An attention based method for offline handwritten Urdu text recognition

Tayaba Anjum, Nazar Khan
Punjab University College of Information Technology (PUCIT)
Lahore, Pakistan
Email: {phdcsf14m005, nazarkhan}@pucit.edu.pk

***Abstract*—Compared to derivatives from Latin script, recognition of derivatives from Arabic handwritten script is a complex task due to the presence of two-dimensional structure, context-dependent shape of characters, high number of ligatures, overlap of characters, and placement of diacritics. While significant attempts exist for Latin and Arabic scripts, very few attempts have been made for offline, handwritten, Urdu script.**

In this paper, we introduce a large, annotated dataset of handwritten Urdu sentences. We also present a methodology for the recognition of offline handwritten Urdu text lines. A deep learning based encoder/decoder framework with attention mechanism is used to handle two-dimensional text structure. While existing approaches report only character level accuracy, the proposed model improves on BLSTM-based state-of-the-art by a factor of 2 in terms of character level accuracy and by a factor of 37 in terms of word level accuracy. Incorporation of attention before a recurrent decoding framework helps the model in looking at appropriate locations before classifying the next character and therefore results in a higher word level accuracy.

***Index Terms*—Handwriting, Urdu, Offline, Recognition, Text, Attention, LSTM, GRU, CNN, DenseNet, Encoder/Decoder**

I. Introduction

Offline handwritten text recognition (OHTR) is the ability of computer to understand readable text from an image which is captured from paper documents, photographs, and screen shots. OHTR can play a vital role in automatic reading of bank checks, postal documents, and other forms. Furthermore, OHTR tools can be used as digital libraries by allowing the entry of image textual information into a computer by detection and recognition methods.

The problem of OHTR is different from traditional Optical Character Recognition (OCR) due to wide range of variations in pen type, writing style, writing size, page background, and rule violations. In handwritten text, two different writing styles are mostly used: cursive and non-cursive. Cursive text is difficult to segment due to inherent ambiguity. Intensive research has been done using various handwritten datasets likes MNIST [1], IAM [2], and CROHME [3]. Arabic script also attracted the attention of various researchers and various datasets are used for recognition like IFN/ENIT [4], CENPARMI [5], OpenHaRT [6], and KHATT [7]. As a research problem, the recognition of *handwritten* Arabic like scripts (Arabic, Urdu, and Farsi) is different from Latin script and exhibits various interesting challenges [8]. Figure 1 demonstrates the presence of two-dimensional structure, character/ligature overlap, context-sensitive shapes of characters, counts and placements of diacritical marks, arbitrary stretching of characters and spacing.

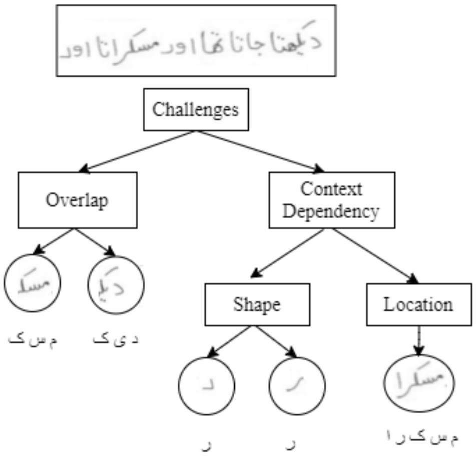

Figure 1: Character recognition challenges in Urdu text.

The most successful architectures for modelling sequential data such as languages are Recurrent Neural Networks (RNN) and their derivative Long Short-Term Memory (LSTM) networks which avoid vanishing as well as exploding gradients during training. Existing methods for recognizing Arabic like scripts employ Bidirectional LSTM (BLSTM) and Multidimensional LSTM (MDLSTM). These approaches use the complete image for prediction of a word at a certain time step. However, a single word can also be predicted by focusing on a certain part of the image instead of the whole image. Attention mechanisms [9], [10] can be used to extract information from the most relevant part of the image instead of the whole image. Additionally, for Arabic like text recognition, LSTM can be used instead of BLSTM and MDLSTM.

In this paper, we introduce an attention based encoder/decoder architecture for OHTR in Urdu text. The model learns to output text sequence from an input image. The model is composed of encoder and decoder. In the encoding phase, high level features are extracted from input image using a DenseNet. In the decoding phase, these high level features are decoded into an output sequence character by character. For prediction of a character, attention mechanism is used to focus on the most relevant part. Unlike previous approaches, proposed model uses single dimensional LSTM instead of BLSTM and MDLSTM. Furthermore, while decoding a character at a time step, only relevant part is focused instead of whole image. Moreover, the model is trained jointly for both encoder and decoder. The joint training not only helps the encoder to extract good features for the decoder to parse, the decoder also helps the encoder by providing contextual information and guides the attention. The paper contributes as follows:

1) We present the first attention-based, encoder-decoder model for recognition of offline handwritten Urdu text.
2) We present a new dataset of offline handwritten Urdu text containing $7,309$ unique text line images with ground-truth.
3) For the first time, we report word level accuracy instead of only character level accuracy.
4) We show that attention can enable uni-directional decoders to out-perform bidirectional decoders.
5) Compared to previous approaches, we report 2x accuracy improvement at character level and 37x at word level.

The new dataset can be accessed via http://faculty.pucit.edu.pk/nazarkhan/.

II. Literature Review

A wide range of literature exists on handwritten text recognition with various languages and data sets. The recognition techniques can be classified as segmentation based (implicit/explicit) and holistic based. In the explicit approaches, three major steps are performed: Over segmentation, grouping, and classification. These approaches are script dependent. Implicit segmentation approaches require images and their corresponding labels. A model is trained to automatically learn segmentation cue points without pre-segmented units of ligatures. Holistic approaches deal with the shape of ligature. The model learns the shape of ligature and recognizes it. Holistic approaches suffer from scalability issue as the shape of ligature may be large especially in handwritten text. Deep learning approaches with implicit segmentation represent the state-of-the-art [11], [12], [13].

In literature, recognition of individual handwritten character recognition has been studied by various researchers[14], [15] and was one of the early applications of neural networks [16]. According to an analysis [14], Convolution Neural Network (CNN) outperforms as compared to other methods on individual handwritten character recognition. Handwritten word/line recognition is most difficult. The problem is character recognition along with character detection by separating characters from their neighbours. In cursive handwritten text, segmentation based techniques cannot be used [11].

In the solution of cursive words recognition, CNN with gradient-based learning methods has been used as an initial step towards handwritten text recognition [17]. As CNN is not recurrent, so the sequence of recognizer outputs was further handled by Graph Transformer Networks (GTN). However, the use of GTN was quite expensive in the general case. Next major upgrade was the use of Hidden Markov Model (HMM) [18]. However, due to the Markovian assumption of only one step dependency, HMM lacks behind Long Term Short Memory (LSTM) in language processing tasks [19]. Grave *et al.* [11] worked on raw images of IFN/ENIT dataset and recognized handwritten Arabic words using MDLSTM and obtained 93.37% character level accuracy. The effectiveness of handcrafted features and automatically learned features has been compared using LSTM and concluded that LSTM outperforms on automatically learned features [20]. IFN/ENIT dataset was used and obtained 89% character level accuracy. For recognition of OpenHaRT dataset, [21] used MDLSTM and obtained 52% word-level accuracy. Arabic KHATT dataset has been recognized [12] using MDLSTM with 75% character level accuracy.

In the context of handwritten Urdu text recognition, there are three existing datasets. In [22], the CENIP-UCCP dataset is presented. The dataset is not publicly available. In [23], the UCOM/UNHD dataset is presented and a BLSTM is used for text recognition with 93.02% character level accuracy. However, data statistics are not reliably reported and only a small portion of the dataset is publicly available which is not enough for training deep networks like LSTM, BLSTM, and MDLSTM as the basic requirement of these networks is the availability of a considerable amount of data. In [13], a custom handwritten Urdu dataset is presented and a CNN is used along with a BLSTM to recognize Urdu text with 83.69% character level accuracy. This dataset is also not publicly available.

Attention-based approaches have been successful in machine translation, image captioning, and speech recognition. These approaches help models in learning correct alignment between input image pixels and corresponding target characters [24]. Attention helps to extract features from the most relevant part of the image [25]. An attention based MDLSTM architecture [26] has been used to recognize text paragraphs of IAM dataset without explicit segmentation into lines or words. For the text recognition from words to lines, a method has been proposed [27] using context-aware reinforcement learning on KHATT, IAM, and RIMES dataset. Feature extraction was done using CNN and feature decoding was done using context-aware LSTM. Attention mechanism has also been used to recognize mathematical expressions [28] which share the two-dimensional structure of Urdu words.

III. Methodology

Images of handwritten text can have varying sizes. The text itself is a variable-length sequence of characters that can be recognized by an encoder-decoder model [24]. The encoder transforms the input image into an intermediate representation. The decoder then decodes this intermediate representation into a sequence of individual characters. Attention mechanism helps the decoder to focus on specific parts of the image in order to produce different characters. The model takes an arbitrary sized raw image as input and outputs corresponding Urdu text sequence

$$Y = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_c\} \tag{1}$$

where $\mathbf{y}_i \in \mathbb{R}^k$ is the 1-of-k encoding of the i-th character in the image text. Variable c represents the total number of characters in the image text. This includes any blank spaces and/or punctuation marks considered part of the alphabet. The number of unique characters in the Urdu alphabet is represented by k.

A. Encoder: DenseNet with bottlenecks

In order to extract useful features from input images, previous attempts at Urdu OHTR have mostly employed a traditional CNN encoder. We replace it by the DenseNet architecture that consists of *dense blocks*. Convolution in a dense block of layers is applied to all previous layers in the block instead of just the previous layer. This enables learning of a more diverse set of features and also helps to maintain the flow of computations during both forward and backward propagations. However, connecting to all previous layers is computationally expensive. A cheaper alternative is to insert bottleneck layers (1×1 convolutions) before the regular convolution layers [29]. This compresses the volume of all previous layers to a shallower depth. The convolutional output volume of layer l is calculated as

$$\mathbf{F}_l = \mathcal{C}_l(\mathcal{B}_l([\mathbf{F}_1; \mathbf{F}_2; \ldots; \mathbf{F}_{l-1}])) \tag{2}$$

where $[\mathbf{F}_1; \mathbf{F}_2; \ldots; \mathbf{F}_{l-1}]$ is the output of all previous layers concatenated in depth, $\mathcal{B}_l(\cdot)$ is the 1×1 convolution of a bottleneck layer and $\mathcal{C}_l(\cdot)$ is a regular convolution. For extraction of features at multiple scales, pooling layers are placed between dense blocks. Our encoder architecture is illustrated in Figure 2.

For a given input image, the output of a DenseNet encoder is a volume of size $h \times w \times d$. That is, we can consider $h \times w$ overlapping blocks of the input image to now be represented by d-dimensional *annotation vectors*. The set of all annotation vectors produced by the DenseNet is represented by

$$A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_s\} \tag{3}$$

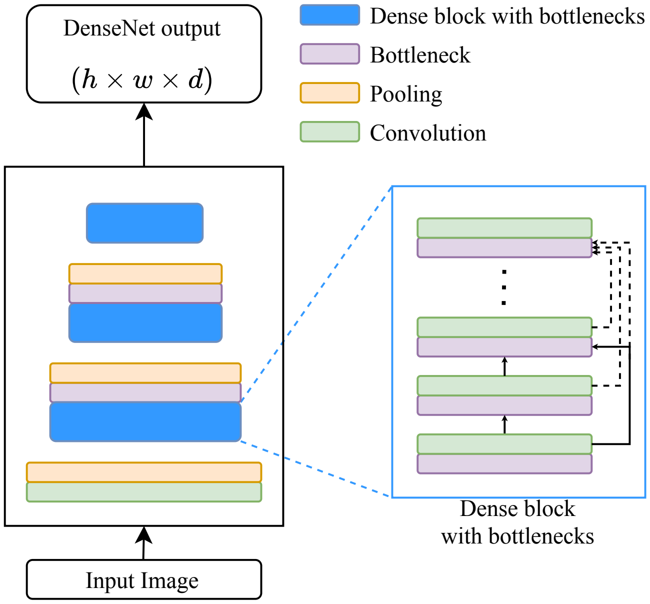

Figure 2: Architecture of DenseNet encoder.

where $\mathbf{a}_i \in \mathbb{R}^d$ and $s = hw$.

B. Decoder: Gated Recurrent Units

When an encoder extracts high-level visual features from variable-sized input images, we can use a GRU [30] to generate a text sequence character by character, conditioned on the previous output character $\mathbf{y}_{t-1}$, the current hidden state $\mathbf{h}_t$ and a context vector $\mathbf{c}_t$.

The current output and hidden state $\mathbf{h}_t$ of a GRU can be computed via the following recurrent equations

$$\mathbf{z}_t = \sigma(\mathbf{W}_{yz}\mathbf{E}\mathbf{y}_{t-1} + \mathbf{U}_{hz}\mathbf{h}_{t-1} + \mathbf{C}_{cz}\mathbf{c}_t) \tag{4}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_{yr}\mathbf{E}\mathbf{y}_{t-1} + \mathbf{U}_{hr}\mathbf{h}_{t-1} + \mathbf{C}_{cr}\mathbf{c}_t) \tag{5}$$

$$\widetilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{yh}\mathbf{E}\mathbf{y}_{t-1} + \mathbf{U}_{rh}(\mathbf{r}_t \otimes \mathbf{h}_{t-1}) + \mathbf{C}_{ch}\mathbf{c}_t) \tag{6}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \otimes \mathbf{h}_{t-1} + \mathbf{z}_t \otimes \widetilde{\mathbf{h}}_t \tag{7}$$

where σ is the logistic sigmoid function, $\otimes$ is a point wise multiplication, $\mathbf{E} \in \mathbb{R}^{m \times k}$ is a learnable lower-dimensional projection matrix for getting rid of the wasteful 1-of-k coding of character vector $\mathbf{y}_{t-1}$ and all other matrices within each equation represent learnable transformations into a constant dimensional space. The matrices can be divided into character embedding transformations $\{\mathbf{W}_{yz}, \mathbf{W}_{yr}, \mathbf{W}_{yh}\} \in \mathbb{R}^{h \times m}$, recurrent transformations $\{\mathbf{U}_{hz}, \mathbf{U}_{hr}, \mathbf{U}_{rh}\} \in \mathbb{R}^{h \times h}$, and context transformations $\{\mathbf{C}_{cz}, \mathbf{C}_{cr}, \mathbf{C}_{ch}\} \in \mathbb{R}^{h \times d}$.

The context vector $\mathbf{c}_t$ is a dynamic representation of the relevant part of the image at time t. It is computed as a weighted sum of the annotation vectors as

$$\mathbf{c}_t = \sum_{i=1}^{s} \alpha_{ti}\mathbf{a}_i \tag{8}$$

where weights α_{ti} determine the parts of an image to focus on at time t. Attention weight α_{ti} depends on the hidden state $\mathbf{h}_{t-1}$, the annotation vector $\mathbf{a}_i$ and a *coverage vector* $\mathbf{f}_i$ that represents a history of the attention already given to the image region corresponding to annotation $\mathbf{a}_i$. The weights are computed as

$$e_{ti} = \mathbf{v}_a^T \tanh(\mathbf{W}_a \mathbf{h}_{t-1} + \mathbf{U}_a \mathbf{a}_i + \mathbf{U}_f \mathbf{f}_i) \tag{9}$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^{s} \exp(e_{tj})} \tag{10}$$

where $\mathbf{v}_a \in \mathbb{R}^n$, $\mathbf{W}_a \in \mathbb{R}^{n \times h}$, $\mathbf{U}_a \in \mathbb{R}^{n \times d}$, and $\mathbf{U}_f \in \mathbb{R}^{n \times q}$ are learnable projection parameters.

Coverage vectors are computed as follows. All previous attention weights at each location are summed in time to obtain an array of size $h \times w$. To smooth out attention histories of adjacent locations, this array is convolved with a learnable, filter of size $f \times f \times q$ to obtain a volume of size $h \times w \times q$. Similar to how annotation vector $\mathbf{a}_i$ is formed, the coverage vector $\mathbf{f}_i$ is the q-dimensional vector at location i of this volume.

The conditional probabilities of the next character are computed as

$$p(\mathbf{y}_t | \mathbf{y}_{t-1}; \mathbf{X}) = \operatorname{softmax}(\mathbf{W}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{W}_h \mathbf{h}_t + \mathbf{W}_c \mathbf{c}_t)) \tag{11}$$

where $\mathbf{W}_o \in \mathbb{R}^{k \times m}$, $\mathbf{W}_h \in \mathbb{R}^{m \times h}$ and $\mathbf{W}_c \in \mathbb{R}^{m \times d}$ are all learnable projection parameters. The optimal sequence of characters $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_c$ is found via beam search [31] using a beam width of 10.

IV. Experiments

A. Dataset

An offline Urdu handwritten dataset is developed. The dataset consists of multiple writing styles, pen types, ink types, text sizes, and background colours. The text is collected from different sources. A hundred undergraduate students between the ages 20-24 years were requested to submit a handwritten Urdu text. Participants were not restricted to use any particular pen, page type, and ink colour. The selection of Urdu text was also not forced. In total, 474 pages containing Urdu handwritten text were submitted. The data was divided randomly into train and test sets. The train set contains data of 88 writers, and the test set contains data of 12 writers. Initially, data was in the form of page images. These images were manually segmented into lines. The 418 pages of the train set were segmented into $6,489$ lines and 56 pages of the test set were segmented into 820 lines. The ground truth for the data was constructed in text format. Table I compares our dataset with existing datasets[1].

B. Experimental Setup

In this paper, the proposed model is trained jointly for recognition of offline handwritten Urdu text. The model predicts output character from the given image based on the previously predicted character and current input image as shown in. The objective of the training is to maximize the probability of the predicted character. The cross-entropy function is used as an objective function. The architecture of the model is based on three components: DenseNet for feature extraction, attention mechanism to focus on a certain part, and GRU for feature decoding. DenseNet contains 3 blocks, each block contains 32 sub-layers, 16 bottleneck layers performing 1×1 convolutions and 16 regular 3×3 convolution layers. Three pooling layers are used in DenseNet with kernel size and stride both set to 2×2. Sizes of the learnable matrices and vectors to produce GRU decoder output and context and coverage vectors are determined by the values $k = 98, h = 256, m = 256, n = 256, d = 684, f = 11$ and $q = 512$. We use batch normalization and dropout to reduce over fitting. The dropout is performed at convolution layers with 20% drop ratio. For better optimization Adadelta algorithm with gradient clipping is used. We resized all training and testing images to 100×800 pixels.

Table I: Comparison with existing offline handwritten Urdu text datasets

Dataset	**Total Lines**	**Total Words**	**Total characters**	**Total Writers**	**Vocabulary Size**
UCOM/UNHD[23]	10,000	312,000	1,872,000	500	59
CENIP-UCCP[22]	2,051	23,833	-	200	-
Custom dataset [13]	6,000	86,400	432,000	600	-
Proposed dataset	**7,309**	**78,870**	**283,664**	**100**	**98**

C. Evaluation Metric

At the level of characters, character error rate (CER) between output and target text is defined as

$$\mathrm{CER} = \frac{\mathrm{ins} + \mathrm{sub} + \mathrm{del}}{\mathrm{n}} \times 100 \tag{12}$$

where ins, sub and del are the numbers of insertions, substitutions and deletions required to transform the target into the output and n is the number of characters in the target text[2]. By replacing characters with words, Equation (12) can be used to compute word error rate (WER) between output and target sentences as well. Both error rates can be converted to accuracies by subtracting from 100.

D. Attention Visualization

In this section, we show how the attention helps to overcome the challenges faced in Urdu character recognition. The model we use automatically segments different characters while resolving the challenges from Figure 1 such as 2-dimensional structure, inter/intra-ligature overlap and context-dependent shapes. While it is non-trivial to segment positions and recognize characters in 2-dimensional, overlapping contexts, Figure 3 demonstrates that our proposed model recognizes character alignments strongly corresponding with human intuition.

[1] The statistics in [23] are not reliably reported. We have included it for completeness.

[2] Theoretically, CER can be greater than 100 but for any reasonably performing recognizer, it is typically less than 100.

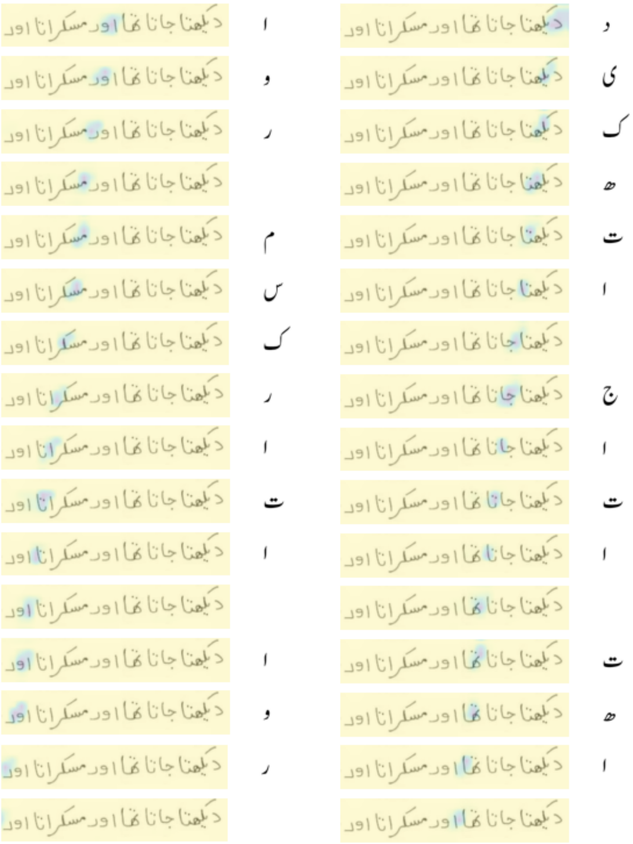

Figure 3: Attention visualization: our model implicitly segments and recognizes every character by focusing only on localized, relevant areas.

E. Results

The proposed model is trained and tested on our proposed offline handwritten Urdu dataset (Section IV-A). Figure 4 shows the recognition examples with some invalid character recognition. Dashed green boxes in ground truth represent those characters that are missing in the output. Orange boxes in output represent incorrectly recognized characters.

The comparison of our proposed method with other models is listed in Table II. The CNN model for generating all results was based on the architecture from [13] and each method was trained on our dataset. From all results, it can be observed that incorporation of the DenseNet encoder with attention-based decoding almost doubles the character level accuracy and increases word level accuracy by a factor of more than 30.

Table II: Comparison of different models for offline handwritten Urdu text recognition on our proposed dataset.

Models	**Character Level Accuracy**	**Word Level Accuracy**
CNN+GRU	39.79	1.24
CNN+BGRU	41.45	1.41
CNN+LSTM	39.26	1.40
CNN+BLSTM[13]	40.90	1.17
DenseNet+GRU+ Attention	**77.05**	**43.35**

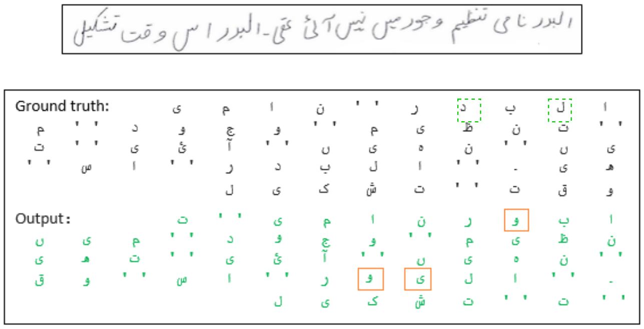

Figure 4: Sample image with five mistakes. Missed characters from the ground-truth are enclosed in green boxes with dashed outline. False detections in output are enclosed in orange boxes with solid outline. All other characters in output are correctly recognized.

This represents a significant improvement in the development of offline, handwritten Urdu recognizers since word level accuracy is the more important evaluation criterion. Incorporation of attention before a recurrent decoding framework helps the model in looking at appropriate locations before classifying the next character. This greatly benefits word level accuracy.

V. Conclusion

We have proposed an encoder/decoder model that uses attention to recognize offline handwritten Urdu text. We have demonstrated that learned attention enhances the interpretability of models. Our model recognizes text by focusing on localized and relevant image regions that correspond to human intuition. We have also collected a new dataset of 7,309 offline handwritten Urdu text lines consisting of 78,870 words written by 100 different writers. Our dataset contains 98 unique Urdu characters written in unrestricted settings. On our dataset, the proposed model improves on BLSTM-based state-of-the-art by a factor of 2 in terms of character level accuracy and by a factor of 37 in terms of word level accuracy. The increase in word level accuracy is due to i) employing a DenseNet based encoder that can learn more diverse features than traditional CNN encoders and ii) incorporation of attention-mechanism in a recurrent decoding framework. It helps the model to focus on appropriate image regions before classifying the next character.

Acknowledgments. This work was supported by the Higher Education Commission (Pakistan) under Grant 8329/Punjab/NRPU/R&D/HEC/2017.

References

[1] Y. LeCun, C. Cortes, and C. J. Burges. (1998) The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Accessed: 19.07.2019.

[2] U. Marti and H. Bunke., "The IAM-database: An English sentence database for off-line handwriting recognition ," *Int'l Journal on Document Analysis and Recognition*, vol. 5, no. 10, pp. 39 – 46, 2002.

[3] H. Mouchere, C. Viard-Gaudin, R. Zanibi, D. H. Kim, J. H. Kim, and U. Garain, "ICDAR 2013 CROHME: Third international competition on recognition of online handwritten Mathematical expressions," in *Proc. ICDAR*, 2013.

[4] H. E. Abed and V. Margner, "The IFN/ENIT-database - a tool to develop Arabic handwriting recognition systems," *Int'l Journal on Document Analysis and Recognition*, vol. 5, no. 10, pp. 39 – 46, 2002.

[5] H. Alamri, J. Sadri, C. Y. Suen, and N. Nobile, "A novel comprehensive database for Arabic offline handwriting Recognition ," *Journal Pattern Recognition*, vol. 60, pp. 378–393, 2016.

[6] A. Tong, M. Przybocki, V. Margner, and H. E. Abed, "NIST 2013 open handwriting recognition and translation evaluation," in *Proc. NIST*, 2013.

[7] S. A. Mahmoud, I. Ahmad, W. G. Al-Khatib, M. Alshayeb, M. T. Parvez, V. Margner, and G. A. Fink, "KHATT: An open Arabic offline handwritten text database," *Pattern Recognition*, vol. 47, no. 3, pp. 1096–1112, 2014.

[8] S. Naz, A. I. Umar, S. H. Shirazi, S. B. Ahmed, M. I. Razzak, and I. Siddiqi, "Segmentation techniques for recognition of Arabic-like scripts: A comprehensive survey," *Educ Inf Technol*, vol. 20, no. 1, 2015.

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[10] K. Xu, J. L. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: neural image caption generation with visual attention," in *International conference on machine learning*, 2015.

[11] G. A. and S. J., "Offline handwriting recognition with multidimensional recurrent neural networks," *In Advances in Neural Information Processing Systems (NIPS)*, vol. 21, pp. 5454–552, 2009.

[12] R. Ahmad, S. Naz, M. Z. Afzal, S. F. Rashid, M. Liwicki, and A. Dengel, "KHATT: A deep learning benchmark on Arabic script," in *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2017.

[13] I.-A. M. A. . S. I. Hassan, S., "Cursive Handwritten Text Recognition using Bi-Directional LSTMs: A Case Study on Urdu Handwriting," in *International Conference on Deep Learning and Machine Learning in Emerging Applications, Deep-ML 2019*, 2019.

[14] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *Intl. Conf. Document Analysis and Recognition*, pp. 958–962, 2003.

[15] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep big simple neural nets excel on handwritten digit recognition," *Neural Computation*, vol. 22, 2010.

[16] S. Marinai, M. Gori, and G. Soda, "Artificial neural networks for document analysis and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 23–35, 2005.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Intelligent Signal Processing*, pp. 306–351, 1998.

[18] J. Hu, S. G. Lim, and M. K. Brown., "Writer independent on-line handwriting recognition using an hmm approach," *Pattern Recognition*, vol. 33, pp. 133–147, 2000.

[19] F. Gers and E. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *Neural Networks IEEE Transactions*, vol. 12, no. 6, pp. 1333–1340, 2001.

[20] Y. Chherawala, P. Roy, and M. Cheriet, "Feature design for offline Arabic handwriting recognition: handcrafted vs automated?" in *12th International Conference on Document Analysis and Recognition*. IEEE, 2013.

[21] O. Morillot, C. Oprean, L. Likforman-Sulem, C. Mokbel, and E. Chammas, "The UOB-telecom Paris tech Arabic handwriting recognition and translation systems for the OpenHart 2013 competition," in *12th International Conference on Document Analysis and Recognition (ICDAR)*. NIST, 2013.

[22] A. Raza, I. Siddiqi, A. Abidi, and F. Arif, "An unconstrained benchmark urdu handwritten sentence database with automatic line segmentation," in *2012 International Conference on Frontiers in Handwriting Recognition*, 2012, pp. 491–496.

[23] S. Ahmed, S. Naz, S. Swati, and M. Razzak, "Handwritten Urdu character recognition using one-dimensional BLSTM classifier," *Neural Computing and Applications Systems (NIPS)*, pp. 1–9, 2017.

[24] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *ArXiv e-prints*, 2014.

[25] K. Cho, A. Courville, and Y. Bengio, "Describing multimedia content using attention based encoder decoder networks," *ArXiv e-prints*, 2015.

[26] T. Bluche, J. Louradour, and R. Messina, "Scan, attend and read: End-to-end handwritten paragraph recognition with MDLSTM attention," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1050–1055.

[27] Gui, Liangke, X. Liang, X. Chang, and A. G. Hauptmann, "Adaptive context-aware reinforced agent for handwritten text recognition," in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[28] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, vol. 71, pp. 196–206, 2017.

[29] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," in *2018 24th international conference on pattern recognition (ICPR)*. IEEE, 2018, pp. 2245–2250.

[30] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint*, 2014.

[31] K. Hwang and W. Sung, "Character-level incremental speech recognition with recurrent neural networks," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5335–5339.