Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Two-stage Multi-class AdaBoost for Facial Expression Recognition

Hongbo Deng, Jianke Zhu, Michael R. Lyu and Irwin King

Abstract— **Although AdaBoost has achieved great success, it still suffers from following problems: (1) the training process could be unmanageable when the number of features is extremely large; (2) the same weak classifier may be learned multiple times from a weak classifier pool, which does not provide additional information for updating the model; (3) there is an imbalance between the amount of the positive samples and that of the negative samples for multi-class classification problems. In this paper, we propose a two-stage AdaBoost learning framework to select and fuse the discriminative feature effectively. Moreover, an improved AdaBoost algorithm is developed to select weak classifiers. Instead of boosting in the original feature space, whose dimensionality is usually very high, multiple feature subspaces with lower dimensionality are generated. In the first stage, boosting is carried out in each subspace. Then the trained classifiers are further combined with simple fusion method in the second stage. Experimental results on facial expression recognition data demonstrate that our proposed algorithms not only reduce the computational cost for training, but also achieve comparable classification performance.**

I. Introduction

AdaBoost is a well-known learning algorithm, which has been extensively studied in recent years [14], [15], [16]. The essence of AdaBoost is to learn a number of simple weak classifiers that are linearly combined into a single strong classifier. The major advantage of AdaBoost algorithm is the adaptive selection of discriminative and complementary features during the training process. In the past, AdaBoost [15] has been successfully applied to many different applications, including object detection [18], face recognition [9], [20], text categorization [16] as well as proving to be suitable for feature selection [17].

The objective of AdaBoost is to find a highly accurate classification rule by fusing many weak classifiers together, and each of them may be only moderately accurate. Moreover, a separate procedure for computing the weak classifier is called the weak learner [16]. In [18], the feature selection is achieved through a simple modification of the AdaBoost procedure: the weak learner is constrained so that each weak classifier returned can depend on only a single feature. As a result, each stage of the boosting process selects a new weak classifier, which can be viewed as a feature selection process. Therefore, AdaBoost has also been employed for the feature selection and fusion [17] with the most discriminative information. Selectivity reduces the dimensionality of the feature space that in turn results in significant of the feature space. Thus, this leads to the significant speed up for the online classification task.

Hongbo Deng, Jianke Zhu, Michael R. Lyu and Irwin King are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. E-mail: {hbdeng, jkzhu, lyu, king}@cse.cuhk.edu.hk.

Although AdaBoost has achieved great success, it still suffers from following problems: (1) the training process could be unmanageable when the number of features is extremely large, and the most time-consuming step is to search the optimal feature by the weak learner; (2) the same weak classifier may be learned multiple times from a weak classifier pool, which does not provide additional information for updating the model; (3) there is an imbalance between the amount of the positive samples and that of the negative samples for multi-class classification problem. To this end, we propose an improved AdaBoost algorithm with a simple weak learner, which is able to deal with these problems.

The contributions in this paper can be summarized as follows. We propose a two-stage AdaBoost learning framework to reduce the computational cost for the high-dimensional features by selecting and fusing the discriminative features effectively. Moreover, an improved AdaBoost algorithm is developed to select weak classifiers.

To evaluate the effectiveness of our proposed AdaBoost algorithms, we apply them to Facial Expression Recognition (FER) tasks. Facial expressions deliver rich information about human emotion and play an essential role in human communications. In order to facilitate a more intelligent and natural human machine interface for new multimedia products, automatic facial expression recognition [3], [5], [13] had been widely studied in the past decade or two, which has become a very active research area in computer vision and pattern recognition. Experimental results on facial expression recognition data demonstrate that our proposed algorithms not only reduce the training computational cost, but also achieve comparable classification performance.

The rest of the paper is organized as follows. Section II reviews the related work in facial expression analysis and AdaBoost. In Section III, we present and formulate our proposed AdaBoost algorithm that can select the most discriminative features and construct a real-time strong classifiers. Experiments and performance evaluations are given in Section IV. Finally, Section V concludes our work.

II. Related Work

In the literature, a number of approaches have been proposed for facial expression analysis, which includes both the image sequences based methods [21], [24] and the still image based ones [4], [23]. As for the image sequences, several successful approaches have been proposed, such as Optical Flow models [21] and Hidden Markov Models (HMM) [24]. On the other hand, the methods for the still image are categorized into the two classes: holistic spatial analysis, such as Eigenfaces and Fisherfaces [1], and local

1-4244-1380-X/07/$25.00 ©2007 IEEE

spatial analysis, like Gabor wavelet [11] and local Principle Component Analysis (PCA) [3].

Since the feature extraction is very important for the facial expression analysis, various feature extraction methods were compared in [3] including Eigenfaces, Fisherfaces, Gabor wavelets, etc. Moreover, the best performance is obtained through the Gabor wavelets representation. The number of Gabor filters used to convolve face images varies with the applications, and usually 40 filters (5 scales and 8 orientations) are used [3], [5], [10]. Due to the large number of convolution operations, the computational cost and memory requirement of such Gabor features are very large. Additionally, the dimensionality is incredibly high compared with the small training samples. In order to ensure a fast online classification task, the FER process must exclude a large majority of the available features, and focus on a small set of critical features. So many AdaBoost algorithms select the most discriminative Gabor features and combine them together optimally to construct an effective and real-time FER system. However, the performance is sensitive to the selected features.

One of the key issues in AdaBoost [14], [15], [16], [17], [18] is how to choose the weak classifier (also feature). The weak classifier could be very simple. In [18], the weak learner determines the optimal threshold classification function for each feature, and successfully applied to face detection. Yang et al. [22] introduced the intra-face and extra-face difference space for each image pairs, but such method may generate tremendous imbalance of the positive and the negative samples. To tackle this problem, we adopt a difference space between the feature and the mean, which can utilize the label information.

For feature selection, classifiers with similar features are more likely to be selected and redundancy will exist among some selected features. FloatBoost [9] checks previous selected weak classifiers and eliminates non-effective classifiers during the learning process. Later, Shen et al. [17] extended the work by incorporating mutual information. However, the computational cost is extremely high especially when the number of features is large. To handle this problem, we adopt a decay factor in the weak learning process. Recently, random subspace method has been applied in various machine learning tasks to attack the high dimensional data. The sample features are divided into n disjoint subspaces [2] or random subspaces [19], then boosting method is used on each of the subsets independently. Inspired by those approaches, we propose a two-stage multi-class AdaBoost algorithm to select and combine discriminative Gabor features, which not only reduces the training computation, but also achieves comparable classification performance.

III. AdaBoost Leaning Algorithm

In this section, we first describe the formal setting in AdaBoost learning, and then review the original multi-class AdaBoost algorithm. Finally, we present and formulate our proposed two-stage multi-class AdaBoost method.

A. Preliminaries

Let denote $\mathcal{X}$ as the sample space, and $\mathcal{Y}$ is a finite set of classes. The size of $\mathcal{Y}$ is defined as $k = |\mathcal{Y}|$. In the multi-class single-label case, a sample pair is represented by (x, Y), where $x \in \mathcal{X}$, $Y \in \mathcal{Y}$. $Y[l]$ for $l \in \mathcal{Y}$ is defined as

$$Y[l] = \begin{cases} +1 & if\ l = Y \\ -1 & if\ l \neq Y \end{cases}. \tag{1}$$

The objective of AdaBoost learning [16] is to find a classifier $f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ with interpretation that, for a given sample x, the labels in $\mathcal{Y}$ should be ordered according to $f(x, \cdot)$. That is, a label l_1 is considered to be ranked higher than l_2 if $f(x, l_1) > f(x, l_2)$. If l_i is the associated label for x, then a successful learning algorithm tend to rank label l_i higher than others. Therefore the output label l corresponding to sample x is determined by

$$l = \arg\max_{l_i \in \mathcal{Y}} f(x, l_i). \tag{2}$$

B. Weak Learner

To simplify the the weak learning process, we choose a simple and effective weak classifier function $h(x, l)$, called a threshold function, on the j_{th} coordinate of x in the n-dimensional space. In order to fully utilize the discriminative information, a difference space between the feature x and the mean is employed to determine the weak classifier

$$h_j(x, l) = sign[(x - \mu_l)^j - b], \tag{3}$$

where μ_l is the mean feature vector of the class l, b is a threshold, and superscript j denotes the j_{th} coordinate of x. For each feature, the weak classifier determines the optimal threshold classification function. Therefore, a weak classifier pool $\mathcal{H}$ is constructed through the threshold function $h(x, l)$.

Assuming l is the label of a sample x, and then $(x - \mu_l)$ is a positive example; otherwise, $(x - \mu_l)$ is a negative example. Comparing to the intra-face space and the extra-face space [9], the advantage of such method is to utilize the discriminative information without generating imbalance of the positive and the negative samples.

C. Original Multi-class AdaBoost Algorithm

The idea of AdaBoost is that the weak classifiers are used to form a highly accurate prediction rule via calling the weak classifiers repeatedly on different distributions over the training samples [15]. Moreover, AdaBoost algorithm maintains a set of weights as a distribution D_t over samples and labels. On each round t, the distribution D_t is passed to the weak learner who searches a weak classifier h_t. The output of the weak learner is a classifier $h : \mathcal{X} \times \mathcal{Y} \rightarrow \{-1, +1\}$. Over a number of T rounds, T weak classifiers try to exhaustively search the one with the maximum weighted classification rate. The parameter r_t is then used to update the distribution D_t so that the weights of misclassified samples are increased and the weights of correct classified samples are decreased. In this way, the algorithm focuses on difficult training samples, increasing their representation in successive training sets. Finally, a strong classifier is constructed through

Algorithm 1 Original Multi-class AdaBoost

Given: m training samples $(x_1, Y_1),\ldots,(x_m, Y_m)$, and the weak classifier pool $\mathcal{H} = \{h_j(x,l)\}$.

1: Initialize: $D_1(i,l) = 1/mk, \quad i = 1\ldots m, \quad l = 1\ldots k$, where $k = |\mathcal{Y}|$.
2: **for** $t = 1\ldots T$ **do**
3: Under the distribution D_t, find a weak classifier $h_t : \mathcal{X} \times \mathcal{Y} \rightarrow \{-1, +1\}$ from $\mathcal{H} = \{h_j(x,l)\}$ to maximize the absolute value of

$$h_t = \arg\max_{h_j \in \mathcal{H}} r_j,$$

where $r_j = \sum_{i,l} D_t(i,l) Y_i[l] h_j(x_i, l)$.
4: Prerequisite: $r_t > 0$, otherwise break the loop.
5: Choose $\alpha_t \in \mathbb{R}$, typically $\alpha_t = \frac{1}{2}\ln\frac{1+r_t}{1-r_t}$.
6: Update the distribution:

$$D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-\alpha_t Y_i[l] h_t(x_i,l))}{Z_t},$$

where Z_t is a normalization factor, so that D_{t+1} will be a distribution.
7: **end for**
8: Output the final strong classifier:

$$f(x,l) = \sum_t \alpha_t h_t(x,l).$$

RETURN $f(x,l)$.

Algorithm 2 Improved Multi-Class AdaBoost

Given: m training samples $(x_1, Y_1),\ldots,(x_m, Y_m)$, the weak classifier pool $\mathcal{H} = \{h_j(x,l)\}$, and the number of the weak classifier T.

1: Initialize: $D_1(i,l) = 1/mk$, $P_1(j,l) = 1$, $i = 1\ldots m$, $j = 1\ldots n$, $l = 1\ldots k$, where $k = |\mathcal{Y}|, n = |\mathcal{H}|$.
2: **for** $t = 1\ldots T$ **do**
3: Under the distribution D_t, find a weak classifier $h_t : \mathcal{X} \times \mathcal{Y} \rightarrow \{-1, +1\}$ from $\mathcal{H} = \{h_j(x,l)\}$ to maximize the absolute value of

$$h_t = \arg\max_{h_j \in \mathcal{H}} r_j,$$

$$P_{t+1}(j,l) = P_t(j,l) + 1,$$

where $r_j = \sum_{i,l} \frac{D_t(i,l) Y_i[l] h_j(x_i,l)}{P_t(j,l)}$.
4: Prerequisite: $r_t > 0$, otherwise break the loop.
5: Choose $\alpha_t \in \mathbb{R}$, typically $\alpha_t = \frac{1}{2}\ln\frac{1+r_t}{1-r_t}$.
6: Update the distribution:

$$D_{t+1}(i,l) = \frac{D_t(i,l)\exp(-\alpha_t Y_i[l] h_t(x_i,l))}{Z_t},$$

where Z_t is a normalization factor, so that D_{t+1} will be a distribution.
7: **end for**
8: Output the final strong classifier:

$$f(x,l) = \sum_t \alpha_t h_t(x,l).$$

RETURN $f(x,l)$.

a weighted vote of the weak classifiers. The details of the original algorithm are described in Algorithm 1.

D. Proposed Two-stage Multi-class AdaBoost Algorithm

Although the Algorithm 1 is simple and effective for boosting the weak classifiers, it has two distinct drawbacks. First, the same weak classifier with very high classification rate may be selected many times, which has no more contribution to the final strong classifier. In the extreme, when r_i almost approximate 1, the worst case is that most of the selected weak classifiers are the same one, which results in generalization error. Second, the training process could be unmanageable when the number of features is extremely large (10,000+).

To tackle the first problem, a decay factor is employed to constrain the selection of duplicated weak classifiers. $P(j,l)$ is denoted as the decay factor for each weak classifier in the weak classifier pool $\mathcal{H}$, which increases one for current selected weak classifier. The optimization solution is formulated as follows:

$$h_t = \arg\max_{h_j \in \mathcal{H}} \sum_{i,l} \frac{D_t(i,l) Y_i[l] h_j(x_i,l)}{P_t(j,l)}. \tag{4}$$

During the weak learning process, all weak classifiers compete to be selected. The basic idea of our proposed AdaBoost is to prevent from learning the previous selected weak classifiers by reducing their probability, and to restrain the previous selected weak classifiers. With the same number of selected weak classifiers, experiments demonstrate that the performance is better than that of the original one. The details of improved multi-class AdaBoost is summarized in Algorithm 2.

The overall time complexity of the algorithm is dominated by Step 3 and Step 6. In Step 3, it involves with searching the optimal weak classifier by weak learner, and the complexity is $\mathcal{O}(mkn)$ ($n \propto |\mathcal{X}|$) in this work. For Step 6, it will cost $\mathcal{O}(mk)$ operations to update the distribution. The overall time complexity of the algorithm is $\mathcal{O}(Tmk(n+1))$ ($\approx \mathcal{O}(Tmkn)$ for $n \gg 1$). Hence, the time-consuming step is to search feature space exhaustively to find the optimal weak classifier, especially for those high dimensional feature.

In order to reduce the computational cost for such high dimensional feature, we propose a two-stage AdaBoost learning framework to address the second problem. According to the sample features, we divide the feature space $\mathcal{X}$ to M disjoint subspaces $\hat{\mathcal{X}}_i$. Then we first perform AdaBoost algorithm on each subspace to obtain M AdaBoost classifiers respectively. Since these classifiers are trained independently, the computation can be parallel on multiple computers. Then a simple sum rule is adopted to combine successively complex classifiers $S(x,l)$. The details of our algorithm are shown in Algorithm 3.

As mentioned above, the complexity of the Step 3 is $\mathcal{O}(\frac{T}{M}mkn_i)$ ($n_i \propto |\hat{\mathcal{X}}_i|$) in Algorithm 3, then the

Algorithm 3 Two-stage Multi-class AdaBoost

Given: m training samples $(x_1, Y_1),\ldots,(x_m, Y_m)$, the weak classifier pool $\mathcal{H} = \{h_j(x, l)\}$, and the number of the weak classifier T.

1: Initialize: Divide $\mathcal{X}$ to M subspace $\hat{\mathcal{X}}_1, \hat{\mathcal{X}}_2, \ldots, \hat{\mathcal{X}}_M$, and $|\mathcal{X}| = \sum_i |\hat{\mathcal{X}}_i|$.
2: **for** $i = 1 \ldots M$ **do**
3: Pass parameter $\hat{\mathcal{X}}_i \times \mathcal{Y}, \frac{T}{M}, \mathcal{H}$ to AdaBoost, and get $\frac{T}{M}$ weak classifiers.
4: Get individual strong classifier $f_i : \hat{\mathcal{X}}_i \times \mathcal{Y} \to \mathbb{R}$.
5: **end for**
6: Output the final combined strong classifier:

$$S(x, l) = \sum_i f_i(\hat{x}_i, l).$$

RETURN $S(x, l)$.

overall complexity to learning T weak classifiers is $\mathcal{O}(\sum_i \frac{T}{M} mkn_i) = \mathcal{O}(\frac{T}{M} mkn)$. We can observe that the complexity is reduced greatly to select T weak classifier. The performance of the two-stage AdaBoost is evaluated in Section IV-D.2

IV. Experimental Results

In this section, we report empirical evaluations of our proposed algorithm for facial expression recognition. We first describe the details of our testbeds, and then discuss our preprocessing and feature extraction methods. Finally, we show the experimental results. In addition, we also implemented PCA and Linear Discriminant Analysis (LDA) as baseline methods for the feature selection. The experiments were conducted on a PC with 3.0GHz CPU and 1GB memory.

A. Experimental Datasets

To evaluate the performance of the proposed method for facial expression recognition, we have collected two different kinds of datasets as our experimental testbeds. One is the JApanese Female Facial Expression (JAFFE) Database [11]. The other is the AR Database [12]. Table I summarizes the detailed information of the datasets used in our experiments.

TABLE I
THE FACIAL EXPRESSION IMAGE DATASETS USED IN THE EXPERIMENTS.

Dataset	#total	#expression (E)	#person (P)	#per E & P
JAFFE	213	7	10	3 - 4
AR	1008	4	126	2

JAFFE dataset contains 213 images of seven facial expressions posed by ten Japanese female persons. Every person posed three or four examples for each of the seven facial expressions (happiness, sadness, surprise, anger, disgust, fear, neutral). For the experiments reported in this section, two images for each expression of each subject are randomly selected for training, while the remaining data are employed for testing. Due to the limited size of the JAFFE dataset, we perform the trial using three-fold cross-validation to obtain the average recognition rate.

AR dataset consists of over 3,200 color frontal facial images belong to 126 subjects. For each subject, these images were recorded in two different sessions separated by two weeks, and each session took 13 images. In our experiment, four images with different facial expressions are selected. Therefore, there are a total of 1,008 images. For each subject, one image of each expression is selected as the training samples, while the other is employed as the testing samples. We perform the trial using two-fold cross-validation to calculate the average recognition rate.

B. Preprocessing

Before extracting the features from facial images, the preprocessing procedure performs the following steps in order to obtain the normalized pure expression images:

- Detect the centers of the eyes manually.
- Rotate to line up the eye coordinates.
- Crop the face region and resize to fixed size of 128×96.
- Use a histogram equalization method to eliminate the illumination effect.

As shown in Figure 1, the facial images are both geometrically and photometrically normalized.

(a) JAFFE dataset (b) AR dataset

Fig. 1. Example images from JAFFE dataset and AR dataset. (a) From left to right: anger, disgust, fear, happiness, neutral, sadness and surprise. (b) From left to right: neutral, smile, anger, scream.

C. Gabor Feature Extraction

Benefited from the optimal localization properties in both spatial analysis and frequency domain, the Gabor filters have been proven to be a very useful tool in computer vision and image analysis [6], [7], [8], [10], [11], [23].

1) Gabor Filters: In the spatial domain, a Gabor filter is a complex exponential modulated by a Gaussian function [8]. The Gabor filter is usually defined as follows,

$$\psi(x, y, \omega, \theta) = \frac{1}{2\pi\sigma^2} e^{-(\frac{x'^2+y'^2}{2\sigma^2})} [e^{i\omega x'} - e^{-\frac{\omega^2\sigma^2}{2}}], \quad x' = x\cos\theta + y\sin\theta, y' = -x\sin\theta + y\cos\theta, \tag{5}$$

where (x, y) is the pixel position in the spatial domain, ω the radial center frequency (scale), and θ the orientation of Gabor filter. Different choice of ω and θ generates different Gabor filter. In this paper, a Gabor filter bank $G(3 \times 8)$ with 3 scales and 8 orientations is selected.

2) Gabor Feature Representation: The Gabor feature representation of an image $I(x, y)$ is the convolution of the image with the Gabor filter $\psi(x, y, \omega_\mu, \theta_\nu)$ as given by:

$$O_{\mu,\nu}(x, y) = |I(x, y) * \psi(x, y, \omega_\mu, \theta_\nu)|, \quad (6)$$

where "$*$" denotes the convolution operator. Figure 2 shows the magnitudes of the convolution outputs of a face image with 3 scales and 8 orientations.

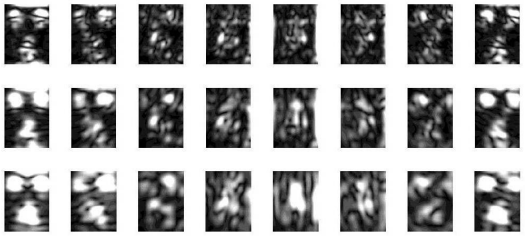


Fig. 2. The magnitudes of the Gabor feature representation.

The Gabor feature set consists of 24 Gabor component-features $S = \{O_{\mu,\nu}\}$ (The output of a Gabor filter is called a Gabor component-feature). To encompass the properties of spatial locality and orientation selectivity, we concatenate [10] all the outputs of Gabor filter bank and derive the Gabor feature vector. Typically, the augmented Gabor feature vector x is constructed by concatenating its rows (or columns), which is able to be defined as follows:

$$x = (o_{11}^T o_{12}^T \dots o_{\mu\nu}^T)^T, \quad (7)$$

where T is the transpose operator. In practice, the dimensionality of a Gabor feature vector is very high. Hence, the computational cost and memory requirements are prohibitively quite large.

D. Experimental Results

Two experiments are designed. In the first experiment, our improved AdaBoost (Algorithm 2) and the original AdaBoost (Algorithm 1) are compared in the classification performance using JAFFE dataset. The second experiment is designed to evaluate the effectiveness of the two-stage learning framework on the JAFFE and the AR datasets. If the feature has 4608 dimensions, Algorithm 1 costs about 14.1 seconds to find one weak classifier, while it only needs 0.43 seconds to get one weak classifier for Algorithm 2, which validates the computation complexity analysis in Section III-D. The baseline experiment is performed using the PCA method on the total Gabor features (3 scales and 8 orientations). In addition, the LDA method is also implemented, similar to the Fisherfaces method [1] method, which applies LDA after PCA dimensionality reduction.

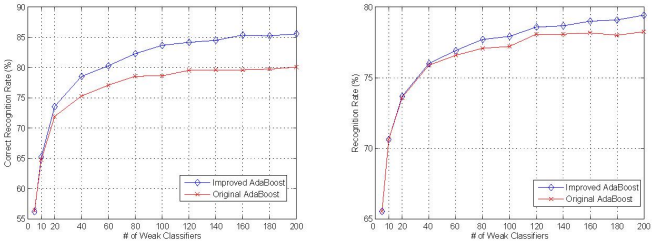
*1) **Experiment 1:*** To compare our improved algorithm with the original one, we applied both AdaBoost algorithms on Gabor component-feature $o_{\mu\nu}$. For each Gabor component-feature, an AdaBoost classifier will be trained to select 200 weak classifiers (features) respectively. In total, 24 individual AdaBoost classifiers are obtained corresponding to 24 Gabor component-features for JAFFE and AR dataset. More specifically, the recognition performance of 24 individual AdaBoost classifiers with 60 selected features are shown in Table II. The row μ corresponds to different scale, and the column ν corresponds to different orientation of Gabor component-features.

We plot the curves of average correct recognition rates corresponding to different number of weak classifiers in Figure 3. It indicates that our proposed algorithm performs much better than original algorithm when the number of selected classifiers increases. Moreover, both algorithms obtain similar recognition rates when only few classifiers are selected. This implies that the redundancy among selected classifiers increases with the number of weak classifiers in original algorithm. As illustrated in Figure 3(a), our proposed algorithm with 60 weak classifiers even outperforms the original algorithm with 200 weak classifiers. From above, we can conclude that the proposed AdaBoost algorithm (Algorithm 2) can achieve better results and restrain the selection of the duplicated weak classifiers.

TABLE II

CORRECT RECOGNITION RATES OF 24 INDIVIDUAL CLASSIFIERS USING ALGORITHM 2 WITH 60 SELECTED FEATURES FOR JAFFE DATASET (%).

$\mu\backslash\nu$	1	2	3	4	5	6	7	8
1	73.3	82.6	74.6	73.3	73.3	76.0	78.6	82.6
2	77.3	81.3	92.0	77.3	70.6	77.3	84.0	84.0
3	80.0	88.0	89.3	81.3	80.0	84.0	88.0	78.6



(a) JAFFE (b) AR

Fig. 3. Performance evaluation for the original AdaBoost algorithm and our improved AdaBoost algorithm.

*2) **Experiment 2:*** We now empirically evaluate the two-stage AdaBoost algorithm on the JAFFE dataset and the AR dataset. In the first stage, 24 individual AdaBoost classifiers corresponding to 24 Gabor component-features are obtained for further fusion of multiple classifiers in our experiment. In the second stage, the final strong classifier is able to be constructed by combining all the individual classifiers according to the summation fusion rule. Actually, the best performance of individual classifier reaches 92.0% in Table II, and the final strong classifier further improves the performance to 96.89%. In the stochastic discrimination theory, classifiers are constructed by combining many components that have weak discriminative power but generalize very well.

Table III reports the correct recognition rates of Two-stage AdaBoost, PCA and LDA on JAFFE and AR datasets.

TABLE III

CLASSIFICATION PERFORMANCE OF TWO-STAGE ADABOOST VS PCA VS LDA (%).

Dataset	PCA	LDA	Two-stage AdaBoost (# of weak classifiers)											
			5	10	20	40	60	80	100	120	140	160	180	200
JAFFE	80.00	95.56	93.34	96.00	96.44	96.89	96.89	**97.78**	97.33	97.33	97.33	96.89	97.33	97.33
AR	83.28	90.09	84.89	86.33	87.80	87.47	88.10	88.53	88.82	87.94	88.44	88.66	**89.16**	89.06

Here # of weak classifiers refers to the number of weak classifiers selected by each individual classifier, not the total number. As for JAFFE and AR dataset, the performance of Two-stage AdaBoost algorithm, with only 5 selected weak classifiers, is much better than the baseline PCA. Moreover, the classification performance improves with the number of weak classifiers. The best performance for JAFFE dataset achieves 97.78%, better result than the LDA method. As for the AR dataset, although the performance of Two-stage AdaBoost is slightly worse than the LDA method, the key point is that a strong classifier can be constructed by choosing only 10 or 20 weak classifiers, which is much better than PCA and LDA for online classification tasks.

V. CONCLUSIONS

In this paper, we propose a novel two-stage multi-class learning framework to attack the problem of computation complexity for high-dimensional features. Moreover, an improved AdaBoost algorithm is developed to restrain previous selected weak classifiers. The advantages of our proposed techniques are explained and demonstrated. We perform the experiments on both JAFFE and AR datasets to evaluate the algorithm. The experimental results demonstrate that our proposed algorithm is effective and promising for reducing the training computation complexity and achieving comparable classification performance in facial expression recognition.

ACKNOWLEDGMENT

The work described in this paper is fully supported by two grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CUHK4205/04E and Project No. CUHK4235/04E).

REFERENCES

[1] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, 1997.

[2] N. V. Chawla, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. Learning ensembles from bites: A scalable and accurate approach. *Journal of Machine Learning Research*, 5:421–451, 2004.

[3] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(10):974–989, 1999.

[4] I. A. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):757–763, 1997.

[5] B. Fasel and J. Luettin. Automatic facial expression analysis: a survey. *Pattern Recognition*, 36(1):259–275, 2003.

[6] L.-L. Huang, A. Shimizu, and H. Kobatake. Classification-based face detection using gabor filter features. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*, pages 397–402. IEEE Computer Society, 2004.

[7] V. Kyrki, J.-K. Kamarainen, and H. Kälviäinen. Simple gabor feature space for invariant object recognition. *Pattern Recognition Letters*, 25(3):311–318, 2004.

[8] T. S. Lee. Image representation using 2d gabor wavelets. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):959–971, 1996.

[9] S. Z. Li and Z. Zhang. Floatboost learning and statistical face detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1112–1123, 2004.

[10] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–476, 2002.

[11] M. J. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In *Third International Conference on Face & Gesture Recognition (FG '98), April 14-16, 1998, Nara, Japan*, pages 200–205. IEEE Computer Society, 1998.

[12] A. Martinez and R. Benavente. The ar face database. Technical report, June 1998. CVC Technical Report #24.

[13] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1424–1445, 2000.

[14] R. E. Schapire. A brief introduction to boosting. In *IJCAI*, pages 1401–1406, 1999.

[15] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[16] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[17] L. Shen, L. Bai, D. Bardsley, and Y. Wang. Gabor feature selection for face recognition using improved adaboost learning. In S. Z. Li, Z. Sun, T. Tan, S. Pankanti, G. Chollet, and D. Zhang, editors, *IWBRS*, volume 3781 of *Lecture Notes in Computer Science*, pages 39–49. Springer, 2005.

[18] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR (1)*, pages 511–518. IEEE Computer Society, 2001.

[19] X. Wang and X. Tang. Random sampling lda for face recognition. In *CVPR (2)*, pages 259–265, 2004.

[20] Y. Wang, H. Ai, B. Wu, and C. Huang. Real time facial expression recognition with adaboost. In *ICPR (3)*, pages 926–929, 2004.

[21] Y. Yacoob and L. S. Davis. Recognizing human facial expressions from long image sequences using optical flow. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(6):636–642, 1996.

[22] P. Yang, S. Shan, W. Gao, S. Z. Li, and D. Zhang. Face recognition using ada-boosted gabor features. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004), May 17-19, 2004, Seoul, Korea*, pages 356–361. IEEE Computer Society, 2004.

[23] Z. Zhang, M. J. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In *Third International Conference on Face & Gesture Recognition (FG '98), April 14-16, 1998, Nara, Japan*, pages 454–461. IEEE Computer Society, 1998.

[24] Y. Zhu, L. C. D. Silva, and C. C. Ko. Using moment invariants and hmm in facial expression recognition. *Pattern Recognition Letters*, 23(1-3):83–91, 2002.