CURSIVE HANDWRITTEN TEXT RECOGNITION USING RECURRENT NEURAL NETWORKS

Aejaz Farooq Ganai[1*], Nasir Nabi[2]

Article History: Received: 5 Feb 2021 Revised: 28 Oct 2021 Accepted: 08 Dec 2021

Abstract

The handwritten Urdu recognition system is the process of reading characters or numbers that have been written on paper and provided as input in the form of a scanned digital image. The recognition of handwritten Urdu offers abundant applications in widespread domains. Big data of offline handwritten Urdu literature written by many different Urdu authors can be made available online, with the help of this offline OCR tool. Offering a variety of applications and convenience to differently-abled people motivates us to pursue our research on the development of the Urdu OCR system. The state-of-the-art approaches employ Convolutional Neural Networks (CNN) for the recognition of handwritten Urdu text. However, in these approaches, it becomes difficult and time-consuming to recognize a long sequence of handwritten Urdu text input because CNN's can't learn long-term dependencies (remembering previous information). Hence to recognize a long sequence of handwritten Urdu text input, an LSTM based Recurrent Neural Network (RNN) model has been proposed to recognize any unconstrained handwritten Urdu text. The proposed approach relies on a holistic approach where features are extracted from handwritten Urdu ligature images using the RNNs and these features train multi-dimensional LSTM-based RNNs. The trained MDLSTM-based RNN is used for the classification and recognition of unconstrained handwritten Urdu text because recurrent neural networks are better able to capture the long-term dependencies in the input sequences. We have employed the benchmark handwritten Urdu dataset UNHD as well as the newly proposed handwritten Urdu dataset UHLD for feature extraction as well as training the LSTM-based RNN model and the trained model recognizes any unconstrained handwritten Urdu text with a better recognition rate of nearly 94% on a large handwritten Urdu ligature dataset of 1500 cluster classes and hence outperforms the state-of-the-art accuracy of 92.75% for handwritten Urdu recognition.

Keywords: OCR, Handwritten Urdu, Recurrent Neural Networks, UNHD

[1*]E&C Engineering Department, NIT Srinagar, J&K India, gaejaz@gmail.com

***Corresponding Author:** Aejaz Farooq Ganai
Email: gaejaz@gmail.com

DOI: - 10.53555/ecb/2021.10.4.15

1. Introduction

Text recognition means recognition of a text written in a particular language. Text recognition may be defined as the method of converting text on an image to editable text. The input text can be in the form of PDFs that are scanned as images, image files containing text (e.g., jpg, png, etc.), or scanned documents such as printed/handwritten text documents. The output of text recognition is an editable/recognized text. The importance of text recognition can be imagined in the fact that it saves a huge amount of time needed for manual typing of text and the editable text obtained can be used for several word processing applications. Text recognition saves a lot of storage needed for storing images comprising text. Text recognition converts handwritten notes to editable texts and documents for easy storage and retrieval. Text recognition is one of the largest applications of natural language processing [1] and has gained much importance after the emergence of various machine learning tools and techniques [2]. In the last decade, there is a rapid improvement and success in the recognition of text written in many different languages using these tools and techniques [2]. Optical Character Recognition (OCR) [3] is a technical term for text recognition. OCR is a set of processes that converts images of printed or handwritten data into editable text. The data from various image sources such as scanned documents, pictures, and scene-text images (for example advertising boards, road signs, etc.) is recognized using this OCR process. The OCR software has been developed for numerous applications such as automatic number plate recognition, passport recognition, information retrieval, traffic sign recognition, data entry for business documents, fast conversion of scanned text documents into actual editable text, language recognition, and many more domain-specific applications.

The OCR process aims to recognize the characters written on an image. These characters may be written in any particular language. The foundation of OCR has been dated to the 90s when the retina scanner's key invention generates the pioneering idea of character identification [4]. After *Nipkow* invents a sequential scanner [4], a significant advancement was made in the development of OCR in 1890, which led to modern TV and reading devices. OCR was viewed as an aid for people who were blind in their early years. But with time, it developed into a limitless scope for creative work. The first documented instance of an optical character recognition framework dates to Germany in 1929, according to *Tauschek's* patent documentation [5]. *Tauschek* later obtained a US patent in 1935 for developing this OCR that helps blind people to read. Later on, Paul Handel was publicly acknowledged in 1933 [6] for developing the first statistical machine. After the development of the Turing machine [7], advanced versions of OCR came into existence. Based on the availability of OCRs for commercial use, the OCRs have been divided into four generations. The first generation of OCR is the business OCR framework reflected in the period of the 1960s. This generation of OCR was dominantly categorized for oblique character reading. However, with time, the multi-font characters showed up. The second generation of OCRs was noticeably better as they could read both handwritten and machine-printed characters. The second generation of OCRs was available from the middle of the 1960s through the middle of the 1970s and could read numbers as well as characters. The primary OCR system of the second generation was IBM 1287, a hybrid system that combined analog and digital technology [8]. Further research was conducted to enhance the recognition of print-based scanned documents, which led to the development of the third generation of OCR. The researchers work with a large set of handwritten characters that are of worse print quality than before. The fourth generation of OCR is suitable for filtering and detecting characters from complex texts such as unconstrained text, poor quality documents, photocopies, fax, colored documents, etc. In this generation, there has been very significant research in the development of OCR for recognizing text written in Latin scripts such as English and Chinese, as well as cursive scripts such as Arabic and Urdu scripts. Tesseract [9] was the first open-source text recognition engine that can be used directly to recognize printed text on an image. The initial version of Tesseract OCR recognizes only English-language text. However, in version 2, many different languages such as French, Italian, German, Spanish, Brazilian, Portuguese, and Dutch were added [10]. The version-3 of Tesseract OCR provides support for both ideographic languages such as Chinese and Japanese, as well as right-to-left languages such as Arabic. Currently, Tesseract OCR provides support for more than 100 languages [11] including the Urdu language.

The general diagram of the OCR process (as shown in Fig 1) consists of an input unit, a processing unit, and an output unit. The input consists of an image with text or a scanned/printed/handwritten document or a PDF document to be read by the OCR process. The processing is performed by the OCR unit. The OCR module consists of several submodules for carrying out various operations to convert the input image into an editable text in the output unit.

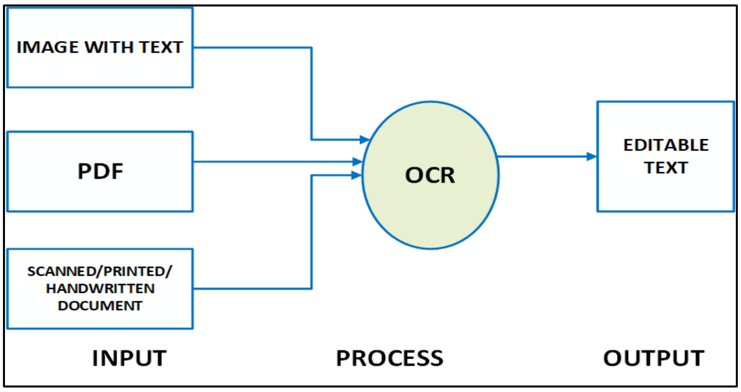


Fig 1. OCR process

The detailed processes in the OCR module are shown in Fig 2 and consist of two phases: the training phase and the testing phase. In the training phase, a dataset of images is collected from different sources. The dataset may be a set of scanned/printed/handwritten documents or text documents such as PDF or images comprising text. Initially, a pre-processing operation is performed on this dataset to remove various types of noise and skew from images. The pre-processing cleans the image and normalizes it to a constant and uniform size. The pre-processing is followed by the segmentation of images (comprising text) into separate text lines and each text line is segmented into words/characters. The features are extracted from these words/characters using various machine-learning tools and techniques. These features train the model and this trained model is thereafter used for the recognition of similar types of images. In the testing phase, the input testing image undergoes pre-processing followed by segmentation and feature extraction in a similar way as in the training phase. These sets of features are input to the trained model for recognition of each word/character of the testing image. The recognized text may undergo a post-processing phase such as re-association of some parts of the word or character. Hence the OCR process reads and recognizes the text in an image and converts it into editable text.

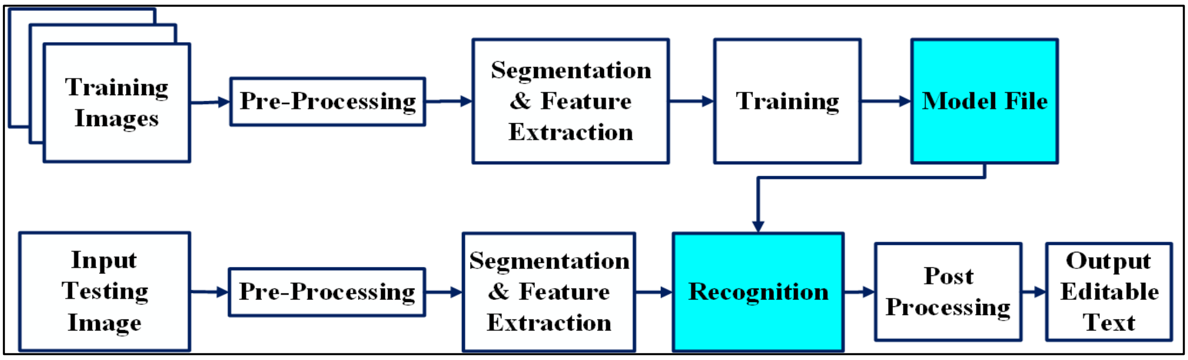


Fig. 2. OCR process in detail

The key phase in the development of the OCR process (Fig.2) is the segmentation phase for a particular script because the segmentation of a text line into words followed by the separation of each word into characters determines the complexity in the development of OCR for that particular script. The complexity increases if the characters are connected in a word while writing the script. The OCR for various Latin scripts such as Roman, English, French, and Chinese script is simple to develop because in their writing script, the letters of a word are not joined and there is a clear white space between letters as well as consecutive words [12]. Hence, the OCR for such languages is developed and commercially available [12]. However, for various cursive scripts such as Urdu, Arabic, Kashmiri, Pashto, Kurdish, Gurmukhi, and Persian script, the recognition of text on an image is a complex problem and poses several challenges [13] such as connected characters in a word, no clear/definite spacing between words, variation in the shape of a character at different positions in a word and presence of a rich variety of dots associated with characters. Hence cursive text detection and recognition is still regarded as a difficult, complex, and unsolved topic despite numerous advancements since the last decade.

After the emergence of various machine learning tools [2], there has been a significant improvement in the development of OCR for cursive languages during the last decade [14]. The recognition of Arabic script has been successfully achieved with great advancements because in Arabic script the ligatures don't overlap each other and there is a clear spacing between words/ligatures [] that makes segmentation of words into ligatures comparatively easier than Urdu script. Hence OCR for Arabic script is developed and commercially available [16], but the recognition of Urdu script offers supplementary challenges to researchers such as the diagonal nature of the script, overlapping of ligatures, and uneven spacing between ligatures/words. However, it has been revealed from the literature survey on Urdu recognition (discussed in chapter 2) that a lot of significant research has been performed in the recognition of printed Urdu [14, 15] since the last decade. Various OCR software is available for printed Urdu such as Nanonets [17], AKhar 2016 [17], Sakhr OCR [17], Tesseract OCR [17,18], Document AI [17], and Microsoft Azure [17]. However, the best Urdu OCR software per the quality and precision is Nanonets [17].

In comparison to printed Urdu, little or no work has been performed in the recognition of handwritten Urdu. The main reasons behind this meagre work in the recognition of handwritten Urdu are the absence of a standard handwritten Urdu dataset that can cover most of the ligature corpus of Urdu text, variation in writing styles of different Urdu writers, variation in writing style based on age, gender and profession, etc. and many more complexities and challenges in handwritten Urdu recognition system. The following section discusses these challenges in detail.

The generalized challenges encountered in the development of handwritten Urdu OCR include:

a) Extreme stroke variation and ambiguity from author to author: there is clear variation in writing styles between different authors belonging to the same field or different fields.
b) Each writer's handwriting style is inconsistent and changes from time to time as well as with age.
c) Deterioration in the original document's or image's quality over time.
d) The printed text is written in a straight line whereas handwritten text usually fluctuates while writing on plain paper. There are chances of the presence of skew among text lines written in a handwritten document which further enhances the difficulty in its recognition.
e) Diacritic identification and separation are difficult in handwritten Urdu. It can be seen in Fig. 3 that the identification of the no. of diacritics and their separation from primary components is difficult. The difficulty enhances in the case of five, six, and seven-character ligatures.
f) The collection and formulation of a labeled handwritten Urdu dataset for learning is also a challenge because there is no uniformity in writing Urdu script.
g) The diagonality of the Urdu script, ligature overlapping, and uneven spacing between ligatures. These challenges become a barrier in the segmentation of Urdu text lines into words/ligatures and each word/ligature into characters.
h) The context sensitivity of characters at different positions in a ligature. The shape of a handwritten Urdu character varies concerning its position in a word/ligature. It has been analyzed that in Urdu script, some characters may have more than 60 different conceivable shapes [19]. Hence this non-uniform shape of characters in a script makes its recognition a complex task.

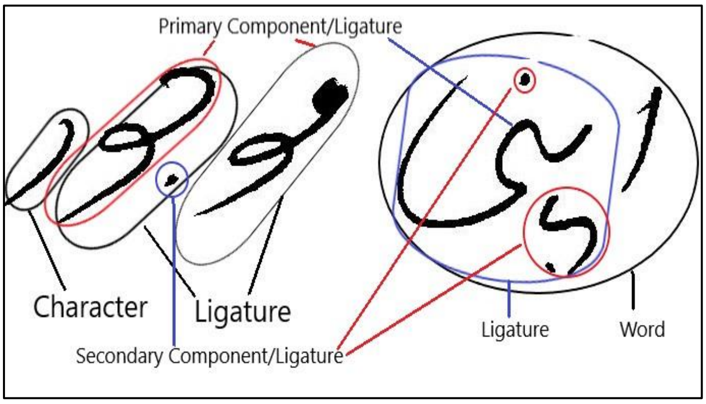


Fig. 3 Handwritten Urdu Ligature Components

We are motivated to pursue our research in the recognition of handwritten Urdu based on the applications and the challenges offered by handwritten Urdu recognition.

2. Related Work

The recognition methods for Urdu script are generally classified into analytical [10] and holistic approaches [11]. Characters are used as recognition units in analytical or segmentation-based approaches. The analytical approaches are classified into implicit segmentation [17] and explicit segmentation [18] approaches. It is extremely challenging to divide the Urdu script into individual characters [4]. Partially words or Ligatures, which are the longest connected parts of a word, are used as units of recognition in holistic or segmentation-free techniques [19, 11]. These methods are based on the recognition of the entire ligature and do not require character-level segmentation as explained in [7]. However, holistic methods require a large number of distinct ligature classes to be recognized. Although there are approximately 26000 ligatures in Urdu [7], the majority of them are rarely used. In addition, many Urdu ligatures have the same shape and only differ in the number and position of diacritics. As a result, when only frequent ligatures are taken into account, the number of classes significantly decreases to just a few thousand unique ligatures, which represent nearly 99 percent of the Urdu corpus [7]. Hence we have attempted to propose a holistic approach to handwritten Urdu recognition. The main contributions to the recognition of handwritten Urdu have been discussed below.

In 2014, Saad et al. started studying handwritten Urdu [20] and generated a new dataset for the Urdu language written in the Nastaliq style called UCOM Dataset. 100 author contributed 6400 text lines to the UCOM dataset. The authors added 300 authors to their database and assigned each user a unique label. This dataset was expanded to include 500 writers later in 2018 [16] and renamed Urdu Nasta'liq Handwritten Dataset [UNHD]. Using a one-dimensional BLSTM classifier for training and classification, the authors found an approximate error of 6.04–7.93 percent.

Mujtaba et al. [21] used CNN to recognize and classify handwritten Urdu characters at the beginning of 2019. A brand-new dataset of handwritten Urdu characters and numerals was created by these authors, who claimed an accuracy of 96% for the character classification and 98% for the numeral classification. Their work can't be applied to real-world handwritten Urdu text recognition because it only recognizes isolated characters and numbers.

Hassan et al. [22] created the non-public Urdu handwritten database in the middle of 2019 with 6000 text lines, seven convolutional layers for feature extraction, and two BLSTM layers for classifying and recognizing Urdu characters. Even though the authors looked into their work on handwritten character recognition from several thousand text lines, their recognition rate was only 83 percent.

Faisal et al [23] 2021 created Handwritten Urdu Character Dataset (HUCD) for Nastaliq's writing style and covering isolated positional characters and numerals. They employed CNN with automatic feature extraction and produces efficient results in comparison to conventional manual feature extraction ways. The proposed technique was trained on a dataset of 74285 samples and evaluated on a dataset of 21223 samples and achieved a recognition rate of 98.82% for 133 classes. However, their approach remains limited to isolated characters and numbers.

Another strategy for utilizing the transfer learning experience of similar patterns on handwritten Urdu text examination was proposed in December 2019 by Saad et al. [8]. By applying its previous learning to UNHD samples, the authors made use of a pre-trained network from MNIST. CNN was used to extract the features, and deep multidimensional long-short-term (MDLSTM) memory networks were used for the experimental evaluation. Even though the authors claim a character accuracy of roughly 93%, their work is limited to the UNHD database, which does not account for the significant variation between other datasets, nor does it investigate the recognition of six and seven-character ligatures.

The summarised literature work related to handwritten Urdu recognition is shown in Table 1. The literature survey reveal that a minor work has been performed in the recognition of handwritten Urdu as presented in Table 1. it is revealed that the existing approaches to handwritten Urdu recognition are mostly analytical approaches ([27], [69], [70], [71]) with lesser recognition rates and huge computational complexity. These approaches don't recognize unconstrained handwritten Urdu text. Moreover, it is difficult to recognize a long sequence of handwritten Urdu text input simultaneously using convolutional neural network (CNN) based approaches as CNNs can't learn long-term dependencies (remembering previous information [14]).

Table 1. Literature Survey on Handwritten Urdu text recognition

Study	Database	Training Dataset	Test Dataset	Framework Used	Unit of Recognition	Error Rate	Critics
Ahmad S.B. et al [30]	UCOM	50 text lines	20 text lines	Proposed dataset only			The only dataset is proposed.
Ahmad S.B. et al [27]	UNHD	6400 Text Lines	1840 text lines	BLSTM Classifier	Character and hence Analytical approach	6-7%	The proposed dataset is constrained
Husnain M. et al [69]	Proposed a database of Urdu Characters.	Information unavailable	Information unavailable	CNN classifier	Individual Characters & Numerals	4% on Characters 2% on Numerals	The dataset consists of characters and numerals only.
Shehbaz et al, 2019 [70]	Claimed their Database is not publicly available	4000 handwritten Lines	1000 text lines	7 CNN layers and 2 BLSTM layers	Character and hence Analytical approach	17%	Dataset not available, the huge error rate in proposed work.
Ahmad S.B. et al [71]	UNHD, MNist Pre-trained Network	10000 Text Lines	2000 Text lines	CNN and MDLSTM	Character and hence Analytical approach	8%	The proposed work is evaluated on only constrained handwritten Urdu text.

Hence we have used LSTM-based RNNs [13] for the classification and recognition of unconstrained handwritten Urdu text because recurrent neural networks are better able to capture the long-term dependencies in the input sequences [14]. A long sequence of handwritten Urdu text input can be simultaneously recognized using the LSTM-based RNNs without increasing the network size. We have employed our handwritten Urdu dataset UHLD as well as benchmark Urdu dataset UNHD [16] for training the LSTM-based RNN model. The proposed approach is computationally efficient and recognizes a long sequence of handwritten Urdu text with a better recognition rate than the state-of-the-art approaches [8, 9, 15]. The proposed research approach encompasses the following objectives.

1) The research proposes a new RNN based holistic handwritten Urdu recognition model that can recognize Urdu ligatures/words up to the six-character length.
2) The proposed approach recognises a large sequence of handwritten Urdu text of length up to 34 ligatures.

3. Proposed Methodology

As stated that becomes difficult and time-consuming to recognize a long sequence of handwritten Urdu text input because convolutional neural networks can't learn long-term dependencies (remembering previous information [92]) and ii) manually identify and classify ligatures into different length ligature clusters. Hence in this approach, to recognize a long sequence of unconstrained handwritten Urdu text input, an RNN model has been proposed wherein LSTM-based RNN recognize any unconstrained handwritten Urdu text. The proposed work also relies on a holistic approach where features are extracted from handwritten Urdu ligature images using the RNNs and these features train multi-dimensional LSTM-based recurrent neural networks (RNNs) as shown in Fig 4. The trained MDLSTM-based recurrent neural network uses recurrence for the recognition of a long sequence of unconstrained handwritten Urdu text as recurrent neural networks are better able to capture the long-term dependencies in the input sequences [92]. A long sequence of handwritten Urdu text input can be recognized using the proposed LSTM-based RNN framework without increasing the network size. In this approach, we have employed the benchmark-constrained handwritten Urdu dataset UNHD [27] as well as the proposed unconstrained handwritten Urdu dataset UHLD for feature extraction as well as training the LSTM-based RNN model and the trained model aims to recognize any unconstrained handwritten Urdu text with a better recognition rate than CNN based approaches. Also these approaches, it was very time-consuming to manually identify and classify ligatures into different length ligature clusters, hence an algorithm that estimates the number of characters in a ligature has been proposed in this approach. The algorithm classifies ligatures into different length ligature clusters and hence saves a huge amount of time needed for manual ligature class construction.

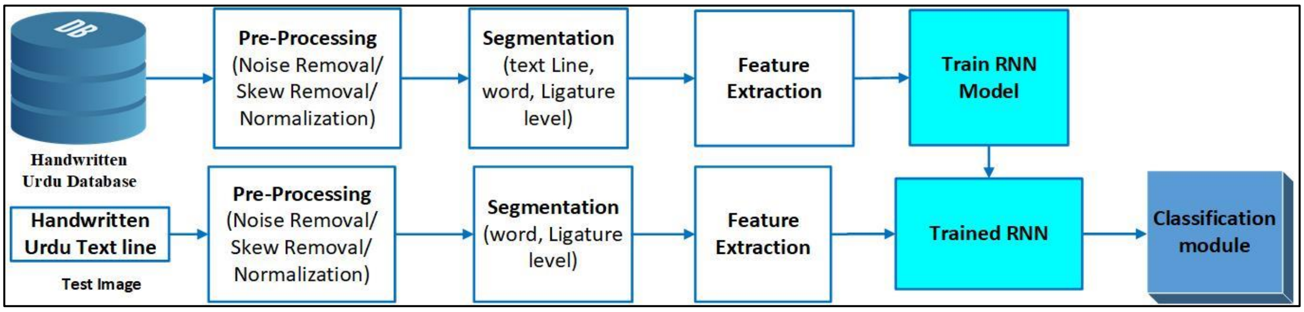


Fig. 4 Proposed approach of handwritten Urdu text recognition

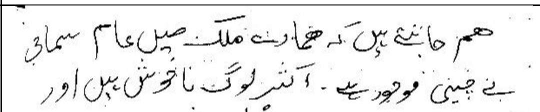
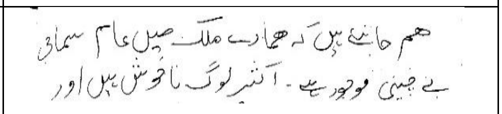
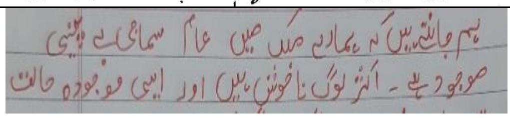
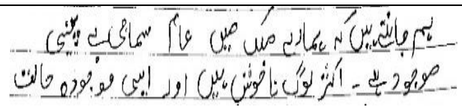
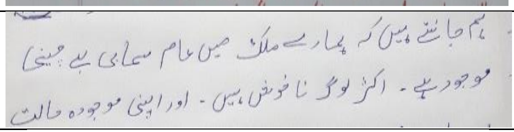
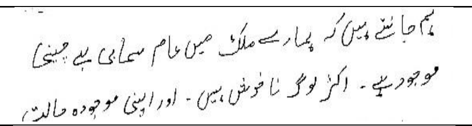
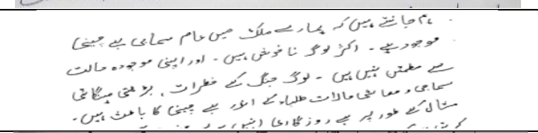
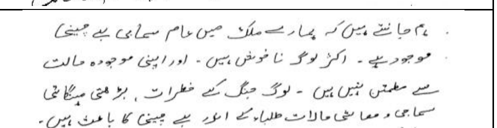
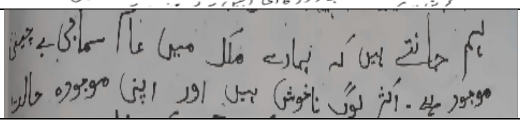
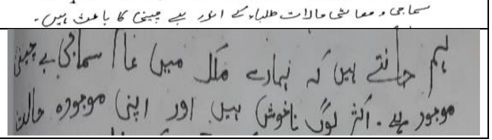
3.1 Dataset Acquisition

While working on handwritten Urdu text recognition, some researchers have proposed their datasets such as UNHD by Saad Bin Ahmad et al. [16] (publicly available) and HUCD by Faisal et al. [23] (publicly unavailable). After acquiring the UNHD database and conducting an analysis, we discovered that, despite its size, the dataset contains ligatures of length only up to five characters, and there are insufficient handwritten Urdu ligatures to cover the majority of the Urdu ligature corpus. Although UNHD claims to have ten thousand text lines, only seven hundred of those lines contain unique semantic content. As a result, the UHLD handwritten Urdu dataset that we have proposed has also been utilized for the experimental evaluation in our proposed work. The UHLD contains ligatures of lengths up to seven characters. The UHLD is unconstrained because it is written on any color of paper, any type of paper (blank or ruled), any color of ink, or any type of pen.

3.2 Pre-Processing the dataset

The main steps involved in pre-processing include de-skewing of handwritten Urdu text images, noise reduction, evenness, and normalization of images. The UNHD [16] dataset was acquired as separate handwritten Urdu text line images without skew. The type of noise visible from UNHD [16] and UHLD handwritten images was usually structural noise that consists of superimposed objects such as lines, marks, etc. We have removed this structural noise using the deep learning process [25]. To eliminate speckle noise in handwritten Urdu image and to binarize it., we have used the 'Thresholding' [26] algorithm as shown in Table 2.

Table 2. Pre-processing of handwritten Urdu text.

Pre-processing Technique	Input handwritten Urdu text image	After pre-processing operation
Noise removal [32]		
Thresholding [73] without noise removal		
Thresholding [35] followed by noise removal		
De-skewing [74] at an angle of +21°		
Smoothing [37]		

3.3 Segmentation

The handwritten Urdu database collected from UNHD [16] and UHLD datasets is in the form of separate text lines. Each text line is divided into words and each word has been segmented into ligatures using the algorithm proposed by Ganai et al [27] and Rehman et al [28]. A ligature is the longest connected component in a word that can be separated from another ligature using [11, 19, 29]. However, it must be noted that segmentation of a ligature into characters is a difficult and challenging task [7, 29] for handwritten Urdu, as characters vary their shapes depending upon their position in a word (context-sensitive nature of Urdu script [4]) and irregular position of diacritics associated to a character. Hence segmentation is performed up to the ligature level only and ligature forms the basis of recognition in the proposed recognition system. The algorithm used in [15, 28] has been used here to extract ligatures from handwritten Urdu text lines taken from benchmark handwritten Urdu dataset UNHD [16] as well as our proposed UHLD. With repetitions, we have nearly extracted 50000 ligatures from the dataset UNHD and 100000 ligatures from the dataset UHLD. These ligatures are associated with secondary components (dots/diacritics) also. To process this huge dataset of ligatures (nearly 150000 ligatures) for feature extraction using RNNs, we have to construct a large number of unique ligature cluster classes. However, the number of ligature classes can be reduced to only a few thousand classes if the secondary components are separated from primary ligatures. It is because most of the Urdu ligatures extracted resemble basic shapes and are differentiated in only the number and position of diacritics [7]. To perform this diacritic separation on a huge handwritten Urdu dataset, we have used the diacritic estimation and separation algorithm proposed by Aejaz et al [43], which separates primary ligatures from secondary components with a reasonable rate of accuracy.

3.4 Ligature Class construction

Most of the primary ligatures obtained after diacritic separation in the segmentation process, have similarities in shape and can be mapped to a single class/cluster of ligatures having the same ligature length as presented in [43]. However, it is a time-consuming process to find the different length ligature classes manually as discussed in [43]. Hence, in this research, we have used an algorithm that automatically determines the no. of characters in a ligature. Using this algorithm, it is easy to classify a ligature into a particular length cluster that forms the basis of different length ligature class construction.

3.5 Normalization

It was very time-consuming to normalize every ligature from a set of nearly 100 thousand ligatures. However, after ligature class construction, normalizing a particular length cluster is easier than the entire ligature set. Hence we have performed normalization of ligatures after generating the different length ligature clusters. The ligatures extracted may contain noisy points or gaps within strokes which is caused by the addition or loss of the data while performing line segmentation and ligature extraction phase. A noise filtering operation is performed on these ligatures using a coordinate logic filter [30] to fill these gaps within strokes and eliminate additional noisy points. Since the styles of writers also differ concerning to skew, slant, height, and width of the ligatures, the ligatures need to be normalized to a constant and uniform size. Hence, all the ligature images are converted to a uniform resolution 28×28 image dataset using a well-known image enhancement technique 'image rescaling' [31] but keeping information and aspect ratio intact. These operations uniform all the ligature images to constant resolution and fixed size and make the feature extraction process feasible.

3.6 Feature Extraction

The MDLSTM based Recurrent Neural Network (RNN) [30] has been used for extracting features from the normalized ligature images. The RNN generate invariant features which are handled directly by the subsequent MDLSTM layers. The features in the subsequent layer are recurrence with distinctive filters to generate additional invariant and abstract features and the process continues till we get every last feature/output which is invariant to further barriers.

We have engaged a 24 cell based LSTM model as shown in Fig. 5 for the extraction of generic and abstract features from a noise-free, normalized ligature image dataset. The first layer of RNN extracts abstract and generic features such as lines, edges, and corner information from the raw pixels of the image. The inner layers are known to extract relatively low-lying features. Therefore, features from the first layer C1 are selected in the form of MDLSTM filters.

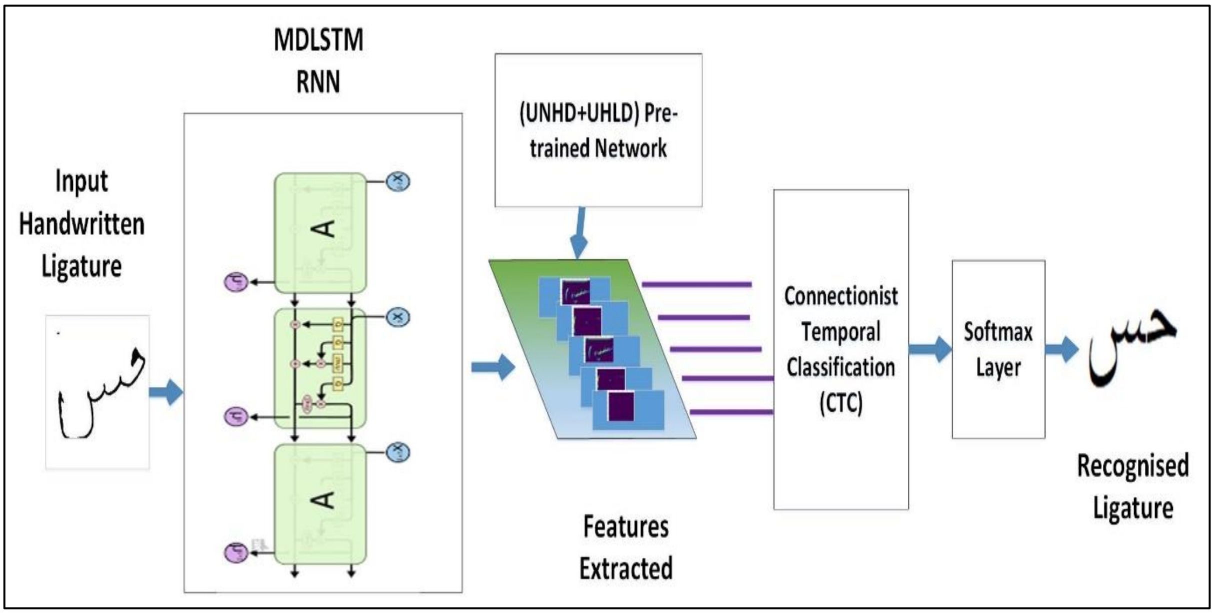


Fig. 5. The proposed RNN model of handwritten Urdu text recognition.

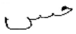
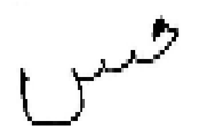
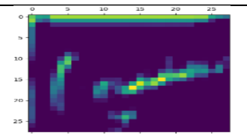
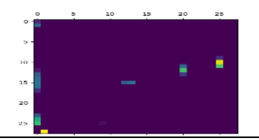
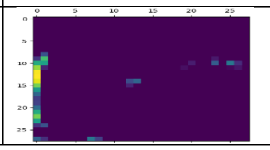
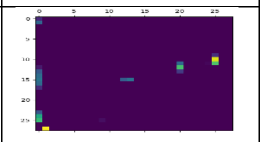
The extracted features of the fourth channel of activation for input ligature () are shown in Table. 3. It is clear that the first MDLSTM layer extracts abstract features such as lines, and edges and retains the shape of the ligature image. The Max-pooling layer summarizes the most activated presence of a feature and calculates the maximum value for each patch of the feature map as shown in Table 3. The second MDLSTM layer extracts less visible features such as corner information and information about principal edges in the image.

Table 3 Activation channels of RNN while extracting features from a given shape of ligature image.

Input Ligature	MDLSTM-1	MDLSTM-2	MDLSTM-3	MDLSTM-4
				

In the output of the feature extraction phase, we get a 28 × 28 feature map with 33 channels. In the wake of plotting every one of the channels of the activation of the primary layer of the RNN model, it has been uncovered that the primary layer is usually holding the full state of the ligatures. Even though there are a few channels that are not activated and are left blank. At that stage, the initial layers hold practically the entirety of the data present in the underlying picture. As we go further in the layers, the activations become progressively conceptual and less outwardly interpretable. The activations start to encode more significant level ideas like single boundaries, corners, and points. Higher introductions convey progressively less data about the visual substance of the picture, and progressively more data identified with the class of the picture. The model structure is very complex to where it may be seen that our last layers not activating by any further extraction of features from the images, hence there's nothing more to learn by then.

The features are extracted from about 150 thousand ligature images classifying them into nearly 1500 unique ligature cluster groups (primary as well as secondary). These features are the kernels of 5×5 size having specific intensity values that maximally match the intensity of the Urdu ligature image of a particular class in the recognition phase. The features extracted are used as kernels to train LSTM-based RNNs discussed in the next section.

3.7 Learning and training

The multi-dimensional LSTM-based Recurrent Neural Network [13] has been used in this research for learning and training of the network because MDLSTM has the property of remembering information and output from previous positions in horizontal, vertical, right, and left directions [14]. A long sequence of handwritten Urdu text input can be simultaneously recognized using the LSTM-based RNNs without increasing the network size. To train an MDLSTM-based recurrent neural network, we employ the features extracted from handwritten Urdu ligature images. We also perform pre-training of the network using the dataset UNHD [16] and our proposed dataset UHLD to evaluate various performance parameters of the network. It must be noted that both in the feature extraction phase and training and recognition phases, the same handwritten Urdu datasets are employed.

In the beginning, the network has been trained on 10 epochs followed by an increment in no. of epochs, with optimized network parameters given in Table 4. We have performed a compilation of the model using the 'Adams optimizer' with a learning rate of 0.001

Table 4: Optimized network parameters for RNN training.

Parameter	Value
Batch Size	256
Number of epochs	20
Momentum	0.9
Weight decay regularization	5×10^{-4}
Base Learning Rate	1×10^{-3}

3.8 Classification and recognition using MDLSTM:

In the recognition phase, a query complete ligature or a text line to be recognized is presented to the MDLSTM-based recurrent neural network. The handwritten Urdu text line is segmented into words/ligatures by applying K U Rehman's Algorithm [28]. The words/ligatures extracted are separated into primary and secondary components using the connected component algorithm. These component images undergo noise removal, normalization, and rescaling to form 28×28 sized images. The noise-free, normalized ligature components are input to the feature extraction phase and the computed feature is input to a trained Multi-Dimensional LSTM-based recurrent neural network as shown in Fig. 5. In the case of pre-trained MDLSTM-based RNNs, the ligature images to be recognized are directly input to the MDLSTM network. The MDLSTM uses a recurrence over both axes of the ligature image allowing it to capture a better context over both axes of the image. This allows for the network model to capture writing variations on both axes and directly work on the raw features extracted. The first MDLSTM layer of the proposed network consists of 28 LSTM cells and a batch size of 128. Initially, the ligature image is input into eight MDLSTM layers, two layers for each scanning direction. The LSTM cell's inner state and output are computed in equation 1, from the states and the output of previous positions in the horizontal, vertical, right, and left directions.

$$LSTM\,(hi,j,qi,j) = LSTM\,(xi,j\ ,hi\ ,j\pm1,h\ i\pm1,j\ ,qi\ ,j\pm1,qi\pm1\ ,j) \quad (1)$$

The features are extracted from each input image by scanning in each direction left, right, up and down directions of input image. These directions are shown in the given formula. In the given formula (1), xi,j represent the input feature vector at position (i,j) and h and q represent the output and inner state of the cell, respectively. The ±1 options in this recurrence depend on which of the four scanning directions is considered.
From Fig. 3, corresponding to an input image, the image skeleton extracted is recurrence with multiple kernels. The output of this recurrence is used as a feature and passed to the MDLSTM classifier as shown in Fig. 5. The MDLSTM compares the features of the query ligature with the kernels extracted, by moving the image on the kernel to extract the feature vector and pass it to the classifier along with the ground truth. Thus the ligature that finds a maximal match with the kernel is recognized as shown in Fig. 5.
However, it must be noted that the output produced by the MDLSTM layer has a continuous form which needs a CTC (connection temporal classification) layer [33] for classification, by selecting the highest probable label (ligature here) from a given sequence of labels using the softmax activation function [34] as shown in Fig. 5.

3.9 Diacritic association

Once the primary or secondary ligatures are recognized, each secondary ligature needs to be assigned to the parent primary ligature. For this purpose, we will find all 'enveloping' as well as 'overlapping' primary ligatures for a given secondary ligature. 'Envelop' refers to a ligature completely occurring within the bounding box of another ligature while 'overlap' refers to the partial occurrence of a secondary ligature within the bounding box of a primary ligature. This approach is presented by Israr Uddin et al [19] for printed Urdu to determine the association of secondary ligatures with their exact primary components. We are applying the same heuristics [19] for handwritten Urdu to associate secondary ligatures with primary components. However, since there can be many characters in a primary component/primary ligature, the dots/diacritics associated with the primary ligature have to be assigned with their actual characters in that primary component. These dots/diacritics are assigned to their respective characters using the pre-computed position and access-order information [19]. A lookup table is maintained that carries information on the allowed set of dots and diacritics with each character class. The information stored in the look-up table is used in associated diacritics to parent ligatures During the association process, each secondary ligature is picked one by one and analyzed for potential assignment to the characters in the primary component after verification from the look-up table. Once a secondary ligature is assigned to a character, it cannot be assigned to another character, and the next secondary ligature in the list is picked for assignment. This complete process of assigning dots/diacritics to their respective character components has been presented by Israr Uddin et al [19] for printed Urdu and has been applied here to handwritten Urdu.

4. EXPERIMENTAL SETUP AND RESULTS:

The experiments were carried out on a ligature set of around 150000 images (with repetitions) extracted from the benchmark dataset UNHD [16] (nearly 50,000) and our proposed dataset UHLD (nearly 100,000). After applying the process of diacritic separation, a large number of ligatures obtained have similarity in shape and hence are reduced to a length of one thousand five hundred unique ligature cluster classes. Also, it is worth mentioning that the ligature dataset extracted whether from UNHD or UHLD is grouped into clusters using our proposed algorithm. The two experimental scenarios considered in our study are summarized in Table 5 below.

Table 5. Distribution of datasets into training and testing datasets.

Dataset	**Training size**	**Testing size**	**Ligature cluster size**
UNHD [16]	50000	10000	500
UHLD (our proposed dataset)	100000	20000	1000
Total Dataset	150000	30000	1500

4.1 Classification report and confusion matrix:

A 'Classification Report' is a performance calculation metric in machine learning. It shows the precision, recall, F1, and support scores for the trained classification model. We have plotted the classification report for the different length ligature cluster classes generated in Table 6. The classification results obtained for these different length ligature clusters are shown in Table 6. Overall aggregate accuracy of 94% has been reported on 1500 unique ligature cluster classes that demonstrate the proposed model is highly accurate and outperforms the state-of-the-art accuracy of 92.75% as obtained in [8, 9] for handwritten Urdu recognition.

Table 6. Summarized classification and accuracy report of all ligature clusters groups

Ligature Group	**No. of Clusters**	**Training Dataset**	**Test Dataset**	**Correct Labels**	**Incorrect Labels**	**Accuracy**
Single Character Ligature	20	11100	2220	2215	05	96.6%
Two Character Ligature	390	24600	4920	4443	477	89.2%
Three Character Ligature	320	23300	4660	4366	294	90.7%
Four Character Ligature	310	22400	4480	4148	332	90.4%
Five Character Ligature	210	19500	3900	3646	254	91.6%
Six Character Ligature	150	17700	3540	3377	163	92.3%
Seven Character Ligature	78	13000	2600	2496	104	93.0%
Urdu Numerals	10	8900	1780	1755	25	96.4%
Secondary Ligatures (Diacritics)	12	9500	1900	1858	42	95.7%
Total Ligature Dataset	1500	150000	30000	28304	1696	**93.8%**

The 'confusion matrix' is used to illustrate the overall accuracy of the classifier in addition to truly predicting positive and negative samples. The 'confusion matrix' of the ligature cluster classification is shown in Fig. 6. The matrix indicates the relation between predicted class labels and actual class labels. It is clear from the plot that the matrix has zero values at all false predictions (where the predicted class label doesn't match the true class label) except diagonal predictions (where the predicted class label exactly equals the true class label). So there is 100% sensitivity (True-Positive-Rate of classifier) and 100% True-Negative-Rate of the classifier. The overall accuracy of the five-character ligature classifier is 93.5%. The 'confusion matrix' for most of the different length ligature classifiers has shown better accuracy of classification that outperforms the state-of-the-art accuracy of 92.75% [8, 9] for handwritten Urdu recognition.

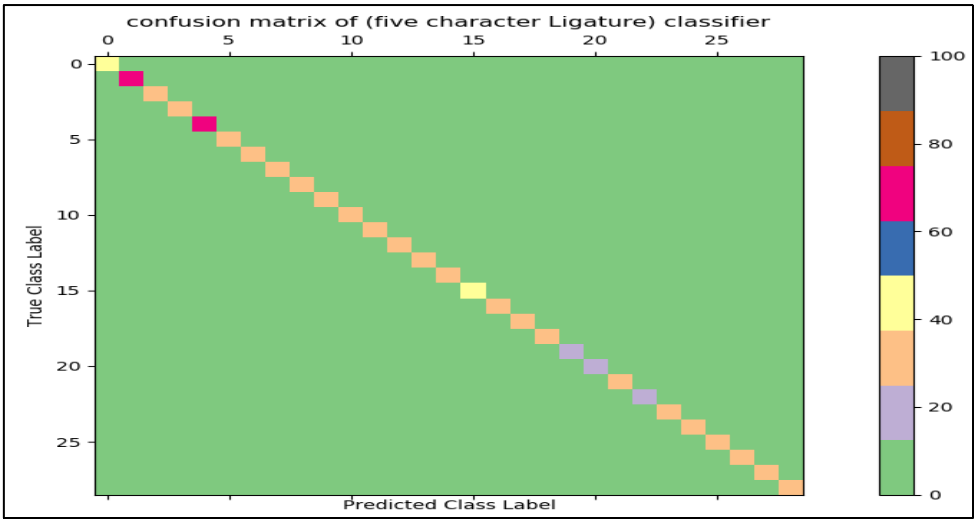


Fig. 6. Confusion Matrix of handwritten Urdu text line comprising 30 ligatures.

The sensitivity, specificity, precision, f1-score, and accuracy parameters for different length ligature cluster classes are shown in Table 7. The average sensitivity of our model is 93% which shows the predictive model is significantly correct. The average specificity of our model is 97% which shows the predictive model is significantly specific. The average precision of our model is 94% which shows the predictive model is appreciably precise. The overall average f1-score of our model is 94.5% which demonstrates the predictive model is appreciably correct.

Table 7: Various accuracy parameters of the proposed system

Performance Parameter	Single Character Ligature	Two Character Ligature	Three Character Ligature	Four Character Ligature	Five Character Ligature	Six Character Ligature	Seven Character Ligature	Secondary Ligatures	Urdu numeral Ligature	Total Ligature dataset
Sensitivity	99.4%	88.7%	94.4%	93.8%	94.5%	95.3%	95.7%	98.7%	99.2%	92.88%
Specificity	98.8%	97.5%	95.7%	95.6%	96.4%	96.5%	97.4%	97.6%	98.6%	96.85%
Precision	99.7%	89.6%	93.8%	94.4%	95.7%	97.2%	98.5%	97.9%	98.8%	93.76%
F1-Score	99.6%	91.8%	92.6%	92.7%	94.4%	92.5%	93.8%	98.6%	98.9%	94.53%
Accuracy	99.8%	90.3%	93.7%	92.6%	93.5%	95.4%	96.0%	97.8%	98.6%	**95.4%**

The accuracy parameters for different length ligature clusters and total ligature dataset are plotted in Table 7 and it can be visualized that these accuracy parameters are noticeably very high for all ligature cluster classes that outperform the state-of-the-art approaches for handwritten Urdu recognition [8, 9]. The values of accuracy parameters for two-character-ligature cluster classes are reported to be comparatively less (hence we have plotted its f1-score above) due to the presence of a larger number of cluster classes in the two-character-ligature cluster, which increases the chances of resemblance in the shape of one class with another and makes classification prone to errors.

4.2 Variation of performance of the proposed model with system parameters:

In an attempt to study the impact of different network parameters on the training performance of the pre-trained network, the training accuracies have been testified as a function of these parameters. In the first series of experiments, we vary the number of LSTM layers of RNN from 1 to 7 while keeping other parameters constant. A similar training recognition pattern is observed on both the datasets (UNHD and UHLD) and it is pertinent to mention that the model report maximum training accuracy rates on 8, 10, and 12 layers of LSTM (Fig 7). However, the accuracy reduces with an increase in the number of LSTM layers because adding more layers will over train the dataset.

In an attempt to analyze the input scalability of the system, we study the performance variation with respect to the length of input text line i,e. the number of ligatures simultaneously presented to the system for recognition. Since the first MDLSTM layer of the network consists of 28 LSTM cells that read a long text line consisting of up to 28 ligatures all together (Fig 8). However, an increase in input sequence length results following impact on performance.

a) Increase in training time as the model needs to process and learn dependencies across longer sequences.

b) Increase in memory usage for storing intermediate states during backpropagation.

c) Reduction in recognition rate as the model needs to capture dependencies and patterns across extended contexts.

d) Overall increase in computational complexity as longer sequence requires more computations during training as well as testing phases.

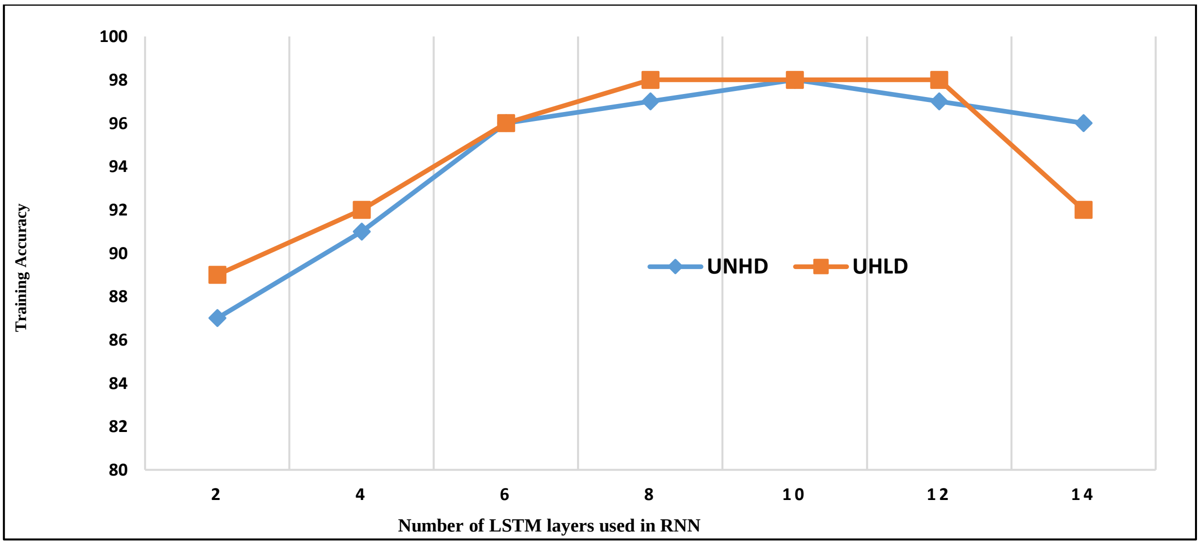


Fig. 7. Variation of training accuracy with increase in depth of model.

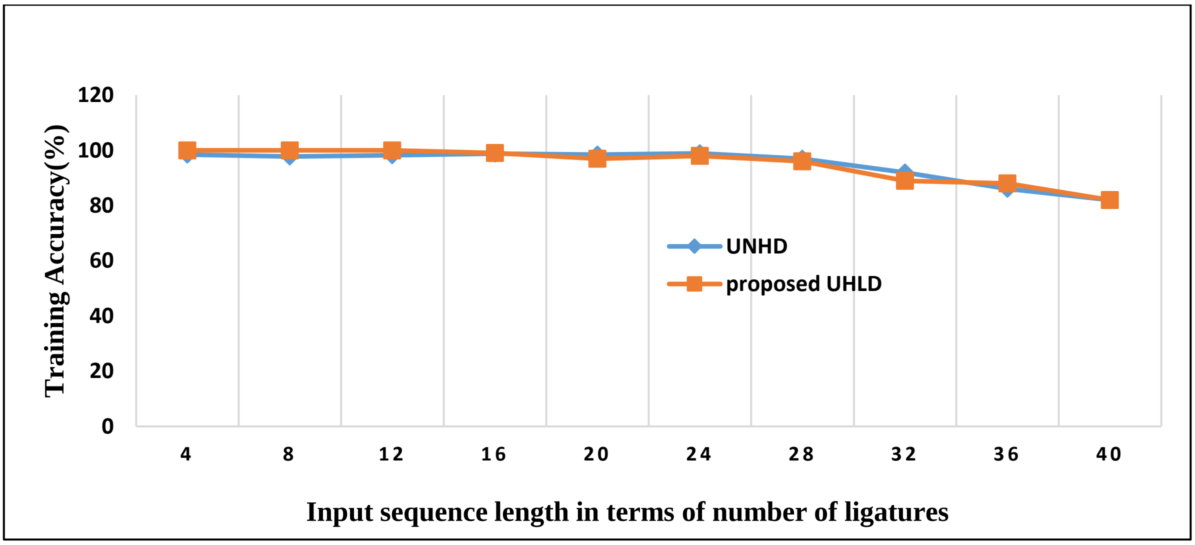


Fig. 8. Variation of training accuracy with the input sequence length

4.3 Complexity of the Proposed system

Finally, in the context of the computational complexity of the proposed technique, there are two important phases; extracting the ligatures and training/recognition. When a text line is presented to the system, ligatures are extracted by applying the ligature segmentation algorithms used in [15, 29]. This technique is relatively easier than the 'sliding window technique' [35] used in [11] to recognize characters in a ligature because in our proposed technique a complete ligature is recognized at a time.

From a training/recognition complexity point of view, training LSTM RNNs naturally is computationally very expensive. However, in all pattern classification systems, it is the test time complexity that is more critical rather than the training complexity as training is carried out offline. In comparison to other techniques where a separate model is trained for each ligature class (for example a separate HMM for each class in [11]), a single LSTM RNN model is required in our proposed system to recognize all ligature classes. Modeling a separate classifier for each class has a test time complexity of O (N) as a query ligature is fed to all N-trained models (N is the number of unique classes) as used in [9]. On the other hand, irrespective of the number of classes, in our technique, a query ligature is fed to only one model and hence has a complexity of O (1) making it efficient in terms of test time computational complexity.

4.4 Comparison of results with notable studies:

As discussed earlier, the Handwritten Urdu recognition has been the least explored to date and our proposed work outperforms the state-of-the-art approaches [8, 9] in terms of larger accuracy and small computational complexity reported and forms the new state-of-the-art technique for handwritten Urdu recognition. A comparison of recognition rates of notable research in handwritten Urdu and that of the proposed work is summarized in Table 8.

Study	Database	Training Set	Test Set	Proposed Approach	Accuracy	True Ligature Recognition	Computational Complexity
Ahmad S.B. et al [20]	UCOM	50 text lines	20 text lines	Not Available	Not Available	Not Available	Not Available
Ahmad S.B. et al [16]	UNHD	6400 Text Lines	1840 text lines	BLSTM Classifier	92%	NO	O(N) N is the number of classes
Husnain M. et al [21]	Proposed a database of Urdu Characters.	Information Unavailable	Information Unavailable	CNN classifier	96% Characters 98% Numerals	NO	O(N) N is the number of classes
Hassan S et al, 2019 [22]	Claimed their Database, is not publicly available	4000 handwritten Lines	1000 text lines	7 CNN layers and 2 BLSTM layers	83.6%	NO	O(N) N is the number of classes
Faisal et al [23]	HUCD	74285 character samples	21223 character samples	CNN	98.8%	NO	O(N) N is the number of classes
Ahmad S.B. et al [8]	UNHD, MNist Pre-trained Network	10000 Text Lines	2000 Text lines	CNN and MDLSTM	93%	NO	O(N) N is the number of classes
Proposed Work	**UNHD [16]**	**5000 text lines**	**1000 text lines**	**RNN model**	**92.6 %**	**YES**	**O (1) Only independent of the number of classes**
	UHLD (our proposed)	**6000 text Lines**	**1000 text lines**	**RNN model**	**93.8 %**	**YES**	
	Both UNHD and UHLD	**11000 text lines**	**2000 text lines**	**RNN model**	**91.4%**	**YES**	

Table 8: Comparison of proposed work with notable research in handwritten Urdu recognition

5. CONCLUSION AND FUTURE SCOPE

Handwritten Urdu recognition is a challenging area in natural language processing. In this research, we have proposed a novel holistic approach to handwritten Urdu text recognition. In this approach, we construct the ligature classes from benchmark datasets UNHD and UHLD. The features are extracted from these classes using RNNs, and these features train MDLSTM RNN to classify handwritten Urdu ligatures and recognize any unconstrained handwritten Urdu text. In this research, several benchmark methods and metrics such as confusion matrix, specificity, sensitivity, accuracy (ACC), and precision were computed to accurately assess the proposed recognition model. The results of these parameters demonstrate that the predictive model depicts better recognition rates. The individual recognition rate of a few ligature clusters like four-character ligature class, five-character ligature class, six-character ligature class, and seven-character ligature class, and even Urdu numerals reports 100% accuracy. The proposed technique is more computationally efficient than the state-of-the-art approaches because the proposed technique offers a constant time complexity of O (1) only (a constant number)

irrespective of the number of classes used whereas the state-of-the-art approaches offer large complexity of O(N) where is n is the number of classes/language models used. The accuracy of our proposed technique has been determined to be better than the state-of-the-art techniques for handwritten Urdu recognition. Hence the proposed approach for ligature recognition of handwritten Urdu forms the new state-of-the-art technique for handwritten Urdu OCR.
In future, the proposed approach may be applied on various challenging cursive text recognition tasks such as patwari Urdu text recognition, scene text recognition and other natural language processing applications.

ACKNOWLEDGEMENTS
We thank Dr. Saad Bin Ahmad for providing the UNHD database. The database UHLD is under development process and will be soon available for research.

CONFLICT OF INTEREST
The authors declare to have no conflict of interest.

REFERENCES
[1] G. G. Chowdhury, "Natural language processing," Ann. Rev. Info. Sci. Tech., vol. 37, no. 1, 51-89. doi:10.1002/aris.144 0370103.
[2] O. Obulesu et al., "Machine learning techniques and tools: A survey". In 2018 international conference on inventive research in computing applications (ICIRCA). IEEE, 2018 Jul. 11, pp. 605-611. doi:10.1109/ICIRCA.2018.8597302.
[3] L. Eikvil, Optical Character Recognition. Citeseer. ist. psu. edu/142042. html, vol. 26, 1993 Dec.
[4] H. F. Schantz, History of OCR, Optical Character Recognition. Manchester, VT, USA: Recognition Technologies Users Association, 1982.
[5] G. Tauschek, "Reading machine". U.S Patent 2026329. Available at: http://www.google.com/patents?vid=USPAT2026329, Dec., p. 1935FLEXChip, Signal Processor (MC68175/D). Motorola, 1996.
[6] P. W. Handel, "Stat IST iCal Machine". U.S Patent 1915993. Available at: http://www.google.com/patents?vid=USPAT1915993, Jun. 1993.
[7] C. E. Shannon, "A universal Turing machine with two internal states," Automata Stud., vol. 34, pp. 157-165, 1956 Apr. 21.
[8] S. Mori et al., "Historical review of OCR research and development," Proc. IEEE, vol. 80, no. 7, pp. 1029-1058, 1992 Jul. doi:10.1109/5.156468.
[9] R. Smith, "An overview of the Tesseract OCR engine" in Ninth international conference on document analysis and recognition (ICDAR 2007), vol. 2. IEEE, 2007 Sept. 23. doi:10.1109/ICDAR. 2007.4376991.
[10] R. W. Smith, "History of the Tesseract OCR engine: What worked and what didn't," SPIE Proceedings. SPIE, vol. 8658, 2013 Feb. 4. doi:10.1117/12.2010051.
[11] S. Akhil, "An overview of the tesseract OCR engine" in A Seminar Report. Department of Computer Science and Engineering National Institute of Technology, Calicut Monsoon 2016.
[12] X. Peng et al., "Multilingual OCR research and applications: An overview" in Proc. 4th International Workshop on Multilingual OCR 2013 Aug 24, 2013, pp. 1-8. doi:10.1145/2505377.2509977.
[13] S. Kompalli et al., "Challenges in OCR of Devanagari documents" in Eighth International Conference on Document Analysis and Recognition (ICDAR '05). IEEE, 2005 Aug. 31, pp. 327-331.
[14] N. H. Khan and A. Adnan, "Urdu optical character recognition systems: Present contributions and future directions," IEEE Access, vol. 6, pp. 46019-46046, 2018. doi:10.1109/ACCESS.2018.2865532.
[15] N. Sabbour and F. Shafait, "A segmentation-free approach to Arabic and Urdu OCR" in, SPIE Proceedings, vol. 8658, 2013, Feb. doi:10.1117/12.2003731. International Society for Optics and Photonics.
[16] S. Naz et al., "'Arabic script based language character recognition: Nasta'liq vs Naskh analysis,' in Proc". World Congr. Comput. Inf. Technol. (WCCIT), Jun. 2013, pp. 1-7.
[17] S. Hussain et al., "Adapting Tesseract for complex scripts: An example for Urdu Nastalique". In'2014' 11th IAPR International Workshop on Document Analysis Systems 2014 Apr 7. IEEE, pp. 191-195.
[18] H. Dawood et al., Improved Arabic Word Classification Using Spatial Pyramid Matching Method.
[19] G. G. Chowdhury, "Natural language processing," Ann. Rev. Info. Sci. Tech., vol. 37, no. 1, 51-89. doi:10.1002/aris.1440370103..

[20] O. Obulesu et al., "Machine learning techniques and tools: A survey". In the 2018 international conference on inventive research in computing applications (ICIRCA). IEEE, 2018 Jul. 11, pp. 605-611. doi:10.1109/ICIRCA.2018.8597302.

[21] L. Eikvil, Optical Character Recognition. Citeseer. ist. psu. edu/142042. html, vol. 26, 1993 Dec.

[22] D. A. Satti and K. Saleem, "Complexities and implementation challenges in offline Urdu Nastaliq OCR" in Proc. Conference on Language & Technology, 2012, Nov., pp. 85-91.

[23] N. H. Khan and A. Adnan, "Urdu optical character recognition systems: Present contributions and future directions," IEEE Access, vol. 6, pp. 46019-46046, 2018. doi:10.1109/ACCESS.2018.2865532.

[24] A. Cheung et al., "An Arabic optical character recognition system using recognition-based segmentation," Pattern Recognit., vol. 34, no. 2, pp. 215-233, 2001 Feb. 1. doi:10.1016/S0031-3203(99)00227-7

[25] G. S. Lehal, "Choice of recognizable units for Urdu OCR" in Proceeding of the Workshop on Document Analysis and Recognition, 2012, Dec., pp. 79-85. doi:10.1145/2432553.2432569.

[26] S. B. Ahmed et al., "Evaluation of handwritten Urdu text by integration of MNIST dataset learning experience," IEEE Access, vol. 7, pp. 153566-153578, 2019. doi:10.1109/ACCESS.2019.2946313.

[27] S. Naz et al., "Segmentation techniques for recognition of Arabic-like scripts: A comprehensive survey," Educ. Inf. Technol., vol. 21, no. 5, pp. 1225-1241, 2016. doi:10.1007/s10639-015-9377-5.

[28] I. U. Din et al., "Holistic optical character recognition for printed Urdu text," EURASIP J. Image Video Process., vol. 2017, no. 1, p. 62, 2017.

[29] S. Albawi et al., "Understanding of a convolutional neural network". At the 2017 International Conference on Engineering and Technology (ICE T). IEEE, 2017 Aug. 21, pp. 1-6. doi:10.1109/ICEngTechnol.2017.8308186.

[30] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," Phys. D Nonlinear Phenom., vol. 404, p. 132306, 2020 Mar. 1. doi:10.1016/j.physd.2019.132306.

[31] A. M. Schaefer et al., "Learning long-term dependencies with recurrent neural networks," Neurocomputing, vol. 71, no. 13-15, pp. 2481-2488, 2008 Aug. 1. doi:10.1016/j.neucom.2007.12.036.

[32] S. B. Ahmed et al., "Handwritten Urdu character recognition using one-dimensional BLSTM classifier," Neural Comput. & Applic., vol. 31, no. 4, pp. 1143-1151, 2019. doi:10.1007/s00521-017-3146-x.

[33] S. B. Ahmed et al., "Evaluation of cursive and non-cursive scripts using recurrent neural networks," Neural Comput. & Applic., vol. 27, no. 3, pp. 603-613, 2016. doi:10.1007/s00521-015-1881-4.

[35] S. Naz et al., "Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks," Neurocomputing, vol. 177, pp. 228-241, Feb. 2016. doi:10.1016/j.neucom.2015.11.030.

[36] I. Uddin et al., "Recognition of printed Urdu ligatures using convolutional neural networks," J. Electron. Imaging, vol. 28, no. 3, p. 033004-, 2019 May 1. doi:10.1117/1.JEI.28.3.033004.

[37] S. B. Ahmed et al., "UCOM offline dataset-an Urdu handwritten dataset generation," Int. Arab J. Inf. Technol. (IAJIT), vol. 14, no. 2, 2017.

[38] M. Husnain et al., "Recognition of Urdu handwritten characters using convolutional neural network," Appl. Sci., vol. 9, no. 13, p. 2758, 2019. doi:10.3390/app9132758.

[39] S. Hassan et al., "Cursive handwritten text recognition using bi-directional LSTMs: A case study on Urdu handwriting" International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML). IEEE, 2019 Aug. 26, pp. 67-72 doi:10.1109/Deep-ML.2019.00021.

[40] F. Mushtaq et al., "UrduDeepNet: Offline handwritten Urdu character recognition using deep neural network," Neural Comput. & Applic., vol. 33, no. 22, pp. 15229-15252, 2021 Nov. doi:10.1007/s00521-021-06144-x.

[41] G. Chen and S. N. Srihari, "Removing structural noise in handwriting images using deep learning" in Proc. 2014 Indian Conference on Computer Vision Graphics and Image Processing (ICVGIP '14). New York, NY, USA: Association for Computing Machinery, 2014, pp. 1-8, article 28. doi:10.1145/2683483.2683511.

[42] B. Sankur, "Survey over image thresholding techniques and quantitative performance

evaluation," J. Electron. Imaging, vol. 13, no. 1, 2004. doi:10.1117/1.1631315.

[43] A. F. Ganai and A. Koul, "Projection profile based ligature segmentation of Nastaleeq Urdu OCR". In 2016 4th International Symposium on Computational and Business Intelligence (ISCBI). IEEE, 2016 Sept. 5, pp. 170-175. doi:10.1109/ISCBI.2016.7743278.

[44] K. U. U. Rehman and Y. D. Khan, "A scale and rotation invariant Urdu nastalique ligature recognition using cascade forward backpropagation neural network," IEEE Access, vol. 7, pp. 120648-120669, 2019 Aug. 19. doi:10.1109/ACCESS.2019.2936363.

[45] G. S. Lehal, "Ligature segmentation for Urdu OCR" 12th International Conference on Document Analysis and Recognition, vol. 2013, 2013. doi:10.1109/ICDAR.2013.229.

[46] S. M. Mostafavi et al., "Noise removal from printed text and handwriting images using coordinate logic filters". In 2010 International Conference on Computer Applications and Industrial Electronics. IEEE, 2010 Dec. 5, pp. 160-164. doi:10.1109/ICCAIE.2010.5735067.

[47] T. Pavlidis, "A thinning algorithm for discrete binary images," Comput. Graph. Image Process., vol. 13, no. 2, pp. 142-157, 1980 Jan. 1. doi:10.1016/S0146-664X(80)80037-2.

[48] C. Szegedy et al., "Going deeper with convolutions" in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1-9. doi:10.1109/CVPR.2015.7298594.

[49] H. Scheidl, "An intuitive explanation of connectionist temporal classificat-ion," Towards Data Sci., 2018 Jun.

[50] S. Sharma et al., Activation Functions in Neural Networks. Towards Data Science vol. 6, no. 12, 2017 Sept, pp. 310-316.

[51] C. H. Lee et al., "Sliding-window filtering: An efficient algorithm for incremental mining" in Proc. Tenth International Conference on Information and Knowledge Management 2001 Oct 5, 2001, pp. 263-270. doi:10.1145/502585.502630

.