Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

# Tandem hidden Markov models using deep belief networks for offline handwriting recognition*

Partha Pratim ROY‡[1], Guoqiang ZHONG[2], Mohamed CHERIET[3]

([1]*Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, Roorkee 247667, India*)

([2]*Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China*)

([3]*Synchromedia Laboratory, École de Technologie Supérieure, Montreal H3C 1K3, Canada*)

E-mail: proy.fcs@iitr.ac.in; gqzhong@ouc.edu.cn; mohamed.cheriet@etsmtl.ca

Received Feb. 15, 2016; Revision accepted June 24, 2016; Crosschecked June 16, 2017

**Abstract:** Unconstrained offline handwriting recognition is a challenging task in the areas of document analysis and pattern recognition. In recent years, to sufficiently exploit the supervisory information hidden in document images, much effort has been made to integrate multi-layer perceptrons (MLPs) in either a hybrid or a tandem fashion into hidden Markov models (HMMs). However, due to the weak learnability of MLPs, the learnt features are not necessarily optimal for subsequent recognition tasks. In this paper, we propose a deep architecture-based tandem approach for unconstrained offline handwriting recognition. In the proposed model, deep belief networks are adopted to learn the compact representations of sequential data, while HMMs are applied for (sub-)word recognition. We evaluate the proposed model on two publicly available datasets, i.e., RIMES and IFN/ENIT, which are based on Latin and Arabic languages respectively, and one dataset collected by ourselves called Devanagari (an Indian script). Extensive experiments show the advantage of the proposed model, especially over the MLP-HMMs tandem approaches.

**Key words:** Handwriting recognition; Hidden Markov models; Deep learning; Deep belief networks; Tandem approach

http://dx.doi.org/10.1631/FITEE.1600996 **CLC number:** TP391

## 1 Introduction

Automatic recognition of handwritten text is a challenging task because of the large variability of writing styles and cursive nature (Senior and Robinson, 1998; El-Yacoubi *et al.*, 1999; Vinciarelli, 2002; Bunke, 2003; Fujisawa, 2008). In the past decades, stochastic approaches, such as hidden Markov models (HMMs), have been widely applied to perform text recognition tasks (Marti and Bunke, 2001; Vinciarelli *et al.*, 2004; Zimmermann *et al.*, 2006; Kessentini *et al.*, 2008; Mohamad *et al.*, 2009). HMMs are effective for modeling unconstrained text-strings. This is mostly due to their ability to cope with non-linear distortions and incomplete information. HMMs perform joint segmentation and recognition, which is useful for avoiding segmentation of cursive words into characters/sub-words (Vinciarelli *et al.*, 2004). These models can deal with observation sequences with variable lengths obtained from text images.

In the general HMM-based approaches, the hidden state sequence is approximated using a first-order Markov chain, where each state $S_t$ at time $t$ depends only on state $S_{t-1}$ at time $t-1$. Given a state sequence, the observations at different time steps are assumed to be conditionally independent (Rabiner, 1989). In practice, HMM-based systems

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61403353)

ORCID: Guoqiang ZHONG, http://orcid.org/0000-0002-2952-6642

are employed under two strategies: holistic or analytical. The holistic process considers word images as a whole and does not segment words into characters or sub-word units. In contrast, analytical approaches model words by concatenation of character/sub-word HMMs. Such approaches are convenient for large vocabulary because unknown words can be modeled by character concatenation.

In the literature, features in HMMs follow a sliding window approach: a fixed-width window shifts column by column from left to right or right to left depending on the writing style (in Arabic, the text is written from right to left). At each position of the window, a feature vector is extracted (Marti and Bunke, 2001; Vinciarelli *et al.*, 2004; Mohamad *et al.*, 2009) and the sequence of feature vectors obtained in this fashion is modeled with HMMs. The performance of HMM-based recognition depends greatly on the discriminative power of the features. Thus, feature extraction has long been a focus of research. In the predominant HMMs paradigm, the observation likelihood is computed from a Gaussian mixture model (GMM). Next, the Viterbi decoding searches the subsequence of an observation that matches best to a given HMM. The GMM-HMM framework presents a generative model, in which training and decoding operate under the maximum likelihood (ML) criterion. Several approaches have introduced discriminative techniques in training: model parameter estimation based on maximum mutual information (MMI) and minimum phone error (MPE) (Dreuw *et al.*, 2009; 2011b). Recently, recurrent neural networks (RNNs) have been shown to perform better than HMMs for handwriting recognition (Graves and Schmidhuber, 2008; Graves *et al.*, 2009). This is because RNNs are discriminative models whereas standard HMMs are generative.

Many researchers have attempted to combine HMMs with other models to build hybrid architectures to improve performance over the GMM-HMM approach. One common approach used at the feature level is to use the tandem method (Schenk and Rigoll, 2006), in which features are discriminatively trained via multi-layer perceptrons (MLPs). The outputs of the MLPs, which can be considered as character/state posterior probabilities, can be either combined at the frame level, or appended to feature vectors. The power of the tandem method stems from non-linear mapping, which is able to maximize the separability between classes in the output space.

In general, the hybrid models use MLPs instead of GMMs as HMMs' state posterior estimator, whereas tandem models take into account both GMMs and MLPs as features. Although MLP-HMM-based approaches have been used successfully in a number of systems (Espana-Boquera *et al.*, 2011), their gradient-based training easily gets stuck in apparent local minima or plateaus (starting from random initialization). As more hidden layers are added in the MLPs, it becomes more difficult to obtain good generalization. Recently, new research on the training strategies of deep belief networks (DBNs) (Hinton *et al.*, 2006) has allowed improved performance in many machine learning and pattern recognition tasks. This deep learning approach has been proved effective in a number of applications, including isolated handwritten character recognition (Hinton, 2002; Thomas *et al.*, 2015), speech recognition (Dahl *et al.*, 2011; Mohamed *et al.*, 2012; Senior *et al.*, 2014), and machine transliteration (Deselaers *et al.*, 2009). Deep networks learn a hierarchy of non-linear feature detectors that can capture complex statistical patterns in data. These advances motivate us in developing deep learning techniques for sequential text recognition.

DBNs use a greedy layer-by-layer pre-training algorithm to initialize the network weights. Because of this advantage, DBNs show significant performance gains over conventional MLPs (Hinton *et al.*, 2006). Nevertheless, until now, there exists no work toward word recognition tasks using a DBN-based HMM system. In this study, we adopt DBNs to extract discriminative features from unconstrained handwritten text images and use these features in a tandem HMM approach. We demonstrate that the existing HMM-based text recognition framework can be further improved by the combined DBN and HMM tandem approach. The primary contributions of this study are the following: (1) exploration of a DBN-HMM tandem model; (2) application of DBNs in unconstrained text recognition problems; (3) a comparative analysis of our results with HMMs and the MLP-HMM tandem system.

## 2 Related work

As discussed earlier, the generative training to optimize a GMM-HMM system can be improved

if the training is tuned discriminatively (Bertolami and Bunke, 2008). On the other hand, because of their discriminative nature, artificial neural networks (ANNs) have been extensively applied to classify characters as part of isolated or continuous handwritten word recognizers (Marinai *et al.*, 2005). Thus, the combination of ANNs and HMMs, as an alternative paradigm to GMM-HMM, has been gaining popularity in many applications (Schenk and Rigoll, 2006; Espana-Boquera *et al.*, 2011). In such systems, neural network (NN) based posterior probabilities may be used to directly compute HMM observation probabilities (e.g., the hybrid approach (Bourlard and Morgan, 1994; Renals *et al.*, 1994; Kozielski *et al.*, 2013)) or for feature extraction (e.g., the tandem approach (Schenk and Rigoll, 2006)). NNs have the advantages of being inherently discriminative to optimize state probabilities.

Recently, a hybrid ANN-HMM system was proposed for text recognition taking advantage of the MLP model and context sensitivity (Espana-Boquera *et al.*, 2011). The hybrid MLP-HMM model was used to model grapheme, while a single NN was used to estimate the emission probabilities. The estimates of the posterior probabilities computed by the NN were divided by the prior state probabilities, resulting in the scaled likelihood, which was used as the emission probability in HMMs.

Tandem modeling (Hermansky *et al.*, 2000; Schenk and Rigoll, 2006) was proposed to combine the discriminative parameter estimation of the ANNs with the sequence modeling ability of the HMMs. The positive effect of the combined feature is that the MLPs perform a non-linear feature transformation into a space which is explicitly oriented for discriminability of characters/states. The transformed feature leads to improved discrimination by the GMM, which describes the output space associated with each HMM state. The advantage of the tandem approach is its robustness to noise (Sharma *et al.*, 2000). The main difference between the tandem and hybrid approaches is that the latter uses the output of MLPs to approximate the probability density function (PDF) of an HMM state. In contrast, the tandem approach uses the standard GMM.

Based on the development of deep learning techniques, in this study, we propose a deep network based tandem model for unconstrained offline handwritten text recognition. DBNs are used to learn compact representations of data and combined with the subsequent HMMs. Although DBNs are introduced as a powerful algorithm and have been explored in many applications, such as handwritten digit recognition and speech recognition, there have rarely been approaches that combine DBNs and HMMs. In particular, we notice that in Mohamed *et al.* (2009), DBNs were employed to classify subphones and then combined with an HMM bi-gram language model for speech recognition. Although the model in Mohamed *et al.* (2009) shares the same building blocks, namely DBNs and HMMs, with our model, it is distinct from our work in both the details of the algorithms and the target of applications. For concreteness, the model in Mohamed *et al.* (2009) uses DBNs for subphone classification, while our model uses them for data representation learning; the model in Mohamed *et al.* (2009) was applied to speech recognition problems, while our model is used for handwritten text recognition tasks. To the best of our knowledge, this work is the first to apply the tandem model of DBNs and HMMs to handwritten word recognition tasks.

## 3 The proposed model

In this section, we present how DBNs can be employed in a tandem framework and analyze how the deep architecture of DBNs helps in handwritten text recognition. The process of producing tandem features is sketched in Fig. 1. The column-wise feature vectors, extracted from the word image, are fed into DBNs. Rather than interpreting the outputs as character class posteriors as in a hybrid ANN-HMM, they are treated as observations in HMMs. The posterior features are subjected to a transformation using dimensionality reduction before processing. Finally, these DBN features are appended to original features in an HMM system with GMM observation distributions.

### 3.1 Training of deep belief networks

DBNs are unsupervised generative models introduced by Hinton *et al.* (2006). In DBNs, the greedy layer-by-layer training is used for efficiently learning a deep probabilistic model from a complex data structure. The learning algorithm first initializes the weights of each layer individually in an unsupervised way and then fine-tunes the entire network

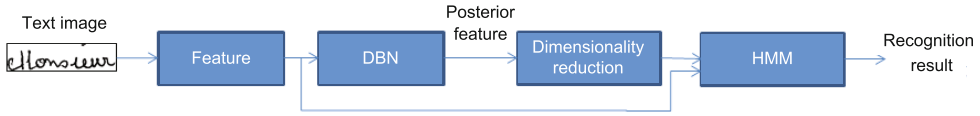


**Fig. 1 Block diagram of the DBN-HMM tandem system**

using labeled data. A DBN is created as a stack of its main building blocks called restricted Boltzmann machines (RBMs). A detailed technical report can be found in Hinton *et al.* (2006). For concreteness, we provide a brief description of RBMs and DBNs in the following.

An RBM is a particular form of a log-linear Markov random field that has a two-layer architecture, in which the visible stochastic units $v$ are connected to the hidden stochastic units $h$. Generally, all visible units are connected to all hidden units, and there are no visible-visible and hidden-hidden connections. In the simplest form of RBMs, both the hidden and visible units are binary and stochastic. Each layer of the latent representation is learned by training an RBM to model the data distribution at the next lower layer, using contrastive divergence (CD) (Hinton, 2002). Given the model parameter $\theta$, the weights of the connections and the biases of the individual units form a joint probability distribution $P(v,\ h|\theta)$ over the visible units $v$ and hidden units $h$. For binary RBMs, this distribution is computed based on the following energy function:

$$E(v,h|\theta)=-\sum_{i=1}^{V}\sum_{j=1}^{H}w_{ij}v_ih_j-\sum_{i=1}^{V}b_iv_i-\sum_{j=1}^{H}a_jh_j, \tag{1}$$

where $\theta=\{w,b,a\}$ are model parameters, $w_{ij}$ is the weight between visible unit $i$ and hidden unit $j$, $b_i$ and $a_j$ are bias terms for visible unit $i$ and hidden unit $j$ respectively, and $V$ and $H$ are the numbers of visible and hidden units respectively. The marginal probability is computed as follows:

$$P(v|\theta)=\frac{\sum\limits_{h}\exp(-E(v,h|\theta))}{Z(\theta)}, \tag{2}$$

where $Z(\theta)$ is known as the partition function, given by

$$Z(\theta)=\sum_{v}\sum_{h}\exp(-E(v,h|\theta)). \tag{3}$$

Since there are no hidden-hidden or visible-visible connections in RBMs, all visible units (hidden units) become independent, given the hidden units (visible units). The conditional distributions $P(v|h)$ and $P(h|v)$ are factorials and can be effectively derived as follows:

$$P(h_j=1|v,\theta)=\sigma\left(\sum_{i=1}^{V}w_{ij}v_i+a_j\right), \tag{4}$$

$$P(v_i=1|h,\theta)=\sigma\left(\sum_{j=1}^{H}w_{ij}h_j+b_i\right), \tag{5}$$

where $\sigma(x)=(1+\mathrm{e}^{-x})^{-1}$ is the sigmoidal function.

Once an RBM is trained, the data can be represented by the RBM. For each data vector $\boldsymbol{v}$, we use Eq. (4) to compute a vector of hidden unit activation probabilities $\boldsymbol{h}$. These hidden activation probabilities are used as training data for a new RBM. Thus, each set of RBM weights can be used to extract features from the output of the previous layer. After the training of RBMs is done, we initialize the weights of the hidden layers of a neural net with the hidden layers of the trained RBMs. After pre-training, we add a randomly initialized softmax output layer and use backpropagation to fine-tune all the weights in the network discriminatively.

### 3.2 DBN-HMM tandem approach

In this study, we present the DBN-HMM tandem systems, as illustrated in Fig. 1. DBN-HMM combines the discriminative parameter estimation of the DBNs with the sequence modeling ability of the HMM. For this purpose, DBNs are integrated into the HMM framework to form the DBN-HMM tandem system. In the HMM system, the probability $P(X|\lambda)$ of the observation sequence is $X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_T\}$, provided that the HMM's $\lambda$ is given. This probability can be computed by adding the probabilities of all possible state sequences $S^k=\left\{s_1^k,s_2^k,\ldots,s_T^k\right\}$:

$$P(X|\lambda)=\sum_{S^k}P(S^k|\lambda)P(X|S^k,\lambda). \tag{6}$$

By using the HMMs' properties, the observation frame depends only on the state which generates

it, and the HMMs used in our word recognition approach are first-order Markov chains. Therefore, Eq. (6) can be derived as follows:

$$P(X|\lambda) = \sum_{S^k} \left(P(s_1^k)P(s_2^k|s_1^k)\cdots P(s_T^k|s_{T-1}^k)\right) \cdot \left(P(\boldsymbol{x}_1|s_1^k)P(\boldsymbol{x}_2|s_2^k)\cdots P(\boldsymbol{x}_T|s_T^k)\right). \tag{7}$$

Eq. (7) can be re-expressed as follows:

$$P(X|\lambda) = \sum_{S^k} P(s_1^k)P(\boldsymbol{x}_1|s_1^k)P(s_2^k|s_1^k)P(\boldsymbol{x}_2|s_2^k) \cdots P(s_T^k|s_{T-1}^k)P(\boldsymbol{x}_T|s_T^k), \tag{8}$$

where $P(s_1^k)$ is the initial probability of state $s_1^k$, $P(s_t^k|s_{t-1}^k)$ the transition probability from state $s_{t-1}^k$ to state $s_t^k$, and $P(\boldsymbol{x}_t|s_t^k)$ the emission probability (posterior) of feature vector $\boldsymbol{x}_t$ given state $s_t^k$.

In contrast to the likelihood of a feature vector $\boldsymbol{x}_t$ given an arbitrary state $s_i$, DBNs produce state posterior probability $P(s_i|\boldsymbol{x}_t)$. Training the DBN requires each observation at time step $t$ in the training data to be aligned to a character label of its transcription. However, the class (e.g., HMM-state) labels are usually not available. To obtain this labeling, a previously trained GMM-HMM is applied to the training data in the forced alignment mode (Espana-Boquera *et al.*, 2011). Fig. 2 shows an illustrative sketch. Then the DBN is trained on the labeled observations. The DBN is trained in a frame-based approach with a softmax output layer. The trained DBN is used to calculate a posterior distribution over the character labels for each observation. In our framework, the posterior probabilities are decorrelated by a dimensionality reduction algorithm using the Karhunen-Loéve transform (KLT) (Kittler and Young, 1973). In a tandem HMM approach, the posterior estimates are considered as observations to train a new HMM (GMM-HMM) in order to perform sequence modeling.

## 4 Experimental evaluation

We carried out experiments on word recognition tasks for three different scripts, namely, Latin (RIMES dataset), Arabic (IFN/ENIT dataset), and Indian (Devanagari dataset) scripts.

The RIMES dataset (*Reconnaissance et Indexation de données Manuscrites et de fac similÉS*/Recognition and indexing of handwritten documents and faxes) (Augustin *et al.*, 2006) has been

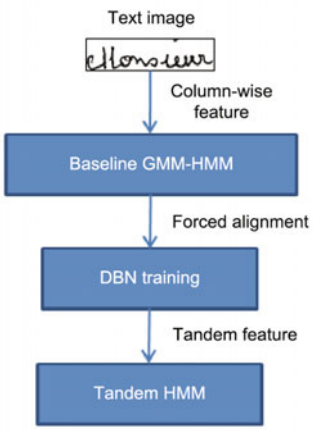


**Fig. 2 Workflow of the DBN-HMM tandem approach used for word recognition**

used in ICFHR and ICDAR competitions to evaluate unconstrained handwriting recognition systems. The dataset contains postal mail or fax by individuals to companies or administrations. It consists of 59 203 word images divided into three subsets: 44 197 images for training, 7542 for validation, and 7464 for testing. In this study, we considered only a reduced test dictionary, the size of which is 1612 words. A few examples from the test dataset are shown in Fig. 3a. Note that it is very challenging to recognize some of these words.

The IFN/ENIT database (Margner and El Abed, 2010) contains 32 492 Arabic handwritten Tunisian town names for recognition of Arabic handwritten words. The words were written by about 1000 writers with a vocabulary size of 937. The database is divided into five subsets, *A*, *B*, *C*, *D*, and *E*. In the presented experiments, sets *A*–*D* were used for training and set *E* was used for testing (see some example images in Fig. 3b).

Devanagari (Hindi) is the most widely used Indian script (Pal and Chaudhuri, 2004) and is used to write languages such as Sanskrit, Devanagari, and Nepali. It is seen that most of the characters have a horizontal line (Matra/Shirorekha) at the upper part. When two or more characters are written side by side, they form a word. A vowel following a consonant takes a modified shape and is placed at the left, right, both left and right, or bottom of the consonant. These modified shapes are called modified characters. These modifiers add extra difficulty in the character segmentation procedure of Devanagari scripts because of their topological positions.

There exists no standard dataset for Devana-

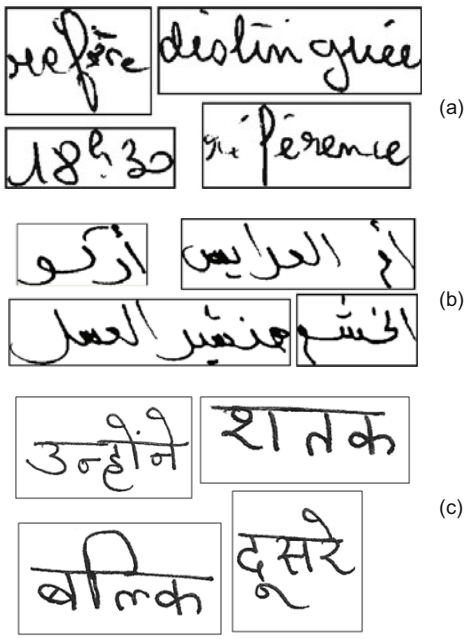

**Fig. 3 Some examples of word images from RIMES (a), IFN/ENIT (b), and Devanagari (c) datasets**

gari word recognition. We collected data from 50 native Hindi speakers. The Devanagari dataset contains a total of 16 128 handwritten word images, out of which 10 667 were used for training, 1872 for validation, and the rest 3589 for testing. These words were considered from 60 handwritten document images from individuals of different professions. The words are considered in a manner so that each character/modifier is uniformly distributed in all the words. We considered 1957 Devanagari words in the lexicon. Some examples are shown in Fig. 3c.

### 4.1 Experimental results

This subsection presents the experimental details of the conventional GMM-HMM baseline system and the DBN-HMM tandem system for word recognition. We also compare the MLP-HMM based framework with the same architecture as DBNs with respect to the number of layers, the number of hidden units per layer, and the activation function.

#### 4.1.1 Word recognition using GMM-HMM

In the basic HMM framework, a textword is modeled by the concatenation of its character models. All character models contain a fixed number of hidden states $\{S_1, S_2, ..., S_N\}$ arranged in Bakis topology without skips (Marti and Bunke, 2001). The word is represented as a sequence of feature vectors $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$, also known as a sequence of frames. The feature extraction process will be explained later. In HMMs, the likelihood of emitting a frame $\boldsymbol{x}_t$ in state $i$ is modeled using a GMM. We consider continuous density HMMs with diagonal covariance matrices of GMMs in each state. Thus, the observation probability density for each state is a mixture of Gaussian distributions. This mixture is obtained by increasing the number of Gaussian distributions in each state, step by step until a convenient HMM topology is obtained. The Baum-Welch algorithm (Baum *et al.*, 1970) is employed for optimizing the parameters of the model. Finally, the trained models are used to decode the test images by the Viterbi algorithm. We use the HMM toolkit (HTK) (Young *et al.*, 2006) for training and recognition.

1. Features for word recognition

Using sliding window techniques, the text image is represented by a sequence of $T$ local feature vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$. In the literature, there are a number of techniques proposed for sliding-window based features for HMMs (Marti and Bunke, 2001; Rodríguez and Perronnin, 2008). Different features are designed to encode the textual information to better describe the handwritten style and scripts. Since in this study we focus on DBN integration into an HMM based tandem approach, an off-the-shelf basic feature is used for the demonstration of DBNs' effectiveness. We employ the popular column-wise feature proposed by Marti and Bunke (2001) to represent the binary word images. Here, the sliding window has a width of one pixel, moving from left to right or right to left. The feature consists of a set of nine local features, including geometrical and contour-gradient information. Three global features capture the fraction of black pixels, the center of gravity, and the second-order moment. The remaining six local features consist of the positions of the upper and lower contours, the gradients of the upper and lower contours, the number of black-white transitions, and the fraction of black pixels between the upper and lower contours.

2. Dynamic feature

The performance of a text recognition system can be greatly enhanced by adding time derivatives to the basic static parameters (Bianne-Bernard *et al.*, 2011). The first- and second-order dynamic features are known as delta and acceleration coefficients. The

delta coefficients are computed using the following regression formula:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{2\sum_{\theta=1}^{\Theta} \theta^2}, \quad (9)$$

where $d_t$ is a delta coefficient at time $t$ computed in terms of the corresponding static coefficients $c_{t-\Theta}$ to $c_{t+\Theta}$. The value of $\Theta$ is set according to the window size. Similarly, delta coefficients are used to obtain acceleration coefficients. These derivative features capture a wider temporal context at the frame level and represent the dynamics of features around the current window. In our approach, we use the nine-dimensional Marti-Bunke (M-B) feature along with its dynamic feature (delta and acceleration in frame-wise feature), and obtain a feature vector of 27 dimensions.

3. Results using GMM-HMM

The raw data from all datasets were used for performance evaluation. No pre-processing was performed to clean the noise or de-slant the text in images. Since the images in the RIMES dataset are in gray tone, we used the Otsu binarization algorithm (Otsu, 1979) to convert them to binary. We optimized the baseline GMM-HMM system by tuning the parameters. Experiments were carried out by varying the number of states ($S \in \{7, 8, 9, 10\}$), and varying the number of mixture components per state. Fig. 4 illustrates the word recognition accuracy on these three datasets with different choices of Gaussian distributions. The numbers of Gaussian distributions were considered from 1 to 256 increasing in steps of the power of 2. It is observed that the recognition accuracy increased consistently when the model complexity increased from 1 to 256. The best accuracies obtained by the conventional GMM-HMM system were 75.27% (RIMES), 88.23% (IFN-ENIT), and 63.94% (Devanagari). We studied the effect of varying the number of states on the recognition performance. It is noted from the experiments that the best performances were obtained with nine states in the three datasets.

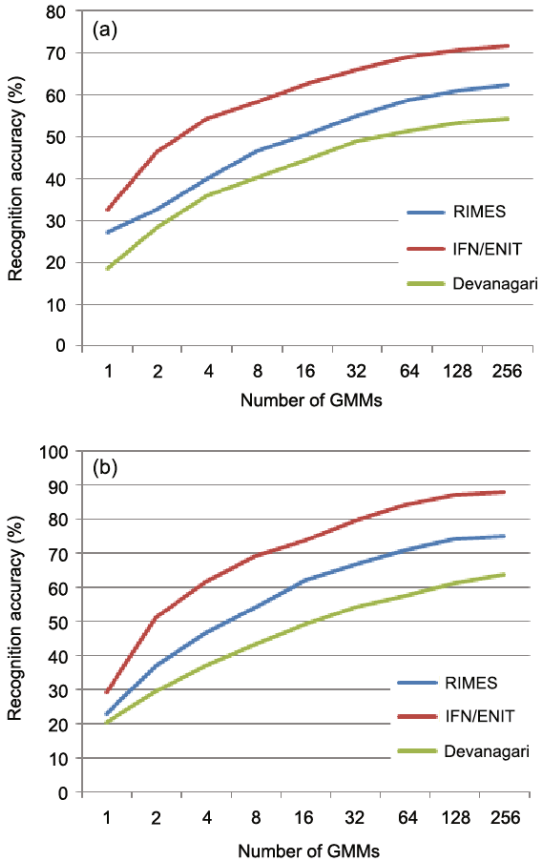


**Fig. 4 Recognition performance with (a) or without (b) context information of GMM-HMM on the RIMES, IFN/ENIT, and Devanagari datasets with different settings of Gaussian distributions**

4.1.2 Performance of DBN-HMM tandem system

With the parameter set in an HMM as explained earlier, we carried out experiments of a tandem HMM with a DBN architecture. Training a DBN requires each observation at time step $t$ in the training data be aligned to a character label of its transcription. For this purpose, the system uses Viterbi forced alignment in the baseline GMM-HMM. Next, DBNs were trained to model frame posterior probabilities using these aligned labels. The outputs of the DBN provided an estimate of the posterior probability distribution of the symbols used. The output posterior probabilities were decorrelated by a dimensionality reduction algorithm. We applied KLT (Kittler and Young, 1973) to the posterior probabilities of the DBNs. This is for reducing the dimensionality and orthogonalizing the feature vectors. Next, the features were normalized by their mean and variance. Finally, these reduced feature vectors were concatenated with the baseline HMM features obtained by dynamic features. Fig. 5 illustrates the performance while varying the dimension of the posterior feature on the three word recognition datasets. From the experiments, when the features were tested without dimensionality reduction, the performance was not good. It is noted that

the best accuracies were obtained with the dimension of 12 in IFN/ENIT and Devanagari and 8 in the RIMES dataset (Fig. 5). During this experiment, a single layer was used in the DBN, and the number of nodes in the hidden layer was kept at 1024.

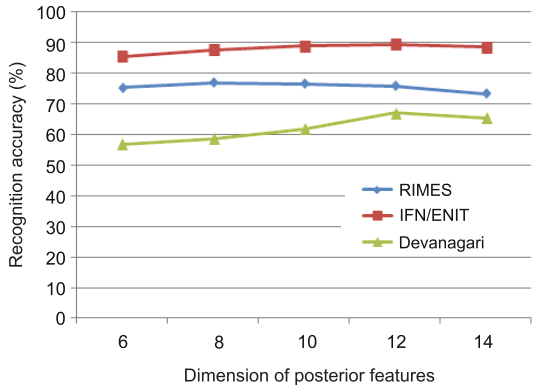


**Fig. 5 Recognition results obtained by DBN-HMM with varying size of posterior features**

All DBNs were pre-trained in an unsupervised manner using RBMs. In the pre-training process, the RBMs were trained using stochastic gradient decent with a mini-batch size of 100 training cases. The weights in RBMs were initialized randomly with a normal distribution with mean 0 and standard deviation 0.1. All RBMs were trained for 50 epochs. The hidden layers used logistic sigmoid non-linearities and a softmax layer was used for the output layer to provide posterior probability estimates for each output class. For fine-tuning, we used the same mini-batch size as in the pre-training step.

We studied the performance of DBNs for word recognition when the number of hidden units increased from 128 to 1536. The performance of the DBN improved with the increase in the number of nodes in the hidden layer. When the hidden layer size was large (more than 512), significant improvement over GMM-HMM was observed. With 1536 nodes in the hidden layer, the accuracies in RIMES, IFN/ENIT, and Devanagari datasets were 76.87%, 89.49%, and 66.84%, respectively. The gains using the DBN from the baseline GMM-HMM were 1.6% (RIMES), 1.26% (IFN/ENIT), and 2.49% (Devanagari). The performance results are detailed in Fig. 6. With more than 1536 hidden nodes the results did not improve. We also tested the effect of varying the number of hidden layers in the DBN. We performed experiments up to three layers for this purpose. During this experiment, the number of nodes in each hidden layer was kept fixed at 1024. With addition of a second layer, the performance improved in all the three datasets. When the number of layers was three, the performance did not improve on the IFN/ENIT dataset. Table 1 shows the performance accuracies according to the number of layers.

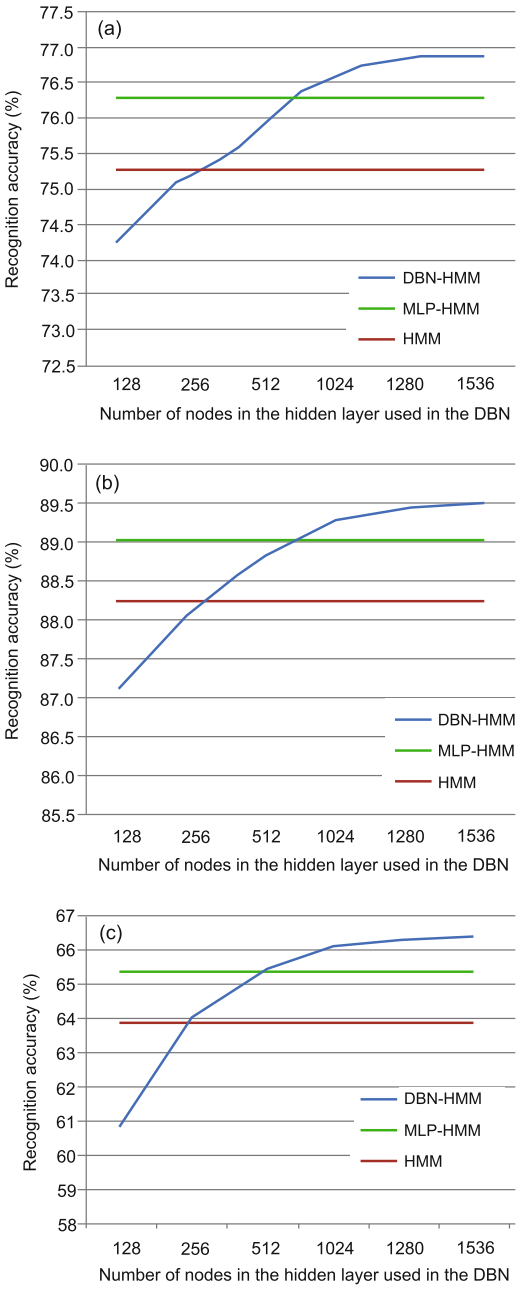


**Fig. 6 Comparison of recognition results on RIMES (a), IFN/ENIT (b), and Devanagari (c) datasets by DBN-HMM, MLP-HMM, and HMM**

**Table 1 Word recognition results on the datasets with different numbers of hidden layers**

| Number of hidden layers | Recognition accuracy (%) | | |
|---|---|---|---|
| | RIMES | IFN/ENIT | Devanagari |
| 1 | 76.74 | 89.29 | 66.08 |
| 2 | 76.92 | 89.46 | 66.46 |
| 3 | 76.98 | 89.41 | 66.84 |

#### 4.1.3 Comparison with the MLP-HMM tandem system

We compared the performance of our DBN-HMM-based tandem approach with the canonical MLP-HMM-based system. In this study, the common neural network called MLP (Haykin, 1998) was used. An MLP is composed of an input layer, an output layer, and a hidden layer. It allows solving problems that are not linearly separable. We used the standard back-propagation algorithm with weights initialized randomly to train the network.

The architecture of the MLP consisted of 27 input units, which corresponded to the M-B feature and its dynamic feature, a varying number of hidden units, and the output units equaled the total number of text symbols. DBNs are different from MLPs only in the unsupervised pre-training process since the weights of MLPs are initialized randomly. We tested MLP with a varying number of hidden units from 128 to 1536. It is observed that increasing the number of hidden units did not improve the word recognition performance. With less than 128 nodes, the recognition performance was worse. We obtained 76.28%, 89.04%, and 65.42% accuracies in RIMES, IFN/ENIT, and Devanagari datasets respectively with 128 nodes. The comparative results for the test sets of DBNs and MLPs are shown in Fig. 6. It is clear that when increasing the number of nodes in the hidden layer, DBN-HMM outperformed the MLP-HMM system. We also tested the MLP-HMM tandem approach by increasing the number of layers in MLP, but the performance did not improve.

### 4.2 Comparison with other systems

In the literature, a number of systems exist using HMM and RNN based recognizers (Graves and Schmidhuber, 2008; Graves *et al.*, 2009; Grosicki and El Abed, 2009; Margner and El Abed, 2010; Bianne-Bernard *et al.*, 2011; Dreuw *et al.*, 2011a). The TUM MDLSTM (Grosicki and El Abed, 2009) system achieved up to 93.2% accuracy in the RIMES dataset. Using MDLSTM in IFN/ENIT, an accuracy of 91.4% was obtained (Graves and Schmidhuber, 2008). One of the reasons for higher accuracy in these datasets is due to the discriminative nature in MDLSTM training. As a consequence, the combination of HMMs and RNNs can provide better performance. There exist some systems (Mohamad *et al.*, 2009) that combine more than one HMM. The comparison with such systems will also not be uniform as we use a single HMM system to evaluate the DBN-based approach. The primary objective of the present work is to evaluate the DBN-based tandem approach with respect to the MLP approach. Hence, a standard off-the-shelf feature was used to compare the performance.

We report some recent performances on Arabic and Latin handwriting datasets using HMM systems. Dreuw *et al.* (2011a) used an MLP-based feature from raw data in the tandem MLP-GMM system. They performed training using discriminative M-MMI criteria and obtained 92.70% accuracy in the IFN/ENIT dataset. In our system with the M-B feature, the accuracy was 89.04% using MLP-HMM in this dataset. With DBN, the accuracy was 89.46% using two layers with 1024 nodes in each layer. The improved performance in Dreuw *et al.* (2011a) might be due to MLP-based feature extraction and discriminative M-MMI training. For the RIMES dataset, Bianne-Bernard *et al.* (2011) reported the accuracy as 73.04% and 79.34% using context-independent and context-dependent systems, respectively. Preprocessing methods like de-slanting and baseline extraction were used in their system. The feature extraction was done based on Mohamad *et al.* (2009). In our approach, only binarization was used as the pre-processing step to use the M-B feature. We used a context-independent system in our approach and obtained 75.27% by GMM-HMM. After adding a DBN layer, we obtained an accuracy of up to 76.98% (corresponding to a 1.71% accuracy gain). To have a fair comparison, in the RIMES dataset we have included the context-independent results from Bianne-Bernard *et al.* (2011). These results are shown in Table 2.

There exists no standard dataset in the Devanagari word dataset. Shaw *et al.* (2014) developed an offline Devanagari word recognition system using a holistic approach. Since it was a holistic approach

**Table 2 Comparison of word recognition performances using the HMM framework**

| Dataset | Method | Recognition accuracy (%) |
|---|---|---|
| IFN/ENIT | Dreuw *et al.* (2011a) | 92.70 |
| | Proposed method | 89.46 |
| RIMES | Bianne-Bernard *et al.* (2011) | 73.04 |
| | Proposed method | 76.98 |

developed for a limited number of words, the performance cannot be compared. The authors reported 81.14% accuracy with a 100-class problem. Our framework of the character-based DBN-HMM tandem model achieved 66.84% accuracy without using a holistic recognition scheme. We conducted a statistical $t$-test by five-fold cross-validation and found that the recognition increment ($2.36\pm0.26$) is significant with the DBN-HMM-based tandem framework over the MLP-HMM-based system.

## 5 Conclusions

We have presented a robust tandem approach of HMMs using DBNs for handwritten word recognition. DBNs have been proved effective for a variety of machine learning problems due to their efficient learning of complex structure data. In this work, we combined the discriminative feature of DBNs with generative model HMMs in a tandem way. The DBN weights resulting from the unsupervised pre-training algorithm have been used to initialize the feed-forward neural network. The model learned during pre-training helped prevent overfitting and used better optimization of the recognition weights.

A comparative evaluation of DBNs and MLPs on three different word recognition datasets, namely RIMES (Latin), IFN/ENIT (Arabic), and Devanagari (Indian), was conducted in our experiments. The results on these datasets demonstrated that substantial gains in performance could be obtained by the use of tandem DBNs. Particularly, with the increase in the number of nodes in each layer, DBNs outperformed MLPs consistently. In our framework, we used a simple off-the-shelf feature to perform text recognition. The performance can be further improved by using sophisticated feature extraction approaches. Hence, existing systems (stand-alone or combination) can be further improved by the DBN-based tandem approach.

## References


Augustin, E., Carré, M., Grosicki, E., *et al.*, 2006. RIMES evaluation campaign for handwritten mail processing. Proc. Int. Workshop on Frontiers in Handwriting Recognition, p.231-235.

Baum, L.E., Petrie, T., Soules, G., *et al.*, 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Statist.*, **41**(1):164-171.

Bertolami, R., Bunke, H., 2008. Hidden Markov model-based ensemble methods for offline handwritten text line recognition. *Patt. Recog.*, **41**(11):3452-3460. http://dx.doi.org/10.1016/j.patcog.2008.04.003

Bianne-Bernard, A.L., Menasri, F., Mohamad, R.A.H., *et al.*, 2011. Dynamic and contextual information in HMM modeling for handwritten word recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, **33**(10):2066-2080. http://dx.doi.org/10.1109/TPAMI.2011.22

Bourlard, H.A., Morgan, N., 1994. Connectionist Speech Recognition: a Hybrid Approach. Springer US, USA.

Bunke, H., 2003. Recognition of cursive Roman handwriting: past, present and future. Proc. 7th Int. Conf. on Document Analysis and Recognition, p.448-459. http://dx.doi.org/10.1109/ICDAR.2003.1227707

Dahl, G., Yu, D., Deng, L., *et al.*, 2011. Large vocabulary continuous speech recognition with context-dependent DBN-HMMs. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, p.4688-4691.

Deselaers, T., Hasan, S., Bender, O., *et al.*, 2009. A deep learning approach to machine transliteration. Proc. 4th Workshop on Statistical Machine Translation, p.233-241.

Dreuw, P., Heigold, G., Ney, H., 2009. Confidence-based discriminative training for model adaptation in offline Arabic handwriting recognition. Proc. 10th Int. Conf. on Document Analysis and Recognition, p.596-600. http://dx.doi.org/10.1109/ICDAR.2009.116

Dreuw, P., Doetsch, P., Plahl, C., *et al.*, 2011a. Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained Gaussian HMM: a comparison for offline handwriting recognition. Proc. 18th Int. Conf. on Image Processing, p.3541-3544. http://dx.doi.org/10.1109/ICIP.2011.6116480

Dreuw, P., Heigold, G., Ney, H., 2011b. Confidence- and margin-based MMI/MPE discriminative training for offline handwriting recognition. *Int. J. Doc. Anal. Recog.*, **14**:273-288. http://dx.doi.org/10.1007/s10032-011-0160-x

El-Yacoubi, A., Gilloux, M., Sabourin, R., *et al.*, 1999. An HMM-based approach for off-line unconstrained handwritten word modeling and recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, **21**(8):752-760. http://dx.doi.org/10.1109/34.784288

Espana-Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., *et al.*, 2011. Improving offline handwritten text recognition with hybrid HMM/ANN models. *IEEE Trans. Patt. Anal. Mach. Intell.*, **33**(4):767-779. http://dx.doi.org/10.1109/TPAMI.2010.141

Fujisawa, H., 2008. Forty years of research in character and document recognition—an industrial perspective. *Patt. Recog.*, **41**:2435-2446. http://dx.doi.org/10.1016/j.patcog.2008.03.015

Graves, A., Schmidhuber, J., 2008. Offline handwriting recognition with multidimensional recurrent neural networks. Proc. 21st Int. Conf. on Neural Information Processing Systems, p.545-552.

Graves, A., Liwicki, M., Fernández, S., *et al.*, 2009. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, **31**(5):855-868. http://dx.doi.org/10.1109/TPAMI.2008.137

Grosicki, E., El Abed, H., 2009. ICDAR 2009 handwriting recognition competition. Proc. 10th Int. Conf. on Document Analysis and Recognition, p.1398-1402. http://dx.doi.org/10.1109/ICDAR.2009.184

Haykin, S., 1998. Neural Networks: a Comprehensive Foundation. Prentice Hall, USA.

Hermansky, H., Ellis, D.P.W., Sharma, S., 2000. Tandem connectionist feature extraction for conventional HMM systems. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, p.1-4. http://dx.doi.org/10.1109/ICASSP.2000.862024

Hinton, G.E., 2002. Training products of experts by minimizing contrastive divergence. *Neur. Comput.*, **14**(8):1771-1800. http://dx.doi.org/10.1162/089976602760128018

Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. *Neur. Comput.*, **18**(7):1527-1554. http://dx.doi.org/10.1162/neco.2006.18.7.1527

Kessentini, Y., Paquet, T., Benhamadou, A., 2008. A multi-stream HMM-based approach for off-line multi-script handwritten word recognition. Proc. Int. Conf. on Frontiers in Handwriting Recognition, p.1-6.

Kittler, J., Young, P.C., 1973. A new approach to feature selection based on the Karhunen-Loeve expansion. *Patt. Recog.*, **5**(4):335-352. http://dx.doi.org/10.1016/0031-3203(73)90025-3

Kozielski, M., Doetsch, P., Ney, H., 2013. Improvements in RWTH's system for off-line handwriting recognition. Proc. 12th Int. Conf. on Document Analysis and Recognition, p.935-939. http://dx.doi.org/10.1109/ICDAR.2013.190

Margner, V., El Abed, H., 2010. ICFHR 2010—Arabic handwriting recognition competition. Proc. Int. Conf. on Frontiers in Handwriting Recognition, p.709-714. http://dx.doi.org/10.1109/ICFHR.2010.115

Marinai, S., Gori, M., Soda, G., 2005. Artificial neural networks for document analysis and recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, **27**(1):23-35. http://dx.doi.org/10.1109/TPAMI.2005.4

Marti, U.V., Bunke, H., 2001. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. J. Patt. Recog. Artif. Intell.*, **15**(1):65-90. http://dx.doi.org/10.1142/S0218001401000848

Mohamad, R.A.H., Likforman-Sulem, L., Mokbel, C., 2009. Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. *IEEE Trans. Patt. Anal. Mach. Intell.*, **31**(7):1165-1177. http://dx.doi.org/10.1109/TPAMI.2008.136

Mohamed, A.R., Dahl, G., Hinton, G., 2009. Deep belief networks for phone recognition. Proc. NIPS Workshop on Deep Learning for Speech Recognition and Related Applications, p.1-9.

Mohamed, A.R., Dahl, G., Hinton, G., 2012. Acoustic modeling using deep belief networks. *IEEE Trans. Audio Speech Lang. Process.*, **20**(1):14-22. http://dx.doi.org/10.1109/TASL.2011.2109382

Otsu, N., 1979. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.*, **9**(1):62-66. http://dx.doi.org/10.1109/TSMC.1979.4310076

Pal, U., Chaudhuri, B.B., 2004. Indian script character recognition: a survey. *Patt. Recog.*, **37**(9):1887-1899. http://dx.doi.org/10.1016/j.patcog.2004.02.003

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**(2):257-286. http://dx.doi.org/10.1109/5.18626

Renals, S., Morgan, N., Bourlard, H., *et al.*, 1994. Connectionist probability estimators in HMM speech recognition. *IEEE Trans. Speech Audio Process.*, **2**(1):161-174. http://dx.doi.org/10.1109/89.260359

Rodríguez, J.A., Perronnin, F., 2008. Local gradient histogram features for word spotting in unconstrained handwritten documents. Proc. Int. Conf. on Frontiers in Handwriting Recognition, p.7-12.

Schenk, J., Rigoll, G., 2006. Novel hybrid NN/HMM modelling techniques for on-line handwriting recognition. Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition, p.1-5.

Senior, A., Robinson, A.J., 1998. An off-line cursive handwriting recognition system. *IEEE Trans. Patt. Anal. Mach. Intell.*, **20**(3):309-321. http://dx.doi.org/10.1109/34.667887

Senior, A., Heigold, G., Bacchiani, M., *et al.*, 2014. GMM-free DNN training. Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, p.1-5.

Sharma, S., Ellis, D., Kajarekar, S., *et al.*, 2000. Feature extraction using non-linear transformation for robust speech recognition on the Aurora database. Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, p.1117-1120. http://dx.doi.org/10.1109/ICASSP.2000.859160

Shaw, B., Bhattacharya, U., Parui, S.K., 2014. Combination of features for efficient recognition of offline handwritten Devanagari words. Proc. 14th Int. Conf. on Frontiers in Handwriting Recognition, p.240-245. http://dx.doi.org/10.1109/ICFHR.2014.48

Thomas, S., Chatelain, C., Heutte, L., *et al.*, 2015. A deep HMM model for multiple keywords spotting in handwritten documents. *Patt. Anal. Appl.*, **18**(4):1003-1015. http://dx.doi.org/10.1007/s10044-014-0433-3

Vinciarelli, A., 2002. A survey on off-line cursive word recognition. *Patt. Recog.*, **35**(7):1433-1446. http://dx.doi.org/10.1016/S0031-3203(01)00129-7

Vinciarelli, A., Bengio, S., Bunke, H., 2004. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *IEEE Trans. Patt. Anal. Mach. Intell.*, **26**(6):709-720. http://dx.doi.org/10.1109/TPAMI.2004.14

Young, S., Evermann, G., Gales, M.J.F., 2006. The HTK Book (Version 3.4). Engineering Department, Cambridge University, UK.

Zimmermann, M., Chappelier, J.C., Bunke, H., 2006. Offline grammar-based recognition of handwritten sentences. *IEEE Trans. Patt. Anal. Mach. Intell.*, **28**(5):818-821. http://dx.doi.org/10.1109/TPAMI.2006.103