Feature Synthesis Using Genetic Programming for Face Expression Recognition

Bir Bhanu, Jiangang Yu, Xuejun Tan, and Yingqiang Lin

Center for research in intelligent systems
University of California, Riverside CA 92521-0425, USA
{bhanu, jyu, xtan, yqlin}@cris.ucr.edu

Abstract. In this paper a novel genetically-inspired learning method is proposed for face expression recognition (FER) in visible images. Unlike current research for FER that generally uses visually meaningful feature, we proposed a Genetic Programming based technique, which learns to discover composite operators and features that are evolved from combinations of primitive image processing operations. In this approach, the output of the learned composite operator is a feature vector that is used for FER. The experimental results show that our approach can find good composite operators to effectively extract useful features.

1 Introduction

Automatic face expression recognition (FER) is desirable for a variety of applications such as human-computer interaction, human behavior understanding, perceptual user interface, and interactive computer games; hence it is not surprising that automatic facial information processing is an important and highly active subfield of computer vision and pattern recognition researches [1]. In an automatic FER system, face detection or localization in a cluttered scene is usually considered the first step. Next, relevant features from the face must be extracted, and finally the expression can be classified based on the extracted features. Unlike face recognition, FER focuses on how to discern the same expressions from different individuals. Since different people may show the same expression differently, FER problem is more challenging.

People classify FER problem into two categories depending on whether an image sequence is the input or a single image is the input. For image sequence, the dynamic characteristics of expressions are analyzed. Approaches based on static difference are focused on distinguishing the face expressions from a single given image. A review of different approaches for face expression recognition can be found in [2]. In this paper, we discuss FER from static images.

Facial feature extraction attempts to find the most appropriate representation of the face images for recognition. There are mainly two approaches: holistic template matching systems and geometric feature-based systems [3]. In holistic system, after the face image is processed as a whole a template can be acquired as a pixel image or

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 896–907, 2004.

a feature vector. Padgett and Cottrell [4] used seven pixel blocks from feature regions to represent expressions. In geometric feature-based systems, major face components and/or feature points are detected in a face image. The distances between feature points and the relative sizes of the major face components are computed to form a feature vector. The feature points can also form a geometric graph representation of the faces. Feature-based techniques are usually computationally more expensive than template-based techniques, but are more robust to variation in scales, size, head orientation, and location of the face in an image.

2 Related Work and Motivation

2.1 Related Work

As compared to face recognition, there is relatively a small amount of research on facial expression recognition. Previous work on automatic facial expression includes studies using representations based on optical flow, principal components analysis and physically-based models. Yacoob and Davis [5] use the inter-frame motion of edges extracted in the area of the mouth, nose, eyes, and eyebrows. Bartlett et al. [6] use the combination of optical flow and principal components obtained from image differences. Lyons et al. [7] [8] and Zhang et al. [9] [10] use Gabor wavelet coefficients to code face expressions. In their work, they extract a set of geometric facial points on the facial expression images, and then they used multi-scale and multi-orientation Gabor wavelets to filter the images and extract the Gabor wavelet coefficients at the chosen facial points. Similarly, Wiskott et at. [11] use a labeled graph, based on a Gabor wavelet transform, to represent facial expression images. They perform face recognition through elastic graph matching.

2.2 Motivation

Facial feature extraction is the key step in facial expression recognition. For conventional methods, human experts design an approach to detect potential feature in images depending on their knowledge and experience. This approach can often be dissected into some primitive operations on the original image or a set of related feature images obtained from the original one. The experts figure out a smart way to achieve good facial feature representations by combining these primitive operations. The task of finding good composite features is equivalent to finding good points in the composite feature space. The final combination of the primitive operators is called composite operators. It is obvious that human experts can only try some limited number of conventional combinations and explore a very small portion of the composite operator space since they are biased with their knowledge and lower computation capability [14]. GP, however, may try many unconventional ways of combining primitive operations that may never be imagined by a human expert. Although these unconventional combinations are very difficult, if not impossible, to be explained by domain experts, in some cases, it is these unconventional combinations that yield exceptionally good detection/recognition results. In addition,

the inherent parallelism of GP and the high speed of current computers allow the portion of the search space explored by GP to be much larger than that by human experts, enhancing the probability of finding an effective composite operator. The search performed by GP is not a random search. It is guided by the fitness of composite operators in the population. As the search proceeds, GP gradually shifts the population to the portion of the space containing good composite operators. Tan et al. [15] propose a learning algorithm for fingerprint classification based on GP. To the best of our knowledge, unconventional features discovered by the computer are never used in facial expression classification.

3 Technical Approach

3.1 Gabor Filter Bank

The Gabor representation has been shown to be optimal in the sense of minimizing the joint two-dimensional uncertainty in space and frequency [12]. The Gabor filters can be considered as orientation and scale tunable edge and line detectors. The general form of a 2-D Gabor function is given as:

$$g(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^{\,2}}+\frac{y^2}{\sigma_y^2}\right)+2\pi jWx\right] \tag{1}$$

And its Fourier transform, $G(u,v)$, can be written as:

$$G(\mu,\nu)=\exp\{-\frac{1}{2}[\frac{(\mu-W)^2}{\sigma_u^2}+\frac{\nu^2}{\sigma_v^2}]\} \tag{2}$$

Where (x,y) is the spatial centroid of the elliptical Gaussian window. W is the frequency of a sinusoidal plane wave along the $x-$axis, and σ_x,σ_y are the space constants of the Gaussian envelop along the x and y axes, respectively. u,v are the frequency components in x and y direction, respectively. $\sigma_u=1/2\pi\sigma_X$ and $\sigma_v=1/2\pi\sigma_y$. Gabor function form a complete but nonorthogonal basis set. Expanding a signal using this basis provides a localized frequency description. Let $g(x,y)$ be the mother Gabor wavelet, then filters with multi-orientation can be obtained by a rigid rotation of $g(x,y)$ through the generating function:

$$g(x,y)=a^{-m}G(x',y'),a>1 \tag{3}$$

Where

$$x'=a^{-m}(x\cos\theta+y\sin\theta),\ \text{and}\ y'=a^{-m}(-x\sin\theta+y\cos\theta) \tag{4}$$

And $\theta = n\pi / K$, θ is the rotation angle and K is the total number of orientations. We designed the Gabor filter bank with the following parameters:

$$a = \left(U_h / U_l\right)^{1/S-1}, \sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2\ln 2}} \tag{5}$$

$$\sigma_v = \tan\left(\pi / 2K\right)\left[W - \frac{(2\ln 2)\sigma_u^2}{W}\right]\left[2\ln 2 - \frac{(2\ln 2)^2 \sigma_u^2}{W^2}\right]^{-\frac{1}{2}} \tag{6}$$

Where $W = a^m U_l$ and $m = 0,1,2,...,S-1$. We define W with the scale factor a^m is to ensure the energy is independent on m. U_h, U_l denote the lower and upper center frequencies of interest, respectively. $n = 0,1,2,...,K-1$. m and n are the indices of scale and orientation, respectively. K is the number of orientations and S is the number of scales. In order to eliminate sensitivity of the filter response to absolute intensity values, the real components of the 2D Gabor filters are biased by adding a constant to make them zero mean. The design strategy is to ensure that the half-peak magnitude support of the filter responses in the frequency spectrum touch each other as shown in Fig.1.

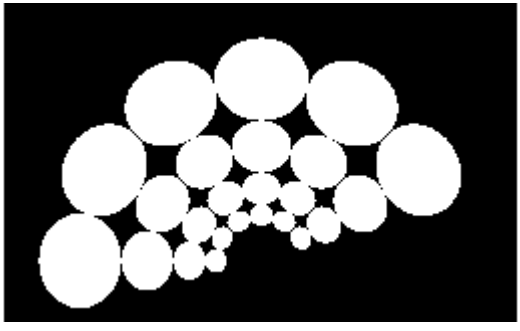

Fig. 1. The filter set in the frequency domain indicates the half-peak magnitude.

3.2 Design Considerations

Figure 2 shows the block diagram of our approach. In our approach, individuals are composite operators represented by binary trees with primitive operators as internal nodes and primitive features as leaf nodes. During the training, GP runs on primitive features generated from the raw facial expression images to generate composite operators. Feature vectors are generated by the learned composite operators, which are used for FER classification. We used Bayesian classifier for classification. During training, fitness value is computed according to the classification result and is monitored during evolution. During testing, the learned best composite operator is

applied directly to generate feature vectors. Since the parameters of Bayesian classifier are determined by the feature vectors from training, the classifier as well the composite operators are learned by GP. Not that, in our approach, we don't need to find any reference point on the image.

- **The Set of Terminals:** The set of terminals used in this paper are called primitive features which is generated from the raw facial expression images filtered by Gabor filter bank at 4 scales and 6 orientations. These 24 images are input to composite operators. For simplicity, we resize the filtered images to 32x32. GP determines which operators are applied on primitive features and how to combine the primitive operators. Figure 3 shows an example of primitive features filtered by Gabor filter bank.
- **The Set of Primitive operators:** A primitive operator takes one or two input images and performs a primitive operation on them and outputs a resultant image and/or feature vectors. In our approach, we designed two kinds of primitive operators: computational operators and feature generation operators. For computational operators, the output are images. For feature generation operators, however, the resultant output includes an image and a real number or vector. The real number or the vectors are the elements of the feature vector, which is used for classification. Table 1 shows different primitive operators and explains the meaning of each one [15].

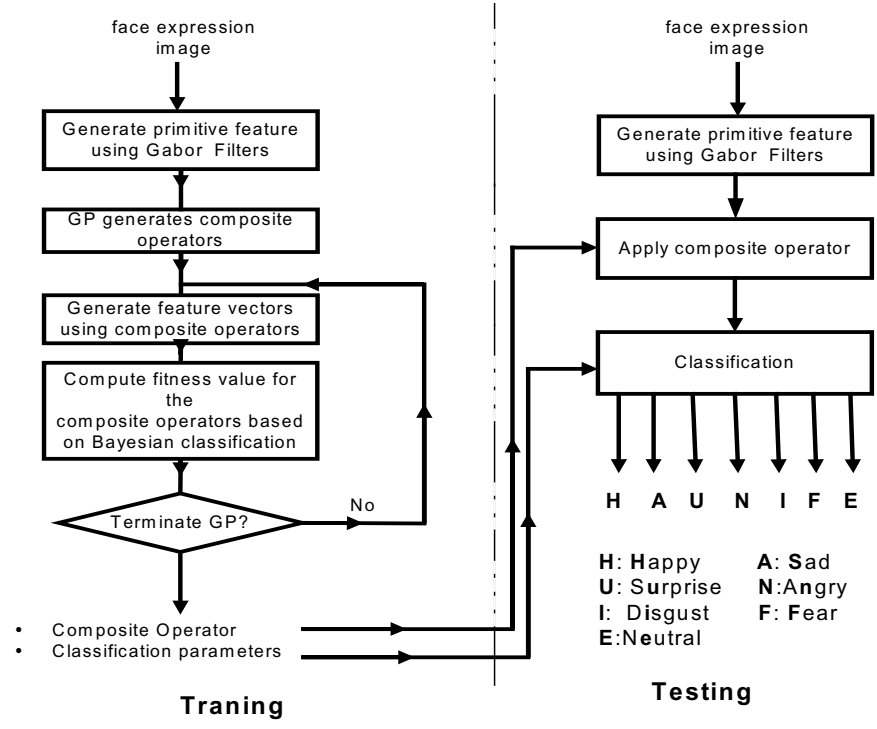

Fig. 2. Block Diagram of our approach

- **The Fitness Value:** During training, at every generation for each composite operator run by GP, we compute the feature vector and estimate the probability distribution function (PDF) for each class using all the available feature vectors for this class. For simplicity, we assume feature vectors for each class have Gaussian distribution. Then, for each class ω_i, we compute the mean and the covariance of this class:

$$u_i = \sum_{j=1,\ldots,N} x_j, \ \Sigma_i = \frac{1}{N}\sum_j \left(x_j - u_i\right)\left(x_j - u_i\right)^T \tag{7}$$

Thus, the PDF of ω_i can be written as:

$$p\left(x \mid \omega_i\right) = \frac{1}{\left(2\pi\right)^{n/2}\left|\Sigma_i\right|^{1/2}} \exp\left(-\frac{1}{2}\left(x - u_i\right)^T \Sigma_i^{-1}\left(x - u_i\right)\right) \tag{8}$$

According to Bayesian theory, we have

$$p\left(\omega_i \mid x\right) = \frac{p\left(x \mid \omega_i\right) p\left(\omega_i\right)}{p\left(x\right)} \tag{9}$$

We assign $x \in \omega_k$

$$iff \ p\left(x \mid \omega_k\right) \cdot p\left(\omega_k\right) = \max_{i=1,2,3,4,5,6,7}\left(p\left(x \mid \omega_i\right) \cdot p\left(\omega_i\right)\right) \tag{10}$$

where n is the size of the feature vector, i is the class and x is a feature vector for the class.
In the classification, the Percentage of Correct Classification (PCC) is used as the fitness value of the composite operator.

$$\text{Fitness Value} = \frac{n_c}{n_s} \times 100\% \tag{11}$$

where n_c is the number of correctly classified facial expressions by GP and n_s is the size of training set.

- **Parameters and Termination conditions:** The parameters to control the run of GP is important. In our approach, we select the maximum size of composite operator 200, population size 100, number of generation 150, crossover rate 0.6, length of maximum feature vector 35, the fitness threshold 0.98, and the mutation rate 0.05. GP stops whenever it finishes the pre-specified number of generations or whenever the best composite operator in the population has fitness value greater than the fitness threshold.

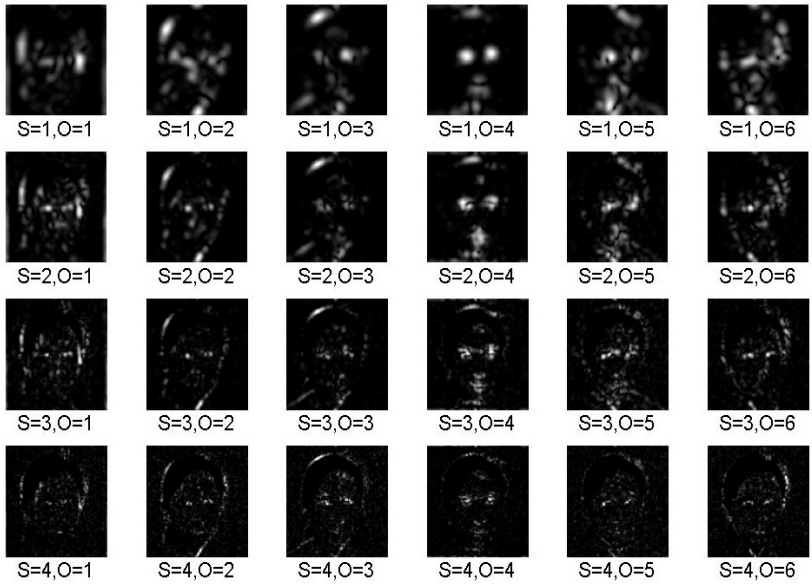

Fig. 3. An example of the primitive feature. S means Scale and O represents Orientation

4 Experimental Results

4.1 Database

The database [7] we use for our experiments contains 213 images of female facial expressions. Each person has two to four images for each of seven expressions: neutral, happy, sad, surprise, anger, disgust, and fear. Each image size is 256x256 pixels. A few examples are shown in Figure 4. This database was also used in [9] [10] [13].

4.2 Results

We perform the experiments 5 times and choose the best result as the learned composite operator. In order to deal with overfitting of this small sample size database, we use 1-fold validation. We divide the database into training set and test set, from which two-third of training data are used to generate composite operators and the remaining one-third is used for evaluating on the tree. Figure 5 shows the fitness values based on the number of the generations in GP. Figure 6 shows the best composite operator for the 7-class classification in LISP notation, which represents the structure of the tree. For 7-class classification, the composite operator's size is 112, out of which there are 19 feature generation operators and the length of the feature vector is 25. Obviously, these composite operators are not easy to be constructed by humans.

Table 1. The primitive operators in our approach

	Primitive Operator	Meaning
Computation Operators	ADD_OP, SUB_OP, MUL_OP and DIV_OP	A+B, A–B, A×B and A/B. If the pixel in B has value 0, the corresponding pixel in A/B takes the maximum pixel value in A.
	MAX2_OP and MIN2_OP	Max (A, B) and min (A, B)
	ADD_CONST_OP, SUB_CONST_OP, MUL_CONST_OP and DIV_CONST_OP	A+c, A-c, A×c and A/c
	SQRT_OP and LOG_OP	$sign\ (A) \times \sqrt{\lvert A \rvert}$ and $sign\ (A) \times \log(\lvert A \rvert)$.
	MAX_OP, MIN_OP, MED_OP, MEAN_OP and STD_OP	Max (A), min (A), med (A), mean (A) and std (A), replace the pixel value by the maximum, minimum, median, mean or standard deviation in a 3×3 block
	BINARY_ZERO_OP and BINARY_MEAN_OP	threshold/binarize A by zero or mean of A
	NEGATIVE_OP	-A
	LEFT_OP, RIGHT_OP, UP_OP and DOWN_OP	Left (A), right (A), up (A) and down (A). Move A to the left, right, up or down by 1 pixel. The border is padded by zeros
	HF_DERIVATIVE_OP and VF_DERIVATIVE_OP	HF (A) and VF (A). Sobel filters along horizontal and vertical directions
Feature Generation Operators	SPE_MAX_OP, SPE_MIN_OP, SPE_MEAN_OP, SPE_ABS_MEAN_OP and SPE_STD_OP	Max2 (A), min2 (A), mean2 (A), mean2 ($\lvert A \rvert$) and std2 (A)
	SPE_U3_OP and SPE_U4_OP	$\mu_3(A)$ and $\mu_4(A)$. Skewness and kurtosis of the histogram of A
	SPE_CENTER_MOMENT11_OP	$\mu_{11}(A)$. First order central moments of A
	SPE_ENTROPY_OP	H (A). Entropy of A
	SPE_MEAN_VECTOR_OP and SPE_STD_VECTOR_OP	mean_vector(A) and std_vector(A). A vector contains the mean or standard deviation value of each row/column of A

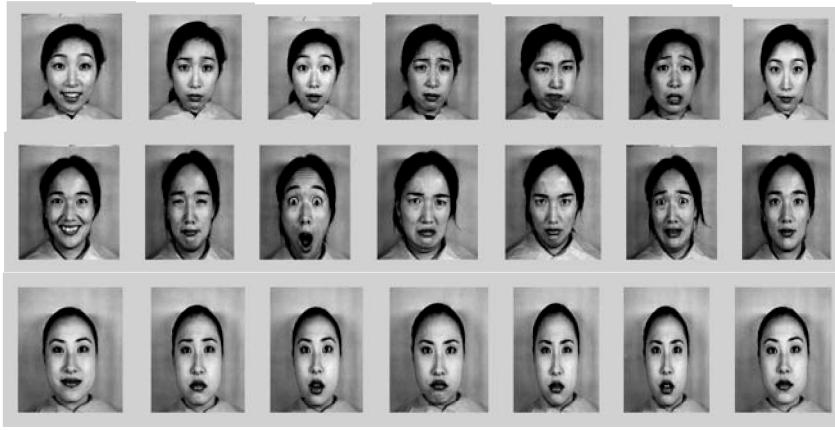

Fig. 4. A few example of the database

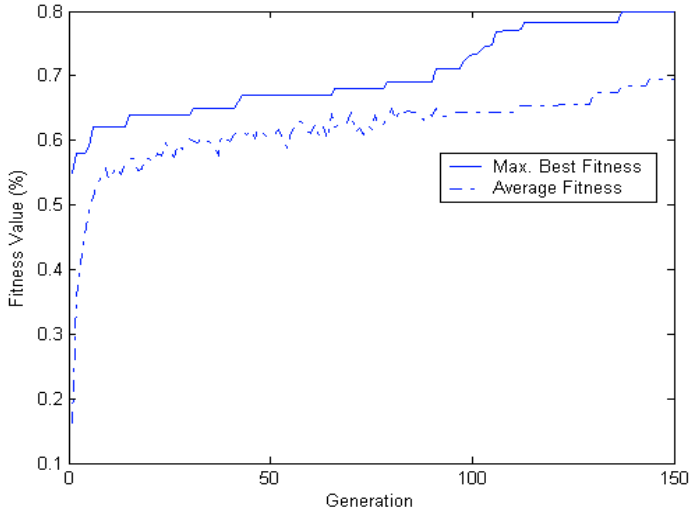

Fig. 5. Fitness value based on the number of generations

4.2.1 Comparison with Previous Approach

In [13], Guo *et al*. used a Bayesian classifier. We compare the result in Table 2.

In Table 2, Bayes All means the Bayes classifier without feature selection, Bayes FS means Bayes with pairwise-greedy feature selection, AdaBoost. From the table, we can find that GP has better performance in both accuracy and length of the feature vector implemented or obtained from [13]. In our approach, we did not do any pre-processing of the raw image. The input image is the raw facial expression image. However, the other methods in Table 2 selected the fiducial points on a face image manully and generated the Gabor coefficients as feature vector.

Table 2. Comparision of the recognition accuracy

	Bayes All [13]	Bayes FS [13]	AdaBoost [13]	GP This paper
Accuracy	63.3%	71.0%	71.9%	72.0%
# Features	612	60	80	25

```
(MEAN_OP(SPE_ABS_MEAN_OP(UP_OP(STDV_OP(SPE_MIN_OP(UP_OP(
ADD_OP(MAX_OP(HF_DERIVATIVE_OP(BINARY_MEAN_OP(HF_DERIV
ATIVE_OP(STDV_OP(MIN2_OP(DIV_OP(INPUT_OP)INPUT_OP))RIGHT_O
P(SPE_MOMENT10_OP(DIV_OP(DIV_CONST_OP(SPE_CENTER_MOMEN
T11_OP(STDV_OP)(ADD_OP(LOG_OP(BINARY_ZERO_OP(ADD_CONST_
OP(DOWN_OP(MAX2_OP(MUL_OP(SPE_MOMENT01_OP(HF_DERIVATI
VE_OP(MUL_OP(BINARY_MEAN_OP(HF_DERIVATIVE_OP(INPUT_OP))I
NPUT_OP))(SPE_MOMENT10_OP)(SPE_MOMENT01_OP(SPE_STD_OP(AD
D_CONST_OP(SPE_MOMENT01_OP(MAX2_OP(SPE_CENTER_MOMENT1
1_OP(MIN_OP(SPE_CENTER_MOMENT11_OP(ADD_OP(VF_DERIVATIVE
_OP(SPE_ABS_MEAN_OP(BINARY_MEAN_OP(MED_OP(INPUT_OP)))))(U
P_OP(MIN2_OP(SUB_OP(INPUT_OP(MAX2_OP(SUB_CONST_OP(INPUT)
OP))(INPUT_OP)))(INPUT_OP)))))))(HF_DERIVATIVE_OP(INPUT_OP))))))))
))))(SPE_STD_OP(MED_OP(SUB_OP(ADD_CONST_OP(INPUT_OP))(LOG_
OP(INPUT_OP)))))))))(HF_DERIVATIVE_OP(HF_DERIVATIVE_OP(MEAN_
OP(LOG_OP(MEAN_OP(SPE_STD_OP(MAX2_OP(BINARY_ZERO_OP(RIG
HT_OP(MIN2_OP(SPE_ABS_MEAN_OP(MED_OP(UP_OP(ADD_OP(INPUT
_OP)(INPUT_OP)))))SPE_ABS_MEAN_OP(MEAN_OP(DIV_OP(INPUT_OP)(
MAX_OP(LOG_OP(DOWN_OP(SPE_MOMENT01_OP(ADD_OP(INPUT_OP)
(INPUT_OP))))))))))(INPUT_OP))))))))))))))))(ADD_CONST_OP(SPE_CENT
ER_MOMENT11_OP(MIN2_OP(UP_OP(SPE_MAX_OP(MIN_OP(MIN_OP(M
EAN_OP(ADD_CONST_OP(INPUT_OP)))))))(BINARY_MEAN_OP(ADD_OP
(INPUT_OP)(INPUT_OP)))))))))))))
```

Fig. 6. Learned Composite Operators in lisp notation

4.3 Discussion

In [9] [10] [13], authors have used SVM, LDA and Neural Network for facial expression recognition. We found SVM and LDA have higher recognition accuracy (about 90%),while in [13] and our GP the perfomances with using Bayesian classifier are much less than that. In the following we present an analysis for this difference.

For a Bayesian classifier, we need to estimate the probability distribution function (PDF) $p(x|\omega_i)$ for each class ω_i. In the small sample case, it is hard that the estimated PDF can accurately approximate the underlying unknown densities. Thus the estimated probability distribution may be biased far away from the real one. As a consequence, low recognition accuracy can be expected. In our approach, the simplified Bayesian theory assumes independent and Gaussian distribution, which simplified the problem of class density estimation. However, in our experiments, we found the problem of overfitting is serious. However, for the margin-based discrimination, one doesn't need to estimate the underlying distribution, thus the recognition accuracy could be higher. For the future, we plan to perform experiments with SVM based classifier and with GP generated features.

5 Conclusions

In this paper, we proposed a learning paradigm for facial expression recognition based on GP. Compared with the previous work with the same classifier, our experimental results show that GP can find good composite operators. Our GP-based algorithm is effective in extracting feature vectors for classification, which are beyond human's imagination. In our approach, we don't need to do any pre-processing of the raw image and we don't need to find any reference points on the face.

References

1. J. Daugman, "Face and Gesture Recognition: An Overview," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 675-676, July 1997
2. M. Pantic and L. J. M. Rothkrantz, Automatic analysis of facial expressions: The state of the art, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(12), 1424-1445, 2000
3. W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, Face recognition: A literature survey. *CVL Technical Report, University of Maryland, October 2000*
4. C. Padgett and G. Cottrell, Identifying emotion in static images, Proc. 2nd Joint Symp. On *Neural Computation*, vol. 5, 91-101, 1997
5. Y. Yacoob and L. Davis. Recognizing facial expressions by spatio-temporal analysis. *In Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 747-749, Oct. 1994.
6. M. Bartlett, P. Viola, T. Sejnowski, L. Larsen, J. Hager, and P. Ekman. Classifying facial action. In D. Touretzky, M. Mozer, and M. Hasselmo, editors, *Advances in Neural Information Processing Systems* 8. MIT Press, Cambridge, MA, 1996

7. M. J. lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, Coding facial expressions with Gabor wavelets. In Proc. *Third IEEE Int. Conf. Automatic Face and Gesture Recognition*, 200-205, 1998
8. M. J. Lyons, J. Budynek, and S. Akamatsu, Automatic Classification of single facial images, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(12), 1357-1362,1999
9. Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, Comparison between geometry-based and Gabor-wavelets-based facial expression recognition using multi-layer perceptron, *Proc. Int'l Conf. Automatic Face and Gesture Recognition*, pp. 454-459, 1998
10. Z. Zhang, Feature-based facial expression recognition: Sensitivity analysis and experiments with a multi-layer perceptron, *Journal of Pattern Recognition and Artificial Intelligence*, 13(6): 893-911,1999
11. L. Wiskott, J. M. Fellous, N. Kruger, and C. Von der Malsburg. Face recognition by bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 19(7): 775-779,July 1997
12. B. S. Manjunath and W. Y. Ma, Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(8), 837-842, 1996
13. G. D. Guo and C. R. Dyer, Simultaneous Feature Selection and Classifier Training via Linear Programming: A Case Study for Face Expression Recognition, *IEEE Conference on Computer Vision and Pattern Recognition,* I, 346-352, June, 2003
14. B. Bhanu and Y. Lin, Learning Composite Operators for Object Detection, *GECCO 2002*, pp. 1003-1010, 2002
15. X. Tan, B. Bhanu, and Y. Lin, Learning Features for Fingerprint Classification, *Audio- and Video-based Biometric Person Authentication 2003*, pp. 319-326, 2003