Neural Computing and Applications
https://doi.org/10.1007/s00521-021-06498-2

ORIGINAL ARTICLE

A convolutional recursive deep architecture for unconstrained Urdu handwriting recognition

Noor ul Sehr Zia[1] · Muhammad Ferjad Naeem[1] · Syed Muhammad Kumail Raza[1] · Muhammad Mubasher Khan[1] · Adnan Ul-Hasan[2] · Faisal Shafait[1,2]

Received: 31 March 2019 / Accepted: 30 August 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

An offline handwriting recognition system for Urdu, a language with a user base of 200 Million and written in Nastaleeq script, has been a challenge for the research community. The key problems include recognition of complex ligature shapes and lack of publicly available datasets. This paper addresses both these problems by (i) proposing an end-to-end handwriting recognition system based on a new CNN-RNN architecture with n-gram language modeling, and (ii) presenting a new unconstrained dataset called NUST-UHWR. We compiled the first unconstrained Urdu handwritten data from around 1000 people from diverse background, age and gender population. The text in this dataset is selected carefully from seven different fields to ensure the presence of commonly used words in different domains. The model architecture is capable of incorporating fine-grained features necessary for handwritten text recognition of complex ligature languages. Our method addresses the limitations of existing architectures and provides state-of-the-art performance on Urdu handwritten text. We achieve a minimum character error rate of 5.28% on Urdu handwriting recognition (UHWR) and establish a state-of-the-art. The paper further demonstrates the generalization ability of the proposed model by training on English language and bilingual (Urdu and English) handwritten data.

Keywords Deep learning · Computer vision · Optical character recognition · Urdu · Nastaleeq · CNN · RNN · Handwriting recognition

1 Introduction

Urdu has the second largest user base in the Indian subcontinent region. Urdu language has 45 characters that join together to form different ligatures depending on their position in a sentence [1]. An estimate puts the unique ligatures in Urdu to be above 26,000 [2]. Other complexities in the script include absence of word spacing [3], no standardized baseline, diagonal writing style, contextual shape change of letters [4] and different writer styles depending on the writer and the medium. The complex nature of Urdu printed text in Nastaleeq script has remained a major challenge in developing a robust Urdu OCR system. The complexity of handwritten text is much more than the printed text; for languages like Urdu where printed OCR is challenging, the added complexities of handwritten text increase the challenges for automated recognition multifolds. Conventional pattern recognition techniques fail to capture the script's intricacies, thereby resulting in unavailability of practical solutions.

There is an increased focus to digitize content in the region; however, handwriting is still the primary mean of information distribution in segments such as government, retail and education. Moreover, there is a plethora of historic literature that is kept in hard copy for centuries. This vast amount of information needs to be connected to the

Noor ul Sehr Zia, Muhammad Ferjad Naeem and Syed Muhammad Kumail Raza have contributed equally to this work.

This work was partially funded by HEC Pakistan's grant for National Center of Artificial Intelligence (NCAI).

✉ Adnan Ul-Hasan
adnan.ulhassan@seecs.edu.pk

[1] School of Electrical Engineering and Computer Science, National University of Sciences and Technology (NUST), Islamabad, Pakistan

[2] Deep Learning Laboratory, National Center of Artificial Intelligence, Islamabad, Pakistan

Published online: 14 September 2021

digital world through an automated system. Historic literature is under the risk of extinction if not digitized due to aging of the scripts it was written on. There have been efforts at government level in Pakistan to digitize this content through manual transcription; however, transcribing this volume of data manually is laborious, time consuming, and expensive.

Arabic, the sister language of Urdu, has attracted more interest from research and industry alike with many commercial systems supporting printed Arabic text. Arabic handwriting has also received major attention with accuracy figures comparing to Latin scripts. Urdu has, however, only recently gained attention from the computer vision community. There is only one commercial OCR solution [5] for printed Urdu text, and there is no commercial system for Urdu handwriting. In the research community, printed Urdu text has gained major attention with Naeem et al. [6] reporting one of the first results of Urdu OCR on real-world data. Urdu handwriting is, however, still in infancy with only one public dataset [7]. This dataset was collected in standardized environment with writers instructed to follow a baseline and hence does not depict performance of the model in the real world. We address this issue by collecting our own Urdu handwriting dataset.

The main contributions of our work are:

1. We collected the first unconstrained dataset for Urdu handwriting that covers more than 73% of Urdu ligatures.
2. We developed a model based on convolutional neural networks (CNN) [8] and long short-term memory networks (LSTM) [9] to achieve state-of-the-art results for Urdu handwriting recognition.
3. We show the generalization ability of our model by testing it with IAM handwriting database and bilingual dataset consisting of Urdu and English.

This paper is further divided in five sections. Section 2 discusses the related works. Section 3 discusses the datasets used in this paper. Section 4 discusses the design cycle of our presented model. Section 5 describes our experimental setup, Sect. 6 presents the results and Sect. 7 concludes the study.

2 Related works

Research in Urdu character recognition is focused on printed Urdu text. Pal and Sarkar [10] proposed one of the first works on printed Urdu script. They proposed a system for individual character recognition based solely on image processing techniques in feature extraction, segmentation and recognition. They reported a 97.8% character level accuracy on isolated characters. Several segmentation free approaches have been proposed that rely on Hidden Markov Models (HMM) [11]. Ud Din et al. [12] proposed a technique using statistical features and HMM for Urdu ligature recognition. The authors in [13] presented a segmentation free approach using context shape matching techniques for Urdu and Arabic OCR. They generated the Urdu Printed Text Image (UPTI) database and found that the system's accuracy is comparable to Arabic OCR as well as Google's Tesseract [14]. Sardar and Wahab [15] presented an OCR system that was independent of fonts and scripts. Ul-Hasan et al. [1] and Ahmed et al. [16] used deep learning approaches for Nastaleeq script recognition. They used bidirectional LSTMs followed by Connectionist Temporal Classification (CTC) layer. Ul-Hasan [4] proposed Hierarchical Sub-sampling LSTM (HSLSTM) networks and reported 2.55% error rate on UPTI dataset. It was demonstrated that HSLSTMs were more efficient and accurate than multi-dimensional LSTM (MDLSTM) networks. Naz et al. [17] used MDLSTMs with statistical features for Urdu Nastaleeq text recognition and achieved an error rate of 5.03%. They later demonstrated that using MDLSTMs with raw pixels for automated feature extraction outperforms manual feature extraction and achieves 50% reduction in error rate [18]. The system presented in [19] uses overlapped windows for extracting statistical features and achieved 3.6% error rate. A hybrid approach consisting of CNN for low level feature extraction followed by MDLSTMs for learning higher level features and classification was presented in [20]. The authors reported an error rate of 1.88%.

Handwritten character recognition has been a great challenge for researchers in the domain of document analysis and recognition. Segmentation of words/ligatures is a challenging task in handwritten script recognition due to cursive and overlapping features of characters. Moreover, complex and continuous ligatures make it hard for the recognition model to classify individual characters. Handwritten script recognition can be broadly divided into online and offline recognition. Offline character recognition is a more complex problem compared to online setting [21]. Graves et al. [22] proposed a system consisting of RNN layers followed by CTC [23] layer for scenarios where it is difficult to segment data. Presented system had a minimum error rate of 11.5% for online data and 18.2% for offline IAM dataset.

For offline handwriting recognition, current state-of-the-art approaches rely on multidimensional recurrent neural networks. Graves and Schmidhuber [24] proposed an offline handwriting recognition system using multidimensional LSTMs (MDLSTMs) and CTC. That system achieved an error rate of 8.6% on 2007's ICDAR Arabic Handwriting Recognition competition. Pal et al. [25] used different classifiers and presented comparison of results of handwriting recognition for multiple Indic scripts. Messina et al. [26] presented a segmentation free approach for Chinese handwriting recognition using MDLSTMs and reported a 16.5% character error rate. They used character level language models that reduced the CER to 10.6%. Authors in [27] use four-layer bidirectional gated recurrent unit (GRU) network for offline Arabic handwriting recognition. They show that the model has greater capacity of generalization than the conventional three layer LSTM approach. Bluche et al. [28] proposed a method for automatic transcription of handwritten text without prior segmentation based on attention models and MDLSTM. Wu et al. [29] replaced traditional MDLSTM-RNN with separable MDLSTM-RNN that uses less computation and resources compared to the traditional MDLSTM-RNN.

Recent work in handwriting recognition focuses on using Hybrid models [30–33]. Adak et al. [32] uses a Lenet-5 CNN architecture followed by RNN and CTC layer for handwritten Bengali word recognition. The authors in [33] use CNN to generate attribute embedding of words followed by BLSTM with CTC layer to get the output. Several approaches have been presented for Devanagari handwriting recognition based on SVM classifiers. Shaw et al. [34] used contour and skeleton based feature representations with multiclass SVM classifier for Devanagari handwriting recognition. Another approach in [35] uses a combination of Gradient, Structural and Concavity (GSC) features and Directional Distance Distribution (DDD) features with multiclass SVM classifier. A hybrid model consisting of CNN followed by BLSTM was proposed in [36] for online recognition of Devanagari and Bangla Handwriting. Dutta et al. [37] presented a CNN-RNN hybrid architecture and benchmarked it on IIIT-HW-Dev dataset as well as their own dataset and achieved state-of-the-art results. A bidirectional LSTM model followed by CTC layer is presented by Chakraborty [38] for online Bangla recognition. Recently, combining classifiers with adaptive boosting and bootstrap aggregating has been proposed for medieval handwritten Gurmukhi character recognition [39].

Transcribed and unconstrained datasets are a prerequisite for training machine learning algorithms for Urdu handwriting recognition. There are only a handful of datasets available for Urdu script. UCOM [7] is a publicly available Urdu handwritten text dataset. UCOM dataset has 600 pages written by 100 individuals and total number of text lines is 4,800, whereas unique text lines are only 48 and unique ligatures are 321. These lines and ligatures are not sufficient for the development of a robust handwriting recognition system as Urdu has more than 26,000 ligatures [2]. Authors of UCOM trained a recurrent neural network-based system on 50 text lines and tested against 20 text lines. They used edit distance for error evaluation and reported an error rate of $0.04 \sim 0.06\%$. Raza et al. [40] presented a database for Handwritten Urdu sentences that consists of 2001 text lines produced by 200 writers and 400 filled forms. The data was generated using 66 unique forms generated from 6 domains. Malik and Khan [41] developed an online handwriting recognition system for Urdu language. Their system extracted different features including hat features, slope and writing direction. A hierarchical database of Urdu characters is maintained with respect to character structure type. Structure of an incoming character is compared against the stored characters in order to classify the character. The system reported 7% error rate.

We identify one constraint and one practical problem with the current state of Urdu handwriting recognition. Urdu has more than 26,000 ligatures [2], the datasets available only contain a small subset of the available ligatures and do not contain a generalized representation of its handwritten script. Hence, the constraint is a lack of comprehensive dataset. The practical problem is the complexity of the language. As the number of complex, continuous and cursive ligatures in Urdu is large, the conventional deep learning recognition architectures such as CNN-RNN hybrids do not have the capacity to account for all of them [36]. Needless to say that equally large amount of data is required to train a model on these complex features. We notice that language modeling improves the OCR accuracy significantly [42, 43] and is a useful addition to handwriting recognition systems. In this paper, we analyze and address both these problems and incorporate language modeling to improve the recognition accuracy.

3 NUST Urdu handwriting dataset (NUST-UHWR)

Our approach towards Urdu handwritten text recognition was to first develop a comprehensive dataset called NUST-UHWR. Dataset preparation was divided into data analysis and database preparation phases, which are described below.

3.1 Data analysis

Data is collected from seven different domains: Columns, News, Urdu Literature, Science & Technology, Religious, Sports and Finance, to capture a significant number of ligatures of Urdu. Data is downloaded from different websites consisting of news and social media sites. The text containing words from other languages e.g. English, Punjabi and Arabic, is removed based on unicode. Three parameters were analyzed on the dataset.

1. The total number of words
2. The total number of unique words
3. The total number of unique ligatures

Urdu alphabets have different joining rules with respect to their position in the text [44], which makes analyzing ligatures taxing and complicated. A rule-based approach was adopted to extract ligatures. Unicode of alphabet and their positioning information was used to classify them into ligatures. Researchers at Center for Language Engineering (CLE) have extracted 2,430 most frequently used ligatures from wide range of domains [45]. All of these ligatures are present in our extracted ligatures. They published around 18,000 unique ligatures for Urdu [45], and more than 73% of these ligatures are present in our dataset.

3.2 Database preparation

Dataset creation process is summarized in Fig. 1. The dataset was created by a collective effort of various institutes around Pakistan. A diverse group of 1,000 writers was reached, belonging to different ethnic and educational backgrounds, age groups and gender. The writers were allowed to write in an unconstrained environment to model writings from real-world scenarios. The writers used different mediums and markers with different line widths. For distinguishing purposes we are calling these handwritten documents "forms".

The forms received had several issues in them. Some words were missing from the start or end of the lines, some lines were interchanged while writing and several words were written incorrectly as shown in Fig. 2. These issues were resolved by manual verification. After scrutiny of the forms, correct ones were scanned with 300 dpi scanner and labeled manually. Scanned forms had some skewness due to human error. To detect and correct the skewness, a projection profile method was used to obtain skew angle and the forms were de-skewed accordingly [46].

Individual lines were extracted from the scanned forms with line segmentation algorithms. Projection profile-based line segmentation methods could not be applied due to variable gaps between the lines (primarily due to superscripts and subscripts). This results in over-segmentation. Overlapping and touching ligatures cause under-segmentation as shown in Fig. 3. The problem of over-segmentation was solved by using a hybrid approach of estimating row height and then using edit distance to merge smaller rows [47]. The under-segmentation was manually corrected. Table 1 presents the database statistics after correction and verification of image labels.

4 Model architecture

4.1 Convolutional recurrent neural network (CRNN)

Handwriting recognition is treated as a sequence to sequence mapping problem [22]. Convolutional recurrent neural network (CRNN) [48] proposed by Shi et al. has been the go to choice for character recognition of complex scripts. CRNN is composed of three main parts [48]. A CNN-based feature extractor [8], an LSTM based sequence labelling component [9] and CTC layer for transcription [23]. The feature extractor consists of 7 layers of CNN with max pooling and batch normalization layers in between. The images are fed to the network at a height of 32 pixels. The aspect ratio is maintained based on the use case. The model is designed to reduce the 3D feature map to a 2D sequence by max pooling the height dimension to 1. This approach results in several limitations that were partially acknowledged in the original paper: the

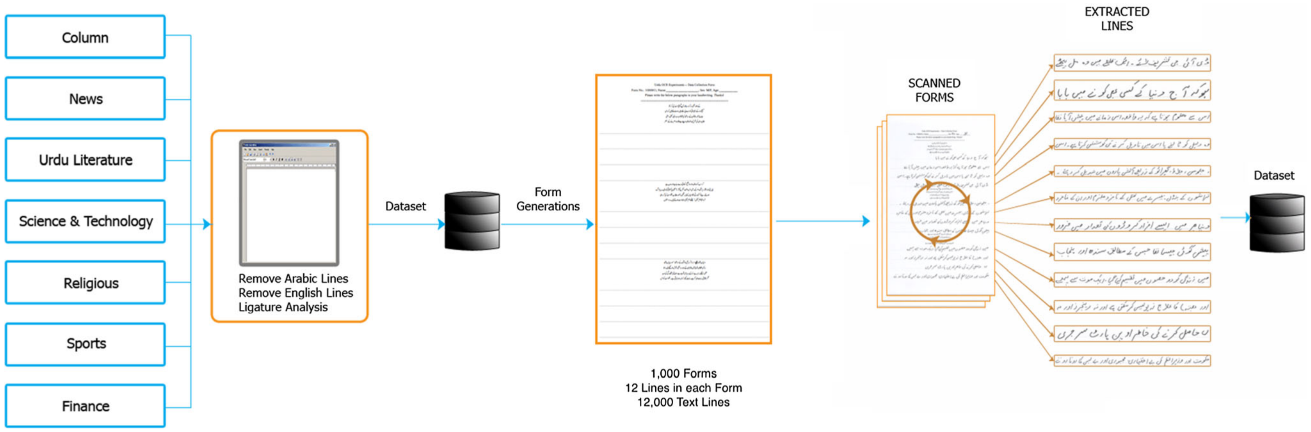

Fig. 1 Process of database preparation: The raw data collected from different writers with different text was first filtered. Then parametric analysis was done and the documents were scanned. The text lines were then segmented and added to the database

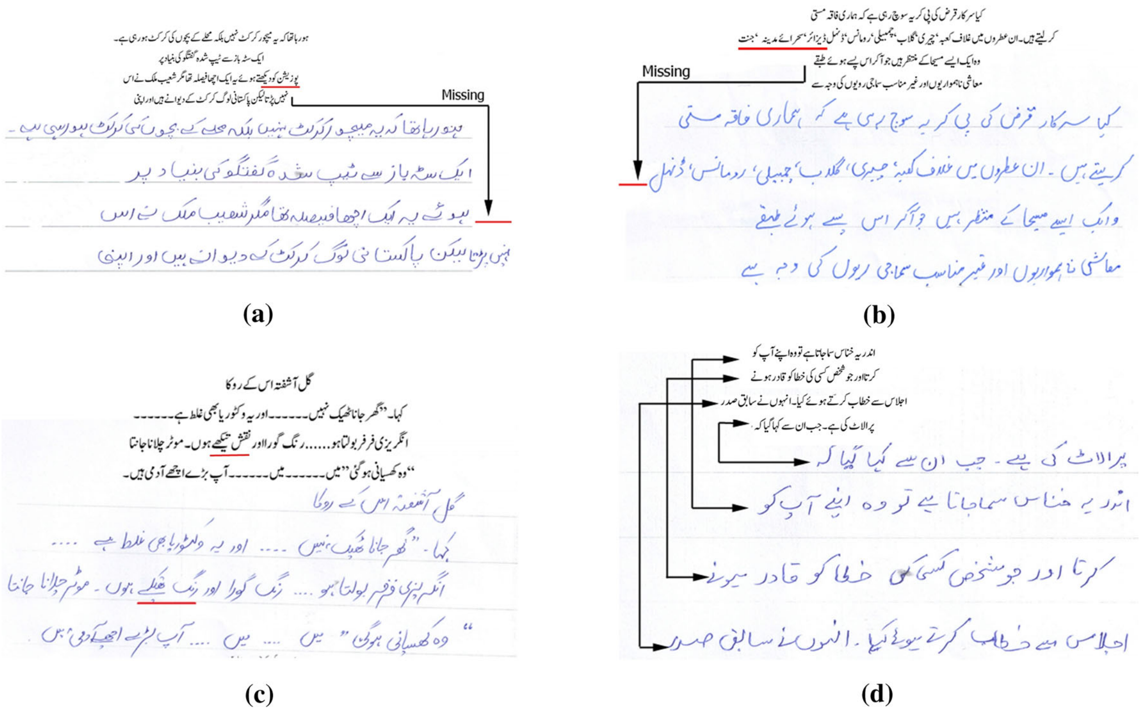
Missing
Missing
(a)
(b)
(c)
(d)

Fig. 2 The figures show the errors in the forms that had to be manually corrected. **a** Words missing from the start of the line **b** Words missing from the end of the line **c** Words written incorrectly **d** Lines interchanged

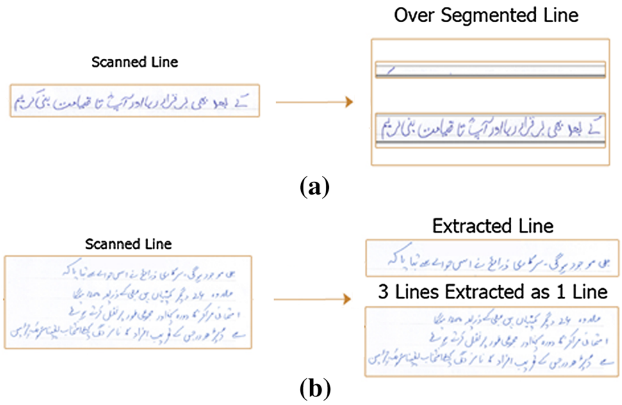

Fig. 3 The figures show the problems in segmenting the handwritten Urdu text lines. **a** Due to superscripts and subscripts the segmentation algorithms fail resulting in over-segmentation. **b** Due to variable gaps in the lines the projection profile based methods fail, resulting in under-segmentation

Table 1 NUST-UHWR dataset statistics

Total no of scanned forms	884
Total no of text lines	10,608
Total no of words	110,940
Total no of unique words	24,327
Total no of unique ligatures	8237

width at low resolution limits the maximum number of outputs in the final layer; moreover, there is a great amount of information that is lost due to resizing to 32 pixels height.

An approach for addressing the mentioned limitation of CRNN is to increase the input resolution of the input image. We fed our dataset at 64 pixel height while maintaining the aspect ratio. In order to maintain the 3D feature volume to a 2D sequence conversion in line with the existing architecture, we introduced an additional max pooling layer after the final convolutional layer (Table 2). However, the model loses important information when it performs excessive max pooling.

4.2 Proposed architecture

We perform some changes to CRNN and propose a new architecture (Fig. 4) to address its limitations. In order to address the limited resolution, we increase the size of the input layer to 128 pixels height while maintaining the aspect ratio. We concatenate the features in depth before feeding them to the LSTM layers instead of eliminating the dimension with excessive max pooling.

We introduce several measures to prevent over-fitting. Batch normalization layers are part of each convolutions block. Random sampling has proved to increase generalization by adding stochasticity [49]. We sample each batch

Table 2 Network configuration of modified CRNN

Layer	Configuration
Conv	1 → 64, 3 × 3
Max Pooling	2 × 2
Conv	64 → 128, 3 × 3
Max Pooling	2 × 2
Conv	128 → 256, 3 × 3
Batch Normalization	–
Conv	256 → 256, 3 × 3
Max pooling	2 × 2
Conv	256 → 512, 3 × 3
Batch Normalization	–
Conv	512 → 512, 3 × 3
Max Pooling	2 × 2
Conv	512 → 512, 3 × 3
Batch Normalization	–
Max Pooling	2 × 1
BDLSTM	256
BDLSTM	256
CTC	Output

An additional max pooling layer is added to allow increased input resolution

randomly from the dataset for each epoch leading to a new combination of images on every update. Moreover, a random distortion layer is introduced before the input layer to randomly distort the image before feeding it to the network for training. These distortions are in the form of translation, smear, rotation and erode as shown in Fig. 5. The random distortions have the effect that the network never sees the same image twice during training.

Moreover, we perform an ablation study over the parameters of the architecture to find the optimal network structure. A detailed overview of the architectures is provided in Table 3.

4.3 Introduction of N-gram language models

Previous works in handwriting recognition make use of explicit language models [50, 51]. Integration of n-gram language models has shown to improve accuracy of recognizers. N-gram language models provide approximation for sentence probabilities based on the relative frequencies of n words [52]. They predict the probability of a word to appear next in a sentence based on the sequence of previous $n-1$ words. The order n of an n-gram defines the context and the number of preceding words that will be used for approximation.

We use interpolated n-gram models to combine the strengths of higher- and lower-order n-grams and smoothing to prevent zero frequency n-gram problems [53], which can arise due to unseen words. For smoothing, we use modified Kneser-Ney smoothing [54], a state-of-the-art technique in language modeling and an extended version of absolute discounting [55], which combines the lower-order and higher-order models using the contextual relationship between the n-gram orders.

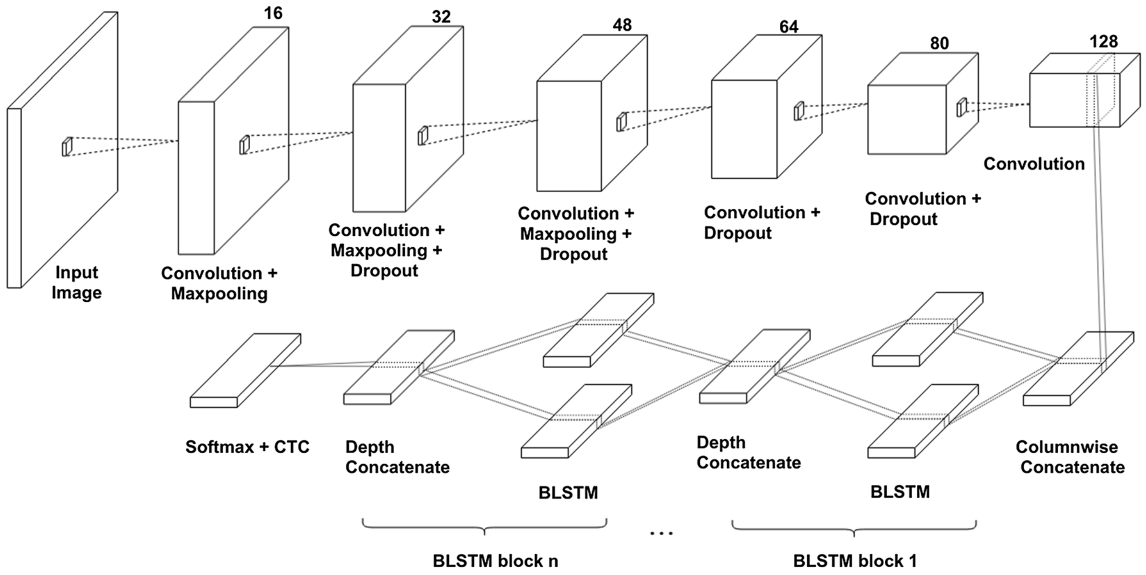

Fig. 4 The network architecture. The architecture consists of convolution blocks followed by a bidirectional LSTM block and a transcription layer

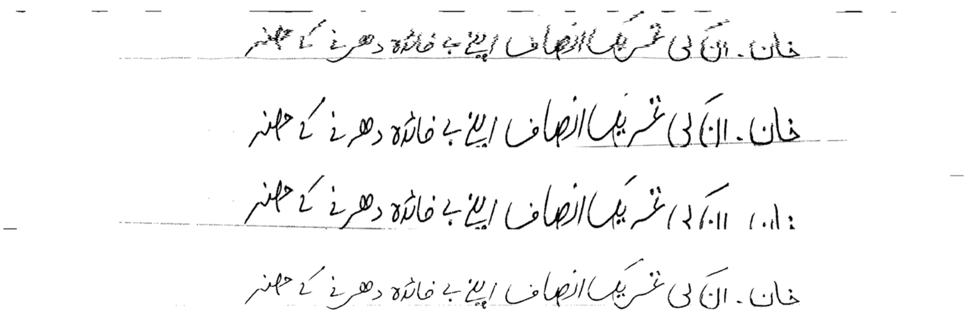

Fig. 5 The figures show some of the images with different kinds of distortions such as smearing, translation, rotation, erosion

Table 3 We test different configurations for the model accuracy, training time and compute resources required

	UHWR6	UHWR5	UHWR4	UHWR3
Conv	1 → 16, 3 × 3	1 → 16, 3 × 3	1 → 16, 3 × 3	1 → 16, 3 × 3
Max Pooling	2 × 2	2 × 2	2 × 2	2 × 2
Conv	16 → 32, 3 × 3	16 → 32, 3 × 3	16 → 32, 3 × 3	16 → 32, 3 × 3
Max Pooling	2 × 2	2 × 2	2 × 2	2 × 2
Dropout	0.2	0.2	0.2	0.2
Conv	32 → 48, 3 × 3	32 → 48, 3 × 3	32 → 48, 3 × 3	32 → 48, 3 × 3
Max pooling	2 × 2	2 × 2	2 × 2	
Dropout	0.2	0.2	0.2	
Conv	48 → 64, 3 × 3	48 → 64, 3 × 3	48 → 64, 3 × 3	
Dropout	0.2	0.2		
Conv	64 → 80, 3 × 3	64 → 80, 3 × 3		
Dropout	0.2			
Conv	80 → 128, 3 × 3			
BLSTM Block				
CTC	Output	Output	Output	Output

We decrease the number of CNN layers from UHWR6 to UHWR3

5 Experimental setup

5.1 Dataset and evaluation

For our Urdu handwriting experiments, we use NUST-UHWR dataset. NUST-UHWR is split into threefold with 8000 training images, 1300 validation images and 1306 test images. Images are flipped horizontally as proposed by Naeem et al. [6] for left to right processing. We tested our models recognition capability by training it on IAM Offline dataset. The dataset consists of unconstrained handwritten text compiled using sentences from LOB text corpus [56]. The IAM database [57] includes 1539 text forms written by 657 different writers. The dataset is partitioned into 6161 training images, 966 validation images and 2915 test images. We evaluate our handwriting recognition model using Character Error Rate (CER) and Word Error Rate (WER).

5.2 Implementation details

We implement our models in Pytorch. For our proposed model, we use a *LeakyReLU* activation function and a *learning rate* of $3e^{-4}$ with Adam Optimizer. We built our language models using SRILM toolkit [58] and Kaldi [59] decoder. The γ value for pseudo-likelihood calculation, acoustic scale factor and beam width were set to 0.2, 1.79 and 65, respectively, as used by [30]. We used our training data ground truth as our text corpus. Our vocabulary consisted of 5482 tokens.

5.3 Experiments performed

We have performed several experiments to evaluate the efficacy of our approach. First of all, we wanted to establish the suitability of CRNN architectures employed in scene text recognition for Urdu handwriting recognition. So, we re-implemented one of the state-of-the-art scene text recognition paper and apply that model for Urdu HWR. The lessons learnt help us develop a better CRNN model architecture suitable for handwriting recognition.

We have shown that performing excessive max pooling causes the model to correctly recognize smaller diacritics marks in Urdu script.

We then evaluated our model for several scenarios. Firstly, we have performed extensive ablation studies to validate our choice of architecture. Our results for changing the CNN layers show that deeper architectures tend to perform better owing to better capability to capture fine details. Ablation study on LSTM layers indicates that having more layers greatly help in learning semantic information.

Secondly, we test the generalizability of our method by training the same architecture for English (IAM offline dataset) handwriting recognition task. Our method was able to achieve the equivalent results on state-of-the-art methods for this dataset.

Thirdly, we also trained a joint model for bilingual handwriting recognition to see the efficacy of our proposed model. In this regard, a joint model on IAM Offline DB and our proposed NUST-UHWR DB comprising of 217 class samples.

Lastly, we evaluate the effect of employing the language model on Urdu handwriting recognition task. Several n-gram models were evaluated to determine the optimal number of “n” for the task in hand.

6 Results and discussion

This section discusses the experimental evaluation of different models and the limitations of the existing models. We further address these limitations in our model and perform an ablation study to study the impact of our design choices.

6.1 Limitations of CRNN for Urdu handwriting recognition

The architectures studied and discussed in Sect. 4.1 highlight a major limitation of adapting CRNNs i.e., scene text recognition architectures for handwriting recognition. Scene text recognition can afford to lose information during max pooling as the task is not dependent on fine-grained differences between characters. We show that this is not the case in handwriting recognition, where small details are required to differentiate between different ligatures. Moreover, these details are especially important in complex and continuous ligature languages. In such languages, diacritics are the primary features for distinguishing different words/alphabets. For CRNN, before our experiments, the implementation was trained on a scene text dataset for sanity check to reproduce Shi et al.’s experiments [48]. Once the results were reproduced, we resized our Urdu handwriting images to height of 32 pixels to feed them to the network for our experiments. We observed during pre-processing that this resizing leads to loss of critical information required for the Urdu script such as ligatures and diacritics. This was further validated when the network failed to learn these details and the loss function output kept oscillating with a very high value.

Following our intuition, we modeled our next architecture on CRNN with an increased resolution as discussed in Sect. 4.1. The model achieved an error rate of 19.34% on the test set. The model was then allowed to train further for 2200 epochs but it did not yield a greater accuracy. The accuracy was far from the numbers achieved on problems such as scene text with similar architectures. These results have been reported in Table 4. After analyzing the failure cases, we build upon our original hypothesis that this discrepancy is due to two major limitations. As shown in Fig. 8a–d, Urdu handwriting has complex superscript and subscript that is crucial to differentiate between very similar letters and ligatures. These complexities are overlooked when the model performs max pooling intensively as in the case of CRNN. Moreover, these fine details are also lost due to resizing the image to a height of 64 or 32 as in the previous case. Hence, higher resolution features are required to achieve comparable results to other domains in OCR.

6.2 Proposed architecture results

The proposed architecture in this paper, in Sect. 4.2, addresses these limitations and achieves state-of-the-art performance. We show comparative results with existing methods in Table 4. The increased resolution allows for fine-grained feature extraction in the convolutional layers for smaller objects or in our case subscripts and superscripts. Researchers in fields such as object detection have

Table 4 Comparison of the character error rate on UHWR dataset with state-of-the-art methods for printed and handwritten text recognition

Method	Valid CER (%)	Test CER (%)
BLSTM [1]	27.39	27.05
Modified CRNN [48]	18.57	19.34
MDLSTM [24]	14.11	19.15
CNN-RNN [60]	13.25	14.12
BGRU [27]	13.50	13.28
Proposed (no LM)	7.25	7.35

Our proposed architecture outperforms the other methods even without language modeling reported similar results for smaller objects with increased resolution of input layer [61].

6.2.1 Ablation study

We perform an ablation study over our network architecture to validate our architectural improvements. For the first set of experiments, we vary the number of CNN layers in the model shown in Fig. 4 from 6 to 3 while fixing the BLSTM block at a layer size of 5 with 256 cells in every layer. We observe that the deepest architecture, UHWR6, naturally achieves the highest accuracy due to the model's high capacity. Moreover, we see small degradations in error rate as we decrease the number of CNN layers in the other architectures. This phenomena is also studied extensively in the Computer Vision community and validated by our experiments. The initial CNN layers capture the low level features that are usually common across different architectures while the higher layers extract high level features. These layers, although beneficial, can be eliminated based on the computational resources available for training and inference with the trade off in accuracy. We observed that UHWR6, UHWR5 and UHWR4 architectures with six, five and four convolutional layers, respectively, had comparable performance. As seen from Table 5, the test CER increases from 7.35 to 7.66 as we go from 6 CNN layers to 4 CNN layers. We see a more drastic increase to 8.29 as we go to 3 CNN layers.

The training and validation error rates are presented in Fig. 6. It is observed during training that the deeper an architecture is, the sooner it converges. This is attributed to the modeling capacity of the model as deeper models can separate the data points to respective classes more easily in higher dimensions. The shallower models have limited parameters to tune leading to an increased training time. However, the shallow architectures are able to approximate the same functions to a close accuracy.

For the next set of experiments, we take the best performing architecture UHWR6 and perform a ablation study over different depths of the BLSTM Block. The number of BLSTM layers is decreased from 5 to 1. We can see from Table 6 that the deepest architecture UHWR6 achieves the best accuracy validating our choice of architecture. Scene text recognition architectures like CRNN [48] often use only 2 LSTM layers. However, tasks like Urdu Handwriting Recognition greatly benefit from the increased semantic information extracted by the deeper RNN blocks as shows in Table 6.

Table 5 UHWR ablation study on CNN layers: The table shows the results of training architectures with different depths of CNN layers on NUST-UHWR dataset

	UHWR6	UHWR5	UHWR4	UHWR3
Epochs	195	210	228	295
Train CER (%)	6.33	6.55	6.63	7.09
Valid CER (%)	7.25	7.34	7.49	8.25
Test CER (%)	7.35	7.45	7.66	8.29

The number of BLSTM layers is kept the same at 5. The best CER figure is 7.35% for UHWR6 architecture, which is also the deepest

6.2.2 Generalizability

We further test the generalizability of the proposed architectures by training the top 3 performing models on IAM Offline Handwriting dataset. The results are presented in Table 7. The deepest architecture achieves the lowest error rates while the shallower architectures closely follow. The results are equivalent to the state-of-the-art performance on IAM handwriting dataset that stands at 5–9% CER [62] and serve as a baseline comparison.

6.3 N-gram language model

OCR results can be further improved by introducing an explicit language model as discussed in sect. 4.3. The performance of n-gram models depends on the order n. For our Urdu dataset, we test different orders of n-grams from 2 to 4 and report the character error rate (CER) and word error rate (WER). The lowest error rate was achieved using 4-gram. CER improved by 1.97% for validation set and 1.86% for test set. A significant improvement of 7.61% is observed in WER. We noticed that orders $n = 2, 3, 4$ produced similar results with negligible performance improvement as order increased, whereas the training time increased significantly. Table 8 summarizes our results with different n-gram orders. Higher-order n-grams do show better performance but the order can be reduced based on the resources available. The performance of n-gram models depends significantly on the text corpus used for training. We use our training ground truth as our n-gram vocabulary. The performance can further be improved by using a bigger corpus. In future, we propose combining other Urdu text sources [6] with the training corpus to estimate n-gram language model parameters.

6.4 Joint training model

Arabic scripts have an alphabet set that is completely different from Latin script. The two scripts are also different in nature and possess different language rules. We further study the capacity and generalization ability of our model by jointly training Urdu with English. We combine the training, validation and test sets of IAM database and

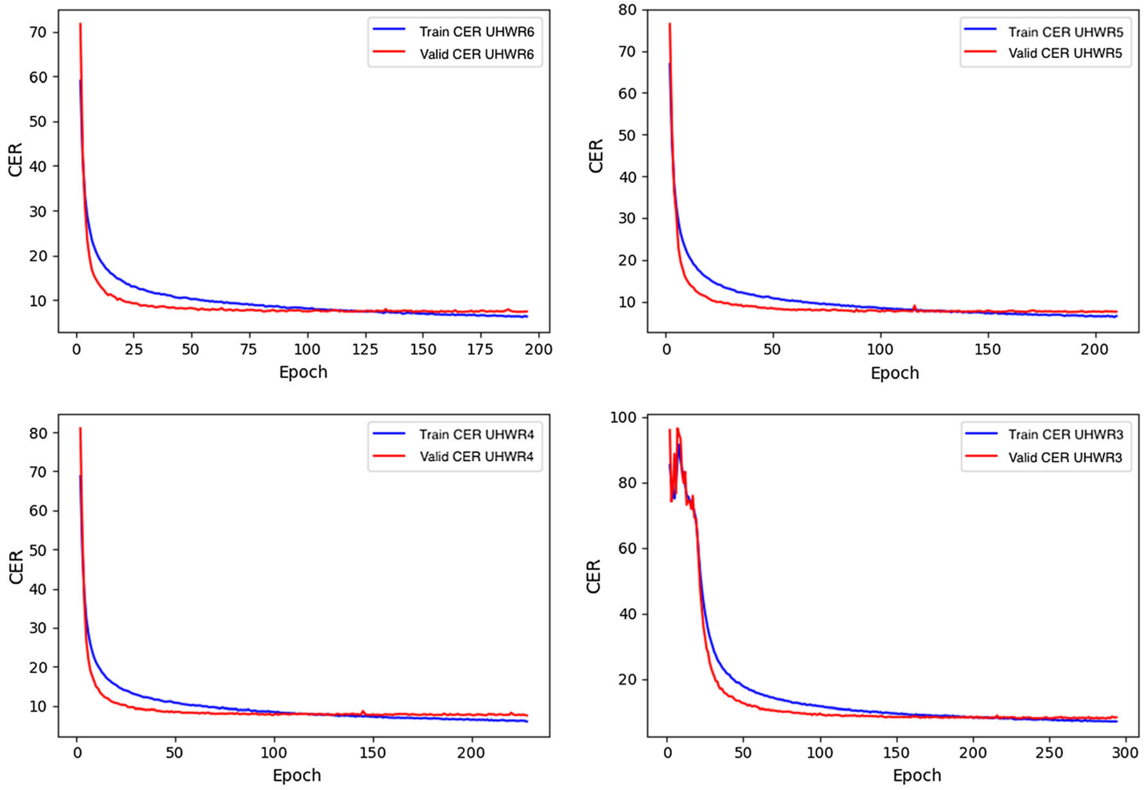

Fig. 6 The figures show the train and validation CER for the architectures shown in Table 3. Curves correspond to the architectures UHWR6, UHWR5, UHWR4 and UHWR3. All the models were trained till convergence with a learning rate 0.0003

Table 6 UHWR6 ablation study on BLSTM layers

	UHWR6	UHWR6L4	UHWR6L3	UHWR6L2	UHWR6L1
BLSTM layers	5	4	3	2	1
Train CER (%)	6.33	5.94	6.01	7.22	10.96
Valid CER (%)	7.25	7.32	7.59	8.47	10.39
Test CER (%)	7.35	7.49	7.68	8.45	10.54

We take the best performing architecture UHWR6 and systematically decrease the number of RNN layers from 5 to 1 to study the impact on the accuracy

Table 7 Results on IAM Offline handwriting dataset: The results are comparable to the state-of-the-art CER for IAM dataset as described in the text

	UHWR6	UHWR5	UHWR4
Train CER (%)	2.87	2.96	2.91
Valid CER (%)	3.71	3.84	3.78
Test CER (%)	5.68	5.91	5.81

The deepest model UHWR6 performs the best with a CER of 5.68%

Table 8 Performance of different orders of n-gram language models, with results in CER as well as WER

n	CER (%)		WER (%)	
	Valid	Test	Valid	Test
no lm	7.25	7.35	27.32	27.00
2	5.39	5.55	19.5	19.48
3	5.30	5.51	19.26	19.48
4	5.28	5.49	19.23	19.39

The CER improves by 1.97% for validation set and 1.86% for the test set. A significant improvement of 7.61% in WER is seen. The orders $n = 2, 3, 4$ produce similar results with negligible gain in performance

NUST-UHWR to have a training set of 14,161 images, validation set of 2266 images and test set of 4221 images. We then take the three top performing architectures, UHWR6, UHWR5, UHWR4 and train them on this combined dataset. The batches are randomly sampled to get an equal representation of Urdu and English during

Table 9 Joint training results: The CER is 6.61% for the combined test set

	UHWR6	UHWR5	UHWR4
Epochs	131	146	195
Train CER (%)	6.99	7.86	7.73
Valid CER (%)	6.35	6.38	6.48
Test CER (%)	6.61	6.72	6.85
Test English CER (%)	6.01	6.07	6.28
Test Urdu CER (%)	7.90	8.06	8.01

However, there is a drop in the overall accuracy, which is due to the increase in the number of classes and different nature of both languages (disjoint character-set, grammar rules, ligature)

training. The random distortion layers are kept and the networks are trained to convergence and tested individually on each test set and the combined test set.

In Table 9 we present the results for our joint training for Bilingual OCR. We observe that the deepest architecture again converges the fastest and achieves the best error rate as discussed previously. The error rate is 6.61% for the combined test set for the deepest architecture. However, there is a drop in over-all accuracy when we apply this model for individual language. This can be attributed to two key reasons. There is a significant increase in the number of classes the network is handling (217 classes), which introduces inherent complexity during softmax. Moreover, the very different nature of both languages also introduces additional complexity. The results are summarized in Table 9. After analyzing the results, one emerging hypothesis can be that the accuracy on English and Urdu can jointly improve if the convolutional blocks are frozen and the LSTM layers are fine tuned for each language after the joint training.

6.5 Model output and failure cases

Some example outputs and failure cases are shown in Figs. 7 and 8. Figure 7b is a handwritten text, consisting of names. These names are not common words of Urdu and they are not used frequently. Despite their infrequency, the model's output is correct. This exhibits a desirable property of good OCR systems, *Context Independence*. A good OCR system should not infer complete spellings of an incomplete word from the context. Moreover, it should not auto-correct any spelling mistakes in the word. Figure 7d shows an example where the text is written in irregular shape, but our model has produced the correct OCR output. Figure 8a and c shows example images where after misclassifying numerals in the middle of the sentence, the model is able to recognize the remaining sentence properly. The other failure case shown in Fig. 8d is when characters in subscript are mis-classified. This happens because some characters in Urdu, used in subscript, are extremely similar. These characters are vital in distinguishing the alphabet being used in the word. Figure 8b shows OCR errors in Urdu characters written in almost the same way.

We also experimented with low-resolution images of Urdu to understand how it effects the output of our architecture. Figure 9 show some of these examples where we take a text line comprising of less detailed ligatures and corresponding outputs. Figure 9a shows a moderately degraded image where our model was able to perfectly predict the output. It also corrected the spelling mistake in the image. Figure 9b shows the same sample but with very less resolution. In this case, our model was not able to correctly predict the output. Figure 9c and d shows two other examples of highly degraded samples comprising of less detailed Urdu ligatures and corresponding outputs.

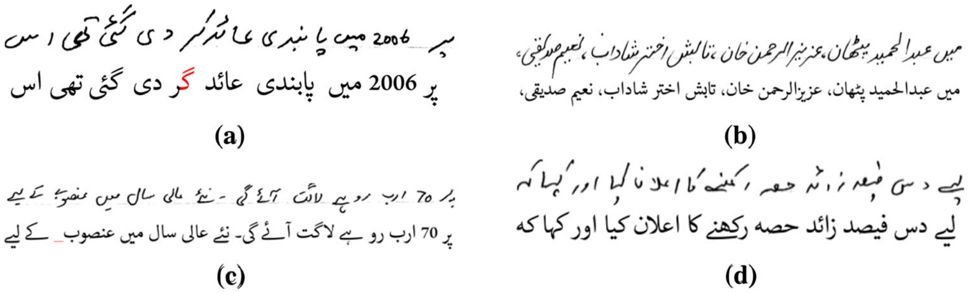
پر 2006 میں پابندی عائد گر دی گئی تھی اس
(a)
میں عبدالحمید پٹھان، عزیزالرحمن خان، تابش اختر شاداب، نعیم صدیقی،
(b)
پر 70 ارب رو ہے لاگت آئے گی۔ نئے عالی سال میں عنصوب کے لیے
(c)
لیے دس فیصد زائد حصہ رکھنے کا اعلان کیا اور کہا کہ
(d)

Fig. 7 The figures show some of the example images and their output written below. The OCR errors have been highlighted in red. Example **a** shows a text-line starting with a number was recognized correctly with only a single character misclassified. Example **b** and **c** illustrate the context independence property of a good OCR engine. In example **b** the text line consists of names that are not part of the common Urdu word set, yet our system was able to recognize them correctly. In example **d** the handwritten text has an irregular shape where some alphabets are incomplete but the OCR output is correct

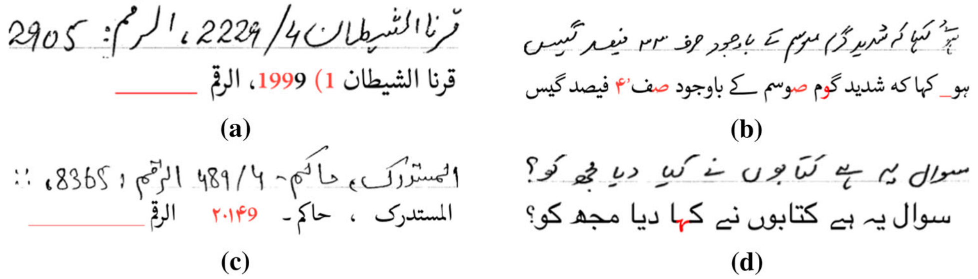
قرنا الشيطان 1) 1999، الرق
(a)
ہو_ کہا کہ شدید گوم صوسم کے باوجود صف ۴' فیصد گیس
(b)
المستدرک ، حاکم- ۲۰۱۴۹ الرق
(c)
سوال یہ ہے کتابوں نے کہا دیا مجھ کو؟
(d)

Fig. 8 The figures show the outputs of some images for which the model failed to produce the correct output. The OCR outputs are shown below each image. The OCR errors have been highlighted in red and the missing characters are shown as red lines. In examples **a** and **c**, numerals were not recognized properly. Example **b** illustrates the failure of the model to correctly classify Urdu characters that have similar shapes. Typical examples are (ﻡ) and (ص). Some subscripts in Urdu are extremely similar but are used to distinguish the type of alphabet being used in a word. Example **d** shows this kind of error in red where (ی) has been confused with (ھ)

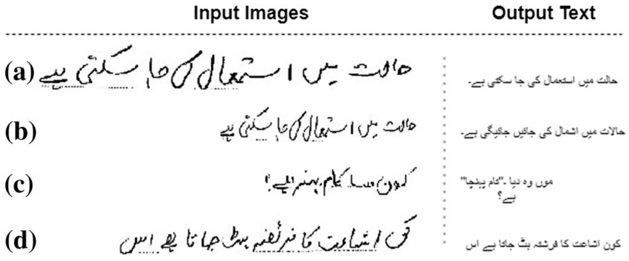
Input Images
Output Text
(a)
(b)
(c)
(d)

Fig. 9 The figure shows output of our proposed architecture on low-resolution images. Example **a** shows a moderately degraded image with less detailed Urdu ligature on which our model was able to perform accurately. Example **b** shows the same sample with very low resolution; our model failed to correctly recognize the text. Examples **c** and **d** also shows some very low-resolution images on which our model failed to perform correctly

7 Conclusion

In this paper, we have presented a new unconstrained dataset, named NUST Urdu Handwriting Dataset (NUST-UHWR), for Urdu handwriting . We also present a hybrid CNN-RNN model and report its results on our dataset.

We tested four different variants of the model architecture with varying CNN layers. We achieved 7.35% CER on the test set with the best performing network. We further used n-gram language models and improved the model's CER to 5.49% for test set. Our model and the new unconstrained dataset fills a large gap in literature with respect to Urdu handwriting recognition. The new dataset and model architecture will allow further research bridging other technological gaps in literature and will pave the way towards digitization of content for historic as well as present Indic scripts. We further demonstrate the generalization capacity of our CNN-RNN model by showing that it is possible to jointly train a left to right and a right to left language with a common feature extractor.

There is a need to introduce models proposed in the literature to commercial products and projects for main stream consumers. The rich and vast knowledge base of the region with respect to art and literature needs to be opened to the world. Urdu Handwriting recognition possesses great potential to impact a very big user base of the Indian subcontinent. Moreover, further research for the recognition of other complex ligature languages can test similar, if not the same model architecture with proposed improvements.

Declarations

Conflict of interest The authors declare that there is no conflict of interest.

References

1. Ul-Hasan A, Ahmed SB, Rashid F, Shafait F, Breuel TM (2013) Offline printed Urdu Nastaleeq script recognition with Bidirectional LSTM networks. In: Document analysis and recognition (ICDAR), 2013 12th international conference on IEEE, pp 1061–1065
2. Khattak IU, Siddiqi I, Khalid S, Djeddi C (2015) Recognition of Urdu ligatures: a holistic approach. In: Document analysis and recognition (ICDAR), 2015 13th international conference on, IEEE, pp 71–75
3. Ahmad R, Zeshan AM, Faisal RS, Liwicki M, Dengel A (2018) Space anomalies in ocrs for arabic like scripts. In: 2018 IEEE 2nd international workshop on arabic and derived script analysis and recognition (ASAR), IEEE, pp 67–71
4. Ul-Hasan A (2016) Generic text recognition using long short-term memory networks. PhD thesis, University of Kaiserslautern
5. Hussain S, Niazi A, Anjum U, Irfan F (2014) Adapting tesseract for complex scripts: an example for Urdu Nastalique. In: Document analysis systems (DAS), 2014 11th IAPR international workshop on, IEEE, pp 191–195
6. Naeem MF, ul Sehr ZN, Awan AA, Shafait Faisal, ul Hasan A (2017) Impact of ligature coverage on training practical Urdu OCR systems. In: Document analysis and recognition (ICDAR), 2017 14th IAPR international conference on, IEEE, vol 1, pp 131–136
7. Bin AS, Saeeda N, Salahuddin S, Imran RM, Iqbal UA, Ali KA (2017) UCOM offline dataset: an Urdu handwritten dataset generation. Int Arab J Inf Technol 14(2):239–245

8. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
9. Sepp H, Jürgen S (1997) Long short-term memory. Neural Comput 9(8):1735–1780
10. Pal U, Sarkar A (2003) Recognition of printed Urdu script. In: International conference on document analysis and recognition, pp 1183–1187
11. Javed Sobia T, Sarmad H, Ameera M, Samia A, Sehrish J, Huma M (2010) Segmentation free Nastalique Urdu OCR. World Acad Sci Eng Technol 46:456–461
12. Ud Din I, Siddiqi I, Khalid S, Azam T (2017) Segmentation-free optical character recognition for printed Urdu text. EURASIP J Image Video Process 1:62
13. Sabbour N, Shafait F (2013) A Segmentation-free approach to Arabic and Urdu OCR. In: Document recognition and retrieval XX. International society for optics and photonics, vol 8658, p 86580N
14. Smith R (2007) An overview of the Tesseract OCR engine. In: Document analysis and recognition, 2007. ICDAR 2007. Ninth international conference on, IEEE, vol 2, pp 629–633
15. Sardar S, Wahab A (2010) Optical character recognition system for Urdu. In: Information and emerging technologies (ICIET), 2010 international conference on, IEEE, pp 1–5
16. Bin AS, Saeeda N, Imran RM, Faisal RS, Zeeshan AM, Breuel Thomas M (2016) Evaluation of cursive and non-cursive scripts using recurrent neural networks. Neural Comput Appl 27(3):603–613
17. Saeeda N, Umar Arif I, Riaz A, Ahmed Saad B, Shirazi Syed H, Razzak Muhammad I (2017) Urdu Nastaliq text recognition system based on multi-dimensional recurrent neural network and statistical features. Neural Comput Appl 28(2):219–231
18. Saeeda N, Umar AI, Ahmed R, Razzak MI, Rashid SF, Shafait F (2016) Urdu nastaliq text recognition using implicit segmentation based on multi-dimensional long short term memory neural networks. SpringerPlus 5(1):1–16
19. Saeeda N, Umar Arif I, Riaz A, Ahmed Saad B, Shirazi Syed H, Imran S, Razzak Muhammad I (2016) Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks. Neurocomputing 177:228–241
20. Saeeda N, Umar Arif I, Riaz A, Imran S, Ahmed Saad B, Razzak Muhammad I, Faisal S (2017) Urdu Nastaliq recognition using convolutional-recursive deep learning. Neurocomputing 243:80–87
21. Réjean P, Srihari Sargur N (2000) Online and off-line handwriting recognition: a comprehensive survey. IEEE Trans Pattern Anal Mach Intell 22(1):63–84
22. Alex G, Marcus L, Santiago F, Roman B, Horst B, Jürgen S (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31(5):855–868
23. Graves A, Fernández S, Gomez F, Schmidhuber J (2006) Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on machine learning, ACM, pp 369–376
24. Graves A, Schmidhuber J (2009) Offline handwriting recognition with multidimensional recurrent neural networks. In: Koller D, Schuurmans D, Bengio Y, Bottou L (eds) Advances in neural information processing systems 21. Curran Associates Inc., pp 545–552
25. Pal U, Jayadevan R, Sharma N (2012) Handwriting recognition in Indian regional scripts: a survey of offline techniques. ACM Trans Asian Lang Inf Process 11(1):1–35
26. Messina R, Louradour J (2015) Segmentation-free handwritten Chinese text recognition with LSTM-RNN. In: Document analysis and recognition (ICDAR), 2015 13th international conference on, IEEE, pp 171–175
27. Chen L, Yan R, Peng L, Furuhata A, Ding X (2017) Multi-layer recurrent neural network based offline Arabic handwriting recognition. In: 2017 1st international workshop on Arabic script analysis and recognition (ASAR), pp 6–10
28. Bluche T, Louradour J, Messina R (2017) Scan, attend and read: end-to-end handwritten paragraph recognition with MDLSTM attention. In: Document analysis and recognition (ICDAR), 2017 14th IAPR international conference on, IEEE, vol 1, pp 1050–1055
29. Wu YC, Yin F, Chen Z, Liu CL (2017) Handwritten chinese text recognition using separable multi-dimensional recurrent neural network. In: Document analysis and recognition (ICDAR), 2017 14th IAPR international conference on, IEEE, vol 1, pp 79–84
30. Puigcerver J (2017) Are multidimensional recurrent layers really necessary for handwritten text recognition? In: Document analysis and recognition (ICDAR), 2017 14th IAPR international conference on, IEEE, vol 1, pp 67–72
31. Bluche T, Messina R (2017) Gated convolutional recurrent neural networks for multilingual handwriting recognition. In: Document analysis and recognition (ICDAR), 2017 14th IAPR international conference on, IEEE, vol 1, pp 646–651
32. Adak C, Chaudhuri BB, Blumenstein M (2016) Offline cursive Bengali word recognition using CNNs with a Recurrent model. In: 2016 15th international conference on frontiers in handwriting recognition (ICFHR), IEEE, pp 429–434
33. Ignacio TJ, Dey S, Fornés A, Lladós J (2017) Handwriting recognition by attribute embedding and recurrent neural networks. In: Document analysis and recognition (ICDAR), 2017 14th IAPR international conference on, IEEE, vol 1, pp 1038–1043
34. Shaw B, Bhattacharya U, Parui SK (2014) Combination of features for efficient recognition of offline handwritten Devanagari words. In: Frontiers in handwriting recognition (ICFHR), 2014 14th international conference on, IEEE, pp 240–245
35. Shaw B, Bhattacharya U, Parui SK (2015) Offline handwritten Devanagari word recognition: information fusion at feature and classifier levels. In: Pattern recognition (ACPR), 2015 3rd IAPR Asian conference on, IEEE, pp 720–724
36. Mukherjee PS, Bhattacharya U, Parui SK (2018) An efficient feature vector for segmentation-free recognition of online cursive handwriting based on a hybrid deep neural network. In: 2018 13th IAPR international workshop on document analysis systems (DAS), IEEE, pp 435–440
37. Dutta K, Krishnan P, Mathew M, Jawahar CV (2018) Offline handwriting recognition on Devanagari using a new benchmark dataset. In: 2018 13th IAPR international workshop on document analysis systems (DAS), pp 25–30
38. Chakraborty B, Mukherjee PS, Bhattacharya U (2016) Bangla online handwriting recognition using recurrent neural network architecture. In: Proceedings of the tenth Indian conference on computer vision, graphics and image processing, ACM, p 63
39. Kumar M, Jindal SR, Jindal MK, Lehal GS (2018) Improved recognition results of medieval handwritten Gurmukhi manuscripts using boosting and bagging methodologies. Neural Process Lett 20:43–56
40. Raza A, Siddiqi I, Abidi A, Arif F (2012) An unconstrained benchmark Urdu handwritten sentence database with automatic line segmentation. In: Frontiers in handwriting recognition (ICFHR), 2012 international conference on, IEEE, pp 491–496
41. Malik S, Khan SA (2005) Urdu online handwriting recognition. In: Emerging technologies, 2005. Proceedings of the IEEE symposium on, IEEE, pp 27–31
42. Urs-Viktor M, Horst B (2001) Using a statistical language model to improve the performance of an HMM-based cursive

handwriting recognition system. Int J Pattern Recognit Artif Intell 15(01):65–90

43. Wassim S (2017) Language modelling for handwriting recognition. Theses, Normandie Université
44. Lehal GS, Rana A (2013) Recognition of Nastalique Urdu ligatures. In: Proceedings of the 4th international workshop on multilingual OCR, ACM, p 7
45. UET (2012) Valid ligatures of Urdu. http://www.cle.org.pk/software/ling_resources/UrduHighFreqLigature.htm. (Accessed 7 Sep 2019)
46. Joost VB, Faisal S, Breuel Thomas M (2010) Combined orientation and skew detection using geometric text-line modeling. Int J Doc Anal Recognit (IJDAR) 13(2):79–92
47. Lehal GS (2013) Ligature segmentation for Urdu OCR. In: Document analysis and recognition (ICDAR), 2013 12th international conference on, IEEE, pp 1130–1134
48. Baoguang S, Xiang B, Cong Y (2017) An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans Pattern Anal Mach Intell 39(11):2298–2304
49. Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. In: Advances in neural information processing systems, pp 2951–2959
50. Voigtlaender P, Doetsch P, Ney H (2016) Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. In: Frontiers in handwriting recognition (ICFHR), 2016 15th international conference on, IEEE, pp 228–233
51. Doetsch P, Kozielski M, Ney H (2014) Fast and robust training of recurrent neural networks for offline handwriting recognition. In: Frontiers in handwriting recognition (ICFHR), 2014 14th international conference on, IEEE, pp 279–284
52. Zimmermann M, Bunke H (2004) N-gram language models for offline handwritten text recognition. In: Ninth international workshop on frontiers in handwriting recognition, IEEE, pp 203–208
53. Martin JH, Jurafsky D (2009) Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Prentice Hall Upper Saddle River, Pearson
54. Chen Stanley F, Joshua G (1999) An empirical study of smoothing techniques for language modeling. Comput Speech Lang 13(4):359–394
55. Hermann N, Ute E, Reinhard K (1994) On structuring probabilistic dependences in stochastic language modelling. Comput Speech Lang 8(1):1–38
56. Johansson S (2008) The tagged LOB corpus: user's manual. http://www.helsinki.fi/varieng/CoRD/corpora/LOB/. (Accessed 15 Dec 2018)
57. Urs-Viktor M, Horst B (2002) The IAM-database: an english sentence database for offline handwriting recognition. Int J Doc Anal Recognit 5(1):39–46
58. Stolcke A (2002) SRILM-an extensible language modeling toolkit. In: Seventh international conference on spoken language processing
59. Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, Hannemann M, Motlicek P, Qian Y, Schwarz P et al. (2011) The Kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding. IEEE signal processing society
60. Safarzadeh VM, Jafarzadeh P (2020) Offline persian handwriting recognition with CNN and RNN-CTC. In: 2020 25th International computer conference, computer society of Iran (CSICC), IEEE, pp 1–10
61. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767
62. Jaramillo JCA, Murillo-Fuentes JJ, Olmos PM (2018) Boosting handwriting text recognition in small databases with transfer learning. In: 2018 16th international conference on frontiers in handwriting recognition (ICFHR), IEEE, pp 429–434

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.