Proc. IWFHR 2006, La Baule, France

Novel Hybrid NN/HMM Modelling Techniques for On-line Handwriting Recognition

Joachim Schenk and Gerhard Rigoll
Institute for Human-Machine Communication
Technische Universität München
Arcisstraße 16, 80333 München
{schenk, rigoll}@mmk.ei.tum.de

Abstract

In this work we propose two hybrid NN/HMM systems for handwriting recognition. The tied posterior model approximates the output probability density function of a Hidden Markov Model (HMM) with a neural net (NN). This allows a discriminative training of the model. The second system is the tandem approach: A NN is used as part of the feature extraction, and then a standard HMM apporach is applied. This adds more discrimination to the features.

In an experimental section we compare the two proposed models with a baseline standard HMM system. We show that enhancing the feature vector has only a limited effect on the standard HMMs, but a significant influence to the hybrid systems. With an enhanced feature vector the two hybrid models highly outperform all baseline models. The tandem approach improves the recognition performance by 4.6 % (52.9 % rel. error reduction) absolute compared to the best baseline HMM.

Keywords: On-line handwriting recognition, HMM, NN, hybrid, tandem, tied posteriors

1. Introduction

Adopted from automated speech recognition (ASR), Hidden-Markov-Models (HMMs, [1]) have proven their power in modelling time-dynamic sequences of variable lengths. HMMs also allow to compensate statistical variations in those sequences. Due to this property, they have become quite popular in on-line (cursive) handwriting recognition [2–4].

In a common recognition system, each symbol (e. g. strokes, letters, and words) is represented by one single HMM. The parameters of each HMM can be trained using the Baum-Welch algorithm [5]. Noteworthy, the parameters of the HMMs are trained independently of all other classes. Besides standard HMMs are not able to take context in terms of inter symbol dependencies into account.

These disadvantages do not count for a neural net classifier (NN, [6; 7]), which is trained discriminatively. That means, each output node is optimized in respect to all other output nodes, respectively all other classes. Also, inter symbol context is included easily by extending the NN's input vector [6]. However, NNs lack the ability to handle time varying sequences and their statistical variations which occur in handwriting recognition.

To maintain all, the handling of sequences with varying lengths, discriminative training, as well as context observation, hybrid NN/HMM approaches were introduced. They join the benefits of both classifiers. These systems have been successfully applied to ASR, showing the advantage of hybrid systems in high recognition rates [6; 8].

In this work we therefore apply hybrid NN/HMM models to the problem of cursive on-line handwriting recognition. We propose two approaches, namely the tied posterior (TP, [8]) and the tandem [9].

The next section gives a brief overview of HMMs and NNs. Afterwards we introduce the two hybrid approaches for handwriting recognition. In section 3 we present the features used for recognition. Our two systems and a baseline system are evaluated in section 4. Finally a conclusion is given in section 5.

2. NN/HMM Hybrid Modelling Techniques

In this section we briefly summarize HMMs and NNs, and give a common notation. We then introduce the hybrid TP and tandem approaches for handwriting recognition.

2.1. HMMs

For recognition with HMMs, each symbol is modelled by one HMM. Each HMM i (and thereby each symbol) is represented by a set of parameters $\lambda_i = (\mathbf{A}, \mathbf{B}, \vec{\pi})$, where $\mathbf{A}$ denotes the transition matrix, $\mathbf{B}$ the matrix of output probabilities, and $\vec{\pi}$ the initial state distribution [1]. There are two basic types of HMMs: discrete and continuous ones. In the case of discrete HMMs, the matrix $\mathbf{B}$ contains discrete probabilities, corresponding to each possible observation. In literature some hybrid NN/discrete HMM systems are known, e. g. a NN vector quantizer is used to derive discrete observations [10]. In the case of continuous HMMs, $\mathbf{B}$ is not directly a matrix, but represents mixtures of Gaussians or some other PDFs. In our work we concentrate on continuous or semi-continuous HMMs [11].

Given some training data $\mathbf{O}_i = (\vec{O}_1, \vec{O}_2, \ldots, \vec{O}_L)$ for class i, an HMM can be trained with the well known EM-method, in the case of HMMs known as Baum-Welch-algorithm [5]. The aim of the training is to maximize $p(\mathbf{O}_i|\lambda_i)$. That is maximizing the probability that HMM i produces the observations $\mathbf{O}_i$. For the training of HMM i only representatives of the class i are used. The resulting models are independent from each other. This is known as non-discriminative training. Considering a handwriting recognition system, where every HMM represents a letter, the HMM corresponding to letter "a" is only trained with examples of the class "a". This HMM neither takes the number of classes into account, nor does it know the similar looking letter "o". Thus this training does not maximize the distance between the classes.

Finally recognition is performed by presenting the unknown pattern $\vec{s}$ to all HMMs λ_i and select the model k_i with

$$k_i = \underset{i}{\operatorname{argmax}}\, p(\vec{s}|\lambda_i) \tag{1}$$

with the highest likelihood. This is done by the Viterbi-algorithm [12], which can also perform a segmentation of the input vector $\vec{s}$.

2.2. Neural Nets

In the above section we showed the main disadvantage of HMMs: non-discriminative training. In contrast to HMMs, NN classifiers allow discriminative training. That means each class is trained in respect to all other classes. In this work two arbitrary NN structures are used: the multi layer perceptron (MLP, [6]) and the recurrent neural net (RNN, [7]).

The MLP in this work consists of M input and N output nodes separated by a number of hidden nodes. The output of each layer forms the input of the consecutive layer. Each output node represents one symbol class. The output vector $\vec{y}$ is, in this case, calculated by

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} = \vec{F}_0 \left(\mathbf{V}^{\mathrm{T}} \vec{F}_h \left(\mathbf{W}^{\mathrm{T}} \vec{x} \right) \right), \tag{2}$$

depending on the input vector $\vec{x}$, the hidden layer $\mathbf{W}$, and the output layer $\mathbf{V}$. $\vec{F}_0$ and $\vec{F}_h$ are a set of non-linear functions. The discriminative training of the MLP is performed by the back propagation algorithm, described in [13]. The initialization of the NN weights is chosen randomly.

In contrast, RNNs use feedback-nodes: Instead of a hidden layer the output nodes are feeded back to the input nodes to perform classification on time varying patterns [7]. The RNN used in this work are trained via resilient propagation (RPROP) [14; 15].

In [6] it is shown, that a NN (either MLP or RNN) is capable of calculating the a posteriori probability $p(\rho_i|\vec{x})$ for a class ρ_i given the input vector $\vec{x}$. To make $p(\rho_i|\vec{x})$ a valid probability, namely $0 \leq p(\rho_i|\vec{x}) \leq 1$ and

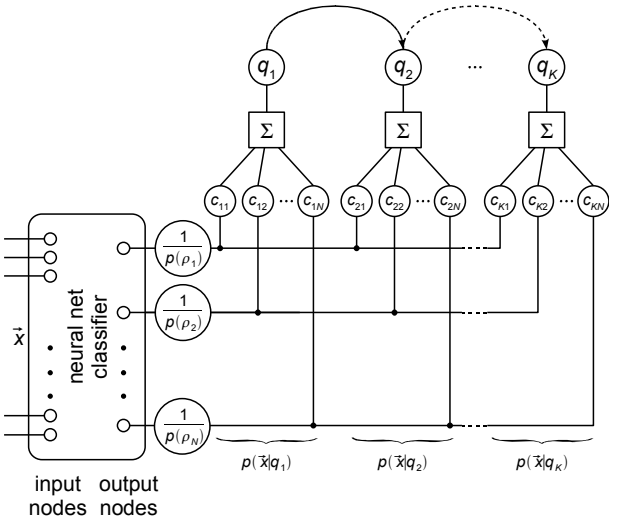

Figure 1. Combination of a neural net and semi-continuous HMM for the tied posterior approach.

$\sum_{j=1}^{N} p(\rho_j|\vec{x}) = 1$ we choose the non-linearity F_0 as the "softmax" function

$$F_0(\xi_i) = \frac{\exp(\xi_i)}{\sum_{i=1}^{N} \exp \xi_i}, \tag{3}$$

where ξ_i is the result of the output layer before applying the non-linearity. Recognition is then performed by choosing the output node with the highest a posteriori probability $p(\rho_i|\vec{x})$ for the unknown sample $\vec{x}$

$$k_i = \underset{i}{\operatorname{argmax}}\, p(\rho_i|\vec{x}). \tag{4}$$

Although NNs support discriminative training, they only work for single, segmented letters with fixed lengths.

2.3. Tied Posteriori

To combine the benefits of both the NN and the HMM classifier, both models are combined in hybrid NN/HMM systems. We now show how such hybrid systems, the TP and the tandem approach, join the advantages of the two models and how they can be applied to handwriting recognition.

The principal idea behind the tied posterior approach is to approximate the output probability density function b_j of each state j of an HMM by the output nodes of an NN. This is illustrated in Figure 1. An input vector $\vec{x}$, consisting of e. g. handwriting features, is preprocessed by a NN classifier estimating the a posteriori probability $p(\rho_i|\vec{x})$ that the input vector $\vec{x}$ belongs to the symbol class ρ_i. The NN's output, weighted by the a priori probability $\frac{1}{p(\rho_i)}$ of each symbol, forms the PDFs used for every state of all HMMs.

As all HMMs use the same NN as source for their PDFs, we are using semi-continuous HMMs. Mathematically the output probability b_j of each state q_j of the HMM

is computed by a weighted sum of a fixed number I of PDFs:

$$b_j(\vec{x}) = p(\vec{x}|q_j) = \sum_{i=1}^{I} c_{ji} \cdot p(\vec{x}|\rho_i). \quad (5)$$

The conditional probability $p(\vec{x}|\rho_i)$ of Eq. 5 is not available from the NN. However, the output of the NN resembles the a posterior probability $p(\vec{x}|\rho_i)$. To replace $p(\vec{x}|\rho_i)$ by $p(\rho_i|\vec{x})$ we can use a scaled likelihood [6]. This probability can be expressed with the a posterior probabilities $p(\rho_i|\vec{x})$ and the priori class probabilities $p(\rho_i)$, which can be estimated using the training data:

$$\frac{p(\vec{x}|\rho_i)}{p(\vec{x})} = \frac{p(\rho_i|\vec{x})}{p(\rho_i)}. \quad (6)$$

Eq. 5 and 6 lead to the tied posterior approach in which the output probabilities b_j can be computed as:

$$b_j = p(\vec{x}|q_j) = \sum_{i=1}^{I} c_{ji} \cdot \frac{p(\rho_i|\vec{x})}{p(\rho_i)}. \quad (7)$$

Hence, the a posteriori probabilities computed by the NN are "tied" together. After training the NNs, the Baum-Welch Algorithm [5] is used to train the semi-continuous HMMs. In contrast to other hybrid approaches, where each output node of the NN is linked to just one state of a HMM [6], we are able to use various HMM topologys as any number of NN output nodes may contribute to any state's output PDF. Therefore the advantages of HMMs mentioned in the introduction are kept whilst discriminance is added by the PDFs used in Eq. 7 generated by the NN.

2.4. Tandem

The TP-approach uses the output of a neural net to *approximate* the PDF of an HMM state. In contrast the tandem approach uses G standard Gaussian mixtures, with

$$b_j = \sum_{m=1}^{G} c_{jm} \mathcal{N}(\vec{\mu}_{jm}, \mathbf{U}_{jm}) \quad (8)$$

to *estimate* the output of the neural net. Thereby, the standard HMM procedures presented in [1] can be used to estimate the mean vector $\vec{\mu}_{jm}$ and the covariance matrix $\mathbf{U}_{jm}$ for the m-th Gaussian mixture component in state j. The neural net therefore acts as a part of the feature extraction, which adds additional discrimination to the features.

Due to their distribution, the probabilities of the output layer cannot be estimated directly by Gaussians [9]. For that reason, either the non-linearity of the neural net's output layer is omitted (namely F_0 in Eq. 2), or the output layer emissions are logarithmized. In this paper, the logarithm is taken by [16]

$$l_i = \log(p(p_i|\vec{x})) - \frac{1}{N} \sum_{j}^{N} \log(p(p_j|\vec{x})). \quad (9)$$

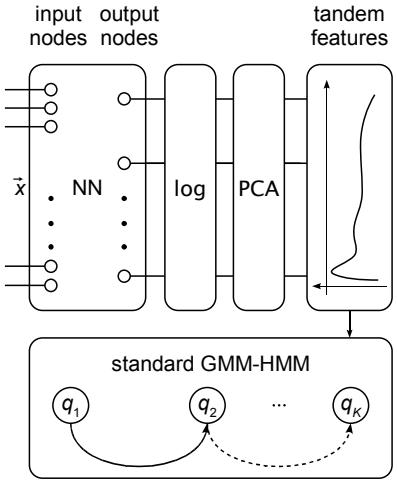

Figure 2. Gaining tandem features from a neural net.

Additionally a PCA is applied to decorrelate the components of the feature vector in oder to use a variance vector instead of the covariance matrix $\mathbf{U}_{jm}$. In Figure 2 the principal use of the neural net in the tandem system is shown. As for the TP aproach, a input vector $\vec{x}$ is preprocessed by the NN. In contrast to the TP, a PCA is applied on the logarithmized output values of the NN. In that way the so called "tandem features" are generated and the standard procedures for continuous HMM training [1] can be performed on them.

As we use standard HMMs, their advantages described in the introduction still count, however by transforming the original features into tandem features additional discriminance is added.

3. Features

In our system handwriting is recorded with a digitizing tableau and stored in cartesian coordinates, including information on the pen's pressure. To describe the trajectory of the pen, data is captured with a constant sample rate of $T_s = 5$ ms. As the sample rate is fixed, two characters with the same size and style result in complete different temporal sequences when written in different speeds. To avoid this time varying effect, the data is resampled to archive equidistant sampling in space rather than in time. To cope with varying sizes each input sequence is normalized to the distance of the upper and lower baseline.

Except for the resampling and normalization, no further preprocessing steps have been taken. Then both on- and off-line features are derived from the pen's input trajectory[17; 18]: As *on-line features* we extract

- the angle α of the spatially resampled strokes (coded as $\sin\alpha$ and $\cos\alpha$),
- the difference of consecutive angles ($\sin\Delta\alpha$ and $\cos\Delta\alpha$), and
- the pen's pressure during writing.

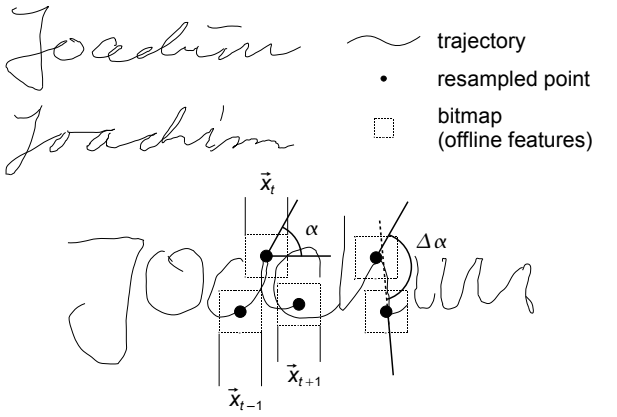

Figure 3. Three examples of the database as well as the sample points and the extracted features.

For some experiments we also derive *off-line features*, namely a 9-dimensional vector representing a subsampled bitmap slid along the pen's trajectory. This is done in order to incorporate a 30×30 pixel fraction of the actual image of the currently written letter.

In sum, five on-line and nine off-line features are used, yielding in a 14-dimensional feature vector

$$\vec{x}_t = (\underbrace{x_1, x_2, x_3, x_4, x_5}_{\text{on-line features}}, \underbrace{x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}}_{\text{off-line features}}) \quad (10)$$

for each spatial sample point t. Figure 3 shows typical sample inputs from the tableau: Three different representations of a name, as well as the extracted on-line features (α and $\Delta\alpha$) and the sliding window for the off-line features.

As mentioned before, NNs are capable of involving temporal context into recognition: To include time context into the feature vector, the input vector is enhanced by several future and past frames. Thus the final observation vector yields

$$\vec{s}_t = \left\{ \vec{x}_{t-N_p}\, \vec{x}_{t-N_p+1}, \ldots, \vec{x}_t, \ldots, \vec{x}_{t+N_f} \right\} \quad (11)$$

with N_p the number of past and N_f the number of future frames. In our system we typically use between four and nine past and future frames. The number of input nodes for the NN is then computed by

$$M = (N_p + N_f + 1) \cdot |\vec{x}| \quad (12)$$

where $|\vec{x}|$ denotes the number of features. This would not be necessary for a pure HMM system, but helps to improve the recognition rate of the NNs significantly.

4. Experiments

For training and evaluation purpose, a handwritten database of German letters and words was used. In sum it contains 64 different symbols, consisting of the 59 letters of the German alphabet and 5 numbers. It was recorded by 21 writers (left handed and right handed, both male and female) and contains over 6000 words. We used 75 % of the words in the database for training and the remaining 25 % for evaluation purposes. All experiments were performed in writer independent mode. The database has already been used in various preceding works [17; 18].

Our novel methods are compared to both a standard continuous and a discrete HMM baseline system. The baseline HMMs consist of 12 states – each HMM represents one symbol. In the discrete case, vector quantization is applied to derive discrete observations. The code book is generated by a k-means algorithm. In all experiments a 2k dictionary is used for final word recognition.

To show the impact of the additional off-line features all tests have been accomplished on both the full feature vector (Eq. 10) and a reduced feature vector consisting only of the on-line features. MLPs and RNNs are known to show different performance due to the time varying sequences. Therefore the TP and the tandem approach have been tested on both NN types. To train the NNs, the training data was aligned to the transcriptions with a Viterbi alignment using a standard HMM system. Afterwards, the NNs and HMMs were trained using the algorithms as explained in section 2. Temporal context is taken into account by enhancing the input vector by either $N_p = 3$ and $N_f = 4$ or $N_p = 9$ and $N_f = 9$ frames. This results in $M_1 = 40$ and $M_2 = 95$ input nodes for using only the on-line, and $M_3 = 266$ using both, on- and off-line features (refer to Eq. 12).

Table 1 shows the recognition results for both the continuous and the discrete baseline HMMs, as well as the proposed TP and tandem approach, both using either a MLP NN or a RNN. All results are shown for both on- and off-line features, as well as only on-line features. For the hybrid methods Table 1 also shows different exemplary configurations of past and future frames N_f and N_p for the neural net.

Table 1. Recognition rates for the baseline and the proposed hybrid NN/HMM systems.

Method		**Features**		
		on-line		*on- and off-line*
Base	*cont.*	83.3 %		84.2 %
	disc.	90.2 %		91.3 %
N_f, N_p		3, 4	9, 9	9, 9
TP	*RNN*	82.5 %	88.7 %	93.3 %
	MLP	83.4 %	89.2 %	94.1 %
tandem	*RNN*	87.2 %	92.4 %	95.2 %
	MLP	87.8 %	92.6 %	**95.9** %

First consider only the baseline system: for both the continuous and the discrete HMM, the enhancement of the feature vector with off-line features has only a very limited effect (around 1% absolute) on the baseline system's recognition rate. This matches the results in [19]. Furthermore it can be seen, that the discrete system outper-

forms the continuous HMM for both feature sets. Again this agrees with the findings in [19]. The best baseline recognition performance of 91.3% is reached with the discrete HMM and on- and off-line features.

Now consider the novel TP and tandem approach. It can be seen that these approaches depend on both the feature set and the number of past and future frames for the NN. For all four model combinations of the TP and the tandem approach the choice of features influences the recognition rate by at least 8% absolute recognition rate between the best and the worst feature configuration for the same model type. Thus all four hybrid models highly benefit from adding off-line features, as well as past and future frames. This can be explained by interpreting the additional information as supervised input nodes which then leads to better NNs. Thus, in contrast to standard HMMs, the hybrid approaches benefit highly from enhancing the feature vector.

The proposed hybrid models with the right choice of features generally perform significantly better compared to the baseline systems. With on- and off-line features all four model combinations are at least 2 % better than the best baseline HMM model. The best result of 95.9 % recognition rate is achieved with the tandem approach: In the configuration with a MLP NN, 14 features, and nine past and future frames this model increases the absolute recognition performance by 4.6 % (52.9 % rel. error reduction) compared to the best baseline HMM. One reason for this significant improvement in recognition rate is the additional discrimination added to the features by the NNs, which leads to more selective HMMs.

5. Conclusion

In this work we proposed two hybrid NN/HMM methods for handwriting recognition. We first introduced the tied posterior approach, where the output probability function of an HMM is approximated with a neural net. This adds neural network discrimination to the advantages of HMMs. The second hybrid model – the tandem – uses a neural net as part of the feature extraction. Thus all characteristics of HMMs are kept, but a significant additional feature discrimination is added to the overall system.

A disadvantage of the proposed hybrid systems is the amount of training data required for the neural nets, which can be higher than for pure HMM systems. Furthermore both the TP and the tandem approach have the drawback of a higher decoding complexity by applying a neural net classification additionally to the Viterbi algorithm.

However in an experimental section we showed that the significant gain in recognition rate is worth this effort: Compared to the best baseline HMM system, the absolute recognition rate was improved by 4.6 % (52.9 % rel. error reduction) with the proposed hybrid tandem approach.

Furthermore in contrast to the baseline HMMs, both hybrid approaches benefit highly from adding additional off-line features. In future we therefore plan to further extend this approach by adding more features for the NN training and modifications of the HMM states in the case of the tandem approach.

References

[1] L.R. Rabiner, "A tutorial on Hidden Markov Models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–285, February 1989.

[2] K. Takahashi, H. Yasuda, and T. Matsumoto, "A fast hmm algorithm for on-line handwritten character recognition," *4th ICDAR97, Proc.*, vol. 1, pp. 369–375, 1997.

[3] H. Yasuda, T-Talahasjo, and T. Matsumoto, "A discrete hmm for online handwriting recognition," *Int. J. of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 5, pp. 675–688, 2000.

[4] R. Koehle and T. Matsumoto, "Pruning algorithms for hmm on-line handwritten recognition," *Technical Reprot of IEICE, PRMU97-5*, pp. 33–39, 1997.

[5] L.E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Annals of Mathematical Statistics*, vol. 37, 1966.

[6] H. Bourland and N. Morgan, "Connectionist speech recognition: A hybrid approach," *Kluwer Academic Publishers*, 1994.

[7] A. J. Robinson, "An application of recurrent nets to phone probability estimation," *IEEE Transactions on Neural Networks*, vol. 5, pp. 298–305, 1994.

[8] J. Rottland and Gerhard Rigoll, "Tied posteriors: an approach for effective introduction of context dependency in hybrid nn/hmm lvcsr," *ICASSP*, vol. 3, pp. 1241–1244, 2000.

[9] Hynek Hermansky, Daniel P. W. Ellis, and Sangita Sharma, "Tandem connectionist feature extraction for conventional hmm systems," *ICASSP*, 2000.

[10] Christoph Neukirchen and G. Rigoll, "A new approach to hybrid hmm/ann speech recognition using mutual information neural networks," *NIPS*, pp. 772–778, 1996.

[11] X. D. Huang and M. A. Jack, "Semi-continuous hidden markov models for speech signals," *Computer Speech and Language*, vol. 3, pp. 1759–1762, 1989.

[12] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–267, 1967.

[13] R. J. Schalkoff, "Artificial neural networks," *McGraw-Hill*, 1994.

[14] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," *IEEE Int. Conference on Neural Networks*, 1993.

[15] C. Igel and M. Hüsken, "Improving the rprop learning algorithm," *Proc. of the Second Int. Symposium on Neural Computation NC'2000*, pp. 115–121, 2000.

[16] Sunil Sivadas and Hynel Hermansky, "Hierachical tandem feature extraction," *ICASSP*, 2002.

[17] G. Rigoll, A. Kosmala, and D. Willett, "An investigation of context-dependent and hybrid modeling techniques for very large vocabulary on-line cursive handwriting recognition," *IWFHR98*, pp. 429–438, 1998.

[18] G. Rigoll, A. Kosmala, J. Rottland, and Ch. Neukrichen, "A comparison between continous and discrete density hidden markov models for cursive handwriting recognition," *Proc. of ICPR '96*, pp. 205–209, 1996.

[19] A. Brakensiek, A. Kosmala, D. Willet, W. Wang, and G. Rigoll, "Performance evaluation of a new hybrid modeling technique for handwriting recognition using identical on-line and off-line data," *5th Int. Conference on Document Analysis and Recognition*, pp. 446–449, 1999.