Krzysztof SIWEK[1], Stanisław OSOWSKI[1,2]

[1]Warsaw University of Technology, Institute of the Theory of Electrical Engineering, Measurement and Information Systems
[2]Military University of Technology, Institute of Electronic Systems

Comparison of methods of feature generation for face recognition

Abstract. *The paper is concerned with the recognition of faces at application of different methods of global feature generation. We check the selected choice of transformations of images, leading to the numerical representation of the face image. The investigated approaches include the linear and nonlinear methods of transformation: principal component analysis (PCA), Kernel PCA, Fisher linear discriminant analysis (FLD), Sammon transformation and stochastic neighbor embedding with t-distribution (tSNE). The representation of the image in the form of limited number of main components of transformation is put to the input of support vector machine classifier (SVM). The numerical results of experiments are presented and discussed.*

Streszczenie *Praca przedstawia analizę porównawczą różnych metod wstępnego przetwarzania obrazów twarzy dla wygenerowania cech diagnostycznych zastosowanych w klasyfikacji. W badaniach uwzględniono metodę transformacji PCA, KPCA, FLD, transformację nieliniową Sammona oraz transformację tSNE. Cechy wygenerowane przy użyciu tych metod stanowią sygnały wejściowe dla klasyfikatora SVM dokonującego ostatecznego rozpoznania. W pracy pokazano i przedyskutowano wyniki przeprowadzonych eksperymentów rozpoznania twarzy przy uwzględnieniu zmiennej liczby cech dla różnej liczby klas. (**Porównanie metod generacji cech dla rozpoznawania twarzy**)*

Keywords: face recognition, transformation of data, classification, SVM.
Słowa kluczowe: rozpoznawanie twarzy, metody generacji cech, klasyfikacja, SVM.

doi:10.12915/pe.2014.04.49

Introduction

A face recognition system is a computer application for automatically identifying or verifying a person from a digital image. The problem of recognition of images, especially the face, is crucial in many different applications [1,2,3]. It is typically used in security systems, fulfilling the role of the first verification of the set of images. The most important point in face recognition is generation of the features, well representing different classes of images.

One approach to feature generation is through extracting the local landmarks, characteristic for the image of the subject's face. Special approach to local feature generation is application of SIFT [4]. The other group of algorithms uses the global approach characterizing the whole image in a statistical way, normalizing a gallery of face images and then compressing the face data, only saving the data in the image that is useful for face detection. In the retrieval stage a probe image is compared with the data representing the set of faces. This approaches typically exploit the linear transformation, like PCA (the eigenface method) or FLD (Fisherface method) [1].

In this paper we will study the features of the face image generated in different way. The linear methods will be extended to the nonlinear ones, including Kernel PCA (KPCA), Sammon transformation (ST) and stochastic neighbor embedding with t-distribution (tSNE). The features generated by different methods will form the input to the class recognition classifiers, implemented here by SVM and random forest. The results of this comparison will be presented and discussed in the paper.

Transformations of the global image for feature generation

The most important point in feature representation of the face is to find the transformation of the highest compression ability of the images, able to pack the global distribution of pixels into smallest possible number of the significant features. In this paper we will limit our considerations to few of them, including PCA, LDA, Kernel PCA, Sammon transformation and stochastic neighbor embedding. The mentioned above transformations have been found to be valuable in efficient visualizing the distribution of different classes of multidimensional systems in 2D coordinate system [5].

The original face image (the matrix) will be represented by the vector $\mathbf{x}$, $\mathbf{x} = [x_1, x_2, ..., x_N]^T$ formed by the succeeding rows of the matrix. Let us assume that the set of such vectors representing the images is of zero mean value. The PCA transformation of any member of such set is described by the linear relation [1]

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{1}$$

of the transformation matrix $\mathbf{W} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K]^T$ defined by the set of fixed K eigenvectors $\mathbf{v}_i$ corresponding to the highest values of the eigenvalues of the covariance matrix $\mathbf{R}_{xx}$, where $\mathbf{R}_{xx}=E[\mathbf{x}\mathbf{x}^T]$. To avoid the problem of processing the $N \times N$ dimension covariance matrix $\mathbf{R}_{xx}$ (N usually very high) we form the small dimension matrix $\mathbf{R}_{sxx}$ of the size $p \times p$, (p – the number of samples), i.e., $\mathbf{R}_{sxx}=E[\mathbf{x}^T\mathbf{x}]$. The true PCA of real (very large) dimension is created on the basis of this small dimension matrix using the eigen-decomposition of $\mathbf{R}_{sxx}$. This decomposition generates the set of small size eigen-vectors $\mathbf{v}_{s1}$, $\mathbf{v}_{s2}$,…$\mathbf{v}_{sp}$. The eignevectors corresponding to the highest eigenvalues form the matrix $\mathbf{V}_s$. The return to the normal (high) size of these vectors is achieved through the transformation $\mathbf{V} = \mathbf{X}^T * \mathbf{V}_s$, in which $\mathbf{V}$ is the matrix representing the original (high size) eigen-vectors (arranged column-wise). Then the final PCA matrix $\mathbf{W}$ is determined as follows $\mathbf{W} = [\mathbf{V}_1, \mathbf{V}_2, ..., \mathbf{V}_K]^T$, where $\mathbf{V}_i$ represents the first succeeding columns of the matrix $\mathbf{V}$ (up to K). Thanks to this approach we avoid the problem of processing very high dimensional matrices in eigen-value decomposition. The features used in image recognition are the elements of the vector $\mathbf{y}$. It was proved that PCA procedure corresponds to the maximization of the objective function

$$\max_{\mathbf{W}} J = \left| \mathbf{W}^T \mathbf{R}_{xx} \mathbf{W} \right| \tag{2}$$

No information of the class membership of the learning samples is taken into account in the process of choosing the matrix W.

The Fisher linear discriminant analysis is the transformation that corrects this defect and takes care of class membership of the learning samples in determination of the transformation matrix. In particular it adjusts **W** in a way to maximize the objective function defined in the following way

$$\mathbf{W} = \arg\max_{\mathbf{W}} \frac{\left|\mathbf{W}^T \mathbf{S}_b \mathbf{W}\right|}{\left|\mathbf{W}^T \mathbf{S}_w \mathbf{W}\right|} = \left[\mathbf{v}_1\ \mathbf{v}_2 \cdots \mathbf{v}_K\right] \tag{3}$$

In this formulation the vectors $\mathbf{v}_i$ (i=1, 2, ..., K) represent the set of the most important eigenvectors associated with the solution of the generalized eigen-problem defined by

$$\mathbf{S}_b \mathbf{v}_i = \lambda_i \mathbf{S}_w \mathbf{v}_i \tag{4}$$

where $\mathbf{S}_b$ and $\mathbf{S}_w$ represent the between- and within-the-class scatter matrices [1], respectively, defined as

$$\mathbf{S}_b = \sum_{i=1}^{M} N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \tag{5}$$

$$\mathbf{S}_w = \sum_{i=1}^{M} \sum_{\mathbf{x}_k \subset klasa_i} (\mathbf{x}_k - \mathbf{m}_i)(\mathbf{x}_k - \mathbf{m}_i)^T \tag{6}$$

where M is the number of classes, N_i – number of data belonging to ith class, **m** – the average vector of data **x** belonging to all classes, $\mathbf{m}_i$ – the average vector of data **x** belonging to ith class.

The nonlinear version of PCA, called Kernel PCA (KPCA) represents the ordinary PCA defined on the nonlinear mapping of the vectors **x** [6]. Instead of original vectors we take in this transformation their nonlinear mapping **φ**(**x**) and the whole procedure is done now on the covariance matrix $\mathbf{R}_{\varphi\varphi}=E[\boldsymbol{\varphi}(\mathbf{x})\boldsymbol{\varphi}(\mathbf{x})^T]$. In practice the true transformation is done on the kernel matrix **K** defined as

$$\mathbf{K} = \begin{bmatrix} K(\mathbf{x}_1,\mathbf{x}_1) & K(\mathbf{x}_1,\mathbf{x}_2) & \cdots & K(\mathbf{x}_1,\mathbf{x}_n) \\ \cdots & \cdots & \cdots & \cdots \\ K(\mathbf{x}_n,\mathbf{x}_1) & K(\mathbf{x}_n,\mathbf{x}_2) & \cdots & K(\mathbf{x}_n,\mathbf{x}_n) \end{bmatrix} \tag{7}$$

where $K(\mathbf{x}_i,\mathbf{x}_j)=\boldsymbol{\varphi}(\mathbf{x})^T\boldsymbol{\varphi}(\mathbf{x})$. The eigenvalue decomposition performed on this matrix allows to find the matrix **W** and the final transformation is done similarly as in (1).

The next transformation (the Sammon approach) belongs to the nonlinear mappings [7]. It is designed to minimize the differences between corresponding inter-point distances in the original and transformed spaces. The method conserves (as much as possible) the distance between each pair of points in both spaces. In mathematical terms the problem is defined to find the mapping of the original vectors $\mathbf{x}_i$ (i=1, 2, …, p) into transformed vectors $\mathbf{y}_i$ minimizing the objective function

$$\min\ E = \frac{1}{c} \sum_{i<j} \frac{\left(d_{ij}^* - d_{ij}\right)^2}{d_{ij}^*} \tag{8}$$

where $d_{ij}^* = d(\mathbf{x}_i,\mathbf{x}_j)$ and $d_{ij} = d(\mathbf{y}_i,\mathbf{y}_j)$ represent the Euclidean pair-wise distances between all points in original and transformed spaces, $c = \sum_{i<j}^{p} d_{ij}^*$.

The last considered transformation is a stochastic neighbor embedding with a t Student distribution [5]. It starts by converting the high dimensional Euclidean distances between data points into the conditional probabilities that represent similarities between objects. It tries to find the map points (vectors $\mathbf{y}_i$ and $\mathbf{y}_j$) of the high-dimensional data points ($\mathbf{x}_i$ and $\mathbf{x}_j$) in a way to minimize a Kullback-Leibler divergence between the joint probability distribution p_{ij} in high-dimensional space and a joint probability distribution q_{ij} in the transformed (lower dimensional) space

$$\min\ C = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{9}$$

It was proved [5], that this transformation very well preserves the original structure in a lower dimensional space and locates the samples of the same class close to each other (reducing standard deviation). Hence the method is naturally well suited for generation of diagnostic features that enable the image recognition.

The Classifiers Used in Face Recognition

After selection of the features, we can go to the final step of image recognition – the classification. In this approach the selected set of features is put to the input of the classifier. To obtain the best results of recognition we have applied the most efficient classifiers, belonging to 2 families: the SVM [6] and Breiman [8] random forest of the decision trees.

The SVM is a linear machine, working in a space formed by the non-linear mapping of the original input vectors **y** into a feature space through the use of a kernel function $K(\mathbf{y},\mathbf{y}_i)$. The learning problem of SVM is defined as the task of separating the learning vectors into two classes of the destination values either d_i=1 (one class) or d_i=−1 (the opposite class), with the maximal separation margin. The great advantage of SVM is the unique formulation of learning problem leading to the quadratic programming with linear constraints, which is easy to solve. The separation margin formed in the learning stage according to the assumed value of the regularization constant C, provides some immunity of this classifier to the noise and hence this solution is known of very good generalization abilities [6].

To deal with a problem of many classes the one against one or one against all approaches working on a principle of the majority voting [6] are usually used. In our solution we have applied the one against one approach, since this approach usually leads to the better total results of recognition.

The Breiman [8] random forest is an ensemble of many multivariate decision trees indicating the class pointed by the majority of the individual trees. The method uses "bagging" idea and the random selection of features for each node of the tree, to construct a collection of decision trees with a controlled variation. In this way the individual trees in the forest are constructed to provide the highest degree of independence.

Let us assume the number of training cases p, and the number of input variables in the classifier N. Let m denotes the number of input variables used to determine the decision at a any node of the tree (m<N). The training set for the tree is selected by choosing n times the sample (with replacement) from all p available training cases. The rest of the cases is used to estimate the error of the tree, by predicting their classes. For each node of the tree, we choose randomly m variables to make decision at that node. Estimate the best split based on these m variables in the training set. The class membership of a new sample is

estimated by pushing it down the set of trees. Each tree assigns the label of the training sample in the terminal leave it ends up in. The procedure is iterated over all trees in the ensemble, and the majority of votes of all trees in the forest decides on the final membership of the sample to the particular class.

The Results of Numerical Experiments

The data base

The numerical experiments have been done on a set of face images representing up to 20 classes of people (both women and men) represented by 20 individuals in different poses and different illumination. The size of original images was 100×100, resized next to 50×50. This base was specially prepared for special research project. Fig. 1 presents the single representatives of the succeeding classes.

Fig. 1. The set of single representatives of the recognized classes of face images

The diversity of poses of faces representing one chosen class is well illustrated in Fig. 2, presenting 5 images of the same person. They differ by lighting, facial expression, turn of the face, background of photo, presence or absence of glasses and the magnification ratio.

Fig. 2. The exemplary representatives of one chosen class of the face images

The visualization of multidimensional vectors

To check the difference between different methods of multidimensional data transformations in a visual way we will use them to reduce the size of the mapped vectors of face images to the value of 2, just to enable to present them in a 2-dimensional coordinate system. Each method of transformation has delivered different results. Fig. 3 presents the distribution of data belonging to only 6 classes by using only PCA and tSNE [9]. We have limited the number of mapped classes to 6 in order to make the visual results more readable.

Fig. 3a corresponds to the linear PCA projection and Fig. 3b to the nonlinear tSNE. The nonlinear tSNE transformation has resulted in much closer locations of samples belonging to the same classes. The mixing of the classes is also much smaller in the case of tSNE. It means that this method of data processing is better suited for generation of the diagnostic features for the face recognition using the automatic classifier.

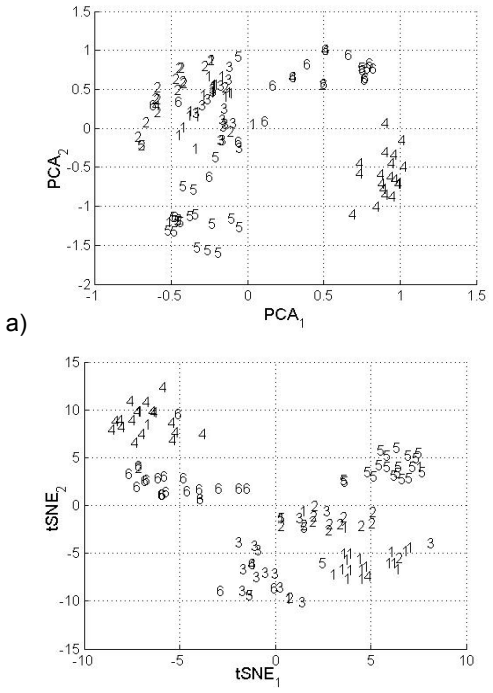

Fig. 3. The results of mapping the original face images of 6 classes into 2D coordinate system: a) PCA transformation, b) tSNE transformation

Adjustment of the optimal size of the feature vectors

Each feature generation method operates on different basis and packs the most important class discriminative information in a different way. Only small portion of the totally available information is important from the class recognition point of view. Therefore the first task was to discover how many features in each method should be used. This part of investigations was performed by learning the SVM classifiers fed by different number of the features obtained in each transformation process. The experiments were done in a 10-fold cross validation approach repeated for the predefined number of the features. According to the results of these experiments we have found that the size of the optimal feature set varies for different number of classes. Table 1 shows these results for 6, 12 and 20 classes.

Table 1. The size of the optimal set of features generated by different methods at recognition of different number of classes

No of classes	PCA	FLD	KPCA	ST	tSNE
6	19	5	14	26	15
12	22	11	18	28	21
20	24	19	25	30	30

As we can see each method of feature generation requires different population of features to get the best results of recognition. Only FLD stayed on the highest number of features, equal the number of classes reduced by one.

The results of face recognition

After adjusting the optimal set of features for each method we were able to check and compare their efficiency of class recognition at application of the same classifier system. In the experiments we have used two classifiers: the SVM working in one against one mode and random forest of decision trees (both implemented in Matlab [10]). In both cases the classification was performed in 10-fold cross validation approach. The final results are in the form of the mean value and standard deviation of the relative recognition error on the testing part of data, not taking part in learning. In the case of SVM the linear kernel was found the best at application of C=1000. The random forest was run at *m*=3 and 100 decision trees in the forest, by using 2/3 of data for training and the remaining 1/3 for testing. 10 repetitions of performing random forest classification have been done and the results averaged.

Table 2. The SVM misclassification ratios (mean+/-std) of different number of classes at application of the features generated by different methods

Number of classes	PCA [%]	FLD [%]	KPCA [%]	ST [%]	tSNE [%]
6	0.33+/-1.05	1.38+/-1.74	0.31+/-1.05	0.67+/-1.41	0.30+/-0.10
12	2.00+/-1.89	4.40+/-1.68	2.17+/-1.93	3.31+/-1.11	1.50+/-1.23
20	3.20+/-1.55	6.00+/-2.52	3.27+/-1.25	3.42+/-1.34	2.80+/-1.98

Table 2 presents the relative misclassification errors obtained at recognition of different number of classes at application of SVM and all methods of feature generation.

Table 3. The random forest misclassification ratios (mean+/-std) of different number of classes at application of the features generated by different methods

Number of classes	PCA [%]	FLD [%]	KPCA [%]	ST [%]	tSNE [%]
6	2.58+/-1.14	1.01+/-1.02	2.50+/-1.03	1.33+/-0.80	0.58+/-0.40
12	2.41+/-0.56	1.04+/-0.65	2.08+/-1.09	4.10+/-0.98	1.80+/-0.26
20	3.17+/-0.63	2.47+/-0.82	3.19+/-0.69	4.54+/-1.02	2.37+/-0.35

The results correspond to the application of 10-fold cross validation approach (the testing of samples not taking part in learning). Analyzing the results we can observe very good performance of the nonlinear tSNE method. The smallest are not only the mean values of the misclassification errors for each number of classes, but also standard deviations of errors in each trial. Surprisingly application of FLD has resulted into relatively high errors. Application of random forest (Table 3) as the classifier has resulted into a slightly different distribution of errors. They were a bit higher (except FLD). However, we can observe better performance of this classifier at the highest number of classes (20), where in most cases the misclassification errors were smaller for almost all methods of feature generation.

Conclusions

The paper has presented and compared different methods (linear and nonlinear) of face image representation in their recognition process. The numerical experiments have shown the superiority of nonlinear tSNE transformation in generation of diagnostic features. Irrespective of the number of recognized classes the tSNE allowed to obtain the smallest misclassification errors in most cases. The advantage of this method is especially well visible at high number of recognized classes.

RFERENCES

[1] Belhumeur P., Hespanha J., Kriegman D., Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *IEEE Trans. PAMI*, 1997, vol. 19, No 7, pp. 711-720

[2] Breiman L., Random forests. Machine Learning, 2001, vol. 45, pp. 5–32

[3] Duda, R.O., Hart, P.E., Stork, P., Pattern classification and scene analysis. Wiley, New York, 2003

[4] Jakubowski J., Ocena możliwości wykorzystania deskryptorów cech lokalnych obrazu twarzy w zadaniu automatycznej identyfikacji osób. Przeglad Elektrotechniczny, 2012, vol. 88, pp. 217–221

[5] Osowski S., Sieci neuronowe SVM w zastosowaniu do klasyfikacji wzorców, Przegląd Elektrotechniczny, 2002, vol. 78, No 2, ss.29-36

[6] Phillips P. J., Moon H., Rizvi S., Rauss P., The FERET evaluation methodology for face recognition algorithms, *IEEE Trans. PAMI*, 2000, vol. 22, No 10, pp. 1090-1104

[7] Phillips P. J., Wechsler H., Huang J., Rauss P., The FERET database and evaluation procedure for face recognition algorithms, Image and Vision Computing J, 1998, vol. 16, No. 5, pp 295-306.

[8] Sammon J. W., A nonlinear mapping for data structure analysis. IEEE Trans. Computers, 1969, vol. C-18 (5), pp. 401–409

[9] Schölkopf B., Smola A., Learning with kernels, Cambridge, *MIT Press, MA.* 2002

[10] Van der Maaten L., Hinton G., Visualising data using t-SNE. Journal MLR, 2008, vol. 9, pp. 2579–2602

[11] Van der Maaten L., Matlab toolbox for dimensionality reduction. v0.7.1b, Delft University of Technology, 2011

[12] Matlab user manual – image toolbox. MathWorks, Natick, USA 2012

***Authors**: dr hab. inż. Krzysztof Siwek, Warsaw University of Technology, Institute of the Theory of Electrical Engineering, Measurement and Information Systems, Email: ksiwek@iem.pw.edu.pl.*

prof. dr hab. inż. Stanisław Osowski, Warsaw University of Technology, Institute of the Theory of Electrical Engineering, Measurement and Information Systems, Military University of Technology, Institute of Electronic Systems, Email: sto@iem.pw.edu.pl.