma. From facial expression to level of interest: A spatio-temporal approach. *Computer Vision and Pattern Recognition*, 2004. 1

[25] X.-C. Yin, C.-P. Liu, and Z. Han. Feature combination using boosting. *Pattern Recogn. Lett.*, 26(14), 2005. 3

[26] J. Yuan, J. Luo, and Y. Wu. Mining compositional features for boosting. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008. 3

[27] Z. Zeng, M. Pantic, G. Roisman, and T. Huang. A Survey of Affect Recognition Methods: Audio, Visual, and Spontaneous Expressions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 31(1), 2009. 1

[28] X. Zhang, L. Liang, X. Tang, and H.-Y. Shum. L1 regularized projection pursuit for additive model learning. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2008. 3