An Efficient Three-Stage Classifier for Handwritten Digit Recognition

Dejan Gorgevik
University "Sv. Kiril i Metodij", Faculty of Electrical Eng., Department of Computer and Information Technology, Karpos II bb, POBox 574, 1000 Skopje, Macedonia, dejan@etf.ukim.edu.mk

Dusan Cakmakov
University "Sv. Kiril i Metodij", Faculty of Mechanical Eng., Department of Mathematics and Computer Science, Karpos II bb, POBox 464, 1000 Skopje, Macedonia, dusan@mf.ukim.edu.mk

Abstract

This paper proposes an efficient three-stage classifier for handwritten digit recognition based on NN (Neural Network) and SVM (Support Vector Machine) classifiers. The classification is performed by 2 NNs and one SVM. The first NN is designed to provide a low misclassification rate using a strong rejection criterion. It is applied on a small set of easy to extract features. Rejected patterns are forwarded to the second NN that uses additional, more complex features, and utilizes a well-balanced rejection criterion. Finally, rejected patterns from the second NN are forwarded to an optimized SVM that considers only the "top k" classes as ranked by the NN. This way a very fast SVM classification is obtained without sacrificing the classifier accuracy. The obtained recognition rate is among the best on the MNIST database and the classification time is much better compared to the single SVM applied on the same feature set.

1. Introduction

The accuracy of an overall recognition system mainly depends on the discriminative capability of the extracted features and the generalization performance of the designed classifier.

Over the last two decades, NNs have been widely used to solve complex classification problems [1]. They are very fast classifiers with relatively fast training time. However, a single NN often exhibits the overfitting behavior which results in a weak generalization performance when trained on a limited set of training data. On the other hand, there is a consensus in machine learning community that SVMs are most promising classifiers due to their excellent generalization performance [2]. Curse of dimensionality and overfitting in NNs, seldom occur in SVMs. However, SVMs for multi-classes classification problems are relatively slow and their training on a large data set is still a bottle-neck.

In this paper, we present an efficient system for handwritten digit recognition attempting to utilize the advantages of both, NN and SVM classifiers. Our system is three-staged where firstly two NNs with rejection criteria are used and then, a reduced one-against-the-rest SVM classification is exploited for the rejected patterns.

In the first stage, our goal was to perform fast classification and to achieve low misclassification rate using a small set of 40 features and NN with strong rejection criterion.

In the second stage, rejected patterns from the first stage are represented by additional 252 features and forwarded to the second NN with a well-balanced rejection criterion.

In the third stage, the rejected patterns from the NN of the second stage together with the class rankings are forwarded to a SVM. Thus, we have obtained a SVM with reduced complexity aimed to classify small number of "hard to classify" patterns considering only the k top-ranked classes of the rejected patterns.

Carefully optimized classifiers and choice of the features have led to fast and accurate recognition system. Among the many handwritten digit recognition systems with high recognition rates e.g. [3], [4], [5], [6], the recognition rate of our system on MNIST database is among the best.

2. The System Architecture

Pre-processed isolated digit images are input for the feature extraction modules, that transfer the extracted features toward NN and SVM classifiers (see Figure 1).

Every image is centered in a square bounding box, and then slant correction is performed. The slant angle is estimated as the inclination of the line connecting the gravity centers of the top 25% part and the bottom 25% part of the image. Then a sub-pixel precision shear transformation is performed in order to remove the estimated inclination. This approach provides more reliable slant correction than the "standard" approaches e.g. [7].

The first set of features contains 40 easy to extract projection-based features that are input for the first NN classifier (40–30–10). From the patterns that are rejected from this network, additional 252 more robust features are extracted. All 292 features are used in the second and the

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

third classification stage where "hard to classify" patterns are firstly forwarded to the second NN (292-30-10). The patterns rejected from the second NN are finally classified by the SVM.

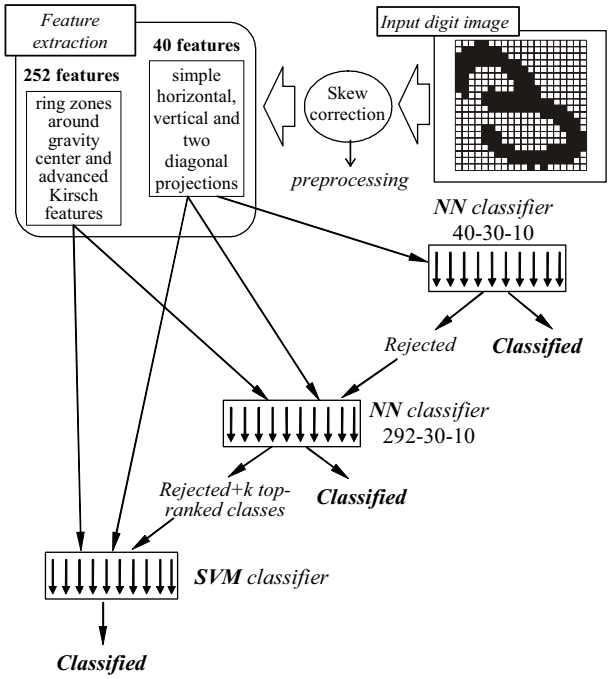

Figure 1. The system architecture

This way, we have obtained a low time consuming classifier with slightly better recognition rate compared to the single SVM applied on the same feature set.

3. Feature extraction

Features that offer better discriminative power are usually more complicated and harder to extract regarding processing time that can not be neglected when building a fast recognition system.

The first 40 features used in the first classification stage are simple horizontal, vertical and diagonal projections. Since the character images are not of the same size, the projection vectors are linearly rescaled in order to obtain 10 features from the horizontal projections, 8 features from the vertical projections, and 11 features from each of the two diagonal projections (see Figure 2).

The second 44 features are used in the second and third stages of classification. They are extracted as pixel counts in rings zones around the gravity center of the image (see Figure 2). We use three rings each divided in different number of equal zones. The outermost ring has a radius r equal to the distance from the gravity center to the furthest black pixel of the image. The first ring with radius $0.2 \cdot r$ provides 4 features and the second ring with radius $0.5 \cdot r$ provides 24 features. The last 16 features of this group are provided from the outermost ring.

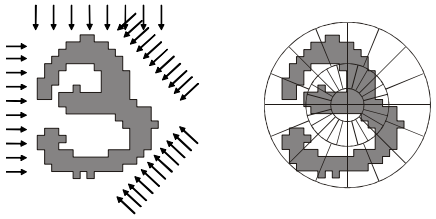

Figure 2. Projection and ring-zone features

The last 208 features are also used in the second and third stages of classification. These features use Kirsch operator to detect local directional information of the edges of the input pattern [8]. Compared with chain code which also describes the edge direction, Kirsch edge detection is more robust even under noisy conditions.

The image is scanned from left to right, top to bottom, right to left and bottom to top, respectively. The first black pixel which the scanning line intersects forms the first outermost periphery. The second black pixel which is the starting point of the second run forms the second outermost periphery (see Figure 3). When the image is scanned in horizontal direction, the vertical and both diagonal Kirsch features are extracted at the outermost periphery. When the image is scanned in vertical direction, the horizontal and both diagonal Kirsch features are extracted at the outermost and second outermost periphery. This way, 3 Kirsch directional features plus one feature representing the position of the periphery measured as the distance from the edge of the bounding box of the image are provided for each periphery pixel. The feature vectors are again linearly rescaled to 40 features coming from the left and right periphery each, 32 features coming from the first outermost top and bottom periphery each, and 32 features coming from the second outermost top and bottom peripheries.

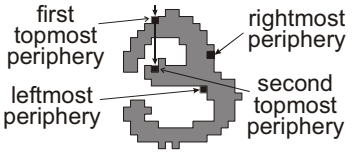

Figure 3. Kirsch features

The image processing is performed on the grayscale digit images using sub-pixel precision. All parameters including the number of features by projection, the radiuses of rings for zone-pattern regions and the number of features coming from the outermost peripheries for Kirsch features are carefully chosen after several iterations using observations about their discriminative power. The features are pre-processed for zero mean and unit variance.

4. Classification

Classification is performed in three stages where the first and second stage classifiers send rejected patterns to the next classifier.

In the first stage, we have used a multilayer perceptron NN with one hidden layer and architecture 40-30-10. In this stage, our goal was to perform fast classification of "easy to classify" patterns keeping low misclassification rate using a strong rejection criterion. So, the input feature set was small (40 features) containing features extracted from pattern projections. The rejection criterion is based on the "top 2" NN outputs. Each sample for which the highest NN output O_1 is smaller than a certain threshold T_1 ($O_1 < T_1$) or for which the difference between the "top 2" classifier outputs is smaller than a certain threshold T_2 ($O_1 - O_2 < T_2$) are rejected. Varying these thresholds to obtain low misclassification rate we have found suitable values $T_1 = 0.994$ and $T_2 = 5.5$ that are used to obtain presented recognition results.

In the second stage, we have used a multilayer perceptron NN with one hidden layer and architecture 292-30-10. This NN uses 252 additional features extracted from the digit image. The rejection criterion was the same as in stage two with parameters $T_1 = 0.985$ and $T_2 = 4.0$.

In the third stage, we have used a SVM with Gaussian kernel $K(x_i, x_j)=exp(-\|x_i-x_j\|^2/\sigma^2)$. In this stage, for the remained "hard to classify" patterns, the complete feature set of 292 features is used again. To reduce SVM complexity and speed up the classification process, the SVM examines only the k top-ranked classes obtained by the second NN.

Since SVMs are binary classifiers, the one-against-the-rest method is used to construct ten-class classifier. That is, each classifier is constructed by separating one class from the rest. The usual classification approach is to present the pattern sequentially to all ten one-against-the-rest SVMs and then to make the decision by choosing the class with the largest classifier output value. However, if strong evidence is provided in advance that certain pattern is not member of some classes, one can decide not to present the pattern to the SVMs for the corresponding classes. Since the SVMs are considered in a sequentially manner, discarding some classes from the consideration saves significant amount of time/processing.

5. The Recognition Result

Our experiments were performed on the well-known MNIST database (http://yann.lecun.com/exdb/mnist) of handwritten digits. MNIST database consists of 60000 training samples and 10000 test samples. All digits have been size-normalized and centered in a 28×28 box.

The NN and SVM classifiers were implemented using the TORCH [9] library while the feature extraction was performed using proprietary C++ code. The C++ programs were compiled by Microsoft Visual C++.NET 7.1. All tests were performed on 2.4GHz P4 processor under Windows XP.

The NNs were trained on the whole training set using stochastic gradient descent. The SVM for every class was carefully optimized for the parameters σ and C using an automated parameter search procedure. For the parameter optimization 40000 samples were used for training and 20000 for validation. After finding the optimal σ and C, the SVM was finally trained on the whole set of 60000 samples.

In Table 1, the individual error rates and the recognition times of the two NNs and the SVM are presented. It is obvious that the SVM has a superior recognition rate but it is a level of magnitude slower then the NNs. The CPU times are given for the recognition of the whole MNIST test set containing 10000 samples (excluding preprocessing and feature extraction).

Table 1. Individual classifier performance

Classifier	Error rate (%)	CPU (s)
NN 40-30-10	3.31	0.129
NN 292-30-10	1.31	0.330
SVM	0.85	196.031

In Table 2, the number of recognized, misclassified and rejected patterns in every stage of the proposed recognition system together with the rejection thresholds and the CPU times are given. To obtain the complete recognition time, the time spent for pre-processing (PP), extraction of the first 40 features (FE40) and extraction of the additional 252 features (FE252) are also given.

Table 2. Characteristics and efficiency of the proposed 3-stage NN-NN-SVM classifier

Action	#Rec.	#Mis.	#Rej.	T_1, T_2	CPU (s)
PP + FE40	on 10000 patterns			–	1.252
1st stage NN	6551	6	3443	0.994, 5.5	0.129
FE252	on 3443 patterns			–	2.492
2nd stage NN	3053	23	367	0.985, 4.0	0.114
SVM top 4	311	54	–	–	3.031
Total	9917	83	–	–	7.018

The lowest error rate of 0.83% (83/10000) was obtained when 4 top-ranked classes from the second NN were considered by the SVM. Of course, faster recognition times (5.656 and 6.354 seconds) could be achieved by considering only 2 or 3 top-ranked classes by the SVM, but the obtained error rates (0.96% and 0.84%) were higher. Increasing the number of top-ranked classes that are considered by the SVM above $k = 4$ has not decreased the error rate. This means that the NN from the second stage manages to keep the right class among the 4

top-ranked classes for the rejected patterns that are going to be recognized with high accuracy by the SVM.

The lowest obtained error rate of 0.83% is slightly better then the error rate of the best individual classifier (SVM) given in Table 1. That is because some of the patterns that would be misclassified by the SVM are correctly recognized by some of the NN classifiers.

In Figure 4, a comparison of the performances of different algorithms tested on the MNIST database is given. The proposed method (3-Stage NN-NN-SVM) was used on the original MNIST database and provided error rate of 0.83% that is better than error rates of most of the classifiers.

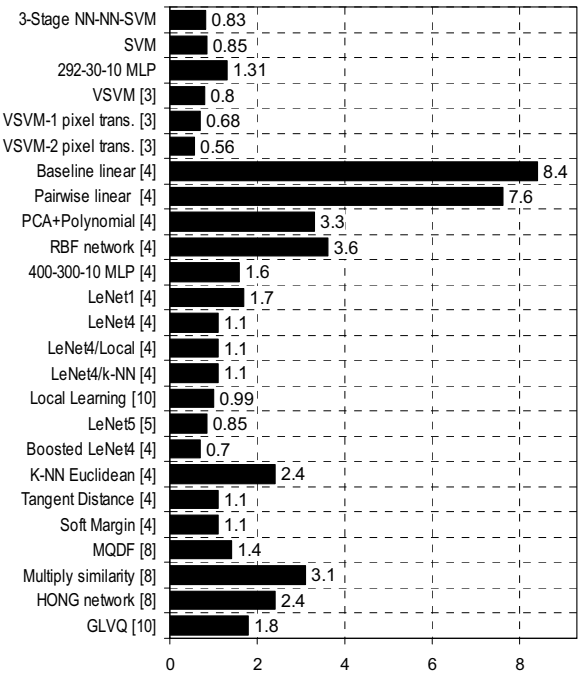

Figure 4. Error rates of different methods on the test set of MNIST database

The best results, provided by Boosted LeNet4 [4] and Virtual SVMs [3] are the state-of-art results, but corresponding classifiers were trained on a perturbed MNIST database, where the training set was augmented with artificially altered versions of the original training samples.

6. Conclusion

This paper proposes an efficient three-stage classification of handwritten digits based on two NNs and one SVM.

In the first stage, a NN is designed to provide a low misclassification rate using strong rejection criterion. In order to be fast it is applied on a small set of easy to extract features.

In the second stage, additional more complex features are extracted from the rejected patterns and forwarded to the second NN. A well-balanced rejection criterion is also applied in order to provide a low misclassification rate.

Rejected patterns from the second stage, together with the class ranking obtained by the NN are forwarded to a SVM. Thus, we have obtained a SVM with reduced complexity because instead of considering all possible classes for every pattern it considers only the k top-ranked classes (best recognition rate was for $k = 4$).

To achieve so high recognition rates, the feature set is carefully chosen as a combination of simple projections, zone-pattern regions around gravity center and powerful Kirsch features extracted from the image peripheries.

Obtained classification time and recognition rates are among the best on MNIST database. They are also better than the recognition time and the recognition rate of single SVM applied on the same feature set.

References

[1] Bishop C.M., *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995, pp. 364-369.

[2] Burges C., "A Tutorial on Support Vector Machines for Pattern Recognition", *Knowledge Discovery and Data Mining*, Vol. 2, 1998, pp. 1-47.

[3] DeCoste D. and Scholkopf B., "Training invariant support vector machines", *Machine Learning*, Vol. 46, No. 1-3, pp. 161-190, 2002.

[4] LeCun Y., Jackel L. D., Bottou L., Brunot A., Cortes C., Denker J. S., Drucker H., Guyon I., Muller U. A., Sackinger E., Simard P., and Vapnik V., "Comparison of learning algorithms for handwritten digit recognition", In F. Fogelman and P. Gallinari, editors, *International Conference on Artificial Neural Networks*, Paris, 1995, pp. 53-60.

[5] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, 86 (11), 1998, pp. 2278–2324.

[6] Dong J.X., "Comparison of algorithms for handwritten numeral recognition", Technical report, CENPARMI, Concordia University, Nov. 1999

[7] Grother P. J., "Karhunen Loève Feature Extraction for Neural Handwritten Character Recognition", *Proceedings of Applications of Artificial Neural Networks III*, SPIE, Orlando, Florida, 1992, pp. 155-166.

[8] Lee S.W., "Multilayer Cluster Neural Networks for totally unconstrained handwritten Numeral Recognition", *Neural Networks*, Vol. 8, No. 5, 1995, pp. 783-792.

[9] Collobert R., Bengio S., and Mariéthoz J., "Torch: a modular machine learning software library", Technical Report IDIAP-RR 02-46, IDIAP, 2002. (http://www.torch.ch)

[10] Dong J.X, Krzyzak A., Suen C.Y., "Local learning framework for handwritten character recognition", *Engineering Applications of Artificial Intelligence*, Vol. 15, 2002, pp. 151–159