automatic face recognition, support vector machines, face detection, feature extraction, multi-method fusion

Michał KAWULOK*

APPLICATION OF SUPPORT VECTOR MACHINES IN AUTOMATIC HUMAN FACE RECOGNITION

This paper presents the possibilities of applying the Support Vector Machines (SVM) in the process of automatic human face recognition. It is described how the existing methods of face recognition can be improved by the SVM. Moreover, a new approach to the multi-method fusion utilising the SVM is proposed. Usefulness of all the methods described in the paper improving the face recognition effectiveness by the SVM is confirmed by the experimental results.

1. INTRODUCTION

At first an idea of automatic face recognition should be described. It can be imagined that there is a system which acquires an image or a set of images as an input data, processes them, detects faces in these images and generates feature vectors describing each detected face. Furthermore, such a system is able to compare two feature vectors and calculate similarity between them in a form of the normalised similarity value. This makes it possible to assess similarity between any two given images containing faces in an automatic way (i.e. without human attendance or interaction). As a result, such a system discriminates between different faces and fulfils four main identification tasks [4]: classification, known-unknown problem (checking whether an image belongs to one of the defined classes or to none of them), verification, full identification.

Automatic face recognition can be divided into following phases of processing [11]: face detection, feature extraction, feature vector comparison which enables identification. There are various methods that are used during these steps, but their detailed description will be omitted in this paper. The interested reader is referred to [5, 11]. However, at every stage the Support Vector Machines (SVM) technique [2] can be used to improve the effectiveness of recognition.

2. SUPPORT VECTOR MACHINES

The SVM is a learning machine which solves two-group classification problems, but it can be enhanced to multi-class cases as well. The SVM based processing consists of two stages: learning and classification.

* Silesian University of Technology, Institute of Computer Science, Akademicka 16, 44-101 Gliwice, Poland

Learning aims at finding an optimal hyper plane separating a classified, linearly separable training data set: $(y_1,\mathbf{x_1})$, ..., $(y_n,\mathbf{x_n})$, $y_i \in \{-1, 1\}$, where $\mathbf{x_i}$ are vectors in N-dimensional input space and y_i are class labels. The hyper plane defined by $\mathbf{w_0} \cdot \mathbf{x} + b_0 = 0$ is found by maximizing the margin ρ between classes (1) and the normal vector $\mathbf{w_0}$ can be expressed as a linear combination of vectors from the training set ($\mathbf{w_0} = \sum_{i=1}^{n} y_i \alpha_i^0 \mathbf{x_i}$), where α_i^0 are non-negative Lagrange multipliers which are obtained during the optimization process. Each vector from the training set is associated with one α and it is worth noticing that relatively small number of α has non-zero values. Vectors with non-zero α are termed support vectors and are used further for classification.

$$\rho(\mathbf{w},b) = \min_{\{\mathbf{x}:y=1\}} \frac{\mathbf{x} \cdot \mathbf{w}}{|\mathbf{w}|} - \max_{\{\mathbf{x}:y=-1\}} \frac{\mathbf{x} \cdot \mathbf{w}}{|\mathbf{w}|} \tag{1}$$

When the training process is finished, the SVM allows for classification of vector $\mathbf{x}$, basing on the calculated decision surface (2). The main disadvantage of this solution is the assumption that the input data set must be linearly separable. However, it has been proved [2] that if this requirement is not satisfied, linear separability can be achieved by adding dimensions to the input space, which can be done by substituting dot-products of vectors in the input space with kernel functions $K(\mathbf{u},\mathbf{v})$. In this case the decision surface is calculated as presented in the Eq. (3).

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i \, \mathbf{x} \cdot \mathbf{x_i} \tag{2}$$

$$f(\mathbf{x}) = \sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}, \mathbf{x_i}) \tag{3}$$

The most popular kernel functions, solving the majority of classification problems are: linear, polynomial and radial basis functions (RBF). The most serious problem concerning the SVM is the choice of a representative training set. Depending on it, the SVM manages or not to find a general rule during learning, which determinates its effectiveness in solving classification problems. The training set can be chosen by simple drawing sets of samples belonging to each class. However, the initial research have shown that genetic algorithms may be quite effective for this purpose.

3. SVM FOR FACE DETECTION

Generally, face detection aims at finding position of eyes in the image, because this unambiguously defines location of a face. This is the first and very important step in the automatic face recognition and it is crucial to minimize detection errors. If a face is detected incorrectly or with little precision, it is virtually impossible to recognize it afterwards.

The SVM can be used to verify whether an image is containing a face or not. However, the SVM operates on normalized images of constant size and with eyes placed in fixed positions. To detect a face of any size in an image only with the SVM, it is necessary to scan the whole image scaled with various factors in order to conform to the normalized size of a face [7]. Such an approach is very time-consuming and cannot be applied in real-time systems. A reasonable method is to divide the detection into two steps. During the first one, areas in which probability of finding a face is high, should be identified. The second step verifies with the SVM whether these areas really contain faces or not.

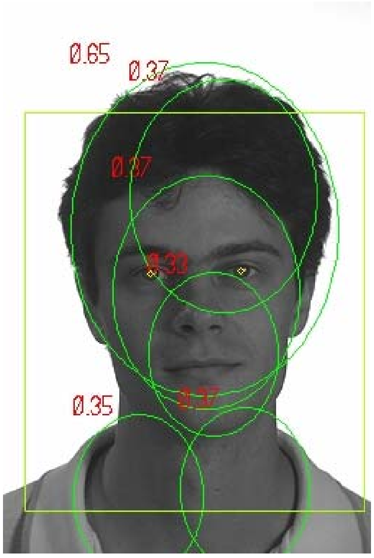

Fig. 1. A face image with detected ellipses. The numbers indicate the confidence of finding an ellipse. In the second step of the detection only one ellipse was classified by the SVM as a face and its eye positions are marked in the image.

The elimination in the first step is done by finding ellipses of different sizes which potentially are face contours (Fig. 1). Then the positions of eyes within these ellipses are found, so that potential face images could be normalized and verified by the SVM. Examples of such images are presented in Fig. 2. Ellipses are detected using the Hough transform, which is a relatively quick method and allows for real-time processing. Thanks to this approach the SVM is used for a small number of images and the detection process is not too time-consuming.

Fig. 2. Geometrically normalized face images after ellipse detection. Such images are verified by the SVM (two images on the left side were not classified as faces, whereas those on the right side were accepted).

The experimental results presented in Tab. 1 have confirmed that the detection precision has significant influence on the recognition rate and that the SVM verification technique is very efficient. Its effectiveness is much higher than the effectiveness of verification based on template matching with an average face image.

4. FEATURE VECTOR COMPARISON

Feature extraction is a highly complicated task due to the multidimensional nature of a human face. However, it has been proved that a greyscale image itself contains redundant information that should be eliminated in order to recognize an object in the image [10]. Feature extraction ($FE : \Re^N \to \Re^n, n < N$) renders projection from the N-dimensional normalized image space, in which primarily a face is located, into a new face space which has much less dimensions. Among the most popular feature extraction methods that are used for face images are the Eigenfaces method [6, 10] and the Elastic Bunch Graph Matching (EBGM) utilizing the Gabor Wavelets [8]. The majority of other methods is based on these two mentioned above.

As a result of the feature extraction, a face is represented by a point in the so called face space which can be treated as a feature vector. However, not only should the feature extraction method generate feature vectors, but it also should be able to measure the similarity between any pair of them ($S(\mathbf{v_1}, \mathbf{v_2})$). The most straightforward way is to calculate it as an inverse of the Euclidean distance between two points in the face space (4). This method usually performs well, but does not take advantage of specific properties of the face space, which effects in losing important information. This may be avoided by utilizing the SVM for comparing feature vectors, because its decision surface may be far better fitted to the face space than the Euclidean distance. To compare two feature vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ with the SVM, it is necessary to calculate the absolute difference vector $\mathbf{v_d}$ (4), which can be classified as having either intra-personal or extra-personal nature.

$$S(\mathbf{v_1}, \mathbf{v_2}) = \frac{1}{1 + \sqrt{\sum_{i=1}^{n} (v_{1i} - v_{2i})^2}}, \quad v_{di} = |v_{1i} - v_{2i}| \tag{4}$$

The intra-personal difference vectors are created by subtracting two feature vectors extracted from two different images of the same face. The extra-personal ones come from two images of different faces. The SVM should be trained with two sets of difference vectors (one set contains intra-personal, the second extra-personal ones). After that the SVM is capable of classifying difference vectors derived from any pair of feature vectors. Such an approach allows for the independence of vectors that are being classified from the training set. The SVM is trained universally and is not suited to a certain group of classes.

The performed experiments confirmed that the SVM for feature vector comparison outperforms the Euclidean distance method for both the Eigenfaces and the EBGM methods in the case of the Feret database [9] (Tab. 2).

5. MULTI-METHOD FUSION

There are various feature extraction methods that can be applied for face recognition. They often take into account different features of a face and they are mutually complementary. For example, the Eigenfaces method presents a holistic approach to face recognition, whereas the EBGM concentrates on local features. It is therefore reasonable to join various methods to improve the effectiveness of recognition [5].

For every single image each feature recognition method generates one feature vector and for each pair of images it produces one similarity S_i. Hence, for k methods each pair of images is assigned with k similarities. Then such a similarity vector must be transformed into one final similarity measure S. In this way, this transformation is a fusion of k methods. The fusion can be done by calculating the average of k similarities S_i (5a), but the experiments revealed that it is more effective to use the weighted mean instead (5b). In this case each method is assigned with an importance factor w_i, which should be proportional to its effectiveness.

$$S = \frac{1}{k}\sum_{i=1}^{k} S_i \ (a) \quad S = \frac{1}{k}\sum_{i=1}^{k} w_i S_i \ (b) \tag{5}$$

Unfortunately, it is quite difficult to find optimal weights w_i. In this case, the SVM is very useful again. A vector of similarities can belong to either of two classes. It can be derived from two images of the same person or of two different ones. It is therefore possible to train the SVM in such a way, that it classifies vectors of similarities. The outcome of the classification is treated then as the total similarity *S*. This approach proved its effectiveness in the experiments (Tab. 3).

6. EXPERIMENTAL RESULTS

The experiments were conducted on 1000 images of 395 different persons from the Feret face image database [9]. The SVM in all cases was trained with data derived from the Feret database, but from samples different from the tested set. During the experiments the tested data set was divided into a template set containing one face image of every person (that is 395 images) and a query set containing the rest (that is 605) images. Every image from the query set was compared with all the images from the template set and the similarities were sorted in descending order. The recognition rank was a place in the sorted list of a correct image, that is belonging to the same person as the tested one from the query set. Hence, if the recognition rank was equal to 1, an image was considered to be recognized correctly. The recognition rate was a percentage of correctly recognized images in the query set.

Face detection precision was measured according to the difference between the original and detected eyes location (Δ_L for the left and Δ_R for the right eye). This difference was divided by the distance between the eyes (*D*) to calculate the detection error $\delta_d = \dfrac{\Delta_L + \Delta_R}{2D}$. The tests were performed for the template matching technique of verification and for four SVM verifiers. The SVM was used with the Radial Basis Functions (RBF) kernel and trained with various training data sets. The first three columns of Tab. 1 contain the percentage of tested images for which the detection error was smaller than δ_d (8%, 11% and 16%). The forth column presents the recognition rates for all the detection cases. The SVM was trained with four different training data sets and it can be noticed that the choice of the training data strongly affects the results (ranging from 71.1 % to 73.4 % in the case of the recognition rate). The best detection result for the SVM (88.8 % for $\delta_d < 8\%$) was achieved by applying a simple version of genetic algorithm [3] for choosing the training set. The rest of the sets, for which the results are poorer, was generated randomly.

During the experiment an influence of detection error on recognition rate was checked as well. For comparison, the recognition rate for perfectly detected eyes (the positions pointed by human) was calculated and presented in Tab. 1. It is noticeable that the recognition rate depends strongly on detection precision.

Table 1 Influence of detection precision on recognition rate.

Detection method	Detection precision			Recognition rate
	$\delta_d < 8\%$	$\delta_d < 11\%$	$\delta_d < 16\%$	
Template matching	79.8 %	86.5 %	90.1 %	65.6 %
SVM	86.2 %	92.9 %	94.6 %	71.1 %
	86.7 %	93.5 %	95.0 %	71.6 %
	87.1 %	93.7 %	95.1 %	72.0 %
	88.8 %	95.1 %	96.9 %	73.4 %
Perfect detection	100 %	100 %	100 %	91.1 %

Table 2. Recognition rate for different feature vector comparison methods.

Feature extraction method	Comparison method	Recognition rate
Eigenfaces	Euclidean distance	71.7 %
	SVM with RBF kernel	72.7 %
	SVM with polynomial kernel	75.9 %
EBGM	Euclidean distance	88.8 %
	SVM with RBF kernel	89.9 %
	SVM with polynomial kernel	89.4 %

The second experiment was conducted to assess the effectiveness of several feature vector comparison methods. In order to avoid detection error propagation, the perfect detection was applied. During the test, feature vectors generated by the Eigenfaces and the EBGM methods were being compared, at first by measuring the Euclidean distance and then by utilizing the SVM classifier. In both cases the SVM gave the best results (Tab. 2). For the Eigenfaces a polynomial kernel $K(\mathbf{u},\mathbf{v}) = (\mathbf{u}\cdot\mathbf{v}+1)^2$ was the most effective, whereas for the EBGM the RBF one $K(\mathbf{u},\mathbf{v}) = \exp(-|\mathbf{u}-\mathbf{v}|/\sigma)$.

The last experiment tested a possibility of improving the recognition results by using the multi-method fusion. At first, four methods were tested separately and after that they were joined. These methods were: the Eigenfaces, the EBGM, the Fisherfaces [1] and the Eigenfaces with Mahalanobis distance (EFMAH).

The fusion was done primarily by calculating an average of partial similarities and then by using the SVM with linear kernel $K(\mathbf{u},\mathbf{v}) = \mathbf{u}\cdot\mathbf{v}$. The results (Tab. 3) allow to conclude that by fusing many methods it is possible to achieve better recognition rate than for single methods. Moreover, the SVM proved to give higher recognition rate in all tested cases.

Table 3. Recognition rate for multi-method fusion.

Single methods		Multi-method fusion		
Method	Recognition rate	Fused methods	Recognition rate	
			Average	SVM
Eigenfaces	75.9 %	Eigenfaces + EBGM	90.9 %	90.9 %
EBGM	89.9 %	Eigenfaces + EFMAH	76.0 %	76.9 %
Fisherfaces	75.4 %	Eigenfaces + EFMAH + EBGM	91.1 %	91.6 %
EFMAH	72.0 %	All methods	90.1 %	91.7 %

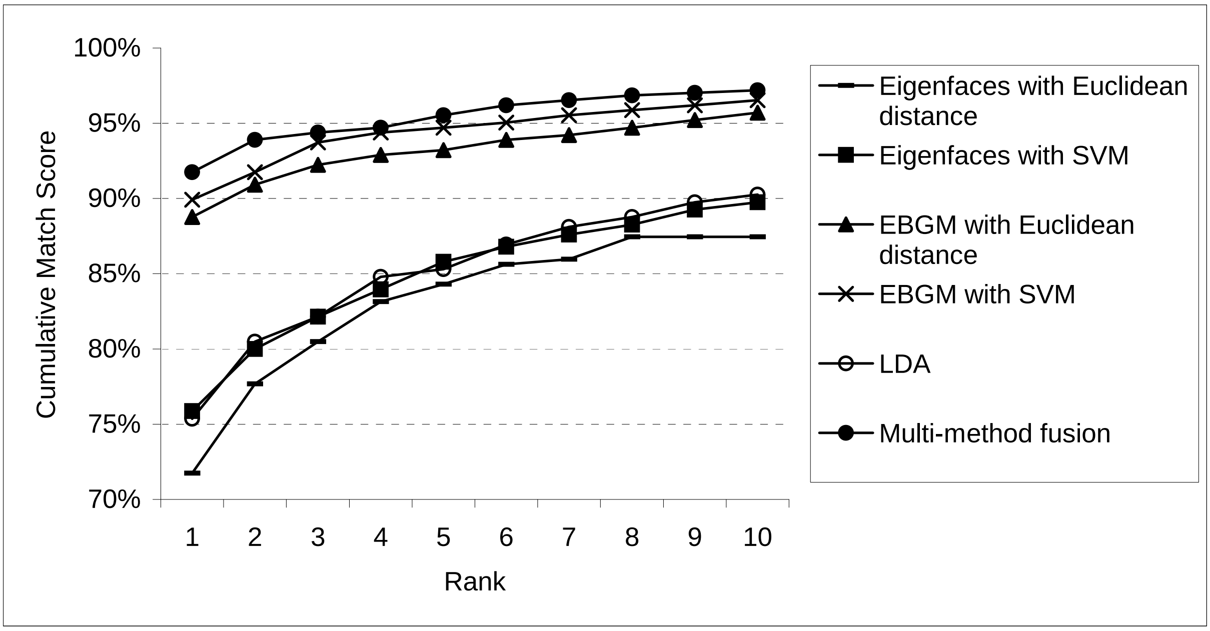

Fig. 3. Cumulative recognition rates for several methods and for multi-method fusion.

In Fig. 3 the cumulative recognition rates are presented. It can be observed that face recognition performs better for all recognition ranks when feature vectors are compared with the SVM. Also the multi-method fusion gives the highest recognition rate for every rank. The described experiments have confirmed that the SVM is very useful for human face recognition and can improve the results significantly.

7. CONCLUSIONS AND FUTURE WORK

In this paper the possibilities of applying the SVM in automatic face recognition have been presented. The performed experiments confirmed that the SVM may improve the results on every stage of this process. The main problem that should be solved in the future is to construct a method of choosing the optimal training data set. The application of genetic algorithms in this area is very promising and should be deeply analyzed.

Moreover, further applications of the SVM are planned to be tested. Particularly, it may be possible to apply specific methods depending on image characteristics. The SVM would classify an image and choose the most effective technique of processing. Initial experimentation suggest that such an approach could be very effective especially for image normalization.

BIBLIOGRAPHY

[1] BELHUMEUR P. N., HESPANHA J. P., KRIEGMAN D. J., Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Trans. Patt. Anal. Mach. Intell. 19, 711–720, 1997

[2] CORTES C., VAPNIK V., Support vector networks. Machine Learning, 20(3):273-297, September 1995.

[3] GOLDBERG D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, 1989, Addison-Wesley Publishing Co.

[4] GONG S., MCKENNA S. J., PSARROU A., Dynamic Vision From Images to Face Recognition, Imperial College Press 1999.

[5] GROTHER P., Face Recognition Vendor Test 2002, Supplemental Report, February 2004, NIST IR 7083.

[6] KAWULOK M., Masks and Eigenvectors Weights for Eigenfaces Method Improvement, ICCVG International Conference on Computer Vision and Graphics 2004, Warsaw, September 2004.

[7] LI Y., GONG S., LIDDELL H., Support Vector Regression and Classification Based Multi-view Face Detection and Recognition. IEEE International Conference on Face Gesture Recognition, Grenoble France, March 2000.

[8] OKADA K., STEFFENS J., MAURER T., HONG H., ELAGIN E., NEVEN H., MALSBURG C., The Bochum/USC Face Recognition System and How it Fared in the FERET Phase III Test, in: Face Recognition: From Theory to Applications, H. Wechsler, P. J. Phillips, V. Bruce, F. F. Soulie, T. S. Huang, Eds. Springer-Verlag, Berlin, Germany, 186–205.

[9] PHILLIPS P. J., WECHSLER H., HUANG J., AND RAUSS P., The FERET database and evaluation procedure for face recognition algorithms, Image and Vision Computing J, Vol. 16, No. 5, pp 295-306, 1998.

[10] TURK M., PENTLAND A., Face Recognition Using Eigenfaces, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1991, p.586 – 591.

[11] ZHAO W., CHELLAPPA R., PHILLIPS P. J., ROSENFELD A., Face Recognition: A Literature Survey, Technical Report CARTR-948, Center for Automation Research, University of Maryland, College Park, MD, 2000.