Journal of ELECTRICAL ENGINEERING, VOL. 59, NO. 4, 2008, 203–209

SUPPORT VECTOR MACHINES, PCA AND LDA IN FACE RECOGNITION

Ján Mazanec — Martin Melišek — Miloš Oravec — Jarmila Pavlovičová *

In this paper, we consider the human face be biometric. We present the results of different statistical algorithms used for face recognition, namely PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis) and SVM (Support Vector Machines). Pre-processed (normalization of size, unified position and rotation, contrast optimization and face masking) image sets from the FERET database are used for experiments. We take advantage of csuFaceIdEval and libsvm software that implement the mentioned algorithms. We also propose a combination of PCA and LDA methods with SVM which produces interesting results from the point of view of recognition success, rate, and robustness of the face recognition algorithm. We use different classifiers to match the image of a person to a class (a subject) obtained from the training data. These classifiers are in the form of both simple metrics (Mahalinobis cosine, LdaSoft) and more complex support vector machines. We present the results of face recognition of all these methods. We also propose the best settings in order to maximize the face recognition success rate.

K e y w o r d s: biometrics, face recognition, principal component analysis, linear discriminant analysis, support vector machines

1 INTRODUCTION

Various automated systems for identification of people based on biometrics are used recently. Along with well-known methods such as fingerprint or DNA recognition, face recognition opens new possibilities. Many prerequisites for putting face recognition into practice, *eg*, face localization in digital cameras, have already been adopted by companies and are commercially available. Face recognition is already being implemented into image organizing software [5], web applications [6]; mobile devices, and passports already contain face biometric data [7]. All this implies that face recognition is an increasingly important field of biometry. The advantages of face recognition are relatively modest requirements on hardware and simple real-time process from the viewpoint of the identified subjects.

2 ALGORITHMS AND METHODS

One of the most challenging problems face recognition deals with is an appropriate separation of the data that belong to the same class. In face recognition, a class represents all data of the same subject, *ie*, all images of the same person. The goal is to implement an automated machine supported system that (after initialization and training by representative sample of images) recognizes person's identity in the images that were not trained before. This can have various practical applications such as automated person identification, recognition of race, gender, emotion, age etc. The area of face recognition is well described at present, *eg*, starting by [8], combining and comparing PCA and LDA in [9, 10], and continuing at present by kernel methods [11, 12].

In this work we examine PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis) and SVM (Support Vector Machines) in the problem of face recognition.

2.1 PCA and LDA

An image can be viewed as a vector of pixel values (*eg*, a 256×256 pixel grey-scale image can be represented as a vector containing 65536 values). Such an image vector can be examined not only in its original space but also in many (theoretically infinite) other subspaces into which the image vector can be transformed by various mathematical / statistical manipulations. PCA and LDA algorithms are examples of such transforms of an image. They transform image vectors into their subspaces (also called "feature spaces") and serve as a feature extraction stage by which it is possible to find a hyperplane that separates data into classes. Both methods implement linear separation of data, which is illustrated in Fig. 1 (simplified 2-dimensional version of the problem).

PCA aims to maximize between-class data separation, while LDA tries to maximize between-class data separation and minimize within class data separation [13].

Basic steps of PCA algorithm [14]

1. Determine PCA subspace (what is analogous to determination of the line in Fig. 1) from training data. i^{th} image vector containing N pixels is in the form

$$\boldsymbol{x}^i = \left[x_1^i, \ldots, x_N^i\right] . \tag{1}$$

* Department of Telecommunications, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 812 19 Bratislava, Slovakia; jmazanec@gmail.com, carheart@gmail.com, milos.oravec@stuba.sk, jarmila.pavlovicova@stuba.sk

ISSN 1335-3632

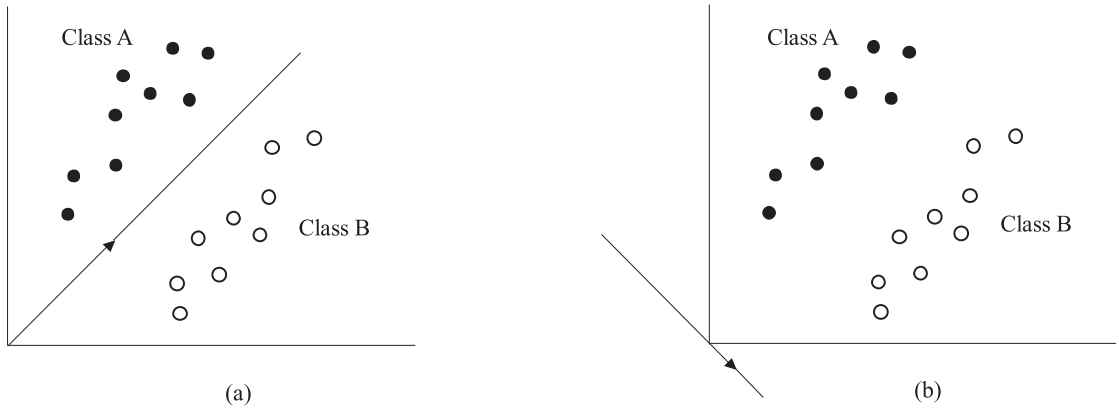

Fig. 1. (a) – Effective PCA data separation, (b) – Effective LDA data separation.

Store all p images in the image matrix

$$\mathbf{X} = \lfloor \boldsymbol{x}^1, \ldots, \boldsymbol{x}^p \rfloor \,. \tag{2}$$

Compute covariance matrix

$$\boldsymbol{\Omega} = \mathbf{X}\mathbf{X}^\top. \tag{3}$$

Compute eigenvalues and eigenvectors

$$\boldsymbol{\Omega}\mathbf{V} = \boldsymbol{\Lambda}\mathbf{V}\,, \tag{4}$$

where $\boldsymbol{\Lambda}$ is the vector of eigenvalues of the covariance matrix.
Order eigenvectors

$$\mathbf{V} = \lfloor \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p \rfloor \,. \tag{5}$$

Order the eigenvectors in $\mathbf{V}$ according to their corresponding eigenvalues in descending order. Keep only the eigenvectors associated with non-zero eigenvalues. This matrix of eigenvectors forms the eigenspace $\mathbf{V}$, where each column of $\mathbf{V}$ is the eigenvector. Visualized eigenvectors of the covariance matrix are called eigenfaces [8].

Basic steps of LDA algorithm [14]

LDA uses PCA subspace as input data, *ie*, matrix $\mathbf{V}$ obtained from PCA. The advantage is cutting the eigenvectors in matrix $\mathbf{V}$ that are not important for face recognition (this significantly improves computing performance).

LDA considers between and also within class correspondence of data. It means that training images create a class for each subject, *ie*, one class = one subject (all his/her training images).

1. Determine LDA subspace (*ie*determining the line in Fig. 2) from training data. Calculate the within class scatter matrix

$$\mathbf{S}_w = \sum_{i=1}^{C} \mathbf{S}_i\,, \quad \mathbf{S}_i = \sum_{\boldsymbol{x} \in X_i} (\boldsymbol{x} - \boldsymbol{m}_i)(\boldsymbol{x} - \boldsymbol{m}_i)^\top, \tag{6}$$

where $\boldsymbol{m}_i$ is the mean of the images in the class and C is the number of classes.

Calculate the between class scatter matrix

$$\mathbf{S}_B = \sum_{i=1}^{N} n_i (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^\top, \tag{7}$$

where n_i is the number of images in the class, $\boldsymbol{m}_i$ is the mean of the images in the class and $\boldsymbol{m}$ is the mean of all the images.

Solve the generalized eigenvalue problem

$$\mathbf{S}_B\mathbf{V} = \boldsymbol{\Lambda}\mathbf{S}_W\mathbf{V}\,. \tag{8}$$

The following steps are performed by both methods

2. All training images are projected onto particular method's subspace

3. Each test image is also projected to the same subspace and compared by distance metrics between the image and training images (distance metrics are different for both methods).

2.2 Metrics Used

Mahalinobis Cosine [3]

Mahalinobis Cosine (MahCos) is defined as the cosine of the angle between the image vectors that were projected into the PCA feature space (the so-called eigenvectors) and were further normalized by the variance estimates. Vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are image vectors in the unscaled PCA space (the so-called eigenvectors) and vectors $\boldsymbol{m}$ and $\boldsymbol{n}$ are their projections in the Mahalinobis space. First we define $\lambda_i = \sigma_i^2$, where λ_i is the PCA eigenvalue, σ_i^2 is the variance along those dimensions and σ_i is the standard deviation. The relationships between the vectors are then defined as

$$m_i = \frac{u_i}{\sigma_i}\,, \tag{9}$$

$$n_i = \frac{v_i}{\sigma_i}\,, \tag{10}$$

the Mahalinobis Cosine is

$$D_{MahCo\sin e}(u, v) = \cos(\theta_{mn}) = \frac{|m||n|\cos(\theta_{mn})}{|m||n|} = \frac{mn}{|m||n|}\,. \tag{11}$$

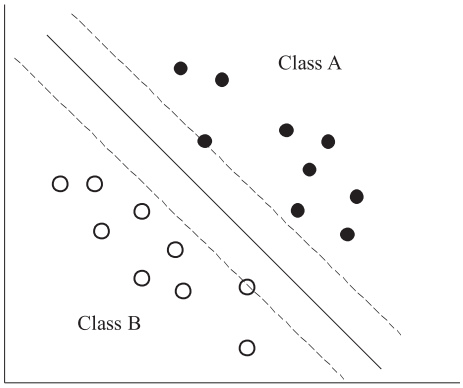

Fig. 2. Separation of data using SVM

Fig. 3. Example of an image after normalization

LDASoft [3]

LDASoft is LDA specific distance metrics. It is similar to the Euclidean measure computed in Mahalinobis space with each axis weighted by generalized eigenvalue λ (also used to compute LDA basis vectors) raised to the power of 0.2. There is a considerable discussion of this setup in Wen Yi Zhao's dissertation [16].

$$D_{LDAsoft}(\boldsymbol{u}, \boldsymbol{v}) = \sum_i \lambda_i^{0.2}(u_i - v_i)^2. \tag{12}$$

2.3 Support Vector Machines

SVM belongs to kernel methods [16]. Kernel algorithms map data from an original space into a higher dimensional feature space using non-linear mapping [17]. An original algorithm from the original space is used in the feature space. Although the high-dimensional space increases the difficulty of the problem (curse of dimensionality), a trick for computing the scalar products in the feature space exists. Computation of the scalar product between two feature space vectors can be done using kernel functions. Using kernel functions, the feature space need not be computed explicitly.

The SVM method was originally developed as a linear classifier [18]. Later it was modified utilizing kernel methods so that it allows also non-linear mapping of data to the feature space. The principle of data separation by SVM is demonstrated on a simplified example in Fig. 2.

SVM separates p-dimensional data using $p-1$ dimensional decision surface (hyperplane) in such a way that it maximizes the margin of the data sets. The margin is defined as the minimal distance of a sample to the decision surface [16]. The distance of the decision surface (the solid line in Fig. 2) from the nearest appearance of the individual data sets should be as large as possible. In Fig. 2, the dashed lines that are parallel with the hyperplane contain support vectors.

In our tests we use SVM with the RBF (radial basis function) kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right), \ \gamma > 0\,, \tag{13}$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j$ are data points from the original space. It is important to find optimal parameters γ (gamma) and C because different parameter setups are suitable for solving different problems [19]. $C > 0$ is the penalty parameter of the error term used in determination of a separating hyperplane with the maximal margin in higher dimensional space by SVM.

3 IMAGE DATABASE

Our tests were performed on a group of 155 greyscale images selected from FERET image database. Gray FERET database [2] contains more than 11000 greyscale images; image size is 256×384 pixels. The images of subjects differ in head position, lightning conditions, beard, glasses, hairstyle, *etc.* We selected images of 10 men and 10 women mostly of Caucasian type, but our image set contains also some Asian or African face types. All of the images are frontal images but with different face expressions (smile, neutral expression) and other signs (with/without glasses, beard, different haircut ...). After pre-processing the image size was 65×75 pixels.

Pre-processing includes five steps of converting a PGM FERET image to a normalized image. The normalization schedule is

- Integer to float conversion — converts 256 grey levels into floating point equivalents.
- Geometric normalization — aligns image according to manually found eye coordinates.
- Masking — crops the image using an elliptical mask and image borders such that only the face from forehead to chin and cheek to cheek is visible.
- Histogram equalization — equalizes the histogram of the unmasked part of the image.
- Pixel normalization — scales the pixel values in order to have a zero mean and unit standard deviation.

Figure 3 shows an example of image after pre-processing. Using pre-processing, we avoid undesirable effects such as "T-shirts recognition" or "haircut recognition" — *ie* we avoid recognition based on non-biometric data such as subject wearing the same T-shirt or having the same haircut on multiple images.

We used 3 different sets of images for training, *ie*, 2, 3 and 4 images per subject in the training set, while the rest of images from the set were used for testing. Figure 4 shows a training set with 2 images per subject.

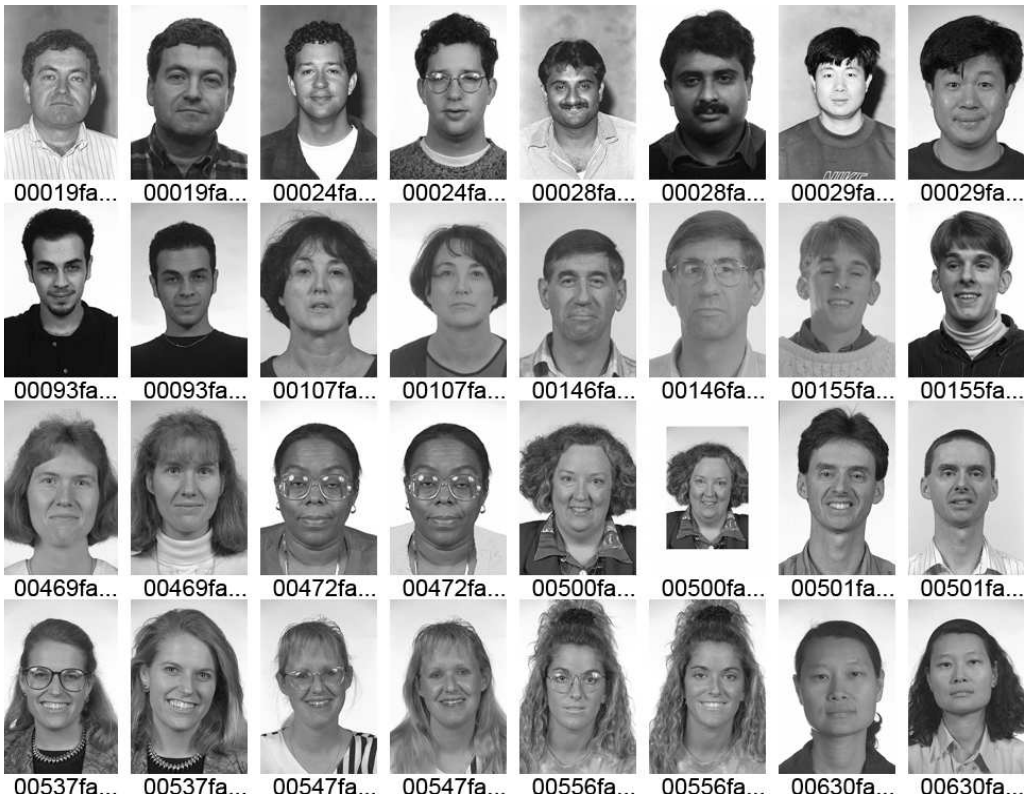

Fig. 4. Example of a training set of 20 images — 10 men, 10 women with 2 images per subject

Table 1. Recognition success rate and optimal SVM parameter setups for used training sets.

Training set	2/subj.	3/subj.	4/subj.
C	0.0312	8.0	32.0
gamma	0.5	$1.220703125 \times 10^{-4}$	$3.0517578125 \times 10^{-5}$
cross-validation rate	45.0 %	70.0 %	87.5 %
successful recognition rate	51.3043 %	90.5263 %	94.6667 %

4 SIMULATION TOOLS

In this work we used software The CSU Face Identification Evaluation System (csuFaceIdEval), Version 5.0 [3] for experiments with PCA and LDA and libsvm [4] for tests with SVM.

4.1 CSU FaceIdEval

This software was developed at the Colorado State University. It is a complex software which implements several algorithms for face recognition, compares them and performs also pre-processing of the images. We used only PCA and the LDA methods (LDA is an implementation of LDA algorithm using Fisher's Linear Discriminants). More about CSU FaceIdEval can be found in [3].

4.2 libsvm

libsvm — A Library for Support Vector Machines [4] (version 2.8. from April 2006) is an integrated software for support vector classification (C-SVC, nu-SVC), regression (epsilon-SVR, nu-SVR) and distribution estimation (one-class SVM). It includes also multi-class classification.

5 EXPERIMENTS AND RESULTS

5.1 Face Recognition Methods

We examined 5 different setups of face recognition experiments. They contain both one-stage and two-stage recognition systems as shown in Fig. 5. All 5 setups are significantly influenced by different settings of parameters that are related to the algorithm used (*ie*, PCA, LDA or SVM). Figure 5, route a), shows one-stage face recognition. In this case SVM is used for classification (*ie*, there is no feature extraction performed). As it will be shown later, this setup does not achieve good success rate for 2 and 3 images per person in training set. Routes b) to e) in Fig. 5 show two-stage face recognition setups including

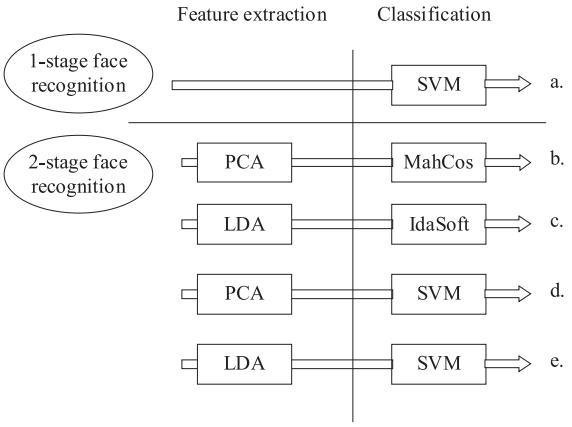

Fig. 5. Methods and classifiers used in our experiments

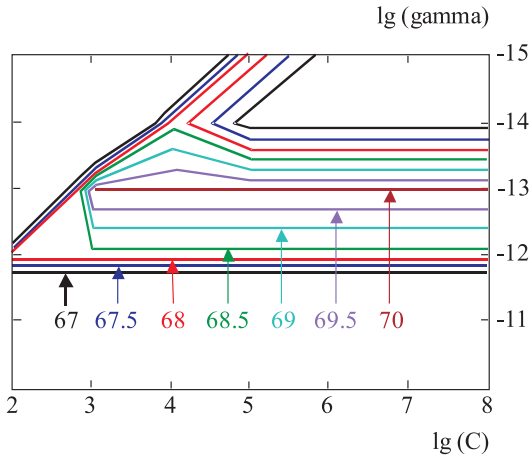

Fig. 6. Example of the output graph — dependence of cross validation rate from the parameters C and gamma for training set with 3 images per subject

both feature extraction and classification. Figures 5b and 5c are standard setups where PCA and LDA are used for feature extraction and MahCos and ldaSoft metrics are used for classification. We propose optimal parameter setups for the best performance of these methods.

At last, we combined PCA and LDA feature extraction with SVM classifier. This is shown in Figs. 5d and 5e, respectively. As we show later, this setup in general does not increase success in recognition rate significantly. On the other hand, it has significant positive influence on robustness and consistency of results across different parameter setups. The most significant improvements were observed in the case of LDA+SVM according to Fig. 5e.

5.2 SVM — One-stage Recognition

For one-stage recognition, we used SVM directly for recognizing faces (see Fig. 5a). Images of size 65 × 75 pixels were used as input for SVM. We created three sets of data each including 2, 3 and 4 images per subject in the training set, while all remaining images were used for testing purposes.

It is important to find optimal parameters C and gamma because different parameter setups are suitable for solving different problems [19]. We used methodology from [19], *ie*, parameter search where the best cross-validation rate performed on training data suggests also the best parameter setup. Figure 6 shows an example of the graph we used for parameter search — the dependence of cross validation rate on parameters C and gamma. The best parameter setups for all training sets are shown in Tab. 1.

More images per subject in the training set resulted in a better recognition rate. In the case of 2 images per subject, the success rate was only slightly above 50 %, while in the case of 4 images per subject the success rate was 94.7 % (in this case 4 incorrectly recognized images). We also tried to find better C and gamma setup manually but we did not manage to find a better performing parameter setup.

5.3 PCA and LDA

For PCA and LDA classifiers followed by MahCos and ldaSoft classifiers (see Figs. 5b and 5c), the results of recognition are significantly influenced by parameters "Dropped from front" and "CutOff".

Dropped from front — denotes number of eigenvectors cut from the beginning of transformation matrix (vectors belonging the highest eigenvalues). These vectors will not be used by image projection to PCA (or LDA) feature space. Reason to truncate these vectors is that it is assumed that these vectors do not correspond to useful information such as lighting variations [3]. Our tests were performed for "Dropped from front" values 0, 1, 2, 3, and 4.

CutOff — represents how many vectors remain in the transformation matrix. Reason to truncate the basis vectors from the end of the matrices (vectors corresponding to the lowest eigenvalues) is to lower the computation difficulty and to eliminate unnecessary information that correlates with noise — and as such is meaningless for recognizing faces [3]. Our tests were performed for CutOff parameter set to 20 %, 40 %, 60 %, 80 % and 100 %.

Again, the experiments were performed using 3 training sets with 2, 3 and 4 images per subject. We determined maximum and minimum recognition rate depending on number of the images per subject in the training set and different parameter setups. This allowed us to propose the optimal parameter setups for different sizes of training sets.

5.4 Summary of Experiments and Results

For each method we tested 25 different parameter setups on 3 different training sets. Figure 18 shows the summary of our experiments. The best performing setups of parameters for PCA and LDA are: CutOff: 60 % or 80 %, and Dropped from front: 0 or 1. Based on these experiments, we can formulate several conclusions

PCA, classifier: MahCosine

Cutoff	Dropped from front 0	1	2	3	4
20%	60	63.5	67.8	67	61.7
40%	81.7	89.6	86.1	86.1	84.3
60%	87.8	90.4	84.3	87	81.7
80%	87.8	90.4	87	86.1	82.6
100%	87	87.8	83.5	82.6	80.9

Max: 90.4 Min: 60 Avg: 81.4

PCA, classifier: MahCosine

Cutoff	Dropped from front 0	1	2	3	4
20%	80	82.1	81.1	78.9	78.9
40%	92.6	91.6	91.6	92.6	91.6
60%	90.5	92.6	92.6	92.6	90.5
80%	90.5	91.6	91.6	92.6	91.6
100%	90.5	90.5	88.4	89.5	87.4

Max: 92.6 Min: 78.9 Avg: 89.0

PCA, classifier: MahCosine

Cutoff	Dropped from front 0	1	2	3	4
20%	94.7	92	92	86.7	89.3
40%	94.7	96	93.3	94.7	94.7
60%	96	96	94.7	96	94.7
80%	94.7	94.7	94.7	94.7	94.7
100%	94.7	93.3	93.3	93.3	93.3

Max: 96 Min: 86.7 Avg: 93.9

LDA, classifier: LdaSoft

Cutoff	Dropped from front 0	1	2	3	4
20%	87	86.1	84.3	82.6	71.3
40%	87	86.1	84.3	82.6	71.3
60%	33	86.1	20.9	84.3	74.8
80%	54.8	66.1	56.5	48.7	40
100%	80.9	67.8	74.8	65.2	62.6

Max: 87 Min: 20.9 Avg: 69.6

LDA, classifier: LdaSoft

Cutoff	Dropped from front 0	1	2	3	4
20%	93.7	90.5	87.4	83.2	78.9
40%	92.6	94.7	87.4	85.3	82.1
60%	95.8	95.8	94.7	91.6	90.5
80%	28.4	54.7	44.2	37.9	25.3
100%	84.2	86.3	78.9	84.2	70.5

Max: 95.8 Min: 25.3 Avg: 77.6

LDA, classifier: LdaSoft

Cutoff	Dropped from front 0	1	2	3	4
20%	96	94.7	96	89.3	89.3
40%	97.3	96	94.7	92	92
60%	97.3	97.3	96	96	96
80%	33.3	44	93.3	97.3	97.3
100%	92	93.3	84	92	78.7

Max: 97.3 Min: 33.3 Avg: 89.0

PCA, classifier: SVM

Cutoff	Dropped from front 0	1	2	3	4
20%	60	59.13	60.87	62.61	56.52
40%	73.04	78.26	83.48	75.65	70.43
60%	83.48	81.74	78.26	80	80.87
80%	90.43	86.09	86.09	83.48	83.48
100%	92.17	92.17	89.57	88.7	88.7

Max: 92.17 Min: 56.52 Avg: 78.6

PCA, classifier: SVM

Cutoff	Dropped from front 0	1	2	3	4
20%	85.26	85.26	80	82.11	75.79
40%	96.84	93.68	91.58	90.53	90.53
60%	96.84	96.84	91.58	91.58	84.21
80%	94.74	96.84	90.53	93.68	89.47
100%	90.53	90.53	86.32	82.11	78.95

Max: 96.84 Min: 75.79 Avg: 89.1

PCA, classifier: SVM

Cutoff	Dropped from front 0	1	2	3	4
20%	96	94.67	96	93.33	89.33
40%	97.33	97.33	96	94.67	93.33
60%	97.33	97.33	97.33	97.33	97.33
80%	97.33	97.33	96	93.33	93.33
100%	97.33	97.33	96	96	96

Max: 97.33 Min: 89.33 Avg: 95.8

LDA, classifier: SVM

Cutoff	Dropped from front 0	1	2	3	4
20%	81.74	80	83.48	79.13	74.78
40%	81.74	80	83.48	79.13	74.78
60%	80.87	87.83	76.52	80	80.87
80%	88.7	86.96	83.48	83.48	80
100%	85.22	79.13	86.96	80.87	78.26

Max: 88.7 Min: 74.78 Avg: 81.5

LDA, classifier: SVM

Cutoff	Dropped from front 0	1	2	3	4
20%	92.63	91.58	87.37	90.53	83.16
40%	95.79	90.53	89.47	85.26	87.37
60%	96.84	96.84	92.63	94.74	91.58
80%	88.42	96.84	96.84	88.42	92.63
100%	91.58	89.47	95.79	93.68	85.26

Max: 96.84 Min: 83.16 Avg: 91.4

LDA, classifier: SVM

Cutoff	Dropped from front 0	1	2	3	4
20%	94.67	96	94.67	90.67	93.33
40%	97.33	97.33	94.67	94.67	86.67
60%	97.33	97.33	96	98.67	93.33
80%	100	97.33	97.33	98.67	98.67
100%	97.33	93.33	94.67	94.67	86.67

Max: 100 Min: 86.67 Avg: 95.3

Legend: above 90% of images successfully recognized; 80 - 90% of images successfully recognized

Fig. 7. Results of experiments for PCA, LDA, PCA+SVM and LDA+SVM

1. The more images per person in the training set, the higher recognition rate is achieved.
2. PCA in Fig. 5b generally performs better than LDA with small training data sets.
3. LDA in Fig. 5c achieves higher maximum recognition rate.
4. LDA with ldaSoft used as classifier (Fig. 5c), despite achieving higher maximum recognition rate, produces very inconsistent results and thus it is hard to suggest best performing parameter setup.
5. Feature extraction by PCA followed by SVM classifier (Fig. 5d) contributes to maximum recognition rate and thus slightly improves performance when comparing with PCA+MahCos setup.
6. Combination of LDA and SVM (Fig. 5e) improves maximum recognition rate and significantly improves robustness of recognition when compared to LDA+ldaSoft method.

Figure 8 shows maximum recognition rates for all methods and training sets as well as the difference between maximum and minimum recognition rates. This difference indicates the robustness degree of the method.

6 CONCLUSIONS

Our experiments with FERET database imply that LDA+ldaSoft generally achieves the highest maximum recognition rate. On the other hand it can be very instable (*ie*, very sensitive to method settings). Thus, LDA alone is not suitable for practical use. At certain parameter settings LDA produced the worst recognition rates from among all experiments. Experiments with the proposed methods PCA+SVM and LDA+SVM produced a better maximum recognition rate than traditional PCA and LDA methods. Combination LDA+SVM produced more consistent results than LDA alone. Altogether we made more than 300 tests and achieved maximum recognition rates near 100 % (LDA+SVM once actually 100 %). In the future, we plan to expand the image sets and find better generalization for the settings.

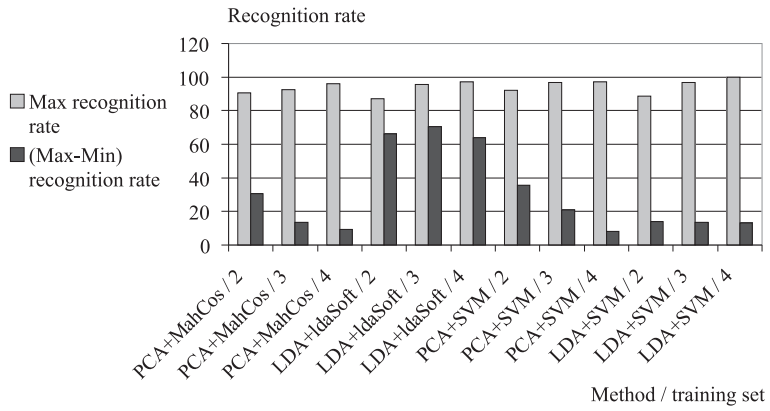

Fig. 8. Maximum recognition rate and difference between maximum and minimum recognition rate for all methods and training sets in our experiments

Acknowledgement

Research described in the paper was done within the grant No. 1/3117/06 of Slovak Grant Agency VEGA .

References

[1] JAIN, A. K.—ROSS, A.—PRABHAKAR, S.: An Introduction to Biometric Recognition, IEEE Trans. Circuits and Systems for Video Technology **14** No. 1 (Jan 2004), 4–20.

[2] FERET Database, http://www.itl.nist.gov/iad/humanid/feret/, NIST, 2001.

[3] BEVERIDGE, R.—BOLME, D.—TEIXEIRA, M.—DRAPER, B.: The CSU Face Identification Evaluation System User's Guide, Version 5.0, http://www.cs.colostate.edu/evalfacerec/algorithms/version5/faceIdUsersGuide.pdf .

[4] LIBSVM — A Library for Support Vector Machine, http://www.csie.ntu.edu.tw/čjlin/libsvm/ .

[5] GRAHAM, A.: A Better Way to Organize Photos?, 2006 http://googleblog.blogspot.com/2006/08/better-way-to-organize-photos.html.

[6] http://riya.wikispaces.com/Face+Recognition, http://www.riya.com/.

[7] EU – Passport Specification, Biometrics Deployment of EU-Passports, Working document (EN) - 28/06/2006, http://ec.europa.eu/justice_home/doc_centre/freetravel/documents/doc/c_2006_2909_prov_en.pdf.

[8] TURK, M.—PENTLAND, A.: Eigenfaces for Recognition, Journal of Cognitive Neuroscience **3** No. 1 (1991), 71–86.

[9] MARCIALIS, G. L.—ROLI, F.: Fusion of LDA and PCA for Face Recognition, Proc. of the Workshop on Machine Vision and Perception, 8th Meeting of the Italian Association of Artificial Intelligence (AI*IA), Siena, Italy, September 10-13, 2002.

[10] MARTINEZ, A. M.—KAK, A. C.: PCA versus LDA, IEEE Trans. Pattern Analysis and Machine Intelligence **23** No. 2 (Feb 2001), 228–233.

[11] WANG, Y.—JIAR, Y.—HU, C.—TURK, M.: Face Recognition Based On Kernel Radial Basis Function Networks, Asian Conference on Computer Vision, Korea, January 27-30, 2004, pp. 174–179, http://www.psych.ucsb.edu/research/recveb/pdfs/4_KRBF.pdf.

[12] YANG, J,—FRANGI, A. F.—YANG, J. Y.—ZHANG, D.—JIN, Z.: KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence **27** No. 2 (Feb 2005), 230–244.

[13] ROTH, A.: Face Recognition, mplab.ucsd.edu/ roth/mypapers/facerec_90734.ppt .

[14] YAMBOR, WENDY S.: Analysis of PCA-Based and Fisher Discriminant-Based Image Recognition Algorithms, http://www.cs.colostate.edu/evalfacerec/papers/tr_00-103.pdf.

[15] ZHAO, W.—CHELLAPPA, R.—PHILLIPS, P. J.: Subspace Linear Discriminant Analysis for Face Recognition, Center for Automation Research, University of Maryland, College Park, Technical Report CAR-TR-914, 1999.

[16] MULLER, K.—MIKA, S.—RATSCH, G.—TSUDA, K.—SCHOLKOPF, B.: An Introduction to Kernel-Based Learning Algorithms, IEEE Transactions on Neural Networks **12** No. 2 (March 2001), 181-201.

[17] ORAVEC, M.—BESZÉDEŠ, M.—ROZINAJ, G.: Detection and Recognition of Human Faces and Facial Features, book chapter in "Speech, Audio, Image and Biomedical Signal Processing Using Neural Networks" (Bhanu Prasad and S. R. Mahadeva Prasanna, eds.), Springer-Verlag, Germany, 2008, pp. 283–306.

[18] VAPNIKV. N.: The Nature of Statistical Learning Theory, Springer, 1995.

[19] HSU, C. W.—CHANG, C. C.—LIN, C. J.: A Practical Guide to Support Vector Classification, 2008 http://www.csie.ntu.edu.tw/čjlin/papers/guide/guide.pdf.

Received 25 January 2008

Ján Mazanec (Bc) received his Bc degree from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in 2006. He is currently studying for master's degree at the Department of Telecommunications. His current research interests include face recognition using PCA and SVM.

Martin Melišek (Bc) received his Bc degree from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in 2006. He is currently studying for master's degree at the Department of Telecommunications. His current research interests include face recognition using LDA and SVM.

Miloš Oravec (Doc, Dr, Ing) received his Ing, Dr, and Doc degrees from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in 1990, 1996, and 2002, respectively. Now, he is Associate Professor at the Department of Telecommunications. His current research interests include neural networks, signal processing and communication networks. He is a member of IET.

Jarmila Pavlovičová (Doc, Dr, Ing) received the MSc, PhD and Assoc. Prof. degrees in telecommunication engineering from the Slovak University of Technology in 1986, 2002 and 2006 respectively. She is with the Department of Telecommunications. Her research interests include image processing, especially image segmentation.