Neural Comput Applic (2017) 28:219–231
DOI 10.1007/s00521-015-2051-4

ORIGINAL ARTICLE

Urdu Nasta'liq text recognition system based on multi-dimensional recurrent neural network and statistical features

Saeeda Naz[1,4] · Arif I. Umar[1] · Riaz Ahmad[2,5] · Saad B. Ahmed[3] · Syed H. Shirazi[1] · Muhammad I. Razzak[3]

Received: 26 April 2015 / Accepted: 26 August 2015 / Published online: 16 September 2015

Abstract Character recognition for cursive script like Arabic, handwritten English and French is a challenging task which becomes more complicated for Urdu Nasta'liq text due to complexity of this script over Arabic. Recurrent neural network (RNN) has proved excellent performance for English, French as well as cursive Arabic script due to sequence learning property. Most of the recent approaches perform segmentation-based character recognition, whereas, due to the complexity of the Nasta'liq script, segmentation error is quite high as compared to Arabic Naskh script. RNN has provided promising results in such scenarios. In this paper, we achieved high accuracy for Urdu Nasta'liq using statistical features and multi-dimensional long short-term memory. We present a robust feature extraction approach that extracts feature based on right-to-left sliding window. Results showed that selected features significantly reduce the label error. For evaluation purposes, we have used Urdu printed text images dataset and compared the proposed approach with the recent work. The system provided 94.97 % recognition accuracy for unconstrained printed Nasta'liq text lines and outperforms the state-of-the-art results.

Keywords Multi-dimensional recurrent neural network · Long short-term memory · OCR · Urdu

✉ Muhammad I. Razzak
imranrazak@hotmail.com

Saeeda Naz
saeedanaz292@gmail.com

Arif I. Umar
arifiqbalumar@yahoo.com

Riaz Ahmad
rahmad@rhrk.uni-kl.de

Saad B. Ahmed
isaadahmed@gmail.com

Syed H. Shirazi
mirpak@gmail.com

[1] Department of Information Technology, Hazara University, Mansehra, Pakistan

[2] University of Technology, Kaiserslautern, Germany

[3] King Saud Bin Abdul Aziz University for Health Sciences, Riyadh, Saudi Arabia

[4] Higher Education Department, GGPGC, No.1, Abbottabad, KPK, Pakistan

[5] Shaheed Benazir Bhutto University, Sheringal, Pakistan

1 Introduction

The excruciating advancement in technology, especially in document image analysis, has been an evident of reliable and efficient OCR systems since last few decades. In an era of globalization, online information access and communication technology have provoked the publishing bodies to make documents available in local and national languages using legacy technology. These documents can be newspapers, novels, stories, proverbs and books. Most of the documents are in the form of images. The legacy technology makes the job tedious for the purpose to transfer, maintain and access such documents over that internet bearing the restriction of low bandwidth. Moreover, such image documents are unsearchable, are uneditable and occupy more storage. Due to invention of android technology and its use in smart phones, tablets and PDAs have made the accessibility and availability of internet with low cost. This prompts the researchers to propose such ideas which facilitate them to see images having text on their handheld devices. This text images can be printed or handwritten documents and images of signboards. There is

an immense demand to make text documents available and publish the local content online either in local or in national languages. For this reason, technologies or softwares like OCR bring to light for regional or local languages.

The OCR is a technology for electronic or mechanical conversion of document images into digitized text form like ASCII/UNICODE [1]. One of the prime objectives of Urdu OCR is to provide text to speech recognition for visually impaired and illiterate people. Furthermore, it makes the document available and readable whenever an individual requires accessing them which in turn will produce fewer headaches for personal PCs to manage the document locally. The backup, archival purposes and machine translation are being served in smart gadgets.

The character recognition rate is mainly dependent on decomposition technique of word/ligature in cursive language. The incorrect segmentation of word into character degrades the classification result. Cursive Arabic script-based character recognition can be either segmentation free or segmentation based. Segmentation-free approach is quite suitable where the segmentation is prone to error. This case is true for Urdu Nasta'liq script where it is difficult to find the segmentation point due to the complexity of this script. But, on the other hand, there are more than 25,000 ligatures; thus, segmentation-free approach for Nasta'liq script is not a suitable choice. Recent developments for cursive script are based on implicit segmentation-based approach. Recurrent neural network has proved its worth for such kind of problems. RNN is a biological inspired classification model that learns features automatically from the input image. The important and daunting step is designing the relevant and good feature, which requires number of heuristic and expert knowledge. Due to the outclass performance of RNN over other machine learning models on English, French, German and Arabic languages, we have used MDRNN for the classification of Urdu Nasta'liq script. MDLSTM has context-capturing property in all directions like up, down, left and right, as well as localization of diacritical marks, and it could perform well for Urdu script. We present a system using multi-dimensional long short-term memory (MDLSTM) and connectionist temporal classification (CTC) based on statistical feature set. The Urdu printed text images (UPTI) [2]. Dataset has been used as a benchmark for Urdu text line recognition. An open-source library named RNNLIB [3] used in our experiment. As it is experienced in [4] that shape variation affected the accuracy, we did not consider the shape variation in this experiment. The rest of the paper is organized as follows: Sect. 2 presents neural network with mainly focus on recurrent neural network, whereas Sect. 3 presents the state of the art. Multi-dimensional RNN-based classification system is presented in Sect. 4, and results are discussed in Sect. 5.

2 Overview of neural network

Artificial neural network (ANN) is inspired from the human biological nervous system [2]. It has layered architecture, i.e., input layer, hidden layer and output layer. It is composed of network of neurons which are joined by weighted connections that take inputs, perform some processing and transmit elaborate patterns of electrical signals. The ANNs are classified into *cyclic ANN* and *acyclic* ANN. The Acyclic ANNs have not cyclical connections which are known as feed-forward neural networks (FNNs). The cyclic ANNs have a cyclic connection which is named as feed-back, recursive or recurrent neural networks (RNNs).

2.1 Recurrent neural network

It was invented in 1980s by Hopfield. It has cyclic path between connections of hidden units (neurons) and internal memory for processing arbitrary sequence of input data [7]. RNN replicates the recurring property of biological neural system. It maintains contextual information and temporally correlates the new events with the previous events; thus, RNN is good at context-aware processing and recognizing patterns occurring in time series [7]. But it cannot retain and correlate information for longer delays. This limitation is known as vanishing gradient problem. The RNN is shown in Fig. 1.

2.2 Long short-term memory (LSTM)

Hochreiter and Schmidhuber [8] fixed the vanishing gradient problem and context access by introducing long short-term memory recurrent neural network (LSTM-RNN). The basic unit of LSTM architecture is memory block. The LSTM memory block comprises one or more different types of memory cells and three adaptive multiplicative gates so-called input gate, forget gate and output gate as shown in Fig. 2. The architecture of LSTM illustrates exactly same as RNN, but the hidden layer units are

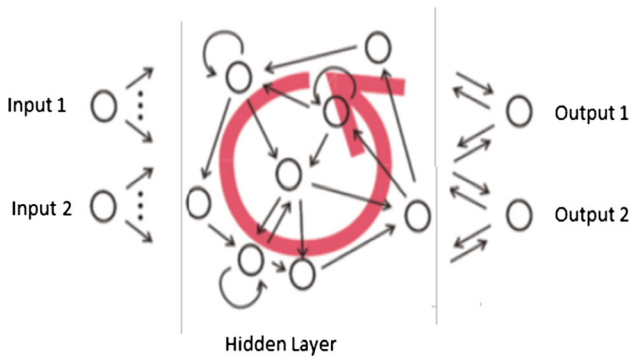


Fig. 1 Recurrent neural network [7]

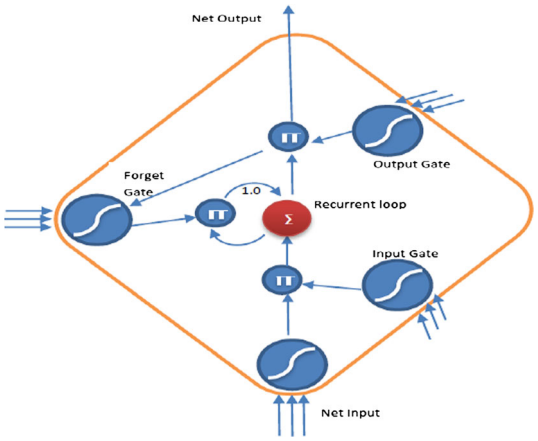


Fig. 2 Long short-term memory [10]

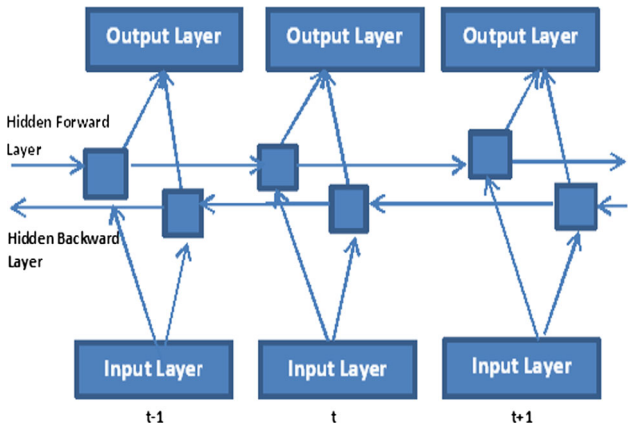


Fig. 3 Bidirectional recurrent neural network [11]

replaced by memory blocks as depicted in Fig. 2. LSTM-RNN outperformed the state-of-the-art techniques for character or word recognition and language learning [9]. The activation of internal unit is controlled by input, forget and output gates. The recurrent connections of cells are controlled by the forget gate that makes RNN to hold the information as long as the forget gate is switched on.

To overcome the limitations of a regular RNN, Schuster and Paliwal [11] introduced the bidirectional recurrent neural network (BRNN) by implementing RNN in forward direction (from left to right) and in backward direction (from right to left) for maintaining long-range context information about past and future using bidirectional long short-term memory (BLSTM) [12]. The architecture of BRNN is depicted in Fig. 3. The BRNNs and LSTM are collectively called the idea of BLSTM.

2.3 Multi-dimensional recurrent neural network

Multi-dimensional recurrent neural network (MDRNN) has various recurrent connections as compared to the single-dimensional RNN that has a single recurrent connection. The MDRNN provides the functionality to apply operations on the multi-dimensional data. The simple LSTM is one dimensional explicitly, and its cell is composed of one connection with itself which activated by one forget gate. This one dimension extended into n-dimensions by replacing one self-connection with n self-connections with n-forget gates. For every dimension, the previous state of each of the cell has one connection as shown in Fig. 4. The MDRNNs and BLSTM in n-dimensions are collectively called the idea of MDLSTM.

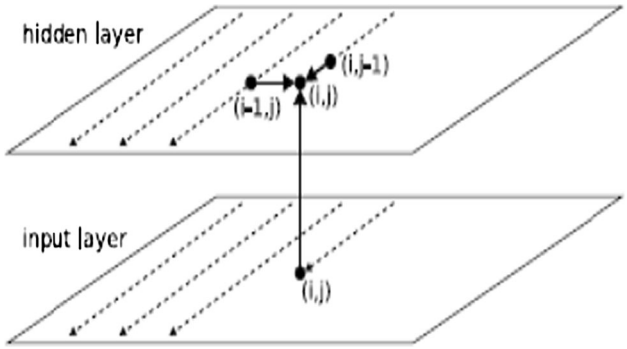


Fig. 4 MDRNN: 2D RNN forward pass [13]

Although BLSTM-RNN outperformed the state-of-the-art algorithm for Urdu character recognition, due to the complexity of Nasta'liq script, i.e., context variation, diacritical marks and character overlapping, MDLSTM could help to improve the accuracy.

The CTC is designed as output layer to label the sequences when there is no alignment between the sequences of input and output labels. Once the system is trained, the CTC chooses label for the unknown input sequence with high value using the conditional probability of label and input data. These models are widely used for modeling sequence of characters or words in text recognition, speech recognition, word spotting, attentive vision, stock market prediction, music composition and protein analysis.

3 Related work

The performance of character recognition is based on the holistic approach or an analytical approach in the segmentation stage. In holistic approach, the word or sub-word is considered for feature extraction, classification and recognition, while in the analytical approach the word or sub-word is segmented into characters or strokes for further steps. The analytical segmentation can be performed explicitly or by implicitly [14, 15]. The holistic approach outperformed than analytical approach for cursive scripts in the literature, but new trend enforces toward the implicit

segmentation of cursive script in particular, which only reads the text using sliding window to get pixels or features with predefined classes in the transcription.

Recent work for character recognition using implicit segmentation not only showed promising result for the character recognition and speech recognition of Latin script [10, 16–20] but also provided very good accuracy for Urdu script-based languages [4–21]. The extensive research has been available on hidden Markov model (HMM), recurrent neural network (RNN) or hybrid for sequential data transcription for different languages and different techniques like word/sub-word or character recognition in the literature. Even though features extraction is one of the bottlenecks for such kind of systems, the researchers are trying to design handcrafted features that show the structure of word/character shape and reduce the dimensionality for extraction automatic pixels-based features. Manual features bring together the topology and geometry of character shapes. To capture the properties of character's shape is explicitly difficult; therefore, different statistical and geometric measures are used to count specific patterns like foreground distribution [22], foreground density, count of pixels of foreground, and upper and lower profile of character shape in a frame [23], mean and variances of skeletonized characters [24, 25] and texture-based measures like contract, energy, correlation and homogeny [26–28]. Most of the existing Urdu OCR systems have been evaluated on custom-developed databases. This makes the quantitative comparison of different methods a difficult task. There has not been reported any work to show the performance of BLSTM-RNN or MDLSTM on manual features for Urdu Nasta'liq text up to our knowledge. We applied our method on the free available UPTI dataset [2] to provide benchmark results for font invariant and unconstrained text lines recognition as a first time using MDLSTM using statistical features and compare with the state-of-the-art techniques using handcrafted features vector.

Ahmad et al. [29] applied variable sliding window and explicitly segmented the words/sub-words into initial shape, medial shape, final shape and isolated shape of each character of Urdu alphabet set and developed 56 unique classes in total. The pixels strength is extracted from the segmented shape for training the feed-forward neural network (FFNN) on the 100 instances of each class. For recognition purpose, text line is used and reported 70 % accuracy rate for self-generated dataset for Urdu Naskh writing style. The size of dataset was not mentioned.

Morillot et al. [30] implemented bidirectional long short-term memory (BLSTM) and reduced the dimensionality for automatic feature using feature vector which have measures for background/foreground transitions, concavity configurations, gravity center position, directional features corresponding and density of pixels. They evaluated the proposed system on NIST handwritten Arabic dataset and reported 52 % recognition rate for word on test set having 12,644 text lines. Graves et al. [17] extracted mean of intensity, center of gravity, second-order vertical moment of the center of gravity, positions of the uppermost and lowermost black pixels, and rate of change with respect to the neighboring windows from the sliding window of offline handwritten English text lines, and BLSTM has been used for classification and recognition. The recognition rate reported up to 81.8 % on IAM-DB dataset.

Chherawala et al. [31] explored different features sets from the literature [22, 32, 33] on IFN/ENIT Arabic dataset using multi-dimensional LSTM and reported 81.1 % results for features reported in [10], 84.2 % results for features reported in [19], 77.6 % results for features reported in [20] and 88.8 % results for all combination of features. Lickwi et al. [17, 19] extracted speed, up and down position of up, hat features, curvature, writing direction, x and y coordinates, slope, aspect, curliness and linearity of vicinity, context map, ascenders/descenders for IAM-OnDB online English dataset using BLSTM and reported 74.4 % accuracy. Morillot et al. [34] also performed the experiment of BLSTM on Rimes handwritten French dataset. The author extracted feature vector of density of foreground, count of foreground/background transitions, its count between adjacent cells and above lower baseline, position and relative position of gravity center, difference of gravity center position next window, density of pixels above upper baseline and below lower baseline, and pixel density for each frame column. The recognition accuracy of characters is up to 43.2 %.

The literature of character recognition proved the worth of RNN performance over other machine learning approaches. RNN-based character recognition system provided excellent result not only for Latin script but also for cursive Arabic script. Due to the complexity of Arabic script, especially when written in Nasta'liq, We present MDLSTM recurrent neural network using geometric and statistical features. We have used sliding window consisted of four columns to extract number of geometric and statistical features. Extracted features are concatenated into feature vector and passed as an input to the RNN classifier for classification and recognition in an unconstraint size of test set and invariant font.

4 Urdu Nasta'liq character recognition

Even though the Urdu script is derivative of Arabic and uses the same character set as of Arabic, work done for Arabic script cannot be applied directly on Urdu script due

to the nature of Nasta'liq script. Recognition of Urdu Nasta'liq text is a challenging task as compared to Arabic Naskh due to the complexity of this script [35]. We presented recurrent neural network based on multi-dimensional long short-term memory using statistical feature as shown in Fig. 5. The proposed methodology consists of three stages: preprocessing and feature extraction, MDLSTM and CTC output layer.

The prior stage of MDLSTM is preprocessing and feature extraction. The sequences of characters in words/sub-word make text, and this sequential behavior of the text is regenerated from the text line grayscale images for text decoding. The sliding window approach is used to decompose the text line into sequence of characters and separated into a frame. The text line is transferred in a sequence of feature vector by extracting different numbers of features from the segmented frame using sliding window approach.

The second stage is the MDLSTM layered approach. The features vectors scan in nxn input block and couple with the corresponding transcriptional values. The final stage is the CTC architecture as an output layer for labeling the sequences. Once the system is trained, the CTC chooses label for the unknown input sequence with high value using the conditional probability of label and input data and recognizes the cursive Urdu Nasta'liq character.

4.1 UPTI dataset

Dataset plays a vital role in evaluating the performance of any pattern or character recognition systems. In supervised classification, class labels are needed to be constructed for data elements in the input space. This is known as ground truth or transcription. RNN is also a supervised learning model. It requires the ground truth values for each image in the input space for training the model.

UPTI dataset consists of 10,000 text lines, 771,339 frequently occurring character samples and 44 labels. We divided 10,000 text lines into three sets: training, testing and validation set with 6800, 1600 and 1600 text lines, respectively. Training, validation and testing sets consist of 644,354, 137,785 and 126,985 characters, respectively. The statistics of dataset are given in Table 1.

We have used 42 unique labels (38 basic characters with extra four common characters ("ں," "ؤ," "ھ" and "ئ" noonghuna, wawohamza, haai and yeahamza, respectively, one for "SPACE" and one extra label for the blank) for character level transcription. The transcription sample "علم بڑی دولت ہے" is as: "aain-laam-meem bay-array-yea dal-wawo-laam-the goalhau-Yea" which is used with its image as an input to the MDLSTM.

4.2 Features extraction

The aim of features extraction is to remove the unnecessary data from the sequences of characters and keep the useful and necessary information in the feature vector which will later load into the recognition engine. We have used right-to-left sliding window to extract features from normalized Urdu Nasta'liq text line images. The text lines are normalized to fix height, whereas the width is variable as per text line length. For features extraction, we have used sliding window of size 4 × 48 (width × height) from right to left and top to bottom by considering Urdu Nasta'liq language properties as shown in Fig. 6. Based on the sliding window, the text line is divided into number of frames having size 48 × 4 and computed the geometric/statistical features from each frame/window. The feature detail is presented in Table 2.

Features F_1 and F_2 are vertical and horizontal edges intensities. Sobel function is applied to compute two-dimensional gradient magnitudes at each point in each frame of the text lines to detect edges along row wise [25]. Then, the total numbers of intensities of extracted horizontal edges are counted and append the feature (F_1) to the feature vector. Likewise, vertical edges are calculated, then all the intensity values are counted, and feature (F_2) is concatenated to the feature vector (Figs. 7, 8).

F_3, the foreground distribution, is the total number of foreground pixels intensities counted for grayscale image that fall in each frame of the text lines as given in Eq. 1.

$$f_3 = \sum_{i,j}^{mn} p(i,j) \quad \text{if } p(i,j) > \theta \tag{1}$$

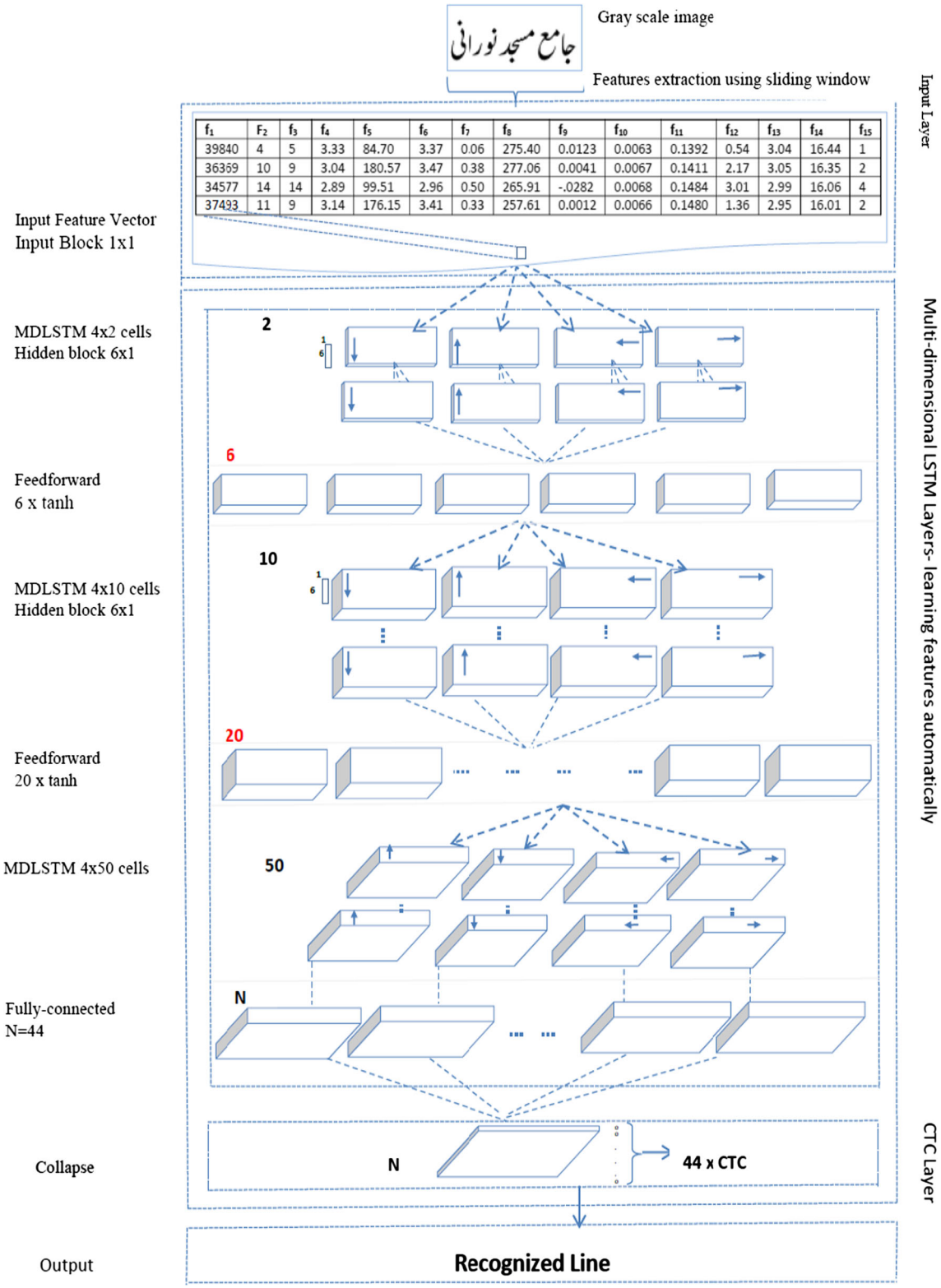
F_4, the density function, is the mean value of the text line foreground. The density function is the ratio of summation of pixels in the foreground in each frame divided by total size of frame and contented to the feature vector of the text line.

$$f_4 = \sum_{i,j}^{mn} p(i,j) / f_{\text{size}} \tag{2}$$

Intensity feature, F_5, is the sum of total numbers of intensity pixels that fall in each frame of text line images as in Eq. 3 and append to the feature vectors. It is called intensity features.

$$f_5 = \sum_{i,j}^{mn} p(i,j) \tag{3}$$

F_6 is the mean of horizontal projection that is calculated using summation of pixels intensities in each row in a frame. The variance is also computed for the summed horizontal projection (F_7) in each frame of the text lines. Similarly, the mean and variance of vertical projection are concatenated as features (F8) and (F9) into the feature vector.



f_1	F_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
39840	4	5	3.33	84.70	3.37	0.06	275.40	0.0123	0.0063	0.1392	0.54	3.04	16.44	1
36359	10	9	3.04	180.57	3.47	0.38	277.06	0.0041	0.0067	0.1411	2.17	3.05	16.35	2
34577	14	14	2.89	99.51	2.96	0.50	265.91	-.0282	0.0068	0.1484	3.01	2.99	16.06	4
37493	11	9	3.14	176.15	3.41	0.33	257.61	0.0012	0.0066	0.1480	1.36	2.95	16.01	2

Fig. 5 MDLSTM-based Urdu Nasta'liq character recognition system

GLCM is statistical measure used to characterize the image texture by calculating how often pairs of pixel occur in special specified relationship. Due to the importance of textural feature for complex object classification, we have extracted four texture features, namely contract, energy, correlation and homogeny.

For each frame, the contrast of intensity (F_{10}) is measured between a current pixel and its neighboring pixels as given in Eq. 4. F_{11} is computed on the gray-level intensities that are squared and then summed which measured the closeness or uniformity among the pixels distributions as given in Eq. 5. F_{12} is the angular second moment that is calculated from each frame from start of the text line to end which is called the homogeneity features of text in Eq. 6.

F_{13} is the correlation of the neighbors' pixels and is calculated in the frame of the text line image in Eq. 7.

$$f_{10} = \sum_{i,j}^{mn} |i-j|^2 p(i,j) \tag{4}$$

$$f_{11} = \sum_{i,j}^{mn} p(i,j)^2 \tag{5}$$

$$f_{13} = \sum_{i,j}^{mn} \frac{p(i,j)}{1+|i-l|} \tag{6}$$

Table 1 Statistics of dataset

Types	Training set	Validation set	Testing set	Total
Text line	6800	1600	1600	10,000
Characters	5,06,569	1,37,785	1,26,985	7,71,339

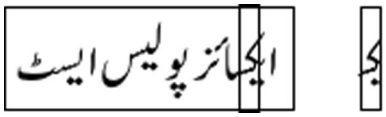


Fig. 6 Frame extracted using sliding window from Urdu Nasta'liq text line

$$f_{12} = \sum_{i,j}^{mn} \frac{(i-\mu i)(j-\mu j)p(i,j)}{\sigma_i \sigma_j} \tag{7}$$

The center of gravity F_{14} and F_{15} is calculated in *x*-direction and *y*-direction for each frame of the text line images and concatenated to the feature vector as given in Table 3.

4.3 MDLSTM recurrent neural network and its parameters

We presented multi-dimensional recurrent neural network (MDRNN) with architecture of multi-directional long short-term memory (MDLSTM), connectionist temporal classification (CTC) output layer and manual nxn-sized features vectors. For the optimal performance of the network, it is required to care in selection of values for network parameters and size of MDLSTM layers. In our experiment, MDLSTM uses 1 × 1 block structure to read the extracted feature vector of text lines for learning the character sequences as discussed in Sect. 2.2 and loads to the next higher layers for further processing such as hidden blocks sizes 6 × 1 and 6 × 1, subsample sizes 6 and 20 and hidden sizes 2, 10 and 50. These parameters are given in Table 4.

Tanh unit is used in the hidden layers of the MDLSTM as an activation function of input and output blocks. Logistic sigmoid is used as activation for gates. The CTC output layer has 44 nodes for 43 characters and one extra is for "blank." All the hidden layers are fully connected with each other, input and output layer, and has given 141,765 weights for printed Urdu Nasta'liq character recognition. The 141,765 uninitialized network weights are randomized uniformly in [−0.1, 0.1]. We trained MDRNN with steepest descent with 0.0001 learning rate and 0.9 momentum = 0.9. The training was stopped when

Table 2 Feature set description

Sr. No.	Feature	Feature description
1	F1: Vertical edges intensities	Total numbers of intensities of extracted vertical edges are counted
2	F2: Horizontal edges intensities	Total numbers of intensities of extracted horizontal edges are counted
3	F3: Foreground distribution	The total numbers of foreground pixels intensities
4	F4: Density function	The mean value of the foreground
5	F5: Intensity features	Sum of total numbers of pixels that fall in each frame of text line images
6–7	F6–7: Mean and variance of horizontal projections	Mean and variance of horizontal projections are calculated using summation of pixels intensities in each row
8–9	F8–F9: Mean and variance of vertical projections	Mean and variance of vertical projection are calculated using summation of pixels intensities in each column
10–13	F10–F13: GLCM features	Contract, energy, correlation and homogeny of each frame
14–15	F14: Center of gravity X and Y	Center of gravity is calculated for x-direction, y-direction

Fig. 7 Horizontal edges in the frame using sliding window

Fig. 8 Vertical edges in the frame using sliding window

the performance was not improving for 30 epochs as given in Table 6.

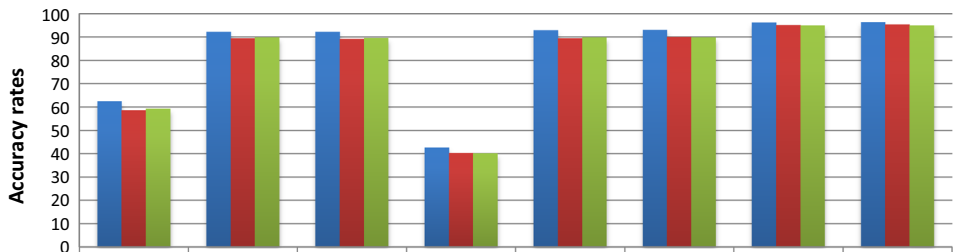
We have trained different networks on different features sets in order to find the optimal feature matrix for Urdu Nasta'liq characters as shown in Fig. 9. The different features sets for network training are discussed in Table 6. The network-1 trains on 48 × 2 features vector (F_1–F_2), network-2 trains on 48 × 7 features vector (F_3–F_9), network-3 trains on 48 × 4 features vector (F_{10}–F_{13}), network-4 trains on features vector 48 × 2 (F_{14}–F_{15}), network-5 trains on F_1–F_9, network-6 trained on F_1–F_{12}, and network-7 trains on F_1–F_{15}. Then, we combined best features such as intensity, foreground distribution, density function, mean and variance of horizontal and vertical projections (F_3–F_9), and texture-based GLCM (F_{10}–F_{13}) and trained the network-8 on 48 × 11 features vector. The training results of all networks are compared in Fig. 9. It is shown

Table 3 Example of feature vector extracted from the input image

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}	F_{14}	F_{15}
39,840	4	5	3.33	84.70	3.37	0.06	275.40	0.0123	0.0063	0.1392	0.54	3.04	16.44	1
36,369	10	9	3.04	180.57	3.47	0.38	277.06	0.0041	0.0067	0.1411	2.17	3.05	16.35	2
34,577	14	14	2.89	99.51	2.96	0.50	265.91	−0.0282	0.0068	0.1484	3.01	2.99	16.06	4
37,493	11	9	3.14	176.15	3.41	0.33	257.61	0.0012	0.0066	0.1480	1.36	2.95	16.01	2

Table 4 Different numbers of parameters for training the network

Parameters	Values	Horizontal sampling	Vertical sampling
Input block size	1 × 1	1	1
Hidden block size	6 × 1 and 6 × 1	1	6
Subsample sizes	6 and 20	–	–
Hidden sizes	2, 10 and 50	–	–
Learn rate	1×10^{-4}	–	–
Momentum	0.9	–	–
maxTests No Best	30	–	–
Total network weight	142,167	–	–



	Network-1 (F1-F2)	Network-2 (F3-F9)	Network-3 (F10-F13)	Network-4 (F14-F15)	Network-5 (F1-F9)	Network-6 (F1-F13)	Network-7 (F1- F15	Network-8 (F3- F13)
train-set	62.36	92.23	92.2	42.55	92.87	92.95	96.12	96.28
valid-set	58.58	89.38	89.16	40.14	89.48	89.96	95.1	95.3
test-set	59.19	89.78	89.58	40.03	89.84	89.87	94.88	94.97

Fig. 9 Comparison of different networks performance for different sets of features on training set, validation set and testing set

that network-8 achieved highest training accuracy of 96.28 % as compared to other networks. Network-7, network-6, network-5, network-2 and network-3 have achieved 96.12, 92.95, 92.87, 92.23 and 92.2 %, respectively. The network-1 and network-4 showed worse results. The performance of trained networks is depicted in Fig. 9.

Thus, based on the training accuracy, network-8 having features (F_3–F_{13}) is considered as the best network. The network-8 took 253 passes through the training set for learning the weights, and each pass has grown with number of weights in average time of 10 min. The training has more number of passes for convergence that could be due to value of "*maxTestsNoBest*" parameters of 30 and also due to rich morphology and large number of shapes for one class, which took 216 numbers of passes to converge and have achieved character classification accuracy rate up to

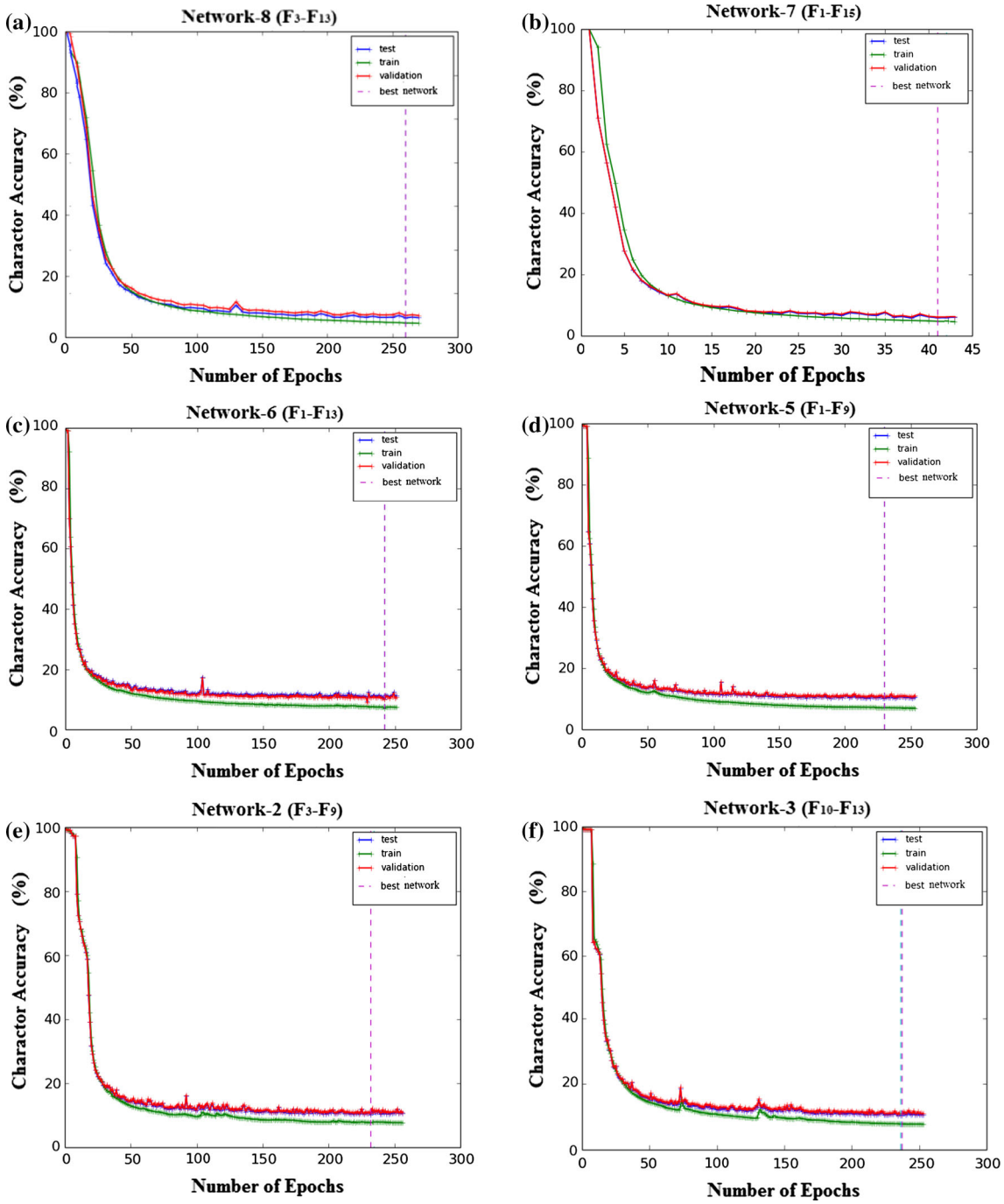


Fig. 10 Training performance of Urdu character recognition system using **a** network-8 trained on F_1–F_{12}, **b** network-7 trained on F_1–F_{12}, **c** network-6 trained on F_1–F_{13}, **d** network-5 trained on F_1–F_9, **e** network-2 trained on F_3–F_9 and **f** network-3 trained on F_{10}–F_{13}

96.28 %. The accuracy rate on validation set is 95.3 % and on test set is 94.97 %. The performance of trained network for best features vector for classification of Urdu Nasta'liq test lines is shown in Fig. 10. We have performed various numbers of experiments to assess the RNN classifier with architecture of sequence labeling algorithm CTC and MDLSTM for Urdu Nasta'liq classification and recognition. The proposed character recognition system (network-8) achieved 3.72 (training set) and 5.03 % (testing set) character error rates (Fig. 10).

5 Results and discussion

Most of the existing Urdu OCR systems have been evaluated on custom-developed databases. This makes the quantitative comparison of different methods a difficult task. We found text line recognition of Urdu Nasta'liq using a benchmark dataset UPTI in [2] and ligature recognition system using UPTI dataset reported in [36]. For comparison of our result on a benchmark dataset, we applied our method using features on the UPTI dataset [2]. There has not been reported any work to show the performance of BLSTM or MDLSTM on manual features for Urdu Nasta'liq text up to our knowledge. In evaluation of the presented system, the recognition is accomplished by using trained MDRNN model on specified test set(s) and provides benchmark results as a first time using MDLST using statistical features. Images of different sizes from the test sets are used. The input test images are normalized using aspect ratio. The character recognition of new text sequences from test set has been tested on each trained network models like network-1, network-2, network-3, network-4, network-5, network-6, network-7 and network-8. In Fig. 11, the input images are shown in a.1 and b.1 and the output texts are shown in a.2 and b.2. The different character recognition rates for each network are given in Table 5. It is shown that recognition rate of network-8 is 94.97 % which is higher than all other networks. The recognition rate of network-7 is 94.88 %, recognition rate of network-6 is 89.87, recognition rate of network-5 is 89.84, recognition rate of network-2 is 89.78, and then network-3 shows 89.58 % recognition rate. Table 6 shows different character error rates and recognition rate of characters on different epochs for network-8 on unseen test set. In order to evaluate the RNN classifier with architecture of sequence labeling algorithm CTC and MDLSTM, we have performed various numbers of experiments on different sets of features and selected the best features for training the MDLSTM with CTC output layer. The proposed character system on selected best features achieved 3.72 (training set) and 5.03 % (testing set) character error rates as a state-of-the-art result in the literature, and there is no work reported for Urdu Nasta'liq using MDLSTM-RNN and handcrafted features. Character recognition rates are 94.97 % on test set as given in Table 8.

In Fig. 11, the recognition results showed that character "yeahamza (ئ)" is substituted with the character "tee (ٹ)," whereas the character "yea (ی)" is deleted due to character overlapping diacritics and excess of diacritics in less space. The overlapping and space between characters made confusion for the recognition system. Likewise, (ڑ) is substituted with laam (ل) and character haai (ہ) deleted from the word. Moreover, there is also an insertion of a

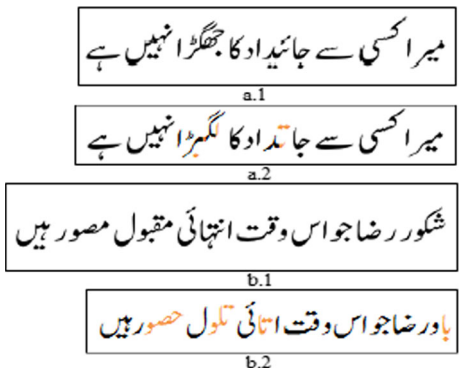


Fig. 11 Input and output of the MDLSTM recognition system for Urdu Nastaliq text: Input images are in figures a.1 and b.1 and output texts are in figures a.2 and b.2

Table 5 Comparison of networks for recognition rate on different features on test set

Networks	Features sets	Recognition rate (%)	Recognition error rate (%)
Network-1	F_1–F_2	59.19	40.81
Network-2	F_3–F_9	89.78	10.22
Network-3	F_{10}–F_{13}	89.58	10.42
Network-4	F_{14}–F_{15}	40.03	59.97
Network-5	F_1–F_9	89.84	10.16
Network-6	F_{10}–F_{13}	89.87	10.13
Network-7	F_1–F_{15}	94.41	5.59
Network-8	F_3–F_{13}	94.97	5.03

new character bay ( ). All these insertion, deletion and substitution have changed the original word into totally a new word having different meaning. The same cases also occurred in image b.1 where number of deletion of characters is three and number of substitution of characters is five which are shown in red color in Fig. 11b.2.

For generalization of recognition rate using MDSLTM on UPTI dataset, we also performed cross-validation scheme known as "*repeated random subsampling validation*." The dataset is shuffled randomly into splits of 68 % training, 16 % validation and 16 % test sets for five times, and experiments are performed for each split. The recognition results are then averaged over all five splits. In our experiments, we achieved average recognition rate of up to 94.97 ± 0.2 % as illustrated in Table 7. The confusion matrix of most problematic characters is given in Table 9.

Table 6 Character recognition rates for network-8 (F_3–F_{13}) on different epochs for testing set

Number of epochs	Character error rate (%)	Recognition rate (%)
50	15.09	84.91
100	5.69	94.31
150	5.56	94.44
200	5.12	94.88
254	5.03	94.97

Table 7 Recognition error rate for five experiments using cross-validation scheme

Experiments	Exp-1	Exp-2	Exp-3	Exp-4	Exp-5
Recognition error rate (%)	4.90	5.0	5.28	5.1	4.87
Average error rate (%)	5.03				

The direct performance comparison of proposed system is not possible with other systems reported in the literature for Urdu Nasta'liq script due to different techniques like holistic [2, 37, 38] or explicit segmentation [39], use of nonstandard dataset for training and testing evaluation or use of RNN on pixels values [36]. However, we are comparing the proposed system with Arabic character recognition system [31] using MDLSTM and features vector. Even though the writing style of Urdu Nasta'liq follows diagonality that shrinks the words or sub-word horizontally and introduces high overlap of intra-ligature and touching of dots or movement of dots from its original place, the results of proposed system are better as compared to Arabic recognition system [31].

Ahmed et al. [29] presented training feed-forward neural network on explicitly segmented Urdu Naskh of text into unique 56 shaped characters, and testing has been performed on Urdu Naskh text lines suing implicit segmentation approach. The dataset is author generated and size is not mentioned. The reported recognition rate is 72 % for Naskh Urdu script. Adnan et al. [36] performed two experiments characters of Urdu language and reported 88.2 % for shaped variation experiment and 94.8 % recognition rate for unshaped variation. Therefore, we implemented and evaluated our proposed system on unshaped characters of Urdu Nasta'liq. The indirect

Table 8 Comparison of proposed system's results with other systems

Authors	Segmentation approach	Recognition approach	Recognition rate (%)
Morillot et al. [30]	Implicit segmentation	BLSTM for handwritten Arabic	52
Morillot et al. [34]	Implicit segmentation	BLSTM for handwritten French	43.2
Graves et al. [17]	Implicit segmentation	BLSTM for online English on features	88.5
		Off-line English on features	81.8
Lickwi et al. [19]	Implicit segmentation	BLSTM for online English	74.4
Schuster and Paliwal [11]	Implicit segmentation	MDLSTM for handwritten Arabic on pixels	93.37
Chherawala et al. [31]	Implicit segmentation	MDLSTM for handwritten Arabic on features	81.1–88.8
Ahmad et al. [29]	Explicit segmentation	FFNN for printed Urdu Naskh on pixels strength	72
Adnan et al. [36]	Implicit segmentation	BLSTM printed Urdu Nasta'liq on pixels	86.4–94.85
MDLSTM (Proposed)	Implicit segmentation	MDLSTM for printed Urdu Nasta'liq on features	94.97

Table 9 Confusion matrix shows number of counts for mis-recognized characters for most frequent characters on test set

Actual label	SP	___	ٹ	ن	ی	م	ت	ن	ل	ب
Predicted label	___ (del)	SP (insert)	ن (subs)	___ (del)	___ (del)	___ (del)	ن (subs)	ٹ (subs)	ن (subs)	ہ (subs)
Counts	980	924	288	179	168	165	163	142	125	113

comparison of neural-network-based character recognition systems for different languages and dataset in [17, 19, 30, 34] with the proposed system is illustrated in Table 8.

We have performed various numbers of experiments to assess the RNN classifier with architecture of sequence labeling algorithm CTC and MDLSTM for Urdu Nasta'liq classification and recognition. The proposed character system achieved 3.72 (training set) and 5.03 % (testing set) character error rates. The proposed approach outperformed the state-of-the-art methods and provided 94.97 % recognition rate on test set as shown in Table 8. The confusion matrix as shown in Table 9, shows number of counts for misrecognized characters for most frequent characters on test set.

6 Conclusion

In this paper, the technique relies on multi-dimensional recurrent neural network (MDRNN and LSTM and CTC output layer) and statistical features. We present a robust feature extraction approach that extracts feature based on right-to-left sliding window. We have extracted 15 sets of features and trained different networks based on several features sets. The performance showed that selected features significantly reduce the label error. For evaluation purposes, we have used UPTI dataset and compared the proposed approach with the state-of-the-art work. The proposed system significantly outperforms the state-of-the-art Urdu character recognition system and RNN-based recognition systems by achieving training error 3.72 and 5.03 % recognition error.

References

1. Naz S, Razzak MI, Hayat K, Anwar MW, Khan SZ (2014) Challenges in baseline detection of arabic script based languages. Intell Syst Sci Inf 542:181–196
2. Sabbour N, Shafait F, Sabbour N, Shafait F (2013) A segmentation-free approach to Arabic and Urdu OCR. In: Proceedings of the SPIE international society for optics and photonics, vol 86580, p 86580 N
3. Graves A (2013) RNNLIB: a recurrent neural network library for sequence learning problems. http://sourceforge.net/projects/rnnl/
4. Ul-Hasan A, Ahmed SB, Rashid F, Shafait F, Breuel TM (2013) Offline printed Urdu Nastaleeq script recognition with bidirectional LSTM networks. In: 12th International conference on document analysis and recognition (ICDAR'13), pp. 1061–1065
5. McCulloch WPWS (1990) A logical calculus of the ideas immanent in nervous activity. Bull Math Biol 52(1–2):99–115
6. Rosenblatt F (1961) Principles of neurodynamics: perceptrons theory brain mechanism. No. VG-1196-G-8. Cornell Aeronautical Lab Inc.
7. Jaeger H (2002) Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the 'echo state network' approach. GMD Rep 159, Ger Natl Res Cent Inf Technol, p 48
8. Hochreiter J, Schmidhuber S (1997) Long short-term memory. Neural Comput 9(8):1735–1780
9. Schmidhuber J, Gers FA (2001) LSTM recurrent networks learn simple context free and context sensitive languages. IEEE Trans Neural Netw 12(6):1333–1340
10. Graves A (2012) Offline arabic handwriting recognition with multidimensional recurrent neural networks. Springer, London
11. Schuster M, Paliwal K (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45:2673–2681
12. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw 18:602–610
13. Graves A, Fernández S, Schmidhuber J (2007) Multidimensional recurrent neural networks. In: Proceedings of the international conference on artificial neural networks
14. Naz S, Umar AI, Shirazi SH, Ahmed SB, Siddiqi I, Razzak MI (2015) Segmentation techniques for recognition of Arabic-like scripts: a comprehensive survey. Educ Inf Technol 20(2). doi:10.1007/s10639-015-9377-5
15. Naz S, Hayat K, Razzak MI, Anwar MW, Madani SA, Khan SU (2013) The optical character recognition of Urdu-like cursive scripts. Pattern Recognit 47(3):1229–1248
16. Nishide S, Okuno HG, Ogata T, Tani J (2011) Handwriting prediction based character recognition using recurrent neural network. In: IEEE international conference on systems, man, and cybernetics (SMC), pp 2549–2554
17. Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31:855–868
18. Graves A, Schmidhuber J (2009) Offline handwriting recognition with multidimensional recurrent neural networks. International conference on neural information processing systems, pp 545–552
19. Liwicki M, Graves A, Bunke H, Schmidhuber J (2007) A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. Proc 9th Int Conf Doc Anal Recognit 1:367–371
20. Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. Icassp 3:6645–6649
21. Ahmed SB, Naz S, Swati S, Razzak MI, Khan AA, Umar AI (2015) Ucom offline dataset: a Urdu handwritten dataset generation. Int Arab J Inf Technol 12(5)
22. Rath TM, Manmatha R (2003) Features for word spotting in historical manuscripts. In: Proceedings of the seventh international conference on document analysis recognition, 2003
23. Al-Hajj Mohamad R, Likforman-Sulem L, Mokbel C (2009) Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31(7):1165–1177
24. Khorsheed MS, Al-Omari HK (2014) System and methods for arabic text recognition based on effective arabic text feature extraction. IS Patent No US 20140219562 A1
25. Khorsheed MS (2007) Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK). Pattern Recognit Lett 28 (12):1563–1571
26. Naeimizaghiani M, Abdullah SNHS, Bataineh B, PirahanSiah F (eds) (2011) Character recognition based on global feature extraction. In: International conference on electrical engineering and informatics (ICEEI), pp 1–4
27. Singla L, Singh S (2014) Offline handwritten devanagari numerals recognition using GLCM features and neural networks. Int J Eng Res Technol 3(6):25
28. Bharathi VC, Geetha MK (2013) Segregated handwritten character recognition using GLCM features. Int J Comput Appl 84 (2):1–7

29. Ahmad Z, Orakzai JK, Shamsher I (2009) Urdu compound character recognition using feed forward neural networks. In: Proceedings of the 2nd international conference on computer science and information technology (ICCSIT'09), pp 457–462
30. Morillot O, Oprean C, Likforman-sulem L, Mokbel C, Chammas E, Grosicki E, Paristech IMT, Ltci C (2013) The UOB-telecom Paristech Arabic handwriting recognition and translation systems for the openhart 2013 competition. NIST-openhart Workshop, Washington
31. Chherawala Y, Roy PP, Cheriet M (2013) Feature design for offline Arabic handwriting recognition: handcrafted vs automated. In: Proceedings of the 12th international conference on document analysis and recognition (ICDAR)
32. Marti UV, Bunke H (2000) 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):6–90
33. Azeem SA, Ahmed H (2013) Effective technique for the recognition of offline Arabic handwritten words using hidden Markov models. Int J Doc Anal Recognit 16(4):399–412
34. Morillot O, Likforman-Sulem L, Grosicki E (2013) New baseline correction algorithm for text-line recognition with bidirectional recurrent neural networks. J Electron Imaging 22(2):023028
35. Naz S, Hayat K, Razzak MI, Anwar MW, Akbar H (2013) Arabic script based character segmentation: a review. In: World congress on computer and information technology (WCCIT'13), pp 1–6
36. Ul-Hasan A, Bin Ahmed S, Rashid F, Shafait F, Breuel TM (2013) Offline printed urdu nastaleeq script recognition with bidirectional LSTM networks. In: Proceedings of the international conference on document analysis recognition, ICDAR, pp 1061–1065
37. Javed ST, Hussain S, Maqbool A, Asloob S, Jamil S, Moin H (2010) Segmentation free Nastalique Urdu OCR. Word Acad Sci Eng Technol 46:456–461
38. Akram QUA, Hussain S, Niazi A, Anjum U, Irfan F (2014) Adapting tesseract for complex scripts: an example for Urdu Nastalique. In: Proceedings of the 11th IAPR international workshop on document analysis systems, pp 191–195
39. Javed ST, Hussain S (2013) Segmentation based Urdu Nastalique OCR, Lecture notes in computer science, vol 8259, pp 41–49