In Proceedings of International Conference on Neural Networks (ICNN'97) -- Houston, TX, vol. 3, pp. 1391-1396, 1997.

Recognition of Handwritten Digits using Structural Information

Sven Behnke
Martin-Luther University, Halle-Wittenberg[1]
Institute of Computer Science
06099 Halle, Germany
{behnke|rojas}@informatik.uni-halle.de

Marcus Pfister
SIEMENS AG
AUT 65, Postfach 4848
90327 Nürnberg, Germany
pfister@scn.de

Raul Rojas[1]

Abstract

This article presents an off-line method for recognizing handwritten digits. Structural information and quantitative features are extracted from images of isolated numerals to be classified by a hybrid multi-stage recognition system.

Feature extraction starts with the raw pixel-image and derives more structured representations like line-drawings and attributed structural graphs. Classification is done in two steps: a) the structural graph is matched to prototypes, b) for each prototype there is a neural classifier which has been trained to distinguish digits represented by the same graph-structure.

The performance of the described system is evaluated on two large databases (provided by SIEMENS AG and NIST) and is compared to other systems. Finally, the combination of the described system and a TDNN classifier is discussed. The experimental results indicate that there is an advantage in using structural information to enhance an unstructured neural classifier.

1. Introduction

Automatic handwriting recognition has a variety of applications at the interface between man and machine. Off-line systems are used to process documents like cheques and for automatic mail sorting.

The performance of a method for handwriting recognition can be evaluated by several of criteria, including size of the alphabet, independence of the writing style, reliability and speed of recognition. The application to ZIP-code recognition restricts the alphabet to digits and allows to emphasize the other criteria.

Recognition of handwritten digits is difficult because of the high variability of the scanned image. This is caused by the peculiar writing style of different persons, the context of the digit, different writing devices and media. This leads to scanned digits of different size and slant, and strokes that vary in width and shape. The structure of the digits is further complicated by superfluous strokes and parts of neighboring digits, caused by segmentation.

The problem of handwriting recognition has been studied for decades and many methods have been developed. Some use only the pixel-image as input to a powerful statistical or neural classifier [8]. Others preprocess the data in order to extract some features that are fed into a classifier. Structural methods of pattern recognition rely on structural information in order to produce a classification decision [2].

The method presented here is a hybrid one, similar to [9]. Structural information, as well as quantitative features, are extracted and are used for classification. The goal is to preserve the information essential for recognition and to discard the unnecessary details. The two-stage decision process matches first the structure of the digit to prototypes that have been extracted from the training set. This selects a neural classifier that has been trained to recognize digits having the same structure by using quantitative features.

Figure 1 shows the stages of the recognition process. Section 2 describes the transformation of the pixel-image into a line-drawing. In section 3 the structural analysis of the line-drawing is described. Section 4 explains how the classification decision is made and experimental results are presented in section 5.

2. Vectorization

The grey-level pixel-images of digits which have been scanned and cut constitute the input to the vectorization routine. They are transformed into a line-drawing. This approach is motivated by the fact that the digits have originally been produced by a set of strokes. Digit representation using connected nodes therefore seems to be more natural than using the raw pixel image. Vectorization is done in two steps: a) the pixel image is preprocessed and a skeletonization operator is applied, b) nodes are positioned and connected.

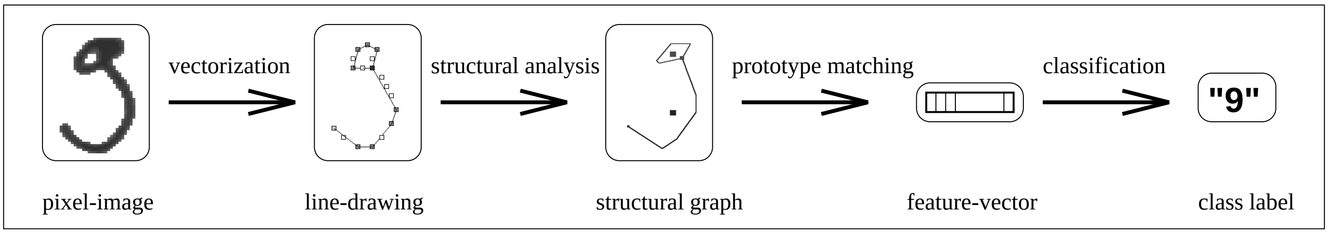

Figure 1. Stages of Recognition.

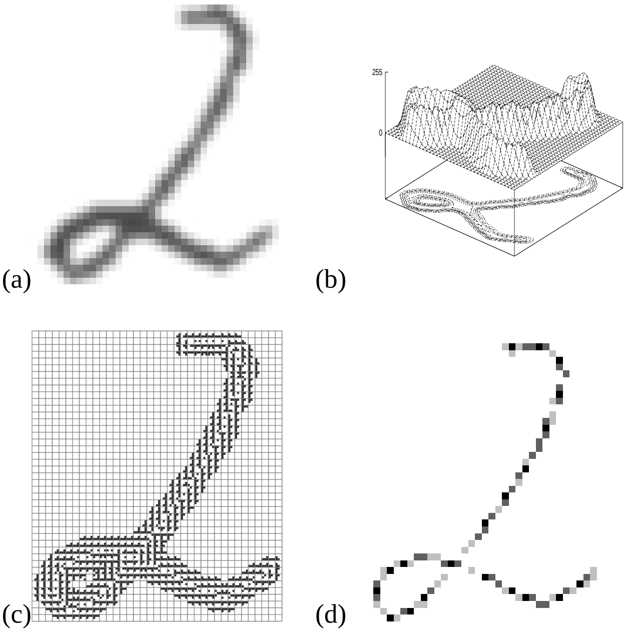

Figure 2. Skeletonization: (a) smoothed pixel-image (b) 3D-visualization, (c) 3×3 regions: neighboring pixels having equal or higher grey-level are marked, (d) Skeleton: peeks are black, ridges are gray.

2.1. Preprocessing and skeletonization

The images are not binarized, but a binarization threshold is used to remove the background, preserving the grey-levels in the foreground (see figure 3(a)). The size of the image is scaled by factors of two to approximately 40×50 pixels. Then the line-width (pen-thickness) is estimated and the image is scaled down, if the line-width is too large. A low-pass filter is applied so that each line cross-section has one maximum only (see figure 2(a,b)).

Skeletonization is used to reduce the line width to approximately one pixel. Unlike morphological methods which start from the border of the line iteratively erode it, the operator used here directly finds a skeleton in the middle of a line. The operator observes 3×3 pixel regions to decide if the central pixel belongs to the skeleton. Pixel having grey-level zero (white) do not belong to the skeleton, but to the background. For all other pixels, the number $c(x, y)$ of neighboring pixels (8-neighborhood) having an equal or higher grey-level is computed (see figure 2(c)) and if this number is less than three the central pixel is added to the skeleton. The resulting skeletons can be seen in figures 2(d) and 3(b). They are similar to the ones produced using a gradient method described in [5] that has a higher computational complexity.

2.2. Placement and connection of nodes

Nodes are placed starting at peaks of the skeleton ($c(x, y) = 0$). Then nodes are placed at pixels belonging to ridges ($c(x, y) = 1$ and $c(x, y) = 2$), but with a minimum distance of two between them.

The nodes now need to be connected to represent strokes. First, the connection structure of the skeleton is reconstructed by inserting connections where 3×3 regions of nodes overlap or touch on the skeleton. In order to insert the few remaining connections necessary to recover the original strokes, more global information is needed. Connections are inserted according to the principles of Gestalt psychology. The goal is to get lines exhibiting good continuity, closure and simplicity.

To achieve this goal candidates for new connections are determined and are evaluated using a measure based on the distance of the nodes to be connected, angles between connections, the gray-level of the pixels between the nodes, and topological information of the connection graph. New connections are inserted, ordered according to this measure, until a topology dependent threshold is reached. After all connections have been inserted, the line drawing is simplified. The lines are smoothed and nodes are taken out at locations of low curvature. Short lines ending in junctions are eliminated and junctions that are close together and connected are merged to form a crossing. The result is shown in figure 3(c).

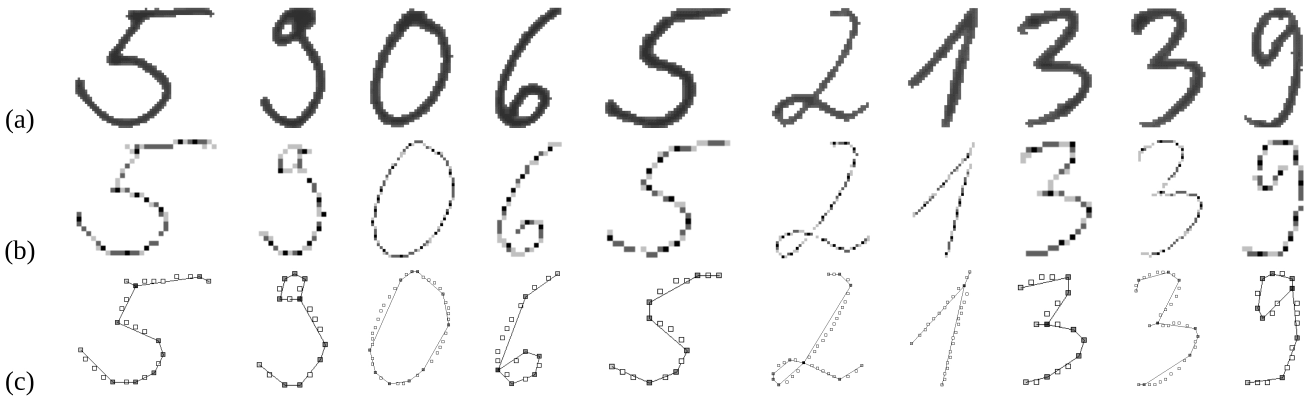

Figure 3. Some digits: (a) background removed, (b) skeleton, (c) line-drawings.

3. Structural analysis

The connected line representation of each digit produced by the previous processing steps is still rather unstructured. A digit is so far represented by some nodes and their connections. The next step derives a more abstract representation consisting of strokes which are joined to form larger curves.

In order to reduce the variability of the input, the direction of the principal axis of the digit is calculated. A shear is then used to correct the slant of the digit so that the principal axis lies in the vertical direction. The image is scaled and centered in a box to normalize its size.

3.1. Search of strokes and connection to curves

A stroke is formed by several lines connected by joints (nodes of degree two), which have a common rotation direction and do not form sharp angles. A stroke has an initial and an end node, such that from the perspective of the initial node, the lines rotate to the right only. Straight strokes run from down to top.

Starting from the nodes having a degree other than two, a topological structure is built by following the connecting lines. The length of the segments and the rotation angle are accumulated for each stroke.

The strokes found touch each other only at the initial or end nodes. The contact points may represent junctions, crossings or changes of rotation direction. A set of strokes can be merged to curves in such a way as to reconstruct the way the digit was drawn.

Two strokes are connected and reduced to a curve only if the rotation direction is preserved and the second constitutes a good continuation of the first. In this step we try to find long curves and the formation of loops is forced. This is done by testing for each common node of two strokes if the two strokes can be merged into a single curve. If this is the case, the candidate is evaluated using the local rotation angle and the total length of the curve. The mergers are performed starting with the best candidates.

Sometimes, defects of the digitalization, noise or small loops and embellishments of the handwritten digits produce short curves which must be eliminated before proceeding to recognize the digit. Using some topological information and the length of the curves it is decided whether it is beneficial to eliminate them from the graph or not. This step simplifies the structural description. Figure 4 shows some simplified curve representations. In this drawing large squares are located at the center of gravity of the curves. Curves run from the middle-sized squares to the small squares.

3.2. Computation of quantitative information

The set of curves found in the previous steps is summarized now using a bipartite graph. Each curve is represented by a node in the left layer of the graph. Nodes in the right layer represent characteristic points such as curve ends, junctions, crossings and saddle points. The edges of the bipartite graph are derived from the curve representation. Each curve is connected to its characteristic points in the same order in which they appear when following

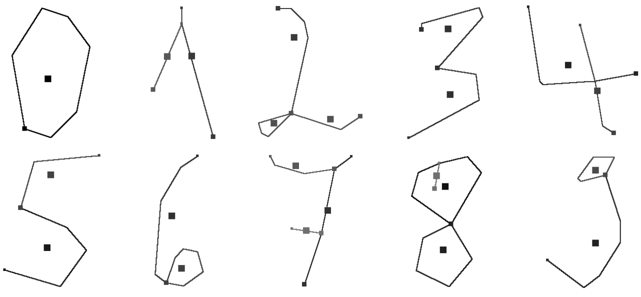

Figure 4. Curve representation of some digits.

the curve. Each node contains attributes, which summarize quantitative information about the curves and points. The curve nodes store the xy-coordinates of the center of gravity of the curve, the accumulated rotation angle, the length, and the distance of end and initial point relative to the total length of the curve. The shape of the curve is summarized by the xy-coordinates of six points distributed uniformly on the curve. The point nodes are described by their xy-coordinates.

4. Classification

Such intensive preprocessing has produced an attributed structural graph that describes the essential features of the digit to be recognized. Recognition is done in two steps: a) the structural graph is matched to prototypes that have been extracted from the training set, b) for each prototype there is a neural classifier which is used to distinguish digits having the same structure based on the extracted quantitative features.

The given digits are split into three sets: a) a training set for the generation of prototypes and training of the classifiers, b) a test set to terminate the training, c) a validation set to evaluate classification performance.

4.1. Graph matching

Two structural graphs are called *isomorph*, if there exists a bijective mapping from the nodes and edges of one graph to the ones of the other graph. Curve nodes can only be assigned to curve nodes and the order of the edges must be preserved. Curves that are almost straight can also be mapped in inverted direction, that means that the order of points belonging to the curve is reversed.

The training set is partitioned into maximal sets that have isomorphic structural graphs. Each partition corresponds to a structural graph that is used as a prototype for the matching step, if the partition contains a significant number of examples. In that way typical structures are extracted, which represent not only perfect digits, but frequent structural deviations as well.

To test for isomorphism first a necessary condition is checked quickly. The number of curve nodes and the number of point nodes of each degree must match. If this holds, the curves of the first graph are permutated and inverted, and the resulting descriptions are matched with the description of the second graph. Sometimes more than one match between the graphs is possible. In this case all matches are used when partitioning the training set, but only the first match is used for recall. If the structural graph does not match any prototype, it is simplified by taking out the shortest curve and is matched again. If there is still no match, the digit is rejected.

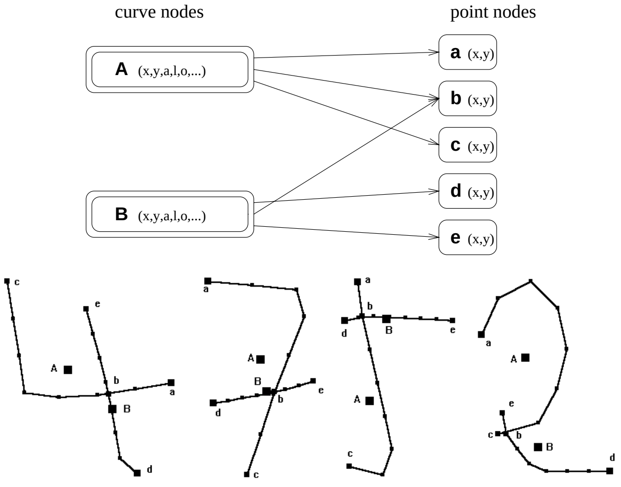

Figure 5. Structural graph and some assigned digits. The first two are typical, the others aren't.

4.2. Neural classifiers

In some cases prototype matching constitutes already a classification decision. There are prototypes that correspond almost only to examples from a single class. Consider for example the structure containing only one loop, which represents zeros, and the structure containing two loops that touch in a cross point, which represents eights. Other prototypes represent digits from more than one class, e.g. sixes and nines, fives and nines, and fours and sevens (see figure 5). The extracted quantitative features are used to discriminate the digits that have the same structure, but belong to different classes. Depending on the complexity of the structure the feature vector that is presented to the classifier has a length ranging from 19 to 128. For each structure a specialized classifier is trained.

Three options were considered for the classifier: a) a k-Nearest-Neighbor classifier (KNN), b) a feed-forward neural network trained with RProp, c) a neural network built using Cascade-Correlation.

For the KNN classifier the examples from the training set that match the structure of the digit to be recognized are used as reference. An optimized distance function is used to find the k closest examples. Based on the minimal distance and the labels of these k examples the digit is either rejected or it is assigned the label of the closest example. The speed and memory requirements of the KNN classifier can be improved by organizing the prototypes in a tree as done in the CNeT architecture [1].

The feed-forward neural network has a three layer architecture. The input layer receives the feature vector and has variable size. The size of the hidden layer varies depending on the difficulty of the problem. The output layer has ten units, each corresponding to one class. The nets were trained using a CNAPS neurocomputer starting with on-line backpropagation [7] and switching to RProp [6] when the performance did not improve any more. The difference between the activities of the most active output unit and the second active one is used as reject criterium. Training is done for different sizes of the hidden layer (5, 10, 20, 40 and 80 units) and different reject criteria. Each run is terminated when the performance on the test set is best. Finally the best combination of network size and reject criterium is selected based on test set performance. In this way overtraining is prevented and a suitable architecture is found.

Cascade-Correlation [3] networks are able to adapt their architecture to the difficulty of the problem. The sizes of the input and output layers are determined by the length of the feature vector and the number of classes. Training starts with no hidden units. As training proceeds a cascade of hidden units is created. The reject criterium is as above. Training stops when the performance on the test set does not improve any more. A number of trials is performed and the reject criterium is varied to find a good network.

5. Experimental results

Two large databases have been used to validate the performance of the proposed classification system. The first one has been provided by SIEMENS AG and contains about 55 000 gray-level images of handwritten digits that have been extracted from German ZIP-codes. The second database consist of about 120 000 handwritten digits from the well known NIST special databases 1 and 3. Unfortunately, the digits of this database have been binarized, which makes intensive low-pass filtering necessary to prepare the images for the skeletonization operator.

The datasets have been portioned as follows:

Dataset	Training	Test	Validation
SIEMENS	43 372	5 310	6 313
NIST	58 646	30 367	30 727

All digits have been preprocessed as described. The average preprocessing time was about 5ms per digit on a Pentium-133 PC. About 500 structures have been extracted from the training set, but only about 300 were frequent enough to be used as prototypes.

The three classifiers described above have been tested on both databases. Recognition time was less than 1ms for Cascade Correlation and RProp. Additionally some neural classifiers that take normalized 12×16 pixel-images as direct input are tested on the SIEMENS database. These are feed-forward neural networks trained with backpropagation

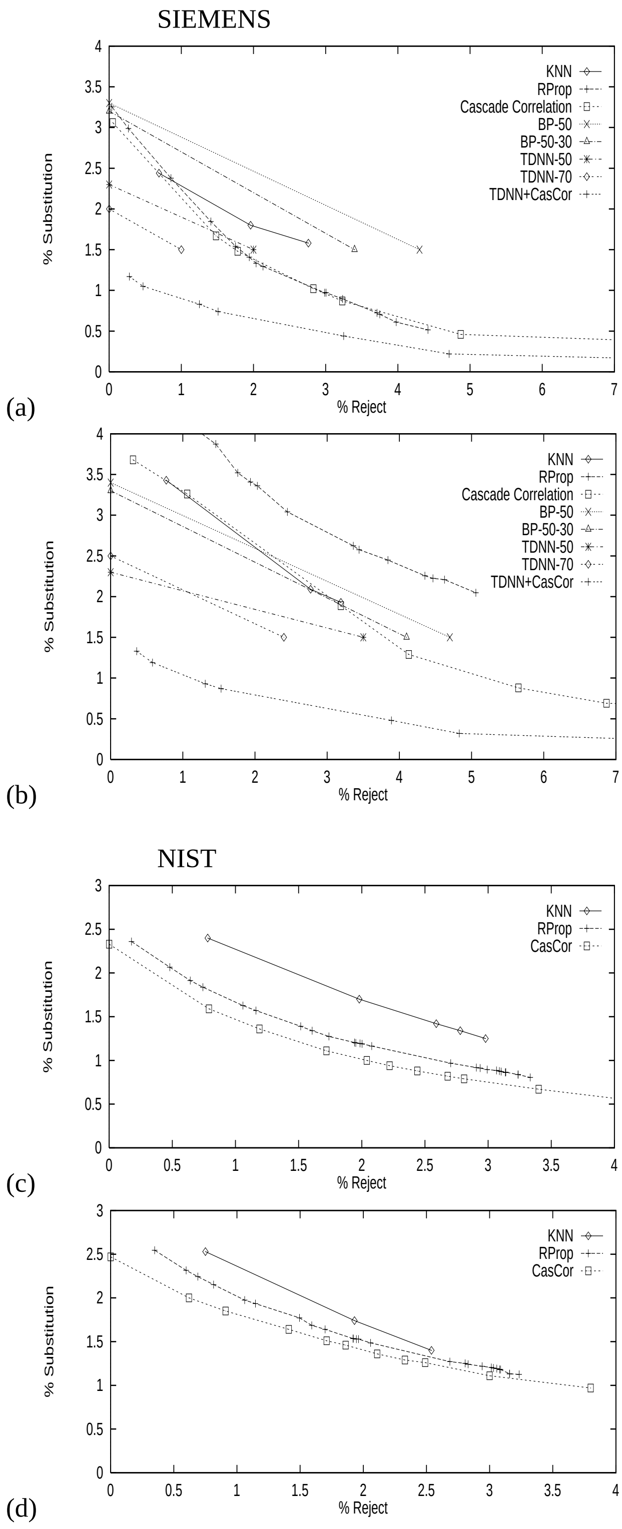

Figure 6. Classification results: (a,b) SIEMENS database, (c,d) NIST database, (a,c) test set, (b,d) validation set

(BP-50, BP-50-30) and Time Delay Neural Networks [7] (TDNN-50, -70).

The results are shown in figure 6. The substitution rates are plotted against the rejection rates. The curves show that there is a tradeoff between reliability and recognition rate. A useful choice of the reject criterium could be such that rejection and substitution rates are equal. In this case the Cascade Correlation classifier recognizes on the SIEMENS database about 96.7% of the test set and about 95.3% of the validation set correctly. The RProp classifier performs almost as well. The KNN recognition rate of 95.4% on the validation set, compared to 96.2% on the test set, indicates that the validation set is more difficult than the test set, since both sets had not been presented to the system prior evaluation. The results of the proposed classifiers are comparable to the other tested systems which reach recognition rates of 97% – 95% on the test set and 96% – 95% on the validation set. For the NIST database the Cascade Correlation classifier has a recognition rate of about 97.5% on the test set and about 96.8% on the validation set. Comparable results have been published by NIST [4].

Most interesting is the combination of the proposed classification system (that uses Cascade Correlation classifiers) and the TDNN classifier. Different methods of combining classifier outputs have been tested on the SIEMENS database. One is to reject a digit, if one of the classifiers rejects or the two labels are different. This yields very low substitution rates of about 0.2%. It is also possible to cascade the classifiers. Here first the TDNN is applied. If the digit is rejected, the structural classifier is applied. About 70% of the rejects were classified correctly, about 8% substituted and about 22% rejected again. This method yields very low rejection rates of about 0.3% while increasing substitutions only slightly. Trading between the extremes such that substitution and rejection rates are equal the recognition rates are about 98.1% on the test set and about 97.8% on the validation set.

6. Conclusion

An off-line method for recognizing handwritten digits has been presented. The method uses intensive preprocessing to extract structural information and quantitative features from images of isolated numerals. Classification is done by a hybrid multi-stage recognition system at a rate of about 170 digits per second.

The performance of the described system has been evaluated on two large databases. The system performs well on the American as well as on the German digits. The proposed method has been combined with a TDNN classifier. The experimental results indicate that both methods are quite orthogonal. Thus there is an advantage in using structural information to enhance an unstructured neural classifier.

In addition to the class label the system delivers a structural description of the digits. This could be used to separate digits from digit blocks on the structural level.

References

[1] S. Behnke and N. B. Karayiannis. Cnet: Competitive neural trees for pattern classification. In *Proceedings ICNN'96–Washington*, volume 1, pages 1439–1444, 1996.

[2] H. Bunke and A. Sanfeliu. *Syntactic and Structural Pattern Recognition – Theory and Applications*. World Scientific Publishing, Singapore, 1990.

[3] S. Fahlmann and C. Lebiere. The cascade correlation learning algorithm. Technical Report CMU-CS-90-100, Carnegie Mellon University, 1990.

[4] P. J. Grother and G. T. Candela. Comparison of handprinted digit classifiers. Technical Report NISTIR 5209, NIST, 1993.

[5] S.-W. Lee and Y. J. Kim. Direct extraction of topographic features for gray scale character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):724–729, 1995.

[6] M. Pfister. *Learning Algorithms for Feed-Forward Neural Networks – Design, Combination and Analysis*. Number 435 in Fortschrittberichte Reihe 10. VDI-Verlag, Düsseldorf, 1996.

[7] R. Rojas. *Neural Networks*. Springer, New York, 1996.

[8] J. Schürmann. *Pattern Classification – A Unified View of Statistical and Neural Approaches*. Wiley-Interscience, New York, 1996.

[9] J. Zhou and T. Pavlidis. Discrimination of characters by a multi-stage recognition process. *Pattern Recognition*, 27(11):1539–1549, 1994.