Segmented Handwritten Text Recognition with Recurrent Neural Network Classifiers

Bolan Su[1], Xi Zhang[2], Shijian Lu[1] and Chew Lim Tan[2]
1. Institute for Infocomm Research, A*Star
1 Fusionopolis Way, 21-01 Connexis (south tower), Singapore 138632
Email: {subl, slu}@i2r.a-star.edu.sg
2. School of Computing, National University of Singapore
Computing 1, 13 Computing Drive, Singapore 117417
Email: {zhangxi, tancl}@comp.nus.edu.sg

***Abstract*—Recognition of handwritten text is a useful technique that can be applied in different applications, such as signature recognition, bank check recognition, etc. However, the off-line handwritten text recognition in an unconstrained situation is still a very challenging task due to the high complexity of text strokes and image background. This paper presents a novel segmented handwritten text recognition technique that ensembles recurrent neural network (RNN) classifiers. Two RNN models are first trained that take advantage of the widely used geometrical feature and the Histogram of Oriented Gradient (HOG) feature, respectively. Given a handwritten word image, the optimal recognition result is then obtained by integrating the two trained RNN models together with a lexicon. Experiments on public datasets show the superior performance of our proposed technique.**

I. Introduction

With the rapid development of computing devices, sensors, and storage facilities, more and more valuable documents including handwriting documents are digitalized and stored in databases for public access. However, recognition of unconstrained handwritten documents is always a challenging task and poor recognition results may lead to unreliable retrieval output and accordingly affect the accessibility of those valuable document information. At the same time, huge amount of important documents such as bank checks require handwriting inputs such as payee's names, money amount, and payer's signature. Accurate and robust handwriting recognition will help greatly to save the manpower and improve the productivity while handling these various types of documents with handwritten text.

Similar to the speech signal, handwritten text can often be viewed as a sequence of continuous signals. A number of techniques that succeeded in speech processing tasks have therefore been applied in the handwritten text recognition domain. Hidden Markov Models (HMMs), one of the most popular technique in speech recognition, has been successfully applied for handwritten text recognition. In particular, Wilfong et. al. proposed to use one HMM to represent one isolated handwritten word [1], though the proposed approach cannot be used for words which do not appear in the training data. Moreover, the approach cannot be scaled to large vocabularies, because a considerable amount of training data is required for each word and every distinct occurring word needs an HMM. To recognize arbitrary words, HMMs are used to represent character models instead of the whole words, and one word or text line are represented by a sequence of linearly connected HMMs [2]. Then the most likely character sequence can be obtained for a given text line by combining the trained HMMs model and the Viterbi algorithm.

However, the HMMs methods have a number of limitations. In particular, the probability of every observation depends only on the current state, which makes it difficult to incorporate the context information. More importantly, HMMs as a generative modle, may not provide better performance than discriminative models, because handwritten document recognition is essentially a discriminative task. Combining HMMs and neural networks has been proposed as a hybrid approach for handwriting recognition [3], [4], [5], [6], [7]. Different types of neural network architectures have been proposed, such as Multilayer Percetrons (MLP) [4], [5], time delay neural network [6], [3], and Recurrent Neural Networks (RNNs) [7], but most still suffer from the limitations of HMMs though they can capture the context information..

In the recent works, Recurrent Neural Network (RNN), with Connectionist Temporal Classification (CTC) output layer has been applied for the unconstrained handwritten document recognition [8], [9]. Compared with traditional RNNs, the RNNs with CTC output layer requires no pre-segmented input data, namely, the whole unsegmented sequence of the input data can be mapped to the output labels directly. Combined with a dictionary, the RNN plus CTC approach outperforms HMMs for both online and offline recognition tasks [8].

In this paper, we extend the RNNs approach by introducing a new set of Histogram of Oriented Gradient (HOG) features [10]. Two RNNs are trained on the HOG features and the traditional geometrical features separately. The classification outputs of the two models are then combined for optimal recognition results. The main contribution of our proposed technique can be summarized as below:

- First, we use the HOG column feature for better handwritten text recognition accuracy.
- Second, we propose an approach to ensemble recognition results of different networks for better performance.
- Third, we develop a handwritten word recognition system based on RNNs and achieve superior performance

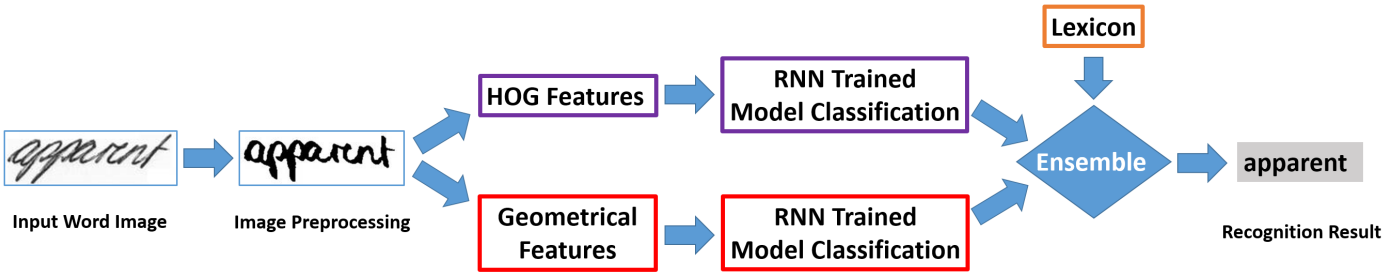

Fig. 1. The overall flowchart of our proposed handwritten word recognition system.

on real world challenging dataset.

II. Proposed Method

Fig 1 shows the overall flowchart of our proposed handwritten word recognition system. Given an input word image, a series of preprocessing is first applied including skew and slant correction [11] and normalization to reduce the text appearance variations. Image binarization [12] is then applied for geometrical feature extraction. After that, the normalized word image is converted into sequences of column feature based on HOG features [13] and geometrical feature [8]. Two multi-layer recurrent neural network (RNN) models with bidirectional Long Short-Term Memory (LSTM) [14] are then trained for the two sets of sequential features. After that, the score of each word in the lexicon is calculated separately for each RNN model by using the CTC [8] technique. Finally the scores of different RNN models are combined for optimal recognition result based on a predefined lexicon.

A. Geometrical Feature Extraction

A set of geometrical features are extracted from a binarized word image. In total, nine geometrical features are extracted from a sliding window, moving from left to right along each word image. These features are defined as follows [8], [9]:

1) the number of text pixels.
2) the center of gravity of the group of pixels.
3) the second order moment of the window.
4) the location of the upper-most text pixel.
5) the location of the lower-most text pixel.
6) the orientation of the upper-most text pixel.
7) the orientation of the lower-most text pixel.
8) the number of text-background transitions.
9) the number of text pixels divided by the number of all pixels between the upper- and lower-most text pixel.

B. HOG Feature Extraction

HOG feature is one of the most widely used features for object recognition in Computer Vision. It also performs well in text recognition due to its tolerance to the illumination variation and different types of geometric transformation. However, HOG features are extracted on image patches, whereas the RNN models require sequential signals as the input. We therefore adapt the convolutional HOG feature [10] to obtain a column feature for RNNs. In particular, the input image is first resized to the same height to obtain the column features with the same height. The normalized image is then partitioned into convolutional patches with step size 1, which is defined as follows:

$$Patch(i,j) = I(i : i+W, j : j+W) \tag{1}$$

where i, j denotes the pixel index of image I, W denotes the windows size. So the range of i, j could be $[1, M-W+1]$, $[1, N-W+1]$, respectively. M, N refer to the height and width of the input image I.

A HOG feature vector is thus extracted for each image patch $P(i,j)$. After that, the average pooling strategy is applied on the HOG feature vectors of the same column as follows:

$$HOG(i,j) = \sum_{p=i-T/2}^{i+T/2} (HOG(p,j))/T \tag{2}$$

where i, j refer to index, HOG denotes the extracted normalized HOG feature vector of corresponding patch, HOG_{avg} denotes the feature vector after averaging pooling, and T denotes the size of neighbouring window for average pooling. A column feature is finally determined by concatenating the averaged HOG feature vectors at the same column. The overall procedure is illustrated in Fig. 2.

C. Recurrent Neural Network

RNNs with CTC output layer is used as handwritten text classifier in our system. It has a number of advantageous characteristics compared with of the traditional neural network or HMMs. First, unlike the HMM that generates observations based only on the current hidden state, RNNs incorporates the context information including the historical states by using the LSTM structure [28] and therefore outperforms the HMM greatly. Second, unlike the traditional neural network, the bidirectional LSTM RNNs model [8] does not require explicit labelling of every single column vector of the input sequence. This is very important for handwritten text recognition because handwritten characters are often connected, where the explicit labelling is often an infeasible task. Third, the RNNs with CTC requires only word level annotated training data, which saves lots of efforts on handwritten data labeling.

The hidden nodes of a RNN are self-connected which also connect to nodes in later time steps. Then information of a

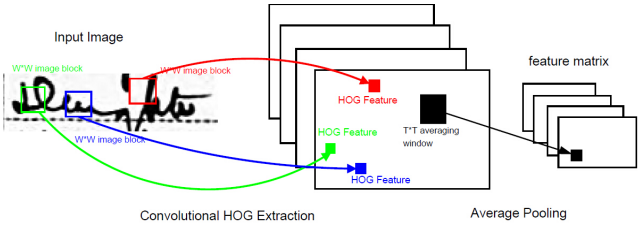

Fig. 2. The overall process of HOG feature extraction.

certain time ranges can therefore be captured and stored in the hidden node of current time step. In order to make full use of all input data, the hidden node is further connected to nodes in the past time steps with an additional backward hidden layer. The forward states are trying to capture the past information, while the backward states are making use of the future information. Such structure is called the Bidirectional Recurrent Neural Network (BRNN) that allows each time step to be evaluated based on both the past and future information.

The BRNN aims to predict the label of current time stamp with the contextual information of past time stamps. It is a powerful classification model. However, there is one severely drawback of the traditional RNN that the the error path integral decays exponentially along the sequence [15]. In order to overcome this limitation, the long short-term memory (LSTM) model was proposed [16] to introduce a new LSTM memory block to replace the hidden nodes of the traditional RNN, as shown in Fig. 3. One LSTM memory block with a single cell has three gates, which control the cell to access information over a long time period.

In our proposed system, the input of the RNNs model is a column vector, either nine geometric features or HOG column feature. Given an input column vector, the output of RNNs model is a probability distribution of assigning all possible labels to the column vector. The node with highest probability value can be set as the label of the input column vector.

Since an input word image consists of a sequence of column vectors $\mathbf{C}$. each column vector will be assigned a label l. The probability of an output label path π given a sequence of column vectors $\mathbf{C}$ is then defined as follows:

$$p(\pi|\mathbf{C}) = \prod_{t=1}^{L} p(\pi_t|\mathbf{C}) = \prod_{t=1}^{L} y_{\pi_t}^t \tag{3}$$

where L denotes the length of the output path and π_t denotes label of output path π at time t. The term y^t denotes the network output of RNN at time t. Therefore $y_{\pi_t}^t$ denotes the probability of π_t at time t.

On the other hand, the output path π corresponds to the target word $\mathcal{W}$ of the input word image. We define the mapping from π to $\mathcal{W}$ by removing all the repeating labels and empty labels in π. For example, an output path $('-','a','-','-','d','d')$ can be mapped to a word ad, where $'-'$ denotes the empty label. So we can further define the probability of an target word $\mathbf{W}$ given a sequence of column vectors $\mathbf{C}$ as follows:

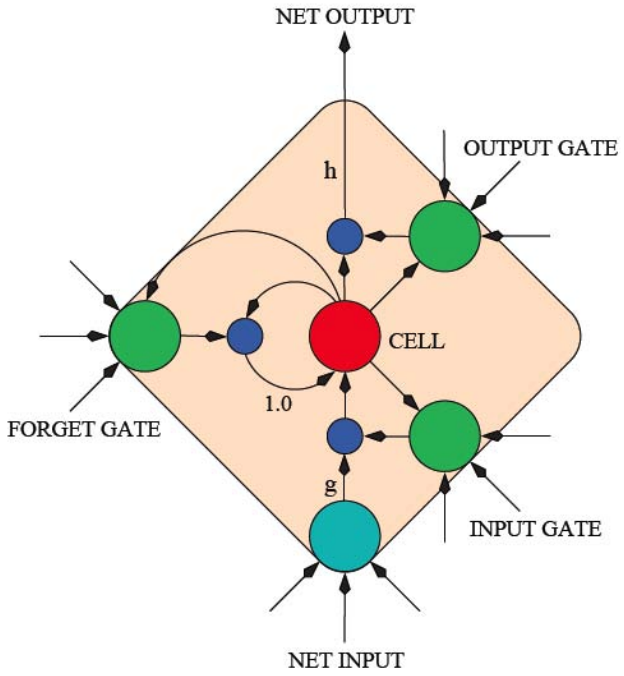

Fig. 3. Structure of LSTM memory block with a single cell [8]. There are three gates: input gate, output gate, and forget gate. These three gates collect the input from other parts of the network and control the information that the cell can accept. The input and output of the cell are controlled by the input gate and output gate, while how the recurrent connection effects the cell is controlled by the forget gate.

$$p(\mathcal{W}|\mathbf{C}) = \sum_{V(\pi)=\mathcal{W}} p(\pi|\mathbf{C}) \tag{4}$$

where V denotes the operator that translates the output path π to target word $\mathcal{W}$. It is worth to note that the translation process V is not unique.

So given a training set of handwritten images and corresponding words, the training process is to find out a network structure that can maximize the overall probability of $p(\mathcal{W}|\mathbf{C})$ in the training set. This can be done by back-propagating the gradient through the output layer.

D. Ensemble RNN models with Lexicon

Once the RNN model is trained, the output of a sequential feature vector can be viewed as a probability matrix. In particular, the RNN will produce a $K \times L$ probability matrix $\mathbf{Y}$, where K denotes the length of the sequence, and L denotes the number of possible output labels. Each entry of $\mathbf{Y}$ can be interpreted as the probability of a label given an input column vector. After that, with a probability matrix $\mathbf{Y}$ and a lexicon set $\mathcal{L}$ of all possible words, the word recognition can be formulated as searching for the best match word w^* with a highest probability.

We first calculate a score of each possible word as follows:

$$score_w = p(w|\mathbf{Y}) = \sum_{V(\pi)=w} p(\pi|\mathbf{Y}) \tag{5}$$

where $p(w|\mathbf{Y})$ is the conditional probability of word w given $\mathbf{Y}$. A direct graph can be constructed for the word w so that each node represents a possible label of w. In another word, we need to sum over all the possible paths that can form a word w on the probability matrix $\mathbf{Y}$ to calculate the score of a word w.

A new word w^i can be generated by adding some blank interval into the beginning and ending of w as well as the neighbouring labels of w, where the blank interval denotes the empty label. The length of w^i is $2*|w|+1$, where $|w|$ denotes the length of w. A new $|w^i| \times L$ probability matrix $\mathfrak{P}$ can thus be formed, where $|w^i|$ denotes the length of w^i and L denotes the length of the input sequence. $\mathfrak{P}(m,t)$ denotes the probability of label w^i_m at time t, which can be determined by the probability matrix $\mathbf{Y}$. Each path from $\mathfrak{P}(1,1)$ to $\mathfrak{P}(|w^i|,L)$ denotes a possible output π of word w, where the probability can be calculated using Eq. 3.

The problem thus changes to the score accumulation along all the possible paths in $\mathfrak{P}$. It can be solved with the CTC token pass algorithm [8] using dynamic programming. The computational complexity of this algorithm is $O(L \cdot |w^i|)$.

If we extracts several sets of features with different parameter settings, there will be more that one trained RNN models. Each RNN model will assign a score to every possible word in the lexicon $\mathcal{L}$. So we can combine the scores given by the two models to obtain the best match word w^* as follows:

$$w^* = \arg\max_{w \in \mathcal{L}} \sum_{i=1}^{n} (\alpha^i * score^i_w) \tag{6}$$

where $score^i_w$ denotes the assigned score to word w by the ith RNN model as defined in Eq. 5, α^i denotes the weight of each RNN models, which can be determined based its recognition accuracy on the training dataset.

III. Experiments and Discussions

The dataset we use in the experiments is taken from the ANWRESH-2014, the first Competition on Word Recognition from Segmented Historical Documents [17] held together with the 14th International Conference on Frontiers in Handwriting Recognition (ICFHR 2014). The dataset was selected from 1930 US Census General Population Schedule data which consists of 2652 images, plus 132,600 records in training dataset, 824 images plus 41200 records in testing dataset. Each record consists of five fields including relation, age, marital status, place of birth and name. These segmented handwritten word images are very challenging due to writer variations and different kinds of degradations, as illustrated in Fig. 4.

The performance is evaluated based on word level accuracy, which is defined as follows:

$$acc = \frac{\text{Corrected Recognized Word Number}}{\text{Total Word Number}} \tag{7}$$

The proposed technique is compared with the best performing submissions to the ANWRESH-2014 as listed in Table II. We tested our system on three out of the five fields as listed in Table II. The recognition performance of place of birth and name fields are not reported in this paper due to the limited time. To train satisfactory models, these two tasks require a very long training time even with super computational power which is not available to us. In fact, only the CIT team reported the result on name recognition in the ANWRESH-2014 competition. The network structure of our proposed system is provided in Table I.

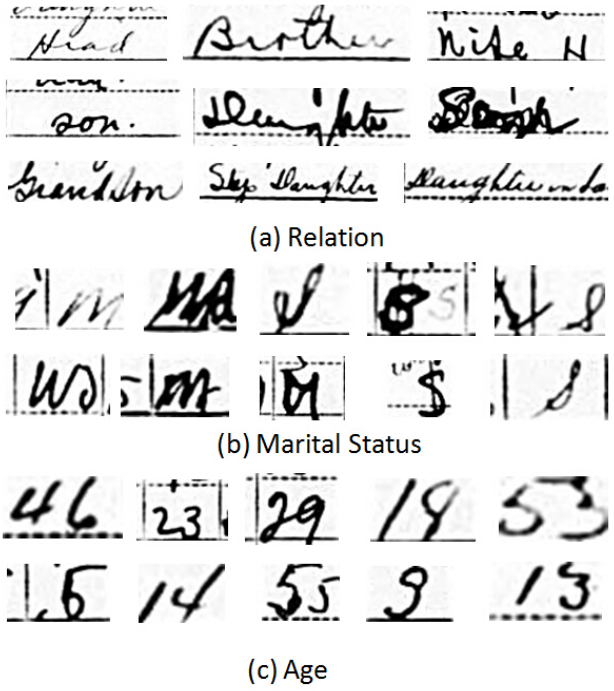

Fig. 4. Some example handwritten text images taken from ANWRESH-2014 dataset that can be successfully recognized by our system.

TABLE I. Network Structure of our Proposed Handwriting Recognition System

Task	Input Neurons	Output Neurons	Hidden Layer	Trainable Weight
Relation Geometric Feature	9	55	1	97846
Relation HOG Feature	40	55	2	364046
Marital Status Geometric Feature	9	7	1	90007
Marital Status HOG Feature	40	7	2	356207
Age Geometric Feature	9	25	1	93424
Age HOG Feature	40	25	2	359624

In particular, D1 [17] in Table II makes use of K-NN classifier on hand crafted features and D2 makes use of the traditional HMMs. Both systems just produce fair recognition results due to the degradation of the input handwritten text. F1 makes use of the Convolutional Neural Network (CNN) to

TABLE II. WORD RECOGNITION ACCURACY ON THE ANWRESH-2014 DATASET(%)

Datsets	Age	Relation	Marital Status	Average
CIT	**90.24**	89.57	97.15	92.32
D1	45.92	79.40	90.73	72.02
D2	54.85	62.77	75.26	64.29
F1	-	88.31	93.58	90.95
I2R	72.90	91.30	95.72	86.64
Geometrical Feature	77.95	85.88	92.62	84.82
HOG Feature	86.65	**92.31**	96.03	91.00
Proposed Combine	88.09	**93.87**	**97.99**	**93.32**

put the problem as a classification problem on the word level. However, it is heavily affected by unbalanced data and high variation of words, especially in the tasks of recognition of place of birth and age.

The top two algorithms including our submitted algorithm (I2R) and CIT system [18] both make use of the RNNs. The good performance demonstrates the robustness and effectiveness of RNNs in handwritten text recognition. Our previous submitted algorithm achieves best performance in relation recognition and second best performance in age and marital status recognition.

The superior performance of our proposed technique can be explained by a few factors. First, we incorporate the HOG column feature and combine it with the traditional geometrical features which helps for better handwritten text recognition accuracy. Second, we propose an ensemble recognition framework that combines different networks for better performance. As Table II shows, the our proposed technique outperforms the best-performing CIT technique [19] for the Relation and Marital Status tasks and obtains similar performance for the Age task. Note that compared with the CIT system that makes used of the multi-dimension RNNs [19] with about 1 million trainable weights, our proposed method is trained on a much simpler RNN model that requires less computational power.

IV. CONCLUSION

In this paper, we propose a new method combining the outputs of two networks, which are trained on different features extracted from the training data. The proposed technique has been tested on public datasets and achieved superior performance. In the future, we try to reduce the time cost for decoding and test the performance by combining more classifiers. Furthermore, we would like to extend our proposed method by combining more than two sets of features to obtain better performance.

REFERENCES

[1] G. Wilfong, F. Sinden, and L. Ruedisueli, "On-line recognition of handwritten symbols," in *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, 1996, pp. 935–940.

[2] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system," in *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, 2001, pp. 65–90.

[3] S. Espana-Boquera, M. J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 4, pp. 767–779, 2011.

[4] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney, "Hierarchical hybrid MLP/HMM or rather mlp features for a discriminatively trained gaussian hmm: a comparison for offline handwriting recognition," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 3541–3544.

[5] S. Marukatat, T. Artieres, R. Gallinari, and B. Dorizzi, "Sentence recognition through hybrid neuro-markovian modeling," in *Document Analysis and Recognition (ICDAR), 2001 6th International Conference on*, 2001, pp. 731–735.

[6] E. Caillault, C. Viard-Gaudin, and A. R. Ahmad, "Ms-tdnn with global discriminant trainings," in *Document Analysis and Recognition (ICDAR), 2005 8th International Conference on*, 2005, pp. 856–860.

[7] J. Schenk, G. Rigoll, and T. U. Mnchen, "Novel hybrid NN/HMM modelling techniques for on-line handwriting recognition," in *Processing of the International Workshop on Frontiers in Handwriting Recognition*, 2006, p. 619623.

[8] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 855–868, May 2009.

[9] X. Zhang and C. L. Tan, "Unconstrained handwritten word recognition based on trigrams using blstm," in *In Pattern Recognition (ICPR), 2014 22nd International Conference on*, 2014, pp. 2914–2919.

[10] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Asian Conference on Computer Vision*, 2014.

[11] A. Vinciarelli and J. Luettin, "A new normalization technique for cursive handwritten words," in *Pattern Recognition Letters*, vol. 22, no. 9, 2001, pp. 1043–1050.

[12] B. Su, S. Lu, and C. L. Tan, "Robust document image binarization technique for degraded document images," *Image Processing, IEEE Transactions on*, vol. 22, no. 4, pp. 1408–1417, April 2013.

[13] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, 2005, pp. 886–893.

[14] A. Graves and J. Schmidhuber., "Framewise phoneme classification with bidirectional lstm and other neural network architectures." in *Neural Networks*, vol. 18, 2005, p. 602610.

[15] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, November 1997.

[17] J. Reese, M. Murdock, S. Reid, and B. Hamilton, "ICFHR2014 competition on word recognition from historical documents: Anncestry word recognition from segmented historical documents (ANWRESH)," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, Sept 2014, pp. 803–808.

[18] G. Leifert, T. Grüning, T. Strauß, and R. Labahn, "CITlab ARGUS for historical data tables," 2014.

[19] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2009, pp. 545–552.