Pyramidal Algorithm for SVM-Classification

Igor Frolov[1], Rauf Sadykhov[2]
[1]BSUIR, Minsk, Belarus
[2]UIIP, Minsk, Belarus
[1]*frolovigor@yandex.ru*, [2]*rsadykhov@bsuir.by*

Abstract— **In this work we consider pyramidal decomposition algorithm for support vector machines classification problem. This algorithm is proposed to solve multi-class classification problem with use some binary SVM-classifiers for the strategy "one-against-one''.**

Keywords— support vector machines; face recognition; decomposition algorithm

I. INTRODUCTION

One of the most problems of computer vision is face recognition with 2D-photograph. The task of face recognition has a lot of solutions and it's based on similar basic principles of image processing and pattern recognition. Our approach is based on well-known methods and algorithms such as PCA and SVM. The process of face recognition consists of several basic stages: face detection, image enhancement of region of interest, image representation as a feature vector, pattern classification with SVM.

II. IMAGE PREPROCESSING

Face detection is the first step of processing in many approaches of face recognition. We use the face-detector trained on our own images based on algorithm of Viola-Jones [1] with use Haar-like features to detect a region of interest on image bounded by lines of brows (see **Ошибка! Источник ссылки не найден.**).

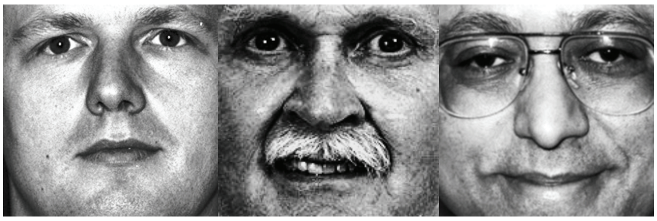

Figure 1. Region of interest bounded by lines of brows

We perform an extension of pixel range values to the whole intensity spectrum and the equalization of histogram of ROI. In order to reduce the dimensionality of the feature space and extract principle components of image the NIPALS algorithm [2] is used. The SVM-classifier solves the problem of training and classification of images.

III. INTRODUCTION TO SUPPORT VECTOR MACHINES

The Support Vector Machines (SVMs) [3] present one of kernel-based techniques. SVMs classifiers can be successfully apply for text categorization, face recognition. A special property of SVMs is that they simultaneously minimize the empirical classification error and maximize the geometric margin. SVMs are used for classification of both linearly separable (see Figure 2.) and unseparable data.

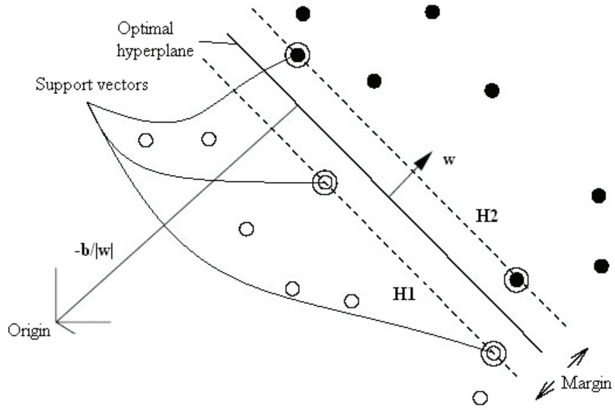

Figure 2. Optimal hyperplane of support vector machines

Basic idea of SVMs is creating the optimal hyperplane 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.

In the context of the Figure 2., 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 $\{(x_1,y_1), (x_2,y_2),\ldots,(x_N,y_N)\}$ where $x_i \in \Re^d$ and $y_i \in \{\pm 1\}$. We would like to learn a linear separating classifier:

$$f(x) = \mathrm{sgn}(w \cdot x - b). \quad (1)$$

Furthemore, we want this hyperplane to have the maximum separating margin with respect to two classes. Specifically, we wish to find this hyperplane $H: y = w \cdot x - b = 0$ and two hyperplanes parallel to it and with equal distances to it:

$$H_1 : y = w \cdot x - b = +1, \quad (2)$$

$$H_2 : y = w \cdot x - b = -1 \quad (3)$$

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 $y = w{\cdot}x - b = -1$.

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 don't cross the planes H_1 and H_2.

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

$$L(w,b,\alpha) \equiv \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i y_i (w \cdot x_i - b) + \sum_{i=1}^{N} \alpha_i \qquad (4)$$

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. Let the transformation be $\Phi(\cdot)$. In the high dimensional space, we solve

$$L_D \equiv \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \qquad (5)$$

Suppose, in addition, $\Phi(x_i)\cdot\Phi(x_j) = k(x_i, x_j)$. 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 $k(x_i, x_j)$ 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).

IV. MULTICLASS CLASSIFICATION WITH SUPPORT VECTOR MACHINES

Support Vector Machines is well-known and reliable technique to solve classification problem. SVMs is method for binary classification.

There are several strategies of use binary classifiers to combine them for multiclass classification. The most famous of them for multiclass SVM-classification are "one-against-one", "one-against-all" and DAGSVM strategies. Each of these approaches has various characteristics and concepts of determination of "winner-class" and distinguished from another methods. All strategies mentioned above consist in dividing the general multiclass problem into minimal two-class problems and use specified procedures of voting.

The "one-against-all" technique [4] builds N binary classifiers to solve N-class problem (see Figure 3.). Each of N binary classifiers train to distinguish one class from all another. In this case each pair of classes has one "winner-class". In validation phase we choose the class which gives maximum of decision function. The i^{th} SVM-classifier is trained with positive label for i^{th} class and with negative label for the rest classes.

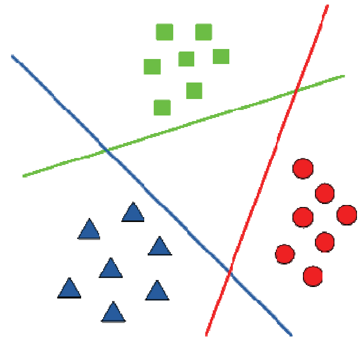

Figure 3. One-against-all strategy of voting

Other basic strategy "one-against-one" calculates all values of possible binary $N(N-1)/2$ SVM-classifiers for N-class problem (see Figure 4.).

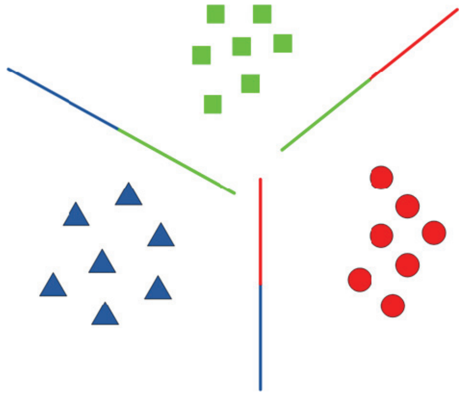

Figure 4. One-against-one strategy of voting

There are several methodologies to choose "winner" in technique "one-against-one". One of them is DAGSVM-strategy that operates Directed Acyclic Graph to estimate the "winner-class" as shown in Figure 5. where each node is associated to a pair of classes and a binary SVM-classifier.

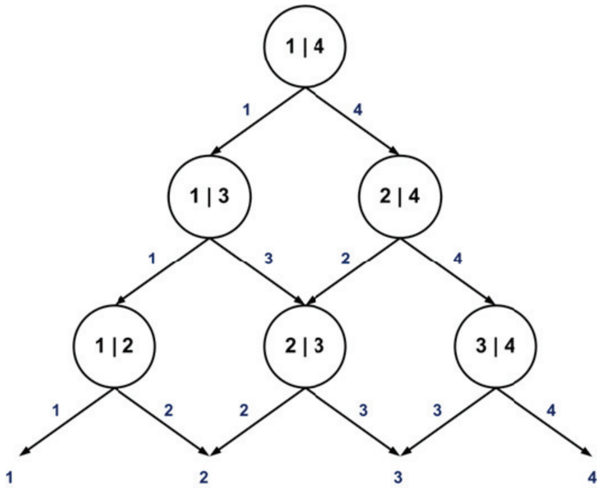

Figure 5. DAGSVM technique

The most effective strategy to identify the target "winner-class" is supposed the voting strategy when the i^{th} class gets a i^{th} vote and the total number of votes for this class is incremented by one. Otherwise total number of votes of j^{th} class is incremented by 1. The recognizable pattern is processed with all binary classifiers. Class-winner is defined as class that scored maximum of votes. The described methodology of detection of "Winner" was called "MaxWin" strategy. In the case when two or more classes have the same number of votes we choose "Winner-class" with the smallest index, although it's not the best decision.

We use the "one-against-one" strategy as the most effective technique to solve multiclassification problem. The source code of SVM-algorithm is implemented in LIBSVM-library [5]. We built the face recognition system with mentioned above methods and algorithms (see Figure 6.).

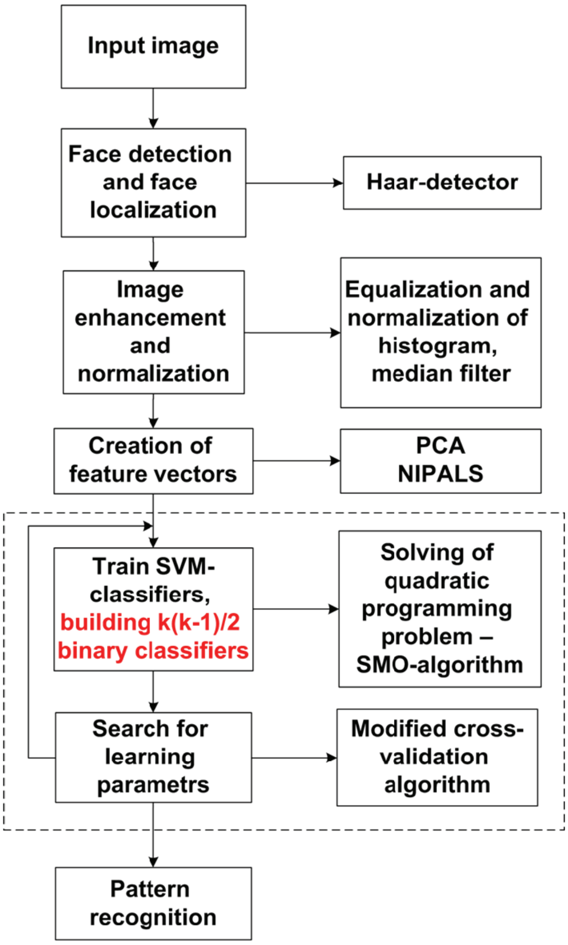

Figure 6. Face recognition system blocks

As in the Figure 6. shown the procedure of train SVM-classifiers is associated with solving of quadratic programming problem and requires searching for learning parameters. Sequential minimal algorithm breaks the optimization problem into an array of smallest possible sub-problems, which are then solved analytically with use special constraints from Karush–Kuhn–Tucker conditions.

Search of optimal to recognize parameters of SVM-classifiers carry out with cross-validation procedure. The searched parameters are the same for all binary SVM-classifiers. Main idea of cross-validation is shown in Figure 7..

Figure 7. Cross-validation algorithm

The search of learning parameters for SVM-classifiers is performed for each binary classifier. The regularization parameter C and parameter γ for radial basis function look for grid as shown in Figure 8..

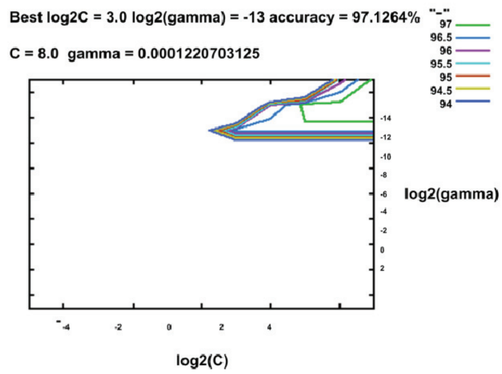

Figure 8. Search learning parameters. Stage 1

This approach of learning parameters is described in [6] and it has a rough estimate because of use a logarithmic scale; we consequently reject the logarithmic scale and we switched over to use immediate value of their bounds. Extreme values of learning parameters are chosen according to boundary points of line corresponding to the highest rate of test recognition (see Figure 9.).

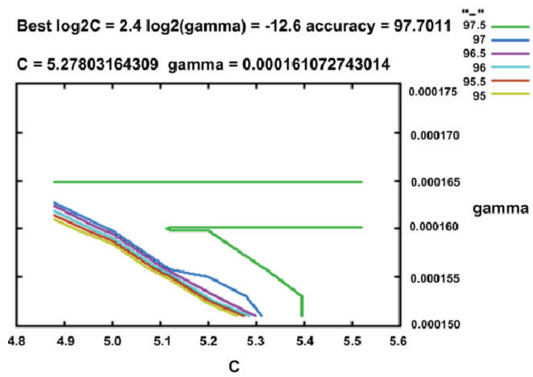

Figure 9. Search learning parameters. Stage 2

To increase the speed of recognition at classification stage we suggested a new scheme of combination binary classifiers to use certain of them to classify patterns (see Figure 10.). At the same time we train all binary SVM-classifiers in learning stage.

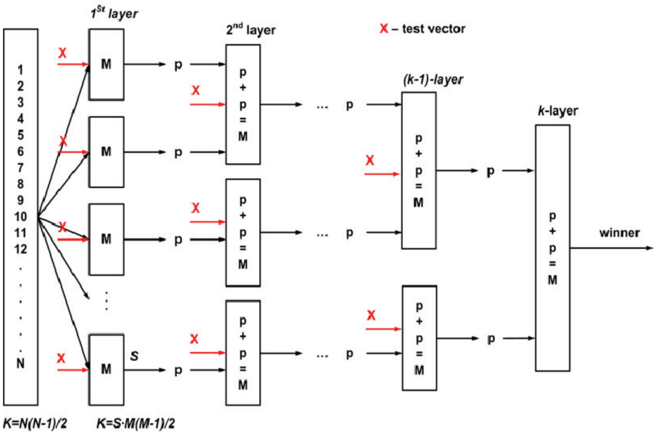

Figure 10. Pyramidal decomposition algorithm for SVM-classification

The general number of binary SVM-classifiers which are used in the scheme "one-against-one" is calculated as shown in equation (6):

$$K = N(N-1)/2 \qquad (6)$$

We carry out the separation of primary training set into M subsets and realize the classification within each of them. The proposed algorithm reduces the number of operations in computing of class-winner. In each subset we use the learned binary SVM-classifiers which correspond to the classes of subset. The quantity of classifiers using in recognition for each layer is computed as

$$K = \frac{N}{M}\cdot\left(\frac{M\cdot(M-1)}{2}\right)+\left(\frac{r\cdot(r-1)}{2}\right), \qquad 3 \le r < M, \qquad (7)$$

$$K = \left(\frac{N}{M}-1\right)\cdot\left(\frac{M\cdot(M-1)}{2}\right)+\frac{(M\cdot(M-1))\cdot(M+r-1)}{2}, \qquad 0 \le r < 3, \qquad (8)$$

where K – is total quantity of binary SVM-classifiers, N – the quantity of binary classifiers for layer, M – a quantity of classes in subset, N/M is rounded to nearest smallest integer, r – the remainder of classes that are not included in subsets.

V. Experiments and Results

We used the sample collection of images with size 512×768 pixels from database FERET [7] containing 611 classes (unique persons) to test our face recognition system based on support vector machines.

TABLE I. Results of Experiments for FERET Database

Recognition rate, %	1st place, %	2nd – 5th place, %	6th – 10th place, %	11th – 50th place, %	51st – 100th place, %	Over 100, %
94,272	80,524	13,748	1,800	2,291	1,146	0,491
93,126	79,051	14,075	1,309	1,964	2,619	2,946
94,435	80,851	13,584	2,291	2,128	0,164	0,982

This collection counts 1833 photos. Each class was presented by 3 images. So, to train SVM-classifier we used 1222 images where 2 photos introduced each class. 611 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 1.

The traditional approach PCA+SVM gives the recognition rate on FERET database gives results at 85% [8] and 95% for ORL database.

On the other hand we evaluated the performance in comparison with traditional algorithms. The results of second speed test are shown in the table 2.

TABLE II. Performance of Multiclass SVM-classification Algorithms

Strategy of multiclassification	***Training time, s***	***Time of 1 face recognition, s***	***Rate of face recognition (1st-5th places), %***
Basic technique "one-against-one"	21,7	1,2	84,8
"One-against-all"	2,8	0,91	85,9
Pyramidal algorithm	21,7	0,029	93,8

VI. Conclusion

In this paper we proposed an efficient technique to combine binary SVM-classifiers, which we called pyramidal decomposition algorithm. This algorithm decreases time of classification and improve index of recognition rate. On the other hand we proposed to use individual learning parameters of binary SVM-classifiers obtained consequently cross-validation. Furthermore we used cross-validation not only with logarithmic scale.

References

[1] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", Computer Vision and Pattern Recognition, 2001 pp. 511-518.

[2] Y. Miyashita, T. Itozawa, H.Katsumi, S.-I. Sasaki "Comments on the NIPALS algorithm", Journal of Chemometrics, vol. 4, Issue 1, 1990, pp. 97–100.

[3] C.J.C. Burges "A Tutorial on Support Vector Machines for Pattern Recognition", Data Mining and Knowledge Discovery, 1998, Vol. 2, pp. 121–167.

[4] C.-W. Hsu, C.-J. Lin "A comparison of methods for multi-class support vector machines", IEEE Transactions on Neural Networks, Vol. 13, 2002, pp. 415–425.

[5] C.-C. Chang "LIBSVM: a library for support vector machines", ACM Transactions on Intelligent Systems and Technology, 2011, http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[6] R. Kohavi "A study of cross-validation and bootstrap for accuracy estimation and model selection", Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Vol. 2(12), pp. 1137–1143.

[7] H. Moon, P.J. Phillips "The FERET verification testing protocol for face recognition algorithms", Proceedings. Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 48–53.

[8] T.H. Le, L. Bui "Face Recognition Based on SVM and 2DPCA", International Journal of Signal Processing, Image Processing and Pattern Recognition, 2011, Vol. 4, No. 3, pp. 85–94.