NPen^{++}: A Writer Independent, Large Vocabulary On-Line Cursive Handwriting Recognition System

Stefan Manke, Michael Finke, and Alex Waibel

University of Karlsruhe
Computer Science Department
D-76128 Karlsruhe, Germany

Carnegie Mellon University
School of Computer Science
Pittsburgh, PA 15213-3890, USA

Abstract

In this paper we describe the **NPen^{++}** *system for writer independent on-line handwriting recognition. This recognizer needs no training for a particular writer and can recognize any common writing style (cursive, hand-printed, or a mixture of both). The neural network architecture, which was originally proposed for continuous speech recognition tasks, and the preprocessing techniques of* **NPen^{++}** *are designed to make heavy use of the dynamic writing information, i.e. the temporal sequence of data points recorded on a LCD tablet or digitizer. We present results for the writer independent recognition of isolated words. Tested on different dictionary sizes from 1,000 up to 100,000 words, recognition rates range from 98.0% for the 1,000 word dictionary to 91.4% on a 20,000 word dictionary and 82.9% for the 100,000 word dictionary. No language models are used to achieve these results.*

1 Introduction

The success and user acceptance of pen computing or multi-modal systems highly depends on the quality of the on-line handwriting recognition engines of these systems. To achieve acceptable recognition performance currently available handwriting recognizers are often either writer dependent or need at least some training for a particular writer to adapt to his handwriting. Additionally people usually have to write in a particular writing style, e.g. hand-printed, or have to use special character shapes defined in the system instead of the usual shapes to get this high performance. All this together makes it very hard for most people using these systems to write as natural as they usually would do on paper. Too small dictionaries are an additional restriction in some of the systems.

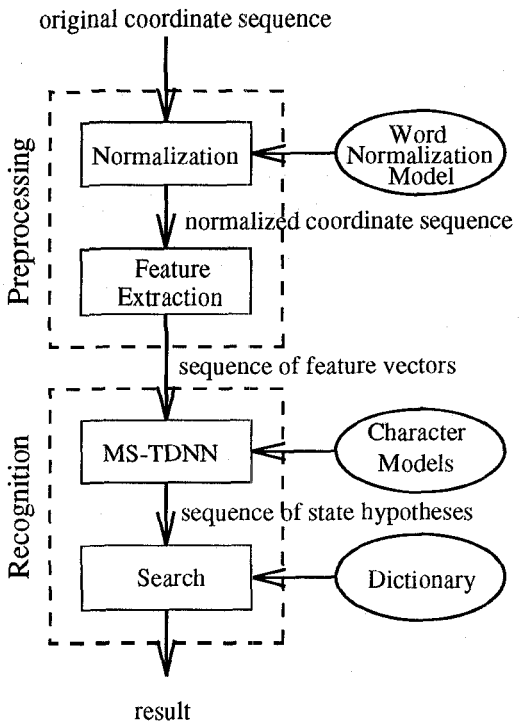

Figure 1: System overview

In this paper we present the **NPen^{++}** on-line handwriting recognition system, which is writer independent and is not constrained to any specific writing style. No training or adaptation for a particular writer is required to achieve high recognition performance even with dictionary sizes up to 100,000 words. **NPen^{++}** can recognize any common writing style, i.e. pure cursive or hand-printed, or a mixture of both. The recognition is not based on bitmaps, which are the only source of information in optical character recognition, but on the dynamic writing information, i.e. the temporal sequence of data points recorded

0-8186-7128-9/95 $4.00

on the LCD tablet or digitizer [10]. The system is designed to make heavy use of this temporal information. **NPen^{++}** (Figure 1) combines a neural network recognizer, which was originally proposed for continuous speech recognition tasks [7, 8], with robust preprocessing techniques, which transform the original sequence of data points into a still temporal sequence of N-dimensional feature vectors.

We have tested the system on the writer independent recognition of isolated words with dictionary sizes from 1,000 up to 100,000 words. Word recognition rates range from 98.0% for the 1,000 word dictionary and 82.9% for the 100,000 word dictionary without using any language model. Even for the largest dictionary used in the experiments the average recognition time for a pattern is still less than 1.5 seconds.

The following section describes the preprocessing techniques used in the system. The architecture and training algorithm of the recognizer are presented in section 3. A description of the experiments to evaluate the system and the results we have achieved on different tasks can be found in section 4.

2 Preprocessing

In optical character recognition (OCR) input usually consists of scanned text (bitmaps) without any temporal information about how the text was written. The fact that this text was generated through a temporal sequence of successive dots is lost in these bitmaps. In contrast to OCR in on-line handwriting recognition the dynamic writing information, i.e. the temporal order of data points produced during handwriting, is recorded on a LCD tablet or digitizer and can be used for recognition [10]. To take advantage of this dynamic writing information it is preserved throughout all our preprocessing steps. The original coordinate sequence $\{(\tilde{x}(t), \tilde{y}(t))\}_{t\in\{0\ldots T'\}}$ recorded on the digitizer is transformed into a new temporal sequence $\boldsymbol{x}_0^T = \boldsymbol{x}_0 \ldots \boldsymbol{x}_T$, where each frame $\boldsymbol{x}_t$ consists of an N-dimensional real-valued feature vector $(f_1(t), \ldots, f_N(t)) \in [-1, 1]^N$.

Several normalization methods are applied to remove undesired variability from the original coordinate sequence [11]. To compensate for different sampling rates and varying writing speeds the coordinates originally sampled to be equidistant in time are resampled yielding a new sequence $\{(\tilde{x}(t), \tilde{y}(t))\}_{t\in\{0\ldots T\}}$ which is equidistant in space. This resampled trajectory is smoothed using a moving average window in order to remove sampling noise. In a final normalization step the goal is to find a representation of the trajectory that is reasonably invariant against rotation and scaling of the input. The idea is to determine the words' baseline and centerline using an Expectation Maximization (EM) approach similar to that described in [6]. The baseline is used to rotate the word to a nearly horizontal orientation and the distance between the baseline and centerline to rescale the word such that the center region of the word is assigned to a fixed size.

From the normalized coordinate sequence $\{(x(t), y(t))\}_{t\in\{0\ldots T\}}$ the temporal sequence $\boldsymbol{x}_0^T$ of N-dimensional feature vectors $\boldsymbol{x}_t = (f_1(t), \ldots, f_N(t))$ is computed (Figure 2). Currently the system uses $N = 15$ features for each data point. The first two features $f_1(t) = x(t) - x(t-1)$ and $f_2(t) = y(t) - b$ describe the relative X movement and the Y position relative to the baseline b. The features $f_3(t)$ to $f_6(t)$ are used to describe the curvature and writing direction in the trajectory [5] (Figure 2(b)). Since all these features are strictly local in the sense that they are local both in time and in space they were shown to be inadequate for modeling temporal long range context dependencies typically observed in pen trajectories [2]. These features can't model effects like that in the neighbourhood of the current data point there might be another part of the trajectory which was written earlier or later as it happens e.g. with t-crossings that cross already written parts of the trajectory. Therefore, nine additional features $f_7(t)$ to $f_{15}(t)$ representing 3×3 bitmaps were included in each feature vector (Figure 2(a)). These so-called context bitmaps are basically low resolution, bitmap-like descriptions of the coordinate's proximity, which were originally described in [2].

Thus, the input representation as shown in Figure 2 combines strictly local features like writing direction and curvature with the context bitmaps, which are still local in space but global in time. That means, each point of the trajectory is visible from each other point of the trajectory in a small neighbourhood. By using these context bitmaps in addition to the local features, important information about other parts of the trajectory, which are in a limited neighbourhood of a coordinate, are encoded.

3 The NPen^{++} recognizer

The **NPen^{++}** recognition component integrates recognition and segmentation of words into a single network architecture, the so-called Multi-State Time Delay Neural Network (MS-TDNN). The MS-TDNN, which was originally proposed for continuous speech

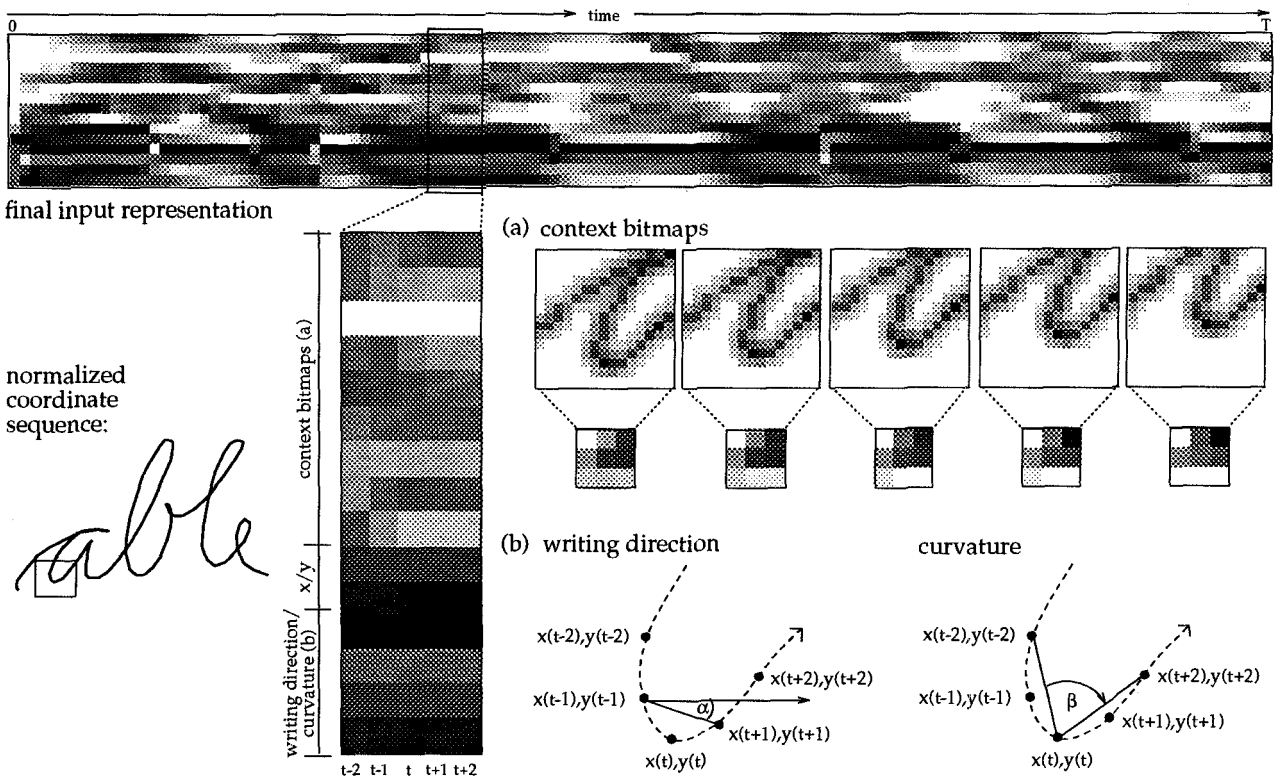

Figure 2: Feature extraction for the normalized word "able". The final input representation is derived by calculating a 15-dimensional feature vector for each data point, which consists of a context bitmap (a) and information about the curvature and writing direction (b).

recognition tasks [7, 8], combines the high accuracy pattern recognition capabilities of a TDNN [9, 5] with a non-linear time alignment algorithm (dynamic time warping) for finding stroke and character boundaries in isolated handwritten words.

3.1 Modeling assumptions

Let $W = \{w_1, \ldots w_K\}$ be a dictionary consisting of K words. Each of these words w_i is represented as a sequence of characters $w_i \equiv c_{i_1} c_{i_2} \ldots c_{i_k}$ where each character c_j itself is modelled by a three state hidden markov model $c_j \equiv q_j^0 q_j^1 q_j^2$. The idea of using three states per character is to model explicitly the initial, middle and final section of the characters. Thus, w_i is modelled by a sequence of states $w_i \equiv q_{i_0} q_{i_1} \cdots q_{j_{3k}}$. In these word HMMs the self-loop probabilities $p(q_{i_j}|q_{i_j})$ and the transition probabilities $p(q_{i_j}|q_{i_{j-1}})$ are both defined to be $\frac{1}{2}$ while all other transition probabilities are set to zero.

During recognition of an unknown sequence of feature vectors $\boldsymbol{x}_0^T = \boldsymbol{x}_0 \ldots \boldsymbol{x}_T$ we have to find the word $w_i \in W$ in the dictionary that maximizes the a-posteriori probability $p(w_i|\boldsymbol{x}_0^T, \theta)$ given a fixed set of parameters θ and the observed coordinate sequence. That means, a written word will be recognized such that

$$w_j \quad = \quad \text{argmax}_{w_i \in W} p(w_i|\boldsymbol{x}_0^T, \theta).$$

In our Multi-State Time Delay Neural Network approach the problem of modeling the word posterior probability $p(w_i|\boldsymbol{x}_0^T, \theta)$ is simplified by using Bayes' rule which expresses that probability as

$$p(w_i|\boldsymbol{x}_0^T, \theta) \quad = \quad \frac{p(\boldsymbol{x}_0^T|w_i, \theta) P(w_i|\theta)}{p(\boldsymbol{x}_0^T|\theta)}.$$

Instead of approximating $p(w_i|\boldsymbol{x}_0^T, \theta)$ directly we define in the following section a network that is supposed to model the likelihood of the feature vector sequence $p(\boldsymbol{x}_0^T|w_i, \theta)$.

3.2 The MS-TDNN architecture

In Figure 3 the basic MS-TDNN architecture for handwriting recognition is shown. The first three layers constitute a standard TDNN with sliding input

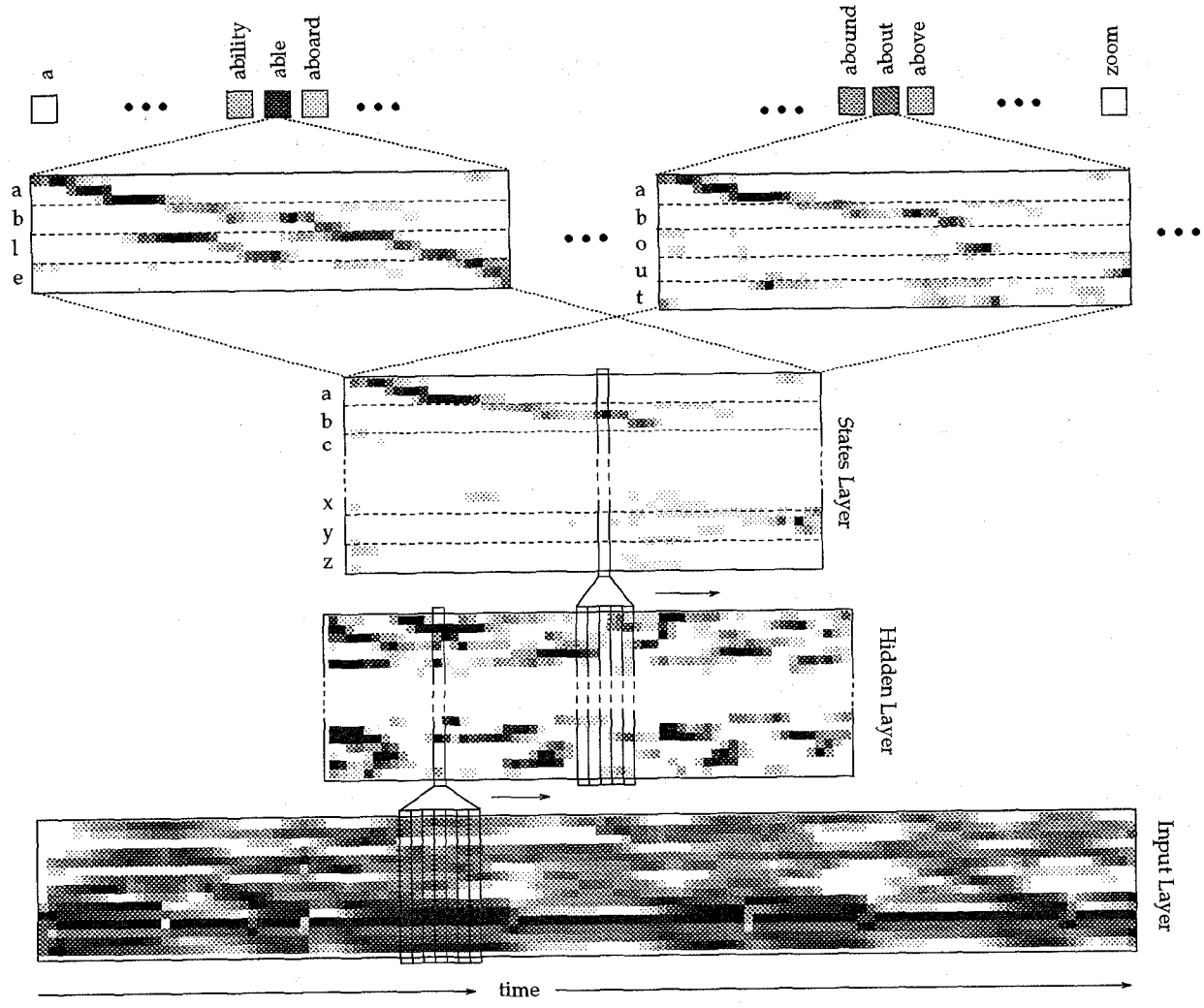

Figure 3: The Multi-State TDNN architecture, consisting of a 3-layer TDNN to estimate the a posteriori probabilities of the character states combined with word units, whose scores are derived from the word models by a Viterbi approximation of the likelihoods.

windows in each layer. In the current implementation of the system, a TDNN with 15 input units, 40 units in the hidden layer, and 78 state output units is used. There are 7 time delays in the input layer and 5 time delays in the hidden layer.

The softmax normalized output of the states layer is interpreted as an estimate of the probabilities of the states q_j given the input window $\boldsymbol{x}_{t-d}^{t+d} = \boldsymbol{x}_{t-d} \ldots \boldsymbol{x}_{t+d}$ for each time frame t, i.e.

$$p(q_j|\boldsymbol{x}_{t-d}^{t+d}) \approx \frac{\exp(\eta_j(t))}{\sum_k \exp(\eta_k(t))} \quad (1)$$

where $\eta_j(t)$ represents the weighted sum of inputs to state unit j at time t. Based on these estimates, the output of the word units is defined to be a Viterbi approximation of the log likelihoods of the feature vector sequence given the word model w_i, i.e. $\log p(\boldsymbol{x}_0^T|w_i)$ is approximated by

$$\begin{aligned} &\max_{q_0^T} \sum_{t=1}^{T} \log p(\boldsymbol{x}_{t-d}^{t+d}|q_t, w_i) + \log p(q_t|q_{t-1}, w_i) \\ \approx &\max_{q_0^T} \sum_{t=1}^{T} \log \frac{p(q_t|\boldsymbol{x}_{t-d}^{t+d})}{p(q_t)} + \log p(q_t|q_{t-1}, w_i). \end{aligned}$$

Here, the maximum is over all possible sequences of states $q_0^T = q_0 \ldots q_T$ given a word model, $p(q_t|\boldsymbol{x}_{t-d}^{t+d})$ refers to the output of the states layer as defined in (1) and $p(q_t)$ is the prior probability of observing a state q_t estimated on the training data.

3.3 Training algorithm

During training the goal is to determine a set of parameters θ that will maximize the posterior probability $p(w|\boldsymbol{x}_0^T, \theta)$ for all training input sequences. But in order to make that maximization computationally feasible even for a large dictionary system we had to simplify that maximum a posteriori approach to a maximum likelihood training procedure that maximizes $p(\boldsymbol{x}_0^T|w, \theta)$ for all words instead.

The first step of our maximum likelihood training is to bootstrap the recognizer using a subset of approximately 2,500 words of the training set that were labeled manually with the character boundaries to adjust the paths in the word layer correctly. After training on this hand-labeled data, the recognizer is used to label another larger set of unlabeled training data. Each pattern in this training set is processed by the recognizer. The boundaries determined automatically by the Viterbi alignment in the target word unit serve as new labels for this pattern. Then, in the second phase, the recognizer is retrained on both data sets to achieve the final performance of the recognizer.

4 Experiments and results

We have tested our system on different writer independent tasks with dictionary sizes ranging from 1,000 up to 100,000 words. The character set used in the dictionaries consists of all lower case and upper case letters. The system was trained on approximately 5,700 patterns from a 7,000 word dictionary, written by 80 different writers. The test was performed on data from an independent set of 40 writers.

All data used in these experiments was collected at the University of Karlsruhe, Germany. Only minimal instructions were given to the writers. The writers were asked to write as natural as they would normally do on paper, without any restrictions in writing style. The consequence is, that the database is characterized by a high variety of different writing styles, ranging from hand-printed to strictly cursive patterns or a mixture of both writing styles (for example see Figure 4). Additionally the native language of the writers was German, but the language of the dictionary is English. Therefore, frequent hesitations and corrections can be observed in the patterns of the database. But since this sort of input is typical for real world applications, a robust recognizer should be able to process these distorted patterns, too. From each of the writers a set of 50-100 isolated words, choosen randomly from the 7,000 word dictionary, was collected.

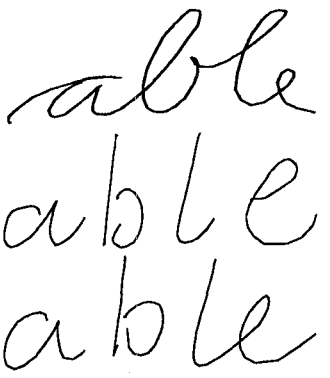

Figure 4: Different writing styles in the database: cursive (top), hand-printed (middle) and a mixture of both (bottom)

All dictionaries used for the experiments were selected randomly from the ARPA Wall Street Journal Task (WSJ), which was originally defined for speech recognition evaluations.

Table 1: Writer independent recognition results

Task	Dictionary Size	Test Patterns	Recognition Rate
wsj_1k	1,000	800	98.0%
wsj_5k	5,000	2,500	95.3%
wsj_10k	10,000	2,500	93.4%
wsj_20k	20,000	2,500	91.4%
wsj_100k	100,000	2,500	82.9%

Word recognition results for dictionary sizes from 1,000 to 100,000 words are shown in Table 1. In the current system the average recognition time (preprocessing + recognition) ranges from 1.0 second for the 1,000 word dictionary to 1.2 seconds for the 20,000 and 1.5 seconds for the 100,000 word dictionary, measured on a DEC Alpha AXP 3000/600. This shows that recognition time is virtually independent of the dictionary size. Approx. 40% of the total recognition time is spent for preprocessing, which has'nt been optimized for speed yet.

For approx. 60% of the errors the system makes on our test set the correct answer is in the list of the 5 best words found by the system, as can be seen in Figure 5. For $N = 1 \ldots 10$ this figure shows the recognition rates for the 1,000, 5,000, 10,000 and 20,000 word dictionaries if the correct word is in the list of the N best hypotheses found by the system. For $N > 10$ no significant performance improvements are observed. Experience from continuous speech recognition and the fact that the correct answer is often very close to an

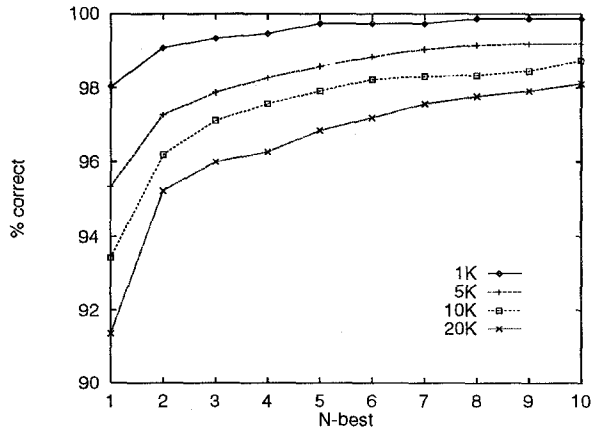

Figure 5: Recognition results with respect to the dictionary size if the $N = 1 \ldots 10$ best words are counted as correct.

incorrect output of the recognizer shows that we can expect further improvements of the word recognition rate by using language models for the recognition of sentences.

5 Conclusions

In this paper we have presented the **NPen**$^{++}$ system, a connectionist recognizer for writer independent on-line cursive handwriting recognition. This system combines a robust input representation, which preserves the dynamic writing information, with a neural network integrating recognition and segmentation in a single framework. This architecture has been shown to be well suited for handling temporal sequences as provided by this kind of input.

Evaluation of the system on different dictionary sizes has shown recognition rates from 98.0% for a 1,000 word dictionary to 82.9% for the 100,000 word dictionary. Even for the largest dictionary used in the experiments the average recognition time for a pattern is still less than 1.5 seconds. These results are especially promising because they were achieved with a small training set compared to other systems (e.g. [4]). As can be seen in Table 1, the system has proved to be virtually independent of the dictionary. Though the system was trained on rather small dictionaries, it generalizes well to completely different and much larger dictionaries. Recognition time doesn't depend on the dictionary sizes, but mainly on the length of the input patterns.

Work is in progress to extend this system to the full character set available on an english computer keyboard and to the recognition of sentences using language models to achieve further improvements of the word recognition rate.

References

[1] S. Manke and U. Bodenhausen, "A Connectionist Recognizer for Cursive Handwriting Recognition", *Proceedings of the ICASSP-94*, Adelaide, April 1994.

[2] S. Manke, M. Finke, and A. Waibel, "Combining Bitmaps with Dynamic Writing Information for On-Line Handwriting Recognition", *Proceedings of the ICPR-94*, Jerusalem, October 1994.

[3] S. Manke, M. Finke, and A. Waibel, "The Use of Dynamic Writing Information in a Connectionist On-Line Cursive Handwriting Recognition System", *Advances in Neural Information Processing 7*, MIT Press, Cambridge (MA), 1995.

[4] M. Schenkel, I. Guyon, and D. Henderson, "On-Line Cursive Script Recognition Using Time Delay Neural Networks and Hidden Markov Models", *Proceedings of the ICASSP-94*, Adelaide, April 1994.

[5] I. Guyon, P. Albrecht, Y. Le Cun, W. Denker, and W. Hubbard, "Design of a Neural Network Character Recognizer for a Touch Terminal", *Pattern Recognition*, 24(2), 1991.

[6] Y. Bengio and Y. LeCun. "Word Normalization for On-Line Handwritten Word Recognition", *Proceedings of the ICPR-94*, Jerusalem, October 1994.

[7] P. Haffner and A. Waibel, "Multi-State Time Delay Neural Networks for Continuous Speech Recognition", *Advances in Neural Information Processing Systems (NIPS-4)*, Morgan Kaufman, 1992.

[8] C. Bregler, H. Hild, S. Manke, and A. Waibel, "Improving Connected Letter Recognition by Lipreading", *Proceedings of the ICASSP-93*, Minneapolis, April 1993.

[9] A. Waibel, T. Hanazawa, G. Hinton, K. Shiano, and K. Lang, "Phoneme Recognition using Time-Delay Neural Networks", *IEEE Transactions on Acoustics, Speech and Signal Processing*, March 1989.

[10] C. Tappert, C. Suen, and T. Wakahara, "The State of the Art in On-Line Handwriting Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8), 1990.

[11] W. Guerfali and R. Plamondon, "Normalizing and Restoring On-Line Handwriting", *Pattern Recognition*, 16(5), 1993.