Human Face Authentication With Support Vector Machines

Varnika Bhardwaj
Gujrat Technological University

Soumik Das
Amity University, Noida

Rana Majumdar
Amity University, Noida

Abstract— This paper describes an attempt to build a component-based face detector using support vector machine classifiers. We present current results and outline plans for future work required to achieve sufficient speed and accuracy to use SVM classifiers in a face recognition system. We take a straightforward approach in implementing our own SVM classifier with a Gaussian kernel that detects face in grayscale images, a first step towards a component-based face detector. Details on design of an iterative bootstrapping process are provided, and we show which training parameter values tend to give best results. Conclusions drawn from our work up to date are consistent with previous research and problems encountered are to be expected by anyone building an object detection system - SVM classifiers with large numbers of support vectors are slow and accuracy depends largely on the quality and variety of training data.

Key Words: HCI, FER, SVM

I. INTRODUCTION

Face can express emotion sooner than people verbalize or even realize their feelings. For human beings, facial expressions reveal a person's emotion and provide important communicative cues during social interaction. This implies that facial expressions form a major modality. Therefore, the research of facial expression recognition has become increasingly popular in computer vision or robotics area. Face identification and recognition has lead to the development of different algorithms for various applications such as automated access control, surveillance, image retrieval etc. Generally pattern recognition problem rely upon the features inherent in the pattern for efficient solution. The challenges associated with face detection and recognition problem are pose, occlusion, expression, varying lighting conditions etc. Facial expression analysis has wide range of applications in areas such as Human Computer Interaction (HCI), Psychological area, Image understanding, Face animation etc. Humans interact with each other both verbally and non-verbally. In an automatic facial expression recognition (FER) system, there are three major components for achieving this goal. First, a face is detected and localized. Facial feature information is extracted from the detected face region. Finally, the facial expression category is classified based on the extracted features. There are many techniques to detect face region from images. In skin color detection technique is used to identify the skin pixels from which face regions are detected [1]. Contour techniques are also used to detect and recognize faces [2]. Geometric based and Appearance based methods are examples of feature extraction methods. In Geometric based method, the shape and location of facial features are extracted as feature vectors. In Appearance based method, image filters are applied either to whole face or to specific regions of facial image to extract facial features. Facial expressions can be classified using a neural network or SVM classifier [3].

II. SUPPORT VECTOR MACHINES

A. Linear case

Consider a set of I vectors {xi}, xi∈Rn , 1 ≤ i ≤ I, representing input samples and set of labels {yi}, yi∈{±1}, that divide input samples into two classes, positive and negative. If the two classes are linearly separable, there exists a separating hyper plane (w, b) defining the function,

$$f(x) =< w \cdot x > +b, \quad (1)$$

and sgn(f(x)) shows on which side of the hyper plane x rests, in other words - the class of x. Vector w of the separating hyper plane can be expressed as a linear combination of xi (often called a dual representation of w) with weights αi :

$$w =\sum \alpha i y i x i , \quad (2)$$

The dual representation of the decision function f(x) is then:

$$f(x)=\sum \alpha i y i < xi \cdot x > +b, \quad (3)$$

Training a linear SVM [4] means finding the embedding strengths {αi} and offset b such that hyper plane (w, b) separates positive samples from negatives ones with a maximal margin. Notice that not all input vectors {xi} might be used in the dual representation of w; those vectors xi that have weight αi > 0 and form w are called support vectors.

B. Non-linear case

In real-life problems it is rarely the case that positive and negative samples are linearly separable. Non-linear support vector classifiers map input space X into a feature space F via a usually non-linear map: X → F, x → (x) and solve the linear separation problem in the feature space by finding weights αi of the dual expression of the separating hyper plane's vector w:

$$w =\sum \alpha i y i \ (xi), \quad (4)$$

while the decision function f(x) takes the form

$$f(x) =\sum \alpha i y i < (xi) . (x) > + b, \quad (5)$$

Usually F is a high-dimensional space where images of training samples are highly separable, but working directly in such a space would be computationally expensive. However we can choose a space F which is induced by kernel K, defined by a kernel function K(x, y) that computes the dot product in F, K(x, y) =< (x). (y) >. The decision function [5] can then be computed by just using the kernel function and it can also be shown that finding the maximum margin separating hyper plane is equivalent to solving the following optimization problem

$$\max \left[\sum \alpha i - \tfrac{1}{2} \sum \alpha i \alpha j y i y j \ K(xi , xj)\right],$$

$$\text{subject to } 0 \leq \alpha i \leq C , 1 \leq i \leq 1 , \sum \alpha i y i = 0, \quad (6)$$

where positive C is a parameter showing the trade-off between margin maximization and training error minimization [6]. Thus knowing the kernel function K we avoid working directly in feature space F. After solving [7], offset b can be chosen so that the margins between the hyper plane and the two classes of sample images are equal. We then have our decision function

$$\text{sgn} (f(x)) = \text{sgn}\left[\sum \alpha i y i \ K (xi , x) + b\right], \quad (7)$$

Commonly used kernels include polynomial kernels $K(x, y) = (x + y) d$ and the Gaussian kernel $K(x, y) = \exp(-\|x-y\|2 / \sigma 2)$. In our implementation we use the Gaussian kernel; however one of the interesting points for further research is approaches for choosing an optimal kernel for the given input data.

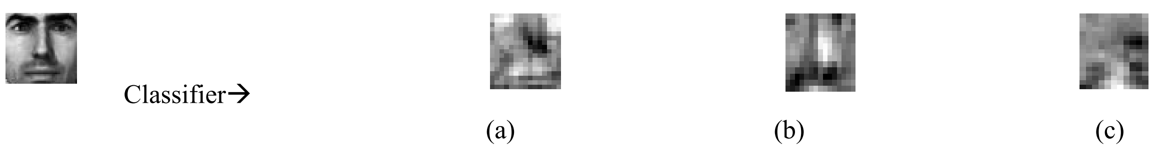
III. COMPONENT BASED APPROACH

The component-based approach is not sensitive to image variations caused by changes in the pose of the face. It does that by independently detecting parts of the face [8]. For small rotations, the changes in the components are relatively small compared to the changes in the whole face pattern. Changes in the 2-D locations of the components due to pose changes are accounted for by a learned, flexible face model.

A. Detection

We implemented a two-level component-based face detector which is described in detail in the paper Face Detection in Still Gray Images [9]. The principles of the system are illustrated in Fig. 1. On the first level, component classifiers independently detected facial components. On the second level, a geometrical configuration classifier performed the final face detection by combining the results of the component classifiers. Given a 58 x 58 window, the maximum continuous outputs of the component classifiers within rectangular search regions around the expected positions of the components were used as inputs to the geometrical configuration classifier. The search regions have been calculated from the mean and standard deviation of the components' locations in the training images. We also provided the geometrical classifier with the precise positions of the detected components relative to the upper left corner of the 58 x 58 window. The 14 facial components used in the detection system are shown in Fig. 2 (a). The shapes and positions of the components have been automatically determined from the training data in order to provide maximum discrimination between face and non-face images; see [5] for details about the algorithm.

First Level: Component Classifier



(a): Output of Eye Classifier
(b): Output of Nose Classifier
(c): Output of Mouth Classifier

Figure 1. Second Level: Detection of Configuration of Components

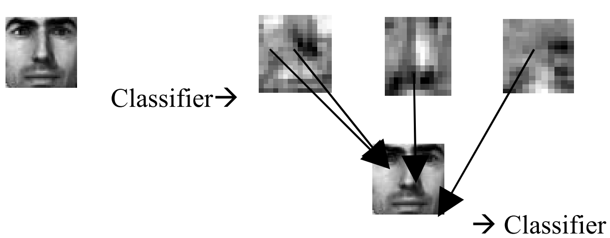
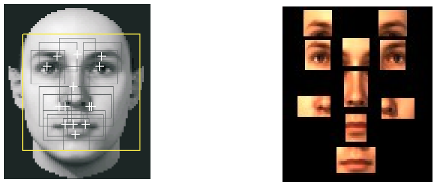


Figure 2. System overview of the component based face detector using four components.

On the first level, windows of the size of the components (solid lined boxes) are shifted over the face image and classified by the component classifiers. On the second level, the maximum outputs of the component classifiers within predefined search regions (dotted lined boxes) and the positions of the detected components are fed into the geometrical configuration classifier.



(a) (b)

Figure 2. (a) shows the 14 components of our face detector. The centers of the components are marked by a white cross. The 10 components that were used for face recognition are shown in (b).

B. Recognition

To train the face recognizer we first ran the componentbased detector over each image in the training set and extracted the components. From the 14 original we kept 10 for face recognition, removing those that either contained few gray value structures (e.g. cheeks) or strongly overlapped with other components. The 10 selected components are shown in Fig. 2 (b). Examples of the component-based face detector applied to images of the training set are shown in Fig. 3. To generate the input to our face recognition classifier we normalized each of the components in size and combined their gray values into a single feature vector. As for the first global system we used a one-vs-all approach with a linear SVM for every person in the database. [10]

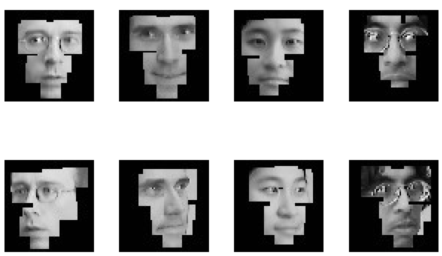


Figure 3. Examples of component-based face detection. Shown are face parts covered by the 10 components that were used for face recognition.

IV. GABOR FILTERS

Features based on Gabor filters have been used in image processing due to their powerful properties. Gabor kernels are characterized as localized, orientation selective, and frequency selective. A family of Gabor kernel is the product of a Gaussian envelope and a plane wave. A 2D Gabor filter is expressed as a Gaussian modulated sinusoid in the spatial domain and as shifted Gaussian in the frequency domain. The Gabor wavelet representation of images allows description of spatial frequency structure in the image while preserving information about spatial relations.

$$P_K(\mu) = k^2/\sigma^2 \exp(-k^2\mu^2/2\sigma^2)\,(\exp(ik\mu) - \exp(-\sigma^2/2)) \qquad (8)$$

where $\mu = (x, y)$ is the variable in spatial domain and k is the frequency vector which determines the scale and direction of Gabor functions.

$k = (kx , ky) = (kv \cos\theta w , kv \sin\theta w)$ and $kv = (k\ max / fv)$. $v = (0, 1, 2, 3, 4)$ is the discrete set of different frequencies and w= (0, 1, 2...7) is the orientation[11]. The multiplicative factor ($k2/\sigma 2$) ensures that filters tuned to different spatial frequency bands have approximately equal energies. A well designed Gabor filter bank can capture the relevant frequency spectrum in all directions. Phase can be taken as a feature because it contains information about the edge locations and other such details in the image. Amplitude at every pixel can be taken as a feature as it contains some oriented frequency spectrum at every point of the image. Many meaningful features can be extracted using the Gabor filter banks. The response image of Gabor filter can be written as a correlation of input image I(x) with Gabor kernel Pk (x)

$$ak\,(x0) = \iint I\,(x)\, Pk\,(x - x0)\, dx \tag{9}$$

Each pixel is then represented by 40 Gabor features. So for a 64X64 image, the size of transformed image is 64X64X58. So, the feature vector consists of all useful information extracted from different frequencies, orientations and from all locations, and hence is very useful for expression recognition. But, in the practical application, evaluating all 40 filters to convolve the face image is quite time consuming. The real part of Gabor feature vectors with eight orientations and five frequencies is given in Fig.4.

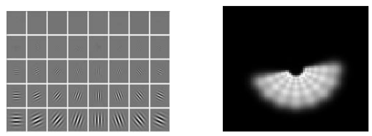


Figure. 4. Gabor filters: Real part of the Gabor kernels at five scales and eight orientations in the spatial and frequency domain.

v. EXPERIMENTAL RESULTS

Two sets of experiments are used to evaluate the method. Test set A is a set of images from the BioID database. This set features a larger variety of illumination, background and face size. It stresses real world conditions. The detection rate obtained for test set A is 96.8% with 6 false detections. Fig. 5 present some of the results obtained for test set A.

Figure. 5. Some detection results on test set A

Test set B is a set from MIT-CMU test sets. As our method addresses detection of frontal and real human faces, those MIT-CMU test images containing large pose-angled faces, line-drawn faces, poker faces, masked faces, or cartoon faces are not included in our experiments. The detection rate achieved is 94.2 % with 8 false detections. Some of the detection results obtained is shown in Fig. 6. The rotated faces are detected by rotating test images to predefined degrees, such as ±5°, ±10°, ±15°, ±20°, and ±25°. Note that because our face detection method is trained using only upright frontal faces, the method cannot detect large pose-angled faces.

The window sliding technique, which scans the pattern at different locations in an image at one-pixel increments horizontally and vertically, is the major reason for the long processing time. The number of enlargement iterations determines the maximum size of the face that can be detected by the system. The user sets this value at runtime. The initial size of search window is 24 × 24 pixels, and for subsequent iterations, the window size is enlarged gradually. It should also be noted that the number of iterations determines the processing time. As the iterations progress, the window size gets bigger, and the number of windows to be examined becomes smaller, thereby reducing the time taken for the iteration when compared with earlier iterations. If the faces to be detected are not expected to be too big, then there is no need to set a value too large for the number of enlargements since that will add to the computation time.

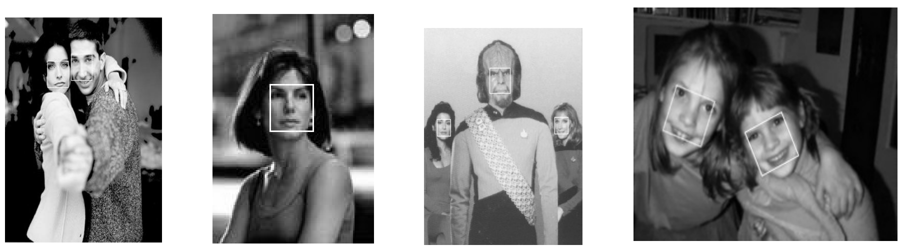


Figure. 6. Some detection results on test set B

SVM has good learning and generalization ability for small sample problems, compared with traditional methods. It shows higher face recognition ability.

Table 1 is the comparison of experiment results of classification of the human face images that are represented by 20 dimension eigenvectors. We have discovered that, compared with traditional methods, such as K-Mean, Euclidean distance, Mahalanobis distance, and neural networks, SVM achieves the highest accuracy. Meanwhile, when the face images are represented by 25 and 30 dimension eigenvectors, SVM has a classification rate of nearly 100%, while the traditional method is 96%.

Classification method	**Accuracy of female section**	**Accuracy of long face section**	**Accuracy of oval face section**
SVM	0.9609	0.9649	0.9745
K-Mean	0.9348	0.9079	0.9562
Euclidean distance	0.9261	0.8965	0.9343*
Mahalanobis distance	0.9261	0.8965	0.9432
Neural networks	0.9410	0.9221	0.9218

Table 1: The comparisons among SVM and traditional methods

The performances of different kernel functions of SVMs are also compared by experiments (Table 2). It is concluded that 2-order polynomial SVM have a better performance when the face image is represented by few eigenvectors, with higher recognition rate and fewer support vectors.

X	Kernel functions of SVM	Accuracy of female section	Accuracy of long face section	Accuracy of oval face section
30	RBF	0.9826/39.3	0.9693*/39.8	0.9964*/38.8
	Polynomial	0.9913*/40.5	0.9605/48.9	0.9964*/44.2
	S-function	0.9826/41.2	0.9565/45.6	0.9964*/44.3
25	RBF	0.987/37.8	0.9693*/38.5	0.9927/38.7
	Polynomial	0.9957*/36.9	0.9561/45.7	0.9964*/39.7
	S-function	0.9826/35.4	0.9561/41	0.9927/43.8
20	RBF	0.9609/37.0	0.9649*/37.3	0.9745/37
	Polynomial	0.9826*/33	0.9430/39.9	0.9891*/36.7
	S-function	0.9652/38.2	0.9386/39.7	0.9708/38.2
15	RBF	0.9348/35.2	0.9167/36.8	0.9635/31.0
	Polynomial	0.9870*/29.6	0.9254*/33.8	0.9672*/27.5
	S-function	0.9652/34.3	0.9167/37.1	0.9518/34.2
10	RBF	0.8826/35.0	0.8509/35.9	0.8940/36.8
	Polynomial	0.9609*/24.4	0.8728*/28.4	0.9015*/27.8
	S-function	0.8565/35.1	0.8486/33.3	0.9708/35.7
				X= Number of eigen vectors

Table 2: Performance comparisons among different kernel function SVMs

VI. CONCLUSIONS AND FURTHER RESEARCH

In this paper, we have shown that the three rectangle features are a good choice for face candidate selection. The features embody the characteristics of eye region, i.e. intensity and symmetry. For face candidate extraction, a rectangular window is scanned over the entire input image and three rectangle features are computed to get eye-pair-like region and then the corresponding square image patches are considered to be face candidate. Finally, the normalized image is sent to SVM for face verification. The component-based system detected and extracted a set of 10 facial components and arranged them in a single feature vector that was classified by linear SVMs.

This method is only applicable to frontal face images; other ways of capturing the profile face models have to be incorporated to make the approach useful for profile face detection. Experimental results show that this face detection method can detect most of the faces, but some of the images in low quality images are missed.

We have showed that using facial components instead of the whole face pattern as input features significantly simplifies the task of face recognition.

The performance of our system is as follows:

- Human face detection accuracy: 97.2% under controlled lighting conditions.
- Human face (70 persons) recognition accuracy: 96.5% (with 20 eigenvectors) and 98.3% (with 30 eigen vectors).

Future research will focus on looking for more effective representation to divide face and non-face pattern, detecting face more accurately. Further work will also focus on the robustness of the system, to develop an algorithm that can detect a face and recognize it in different lighting conditions and against complex backgrounds.

VII. REFERENCES

[1] Q. Wang, W.K. Yang, H. Wang et al., “Face Detection using binary template matching and SVM,” LNAI 4099, 2006, pp. 1237 – 1241.

[2] Analysis of Facial Expression using Gabor and SVM by Praseeda Lekshmi.V., Dr.M.Sasikumar

[3] C. J. C. Burges. Simplified support vector decision rules. In International Conference on Machine Learning, pages 71–77, 1996.

[4] V. Vapnik. The Nature of Statistical Learning Theory. Springer, N.Y., 1995.

[5] J. C. Platt. Using analytic qp and sparseness to speed training of support vector machines. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, Advances in Neural Information Processing Systems, volume 11, Cambridge, MA, 1999. MIT Press.

[6] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In COLT ’92: Proceedings of the fifth annual workshop on Computational learning theory, pages 144–152, New York, NY, USA, 1992. ACM Press.

[7] S. Romdhani, P. Torr, and B. Sch¨olkopf. Efficient face detection by a cascaded support-vector machine expansion. Royal Society of London Proceedings Series A, 460:3283–3297, November 2004.

[8] B. Heisele, T. Serre, and T. Poggio. A component-based framework for face detection and identification. International Journal of Computer Vision, 74(2):167–181, August 2007.

[9] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. AI Memo 1687, Center for Biological and Computational Learning, MIT, Cambridge, MA, 2000

[10] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: an application to face detection,” In Proc. IEEE Conf. Computer Vision Pattern Recognition. 1997, pp. 130–136.

[11] Gabor-Based RCM Features for Ear Recognition Ali Pour Yazdanpanah and Karim Faez.