Unconstrained OCR for Urdu using Deep CNN-RNN Hybrid Networks

Mohit Jain, Minesh Mathew and C.V. Jawahar
Center for Visual Information Technology, IIIT Hyderabad, India
{mohit.jain,minesh.mathew}@research.iiit.ac.in, jawahar@iiit.ac.in

***Abstract*—Building robust text recognition systems for languages with cursive scripts like Urdu has always been challenging. Intricacies of the script and the absence of ample annotated data further act as adversaries to this task. We demonstrate the effectiveness of an end-to-end trainable hybrid CNN-RNN architecture in recognizing Urdu text from printed documents, typically known as Urdu OCR. The solution proposed is not bounded by any language specific lexicon with the model following a segmentation-free, sequence-to-sequence transcription approach. The network transcribes a sequence of convolutional features from an input image to a sequence of target labels. This discards the need to segment the input image into its constituent characters/glyphs, which is often arduous for scripts like Urdu. Furthermore, past and future contexts modelled by bidirectional recurrent layers aids the transcription. We outperform previous state-of-the-art techniques on the synthetic UPTI dataset. Additionally, we publish a new dataset curated by scanning printed Urdu publications in various writing styles and fonts, annotated at the line level. We also provide benchmark results of our model on this dataset.**

***Keywords*-OCR, Urdu OCR, Deep Learning, Urdu Dataset, Text Recognition.**

I. Introduction

Though a lot of research has been done in the field of text recognition, the focus of the vision community has been primarily on English. While, Arabic script has started to get some attention as far as text recognition is concerned [1], [2], [3], works on other languages which use the *Nabataean* family of scripts, like Urdu and Persian, are very limited. This is quite aberrant considering the fact that there are about 60 to 80 million speakers of the Urdu language. It is ranked as the fifth most spoken language catering to 4.7 percent of the world's population, spoken widely in Pakistan where it is the national language, and India where it is recognized as one of the 22 official languages.

Digitizing historical documents is crucial in preserving the literary heritage. With the availability of low cost mobile capturing devices, institutions all over the world are preserving their literature in the form of scanned documents. Huge amounts of valuable Urdu literature from philosophy to sciences is vanishing and being rendered useless because it has not been digitized till now. Moreover, many of the native speakers of this language can only read and write in Urdu and hence there's a scarcity of information and data for them on the internet and in digitized form. A major barrier to this process of digitization is the huge overhead introduced in indexing and retrieval of such documents. All these problems indicate towards the strong need for developing a robust OCR system for Urdu.

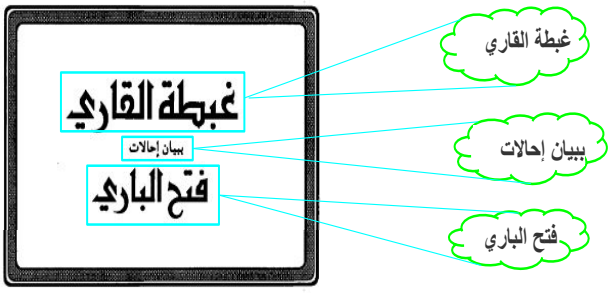

Figure 1: Example of Urdu Optical Character Recognition (OCR). Our work deals only with the recognition of cropped words/lines. Bounding boxes were provided manually.

A. Intricacies of Urdu

While Urdu can be written in various styles like *Naskh, Nastaliq, Kofi, Thuluth, Devani* and *Riqa*, the most commonly used writing styles are *Naskh* and *Nastaliq*. Printed media like magazines, newspapers and books generally follow the *Nastaliq* style of writing, whereas online material is mostly available in the *Naskh* style of writing. Both these styles are written in a semi-cursive fashion from right-to-left, similar to Arabic. However, a prominent distinction between the two styles is the flow in which these scripts are written. *Naskh* has a horizontal writing flow from right to left while *Nastaliq*'s flow is diagonal from right-top to left-bottom (Fig. 2). Another peculiar characteristic of Urdu scripts is that unlike words, numerals are written from left-to-right. Since the problem of OCR caters to printed text in documents, we would primarily be dealing with the *Nastaliq* style of writing.

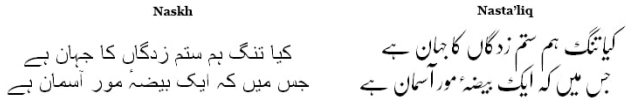

Figure 2: Two commonly used styles for Urdu scripts; *Nastaliq* (notice the flow from right-top to left-bottom) and *Naskh*.

The scripts for Urdu use 45 characters as their basic building blocks. Of these, 5 characters appear only in isolation, 10 appear only at the beginning or at the end of a word and 1 character is bound to occupy only middle positions. The other 27 characters are free to occur in isolation or at the beginning, middle or end of a word. Additionally, there are 26 punctuation marks, 8 honorific marks and 10

digits that complete the character-set for Urdu. However, from an OCR problem's perspective, English numerals and punctuation marks are also a common occurrence in the printed Urdu documents domain and hence need to be recognised by any practical solution. Also, the position of a character in a word (at the beginning, middle or end) changes the shape of the glyph used to represent the character completely. Accounting for all the above mentioned variations, there are a 192 distinct glyphs that might occur in an Urdu publication.

Non-standardization of fonts and their rendering schemes, especially for the publications printed prior to the emergence of Unicode, has made the development of an Urdu OCR further challenging. Moreover, due to the mismatch between the basic units for representation and rendering (Unicode characters v/s various fonts and writing styles), creation of synthetically rendered data samples to employ fully supervised machine learning methods is all the more difficult.

B. Related Work

Recognizing cursive scripts has been an active field of research. Initial work in this domain was by [4], who presented an OCR solution for languages with large character-sets (like Japanese, Chinese, etc.). They utilised an approximate character shape similarity on top of a word segmentation algorithm using language models. Later on, [5] proposed a segmentation based approach for OCR using Support Vector Machine (SVM). They computed local and global features on top of segmented cursive characters.

However, Urdu OCR still remains in a nascent stage as compared to other cursive scripts used in the Asian continent. Among the initial works, [6] used morphological operations and character-specific filters to preprocess each segmented character/glyph from a line image. They used a heuristics based approach on the character-chain-code created to figure out which class label (Urdu glyph) the segmented image gets assigned. Most of the work in the domain of Urdu OCR utilised handcrafted features and used nearest neighbour techniques to perform classification. Like, [7] used connected components information along with stroke information computed using baseline detection as feature descriptors. Features were also obtained by passing sliding windows of various filters over the image. Similarly, [8] used contour extraction techniques and shape context information to create feature descriptors for each ligature/glyph. Finally, classification was done using k-Nearest Neighbour techniques in both of these works.

Segmentation free methods have come into light recently. These methods are generally based on Hidden Markov Models (HMM) or Recurrent Neural Networks (RNN). In one such work, [9] train multiple HMMs for each type of ligature and text-body of the characters. These models are used to create a feature matrix based on the number and positions of diacritics to perform classification on the text in a segmentation-free fashion. Some methods improved the transcription accuracy with the help of language models [10]. They used uni-gram, bi-gram and tri-gram counts for words and ligatures to rank possible word predictions based on probabilities derived from a lexicon of the most frequently used Urdu words. Recently, [11] use Convolutional Neural Networks (CNN) features with a Multi-dimensional Long Short-Term Memory (MDLSTM) layer [12], and achieve comparable results to our proposed solution. Here, a single CNN layer extracts low level features from input data and feeds them to a MDLSTM layer. The concatenation of all the resulting vectors forms a final feature vector for a Connectionist Temporal Classification (CTC) [13] output transcription layer.

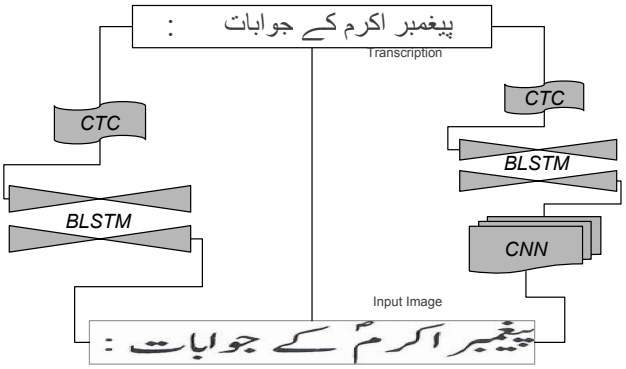

Figure 3: Flow diagram representing the two solution architectures; BLSTM (left) and HYBRID CNN-RNN (right).

The rest of the paper is organized as follows; Section II describes the solution architectures used for performance comparison. Section III focuses on the problem of Urdu OCR from a transcription perspective. Section IV presents our results on the UPTI database and our new benchmark dataset, followed by a discussion based on the qualitative results. Section V concludes with the findings of our work.

II. Solution Architectures

Segmentation of characters in cursive scripts is a challenging task and prone to errors. Therefore, there has been a shift in research interests to try segmentation-free approaches which have proven to be quite effective for Indian Scripts OCR [14], [15]. On similar lines, [16], [17] use Bi-directional LSTM networks [18] along with the (CTC) loss on raw image features to perform transcription in an end-to-end fashion for Arabic script.

A novel approach combining the robust convolutional features and transcription abilities of RNNs was introduced for English scene text by [19]. Our hybrid CNN-RNN solution is inspired from this work with changes made to cater for the intricacies of Urdu scripts. The unconstrained hybrid CNN-RNN architecture can be trained in an end-to-end fashion using the CTC loss. By providing an *unconstrained* solution, we mean that our model is not bounded by any language-lexicon and any possible combination of the script's character-set can be recognized. A visual comparison of the two methods can be seen in Fig. 3.

Figure 4: Visualization of the hybrid CNN-RNN architecture with a 7-layer-deep convolutional block. The symbols `k', `s' and `p' stand for kernel size, stride and padding size respectively.

A. BLSTM Architecture

RNNs are an optimal choice for our unconstrained end-to-end system as they have a strong capability of capturing contextual information within a sequence. Additionally, RNNs are capable of handling variable length sequences. Since the number of parameters in an RNN is independent of the length of the sequence, we can simply unroll the network as many times as the number of time-steps in the input sequence. This helps us perform unconstrained recognition, where the predicted output can be any sequence of labels from the entire label set (192 unique characters/glyphs and punctuation marks in our case).

Traditional RNN units (vanilla RNNs) face the problem of vanishing gradients [20] and hence LSTM (LSTM) units are used, which were specifically designed to address the vanishing-gradients problem [21], [18]. In a text recognition problem, contexts from both directions (left-to-right and right-to-left) are useful and complementary to each other in performing correct transcription. Therefore, a combined forward and backward oriented LSTM is used to create a bi-directional LSTM unit. Multiple such bi-directional LSTM layers can be stacked to make the network deeper and gain higher levels of abstractions over the image-sequences as shown in [22].

The transcription layer at the top of the network is used to translate the predictions generated by the recurrent layers into label sequences for the target language. The CTC layer's conditional probability is used in the objective function as shown in [13]. This objective function calculates a cost value directly from an image and its ground truth label sequence, eliminating the need to manually label all the individual components in a sequence.

B. Hybrid CNN-RNN Architecture

The proposed hybrid CNN-RNN networks have multiple convolutional layers stacked at the head of the BLSTM architecture described in the previous subsection. They consist of three major components; initial convolutional layers, middle recurrent layers and a final transcription layer but vary in the number of convolutional layers. The convolutional layers obtain robust feature representations from the input images. These features are then passed on to the recurrent layers which transcribe them into an output sequence of labels representing the Urdu characters/glyphs.

The convolutional layers follow a VGG [23] style architecture without the fully-connected layers. The input image first goes through the convolutional layers where feature maps are extracted from it. All the images are scaled to a fixed height before being fed to the convolutional layers. After the convolutional operations, the sequence of feature maps obtained are split column-wise to create feature vectors which act as time-steps for the recurrent layers. These feature descriptors are highly robust and most importantly can be trained to be adopted to a wide variety of problems [24], [25], [26].

The recurrent layers take each feature vector from the feature sequence generated by the convolutional layers and make predictions. The sequence-to-sequence transcription is achieved by using a CTC loss layer at the output. A visualization of this process with complete network configurations can be seen in Fig. 4.

III. Urdu Text Transcription

The utility of sequence-to-sequence transcription resides in its ability to predict output sequences directly from input sequences without the need for a target label associated with each time-step. In addition, contextual modelling of the sequence in both the directions, forward and backward, is achieved by the bidirectional LSTM block. Contextual information is critical in making accurate predictions for a language like Urdu where there are many similar looking characters/glyphs. The hybrid architecture replaces the need for any handcrafted features to be extracted from the image. Instead, more robust convolutional features are fed to the RNN layers. The CNN-RNN hybrid network, though composed of multiple modules of convolutional and recurrent layers, can be trained using a single CTC loss function and is end-to-end trainable.

A. Datasets

The hybrid CNN-RNN model was trained on a dataset consisting of approximately 1500 pages of scanned Urdu text annotated at the page-level. This dataset was collected as part of a consortium project for the Government of India [27]. Bounding boxes for the lines on these pages were manually added and after cleaning the data, a total of 29876 line images were obtained. From these line images, 27000 images were used for training the model and 1876 images were kept aside for validation.

For testing the trained models performance, we compare our results on the UPTI (Urdu Printed Text Images)

dataset [8] and provide benchmark results for the IIIT-Urdu OCR dataset we release.[1]

The UPTI dataset consists of 10,063 synthetically generated Urdu text line images. The dataset consists of both ligature and line versions, however, we only use the line version in this work. To better compare our transcription accuracy, we follow the data augmentation techniques used by [17]. The images are degraded using techniques described in [28] and split into 12 sets depending on the degradation parameters, namely, elastic elongation, jitter, sensitivity and threshold. Thereafter, the line images are divided in training (46%), validation (34%) and testing (20%) sets by evenly distributing the clean and degraded images.

We also release an IIIT-Urdu OCR dataset consisting of 2000 Urdu text line images along with their corresponding annotations. To incorporate maximum variance in terms of writing styles and fonts, as predominantly seen in Urdu publications, text pages from various Urdu books and magazines were scanned at a high-resolution (Fig. 5). Subsequently, bounding boxes around text lines were manually made and annotations for the same were provided by language experts. The dataset is being made publicly available for future researchers to compare the performance of their solutions against our benchmark.

Figure 5: Sample images from the scanned IIIT-Urdu OCR dataset we release. Various writing styles and fonts were incorporated to cater the diverse writing styles observed in Urdu publications.

B. Implementation Details

Although the convolutional block is inspired from the VGG-style architecture, minor modifications were made in the layers to fit an Urdu OCR setting better. Precisely, in the 3rd and 4th max-pooling layers, the pooling windows used are rectangular instead of the usual square windows used in VGG. This allows us to obtain feature maps which are wider and hence create longer feature sequences for the recurrent layers that follow. The images are horizontally flipped before feeding them to the convolutional layers since Urdu is read from right to left. To enable faster batch learning, all input images are re-sized to a fixed height and width. We observed that re-sizing all images to a fixed aspect ratio didn't affect the accuracy much however, it significantly reduced the time and GPU memory required for training our model. With a batch-size of 64, training reaches convergence in about 14 hours on a single Nvidia TitanX GPU occupying less than 3Gbs of memory when the images fed in are all of fixed dimensions. However, incorporating zero-padding and allowing variable length images shoots up the training time to 24 hours and occupies close to 8Gbs of space on the same TitanX GPU.

To address the problems of training such deep convolutional and recurrent layers, we used the batch normalization [29] technique. Adding two batch-norm layers after the 5th and 6th convolutional layers respectively, accelerated the training process greatly. The network is trained using stochastic gradient descent (SGD). Gradients are calculated by the back-propagation algorithm. Precisely, the transcription layers' error differentials are back-propagated with the forward-backward algorithm, as discussed in [13]. While in the recurrent layers, the Back-Propagation Through Time (BPTT) [30] algorithm is applied to calculate the error differentials. While doing the gradient descent, an adaptive learning rate method like ADADELTA [31] is used.

[1] IIIT-URDU OCR dataset : https://goo.gl/2G2oAS

IV. Results and Discussion

In this section we discuss the efficacy of the above architecture in recognizing printed Urdu text. We perform our experiments with the assumption that cropped line images are available, and not full page/text images. We compare the results of our hybrid CNN-RNN architecture against the previous state-of-the-art bi-directional LSTM network presented in [17]. The transcription accuracy is compared on the UPTI dataset [8].

Results for the Urdu OCR task are presented in Table I. The metric used to compare the performance of these solutions is CRR - Character Recognition Rate. In the below equation, RT and GT stand for recognized text and ground truth, respectively.

$$CRR = \frac{(nCharacters - \sum EditDistance(RT, GT))}{nCharacters}$$

Table I presents the variants for our solution architecture; HYBRID X-CNN-RNN, HYBRID X-CNN-RNN-FINE and HYBRID X-CNN-RNN-MIX, where X stands for the number of convolutional layers in the model. HYBRID X-CNN-RNN-FINE model was fine-tuned on the train set of UPTI dataset, while the HYBRID X-CNN-RNN models were not fine-tuned and neither did they see any images of the UPTI dataset whilst training. It should be noted that the HYBRID CNN-RNN architectures achieve higher transcription accuracy than the previous state-of-the-art even without fine-tuning.

We noticed that fine-tuning on the UPTI synthetic dataset reduces our model performance on URDU OCR DATASET. We trained the HYBRID X-CNN-RNN-MIX model by fine-tuning on an equal sampling of UPTI and URDU OCR DATASET and observed that the model performance remains similar on UPTI while improves on IIIT-Urdu OCR dataset as compared to it's FINE variant. This behaviour can be attributed to the higher diversity among fonts and styles in our scanned data as compared to UPTI's synthetic images.

Figure 6: Visualization of the robust convolutional features learnt by the HYBRID 7-CNN-RNN model. The partitions show some of the activations for a given input created by the model which can be interpreted as solving a particular type of detection task (edges, text-body, diacritics). Notice how going deeper in convolutions brings more insights into detection - Diacritics appearing below the text-body get separate filters, adding a sense of relative positioning.

Table I: Transcription accuracy for Urdu OCR

URDU OCR	UPTI DATASET	IIIT -URDU OCR DATASET
	CRR (%)	CRR (%)
BLSTM [17]	86.43	-
BLSTM [16]	93.38	-
CNN-MDLSTM [11]	98.12	-
HYBRID 4-CNN-RNN	73.10	70.38
HYBRID 5-CNN-RNN	91.60	81.89
HYBRID 6-CNN-RNN	91.72	81.94
HYBRID 7-CNN-RNN	92.04	89.84
HYBRID 7-CNN-RNN-FINE	98.80	78.97
HYBRID 7-CNN-RNN-MIX	97.81	82.45

The HYBRID CNN-RNN model outperforms BLSTM based method on the UPTI dataset without needing fine-tuning.

The trained model scaling well to a new dataset (UPTI), demonstrates the robustness of learnt convolutional features. A qualitative analysis of the model performance can be seen in Fig. 7. We can see that the model fails to accurately predict the diacritics in certain cases. Also, creating transcription for images containing English numerals is erroneous.

To get further insights into the working of the convolutional layers, we visualize the layer activations on passing an image through the trained network (Fig. 6). It is interesting to see that the model tried to learn the innate patterns in Urdu text. The first convolution layer learns to detect text-edges, text-body and diacritics in the text. As we go deeper in the convolutional layers, additional complex insights are brought into the detection process. The filters now differentiate between diacritics appearing above and below the main text-body.

V. Conclusion

We demonstrate the efficacy of newer script and language agnostic approaches for low resource languages like Urdu, for which traditional methods were often constrained by language specific modules. For cursive script languages, like Urdu, segmentation of individual characters/glyphs was challenging. CTC loss layer enabled segmentation-free transcription, and end-to-end training of the transcription module. Furthermore, Bi-directional RNN's made it possible to capture contexts in both the forward and backward directions. We have shown how state-of-the-art deep learning techniques can be successfully adapted to some rather challenging tasks like Urdu OCR. Additionally, we provided insights into the robust representations learnt by the convolutional layers through visualization of activations.

With the availability of better feature representations and learning algorithms, we believe that the focus of the vision community should now shift towards more complex cursive scripts which are generally also low on resources. Another possible direction to take this work forward would be the incorporation of Attention Modelling into text recognition, which has been proven quite effective for rather complicated tasks like object detection and captioning [32]. We hope that the introduction of a new IIIT-Urdu OCR dataset and our benchmark results would instill interest among the community to take this field of research further.

Acknowledgment

The authors would like to thank Suhana Suha and Silar Shaik for helping with the annotation for our Urdu OCR dataset. We also thank Himani, Kalpit, Somya, Virali and Ankita for their timely help and support.

References

[1] R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan, "Improvements in hidden markov model based arabic ocr," ICPR, 2008. 1

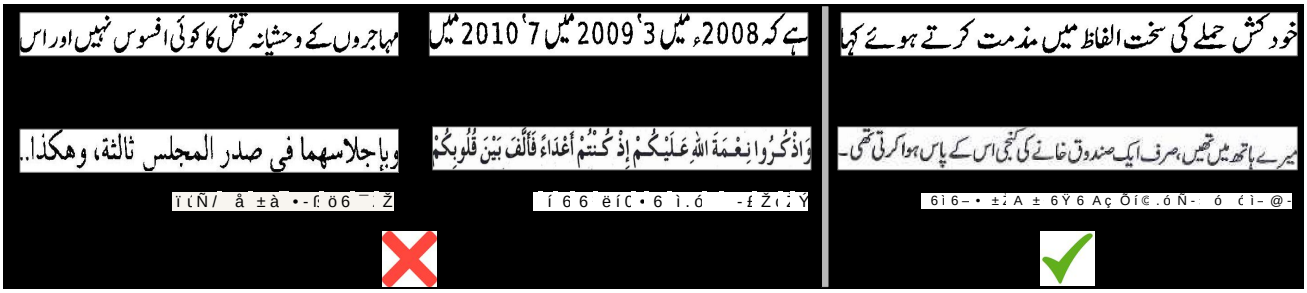

Figure 7: Qualitative analysis of the HYBRID 7-CNN-RNN model on the UPTI dataset (top) and the *IIIT-Urdu OCR dataset* (bottom). The printed text images are enclosed in gray rectangles with the transcription made by our model given below each image.

[2] M. Yousefi, M. Soheili, T. Breuel, and D. Stricker, "A comparison of 1d and 2d lstm architectures for the recognition of handwritten arabic," *SPIE/IS&T Electronic Imaging*, 2015. 1

[3] M. Jain, M. Mathew, and C. V. Jawahar, "Unconstrained scene text and video text recognition for arabic script," *ASAR Workshop*, 2017. 1

[4] M. Nagata, "Japanese ocr error correction using character shape similarity and statistical language model," *ACL*, 1998. 2

[5] F. Camastra, "A svm-based cursive character recognizer," *PR*, 2007. 2

[6] T. Nawaz, S. Naqvi, H. ur Rehman, and A. Faiz, "Optical character recognition system for urdu (naskh font) using pattern matching technique," *IJIP*, 2009. 2

[7] S. Sardar and A. Wahab, "Optical character recognition system for urdu," *ICIET*, 2010. 2

[8] N. Sabbour and F. Shafait, "A segmentation-free approach to arabic and urdu ocr," *DRR*, 2013. 2, 4

[9] S. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, "Segmentation free nastalique urdu ocr," *World Academy of Science, Engineering and Technology 46*, 2010. 2

[10] M. Akram and S. Hussain, "Word segmentation for urdu ocr system," *Proceedings of the 8th Workshop on Asian Language Resources*, 2010. 2

[11] S. Naz, A. I. Umar, R. Ahmad, I. Siddiqi, S. B. Ahmed, M. I. Razzak, and F. Shafait, "Urdu nastaliq recognition using convolutional–recursive deep learning," *Neurocomputing*, vol. 243, pp. 80–87, 2017. 2, 5

[12] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in neural information processing systems*, 2009, pp. 545–552. 2

[13] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *ICML*, 2006. 2, 3, 4

[14] M. Mathew, A. Singh, and C. V. Jawahar, "Multilingual ocr for indic scripts," *DAS*, 2016. 2

[15] N. Sankaran and C. V. Jawahar, "Recognition of printed devanagari text using blstm neural network," *ICPR*, 2012. 2

[16] S. Naz, S. B. Ahmed, R. Ahmad, and M. I. Razzak, "Zoning features and 2dlstm for urdu text-line recognition," *Procedia Computer Science*, vol. 96, pp. 16–22, 2016. 2, 5

[17] A. Ul-Hasan, S. Ahmed, F. Rashid, F. Shafait, and T. Breuel, "Offline printed urdu nastaleeq script recognition with bidirectional lstm networks," *ICDAR*, 2013. 2, 4, 5

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 1997. 2, 3

[19] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE TPAMI*, 2016. 2

[20] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *NN*, 1994. 3

[21] F. Gers, N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *JMLR*, 2002. 3

[22] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *ICASSP*, 2013. 3

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014. 3

[24] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," *NIPS*, 2012. 3

[25] R. Girshick, J. Donahue, T. Darrel, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CVPR*, 2014. 3

[26] M. Jaderberg, K. Simoyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *IJCV*, 2015. 3

[27] D. Arya, C. Jawahar, C. Bhagvati, T. Patnaik, B. Chaudhuri, G. Lehal, S. Chaudhury, and A. Ramakrishna, "Experiences of integration and performance testing of multilingual ocr for printed indian scripts," in *Proceedings of the 2011 joint workshop on multilingual OCR and analytics for noisy unstructured text data*. ACM, 2011, p. 9. 3

[28] H. Baird, H. Bunke, and K. Yamamoto, "Document image defect models," *Structured Document Image Analysis, Springer-Verlag*, 1992. 4

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ICML*, 2015. 4

[30] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE 78.10*, 1990. 4

[31] M. Zeiler, "Adadelta: an adaptive learning rate method," *CoRR*, 2012. 4

[32] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *ICML*, 2015. 5