Recognition of Handwritten ZIP Codes in a Postal Sorting System

M. Pfister, S. Behnke, R. Rojas

In this article, we describe the OCR and image processing algorithms used to read destination addresses from non-standard letters (flats) by the Siemens postal automation system currently in use by the *Deutsche Post AG*.The article concentrates mainly on the two classifiers used to recognize handprinted digits. One of them is a complex *time delayed neural network (TDNN)* used to classify scaled digit-features. The other classifier extracts the structure of each digit and matches it to a number of prototypes. Different digits represented by the same graph are then discriminated by classifiying some of the features of the digit-graph with small neural networks.

1 Introduction

The automatic sorting of so-called 'standard letters' (with envelopes smaller than 11.5 x 23 cm in Germany) is a problem on which companies like AEG ElectroCom (now Siemens ElectroCom) have been working since 1970. The work done by J. Schürmann from the Daimler-Benz Research Center in Ulm, Germany, has been reported in a number of publications, the most important one being [8]. Also N. Srihari from the CEDAR Institute in Buffalo has concentrated on these kinds of problems (see e.g. the CEDAR homepage www.cedar.buffalo.edu or [4]).

However, so-called *non-standard letters* (in this paper shortly called *flats*), constitute a large fraction of the daily postal items to deliver and sort. In Germany flats are larger than standard letters but smaller than 35.3 x 25.3 x 2 cm. Sorting flats is slightly more difficult than sorting standard letters, mainly because of two reasons. Firstly, the adress block is not located in a specific region (like in standard letters) and, secondly, the variety of handwriting that we find is certainly larger than for standard letters.

As the result of a worldwide competition in 1994, the German *Deutsche Post AG (DPAG)* awarded Siemens AG a contract to install a prototype of a flat sorting machine (in German: *Großbrief-Sortieranlage*, abreviated in the sequel as *GSA*) The official name is *SICALIS FSS-C200*. The system described was ordered, developed and installed three years before the acquisition of ElectroCom GmBH (SEC) by Siemens. It is not identical with the SEC OCR. At the time of this writing 150 GSAs have been delivered and installed in about 80 sorting centers all over Germany. They sort millions of flats daily with a GSA throughput up to 20,000 flats per hour. More than 85% of the addresses are found and read correctly by the machines with an error rate of less than 1%. Those letters rejected by the GSA-OCR system are sent to so called *Video Coding Places (VCPs)* to be classified by human experts using a numeric keyboard. Assuming that about 15% of the flats contain handwritten addresses, this means that the recognition rates are 78% for handwritten, and far over 90% for typewritten adresses.

Figure 1 shows a diagram of a GSA postal sorting machine. The envelopes are fed into the GSA and are separated by four *feeders* (1). They are transported on a *conveyor belt* (2) running at about 2m/s. They go below the *linescanning camera* (3), which captures a greyscale image of each letter. While the letter is passed to the *sorter* (6), the GSA-OCR (4) starts processing the image to recognize the destination address. If no valid address is found, the image is delivered to video coding places VCPs (5), where postal workers handle it. If the ZIP code information can be automatically recognized or is entered by a human operator, the letter is dropped in one of 200 mailboxes (7), each one covering different ZIP code ranges.

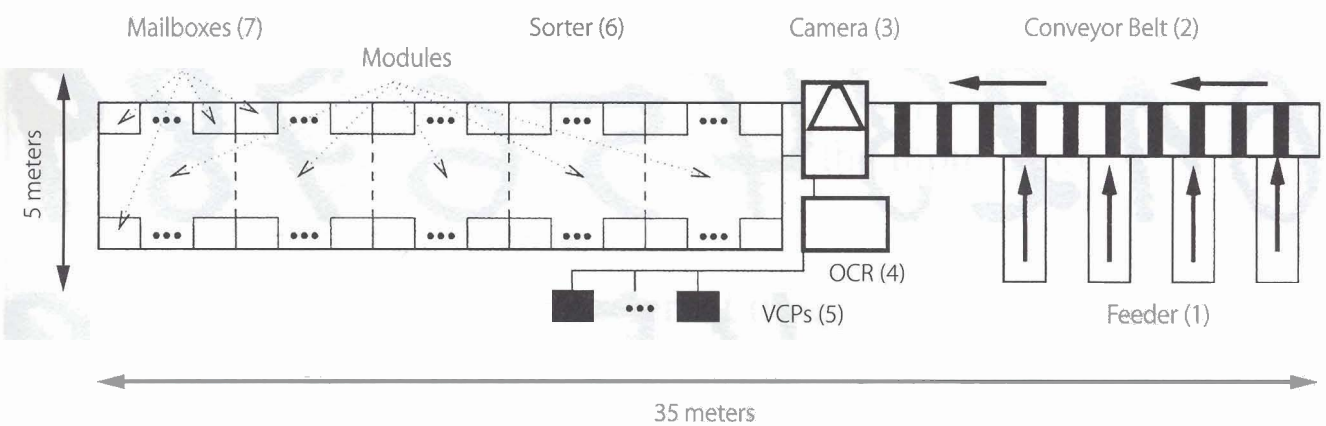


Fig. 1: Siemens flat sorting machine GSA installed in a DPAG sorting center.

In order to solve the difficult problem of automatically reading flat mail addresses with such high accuracy and at an average speed of 6 flats/second, several algorithms out of the image processing, pattern recognition and neural networks field had to be used. Also new algorithms were developed and special hardware was used. In this article we discuss several of the GSA algorithms, some of them roughly, and others, such as the handprinted ZIP code classifiers, more in detail. Together they constitute a real world application of 'applied intelligent algorithms'.

The problem of automatic postal sorting non-standard-letters, as done by the GSA, is very complex. It principally consists of the two main subproblems *finding* the address on the letter and then *reading* it. Both of these tasks include of course many other subtasks.

Since we can not be sure that a region of the letter surface that looks like an address region actually contains the address, these "potential address regions" will further be called *Areas Of Interest, AOIs*. The following list shows the sequence of subtasks that have to be solved to analyze AOIs to finally get the destination ZIP code from the letter.

1. AOI determination,
2. AOI binarization,
3. line segmentation,
4. word segmentation,
5. character segmentation,
6. character classification,
7. interpretation of the classification results,
8. eventual alternative handwritten ZIP code processing,
9. address verification.

This article concentrates on the recognition of handwritten ZIP codes. Anyway these cannot be processed or verified without solving the other tasks of the AOI analysis as listed above.

Nota bene: Developing the GSA and its OCR was a Siemens project involving many researchers and developers from many Siemens divisions. The development of some methods was done in collaboration with the Freie Universität Berlin, namely the ones described in detail in this article.

2 Interpretation of handwritten ZIP codes

We now assume to be faced with the isolated ZIP code block, i.e., in Germany a text block containing five digits, we now want to segment this block into its single digits and classify them. Since the task is rather trivial for printed digits, we concentrate exclusively on the handwriting problem in this article.

2.1 Classifying isolated handprinted digits

The heart of any OCR processing system is a high performance character classification system, since this is the place where the unstructured pixel patterns get their 'meaning', i.e. they are identified as a '6' or an 'A'.

Since the methods described up to here were designed to read handprinted ZIP codes, the problem reduces basically to the recognition of the digits 0 to 9. On the other hand, very high reliability and writer-independency is required for this application. The system has to deal with widely different sizes and slants, with different shapes and width of the strokes.

To produce a system as writer independent and reliable as possible, we decided to combine two methods: a fast but reliable method which analyzes the structure of the digit, and in a second step, a powerful but more computationally intensive pixel oriented neural classifier. Both methods are described in the following sections together with remarks on the combination of the two classifiers.

2.1.1 The TDNN classifier

The TDNN (*Time Delayed Neural Network*) classifier is a high performance neural classifier, which takes the image of a binarized digit, scaled to a fixed size, as its input. In a preprocessing step, some of the digit's variance is removed. The most important ones are the slant and size, as shown in figure 2. For better visualization, the digits were all scaled to the same width.

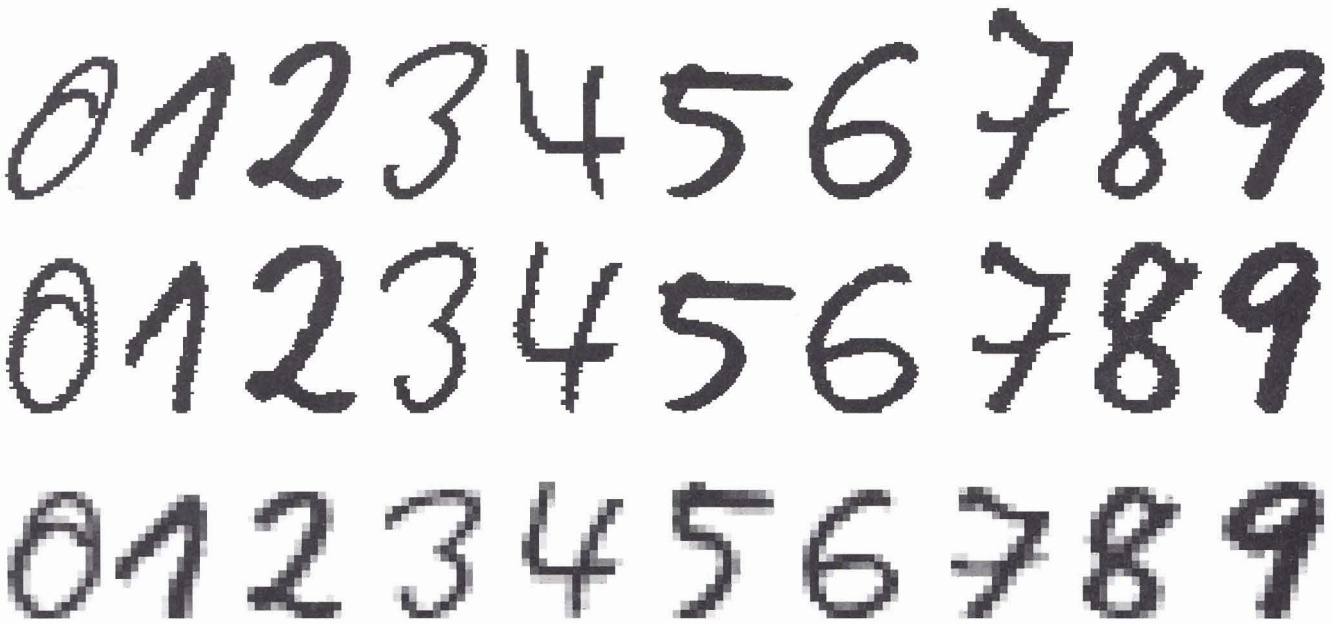


Fig. 2: Normalization of some digits slant and size.

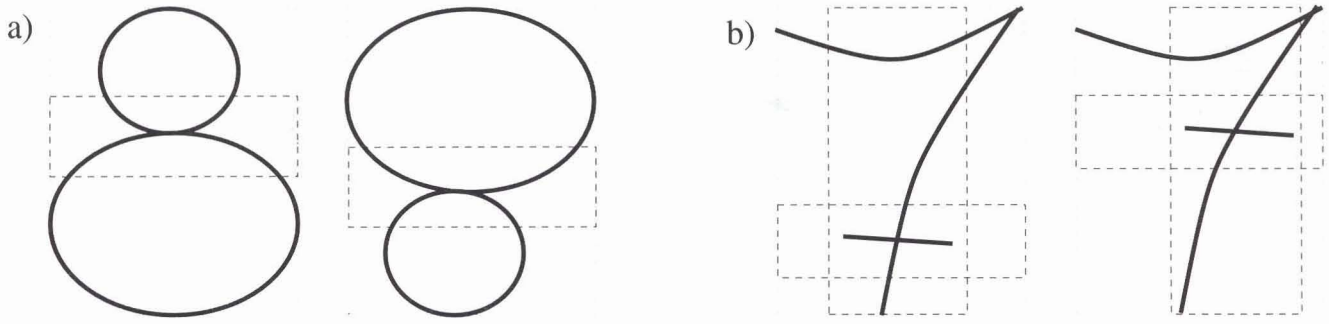


Fig.3: Locations of characteristica of the same digit depend on the writer.

The principal axis of the digit is estimated and then the digit is sheared, so that its axis is vertical after the transformation. Then the resulting digit is scaled to a fixed size of 16 pixels height and 12 pixels width. After this process, the digit is not binarized anymore, so-called *pseudo-grey values* occur.

Beside the variance to be removed by preprocessors, the system has to detect those characteristic features of the digit which also help us humans to discriminate and 'classify' it. These features may be in different locations of the pixel-image, due to nonlinear deformations, as suggested in figure 3.

These characteristic features may be shifted in horizontal (figure 3 a), vertical, or both directions (figure 3 b), depending on the writer, the digit and the preprocessing of the digit.

The general architecture of a TDNN is shown in figure 4 (for more details see e.g. [7]). Each group of input nodes (called the *receptive fields* with shared weights) 'sees' only a small window of the input stream, which 'marches' through the window one position in each time step. The output of the hidden layer is also covered with receptive windows using shared weights. The network's output consists of the sum of squares of the different time steps of the output neurons. This has the advantage that small individual outputs tend to become less important [7].

The *input* of the OCR-TDNN consists of the binarized image of an isolated digit, scaled to a fixed height P_H=16 pixels and a fixed width P_w=12 pixels. There are R_1 receptive fields, each one 'sees' R_0 columns of the picture. These input windows are

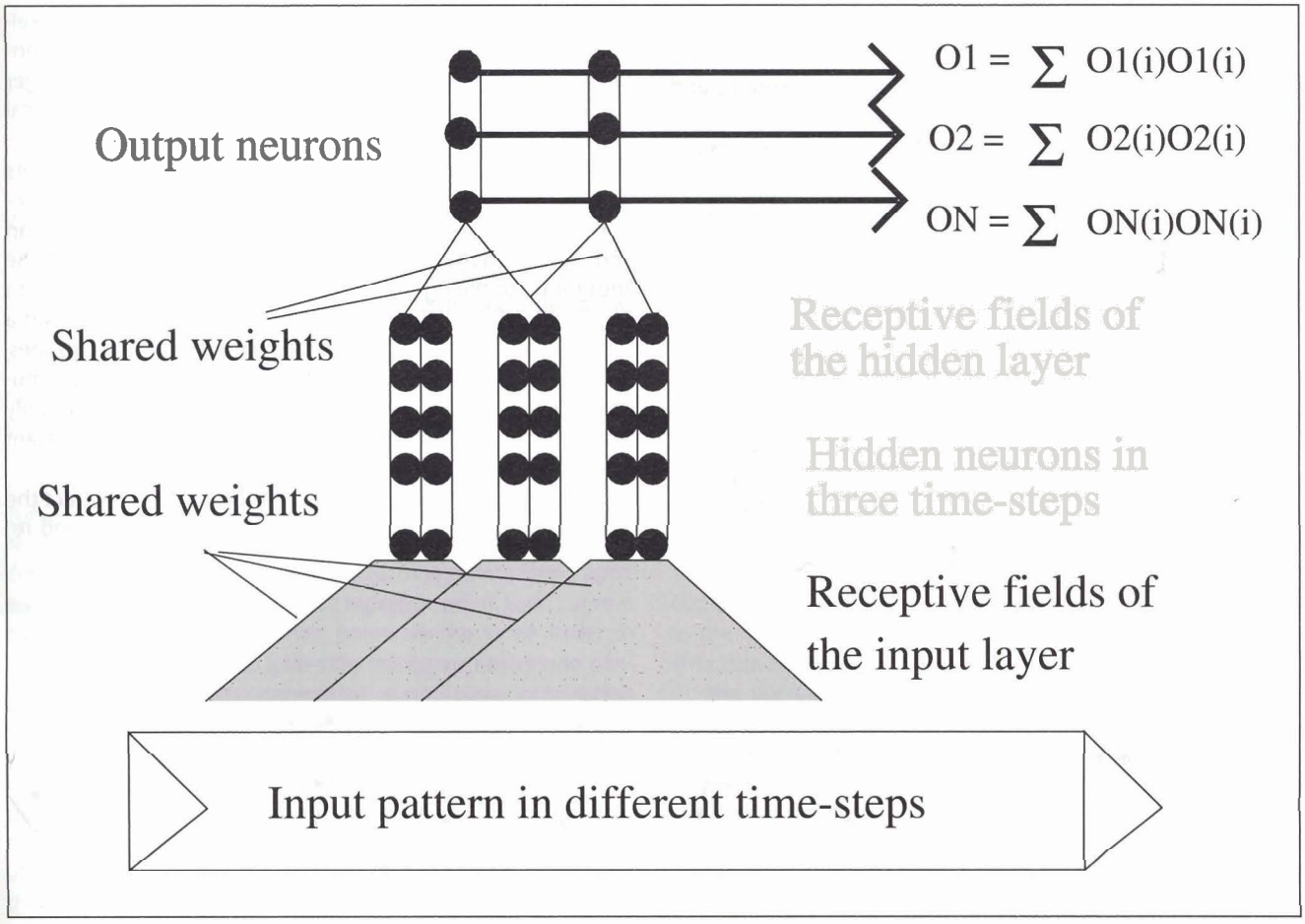


Fig.4: Receptive fields with shared weights in different layers of the TDNN.

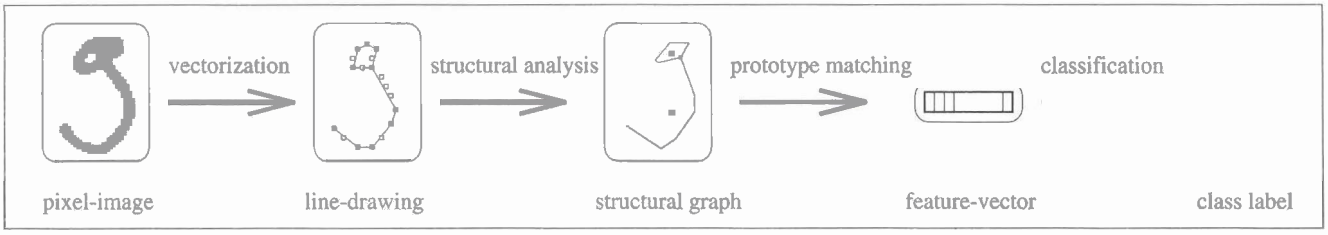


Fig. 5: Stages of the Structural Digit Recognition.

sharing their weights and are shifted by one column of the pixel-image. During the recognition process the columns are moved from left to right through the receptive fields, i.e., a horizontal scanning of the digit is performed. Best results have been obtained with field-sizes $R_0 \approx P_W$, e.g. $R_0=11$ or $R_0=13$. The number of time-states should be limited to $R_1 \leq 5$ because of the computational complexity. Anyway best results were obtained with $R_0+R_1>P_W$, so the total input window should be larger than the digit's width. During the scanning, the image is never moved out of the total input window, and the digit is (virtually) enlarged with white columns left and right to have a well defined input for each node of all receptive fields.

The *hidden layer* consists of N_H hidden nodes in R_1 time-states. This is realized by connecting each group of N_H hidden nodes with the corresponding input window.

The *output layer* of the network consists of 10 nodes, each one representing one class of the digits '0' to '9', which are fully connected to *all* hidden nodes. The output layer thus works without receptive fields. This modification of the standard TDNN is motivated by two ideas. First of all, it accelerates and simplifies the learning algorithm of the TDNN. Secondly, working in this manner, the output layer gets a 'full view' over all time states of all hidden nodes, which also significantly improves the network's performance. During the discrete steps of the scanning process, the output of the output neurons is monitored. The most confident output is regarded as the final recognition, where results obtained from a more centered position of the digit get a little higher scores.

The performance of the TDNN classifier was evaluated on the NIST special database [5], [2]. The TDNN was trained using about 120.000 digits and then tested on an independent validation set. On this set, the TDNN reached maximum recognition rates of up to 99.1% when substituting the other 0.9% and rejecting none. The substitution rate could be lowered to 0.1% when about 4.8% of the digits were rejected.

2.1.2 Classifying digits using structural information

The second classifier implemented in the GSA-OCR uses structural information for the recognition of isolated handprinted digits. This hybrid classifier is described in more detail in [1]. Structural information and quantitative features are extracted in a multi-stage process from the digit's pixel image. The goal is to preserve the information essential for recognition and to discard unnecessary details. Figure 5 shows the stages of the recognition process.

The preprocessing consists of a vectorization of the digit. This produces a line-drawing which is analyzed to construct a structural graph representation. The two-stage decision process matches first the structure of the digit to a structural prototype whose associated neural classifier has been trained to distinguish digits that have the same structure based on extracted quantitative features.

First a vectorization of the image is computed from the skeleton and the next step derives a more abstract digit representation consisting of strokes which are merged to form larger curves. In order to reduce the variability of the input the vertical principal axis and the size of the digits are normalized.

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 to reconstruct the way the digit was drawn. Two strokes are connected and re-

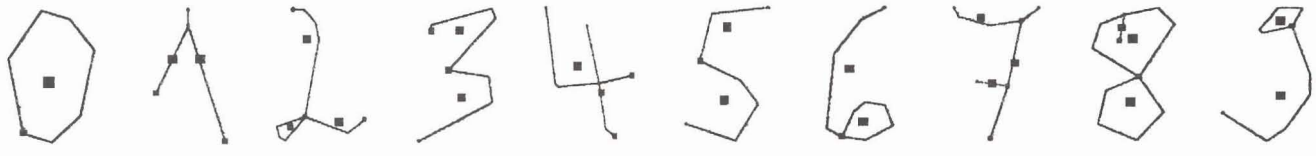


Fig. 6: Curve representation of some digits. Large squares are located at the center of gravity of the curves. Curves run from the middle-sized squares to the small squares.

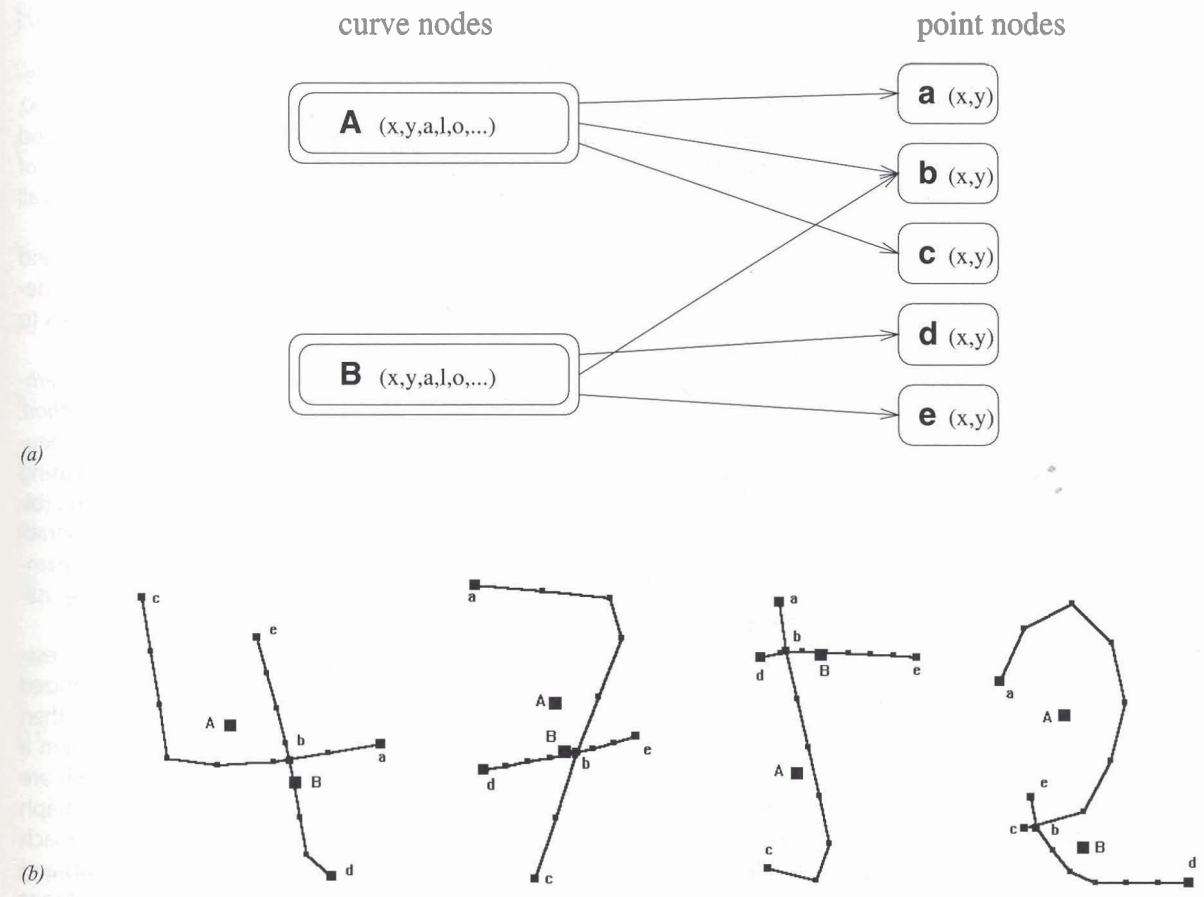


Fig. 7: Attributed structural graph (a) and some assigned digits (b). The first two are typical, the others aren't.

duced 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.

The set of curves found in the previous steps is described now using a bipartite graph as can be seen in figure 7(a). 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 turning points. The graph's edges are derived from the curve representation. Each curve is connected to its characteristic points in the same order in which they appear when following 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.

Prototype matching: The attributed structural graph describes the essential features of the digit to be recognized. Recognition is done in two steps:

i) the structural graph is matched to prototypes that have been extracted from the training set,
ii) for each prototype there is a neural classifier which is used to distinguish digits having the same structure based on the extracted quantitative features.

Classification: In some cases prototype matching constitutes already a classification decision. There are prototypes that correspond almost only to examples from a single class, e.g. perfect zeros or eights. Other prototypes represent digits from more than one class, e.g. sixes and nines, fives and nines, and fours and sevens (see figure 7(b)). 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 neural classifier is trained.

We use Cascade-Correlation [3] networks, since they 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. Training stops when the performance on a test set does not improve any more. A number of trials is performed and a reject criterion is varied to find a good network.

Experimental Results: To validate the performance of the described structural digit classification system, again the well known NIST special databases 1 and 3 have been used [5], [1],

[2]. Unfortunately, the digits of this database have been binarized, which makes intensive low-pass filtering necessary to prepare the images for the skeletonization operator.

About 500 structures have been extracted from the training set, but only about 300 were frequent enough to be used as prototypes. The recognition results show that there is a tradeoff between reliability and recognition rate. A useful choice of the reject criterion could be such that rejection and substitution rates are equal. In this case the structural classifier has recognition rates of about 97.5% on the test set and about 96.8% on the validation set.

These recognition rates by itself would not justify the employment of the structural digit recognition, but the combination with the TDNN makes the hybrid system more reliable. Its distinctive features are its speed, its ability to recognize deformed digits and its high reliability for higher reject rates. The throughput of the entire classification is about 500 characters/second on a Pentium-II/266 system. It is able to classify deformed digits as long as the typical structure is retained. Most substitutions occur due to structural defects of the digits and can be avoided when allowing the classifier to reject ambiguous digits. For the NIST data set a substitution rate of only 0.19% is observed when rejecting 11.55% of the digits.

2.1.3 Combining classifiers

Now that we have two 'digit classifying experts', we have the problem of combining their (eventually) conflicting decisions. There are several ways to deal with such classifier combinations [2], [8]. The two main alternatives are *parallel* and *sequential* combination.

For the parallel combination, both classifiers are run and their results are merged, usually by some kind of voting mechanism or a small third classifier [2], [8]. This kind of combination usually yields very low error rates, but has the disadvantage that *both* classifiers have to be run, which is a very time consuming process and not neccessary for 'easy to recognize' digits.

For the sequential combination, the simplest and thus lesser time consuming classifier C_0 is run first. If it recognizes the digit with high confidence, we are done without having to ask the second classifier C_1. If C_0 does not recognize the digit, the more powerful classifier C_1 is run and the results of both are merged. The disadvantage of this method is of course that any misclassification done by C_0 cannot be overruled by C_1, therefore we must ensure that C_0 yields very low error rates.

In our case, we decided to run the structural classifier first. It is very fast and it also yields very low error rates. It also has the advantage that, due to the structural analysis of its input pattern, it is also able to tell if a pattern is 'far away from being a digit' (e.g. segmentation alternatives, see below). In these cases we can also avoid to run the TDNN.

Combining the two classifiers in such a way, we were able to obtain recognition rates of about 97.5% with less than 0.1% substitution rate on the NIST handprinted digits dataset, or a recognition rate of 99.5% with 0.5% substitutions [5], [2]. This is a significant improvement over the recognition rates yielded by the two classifiers alone.

Conclusion

In this article we have described the OCR system implemented in the Siemens flat reading system *SICALIS FSS - C200*, GSA for short. The GSA is a real world application of neural and image processing algorithms; the machine sorts millions of non-standard letters every week in about 80 sorting centers all over Germany with significant speed and recognition rates.

We especially concentrated on the problem of reading and segmenting handwritten ZIP codes. Two methods were described for classifying handprinted digits, and two methods to segment the ZIP code block into its single digits.

The final classifier is a combination of a *pixel oriented method*, the neural TDNN classifier, and a *structure analyzing method*. Both offer some advantages and disadvantages. With the neural approach, very high recognition rates can be obtained using very little prior knowledge. After the usual preconditioning (binarization, uprighting and scaling to fixed size), feature extraction and classification is done automatically by the TDNN learning algorithm. This requires, on the other hand, more computational effort than the structural approach.

The structural approach is based on extensive preprocessing of the digit to be recognized. The image is first skeletonized and then converted to a graph of strokes, which is then matched against a set of prototype graphs extracted from a training set. Different digits resulting in the same graph are then discriminated using quantitative features of the graph used as input to small neural networks. This structural approach exploits of course extensive prior knowledge about digits and the way they 'could have been' written. This leads to significant recognition speed and very low error rates, since only digits exhibiting a typical structure are recognized. The drawback is, of course, the much larger effort involved in building and training the classifier. The recognition rates are also slightly lower than those of the TDNN.

The same hybrid approach was used for the segmentation algorithms. We have a pixel oriented method, which analyzes binarized connected components and a second structural approach used as backup. In both tasks (classification and segmentation) the two used orthogonal approaches complement each other strongly. The advantage is obvious for the classifiers: The bulk of the digits which are carefully written and thus possess a typical structure are fast and reliably recognized by the structural method. When in doubt, the digits are rejected and are recognized by the powerful TDNN, which relies more on the 'general appearance' of the digit's image, not on structural similarities. When used in combination, more than 99% of the NIST digits can be recognized correctly.

Also segmentation states complement each other. If the 'free' segmentation (i.e. the number of digits to separate is undetermined), done by analysis of the component's outlines, fails, the structural segmentation is called. Again its orthogonal vectorization approach and also its more strict segmentation goal (strictly dividing the ZIP code into five clusters) as well as the eventual prior knowledge will very likely complete the task.

This paper has thus shown that a complex classification task can be better solved by using a hybrid approach: when two or more classifiers base their decision on different sets of features, they can be combined to produce a more reliable system. This hybrid approach is followed consistently in the non-standard letter sorting machine GSA designed and built by Siemens.

References

[1] S. Behnke, M. Pfister and R. Rojas: *Recognition of Handwritten digits using Structural Information*. Proceedings of the ICNN '97, Houston/Texas, 1997.

[2] S. Behnke, M. Pfister and R. Rojas. *A Study on the Combination of Classifiers for Handwritten Digits Recognition*. Proceedings of the ªrd International Workshop on Neural Networks in Applications NN '97, Magdeburg/Germany, 1998.

[3] S. Fahlmann and C. Lebiere: *The Cascade Correlation Learning Algorithm*.Technical Report CMU-CS-90-100, Carnegie Mellon University, 1990.

[4] R. Fenrich, S. Lam and S.N. Srihari: *Optical character readers*newblock Encyclopedia of Computer Science and Engineering, Third Edition, A. Ralston, ed., Van Nostrand, 1992, pp. 993-1000.

[5] M. D. Garris *et al*: *NIST Form-Based Handprint recognition System*. NIST Internal Report 5959, 1994.

[6] M. Pfister: *Learning Algorithms for Feed-forward Neural Networks — Design, Combination and Analysis*. PhD Thesis, FU Berlin, 1995.

[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] M. Schüßler and H. Niemann: *A System for Reading Handwritten Addresses*. Accepted at the 6th International Workshop on the Froniers of Handwriting Recognition, Taejon City/Corea, 1998.

[10] P. Simard, Y. LeCun and J. Drucker: *Improving Performance in Neural Networks using a boosting Algorithm*. newblock Advances in Neural Information Processing Systems. Vol. 5, pp. 42—49, 1993

[11] A. Waibel *et al*: *Phoneme Recognition using time-delay Neural Networks*. IEEE Transactions on Acoustics, Speech and Signal Processing. Vol 37, No. 3, pp. 328—339, 1989.

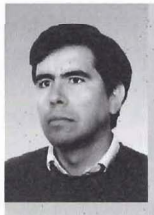
Contact:
Raùl Rojas
Freie Universität Berlin
FB Mathematik und Informatik
Takustraße 9
D-14195 Berlin
email: rojas@inf.fu-berlin.de

Marcus Pfister
Siemens AG
A&D SH 53
Gleiwitzerstraße 555
D-90475 Nürnberg
email: marcus.pfister@nbgm.siemens.de

Sven Behnke
email:
behnke@inf.fu-berlin.de

Marcus Pfister works for the Siemens AG Automation division in Nürnberg, Germany. He received his degree in Mathematics from the Free University of Berlin (FU Berlin) in 1992. In 1996 he received his PhD for his work about „Fast Learning algorithms for Feed-forward Neural Networks", also at the FU Berlin. In 1995 Dr. Pfister joined the Siemens AG to work in the postal sorting machine project. He was responsible for the recognition and classification of hand-printed characters and ZIP codes. His main field of research and interest is image and signal processing, as well as real world applications of pattern recognition and artificial neural networks.

Sven Behnke received his degree in Computer Science in 1997 from Martin-Luther University at Halle-Wittenberg. From May 1996 to November 1998 he did research on handwritten digit recognition for Siemens AG. Currently he is working towards a Ph.D. at the Computer Science Institute of the Free University of Berlin. His research interests include neural architectures for image interpretation, hierarchical image analysis, and neural control.



Raùl Rojas has been a Professor of Computer Science at the Free University of Berlin since 1997. From 1994 to 1997 he was a Professor at the CS department at the University of Halle-Wittenberg. He has been active in the field of neural networks since 1989, after having collaborated with a group at the GMD in Germany that built a parallel Prolog machine. Dr. Rojas got two mathematics degrees from Mexican universities, and afterwards got his PhD and Habilitation at the Free University of Berlin. His main field of research is the theory and applications of connectionist systems.