Comparisons of Features for Automatic Eye and Mouth Localization

Hakan Cevikalp, Hasan Serhan Yavuz, Rifat Edizkan, Hüseyin Gündüz, Celal Murat Kandemir

Eskisehir Osmangazi University, Electrical and Electronics Engineering, Eskisehir, Turkey.
hakan.cevikalp@gmail.com, {hsyavuz, redizkan, kandemir@ogu.edu.tr}, husamsoft@gmail.com

Abstract— **Localization of the eyes and mouth in face images is very important for accurate classification in automatic face recognition systems. The alignment of unknown face images with templates generally improves the performance of the face recognition system, and this process uses locations of the eyes and mouth. In this work, we compare different features (gray-level values, distance transform features, gradients and local binary patterns) for automatic localization of eyes and mouth. To this end, we use the sliding window approach using the linear and nonlinear support vector machine (SVM) classifiers. We created new frontal face data sets to train and test our algorithms. The experimental results show that the SVM classifier using the Gaussian kernel yields better results than the linear kernel. Among the four feature extraction methods, the performance of the local binary pattern features draws the attention for having better detection rates in both the linear and the nonlinear cases with smaller feature size.**

Keywords – face recognition; support vector machine classifiers; eye detection; mouth detection

I. INTRODUCTION

Face recognition is a biometric method that is widely used in many areas including military, commercial and law enforcement applications. There are many challenging issues for accurate and efficient face recognition: Illumination changes, variations in pose, orientation and scale to name a few. The detection of faces along with the detection of some facial fiducial points such as the eyes and mouth is the first stage of the face recognition system, and this step largely affects the performance of the overall system.

Alignment of an unknown face image with a template is a requirement in most appearance-based methods to improve the classification performance. The face detection algorithms generally do not give orientation and pose information about the detected face image. Therefore, an unknown face must be aligned to the templates using the locations of the facial fiducial points. Facial fiducial point detection methods can be generally classified into four categories: rule-based methods, feature invariant methods, template matching techniques, and appearance-based methods [1-7].

In this study, we use appearance-based methods. In appearance-based methods, there are two important factors that determine the success of localization of the facial fiducial points: the features used for representing the eyes and mouth and the learning algorithm that implements the detection. There are various feature extraction techniques which can be employed for detection. The features must be able to represent images including facial fiducial points in a manner that renders them invariant to intra-class variations in challenging viewing conditions, but at the same time distinguishes the instances of object class of interest (such as the left-right eye or mouth) from the remaining face region. In this study, three more different types of features have been extracted in addition to the gray level intensity values. These are the local binary patterns, the distance transform and the gradient values. As a whole, four categories of features have been used to detect the eyes and mouth regions in the face image. The learning algorithm implementing the detection treats the detection problem as a binary classification problem, namely, separating the facial fiducial points from the remaining face regions. Here we used the SVM classifier for detection as in [8].

In this work, we created a new frontal face database in which face images come from the web and different face recognition databases. Two different kernels have been used in the SVM classifier training: the linear and the Gaussian kernels. After we learn facial fiducial point models during the training phase, the detection is performed by using the sliding window approach. In sliding window approach, each image is densely scanned from the top left to the bottom right with rectangular sliding windows in different scales. For each sliding window, feature vector is extracted first and the feature vector is fed to the classifier which classifies the rectangular window as the facial fiducial point or background. To assess the detection results, we use the PASCAL VOC detection metric [9].

The rest of the paper is organized as follows. In Section II, we give some brief information about the feature extraction techniques used in the representation of the eyes and the mouth regions. Theoretical background about SVM is given in Section III. Experimental results are given in Section IV, and we conclude the paper in Section V.

II. FEATURE EXTRACTION

There are numerous feature extraction techniques which can be used in the facial fiducial point representation. Each feature can supply various performance results under different conditions. In our study, we evaluate the most appealing four different feature schemes for the localization of the eye and the mouth regions in a face image: gray level values, the local binary patterns, the distance transform and the gradient values. The eyes and mouth regions are cropped from the training face images and they are resized to fit fixed size rectangular

windows, whose sizes are separately specified for the eyes and mouth. The feature representations are obtained from these rectangular image windows, and then they are used in the training of the SVM classifier. Finally, the SVM classifier is used for detection of facial fiducial points in new face images.

A. Gray Level Values

The digital image pixels are usually represented in gray-level scale notation where the gray level value varies between 0 and 255. The feature vector for the image is obtained by stacking the row-or-column pixel intensity values (gray level values) consecutively. When the illumination variations are not significant in the images, the gray-level representation may give enough discriminative information for detection of facial fiducial points.

B. Local Binary Patterns

The Local Binary Patterns (LBP) is a method which makes pair wise comparison of a pixel value with its neighboring pixels in order to assign a label to every pixel of the image resulting in a binary number [10]. The neighborhood size may differ to deal with textures at different scales but the 3×3 neighborhood is the most common one. Basically, the LBP pattern is obtained as follows: The center pixel is chosen as the reference pixel. The intensity value of the reference pixel is compared to the other pixels in its neighborhood. If the reference pixel value is greater than or equal to the neighboring pixel, the value of the neighboring pixel is set to 1; otherwise it is set to 0. The binary pattern for the reference pixel is then obtained by combining binary digits 1 and 0's from its neighboring pixels. The final resulting binary value is converted into its decimal representation. By this method, especially the lighting intensity changes in the image can be reduced. In our paper, we extracted LBP features using circular (8,1) neighborhoods as shown in Fig.1.

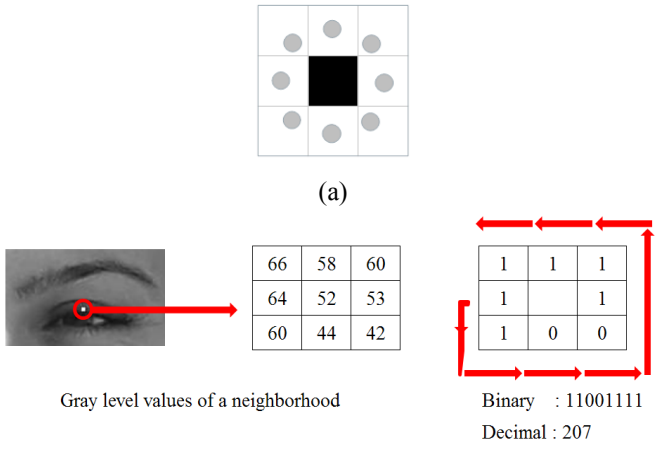


Figure 1. (a) A circular (8,1) neighborhood, (b) An LBP extraction using radius 1 and 8 neighboring pixels.

C. Gradient Values

Edge detection is a method which is used not only for feature extraction but also for object segmentation. Edge detection is basically used for identifying points in a digital image at which the image brightness changes sharply. Fig. 2 shows an edge detected image (or simply the gradient image) with the original one.

Figure 2. Original and gradient images.

In the edge detection, gradients of the image are computed, and the Roberts, Prewitt and Sobel operators are the well-known gradient-based edge detection operators. We used the Euclidean norm of the horizontal and vertical gradients as the feature, i.e., $\sqrt{G_x(i,j)^2 + G_y(i,j)^2}$ where G_x and G_y are given in Fig. 3. Gradient is more robust to illumination changes and it can be used as a feature to represent the fiducial points.

-1	0	1
-2	0	2
-1	0	1

G_x

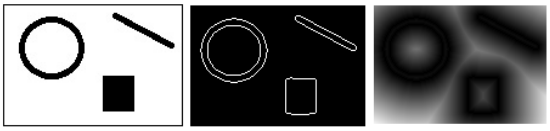
-1	-2	-1
0	0	0
1	2	1

G_y

Figure 3. The horizontal and the vertical gradients.

D. Distance Transform

The distance transform (DT) is basically a type of transformation of the image gray levels designed to be used for real-time machine vision or inspection tasks. DT operates on binary images and it assigns each bright (dark) pixel of a binary image a value equal to its distance to the nearest dark (bright) pixel [11]. The transformation utilizes distance between the pixels instead of edge information and some useful information between pixels can be extracted by this way. DT has been used in many computer vision applications as well as facial fiducial point detection [12]. Fig. 4 shows a DT applied image compared to the Canny edge detection applied version.



(a) (b) (c)

Figure 4. (a) Original image, (b) image with Canny edge detection operator, (c) image after distance transform.

III. SUPPORT VECTOR MACHINES

Support Vector Machine (SVM) classifier is considered one of the best methods to deal with tough classification problems, such as those arising in speech recognition, visual object

classification, text classification, etc [13,14]. SVM was originally proposed as a binary classification method and it finds the optimal separating hyperplane that maximizes the distance from the closest points of the classes to the separating hyperplane. Therefore, it is also called the maximum margin classifier [15]. Maximizing the margin between two classes on the training data usually leads to a better classification performance on the test data, especially in high-dimensional spaces with the limited number of samples. Fig. 5 demonstrates how SVMs work for two linearly separable classes. As can be seen in the figure, the margin between classes is determined by the nearest data samples which are also called the support vectors.

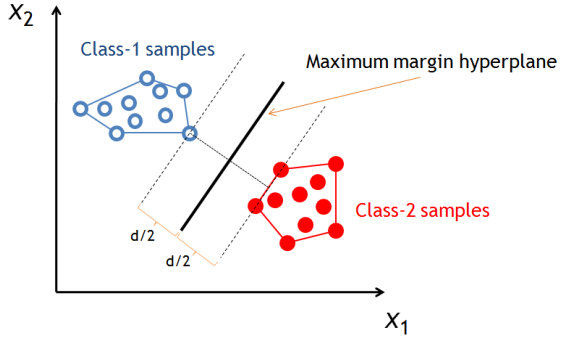


Figure 5. Classification of two classes by SVM classifier.

Now, let's consider a binary classification problem with the training data given in the form as in (1):

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N) \quad \mathbf{x}_i \in \Re^N, \; y_i \in \{-1, 1\} \tag{1}$$

Here $\mathbf{x}_i$ denotes the data samples and y_i denotes the class label (the positive or the negative class). The points $\mathbf{x}$ which lie on the separating hyperplane satisfy $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$, where $\mathbf{w}$ is the normal of the separating hyperplane, $|b| / \|\mathbf{w}\|$ is the perpendicular distance from the hyperplane to the origin, and $\|\mathbf{w}\|$ is the Euclidean norm of $\mathbf{w}$. For any separating hyperplane, all points $\mathbf{x}_i$ in the positive class satisfy $\langle \mathbf{w}, \mathbf{x}_i \rangle + b > 0$ and all points $\mathbf{x}_i$ in the negative class satisfy $\langle \mathbf{w}, \mathbf{x}_i \rangle + b < 0$, so that $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$ for all training data points. In the linearly separable case, the class separations are realized according to the sign of the following function f including nonzero α_i coefficients:

$$f(\mathbf{x}_{test}) = \sum_{i=1}^{N} \alpha_i y_i (\mathbf{x}_i^T \mathbf{x}) + b \tag{2}$$

For the linearly non-separable data, the data samples are mapped into a higher-dimensional space where the classes become separable and we find the best separating hyperplane in the mapped space. By using the kernel trick – i.e., replacing $< \mathbf{x_i}, \mathbf{x_j} >$ with the kernel function $k(\mathbf{x_i}, \mathbf{x_j}) = < \phi(\mathbf{x_i}), \phi(\mathbf{x_j}) >$, where $\phi : \Re^d \rightarrow \Im$ is the mapping function from the input space to the mapped space $\Im$ – we can find the best separating hyperplane features in the mapped space [16]. As a result, more complex nonlinear decision boundaries between classes can be approximated by using this trick. Separation of classes with the kernel is written as given in (3).

$$f(\mathbf{x}) = \left(\sum_{i=1}^{N} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \right) \tag{3}$$

In this study, we used two types of kernels:

- The linear kernel:

$$k(\mathbf{x}_i, \mathbf{x}_i) = \langle \mathbf{x}_i, \mathbf{x}_i \rangle \tag{4}$$

- The Gaussian kernel:

$$k(\mathbf{x}_i, \mathbf{x}_i) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2} \tag{5}$$

IV. EXPERIMENTAL WORK

Here we compare different features for automatic facial fiducial point detection. First we manually crop the face and mouth regions from the training face images and extract 4 different types of features (gray-level values, distance transform features, gradients and local binary patterns). Then, these features are used to train the SVM classifiers to find the best decision boundaries separating two classes. Finally the detection performance of the system has been computed from the test set of the database. The details of the experimental work are given below.

A. Database

In this study, we constructed a face database including 5968 digital face images which are used for training and 200 digital face images used for testing. The images are collected from the web and well-known face databases and hence their dimensions are quite different. The images have also large variations in illumination conditions and there are some partial occlusions. The initial face detection is performed by using the cascade classifier of Viola and Jones face detector that comes with OpenCV library [17]. Some face image samples from the database are shown in Fig. 6.

Figure 6. Some face images from the database.

B. SVM Training

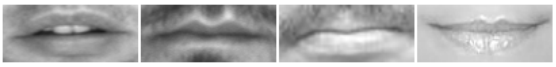
The eyes and mouth regions in the detected face images are cropped manually. The aspect ratios of the search windows are determined by using sizes of these manually cropped images. The cropped images are then resized according to the selected fixed window size. The window size for each eye is set to 38×22 and the window size of the mouth is set to 50×22. Some sample images belonging to the eyes and mouth are shown in Fig. 7.

For the negative class, some arbitrary regions (which do not include eyes or mouth) in the images are randomly chosen. Then, these selected regions are used to construct the negative

class samples of the data. The final size of the negative class for both the eyes and mouth detection was 14296. Fig. 8 shows some sample images from the negative classes.

(a) The left eye

(b) The right eye



(c) The mouth

Figure 7. Examples of eye and mouth images (positive class images) used in the training of the SVM classifiers.

After collecting the positive and negative class image samples, we applied the feature extraction schemes given in Section II. Then, the extracted features have been used to train two-class SVM classifiers. The optimal SVM parameters have been determined based on 5-fold cross-validation. Then, the trained classifiers were used in the testing images to get the correct detection rates of the eyes and mouth regions.

(a) Not-eye image samples

(b) Not-mouth image samples

Figure 8. Examples of not-eye and not-mouth images (negative class images) used in the training of SVM classifiers.

C. *Tests and Results*

In the test set, there are 200 face images which are not used in training. As in the previous case, we first detected face regions by using cascade classifiers of Viola and Jones. We then manually determined the true locations of the facial fiducial points. During detection, we divided the face image into 3 partitions as shown in Fig. 9, and searched the corresponding fiducial points only in these regions by using the sliding window approach.

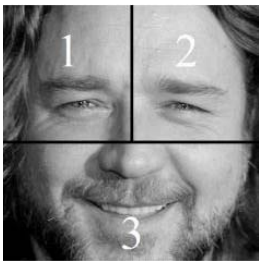


Figure 9. Face partitions.

During the search in the image, the window size is kept constant and the fiducial points are scanned in different scales. We set the scales by a factor of 1.2. Each rectangular window is classified as fiducial point or background based on the output of the SVM classifier. To assess the detection performance we used PASCAL VOC metric [9]. In this metric, detections are considered as true or false based on the overlap with manually annotated bounding boxes. The overlap between the bounding boxes B_p returned by the classifier and the annotated bounding box (ground truth) B_{gt} is computed as below

$$a_0 = \frac{area\left(B_p \cap B_{gt}\right)}{area\left(B_p \cup B_{gt}\right)}. \qquad (6)$$

If the resulting ratio a_0 is greater than 0.5, the detected region was decided to be the correct detection, otherwise it was decided to be the false detection. We count the number of true detections and divide the number of total images in the test set to compute the correct detection rates. The number of false positives will be equal to the difference between the total number of images and the number of correct detections. The results were given in Tables I and II for the linear SVM and the Gaussian SVM classifiers respectively.

TABLE I. CORRECT DETECTION RATES FOR THE LINEAR KERNEL

Features	Left Eye	Right Eye	Mouth
Distance T.	176/200	177/200	108/200
Gradients	170/200	182/200	91/200
LBPs	180/200	182/200	165/200
Gray Levels	162/200	170/200	60/200

TABLE II. CORRECT DETECTION RATES FOR THE GAUSSIAN KERNEL

Features	Left Eye	Right Eye	Mouth
Distance T.	180/200	183/200	138/200
Gradients	190/200	190/200	164/200
LBPs	192/200	190/200	182/200
Gray Levels	190/200	191/200	138/200

Table I shows that the performance of the gray level features is the worst among all tested features in the detection of all three fiducial points when the linear SVM classifiers are used. This implies that the adoption of other features can improve the performance of the detection system. For the detection of the left eye, the best result is obtained by using LBPs followed by the distance transform features, gradients and gray levels, respectively. For the detection of the right eye, LBPs and gradients perform equally achieving the best result. They are followed by the distance transform features and gray levels. In the detection of the mouth, the LBP features significantly outperform other features.

Using the Gaussian kernel in SVM classification generally improves the correct detection rates as seen in Table II. In this case, the performance of the distance transform features seems to be the worst among all tested features. Gray level features and gradient features achieve similar results for the eyes, but

gradient features achieve better performance for the mouth detection. As in the previous case, LBP features perform the best among all tested features for the detection of the mouth and its detection rate is significantly higher than the others. For mouth detection, using the Gaussian kernels significantly improves the performance compared to the linear kernel for the gray levels. Similar observation is also true for gradients as well. Overall, LBP features seem to be the best performer for detection of facial fiducial points. They are also the most efficient ones since the size of LBP features is much smaller than all other features (size of LBPs is always 59 whereas the sizes of other tested features are equivalent to the number of pixels in the sliding windows)

V. SUMMARY AND CONCLUSION

In this work, we have compared different feature extraction methods for automatic facial fiducial points detection. To this end, we have collected a new frontal face database where the images come from the web and different face recognition databases. We have studied four different feature extraction techniques, namely, the gray level values, distance transform, gradient and local binary patterns features. As a learning algorithm we used SVM classifiers using the linear and Gaussian kernels. In the study, 5968 face images were used to train the SVM classifiers. The detection performance is tested on a data base including 200 images. The results showed that when the linear SVM classifiers are used, the detection performance of gradients, distance transform features and local binary patterns were better than the detection performance of gray level value features for all fiducial points. This implies that those features can correspondingly be used for the gray level features to improve the detection performance. When the Gaussian kernel was used in the SVM classification, the performances of all feature extraction methods improved in general. However, the local binary pattern features are the best features of all especially with its high performance in the mouth detection. LBPs are also the most efficient feature since the size of it is much smaller than the other tested features. To conclude, LBP features seem to be the most accurate and efficient feature for automatic detection of facial fiducial points.

ACKNOWLEDGMENT

This work has been supported by Eskisehir Osmangazi University under the scientific research project with the project number 200915015 and the Scientific and Technological Research Council of Turkey (TUBİTAK) under Grant number EEEAG-109E279.

REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, “Face Recognition: A Literature Survey”, ACM Computing Surveys, vol. 35, no. 4, pp 399-458, Dec 2003.

[2] W. Min Huang, R. Mariani, “Face Detection and Precise Eyes Location”, ICPR, 2000.

[3] Z. H. Zhou and X. Geng, “Projection Function for Eye Detection”, Pattern Recognition, vol. 37, no. 5, pp. 1049-1056, 2004.

[4] Huang J. and Wechsler H., “Eye Detection Using Optimal Wavelet Packets and Radial Basis Functions (RBFs)”, International Journal of Pattern Recognition and Artificial Intelligence, vol. 13 no. 7, 1999.

[5] C.C. Tsai, W.C. Cheng, J.S. Taur and C.W. Tao, “Face Detection Using Eigenface and Neural Network,” IEEE International Conference on Systems, Man, and Cybernetics, pp 4443-4347, 2006.

[6] H. Proença and S. Filipe, “Combining Rectangular and Triangular Image Regions to Perform Real-Time Face Detection,” ICSP Proceedings , pp 903-908, 2008.

[7] M. H. Mahoor and M. abdel-Mottaleb, “Facial features extraction in color images using active shape model,” International conference on Automatic Face and Gesture Recognition, 2006.

[8] H. Jee, K. Lee and S. Pan, “Eye and Face Detection using SVM,” ISSNP, pp 577-580, 2004.

[9] Available at http://pascallin.ecs.soton.ac.uk/challenges/VOC/.

[10] Ahonen, T., Hadid, A. and Pietikäinen, M., “Face Description with Local Binary Patterns: Application to Face Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 28, 12, 2006.

[11] D. -J. Lee, J. Archibald, X. Xu, and P. Zhan, “Using Distance Transform to Solve Real-Time Machine Vision Inspection Problems,” Machine Vision and Applications, vol. 18, no. 2, pp. 85–93, 2007.

[12] S. Asteriadis, N. Nikolaidis, I. Pitas, M. Pardas “Detection of facial characteristiscs based on edge information,” International Conference on Computer Vision Theory and Applications, VISAPP 2007 Barcelona, March 8-11, 2007.

[13] C. Cortes, V. Vapnik, “Support vector networks,” Machine Learning. 20, pp. 273–297, 1995.

[14] Chapelle O., Haffner P., and Vapnik V. N., “Support Vector Machines for Histogram-Based Image Classification,” IEEE Transactions on Neural Networks. 10, No: 5, pp.1055-1064, 1999.

[15] C.J.C. Burges, “Tutorial on support vector machines for pattern recognition,” Data Min. Knowl. Discovery 2, pp. 121–167, 1998.

[16] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An Introduction to Kernel-Based Learning Algorithms”, IEEE Trans. Neural Networks, vol. 12, pp. 181–202, Mar. 2001.

[17] Viola, P. and Jones, M., “Robust real-time face detection,” International Journal of Computer Vision. 57 (2): 137-154, 2004.