2013 12th International Conference on Document Analysis and Recognition

Offline Printed Urdu Nastaleeq Script Recognition with Bidirectional LSTM Networks

Adnan Ul-Hasan[†*], Saad Bin Ahmed[†*], Sheikh Faisal Rashid[*], Faisal Shafait[‡] and Thomas M. Breuel[*]
[*]Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany
Email: {adnan,tmb}@cs.uni-kl.de, isaadahmed@gmail.com, rashid@iupr.com
[‡]Department of Computer Science and Software Engineering, The University of Western Australia, Perth, Australia
Email: faisal.shafait@uwa.edu.au

***Abstract*—Recurrent neural networks (RNN) have been successfully applied for recognition of cursive handwritten documents, both in English and Arabic scripts. Ability of RNNs to model context in sequence data like speech and text makes them a suitable candidate to develop OCR systems for printed Nabataean scripts (including Nastaleeq for which no OCR system is available to date). In this work, we have presented the results of applying RNN to printed Urdu text in Nastaleeq script. Bidirectional Long Short Term Memory (BLSTM) architecture with Connectionist Temporal Classification (CTC) output layer was employed to recognize printed Urdu text. We evaluated BLSTM networks for two cases: one ignoring the character's shape variations and the second is considering them. The recognition error rate at character level for first case is 5.15% and for the second is 13.6%. These results were obtained on synthetically generated UPTI dataset containing artificially degraded images to reflect some real-world scanning artefacts along with clean images. Comparison with shape-matching based method is also presented.**

I. Introduction

Recurrent neural network (RNN) are good at context-aware processing and recognizing patterns occurring in time-series [1]. The main drawbacks of traditional RNNs are the requirement of pre-segmented input and that the input on the hidden layer either decays or blows-up exponentially [2], [3]. LSTM architecture have given Recurrent Neural Networks (RNN) a rebirth by overcoming many limitations and problems of earlier RNN architectures like [4], [5], [6]. The hidden layer of an LSTM network consists of recurrently connected blocks that in turn contains internal units whose activation is controlled by input, forget and the output gates. The recurrent connections of cells are controlled by the forget gate. So, the network can hold the information as long as the forget gate is switched on. More details on RNN and LSTM architecture may be found in [7]. Graves [7] introduced Bi-directional LSTM (BLSTM) architectures for accessing context in both forward and backward directions. BLSTM is a combination of bi-directional neural network (BRNN) and LSTM architectures and it uses two hidden layers, one for forward pass (from left to right) and the other for backward pass (from right to left). Both layers are then connected to a single output layer. Pre-segmented input data is a peculiar requirement of the original RNNs, which limited the utility of traditional RNN for sequence data such as speech and handwriting recognition [7]. To avoid this requirement Graves et al. [8] used a forward-backward algorithm to align transcripts with the output of the neural network (referred to as Connectionist Temporal Classification (CTC)).

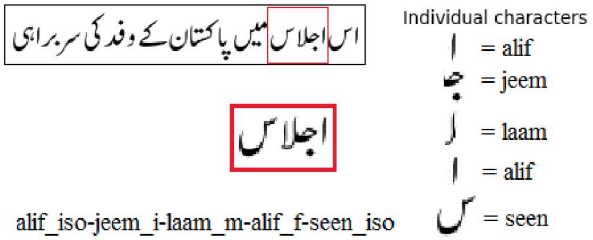

Fig. 1. Decomposition of Urdu words into characters. The word in red block is a combination of four characters. There are three glyphs and four ligatures in this word. Individual characters are shown on right side. *_iso* is used for character in isolation, *_i* for initial position, *_m* for middle position and *_f* is used to indicate final position.

We have started benchmarking various LSTM architectures for printed text recognition at our group[1]. LSTM networks have shown higher recognition accuracies on other sequence labelling tasks like speech [9] and handwriting recognition [10]. In our knowledge, no work has been reported to-date showing performance of LSTM-based RNNs on printed text recognition. We started benchmarking LSTM networks for English and Fraktur (German historical script) first[2], and found in preliminary experiments that the 1D LSTM networks performs better than their multidimensional siblings. Seeing their performance on Latin scripts, we decided to apply 1D LSTM to Urdu Nastaleeq script as well. In this paper, we demonstrate the application of 1D bidirectional LSTM networks to the printed Urdu Nastaleeq recognition. 1D BLSTMs are different than 2D or multidimensional BLSTMs in how input sequence is given to the network.

Urdu is the national language and lingua franca of Pakistan and is considered as one of the important languages of the Indian subcontinent. It belongs to the family of Nabataean scripts and shares many common properties of other family members like Arabic and Persian. Some of its salient features are writing from right to left, presence of huge number of ligatures *(connected set of components with associated dots and diacritics)*, variations in the character's shape depending

[†] These authors contributed equally

[1]Image Understanding and Pattern Recognition Research, www.iupr.com

[2]see our paper "High-Performance OCR for Printed English and Fraktur using LSTM Networks" in ICDAR-2013 proceedings.

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.212

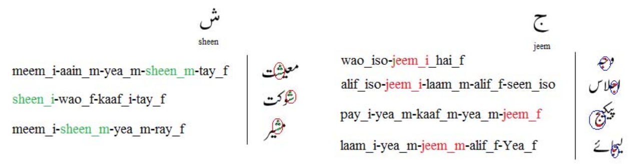

Fig. 2. Urdu characters in their actual form are shown at the top of each block. Shape of the same character at various locations are shown below. $\_x, x \in \{iso, i, m, f\}$ indicates the position of a character in the ligatures.

on its location in a ligature (context), kerning, etc. Due to its inherit complexity with respect to connecting ligatures, identifying glyphs during word generation as shown in Figure 1 is considered a subtle task and poses unique challenges to be recognized.

An important issue in Urdu language is change in shape of a character depending upon its position in a word. Context plays an important role in determining the particular shape of a character at a particular position. The shape of a character when it is located in isolation or at initial, middle or final position in a word may differ significantly. Figure 2 shows different shapes of some of the Urdu characters when isolated and at other places in a ligature. Dots and diacritics give meaning and identity to any character in a ligatures. Associating dots and diacritics to their base character is also a challenging task in Urdu because of their relatively smaller sizes. There are two prominent writing styles in Urdu: *Naskh*[3] and *Nastaleeq*[4]. Figure 4 shows these two styles. The standard Urdu language written in magazines, newspaper, and books are in Nastaleeq script, while most of Urdu online material is available in Naskh. An important distinction between the two styles is that the Naskh's flow is horizontal from right to left, while the Nastaleeq's flow is diagonal from right top to left bottom. This makes Nastaleeq to occupy less space for a ligature than the Naskh font. The scope of the current work is confined only to Nastaleeq script.

Urdu script consists of 45 basic characters. Five (05) characters can only occur in isolation, 10 can only occur in first position or at last position, 2 characters can occur only at the end of a ligatures, and only 1 character can occupy position in middle; it can't be located in any other position. Remaining 27 characters may occur in isolation, at the beginning, at the end or in the middle of a ligature. Moreover, there are 26 punctuation marks, 8 honorific marks, and 20 digits. Some common punctuations (like $\%, <, >$, parentheses, etc.) and English numerals are also used in Urdu publications frequently; so they are also included in the list of possible characters/class-labels (in terms of machine learning terminology). Characters belonging to above-mentioned eight categories are shown in Figure 3. So, in total there are 99 individual labels. Moreover, if we take the shapes of various characters as a separate label, then there are 191 labels. The last column in Figure 3 details the number of classes in each category as per their number of shapes depending on their position in a ligature.

The recognition of cursive characters is an active research field. One such work is proposed by [11] for cursive character

[3]http://en.wikipedia.org/wiki/Naskh_(script)

[4]http://en.wikipedia.org/wiki/Nastaleeq

Category	Urdu Characters	Classes
Characters that can occur only in isolation. (CAT-I)	آ، ژ، ئے، ؤ، ء	5
Characters that can not occur in the beginning or in middle of a ligatures. (CAT-II)	ا، د، ڈ، ذ، ر، ڑ، ز، ں، و، ے	20
Characters that may occur in isolation, at the beginning, at the end or in the middle of a ligature. (CAT-III)	ب، پ، ت، ٹ، ث، ج، چ، ح، خ، س، ش، ص، ض، ط، ظ، ع، غ، ف، ق، ک، گ، ل، م، ن، ہ، ھ، ی	108
Characters that can occur only in isolation or in the middle of a ligature. (CAT-IV)	ئ	2
Characters that can occur only at the end of a ligature. (CAT-V)	ۃ، ۂ	2
Honorific marks (CAT-VI)	ﷲ، ﷺ، ﷽، ؑ، ؓ، ؒ، ؐ، جل جلالہ	8
Punctuation marks (CAT-VII)	،، (،)، [،]، {، }، /، >، <، !، ٪، ؛، :، "، ‘‘، ’’، ؟، ـ، ۔، ؍، ؎، ؏، ؂	26
Numbers (CAT-VIII)	0123456789, ۰۱۲۳۴۵۶۷۸۹	20

Fig. 3. Character categorization. There are a total of 45 characters in five categories based on the position in a ligature.

recognition using Support Vector Machine (SVM) which is based on segmentation. They performed experiments on isolated characters and computed local and global features of it. Another work in relation to cursive script is represented by Nagata [12]. They presented an OCR approach for cursive characters of a language which has a large character set (like Chinese, Japanese etc). They used approximate character shape similarity and a word segmentation algorithm with support of language model. Graves [13] evaluated multidimensional LSTM (MDLSTM) for offline Arabic handwriting recognition. They first divided the input image into 3×4 sub-images and then scanned them by four MDLSTM layers. They scanned the image in all four directions (right-to-left, left-to-right, top-to-bottom and bottom-to-top) to capture the context. Their system won both ICDAR-2007 [14] and ICDAR-2009 [15] competitions. Sankaran and Jawahar [16] applied BLSTM networks for Devanagari script OCR problem. Frinken et al [17] applied BLSTM networks to word spotting problem by modifying the CTC token-passing algorithm.

Unfortunately, Urdu script has not yet thoroughly been introduced to state-of-the-art research and till date no significant OCR system specialized for Urdu Nastaleeq script has been reported. However, there has been an increase in the interest of the research community especially from the Indian subcontinent to address the Urdu Nastaleeq OCR problem. Some earlier work was done by Pal et al. [18] to recognize individual Urdu characters. They recognized these characters using a combination of topological, contour and water-reservoir concepts. Segmentation of Urdu ligatures is very complex and error-prone, therefore, there has been a shift in interest to try segmentation-free approaches. Javed et al. [19]

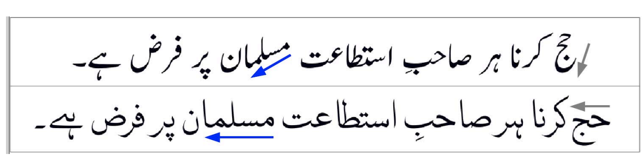

Fig. 4. Two commonly used styles for Urdu scripts. Nastaleeq script (above) is predominately used for Urdu publications, while Naskh (below) is used for web-viewing. Arrows show the direction of reading flow. Note that Nastaleeq flow is from top-right to left-bottom.

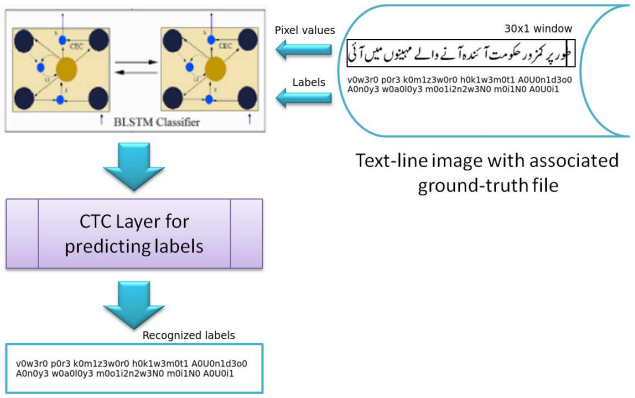

Fig. 5. Training pipeline. A 30×1 window traversed the height-normalized image and the BLSTM network is fed with a 1D sequence of corresponding pixel values. CTC layer performs the output-transcription alignment.

proposed a HMM-based segmentation-free Urdu OCR system for 1500 frequently used Urdu ligatures. Sabbour et al. [20] presented a segmentation-free approach for Urdu and Arabic scripts. They modified the traditional shape-context features method [21] to extract features from Urdu/Arabic ligatures and then applied k-nearest neighbour classifier to recognize the ligatures. Instead of computing shape-context features of whole ligature, they first divided the ligature image into four parts; then they computed the shape-context of points in each of the region separately. Subsequently, these features were concatenated to define the cumulative feature vector.

The aim of current work is to further extend the research towards reliable OCR for Nastaleeq script. The next section (Section II) describes the preprocessing and feature extraction step. Configuration and training procedure for our BLSTM network is outlined in Section III. Section IV presents the experimental evaluation of LSTM networks for Nastaleeq script and the results are discussed in Section V.

II. Preprocessing and Feature Extraction

Baseline information of a text line is an important feature in distinguishing a number of common characters. It is therefore necessary to normalize the input images to a specific height, so that this information is uniformly available. Currently, there are no Nastaleeq-specific normalization method reported. In the current work, each text-line image was rescaled to a fixed height. Raw pixel values are used as features and no other sophisticated features were extracted. A 30×1 window is traversed over the text-line image and the resulting 1D sequence is fed to BLSTM network for training.

III. BLSTM Network configuration

As mentioned earlier, BLSTM architecture with CTC output layer was employed to evaluate RNN for Urdu script. A publicly available RNN library [22] was used for evaluation. Implementation of both 1D and multidimensional BLSTM networks is provided in this library along with CTC output layer. Size of hidden-layer, learning rate and momentum are other tunable parameters.

For training purpose, the normalized gray-scale input text-line image was scanned from left to right to extract the features. The corresponding transcriptions were reversed to make it consistent with the input image (Urdu is read from right to left). Figure 5 shows the complete training pipeline. Normalized text-line images along with their transcriptions were fed to the network, which performed the forward propagation step first. Alignment of output with associated transcriptions is done in the next step and then finally backward propagation step was performed. After each epoch, training and validation error were computed and the best results were saved. When there was no significant change in training and validation errors for a pre-set number of epochs, the training stopped. Training and validation errors were recorded and the network was evaluated on test set. There are four parameters, which need to be tuned; namely input-image size, hidden-layer size, learning rate and the momentum. The input image height was set to 30 and was not altered. Momentum value was also kept fixed at 0.9. Other parameters were changed and the suitable parameter was found. Parameter tuning is discussed in details in Section IV-C. Briefly, best parameters for hidden-layer size and learning rate were 100 and 0.0001 respectively. For this network with best parameters, training and validation errors as a function of number of epochs are shown in Figure 6. This network took 77 epochs to converge. However, it can be seen that the validation error is minimum after 41 epochs (marked as dotted-line in Figure 6). This network is returned as the best network.

IV. Experimental Evaluation

This section discusses the results of evaluating BLSTM architecture on printed Urdu script. Two kinds of evaluations were performed for Urdu Nastaleeq script. In the first evaluation, shape variations at all four positions (isolation, beginning, middle and end) were considered (191 classes). In second evaluation, only basic labels were considered (99 classes).

A. Database

A synthetic database used by Sabbour et al [20], called UPTI *(Urdu Printed Text Images)*-dataset, was used for evaluation. This Urdu dataset consists of $10,063$ synthetically generated text lines. Various degradation techniques [23] were applied to increase the size of dataset. 12 sets were generated by varying four parameters, namely, *elastic elongation, jitter, sensitivity* and *threshold*. This dataset contains both ligatures

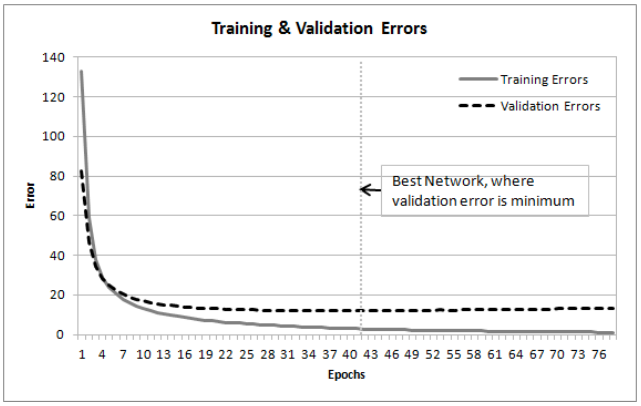

Fig. 6. CTC Error rate (on character level) during training. The validation error is minimum at 41^{st} epoch.

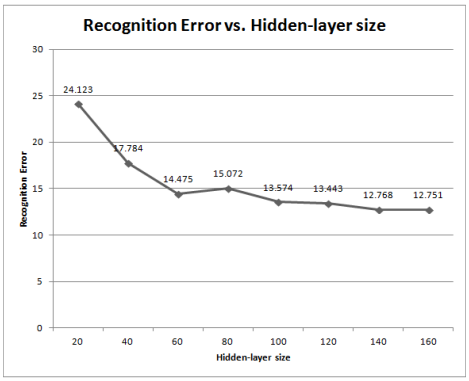

Fig. 7. Recognition error decreases with increasing the number of units in hidden layer. However, it takes more time to train the network at the same time.

and lines versions; however, only lines dataset was used for the present work. These lines were divided into three sub-categories, training (46%), validation (34%) and testing (20%). Each set was build such that text-lines from all 12 degraded categories and 1 clean category were taken in equal proportions. The ground-truth of these text-line images was also available.

B. Evaluation Metric

As suggested in [24], CTC Error is the most appropriate error measure to be used as it gives faster convergence than other options like Character Error measure. So, the same error criterion was used in the present work. The overall accuracy is calculated using ratio of insertions, deletions and substitution w.r.t. total number of characters in transcription.

C. Parameter Selection

In the present work, two parameters namely *learning rate*, *momentum* and *number of hidden-layers* were evaluated for their respective effect on the recognition accuracies. Parameter selection was done for case where we considered the ligature shape variations (191 classes), and then the optimal parameters found were useRegarding not receiving your emails:d for other case where ligature shape variation was not considered. First, the most appropriate number of hidden-layers were determined keeping learning rate constant at 0.0001. We trained BLSTM networks with hidden-layer of sizes 20, 40, 60, 80, 100, 120,

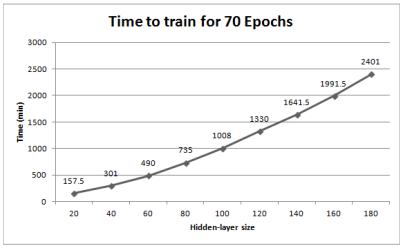

Fig. 8. Training time as a function of Hidden-layer size. This time is taken for computing 70 epochs during training on Intel Xeon 2.53 GHz, 40GB RAM, Ubuntu 12.04 OS.

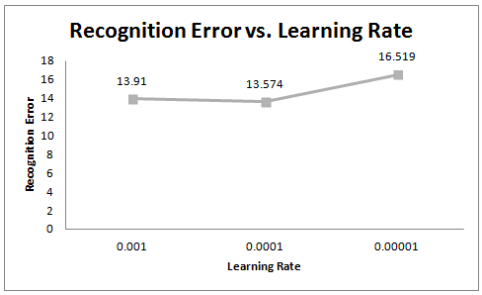

Fig. 9. Learning rate of 0.0001 gives the lowest recognition error.

140 and 160. The comparison of respective recognition-errors on test set is shown in Figure 7. The training time as a function of hidden-layer sizes is shown in Figure 8. From Figure 7 and Figure 8, we can deduce two points; first that increasing the number of hidden-layer sizes decreases the recognition error but at the same time, training for network with large number of hidden-layers requires more time. Moreover, it is also noted that increase in training-time is almost linear, while increase in hidden-layer sizes does not increase accuracy more than 5% when the hidden-layer size is from 100 to 160. So, it was decided to select 100 as the optimal hidden-layer size for the present work.

In the next step, keeping the hidden-layer size to 100, the learning rate was varied between 0.001, 0.0001 and 0.00001. The comparison of respective recognition errors on test set is shown in Figure 9. It is evident from the figure, that a learning rate of 0.0001 is most suitable. Similar network parameters have been reported in [10] and [24].

D. Results

There were $2,003$ text-line images in the test set. As mentioned in Section IV, BLSTM networks have been evaluated for two scenarios: considering ligature shape variations and ignoring shape variations in the ligatures. For the first case, the recognition error was 13.574%, (Total No. of labels, $N = 74,279$), while for the second case, the recognition error was 5.15% ($N = 74,279$).

As mentioned in Section I that there have not been many OCR systems available for Urdu Nastaleeq script. Only shape-matching based OCR system proposed by Sabbour et al [20] is reported in recent times. They evaluated their system on clean printed text as well on some of the artificially degraded versions of the clean dataset. They achieved 11.2% letter error rate on clean images. They also reported error rates for various degradation effects on individual basis. There is no error rate reported for mixed dataset that we used in our evaluations. Moreover, they did not consider the case where ligature shape variations are not considered (where we achieved 5% error rate). It is therefore not possible to do one to one comparison in true sense, but it can be seen that our system performed better taking into consideration that clean images are only $\frac{1}{13}$ of our test-dataset. Secondly, performance of their system changes significantly by changing degradation parameters' values.

Some sample outputs images along with their original images are shown in Figure 10. BLSTM network performs generally well for most labels; however, it appears that it

Fig. 10. Input/output from the BLSTM-based OCR illustrating capabilities and errors. Figures (a), (c) and (e) represent the original images, whereas Figures (b), (d) and (f) represent the output of BLSTM network.

sometimes fails to recognize the location of dots and diacritics (e.g. Figure 10-(a)-(e)). As mentioned earlier, the dots and diacritics are very important to give meaning to a ligature. Other errors are mostly due to very similar shapes of a ligature (e.g. Figure 10-(e) and (f)).

V. Discussion

The results of two evaluations are surprising, because, at the beginning of experiments, it was perceived that by incorporating the shape variations as separate classes would increase the recognition accuracy because we have less variations within a specific class. There could be two issues: by ignoring the shape variations, no. of samples per class definitely increases. So, increased numbers per class is resulting in better training and thereby reducing recognition errors. On the other side, when considering shape variations, number of samples per class are small and that could lead to insufficient training and thus resulting in higher recognition errors. One may argue that less no. of classes generally means better classification accuracy; however, it should be noted that the dataset remains the same, so by merging many classes, we actually are increasing the variations. This conflict may be solved by having such a dataset in which samples per class in both variations are equivalent.

The context-capturing property of RNN makes it a better candidate for Nabataean scripts like Arabic, Urdu, Persian, etc. than other neural networks based methods. The next step to extend this research is to apply multidimensional LSTMs [25] and see whether they perform better than 1D LSTM. It is possible that multidimensional networks would localize the position of dots and diacritics better, thereby further lowering the error rates.

References

[1] A. Senior and T. Robinson, "Forward-backward retraining of recurrent neural networks," in *NIPS*, D. Touretzky, M. Mozer, and M. Hasselmo, Eds., vol. 8, San Mateo, CA., 1996, pp. 743–749.

[2] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," in *A Field Guide to Dynamical Recurrent Neural Networks*, Kremer and Kolen, Eds. IEEE Press, 2001.

[3] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Trans. on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[4] J. L. Elman, "Finding Structure in Time." *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[5] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: a new framework for neural computation based on perturbations." *Neural Computations*, vol. 14, no. 11, pp. 2531–2560, Nov. 2002.

[6] H. Jaeger, "Tutorial on Training Recurrent Neural Networks, Covering BPTT, RTRL, EKF and the 'Echo State Network' approach," Sankt Augustin, Tech. Rep., 2002.

[7] A. Graves, "Supervised sequence labelling with recurrent neural networks." Ph.D. dissertation, Technical University Munich, 2008.

[8] A. Graves, S. Fernàndez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Nets," in *ICML*, 2006.

[9] A. Graves, D. Eck, N. Beringer, and J. Schmidhuber, "Biologically Plausible Speech Recognition with LSTM Neural Nets." in *BioADIT*, ser. Lecture Notes in Computer Science, A. J. Ijspeert, M. Murata, and N. Wakamiya, Eds., vol. 3141. Springer, 2004, pp. 127–136.

[10] A. Graves, M. Liwicki, S. Fernàndez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, 2009.

[11] F. Camastra, "A SVM-based cursive character recognizer." *Pattern Recognition*, vol. 40, no. 12, pp. 3721–3727, 2007.

[12] M. Nagata, "Japanese OCR Error Correction using Character Shape Similarity and Statistical Language Model." in *Int. Conf. on Computational Linguistics*, 1998, pp. 922–928.

[13] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*, ser. Studies in Computational Intelligence. Springer, 2012, vol. 385.

[14] V. Märgner and H. E. Abed, "Arabic Handwriting Recognition Competition." in *ICDAR*. IEEE Computer Society, 2007, pp. 1274–1278.

[15] ——, "ICDAR 2009 Arabic Handwriting Recognition Competition." in *ICDAR*. IEEE Computer Society, 2009, pp. 1383–1387.

[16] N. Sankaran and C. V. Jawahar, "Recognition of printed Devanagari text using BLSTM Neural Network." in *ICPR*. IEEE, 2012, pp. 322–325.

[17] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A Novel Word Spotting Method Based on Recurrent Neural Networks." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 211–224, 2012.

[18] U. Pal and A. Sarkar, "Recognition of Printed Urdu Text," in *ICDAR*, Eidenburg, UK, Aug. 2003, pp. 1183–1187.

[19] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, "Segmentation Free Nastalique Urdu OCR," *World Academy of Science, Engineering and Technology*, vol. 46, pp. 456–461, 2010.

[20] N. Sabbour and F. Shafait, "A Segmentation Free Approach to Arabic and Urdu OCR," in *DRR XX (Part of the IS&T/SPIE 25th Annual Symposium on Electronic Imaging)* , Feb. 2013.

[21] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[22] A. Graves, "RNNLIB: A recurrent neural network library for sequence learning problems." [Online]. Available: http://sourceforge.net/projects/rnnl

[23] H. S. Baird, "Document Image Defect Models ," in *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, Eds. New York: Springer-Verlag, 1992.

[24] A. Graves, "Offline Arabic Handwriting Recognition with Multidimensional Recurrent Neural Networks," in *Guide to OCR for Arabic Scripts*, V. Märgner and H. E. Abed, Eds. Springer, 2012, ch. 12, pp. 297–313.

[25] A. Graves, S. Fernàndez, and J. Schmidhuber, "Multi-Dimensional Recurrent Neural Networks," in *NIPS*, vol. 22. Vancouver: MIT Press, 2009, pp. 545–552.