Combined Classification of Handwritten Digits using the 'Virtual Test Sample Method'

Jörg Dahmen, Daniel Keysers, and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department
RWTH Aachen - University of Technology
D-52056 Aachen, Germany
{dahmen, keysers, ney}@informatik.rwth-aachen.de

Abstract. In this paper, we present a combined classification approach called the 'virtual test sample method'. Contrary to classifier combination, where the outputs of a number of classifiers are used to come to a combined decision for a given observation, we use multiple instances generated from the original observation and a single classifier to compute a combined decision. In our experiments, the virtual test sample method is used to improve the performance of a statistical classifier based on Gaussian mixture densities. We show that this approach has some desirable theoretical properties and performs very well, especially when combined with the use of invariant distance measures. In the experiments conducted throughout this work, we obtained an excellent error rate of 2.2% on the original US Postal Service task.

1 Introduction

In this paper, we present a combined classification approach called the 'virtual test sample method' (VTS), which is based on the idea of using a single classifier to classify a set of observations which are known to belong to the same class. This approach is somewhat contrary to the common idea of classifier combination, where the outputs of different classifiers are combined to come to a final decision for a given observation. In our approach, a number of instances is created from the original observation using prior knowledge about the classification task. For example, in handwritten digit recognition, invariance to image shifts and other affine transformations plays an important role. Thus, VTS can be considered a counterpart of the common creation of virtual training data ('perturbation of the training data'). In the experiments, it is used to improve the performance of a Bayesian classifier based on Gaussian mixture densities. We show that using VTS not only yields state-of-the-art results on the well known US Postal Service handwritten digit recognition task (USPS), but that it also has some desirable theoretical properties.

In the next section, we describe the US Postal Service database used in our experiments and present some state-of-the-art results that were reported on this database in the last years. In Section 3, we briefly discuss the idea of classifier combination and one particular classifier combination scheme, namely the very popular sum rule. In Section 4, the VTS method is presented and its theoretical

Table 1. Results reported on USPS.

Author	Method	Error [%]
Simard+ 1993	Human Performance	2.5
Vapnik 1995	Decision Tree C4.5	16.2
Freund & Schapire 1996	AdaBoost & Nearest Neighbour	*6.4
Simard+ 1998	Five-Layer Neural Net	4.2
Schölkopf 1997	Support Vectors	4.0
Schölkopf+ 1998	Invariant Support Vectors	3.0
Drucker+ 1993	Boosted Neural Net	*2.6
Simard+ 1993	Tangent Distance & Nearest Neighbour	*2.5
This work:	Gaussian Mixtures, Invariances	2.2

*: 2418 machine printed digits were added to the training set

properties are discussed. In Section 5, the statistical classifier we used in our experiments (in combination with VTS) is described. In this context, we will also discuss possibilities to incorporate invariances into the classifier which go beyond the use of virtual test samples. This is done by creating virtual training data and by using an invariant distance measure called tangent distance, which was proposed by Simard in 1993 [14]. After presenting some experimental results in Section 6, the paper is ended by drawing some conclusions and giving an outlook to future work in Section 7.

2 The US Postal Service Task & Feature Analysis

The USPS database (`ftp://ftp.kyb.tuebingen.mpg.de/pub/bs/data/`) is a well known handwritten digit recognition task. It contains 7,291 training examples and 2,007 test examples. The digits are isolated and represented as 16×16 pixels sized grayscale images (see Figure 1). Making use of 'appearance based pattern recognition', we interpret each pixel as a feature in our experiments, resulting in 256-dimensional feature vectors. Because of this rather high-dimensional feature space, we optionally apply a linear discriminant analysis (LDA, [5]) for feature reduction. As the maximum number of features that can be extracted by applying the LDA to a K-class problem is $K-1$, we create four pseudoclasses for each USPS digit class by training a mixture with four densities using the algorithms described in Section 5. Thus, the resulting feature vectors are 39-dimensional [2]. One of the advantages of USPS is that many recognition

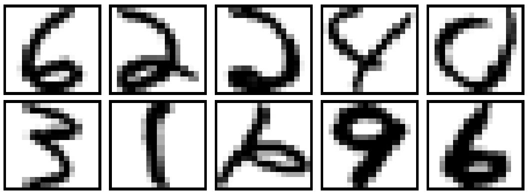

Fig. 1. Example images taken from USPS.

results have been reported by various research groups throughout the last years. Because of that, a meaningful comparison of the different classifiers is possible, with some results given in Tab. 1. Error rates marked with an asterisk were obtained using a modified USPS training set, which – resulting in restricted comparability – was extended by adding 2,418 machine printed digits.

3 The Idea of Classifier Combination

The idea of classifier combination the following: Given a particular pattern recognition problem, the goal is usually to implement a system which achieves the best possible recognition results on unseen data. Thus, in many cases, a variety of pattern recognition approaches is evaluated and the one performing best is chosen to solve the task. Unfortunately – in that approach – all other systems that have been developed are useless. In opposite to this, the idea of classifier combination is to use all classifiers $C_m, m = 1, ..., M$ for classification and to come to a final decision by combining the outputs in a suitable way (cp. Fig. 2). In the last years, many combination approaches have been considered, among them the product rule, the sum rule, or the median rule, where in some cases the 'vote' of a classifier is weighted according to its performance on the training set (i.e. boosting methods). Note that if such combination rules should be meaningful, the outputs of the classifiers must be normalized. Thus, we assume that - given the observation $x \in \mathbb{R}^D$ - each classifier C_m computes posterior probabilities $p_m(k|x)$ for each class $k = 1, ..., K$, which are normalized in the sense that $\sum_{k=1}^{K} p(k|x) = 1$. It should be noted that – for instance – the outputs of an artificial neural net approximate such posterior probabilities [10], assuming that a sufficient amount of training data is available. Thus, normalization comes for free in many applications.

For a single classifier, the Bayesian decision rule can be used for classification:

$$x \mapsto r(x) = \underset{k}{\operatorname{argmax}} \{p(k|x)\} = \underset{k}{\operatorname{argmax}} \left\{ \frac{p(k) \cdot p(x|k)}{\sum_{k'=1}^{K} p(k') \cdot p(x|k')} \right\}, \qquad (1)$$

where $p(k)$ is the prior probability of class k and $p(x|k)$ is the class conditional probability for the observation x given class k. Note that the denominator of Eq. (1) is independent of k and can be neglected for classification purposes. If different classifiers C_m are available (computing posterior probabilities $p_m(k|x)$) the final decision can – for instance – be obtained using the sum rule

$$x \mapsto r(x) = \underset{k}{\operatorname{argmax}} \left\{ \sum_{m=1}^{M} p_m(k|x) \right\}. \qquad (2)$$

Although Eq. (2) is widely accepted to yield state-of-the-art results in many applications, KITTLER assumed in his derivation of the sum rule that the posterior probabilities $p_m(k|x)$ computed by the different classifiers do not differ much from the prior probabilities $p(k)$ [9]. In other words, the derivation of

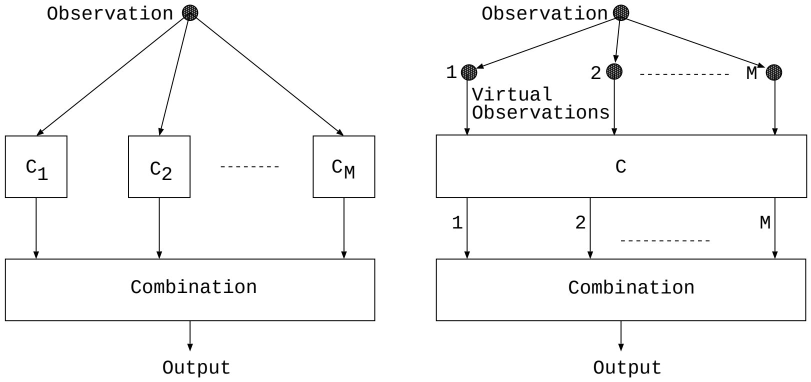

Fig. 2. Classifier combination (left) vs. the virtual test sample method (right).

the sum rule for classifier combination is based on the strong assumption that the features extracted from the data contain no discriminatory information. Interestingly, KITTLER also observed that the good performance of the sum rule could possibly be explained by its error tolerance: Using the sum rule, errors in estimating the real (and therefore usually unknown) posterior probabilities are dampened, while for instance in the case of the product rule, these estimation errors are amplified [9].

If no set of classifiers exists for combination, techniques like 'bagging' [1] or 'boosting' [13] exist, which generate a variety of classifiers using different subsets (bagging) respectively differently weighted versions of the training data (boosting) for training. Here, it is assumed that the classifiers are 'instable', i.e. that modifications of the training set have a significant impact on the resulting classifier. Otherwise, combination of the resulting classifiers would be pointless.

4 The Virtual Test Sample Method

The basic idea of the virtual test sample method is to create a number of 'virtual test samples' starting from the original observation, to classify each of these separately using a single classifier C and to suitably combine these decisions to a final decision for the original observation (cp. Fig. 2). In handwritten digit recognition, invariance to affine transformations is usually desired, but generally speaking all transformations respecting class membership can be considered here. Thus, given the observation x, we can create virtual test samples $x(\alpha) = t(x, \alpha)$, $\alpha \in \mathcal{M}$, with $M = |\mathcal{M}|$, where $t(x, \alpha)$ is a transformation with parameters $\alpha \in \mathrm{I\!R}^L$. In the experiments, ± 1 pixel shifts were applied, i.e. $M = 9$ (eight shifts and the original image). As an image cannot be shifted into different directions at the same time, the resulting 'events' $x(\alpha)$, $\alpha \in \mathcal{M}$ can be regarded as being mutually exclusive and a final decision can be computed as follows:

$$x \longmapsto r(x) \quad = \quad \operatorname*{argmax}_k \{p(k|x)\}$$

$$
\begin{aligned}
&= \operatorname*{argmax}_k \left\{ \sum_{\alpha \in \mathcal{M}} p(k, \alpha|x) \right\} \\
&= \operatorname*{argmax}_k \left\{ \sum_{\alpha \in \mathcal{M}} p(\alpha|x) \cdot p(k|x, \alpha) \right\} \\
&\stackrel{model}{=} \operatorname*{argmax}_k \left\{ \sum_{\alpha \in \mathcal{M}} p(\alpha) \cdot p(k|x(\alpha)) \right\} \qquad (3)
\end{aligned}
$$

Here, the simultaneous occurrence of an observation x and a parameter vector $\alpha \in \mathbb{R}^L$ is modeled by the virtual test sample $x(\alpha)$, i.e. by applying the respective transformation to the observation. Furthermore, α is assumed to be independent of x. Thus, to come to a final decision for the original observation, we only have to add the posterior probabilities $p(k|x(\alpha))$, weighted with the prior probabilities $p(\alpha)$ of the transformation parameters. In the experiments conducted throughout this work - if nothing else is said - these transformation parameters are assumed to be uniformly distributed. Thus, the prior probabilities $p(\alpha)$ can be neglected for classification purposes and Eq. (3) reduces to

$$
x \longmapsto r(x) = \operatorname*{argmax}_k \left\{ \sum_{\alpha \in \mathcal{M}} p(k|x(\alpha)) \right\} \qquad (4)
$$

The only assumption needed here is the mutual exclusiveness of the virtual test samples. As each of these is the result of applying a unique transformation to the given observation, this assumption seems reasonable. This 'virtual test sample method' has a number of desirable properties:

Computational Complexity:
The computational complexity of the VTS recognition step is the same as that of classifier combination. Yet, only one classifier has to be trained in the VTS training phase, which is especially important for statistical classifiers, where the training step is computationally expensive in many cases.

Theoretical Basis:
In contrast to the derivation of the sum rule in the framework of classifier combination, VTS sum rule is straightforward to derive, with the assumption of mutual exclusiveness of the $x(\alpha)$ sounding reasonable.

Increased Transformation Tolerance/ Invariance:
Obviously, invariance properties with respect to the transformations used for virtual test data creation are incorporated into the classifier.

Ease of Implementation & Effectiveness:
Assuming a suitable normalization of the classifier's output, VTS is very simple to embed into an existing classifier. Furthermore, using VTS significantly reduced the error rates in the experiments conducted throughout this work. For real-time applications, VTS is straightforward to parallelize (just like classifier combination), as it is inherently parallel.

Applicable together with Classifier Combination:
In principle, VTS and classifier combination can be used at the same time. Doing so, our best VTS result could in fact be slightly improved (cp. Section 7).

Incorporation of Prior Knowledge about Transformation Probabilities:
Finally, it is possible to incorporate prior knowledge into VTS classification via an appropriate choice of the probabilities $p(\alpha)$ (our model) respectively $p(\alpha|x)$. For instance, in a statistical framework as the one presented in the next section, these probabilities could be learned from the training data.

5 The Statistical Classifier

In this section, we describe the statistical classifier which we used in combination with the VTS method in our experiments. To classify an observation $x \in \mathbb{R}^D$, we use the Bayesian decision rule as given in Eq. (1), which is known to minimize the number of expected classification errors in the case that the true distributions $p(k)$ and $p(x|k)$ are known. Naturally, as these are unknown in most practical applications, we have to choose models for them and learn the respective parameters using the training data. In the experiments, we choose $p(k) = \frac{1}{K},\ k = 1, ..., K$, and model the class conditional probabilities $p(x|k)$ using Gaussian mixture densities (GMD) (see Eq. (6)) respectively Gaussian kernel densities (GKD). In order to keep the number of free model parameters small (and thus allow for reliable parameter estimation), we make use of a globally pooled covariance matrix

$$\Sigma = \sum_{k=1}^{K} \sum_{i=1}^{I_k} \frac{N_{ki}}{N} \cdot \Sigma_{ki}, \qquad (5)$$

where Σ_{ki} is the covariance matrix of component density i of class k and N_{ki} is the number of observations that are assigned to that particular density. Thus, we obtain the following expression for the class conditional probabilities:

$$p(x|k) = \sum_{i=1}^{I_k} c_{ki} \cdot \mathcal{N}(x|\mu_{ki}, \Sigma), \qquad (6)$$

where I_k is the number of component densities used to model class k, c_{ki} are weight coefficients (with $c_{ki} > 0$ and $\sum_{i=1}^{I_k} c_{ki} = 1$, which is necessary to ensure that $p(x|k)$ is a probability density function) and μ_{ki} is the mean vector of component density i of class k. Furthermore, we only use a diagonal covariance matrix, i.e. a variance vector. Note that this does not lead to a loss of information, since a Gaussian mixture of that form can still approximate any density function with arbitrary precision. Maximum likelihood parameter estimation is now done using the Expectation-Maximization algorithm [3]. Concerning Gaussian kernel densities it should be pointed out that these can be regarded an extreme case of a Gaussian mixture, where each reference sample x_n defines a Gaussian normal distribution $\mathcal{N}(x|x_n, \Sigma)$ [8].

Note that the approach presented above is only invariant with respect to transformations that are present in the training data. In the following, we therefore briefly describe two possibilities to enhance the invariance properties of the statistical classifier that go beyond the usage of VTS.

5.1 Creation of Virtual Training Data

A typical drawback of statistical classifiers is their need for a large amount of training data, which is not always available. A common solution for this problem is the creation of virtual training data. Here, just like for the VTS method, ± 1 pixel shifts were chosen, resulting in $9 \cdot 7{,}291{=}65{,}619$ reference samples.

5.2 Incorporating Invariant Distance Measures

Another way to incorporate invariances is to use invariant probability density functions or - equivalently - invariant distance measures [7]. Here, we choose tangent distance (TD), which proved to be especially effective for optical character recognition. In [14], the authors observed that reasonably small transformations of certain objects (like digits) do not affect class membership. Simple distance measures like Euclidean or Mahalanobis distance do not account for this and are very sensitive to transformations like translations or rotations. When an image x of size $I \times J$ is transformed (e.g. scaled and rotated) with a transformation $t(x, \alpha)$ which depends on L parameters $\alpha \in \mathbb{R}^L$ (e.g. the scaling factor and the rotation angle), the set of all transformed images $M_x = \{t(x, \alpha) : \alpha \in \mathbb{R}^L\} \subset \mathbb{R}^{I \times J}$ is a manifold of at most L dimensions. The distance between two images can now be defined as the minimum distance between their according manifolds, being truly invariant with respect to the L transformations regarded. Unfortunately, computation of this distance is a hard optimization problem and the manifolds needed have no analytic expression in general. Therefore, small transformations of an image x are approximated by a tangent subspace $\hat{M}_x$ to the manifold M_x at the point x. Those transformations can be obtained by adding to x a linear combination of the vectors $x_l, l = 1, ..., L$ that span the tangent subspace. Thus, we obtain as a first-order approximation of M_x:

$$\hat{M}_x = \left\{ x + \sum_{l=1}^{L} \alpha_l \cdot x_l \ \ : \alpha \in \mathbb{R}^L \right\} \subset \mathbb{R}^{I \times J} \tag{7}$$

Now, the single sided TD $D_T(x, \mu)$ between two images x and μ is defined as

$$D_T(x, \mu) = \min_{\alpha} \left\{ ||x + \sum_{l=1}^{L} \alpha_l \cdot x_l - \mu||^2 \right\} \tag{8}$$

The tangent vectors x_l can be computed using finite differences between the original image x and a small transformation of x [14]. Furthermore, a double sided TD can also be defined by approximating M_x and M_μ. In the experiments, we computed seven tangent vectors for translations (2), rotation, scaling, axis

$$\frac{1}{16}\boxed{\begin{array}{ccc}1&2&1\\2&4&2\\1&2&1\end{array}}$$

Fig. 3. Prior probabilities $p(\alpha)$ chosen for image shifts in the experiments. For example, the prior probability for the original (i.e. unchanged) image is $(4/16) = 0.25$.

deformations (2) and line thickness as proposed by Simard [14]. Given that the tangent vectors are orthogonal, Eq. (8) can be solved efficiently by computing

$$D_T(x,\mu) = ||x-\mu||^2 - \sum_{l=1}^{L} \frac{[(x-\mu)^t \cdot x_l]^2}{||x_l||^2} \quad (9)$$

The combination of TD with virtual data creation makes sense, as TD is only approximately invariant with respect to the transformations considered. Thus, creating virtual training data yields a better approximation of the original manifold, as the virtual training images lie exactly on it.

For the calculation of the Mahalanobis distance, the observation x and the references μ_{ki} are replaced by the optimal tangent approximations $x(\alpha_{opt})$ respectively $\mu_{ki}(\alpha_{opt})$ in the TD experiments. When calculating single sided TD, the tangents are applied on the side of the references. Note that TD can also be used to compute a 'tangent covariance matrix', which is defined as the empirical covariance matrix of all possible tangent approximations of the references [2]. Further information on a probabilistic interpretation of TD is given in [7].

6 Results

The experiments were started by applying the statistical approach described in Section 5 to the high-dimensional USPS data, using different combinations of virtual training and test data. In Tab. 2, the notation '*a*-*b*' indicates that we increased the number of training samples by a factor of a and that of the test samples by a factor of b. Thus, b=9 indicates the use of VTS. As can be seen, VTS significantly reduces the error rate on USPS from 8.0% to 6.6% (without virtual training data) respectively from 6.4% to 6.0% (with virtual training data). These error rates can be further reduced by applying an LDA. Thus, the best error rate decreases from 6.0% to 3.4%, which is mainly because parameter estimation is more reliable in this rather low-dimensional feature space. Note that in this case, applying VTS reduced the 9-1 error rate from 4.5% to 3.4%, being a relative improvement of 24.4%. This error rate could be slightly improved to 3.3% by assuming a Gaussian distribution for the prior probabilities $p(\alpha)$, resulting in the template depicted in Figure 3. As a key experiment, we boosted the statistical classifier based on 39 LDA features using AdaBoost.M1 [6] for $M = 10$. Indeed, we were able to reduce the 9-1 error rate from 4.5% to 4.2%, yet VTS - by reducing the error rate from 4.5% to 3.4% - significantly outperformed AdaBoost on this particular dataset.

In another experiment, we investigated on the use of TD in combination with VTS. These experiments were performed in the high-dimensional feature space

Table 2. USPS results for Mahalanobis/ tangent distance, with/ without LDA.

Method:	Error rate [%]			
	1-1	1-9	9-1	9-9
GMD,	8.0	6.6	6.4	6.0
GMD, LDA,	6.7	5.9	4.5	3.4
GMD, tangent distance	3.9	3.6	3.4	2.9
GKD, tangent distance	3.0	2.6	2.5	2.4

(no LDA), as TD in its basic form is defined on images (although it can also be defined on arbitrary feature spaces, where the tangent vectors are learned from the data itself [7]). Using single sided TD, the best error rate could be reduced from 3.4% for the LDA based statistical classifier to 2.9% (using single sided TD and about 1,000 normal distributions per class). This error rate could be further reduced to 2.7% by using double sided TD. Replacing the mixture density approach by kernel densities, the VTS error rate was reduced to 2.4%. In these experiments, standard deviations were used instead of diagonal covariance matrices. Finally, by combining the outputs of five VTS based kernel density classifiers (using different norms in the distance calculations and different kinds of training data multiplication), the error rate could be further reduced to 2.2%. To make sure that these good results are not the result of overfitting, we also applied our best kernel density based USPS classifier (error rate 2.4%) to the well known MNIST task without further parameter tuning, obtaining a state-of-the-art result of 1.0% (1.3% without VTS). Although this result is not the best known on MNIST (the best error rate of 0.7% was reported by Drucker in [4]), it shows that the algorithms presented here generalize well.

7 Conclusions & Outlook

In this paper, we presented a combined classification approach called the 'virtual test sample method', which is based on using a single classifier in combination with artificially created test samples. Thus, it can be regarded as a counterpart to the creation of virtual training data, which is a common approach in pattern recognition. We showed that the proposed method is straightforward to justify and has some desirable properties, among them the possible incorporation of prior knowledge and the fact that only a single classifier has to be trained. In our experiments, the approach was used to improve the performance of a statistical classifier based on the use of Gaussian mixture densities in the context of the Bayesian decision rule. The results obtained on the well known US Postal Service task are state-of-the-art, especially when the virtual test sample method is combined with the incorporation of invariances into the classifier, which was done by using Simard's tangent distance and resulted in an error rate of 2.2%. Finally, the approach seems to generalize well, as a state-of-the-art error rate of 1.0% was also obtained on the MNIST handwritten digit task. Besides developing better models for the probabilities $p(\alpha)$ respectively $p(\alpha|x)$ considered in the virtual test sample method, future work will also include investigating its effectiveness in other pattern recognition domains.

Acknowledgement
The authors wish to thank the student researchers Mark Oliver Güld and Ralf Perrey for implementing parts of the methods and conducting some of the experiments described in this paper.

References

1. Breiman, L.: Bagging Predictors. Technical Report 421, Department of Statistics, University of California at Berkeley, 1994.
2. Dahmen, J., Keysers, D., Ney, H., Güld, M.: Statistical Image Object Recognition using Mixture Densities. Journal of Mathematical Imaging and Vision, Vol. 14, No. 3, Kluwer Academic Publishers, pp. 285-296, May 2001.
3. Dempster A., Laird, N., Rubin, D.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, 39(B), pp. 1–38, 1977.
4. Drucker, H., Schapire, R., Simard, P.: Boosting Performance in Neural Networks. Int. Journal of Pattern Recognition and Artificial Intelligence, Vol. 7, No. 4, pp. 705–719, 1993.
5. Duda, R., Hart, P.: Pattern Classification and Scene Analysis. John Wiley & Sons, 1973.
6. Freund, Y., Schapire, R.: Experiments with a New Boosting Algorithm. 13th Int. Conference on Machine Learning, Bari, Italy, pp. 148-156, July 1996.
7. Keysers, D., Dahmen, J., Ney, H.: A Probabilistic View on Tangent Distance. 22nd Symposium German Association for Pattern Recognition (DAGM), Kiel, Germany, pp. 107-114, 2000.
8. Keysers, D., Dahmen, J., Theiner, T., Ney, H.: Experiments with an Extended Tangent Distance. 15th Int. Conference on Pattern Recognition, Barcelona, Spain, Vol. 2, pp. 38-42, September 2000.
9. Kittler, J.: On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 3, pp. 226–239, March 1998.
10. Ney, H.: On the Probabilistic Interpretation of Neural Network Classifiers and Discriminative Training Criteria, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 2, pp. 107–119, February 1995.
11. Schölkopf, B.: Support Vector Learning. Oldenbourg Verlag, Munich, 1997.
12. Schölkopf, B., Simard, P., Smola, A., Vapnik, V.: Prior Knowledge in Support Vector Kernels. Advances in Neural Information Processing Systems 10, MIT Press, Cambridge, MA, pp. 640–646, 1998.
13. Schapire, R., Freund, Y., Bartlett, P., Lee, W.: Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. The Annals of Statistics, Vol. 26, No. 5, pp. 1651–1686, 1998.
14. Simard, P., Le Cun, Y., Denker, J.: Efficient Pattern Recognition using a New Transformation Distance. Advances in Neural Information Processing Systems 5, Morgan Kaufmann, San Mateo CA, pp. 50–58, 1993.
15. Simard, P., Le Cun, Y., Denker, J., Victorri, B.: Transformation Invariance in Pattern Recognition – Tangent Distance and Tangent Propagation. Lecture Notes in Computer Science, Vol. 1524, Springer, pp. 239–274, 1998.
16. Vapnik, V.: The Nature of Statistical Learning Theory, Springer, New York, pp. 142-143, 1995.