Proceedings of the International Multiconference on
Computer Science and Information Technology pp. 121–128
ISBN 978-83-60810-27-9
ISSN 1896-7094

The development features of the face recognition system

Rauf Sadykhov
United Institute of Informatics Problems
6, Surganov str., Minsk, Belarus
Email: rsadykhov@bsuir.by

Igor Frolov
Belarusian State University
of Informatics and Radioelectronics
6, P.Brovka str., Minsk, Belarus
Email: frolovigor@yandex.ru

***Abstract*—Nowadays personal identification is a very important issue. There is a wide range of applications in different spheres, such as video surveillance security systems, control of documents, forensics systems and etc. We consider a range of most significant aspects of face identification system based on support vector machines in this paper. At first we propose improved face detector to get the region of interest for next face recognition. In paper the technique of face detection jointly image normalization is introduced. We compare three algorithms of feature extraction in application on face identification (PCA NIPALS, NNPCA, kernel PCA). The presented system is intended for process the image with low quality, the photo with the different facial expressions. Our goal is to develop face recognition techniques and create the system for face identification.**

I. Introduction

THE PERSON identification systems are increasingly becoming popular in modern society. The producers of security systems are interested in the new technologies for the automation of the person identification process. This fact is due to rise the level of these systems reliability because of depreciation of the components of used hardware in the designing and the construction of ones.

The range of the biometric systems identification is wide enough, there're the identification on the fingerprints, the iris identification, the face recognition methods and etc. All these technologies are different by the algorithms, methods and techniques that used for the system development. The considerable quantity of solutions are proposed in the field of face recognition and in the sphere of the person identification by photo. Nowadays the development of the automatic personal identification system is a very important issue because of the wide range of applications in different spheres, such as video surveillance security systems, control of documents, forensics systems and etc.

It should be noted that the process of pattern recognition in the field of image processing consists of several required stages before getting final result. There are the preprocessing of the source patterns (the image data processing such as the readjustment of light conditions, the detection of region of interest, the image resizing), the dimension reduction of source data space by data transformation (to remove the noise and to approximate the data), the selection and the implementation of techniques for the data classification.

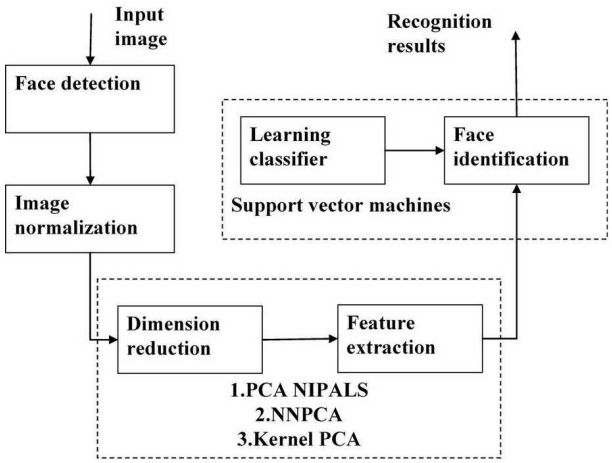


Fig. 1. The structure of face identification system

In this paper we describe the experimental face identification system based on support vector machines (SVM) [1] and we consider some more interesting aspects with designing and constructing of the person system identification by photo. Our system consists of several typical modules (see fig. 1) that are important for the systems of this type such as block of the region of interest (face) detection, block of the image normalization (with the functions of enhancement of brightness and contrast characteristics), the features' extraction block for the dimension reduction of source data space, the module of face recognition (identification) with the functions for the training SVM-classifier and the functions of classification of the processed pattern by the definition of the test image to the certain class.

We researched automated biometric identification systems that were tested on data of the National Institute of Standards and Technology (USA) in particular FERET database. So we propose you to read the results of testing several systems that were developed by different companies (see fig. 2). There are automated portrait identification system "Portrait 2005", developed by specialists LLC "Bars International", LLC "Portland" (Russian Federation) and LLC "ASPI-Soft" (Belarus); automated portrait identification system "Crime Face", developed by RPLLC "Todes" (Belarus); hardware-software complex "Image++"(or"SOVA"), developed by LLP

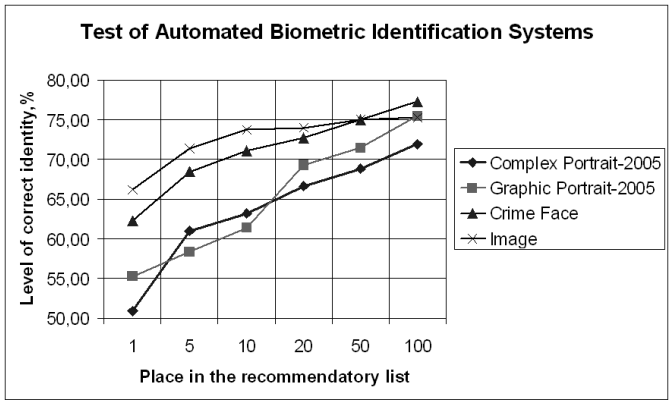


Fig. 2. Test of Automated Biometric Identification Systems

"DANA" (Kazakhstan). Information with test results of automated biometric identification systems that were tested on data of the National Institute of Standards and Technology (USA) is presented at figure 2. In February-July 2006 the State Expert Forensic Center of Ministry of Internal Affairs of Belarus carried out a test of these automated biometric identification systems based on technology of face recognition. The source of this research is [2].

This paper is organized as follows. In the next section the methods of face detection are described. In section III the approaches of the preparation and normalization of the images are introduced. In section IV the neural network approaches as tool for reducing source space of data is considered. The section V contains the description of classifier for pattern recognition based on support vector machines. Finally, the section VI collect some experimental results and brief conclusions.

II. Face detection approaches

A precise detection of face at image strongly simplifies the process of classification. At stage of system development we have realized some experiments and established that at first face detection procedure must be executed due to its importance. The module of image normalization should work at second stage. The results of performance of these procedures in determinate order like this are displayed in a fig. 3 and fig. 4. Presence of background and of facial parts (ears, hair) have effect on obtained results of these experiments. We apply the image enhancement unit in detected region of interest (face in particular) to get more contrasting images that are more suitable for a consequent procedure of reduction of original data space and pattern recognition process. At the beginning the unit of face detection performs detection of facial region using the famous algorithm of Viola-Jones [3]. This method uses an image representation called the "Integral Image" which allows the features used by detector to be computed very quickly [4]. A simple and efficient classifier is built using the AdaBoost learning algorithm [5] to select a small number of critical visual features from a very large set of potential features. Viola-Jones proposed a method for

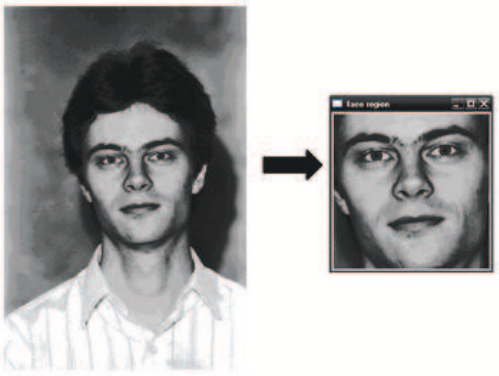


Fig. 3. Image enhancement by "normalization-detect"

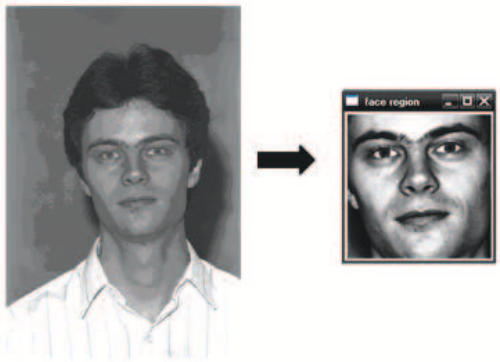


Fig. 4. Image enhancement by "detect-normalization"

combining classifiers in a "cascade" which allows background regions of the image to be quickly discarded while spending more computation on promising face-like regions. However, the results of initial algorithm are not acceptable to further process of face recognition. We can observe a lot of noised data such as background, hair, clothes (see fig. 5).

These elements of image are not interested for process of pattern recognition. To achieve a higher level of authenticity in face recognition it is necessary to select a more narrow region of interest. We have developed the technique to detect region of face only. Our method intends for extracting facial features only without any noised data. It is based on applying of information about human anthropometric measurements. The results of application of presented algorithm are shown in fig. 7. Our approach is based on discrete adaboost ap-

Fig. 5. The face region with the noised data

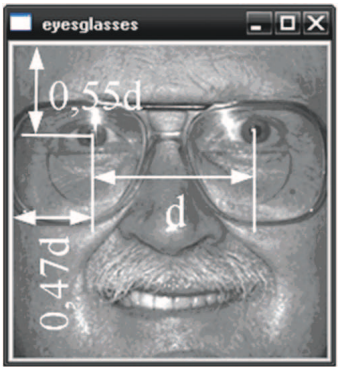
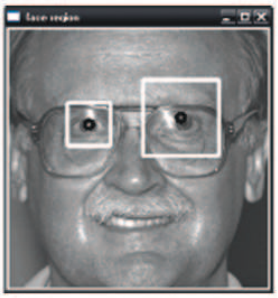
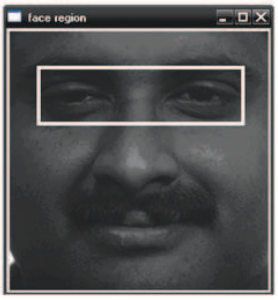


Fig. 6. Facial anthropometric Measurements

Fig. 7. The region of interest





a) b)

Fig. 8. Face detect with: a)- eyeglasses, b)- pair eyes

plication [6] to select simple classifiers based on individual features drawn from a large and over-complete feature set in order to build strong stage classifiers of the cascade. This technique executes the iris area only, not whole face as the previous method. The input image for iris detection procedure is image obtained at the previous stage of image processing that contains parts of clothes, hair etc.

At first we perform the search of left-hand eye area only. The procedure of search of right-hand eye starts if the previous stage was finished successfully. When we get a beneficial effect the distance between pupils in pixels is calculated. At the next step we compute coordinates of left upper point of region of interest. The facial region is presented as squared area with left-upper point calculated at previous stage.

To compute the coordinates of each of four points we use the distance between pupils and some of facial anthropometric data. A length from left iris to upper boundary is calculated as $0,55 \cdot D$ and length from the left iris to the left boundary of the region of interest is equal $0,47 \cdot D$, where D is the distance between pupils in pixels. Thus the length of the side of square of face region is calculated as $2 \cdot 0,47 \cdot D + D = 1,94 \cdot D$. All of these estimated coefficients were obtained experimentally(see fig. 6).

Our approach allows of finding the specified face region on the majority tested images. Failing of iris detection we use additional techniques to solve this problem and to find the region of interest. This technique is enhancement of basic detector and it is designed to find eyeglasses. The current detector was trained to find each part of eyeglasses and after that to detect the iris area. The results of search of eyeglasses are represented in fig. 8(a). The search of region of pair eyes starts after unsuccessful attempt of search of eyeglasses (see fig. 8(b) with results). Our algorithm has found the face regions of interest on all tested images. However, when nothing found at work image we provide the original entry image in the capacity of region of interest.

The region of interest is presented as squared area which contains necessary data (features of face) with minimal noise level to face recognition process. The initial size of facial region has arbitrary size but then we scale it to size of 169×169 pixels. Square side is chosen subject to several important factors. The most important restriction imposed on the procedure of identification is the quality of processed images, primarily due to the resolution image and due to conditions of exposure. The minimum distance between eyes of frontal images that are used at face identification is determined by international standards as “ISO 19794-5:2005” [7], “ANSI/INCITS 385-2004 (Information technology - Face Recognition Format for Data Interchange)” [8]. This parameter is equal 90 pixels. If we keep constraint about minimal distance between eyes then the square side of region of interest is equal about 170×170 pixels for our system face identification. We have chosen the square side size equal 169 pixels subject to some details of image processing with artificial neural networks PCA.

III. Image Enhancement Techniques

At next stage our system performs the procedures of image normalization. We perform an expansion of pixels values to the whole intensity range and the equalization of histogram. The first approach maps the values in intensity image to new values such that values between low and high values in current image map to values between 0 and 1. Thus new pixel values allocate to whole intensity range. After that we perform the histogram equalization which enhances the contrast of images by transforming the values in an intensity image so that the histogram of the output image approximately matches a specified histogram. After use these methods image contains some distortions as sharp face lines. That’s why we apply the median filter to dither face features. This method performs median filtering of the input image in two dimensions. Each output pixel contains the median value in the 3×3 neighborhood around the corresponding pixel in the input image. This part of image processing removes significantly the illumination

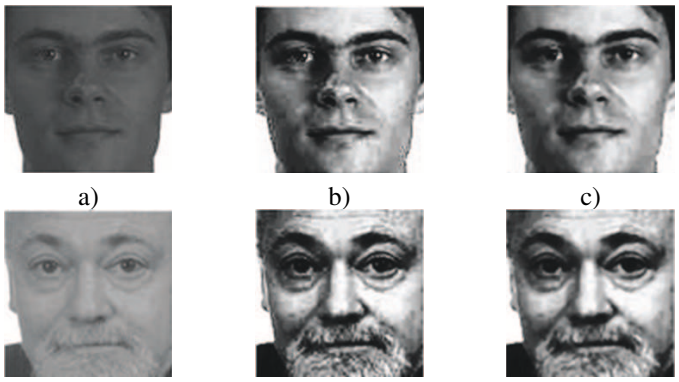


Fig. 9. Examples of normalized images: a - input images, b - after application adjusting image intensity values and histogram equalization, c - after using median filter

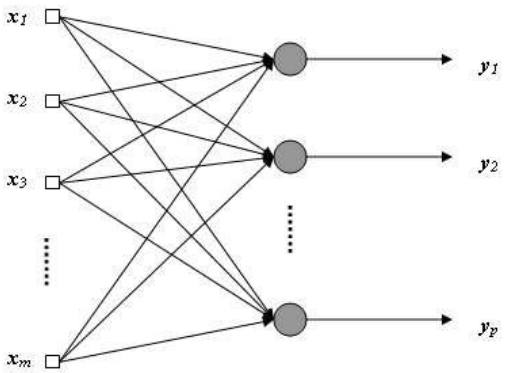


Fig. 10. A single-layered feedforward neural network for extracting p principal components

changes among the images. The fig. 9 illustrates the results of introduction the image pre-processing methods described above. We use these methods in the following sequence: expansion of pixels values to the whole intensity range - the equalization of histogram - median filtering.

IV. Dimension reduction and feature extraction

Some classification-based methods use the intensity values of window images as input features of classifier. However, direct use of intensity values of image pixels are dramatically increases the computation time. On the other hand the huge capacity of data contains many waste data being overfull. In our system we extract features via method of principal component analysis. We use three techniques for implementation of this approach. There are the algorithm NIPALS (non-linear iterative partial least squares) [9] for compute the principal components, the neural network PCA (NNPCA) [13], [16], and the kernel principal component analysis [17]. These methods have different computational cost and various confidence levels at recognition stage. The user of system can choose the method to work by oneself.

Principal Component Analysis (PCA) - is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension. PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

The NIPALS ("Nonlinear Iterative Partial Least Square") algorithm is one of the many methods that exist for finding the eigenvectors (another example is SVD [10]). It was originally made for PCA, but it has been used in other methods as well. Algorithm is used in principal component analysis to decompose a data matrix into score vectors and eigenvectors (loading vectors) plus a residual matrix. This is an overview of the algorithm:

X is a mean centered data matrix, $E_{(0)} = X$. The E-matrix for the zero-th PC (PC_0) is mean centered X, t vector is set to a column in X, t will be the scores for PC_i, p will be the loadings for PC_i, $threshold = 0,00001$, just a low value, to do the convergence check.

Iterations ($i = 1$ to number-of-PCs):

1.Project X onto t to find the corresponding loading p

$$p = (E_{i-1}^T / t^T t) \tag{1}$$

2.Normalise loading vector p to length 1

$$p = p \cdot p(p^T p)^{-0.5} \tag{2}$$

3.Project X onto p to find corresponding score vector t

$$t = (E_{(i-1)} p / (p^T p)) \tag{3}$$

4.Check for convergence. If difference between eigenvalues $\tau_{new} = (t^T t)$ and τ_{old} (from last iteration) is larger than $threshold \cdot \tau_{new}$ return to step 1.

5.Remove the estimated PC component from $E_{(i-1)}$

$$E_{(i)} = E_{(i-1)} - (tp^T) \tag{4}$$

Principal components can be extracted using single-layer feed-forward neural networks [11]. These networks learn unsupervised by using variants of the Hebbian rule. The Generalized Hebbian Algorithm (GHA) [12], also known in the literature as Sanger's rule, is a linear feedforward neural network model for unsupervised learning with applications primarily in principal components analysis. It is similar to Oja's rule in its formulation and stability, except it can be applied to networks with multiple outputs.

$$y_j(n) = \sum_{i=1}^{p} w_{ij} x_i(n), \qquad j = 1, 2, ..., m. \tag{5}$$

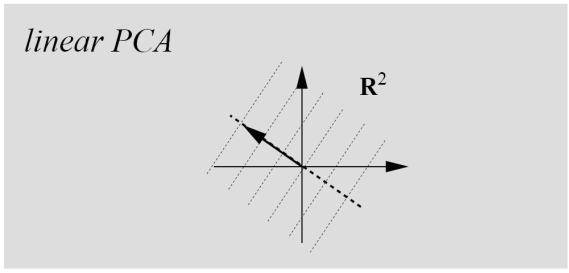


Fig. 11. Linear PCA

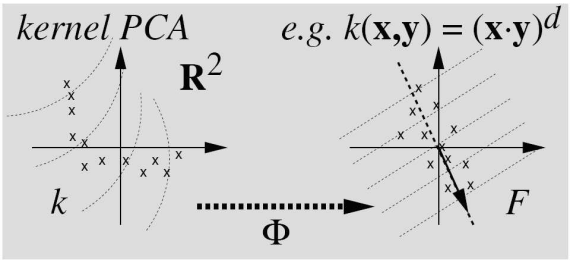


Fig. 12. The basic idea of kernel PCA. In some high dimensional feature space F (see fig. 12 right), we are performing linear PCA, just as a PCA in input space(figure 11). Since F is nonlinearly related to input space (via Φ), the contour lines of constant projections onto the principal Eigenvector (drawn as an arrow) become nonlinear in input space. Note that we cannot draw a pre-image of the Eigenvector in input space, as it may not even exist. Crucial to kernel PCA is the fact that we do not actually perform the map into F, but instead perform all necessary computations by the use of a kernel function k in input space (here: R^2).

$$\Delta w_{ij}(k) = \eta[y_j(n)x_i(n) - y_j(n)\sum_{k=1}^{j} w_{ki}(n)y_k(n)] \quad (6)$$

The neural network PCA (NNPCA) is used in our work. The Generalized Hebbian Algorithm by Sanger [6] is one among the best known learning algorithms that allow a neural network (see fig. 10) to extract a selected number of principal components from a multivariate random process. It applies to a single-layered feedforward neural network that may be described by equation(5) with the rule for updating weights (see equation 6).

The weight of first eigenvector has been estimated and its value lies within the range from 75 to 84%. Therefore, to decrease the computation expenses we used only one eigenvector for calculation one principal component.

Kernel principal component analysis (kernel PCA) [14] is an extension of principal component analysis using techniques of kernel methods. Instead of directly doing a PCA, the original data points are mapped into a higher-dimensional (possibly infinite-dimensional) feature space defined by a (usually non-linear) function Φ through a mathematical process called the "kernel trick":

$$\Phi : \mathbf{R}^N \to F,\ x \to \mathbf{X} \quad (7)$$

The kernel trick [15] transforms any algorithm that solely depends on the dot product between two vectors. Wherever a dot product is used, it is replaced with the kernel function. Thus, a linear algorithm can easily be transformed into a non-linear algorithm. This non-linear algorithm is equivalent to the linear algorithm operating in the range space of Φ. However, because kernels are used, the Φ function is never explicitly computed. This is desirable, because the high-dimensional space may be infinite-dimensional (as is the case when the kernel is a Gaussian).

Like in PCA, the overall idea is to perform a transformation that will maximize the variance of the captured variables while minimizing the overall covariance between those variables. Using the kernel trick, the covariance matrix is substituted by the Kernel matrix and the analysis is carried analogously in feature space. An Eigen value decomposition is performed and the eigenvectors are sorted in ascending order of Eigen values, so those vectors may form a basis in feature space that explain most of the variance in the data on its first dimensions.

However, because the principal components are in feature space, we will not be directly performing dimensionality reduction. Suppose that the number of observations m exceeds the input dimensionality n. In linear PCA, we can find at most n nonzero Eigen values. On the other hand, using Kernel PCA we can find up to m nonzero Eigen values because we will be operating on a $m \times m$ kernel matrix.

Each time the features extracted vector is presented as the sequence of more significant coefficients of the principal components. In our work the size of face region extracted in face detection block is 169×169 pixels. Thus the original dimension of data space counts 28.561 points. We form sequence with 169 features only by use of most important coefficients from process of feature extraction. The second part of data (less significant coefficients) is rejected. Thus we use three different approaches to extract the vector of features from original data set (region of interest that contains a facial image).

V. Face identification with SVMs

The Support Vector Machines (SVMs) [1] present one of kernel-based techniques. SVMs based classifiers [18] can be successfully apply for text categorization, face identification. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers. SVMs are used for classification of both linearly separable (see fig. 13) and unseparable data. SVMs based classifiers can be successfully apply for text categorization, face identification.

Linear classifiers are not complex enough sometimes. SVM solution: map data into a richer feature space including non-linear features, then construct a hyperplane in that space so all other equations are the same. Basic idea of SVMs is creating the optimal hyperplane and calculating the decision function for linearly separable patterns. This approach can be extended to patterns that are not linearly separable by transformations of

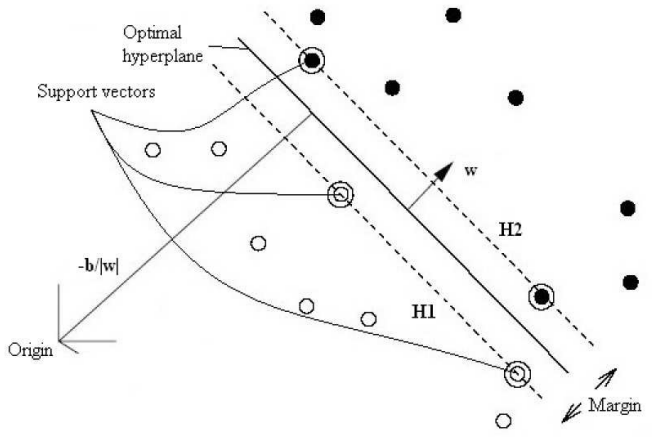


Fig. 13. Linear separating hyperplanes for the separable case.

original data to map into new space due to using "kernel trick". In the context of the Fig. 13, illustrated for 2-class linearly separable data, the design of the conventional classifier would be just to identify the decision boundary w between the two classes. However, SVMs identify support vectors (SVs) $H1$ and $H2$ that will create a margin between the two classes, thus ensuring that the data is "more separable" than in the case of the conventional classifier.

Suppose we have N training data points $(x1, y1), (x2, y2), \ldots, (x_N, y_N)$ where $x_i \in \Re^d$ and $y_i \in \pm 1$. We would like to learn a linear separating classifier:

$$f(x) = sgn(w \cdot x - b) \tag{8}$$

Furthemore, we want this hyperplane to have the maximum separating margin with respect to two classes. Specifically, we wish to find this hyperplane $H : y = w \cdot x - b$ and two hyperplanes parallel to it and with equal distances to it:

$$H_1 : y = w \cdot x - b = +1 \tag{9}$$

$$H_2 : y = w \cdot x - b = -1 \tag{10}$$

with the condition that there are no data points between H_1 and H_2, and the distance between H_1 and H_2 is maximized.

For any separating plane H the corresponding H_1 and H_2 we can always "normalize" the coefficients vector w so that H_1 will be $y = w \cdot x - b = +1$, and H_2 will be $y = w \cdot x - b = -1$ as shown [1].

We want to maximize the distance between H_1 and H_2. So there will be some positive examples on H_1 and some negative examples on H_2. These examples are called support vectors because only they participate in the definition of the separating hyperplane, and other examples can be removed and moved around as long as they do not cross the planes H_1 and H_2.

In the space the distance from a point on H_1 to $H : w \cdot x - b = 0$ is $|w \cdot x - b|/||w|| = 1/||w||$, and the distance between H_1 and H_2 is $2/||w||$. Thus, to maximize the distance we should minimize $||w|| = w^T w$ with the condition that there are no data points between H_1 and H_2:

$$w \cdot x - b \geq +1, \text{ for positive example } y_i = +1 \tag{11}$$

$$w \cdot x - b \leq -1, \text{ for negative example } y_i = -1 \tag{12}$$

These two conditions can be combined into

$$y_i \cdot (w \cdot x_i - b) \geq 1 \tag{13}$$

So, this problem can be formulated as

$$\min_{w,b} \frac{1}{2} w^T w \text{ subject to } y_i \cdot (w \cdot x_i - b) \geq 1 \tag{14}$$

This is a convex quadratic programming problem (in w, b) in convex set.

Introducing Lagrange multipliers $\alpha_1, \alpha_2, \ldots, \alpha_N \geq 0$, we have the following Lagrangian:

$$L(w, b, \alpha) \equiv \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i y_i (w \cdot x_i - b) + \sum_{i=1}^{N} \alpha_i \tag{15}$$

We can solve the wolfe dual insread: maximize $L(w, b, \alpha)$ with respect to α subject to constrains that the gradient of $L(w, b, \alpha)$ with respect to the primal variables w and b vanish:

$$\frac{\partial l}{\partial w} = 0 \tag{16}$$

$$\frac{\partial l}{\partial b} = 0 \tag{17}$$

and that $\alpha \leq 0$

From equations (16) and (17) we have

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{18}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{19}$$

Substitute them (18), (19) into $L(w, b, \alpha)$ we have

$$L_D \equiv \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j \tag{20}$$

in which the primal variables are eliminated.

When we solve α_i, we can get $w = \sum_{i=1}^{N} \alpha_i y_i x_i$ and we can classify a new object x with:

$$\begin{aligned} f(x) &= sgn(w \cdot x + b) \\ &= sgn((\sum_{i=1}^{N} \alpha_i y_i x_i) \cdot x + b) \\ &= sgn(\sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x) + b) \end{aligned} \tag{21}$$

Note that in the objective function and solution, the training vector x_i is occurred only in the form of dot product.

If the surface separating the two classes are not linear we can transform the data points to another high dimensional space such that the data points will be linearly separable [19]. Let the transformation be $\Phi(\cdot)$. In the high dimensional space, we solve

$$L_D \equiv \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \tag{22}$$

Suppose, in addition, $\Phi(x_i) \cdot \Phi(x_j) = k(x_i \cdot x_j)$. That is, the dot product in that high dimensional space is equivalent to a kernel function of the input space. So we need not be explicit about the transformation $\Phi(\cdot)$ as long as we know that the kernel function $k(x_i \cdot x_j)$ is equivalent to the dot product of some other high dimensional space. There are many kernel functions that can be used this way, for example, the radial basis function (Gaussian kernel):

$$K(x_i, x_j) = e^{-||x_i - x_j||^2/2\sigma^2} \tag{23}$$

Formally, preprocess the data with $\Phi : \Re^N \rightarrow F$, then a data set that is not linearly separable in the input data space (as in the left hand side of fig. 14) is separable in the nonlinear feature space (right hand side of fig. 14) defined implicitly by the non-linear kernel function.

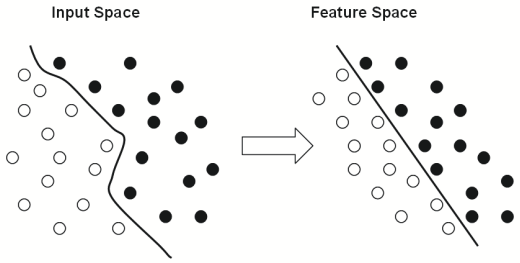


Fig. 14. Kernel trick for the unseparable case

For multi-class classification we use the "one-against-one" approach in which $k \cdot (k-1)/2$ classifiers are constructed and each one trains data from two different classes. In classiffication we use a voting strategy: each binary classiffcation is considered to be a voting where votes can be cast for all data points x - in the end point is designated to be in a class with maximum number of votes. We implemented probabilistic approach to identify the processed pattern by calculating the confidence level for this face. And we construct a list of 6 samples by descending to make a final decision.

Basic idea of SVMs relative to the Nearest Neighbor [20] approach is creating the optimal hyperplane and calculating the decision function for linearly separable patterns. This approach can be extended to patterns that are not linearly separable by transformations of original data to map into new space due to using kernel trick.

To train the SVM, we search trough the feasible region of the dual problem and maximize the objective function. The optimal solution can be checked using the Karush-Kuhn-Tucker (KKT) conditions [1].

The KKT optimality conditions of the primal problem are

$$\alpha_i[y_i(w^T x_i - b) + \xi_i - 1] = 0 \tag{24}$$

$$\sum_{i=1}^{N} \mu_i \xi_i = 0 \tag{25}$$

To solve this quadratic programming problem we used the sequential minimal optimization (SMO)-type decomposition method[21] for support vector machines [22].

The SMO algorithm searches through the feasible region of the dual problem and maximizes the objective function

$$L_D \equiv \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \tag{26}$$

$$0 \leq \alpha_i \leq C, \quad \forall i$$

It works by optimizing two α_i's at time (with the other α_i's fixed) and uses heuristics to choose the two α_i's for optimization [22].

The decision function is

$$sgn(\sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b). \tag{27}$$

VI. Experiment results

Our system contains two basic blocks. There are training SVM-classifier module and face identification unit developed on SVM-classifier. At first we have to create the model for following pattern recognition. At this stage we train our SVM-classifier by the algorithm proposed Jones C.Platt [22]. In our system we used the libsvm implementation [23] of this algorithm. The one-type input feature vector containing the significant coefficients from PCA is used both for train and classification.

Scaling data before applying SVM is very important. [24] explains why we scale data while using Neural Networks, and most of considerations also apply to SVM. The main advantage is to avoid attributes in greater numeric ranges dominate those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel, large attribute values might cause numerical problems. We linearly scale each attribute to the range $[0; 1]$. Of course we use the same method to scale testing data before testing.

To increase the level of correct identity we applied choose the parameters of C-support vector classification with cross validation via parallel grid search. There are two parameters while using RBF kernels: C and γ. It is not known beforehand which C and γ are the best for one problem; consequently some kind of model selection (parameter search) must be done. Therefore, a common way is to separate training data into two parts of which one is considered unknown in training the classifier. Then the prediction accuracy on this set can more precisely reflect the performance on classifying unknown data. An improved version of this procedure is cross-validation.

In v-fold cross-validation, we first divide the training set into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining $v - 1$ subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

We used the sample collection of images with size $512 - by - 768$ pixels from database FERET [25] containing 100 classes (unique persons) to test our face recognition system

TABLE I
RESULTS OF TESTING PERSON IDENTIFICATION SYSTEM

	Recognition rate, percent	Feature extraction time for each vector, s	Training time, s
PCA NIPALS	80	0,6	28,4
NNPCA	84	12	28,8
Kernel PCA	81	0,8	28,3

based on support vector machines. This collection counts 300 photos. Each class was presented by 3 images. So, to train SVM-classifier we used 200 images where 2 photos introduced each class. 100 images were used to test our system. Note, that any image for testing doesn't use in training process. The results of realized experiments are shown in the table I. In this paper we proposed an efficient face identification system based on support vector machines. This system performs several algorithms to ensure the full process of pattern recognition. Thus, our system is intended for face identification by processing the image even low quality. The face detection region procedure without any noise is a very important stage of the person identification process. The angle of inclination and the rotation angle of head influence on the level of validity of recognition. These factors are the most significant in person identification system.

REFERENCES

[1] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", *Data Mining and Knowledge Discovery*, vol. 2, 1998, pp.121-167.

[2] http://www.portret.tomsk.ru/index.php?page=informations&subject=gabitoskopia

[3] P. Viola, M. J. Jones, "Robust Real-Time Face Detection", *International Journal of Computer Vision*, vol. 57 (2), 2004, pp.137-154.

[4] Bae, H. and S. Kim, "Real-time face detection and recognition using hybrid-information extracted from face space and facial features", *Image and Vision Computing*, vol. 23, 2005, pp.1181-1191.

[5] K. Tieu, P. Viola, "Boosting image retrieval", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[6] R. E. Schapire, Y. Freund, "A short introduction to boosting", *Journal of Japan Society for Artificial Intelligence*, vol. 5 (14), 1999, pp.771-780.

[7] http://www.iso.org/iso/catalogue_detail.htm?csnumber=38749

[8] http://webstore.ansi.org/RecordDetail.aspx?sku=ANSI+INCITS+385-2004

[9] H. Risvik, "Principal Component Analysis (PCA) & NIPALS algorithm", http://folk.uio.no/henninri/pca_module/pca_nipals.pdf, 2007.

[10] Wall, E. Michael, A. Rechtsteiner, L. M. Rocha, "Singular value decomposition and principal component analysis", *in A Practical Approach to Microarray Data Analysis*, 2003, pp. 91–109.

[11] Oja, Erkki, "Simplified neuron model as a principal component analyze". *Journal of Mathematical Biology*, vol.15 (3), 1982, pp. 267–273.

[12] S. Haykin, "Neural Networks: A Comprehensive Foundation (2 ed.)", Prentice Hall, 1998.

[13] T. D. Sanger, "Optimal Unsupervised Learning in A Single-Layer Linear Feedforward Neural Network", *Neural Networks*, vol. 2, 1989, pp. 459-473.

[14] B. Scholkopf1, A. Smola, K.R. Muller, "Kernel Principal Component Analysis", http://cseweb.ucsd.edu/classes/fa01/cse291/kernelPCA_article.pdf

[15] M. Aizerman, E. Braverman, and L. Rozonoer, "Theoretical foundations of the potential function method in pattern recognition learning", *Automation and Remote Control*, 1964, pp. 821–837.

[16] S. Knerr, L. Personnaz, G. Dreyfus, "Single-layer learning revisited: a stepwise procedure for building and training a neural network", *In J. Fogelman, editor, Neurocomputing: Algorithms, Architectures and Applications*, 1990, Springer-Verlag.

[17] K. Varmuza, P. Filzmoser, "Introduction to Multivariate Statistical Analysis in Chemometrics", 2009, p. 321.

[18] V. Vapnik, "Universal Learning Technology: Support Vector Machines", *NEC Journal of Advanced Technology*, vol. 2, 2005, pp. 137–144.

[19] E. Osuna, R. Freund, and F. Girosi, "An Improved Training Algorithm for Support Vector Machines", *Proceedings IEEE Neural Networks for Signal Processing VII Workshop*, 1997, pp. 276–285.

[20] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, A. Y. Wu, "An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions", *Journal of the ACM*, vol. 45(6), 1998, pp. 891–923.

[21] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training SVM", *Journal of Machine Learning Research*, 2005, http://www.csie.ntu.edu.tw/ cjlin/papers/quadworkset.pdf.

[22] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines", *Technical Report MSR-TR-98-14 Microsoft Research*, 1998, p.21.

[23] C. W. Hsu, C. C. Chang, C. J. Lin, "A practical guide to support vector classification", http://www.csie.ntu.edu.tw/c̆jlin.

[24] W. S. Sarle, "Neural Network FAQ. Periodic posting to the Usenet newsgroup comp.ai.neural-nets", 1997.

[25] http://www.face.nist.gov