Scientia Iranica D (2022) 29(5), 2419–2436

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

A novel similar character discrimination method for online handwritten Urdu character recognition in half forms

Q.A. Safdar[a,*], K.U. Khan[a,*], and L. Peng[b]

a. *Department of Electrical Engineering, Pakistan Institute of Engineering and Applied Sciences, Islamabad, Pakistan.*
b. *Department of Electronic Engineering, Tsinghua University, Beijing, China.*

Received 4 July 2017; received in revised form 25 November 2017; accepted 11 August 2018

KEYWORDS
Online;
Handwritten character recognition;
Half forms;
Multistroke Urdu characters;
Wavelets;
ANN;
SVM;
RNN;
DBN.

Abstract. Online handwritten Urdu character recognition is one of the key technologies for the intelligent interface on smart phones and touch screens. It is a challenging research topic as Urdu script has many similar character groups. A novel similar characters discrimination method for online handwritten Urdu character recognition is proposed in this paper, which includes pre-classification, feature extraction, and fine classification process. The pre-classifier enabled the discrimination of similar characters by putting them in distinct smaller subsets according to stroke number and diacritics. Then, structural features and wavelet features were extracted. Finally, Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Recurrent Neural Network (RNN) classifiers were compared for fine classification within subsets. Results of RNN classifier without using the proposed pre-classifier and features were also obtained to check the end-to-end capability of the RNN classifier. Experimental results showed that the proposed method was efficient and achieved an overall accuracy of 96% for a large-scale self-collected dataset. It is feasible to extend this method for other Arabic scripts.

1. Introduction

Development of portable computing devices, writing pads, and smartphones with non-keyboard-based input interface is receiving considerable attention in the research communities and commercial sector. The provision of an interface that can recognize handwritten inputs efficiently is a non-trivial task owing to complexities involved in handwriting, limited memory, and relatively low processing resources available in mobile devices. From the point of view of a user, online recognition systems are receiving greater acknowledgment than offline ones are, because of their convenience in writing compared to typing, their usefulness in situations that typing is hard, inadequate keyboard facility on small computers, and difficulty to type in some languages for their large numbers of letters [1]. From a developer's point of view, the advantage of a pen-tablet environment (the online system) is that it facilitates the process of recognition with some important information, which is missing in its counterpart (that is, the offline system). For example, the handwritten stroke coordinates are available in online systems as a function of time along with pressure values of the respective pen. Moreover, auxiliary information of writing speed, stroke order, and pen-up/down events can be tracked in these online systems. In spite of

*. *Corresponding authors.*
E-mail addresses: aineemalik@hotmail.com (Q.A. Safdar); kamranaman@yahoo.com (K.U. Khan); penglr@tsinghua.edu.cn (L. Peng)

doi: 10.24200/sci.2018.20826

these difficulties, some work on development of *state of the art* character-recognition systems (either in offline or online mode) for alphanumeric handwriting has been reported for various languages, like English [2], Persian [3], Chinese [4], Japanese [5], Mongolian [6], Kannada [7], and Arabic [8–14]. In fact, there is well-matured online handwriting recognition software for Latin-script-based (like English) and character-based languages (such as Chinese).

Urdu is one of the script-based languages derived from Arabic and Persian. According to estimates, it is written, spoken, and used by more than 200 million people around the world. Urdu is officially recognized in India due to the existing 70 million native Urdu speakers. It is also spoken and understood in Nepal, Bangladesh, the Middle East, Fiji, USA, and many other countries around the globe, including UK (having about 400,000 native Urdu speakers). In Pakistan, populated with approximately 200 million speakers, Urdu is the primary language of communication and there are about 130 million mobile phone users [15]. According to market estimates, based on the current trends in the e-commerce sector, there can be 40 million smart phones in Pakistan by December 2016 [16]. In that scenario, there is a need for carrying out research in the field of design and development of online Urdu handwriting recognition systems for computing devices (like smartphones) to provide benefit for the large Urdu speaking population of the world. Online Urdu handwriting recognition system can also extend its benefits to the users of other Arabic script based languages like Persian, Uyghur, Sindhi, Punjabi, and Pushto with minor modifications.

Urdu script comprises a larger character set with cursively written and contextually dependent alphabet. Being context dependent, Urdu letters adjust their shapes according to the 'preceding and following' characters. In this way, for the Urdu alphabet, there are one full and at least three different half forms with few exceptions. Moreover, complexities of Urdu handwriting recognition arise not only from cursive and context-dependent nature of the alphabet, but also from the very nature of a letter structure, word formation in a particular font style, and diacritics involved in letters. Overlapping ligatures; delicate joints of characters in a word; aslant traces; neither fixed baseline nor standard slope (in Nastalique font style); associated dots and other diacritics, which may be above, below, or within the character; and displacement of dots with the slope and context of the base stroke [17–19] are a few to shed light on the complexities of Urdu script.

On the basis of the target set, Urdu handwriting recognition (both offline and online) can be placed into three categories, namely isolated or full-form character recognition [20–23], selecting ligatures for recognition or holistic approach (also known as segmentation-free approach) [17,24–28], and segmentation-based or analytical approach [29–35].

Moreover, different researchers have tried to address the recognition problem by focusing on different aspects. For example, the authors in [36] worked out the baseline (an imaginary line on which characters were combined to form the ligatures) of the character stroke; the work in [37] discussed the diacritical marks associated with characters and ligatures; and the approach in [38] emphasized the pre-processing operations.

Following the analytical approach along with dictionary based search to obtain valid characters and words, Malik and Khan [20] recognized 39 isolated characters with an overall accuracy of 93% and 200 two-unattached-characters ligatures with an accuracy of 78%. Hussain and Khan [25] preferred the holistic approach, proposed spatial temporal artificial neuron for the recognition, and reported an accuracy of 85% for 15 selected ligatures only. However, their data set lacked the aspect of generality as it was acquired from only two different writers. Husain et al. [26] investigated the recognition system for one-, two-, and three-character ligatures and obtained separate results of 93% and 98% for base and secondary strokes, respectively. Shahzad et al. [21] studied the recognition of 38 isolated Urdu characters using 9 geometric features for primary stroke and 4 for secondary stroke to achieve the accuracy of 92.8% for the data obtained only from two native writers; however, the recognition rate diminished to 31% when the characters were scribbled by an untrained non-native writer. With the scribbled data of the trained non-native writer, the recognition rate barely increased to 73%. Razzak et al. [27,28] investigated the recognition system for 1800 ligatures. By utilizing the features based on fuzzy rules and hidden Morkov model, they secured 87.6% recognition rate for Urdu Nastalique font and 74.1% for Naskh font. Most of the work available in the online domain of Urdu character recognition deals with ligatures and full-form recognition. Segmentation-based approaches have been applied either to distinguish the ligatures present in a word from each other or to dissociate the diacritics from the base character [19]. It is noteworthy that, to the best of the authors' knowledge, no work is found using wavelet analysis for recognition of Urdu characters. However, studies have been reported for Arabic and Persian characters recognition using wavelets. Therefore, on the basis of alike-script and wavelet analysis, the present paper is compared with Arabic and Persian work as well. Table 1 accounts for the comparison of the proposed work with Arabic and Persian recognition systems using wavelet analysis.

Inspired by [22,23,39], the authors propose in this work the online Urdu character recognition problem for context-dependent shapes of Urdu characters, that is,

Table 1. Comparison of the online Urdu handwritten character recognition (proposed) with Arabic and Persian work.

Authors	Type	Character Set × Samples	Language	Features	Classification	Participants	Accuracy
Proposed work	Multistroke characters (IHF, MHF, THF)*	77 × 100	Urdu	Structural, wavelet Coefficients	BPNN, SVM	100	87.5% to 100%
Jannoud [67]	Isolated, IHF, MHF, THF	Not reported	Arabic	Discrete wavelet transformation	MLE	Not reported	99% for isolated, more than 90% for IHF and THF, 91% for MHF
Asiri and Khorsheed [68]	Isolated, IHF, MHF, THF	30 × 500	Arabic	Haar wavelet transform	ANN	Not reported	For 3 different sets of wavelet coefficients: 74%, 82%, and 88%
Mowlaei et al. [43]	Isolated	32 × 190	Persian	Haar wavelets	MLP	200	92.3%
Aburas and Rehiel [42]	Isolated	28 × 48	Arabic	Wavelet coefficients	Codebook search & Euclidean distance measure	48	45.8% to 97.9%
Broumandnia et al. [69]	Words	100 × 8 rotations of each word	Persian	2D M-band wavelet packets	Mahalanobis classifier	12	65% to 96%
Jenabzade et al. [44]	Isolated	33 × 200	Persian	Haar wavelets	MLP	Not reported	86.3%

*IHF: Initial Half Form; MHF: Medial Half Form; THF: Terminal Half Form

for half forms. For the development of online cursive Urdu handwriting recognition system, recognition of half-form Urdu characters is a primary step because of the following four reasons: First, Urdu characters appear in half forms in a word. Although full-form letters are also used within a word, the role of half forms is much more than that of full forms. Second, half-form characters are the building blocks of ligatures and therefore, segmentation-based systems eventually attempt to recognize the constituent half forms [29,33–35]. Third, there are a lot more ligatures in Urdu, which cannot be entirely enclosed within the scope of a single study. That is why researchers have tried to recognize selective numbers of ligatures through which many words, not all, can be composed. Consequently, such systems have limited vocabulary available for processing [27,38,19]. Furthermore, for acquiring a valid ligature or finding an optimum word, dictionary-based search becomes a necessary part of the research [26]; however, this is not the case with the half forms. Last, targeting half forms would mean independence from dictionary. Even new words not present in dictionary can be recognized.

Instead of putting all the characters at once into a single recognizer, we opted to pre-classify the larger half-form character set into smaller subsets. A pre-classifier is proposed, which puts similar characters into distinct smaller subgroups. Then, these smaller subgroups are targeted for further classification through Artificial Neural Network (ANN) and Support Vector Machines (SVM) classifiers by employing wavelet and structural features, and by RNN classifier using the raw stroke data (without using the extracted features). In character recognition problems, wavelet transform has been used for languages like English [40], Chinese [41], Arabic [42], Persian [43,44], and different Indian languages [45,46]. A database is available for Arabic handwritten words (Arabic DAtaBase: ADAB [47]), but for Urdu, there is no standard handwritten character database. The end-to-end recognition capability of the RNN classifier [48,49] has also been utilized in which all the characters are fed to the RNN classifier without any feature extraction or pre-classification. A large database of Urdu handwritten characters has also been developed by the authors, which is provided for research purposes. The main contributions of this work are as follows:

1. A framework for development of online Urdu handwriting recognition in smartphones has been presented;
2. Based on the number of strokes in a character as well as the position and shape of diacrtics, segregation of larger character set into smaller subsets is performed through the proposed pre-classification in contrast to the previous online Urdu character recognition approaches like [20,21,25–28,38];
3. To cope with the demand of robust and accurate

recognition along with the relatively low computational power and limited memory available to mobile devices, banks of computationally less complex classifiers are developed from which the appropriate classifier would be loaded to the memory to achieve the recognition task;

4. A comparison of different classifier/feature combinations is presented in this study to distinguish be- tween features' discrimination and classifiers' recognition ability;
5. A comparison of feature-based classifiers (ANN, SVM) and end-to-end classifier (RNN) is presented;
6. Noting the small databases of the existing Urdu character recognition studies [21,25,27,28], a large database of handwritten Urdu characters is developed and employed in this study, which contains 10800 samples of all Urdu half-form characters (100 samples of each character) acquired from 100 writers. The database can be obtained from the authors for the research purposes.

For different classifier/feature combinations, the overall accuracies obtained through the proposed methods are 81.9%, 92.8%, 95.8%, 96.1%, 84.7%, 87.2%, and 60% (to be detailed in results). The best overall recognition rate is procured by SVM. For individual characters, the recognition rates obtained are up to 100% by application of the resultant schemes.

The organization of the paper is as follows. A brief introduction of the Urdu character set in half forms is provided in Section 2. The proposed online Urdu handwriting recognition system is explained in Section 3. Results and discussions are presented in Section 4. The paper is concluded in Section 5.

2. About the Urdu character set in half forms

In this section we will analyze the way in which Urdu words are handwritten by the native writers. Urdu handwriting is inherently cursive and there are many Urdu font styles available, such as *Naskh, Nastalique, Kofi, Thuluth, Diwani, and Rouq'i. Nastalique* style is mostly adopted for Urdu writing, whereas Arabic is penned in *Naskh* style. With respect to the position in a word, *Nastalique* font style reveals writing with atilt ligatures and distinctive variations in letters [19]. For example, the character 'ت' adapting three different shapes as per context is shown in Figure 1.

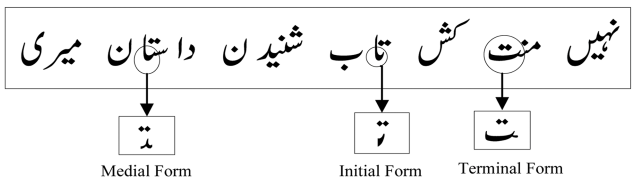

Figure 1. Urdu character 'ت' in half forms.

Most of the characters in Urdu words appear in three different forms as shown in Figure 1 (see also [50]). The form in which a character appears in a word depends on the position in which it occurs in the word. These forms are described below:

- **Full form:** Every character in Urdu has a full form. The full forms always occur in isolated positions in a word (not ligature). Urdu character set consists of 37 characters (letters/alphabet) in full form [23]; however, this count is reported 39 in [19]. The difference is due to the addition of some characters to the basic Urdu character set.
- **Initial half form:** A character falling in the beginning of a word (more generally, a ligature) adopts the initial form. Not every character has an initial half form. There are 36 initial half forms.
- **Medial half form:** Characters falling in the middle of a word (or ligature) adopt their medial forms. Some characters do not have medial half forms; they are 30 in number.
- **Terminal half form:** Characters falling at the end of a word (or ligature) adopt the terminal half form. Every character has a terminal form. The terminal forms have very much similar shapes to their corresponding full forms. There are 42 terminal half forms.

All Urdu characters in half forms (108 in number) are shown in Figure 2. The aim of this work is to classify all these 108 shapes while handwritten online. Some characters possess more than one *shape* at initial, medial, or terminal positions. The usage of those shapes depends on the **context** in which that character appears. The context means which other characters appear, in a ligature, before and after a particular character.

Analyzing Urdu characters in further detail, we find that an Urdu character consists in a major stroke and may have none, one, two, or three minor strokes. There are few groups of characters in which the major stroke is common to the group and the distinction among the characters is made on the basis of the type, count, and position of the minor strokes. Depending on

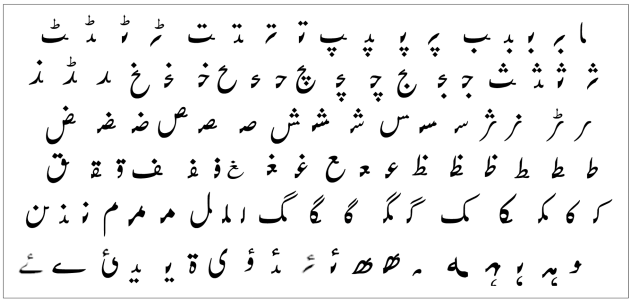

Figure 2. All Urdu characters in all half forms.

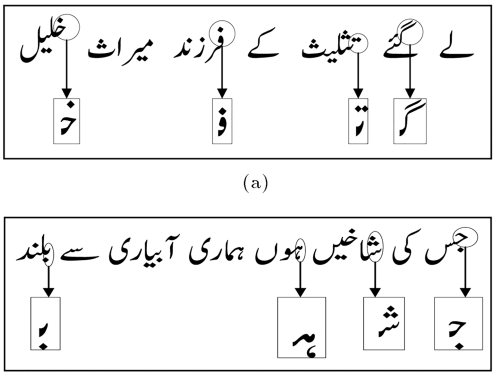

Figure 3. Examples: Use of initial half form multi stroke characters.

the number of strokes, we categorize Urdu characters into 4 subsets, namely single-, two-, three-, and four-stroke characters. A few examples specifying the use of two-, three-, and four-stroke characters as initial half forms have been shown in Figure 3.

There are five different types of minor strokes on the basis of the shapes drawn: dot or *nuqta* ('ٔ'), towey ('ط'), inverted *hay* ('ۂ'), *hamza* ('ء') and *kash* ('ٕ'). Moreover, minor stroke(s) may be placed above or below a major stroke depending upon the character. Multistroke characters can also be grouped on the basis of the position of secondary stroke(s) with respect to the primary stroke as well as on the basis of the shape of the secondary stroke.

Due to the presence of similar characters, various half forms, context dependency of shape of a character (108 shapes), and different types of minor strokes, recognition of online handwritten Urdu characters is a complex and challenging pattern recognition problem.

3. Proposed online handwritten Urdu character recognition system

In this section, we present the proposed online handwritten Urdu character recognition system. The whole system consists of data acquisition, preprocessing, preclassification, feature extraction, and classification stages. The block diagram of the whole system is shown in Figure 4 and all the stages are explained in the following subsections.

3.1. Data acquisition

Handwriting data can be acquired using a pen-tablet device connected to a computer. Data may also be acquired by writing on the touch-sensitive screen of a smartphone. In our study, 100 native Urdu writers of different age groups provided their handwriting samples using a stylus and digitizing tablet. Online handwritten character signals contain the information of the digitized coordinates $(x(t), y(t))$ and the pressure values and time-stamps for each point $(x(t), y(t))$.

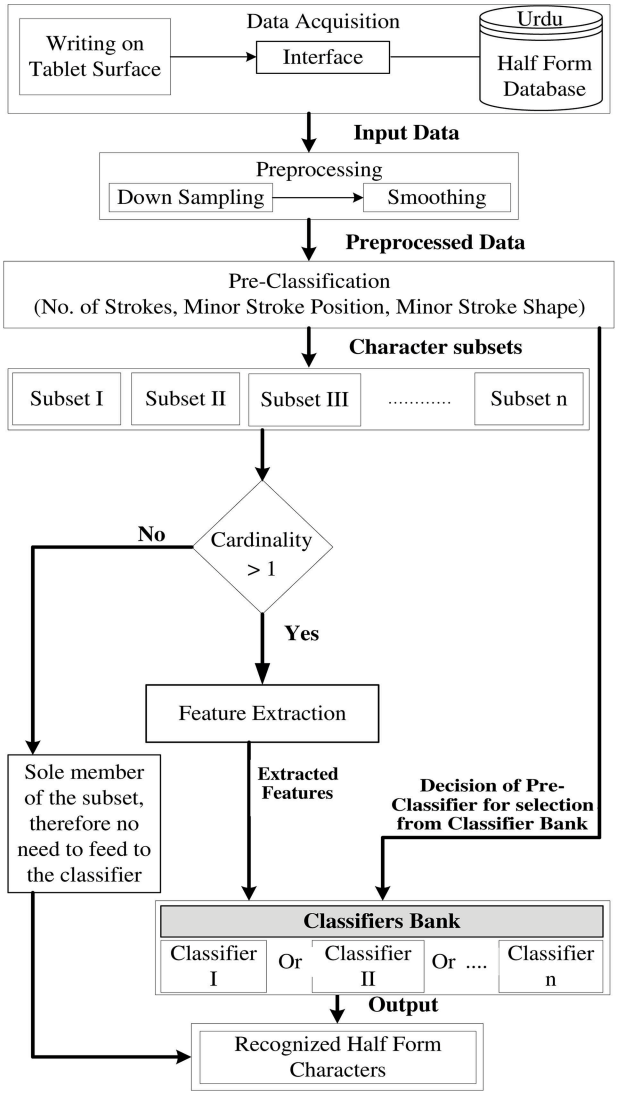

Figure 4. Block diagram of the proposed online Urdu character recognition system: From data acquisition to preprocessing to pre-classification to feature extraction to final classification.

During the data acquisition, the following attributes of character strokes were acquired:

1. Number of times the pen is up/down;
2. Number of strokes in a character;
3. tarting/ending index of each stroke;
4. Temporal order of each sample of $(x(t), y(t))$ coordinates;
5. Pressure value at $(x(t), y(t))$. Note: Pressure value is not utilized in this work except for detecting pen up/down events.

3.1.1. About the data

The data obtained from the writers is in segmented form. Figure 5 shows few examples of full Urdu words and ligatures composed of the segmented characters

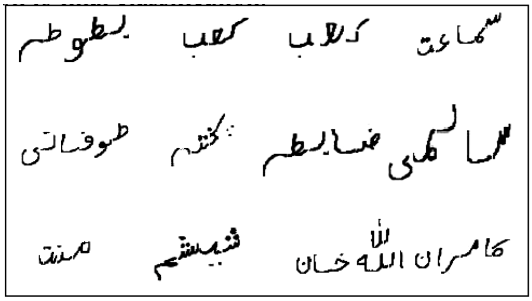

Figure 5. Examples of words composed of (segmented) handwritten half-form characters.

obtained from the participating writers, and demonstrates that the words composed of these segmented characters do resemble the words as if written continuously. To use a recognition system based on our proposed method in its current form, it is required to draw the characters in their segmented forms. If the visual feeling of continuous word is required, then the segmented characters should be drawn at appropriate positions as shown in Figure 5. We are also working on segmentation of characters from ligatures, which will be reported in the future. A related work on segmentation of handwritten Arabic text can be found in [51], which presents an efficient skeleton-based grapheme segmentation algorithm. With some modifications, this segmentation algorithm along with our proposed methodology may serve as a full system for online Urdu handwriting recognition. Segmentation of printed Urdu script can be found in [33–35].

3.1.2. Instructions for writing

For non-native audience, here, we present some instructions that should be followed while writing Urdu characters (these instructions are implicitly followed by native Urdu writers).

- There should be no pen-up event while drawing the major stroke, i.e., the major stroke should be drawn continuously without raising the pen;
- In case of multistroke characters, the major stroke should precede the minor stroke(s);
- Minor strokes should be penned one at a time, i.e., there must be pen-up events between two or three dots or between two '*kashes*.' In some cases, this instruction is violated by the native writers, but now, we stress on following this instruction.

Although two minor strokes drawn together (for example, two dots) can be separated using the variation in pressure values, this is not implemented in this paper.

3.2. Preprocessing

The raw data obtained from hardware contains artifacts like jitters, hooks at the start and end of a stroke,

procedure REMOVEREPEATEDDATAPOINTS(S)
▷ $S_{(}Mx2)$ contains X and Y coordinates of a given stroke
  initialize $k \leftarrow 1$
  initialize $Sr(k) \leftarrow S(1)$
  for $i = 2$ to M **do**
    if $\|(S(i-1) - S(i)\|_2 = 0$ **then**
      $Sr(k) \leftarrow S(i)$
    else
      $k \leftarrow k + 1$
      $Sr(k) \leftarrow S(i)$
    end if
  end for
  return Sr
end procedure

Algorithm 1. Repeated elements removed.

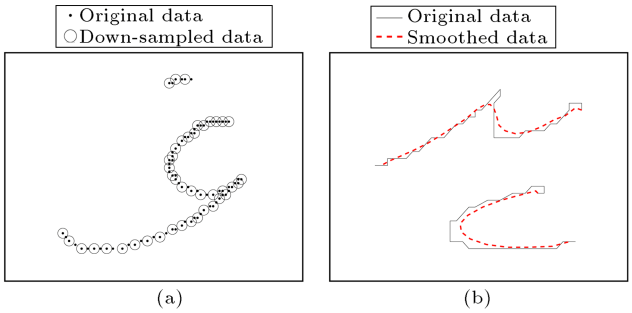

Figure 6. Preprocessing: (a) Resampling and down-sampling of character 'غ', and (b) smoothing of character 'ح'.

speed variations, etc. To reduce the effect of artifacts, the following preprocessing steps have been performed on the raw data.

3.2.1. Re-sampling

Algorithm 1 has been implemented to remove repeated data samples (those occurring consecutively in temporal order). Then, a downsampled version of this signal has been obtained by keeping every second data sample starting with the first. Few samples of downsampled data are shown in Figure 6(a).

3.2.2. Smoothing

Drawing on a tablet by inexperienced users, or roughness of pen tip or writing surface may result in jitters and trembles in writing. To mitigate jittering effects, the character data is smoothed using a 5-point moving average filter given by the following difference equation:

$$y_s(i) = \frac{1}{2N+1}(y(i+N) + y(i+N-1) + \ldots + y(i-N)), \quad (1)$$

where $y_s(i)$ is smoothed value for the ith data point, N is the number of neighboring data points on either side of $y_s(i)$ (in this case $N = 2$), and $2N + 1$ is the

span. The results of smoothing function are shown in Figure 6(b).

3.3. Pre-classification

There are many groups of characters in Urdu that share the same major stroke and differ from each other in minor strokes. These similar characters pose difficulty in classification. A concept of pre-classifier is presented here. The pre-classifier classifies the characters into smaller subgroups. The classification criterion is derived from the properties of Urdu characters as presented in Section 2. Only the pre-classification of initial half forms is explained here, because the pre-classification of medial and terminal half forms is the same.

In the first phase of pre-classification, the character set (initial half forms) is divided into different groups on the basis of the number of pen-up events. The number of pen-up events actually represents the number of strokes in a character. Four subsets are considered (see details in Section 2), namely single-, two-, three-, and four-stroke. In the second phase of pre-classification, on the basis of the position of the diacritics, every multistroke subset obtained in the first phase is segregated into two sub-subsets. Position of the diacritics for multistroke Urdu characters is either above or below the major stroke. Therefore, at the end of the second phase of pre-classification, we get 6 sub-subsets of multistroke initial half-form characters.

For Urdu characters, we place diacritics into two types on the basis of shape:

1. *Dot or nuqta* (‘ٕ’) diacritic.
2. *Other-than-dot* diacritic. They are *towey* (‘ط’), inverted *hay* (‘ہ’), *hamza* (‘ء’), and *kash* (‘ٔ’).

In the third phase of pre-classification, sub-subsets obtained in the second phase are further divided on the basis of the shape of the diacritic to produce 9 sub-sub-subsets. As a result, we get 10 subsets for the initial half-form characters. Figure 7 graphically represents pre-classification of initial half-form characters. Table 2 shows the pre- classification of Urdu character set in initial, medial, and terminal half forms.

In Urdu, it is impossible to have 3-stroke characters with *other-than-dot* diacritic below the major stroke. Four-stroke characters occur only with *dot* diacritic and no *other-than-dot* diacritic is present there. In this way of subdivision, we see that character ‘ے’, in the three-stroke group of characters, stands alone in its subset having no competition for classification. Table 3 shows the characters that face no competition in their respective subsets. They are fully recognized at pre-classification stage and do not require any further recognition.

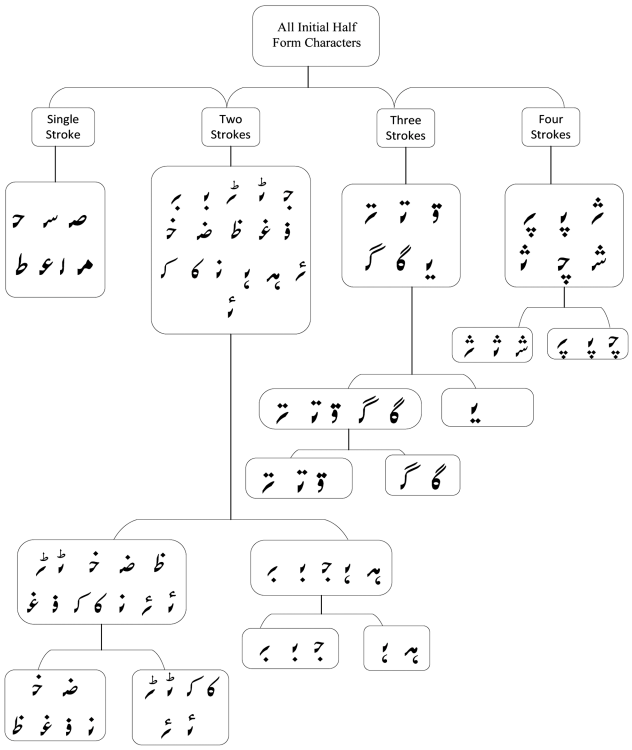

Figure 7. Pre-classification of initial half forms on the basis of stroke count, position and shape of minor strokes.

With the small subsets produced by the pre-classifier, it becomes possible to design banks of simple ANN or SVM classifiers for fine classification within the subsets.

3.4. Feature extraction

Selection of appropriate features for recognition tasks is necessary to achieve high performance [52]. Computing suitable features, in every online system, helps in reducing the computational complexity of a pattern recognition problem [45]. However, selection and extraction of such features do not follow any specific technique. Variations involved in one kind of problem manifests that a feature set designated for a particular problem may not necessarily be satisfactory for a similar problem. One can deduce that no widely accepted feature set exists yet that is universally applicable to all problems of one type [53]. To reduce computational complexity, prominent features are acquired from the preprocessed data. However, the optimum size of feature vector to recognize a handwritten character depends on the complexity involved.

For Arabic/Urdu handwritten characters recognition, different types of features have been presented in the literature, namely structural features, statistical features, and global transformation features. Using structural features [20,25,26], a model/standard template is designed for each class of letters that contains all the significant information with which test classes are compared. Statistical approach uses the

Table 2. Pre-classification of Urdu character set; the encircled numbers indicate the cardinality of final stage subsets that could be obtained with the help of the proposed pre-classifier

	Subset	Number of characters in subset	Division on minor stroke position w.r.t major stroke (above/below)	Number of characters in sub-subset	Division on diacritic type (*dot*/*other-than-dot*)	Number of characters in sub-sub-subset
Initial half forms (36 characters)	Single-stroke	**7**	×	×	×	×
	Two-stroke	17	Above	12	*dot*	**6**
					other-than-dot	**6**
			Below	5	*dot*	**3**
					other-than-dot	**2**
	Three-stroke	6	Above	5	*dot*	**3**
					other-than-dot	**2**
			Below	**1**	×	×
	Four-stroke	6	Above	3	*dot*	**3**
					other-than-dot	×
			Below	3	*dot*	**3**
					other-than-dot	×
Medial half forms (30 characters)	Single-stroke	**8**	×	×	×	×
	Two-stroke	13	Above	10	*dot*	**6**
					other-than-dot	**4**
			Below	3	*dot*	**2**
					other-than-dot	**1**
	Three-stroke	5	Above	4	*dot*	**2**
					other-than-dot	**2**
			Below	**1**	×	×
	Four-stroke	4	Above	2	*dot*	**2**
					other-than-dot	×
			Below	2	*dot*	**2**
					other-than-dot	×
Terminal half forms (42 characters)	Single-stroke	**16**	×	×	×	×
	Two-stroke	17	Above	16	*dot*	**9**
					other-than-dot	**7**
			Below	**1**	×	×
	Three-stroke	4	Above	4	*dot*	**3**
					other-than-dot	**1**
			Below	×	×	×
	Four-stroke	5	Above	4	*dot*	**4**
					other-than-dot	×
			Below	**1**	×	×

information of the underlying statistical distribution of some measurable events or phenomena of interest in the input data [22,23]. With global transformation features, the recognition problem is taken up in frequency domain using transformations like Fourier, discrete cosine, Gabor, Walsh-Hadamard, etc. [19]. However, to determine and analyze localized features of a signal/image, a time-scale representation of that signal/image, i.e., wavelet transform, is used. In [54], wavelet transform has been used for optical character

Table 3. Characters recognized at pre-classification stage that do not require any further classification.

Target group	Subset	Character	Recognition rate (%)
Initial half forms	3-stroke dot below	ڈ	100
Medial half forms	2-stroke other below	؟	100
Medial half forms	3-stroke dot below	ڈ	100
Terminal half forms	2-stroke dot below	ب	100
Terminal half forms	3-stroke other above	گ	100
Terminal half forms	4-stroke dot below	پ	100

recognition of multi-font English text. The wavelet transform is a multiresolution technique that clips data into different frequency components and then, analyzes each component with a resolution matched to its scale [55]. Wavelet series expansion of a function $f(x)$ is given in Eq. (2):

$$f(x) = \sum_{k} c_{j_o}(k)\varphi_{j_o,k}(x) + \sum_{j=j_o}^{\infty}\sum_{k} d_j(k)\,\Psi_{j,k}(x), \quad (2)$$

where $c_{j_o}(k)$ are approximation (or scaling) coefficients and $d_j(k)$ are detailed (or wavelet) coefficients [56]. Details about the wavelets can be taken from [55]; however, for a brief review of wavelet properties, one can refer to [57].

3.4.1. Wavelet Features

To discriminate characters from each other, a human reader looks for the exact location of smooth regions, sharp turns, and cusps as the landmarks of interest. With structural, statistical, and global transformation features such as those used in [19,20,22,23,25,26], it is not possible to find out these landmarks exactly. In the proposed study, wavelet transformation of handwritten stroke data enables us to accurately pinpoint the mentioned landmarks and helps to attain better recognition rates. To verify the discriminating potential of wavelet features, a multilevel one-dimensional wavelet analysis is applied to the preprocessed data. Approximation and detail coefficients are obtained for the $x(t)$ and $y(t)$ coordinates of the handwritten strokes. In order to obtain better classification accuracy and keep the feature vector as small as possible, it has been found after some trials; level 2 approximation coefficients and level 4 detail coefficients provide the best classification accuracy. The feature vector is:

$$\mathbf{W} = \left[\begin{array}{cccc} \overrightarrow{cA2}_x & \overrightarrow{cD4}_x & \overrightarrow{cA2}_y & \overrightarrow{cD4}_y \end{array}\right]^T \in \mathbb{R}^n, \quad (3)$$

where $\overrightarrow{cA2}_x$ and $\overrightarrow{cA2}_y$ are the vectors of level-2 approximation coefficients, and $\overrightarrow{cD4}_x$ and $\overrightarrow{cD4}_y$ are the vectors of level-4 detail coefficients of the one dimensional $x(t)$ and $y(t)$ signals of the stroke coordinates $(x(t), y(t))$. C++ or Matlab codes may be used to obtain the wavelet coefficients.

Figures 8 and 9 show four different handwritten characters in medial half form. Each of these figures shows the handwritten stroke and $x(t)$, and $y(t)$ presents the major stroke in the top row; the second row shows the $\overrightarrow{cA2}_x$ and $\overrightarrow{cA2}_y$ coefficients, while the third row shows the $\overrightarrow{cD4}_x$ and $\overrightarrow{cD4}_y$ coefficients.

Figure 10 provides the case in which *other-than-dot* minor stroke is involved. In this case, there are characters that have similar major strokes and are distinguishable from each other only based on the shapes of their minor strokes. Since the minor stroke in this case is significantly long, the wavelet coefficients of the minor stroke are also included along with the wavelet coefficients of the major stroke to form the feature vector.

It can easily be observed in Figures 8 to 10 that the wavelet coefficients of different characters are quite different from each other. Such variability raises the hope for the wavelet features to present good discrimination power. The results have verified that using wavelet features, in the way presented above, provides high recognition rates.

3.4.2. Structural features

In this study, for the comparison purpose, in addition to wavelet based features, structural features proposed by Khan and Haider [22,23] are also employed and tested. It is shown in the results (Section 4) that with wavelet features, the recognition accuracy is far better than that with structural features.

3.5. Classification

For fine classification of each character within the subsets produced by the pre-classifier, a dedicated classifier is designed for each of the subsets. In this work, the responses of ANN and SVM classifiers along with different input features are studied. Moreover, RNNs are also applied to compare the responses obtained through ANN and SVM.

3.5.1. Artificial Neural Networks (ANNs)

For pattern recognition problems, developing a Multilayer Perceptron (MLP) neural network with backpropagation algorithm is a very popular approach [58–62]. The ANNs used in this work are single or multilayer Back Propagation Neural Networks (BPNN). For each

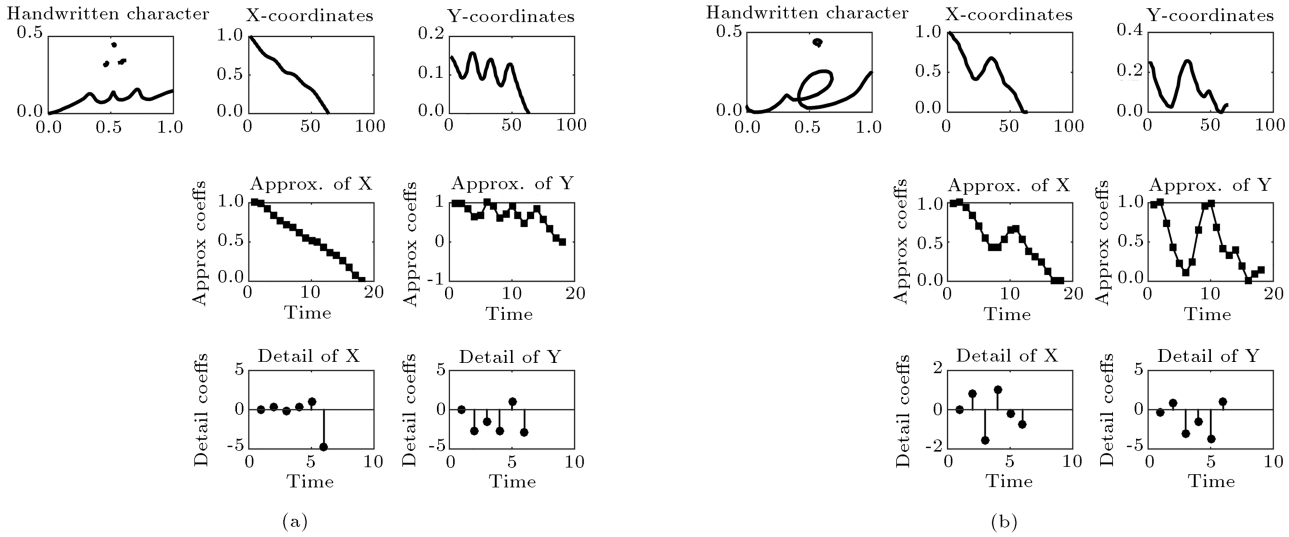

Figure 8. (a) Top row shows character *sheen* in medial form as well as $x(t)$ and $y(t)$ of its major stroke; the second and third rows show level-2 db2 wavelet approximation and level-4 db2 wavelet detail coefficients of $x(t)$ and $y(t)$, respectively. (b) Top row shows character *zwad* in medial form as well as $x(t)$ and $y(t)$ of its major stroke, the second and third rows show level-2 db2 wavelet approximation and level-4 db2 wavelet detail coefficients of $x(t)$ and $y(t)$, respectively.

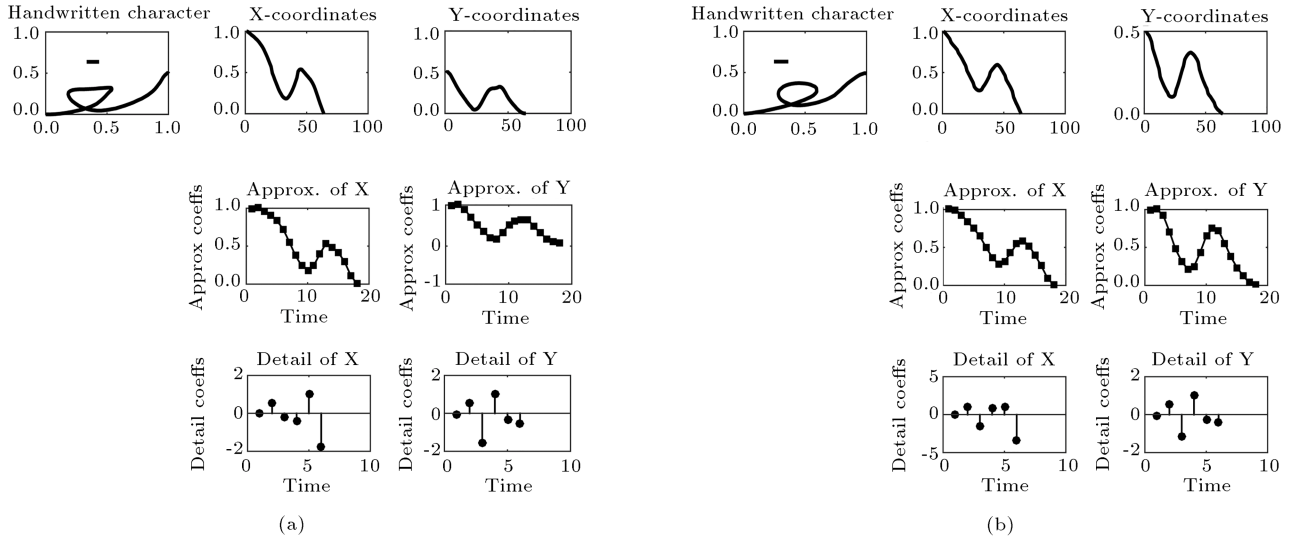

Figure 9. (a) Top row shows character *Ghain* in medial form, as well as $x(t)$ and $y(t)$ of its major stroke; the second and third rows show level-2 db2 wavelet approximation, and level-4 db2 wavelet detail coefficients of $x(t)$ and $y(t)$, respectively. (b) Top row shows character *fay* in medial form, as well as $x(t)$ and $y(t)$ of its major stroke; the second and third rows show level-2 db2 wavelet approximafayon and level-4 db2 wavelet detail coefficients of $x(t)$ and $y(t)$, respectively.

of the 19 subsets (cardinality $\geq$ 2), an ANN is configured, trained, and tested. In this way, a bank of ANNs is obtained in which each neural network serves to recognize a specific character subset. There are two different banks of ANNs:

1. ANNs which are trained using *structural* features.
2. ANNs which are trained using wavelet *db2* approximation and detailed coefficients. Table 4 presents configurations of these ANNs.

Using MATLAB environment, all ANNs have been trained by 40% (40 instances for each character) and tested by the remaining 60% (4260 samples) of the data set.

3.5.2. Support Vector Machines (SVMs)

SVMs are also widely used for pattern classification and recognition [62,63]. The specific capability of SVM is that minimization of empirical classification error and maximization of geometric margins occur simultaneously in it. To make a comparison of recognition results obtained through ANN classifiers using wavelet features, recognition results using SVM have also been obtained. Two banks of SVM classifiers

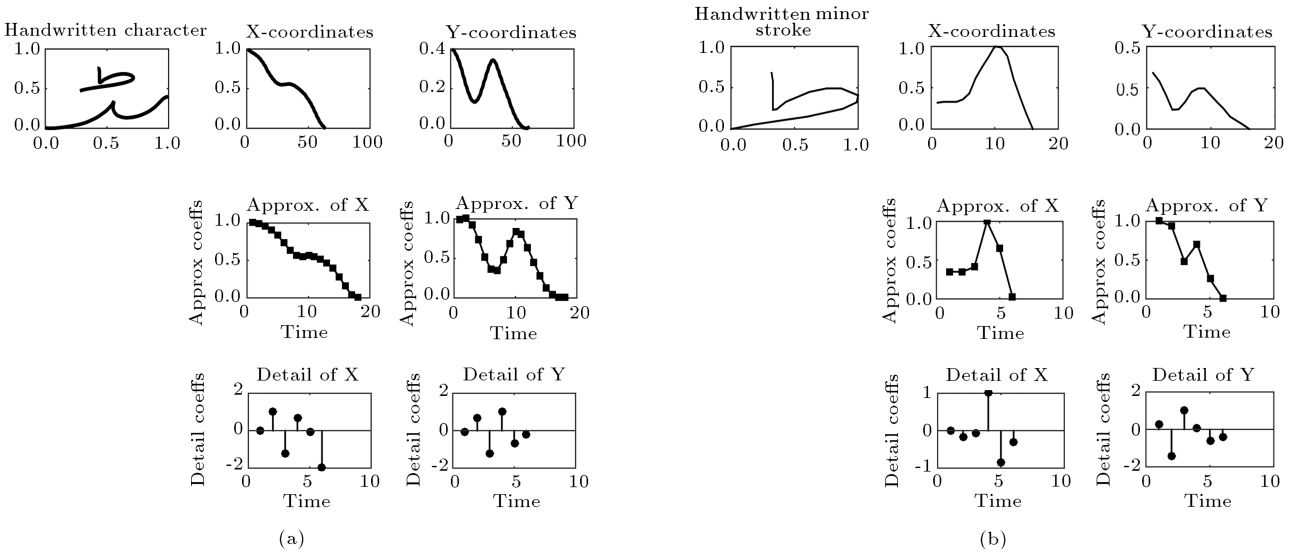

Figure 10. (a) Top row shows character *Tay* in medial form as well as $x(t)$ and $y(t)$ of its major stroke; the second and third rows show level-2 db2 wavelet approximation and level-4 db2 wavelet detail coefficients of $x(t)$ and $y(t)$, respectively. (b) Top row shows the minor stroke of *Tay* as well as its $x(t)$ and $y(t)$ coordinates; the second and third rows show the level-2 db2 wavelet approximation and level-2 db2 wavelet detail coefficients of $x(t)$ and $y(t)$ of minor stroke, respectively.

Table 4. ANN configurations (trained using wavelet *db2* approximation and detailed coefficients).

Target group	No. of hidden layers	Neurons in hidden layer 1	Neurons in hidden layer 2	Recognition rate (%)
ANN configuration: Initial half forms				
2-stroke *dot* above	2	9	6	90.2
2-stroke *other-* above	2	9	6	87.7
2-stroke *dot* below	1	1	-	94.4
2-stroke *other-* below	1	1	-	97.5
3-stroke *dot* above	2	2	3	97.7
3-stroke *other-* above	2	2	3	98.3
4-stroke *dot* above	2	6	3	89.4
4-stroke *dot* below	2	4	3	89
ANN configuration: Medial half forms				
2-stroke *dot* above	2	9	9	81.6
2-stroke *other-* above	2	8	6	91.6
2-stroke *dot* below	1	1	-	99.1
3-stroke *dot* above	2	3	3	98.3
3-stroke *other-* above	1	2	-	97.5
4-stroke *dot* above	2	4	2	97.5
4-stroke *dot* below	2	4	2	100
ANN configuration: Terminal half forms				
2-stroke *dot* above	2	7	9	93.3
2-stroke *other-* above	2	7	7	95.7
3-stroke *dot* above	1	2	-	95.5
4-stroke *dot* above	2	4	2	97.9

using wavelet features (*db2* and *bior1.3*) are trained and tested. SVM has been set up using LIBSVM (Matlab) [64]. LIBSVM offers selecting different types of kernel functions (e.g., linear, polynomial, radial basis function (RBF), sigmoid, etc.) with various parameters of the kernels. For the proposed study, C-SVM (multiclass classification) with radial basis function is employed. For the selection of good parameters, the training set is used with 5-fold cross validation and optimized values are obtained (for cost of constraint violation C and γ in radial basis function). All the SVMs are then trained with 40% of sample data randomly selected and tested on the remaining 60% of the data.

3.5.3. Recurrent neural networks: Long short-term memory

Recurrent Neural Networks (RNNs) introduce the notion of time to the traditional feedforward ANN; this enables the network to make use of the temporal patterns present in the sequential data. In a sequential set of data, the current output depends on the previously computed values. RNNs are elevated with the inclusion of edges that span the adjacent time steps. For sequence learning, Long Short-Term Memory (LSTM) and Bidirectional Recurrent Neural Networks (BRNNs) are considered to be the most successful RNN architectures. In LSTM RNNs, traditional nodes in the hidden layer of a network are replaced by a memory unit. The architecture of bidirectional recurrent neural networks utilizes the information from both the past and the future to compute the output at any point in the sequence [65]. It helps the recurrent neural networks to be applicable to cursively handwritten scripts more efficiently.

In this work, using RNNLIB [66], RNNs with LSTM architecture, without any feature extraction, and with/without using the proposed pre-classification are applied to the handwritten data. By the proposed pre-classification, each subset is presented to a recurrent neural network which is specifically trained for that subset. Results of RNN classifier without using the proposed pre-classifier have also been obtained to check the end-to-end capability of the RNN classifier. Using the raw stroke data, each RNN is trained, validated, and tested by 30%, 20%, and 50% of the randomly selected subsets of the data set, respectively.

4. Results and discussion

The pre-classifier produces a total of 28 subsets from the set of 108 half-form characters (Table 2). Out of these 28 subsets, there are 6 subsets containing only one character that do not need any further classification (Table 3). There are 3 subsets containing single-stroke characters for which some results are presented in [39]. The remaining 19 subsets contain multi-stroke characters for which 6 different combinations of classifiers and features are tried to classify the individual characters in the subsets:

1. ANN classifiers using structural features;
2. ANN classifiers using Daubechies' family *db*2 wavelet features;
3. SVM classifiers using Daubechies' family *db*2 wavelet features;
4. SVM classifiers using Biorthogonal family *bior*1.3 wavelet features;
5. RNN classifiers using single LSTM hidden layer of size 100 (no feature extraction with pre-classification);
6. RNN classifiers using multi LSTM hidden layers of varying sizes (no feature extraction with pre-classification);
7. RNN classifier using multi LSTM hidden layers of varying sizes (no feature extraction no pre-classification).

These 19 subsets contain 8, 7, and 4 subsets of initial, medial, and terminal half-form characters, respectively, containing a total of 71 multi-stroke characters. The discrimination of similar characters from each other is made easier by the pre-classifier, because it puts similar characters into different subsets. Since a subset contains quite dissimilar characters, the pre-classifier also allows the use of computationally simpler ANN or SVM classifiers for fine classification of individual characters within a subset.

The recognition results are shown in Table 5. Among these 7 classifier/feature combinations, the best overall recognition accuracy of 96.1% is obtained by using *db*2 wavelet features with SVM classifier, but SVM with *bior1.3* wavelet features also provides comparable overall accuracy of 95.8%. ANN with *db*2 wavelet features provides lower overall accuracy of 92.8%. For ANNs with structural features, the overall accuracy is 81.9%, which is significantly low as compared to those for the other 3 combinations. Note that the dataset contains 100 instances for each character; for each ANN, 40 instances are used for training purpose, while 60 instances are used for testing of the classifiers. Overall recognition results using RNNs are 84.7% (RNN with single LSTM hidden layer of size 100) and 87.2% (RNN with multi LSTM hidden layers of varying sizes).

The end-to-end recognition capability of RNN has also been checked without utilizing the proposed pre-classifier and any features. The raw stroke data for all the 108 character classes have been used to train a single RNN classifier. Different configurations have been tried for the RNN classifier. The best recognition rate obtained is 60% and the training time is more than

Table 5. Recognition rates for each subset (cardinality >1) of multistroke half form Urdu characters obtained by the pre-classifier; the results obtained using ANN, SVM, and RNN with different features are presented for comparison.

Half Form	Character Subset	Number of characters in subset	Recognition rate (%) using ANN		Recognition rate (%) using SVM		Recognition rate (%) using RNN	
			Structural features	Wavelet (db2)	Wavelet (db2)	Wavelet (bior1.3)	Single LSTM hidden layer of size 100	Multi LSTM hidden layers of varying sizes
Initial half forms (8 subsets, 28 chars)	2-Stroke *dot* above	6	81.3	90.2	99.1	98	78	84.7
	2-Stroke *other-than-dot* above	6	76.3	87.7	91.9	87.2	72	73.3
	2-Stroke *dot* below	3	92.2	94.4	97.2	97.7	88.7	94
	2-Stroke *other-than-dot* below	2	90	97.5	98.3	96.6	79	90
	3-Stroke *dot* above	3	88.8	97.7	94.4	95.5	88.7	91.3
	3-Stroke *other-than-dot* above	2	99.1	98.3	100	100	96	97
	4-Stroke *dot* above	3	77.7	89.4	88.8	87.7	83.3	85.3
	4-Stroke *dot* below	3	88.8	89	92.7	93.3	84.7	88.7
Medial half forms (7 subsets, 20 chars)	2-Stroke *dot* above	6	58	81.6	93.6	91.3	74	74
	2-Stroke *other-than-dot* above	4	80.4	91.6	93.3	94.1	73	73.5
	2-Stroke *dot* below	2	99	99.1	98.3	100	94	96
	3-Stroke *dot* above	2	95	98.3	95	98.3	81	90
	3-Stroke *other-than-dot* above	2	94.1	97.5	95.8	97.5	84	94
	4-Stroke *dot* above	2	87.5	97.5	95.8	95.8	86	86
	4-Stroke *dot* below	2	97.5	100	100	100	89	99
Terminal half forms (4 subsets, 23 chars)	2-Stroke *dot* above	9	66.6	93.3	96.7	97.2	92	92
	2-Stroke *other-than-dot* above	7	82.6	95.7	99	99.2	89.3	91.8
	3-Stroke *dot* above	3	93.3	95.5	99.4	100	96	99.3
	4-Stroke *dot* above	4	94.1	97.9	99.6	99.1	96.5	97.5
Overall Accuracy		81.9	92.9	96.1		95.8	84.7	87.2

100 hours. Note that for the RNNs, 30%, 20%, and 50% of sample data have been randomly selected for training, validation, and testing purposes, respectively.

4.1. Error analysis using confusion matrices

Some confusion matrices will be presented in this section for the best and worst cases of the best classifier/feature combination, i.e., SVM+db2-wavelet-features.

Table 6 shows the confusion matrix of a subset containing 6 characters. The recognition accuracy of 91.9% for this subset is among the lowest accuracies obtained with the SVM+db2 wavelet features combination. The character 'ﻋ' has been misclassified 3 times as ﻉ and 4 times as ﻌ. This is expectable because of the shape similarity of these characters. Similarly, ﻊ has been misclassified 7 times as ﻋ and 4 times as ﻌ for the same reason.

Table 7 shows the confusion matrix for another subset yielding low overall recognition accuracy (93.6%) with the SVM+db2 wavelet features combination. The main causes for the low accuracy in this subset are the characters ﻍ and ﻎ. Although ﻍ and ﻎ have distinct major strokes in standard form with ﻎ

Table 6. Confusion matrix for initial half-form 2-stroke characters with *other-than-dot* diacritic above the major stroke. Overall accuracy for this subset is 91.9%.

	ﺋ	ﺉ	ﮐ	ﮎ	ﭩ	ﭨ	Unknown	
ﺋ	57	2	0	0	1	0	0	60
ﺉ	3	52	1	0	4	0	0	60
ﮐ	0	0	60	0	0	0	0	60
ﮎ	0	0	1	59	0	0	0	60
ﭩ	0	4	0	0	49	7	0	60
ﭨ	2	2	0	0	2	54	0	60
	62	60	62	59	56	61	0	

Table 7. Confusion matrix for medial half-form 2-stroke characters with *dot* diacritic above the major stroke. Overall accuracy for this subset is 93.6%

	ﻏ	ﻍ	ﺦ	ﺯ	ﻅ	ﺿ	Unknown	
ﻏ	51	5	0	1	0	3	0	60
ﻍ	7	51	0	1	0	1	0	60
ﺦ	0	0	60	0	0	0	0	60
ﺯ	1	1	0	57	1	0	0	60
ﻅ	0	0	1	0	59	0	0	60
ﺿ	0	0	0	0	1	59	0	60
	59	57	61	59	61	63	0	

Table 8. Confusion matrix for medial half-forms 2-stroke characters with *other-than-dot* diacritic above the major stroke. Overall accuracy for this subset is 93.3%

	ﮈ	ﮑ	ﮒ	ﮋ	Unknown	
ﮈ	56	1	0	3	0	60
ﮑ	0	60	0	0	0	60
ﮒ	0	3	57	0	0	60
ﮋ	9	0	0	51	0	60
	65	64	57	54	0	

having a cusp in its major stroke, many writers ignore this cusp when handwriting ﻍ casually. Therefore, ﻍ appears very much similar to ﻒ. This is confirmed by the confusion matrix, which shows that ﻍ has been misclassified 7 times as ﻒ. Removing ﻍ and ﻒ from this sub set results in 97.9% accuracy. Removing only ﻒ results in 95% accuracy, while removing only ﻍ gives 97.9% accuracy.

The confusion matrix of another subset yielding low overall accuracy of 93.3% is presented in Table 8. Here, ﮈ and ﮋ are responsible for the low recognition rate. Both characters have the same major stroke, but distinct minor strokes; thus, minor stroke has also been utilized for feature vector formation. However, casual penning of minor strokes results in similar shapes of the minor strokes. Consequently, ﮋ has been misclassified 9 times as ﮈ.

Tables 9 and 10 present two subsets showing high overall recognition accuracy.

4.2. Confusing characters

In Urdu, there are few groups of characters in which the major stroke is common to the group and the discrimination is made on the basis of minor strokes. This similarity is inherent to Urdu. The similar characters are put into different subsets by the pre-classifier. There is another kind of similarity between different characters, which arises from the careless

Table 10. Confusion matrix for terminal half-form 4-stroke characters with *dot* diacritic above the major stroke. Overall accuracy for this subset is 99.6%

		ﭽ	ﺙ	ﺵ	ﮊ	Unknown
ﭽ	60	0	0	0	0	60
ﺙ	0	60	0	0	0	60
ﺵ	0	0	59	1	0	60
ﮊ	0	0	0	60	0	60
	60	60	59	61	0	

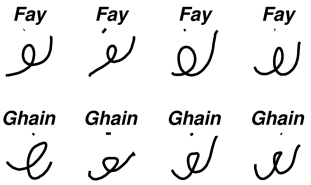

Figure 11. Handwritten samples of ﻒ (*Fay*) and ﻍ (*Ghain*). The ﻍ (*Ghains*) are confusingly similar to the ﻒ (*Fays*).

writing by the user. This user imposed similarity occurs inside the subsets produced by the pre-classifier and results in confusing pairs of characters within a subset.

Figure 11 shows few handwritten samples of two confusing characters ﻒ (*Fay*) and ﻍ (*Ghain*) present in the subset presented in Table 7. If drawn according to rules, the character ﻍ should have a well-defined cusp in its major stroke. Some users do not draw the cusp while writing casually or in hurry. The ﻍ drawn in this way appears like ﻒ to a human reader, as can be seen in Figure 11. The classifier also misclassifies ﻍ as ﻒ many times, as shown in Table 7.

Another pair of confusing characters is shown in Figure 12. They are ﮈ (*hamza*) and ﮋ (*Tay*) in medial form. The major strokes for both the characters is the same and the discrimination is made based on the minor stroke. Many users casually draw the minor

Table 9. Confusion matrix for terminal half-form 2-stroke characters with *dot* diacritic above the major stroke. Overall accuracy for this subset is 96.7%

	ﻑ	ﻍ	ﺝ	ﺥ	ﻥ	ﺯ	ﺭ	ﻅ	ﺹ	Unknown	
ﻑ	58	0	0	0	0	0	0	0	2	0	60
ﻍ	0	54	0	6	0	0	0	0	0	0	60
ﺝ	0	1	59	0	0	0	0	0	0	0	60
ﺥ	0	4	0	56	0	0	0	0	0	0	60
ﻥ	0	0	0	0	60	0	0	0	0	0	60
ﺯ	0	0	0	0	0	58	2	0	0	0	60
ﺭ	0	0	0	0	0	0	60	0	0	0	60
ﻅ	0	0	0	1	0	1	0	58	0	0	60
ﺹ	1	0	0	0	0	0	0	0	59	0	60
	59	59	59	63	60	59	62	58	61	0	

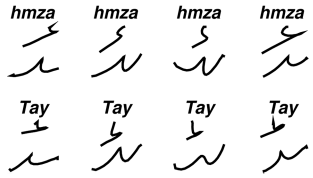

Figure 12. Handwritten samples of ئ (*hamza*) and ﻁ (*Tay*). The ﻁ (*Tays*) are confusingly similar to the ئ (*hamzas*).

stroke of ﻁ in a very much similar way to the minor stroke of ئ. The confusion matrix in Table 8 for this subset confirms this; it can be seen that ﻁ has been misclassified 9 times as ئ.

4.3. RNN and pre-classification

Urdu character subsets achieved by pre-classification are introduced (configuration-wise) to two types of RNNs. In the first set of RNNs, the configuration of each RNN consists in a single LSTM hidden layer of size 100 trained separately for each character subset. Its overall recognition rate is 84%. In the second set of RNNs, each RNN is configured with multi LSTM hidden layers of different sizes. After many hit and trials, the best RNNs are selected and the recognition rate obtained is 87%. Comparison of recognition rates obtained through fixed-size and varying-size configurations shows that the latter is more acceptable. Recurrent neural networks are also applied to handwritten Urdu data without going through preclassification process. All the 108 classes are introduced to only one RNN. After many hit and trials, the best RNN thus obtained achieves a recognition rate of 60%. It can be observed that without pre-classification, the recognition rate is substantially lower than that with pre-classification. The recognition rate may be further improved for the RNN if more data is added to the handwritten Urdu database. However, in the context of the current study, the difference among the three results obtained with RNNs shows that due to the complexity and similarity of Urdu characters, the proposed pre-classification is helpful in obtaining better results.

5. Conclusions

In this study, a novel character recognition system for online handwritten Urdu language characters is presented. All multi-stroke initial, medial, and terminal half-form characters were recognized. A large-scale handwriting data set was obtained from 100 native Urdu writers of different age groups and educational qualifications. The data was acquired using a digitizing tablet. Spatial coordinates in temporal order with their respective pressure values and pen up/down events were recorded. The raw data was refined after its manipulation with different preprocessing operations. A novel preclassifier was designed to preclassify Urdu characters set into smaller subsets. The preclassifier yielded smaller subsets based on the number of strokes to give two-, three-, and four-stroke subsets. The pre-classifier further divided the subsets based on the position of the minor stroke with respect to the major stroke, and on the basis of whether the minor stroke was a *dot* or *other-than-dot*. The pre-classifier helped in discriminating similar characters from each other by putting them in different subsets. Two types of features, namely structural and wavelet transform, were extracted. Wavelet features were obtained using Daubechies *db2* coefficients and Biorthogonal *bior1.3* coefficients. Artificial Neural Network (ANN), Support Vector Machines (SVM), and Recurrent Neural Network (RNN) classifiers were used for fine classification of the individual characters in the subsets generated by the pre-classifier. Results of RNN classifier without using the proposed pre-classifier and features were also obtained to check the end-to-end capability of the RNN classifier. Since there was no sufficient previous work for comparison, different combinations of features and classifiers were tried to find the best recognition results. Seven different classifier/feature combinations were tried, which resulted in overall accuracies of 81.9%, 92.8%, 95.8%, and 96.1% with classical approaches and 84.7%, 87.2%, and 60% with RNNs. The best overall recognition rate of 96.1% was found for SVM+db2-wavelet-features combination. For individual characters, recognition rates obtained were between 80% to 100% and overall accuracy for different subsets was between 88.8% to 100% for SVM+db2-wavelet-features combination. We followed the segmentation-based approach, which required extraction of half forms of characters from the ligatures. The data was actually obtained in segmented form from the users. Research on segmentation of ligatures into half-form characters was also carried out parallel to this work. The end-to-end recognition capability of RNNs was also explored, and it yielded inferior results to the classical feature-based approaches of SVM and ANN. The results with RNNs may be improved if more data is added to the database. In future, with increase in the size of database, other deep learning methods like deep belief networks and convolutional neural networks may be employed. Other kinds of features may also be explored.

Acknowledgement

The authors are thankful to the Higher Education Commission (HEC) of Pakistan for funding this work.

References

1. Ghods, V. and Sohrabi, M.K. "Online Farsi handwritten character recognition using hidden Markov model",

Journal of Computers, **11**(2), pp. 169-175 (2016).

2. Mahasukhon, P., Mousavinezhad, H., and Song, J.Y. "Hand-printed English character recognition based on fuzzy theory", *2012 IEEE International Conference on Electro/Information Technology*, pp. 1–4, ISSN: 2154-0357 (2012). DOI:10.1109/EIT.2012.6220772
3. Khodadad, I., Sid-Ahmed, M., and Abdel-Raheem, E. "Online Arabic/Persian character recognition using neural network classifier and DCT features", *IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS)* (2011).
4. Yao, C. and Cheng, G. "Approximative bayes optimality linear discriminant analysis for Chinese handwriting character recognition", *Neurocomputing*, **207**, pp. 346–353, ISSN: 0925-2312 (2016). DOI: https://doi.org/10.1016/j.neucom.2016.05.017. URL: http://www.sciencedirect.com/science/article/pii/S0925231216303551
5. Xiang-Dong, Z., Da-Han, W., Feng, T., et al. "Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**, pp. 2413-2426 (2013). doi: 10.1109/TPAMI.2013.49
6. Peng, L., Liu, C., Ding, X., et al. "Multi-font printed mongolian document recognition system", *International Journal on Document Analysis and Recognition (IJDAR)*, **13**(2), pp. 93–106, ISSN 1433-2825 (2010). DOI:10.1007/s10032-009-0106-8
7. Rao Kunte, R.S. and Sudhaker Samuel, R.D. "Online character recognition for handwritten Kannada characters using wavelet features and neural classifier", *IETE Journal of Research*, **46**, pp. 387-393 (2000).
8. Tagougui, N., Kherallah, M., and Alimi, A.M. "Online arabic handwriting recognition: a survey", *International Journal on Document Analysis and Recognition (IJDAR)*, **16**(3), pp. 209–226, ISSN1433-2825 (2013). DOI:10.1007/s10032-012-0186-8URL: http://dx.doi.org/10.1007/s10032-012-0186-8
9. Harouni, M., Mohamad, D., Shafry, M.M.R., et al. "Handwritten Arabic character recognition based on minimal geometric features", *International Journal of Machine Learning and Computing*, **2**, pp.578-582 (2012).
10. Kherallah, M., Bouri, F., and Alimi, A. "On-line arabic handwriting recognition system based on visual encoding and genetic algorithm", *Engineering Applications of Artificial Intelligence*, **22**(1), pp. 153–170, ISSN 0952-1976 (2009). DOI: https://doi.org/10.1016/j.engappai.2008.05.010 URL: http://www.sciencedirect.com/science/article/pii/S0952197608001176
11. Kherallah, M., Haddad, L., Alimi, A.M., et al. "On-line handwritten digit recognition based on trajectory and velocity modeling", *Pattern Recognition Letters*, **29**(5), pp. 580–594, ISSN 0167-8655 (2008). DOI:https://doi.org/10.1016/j.patrec.2007.11.011 URL: http://www.sciencedirect.com/science/article/pii/S0167865507003662
12. Mahmoud, S.A. and Mahmoud, A.S. "Arabic character recognition using modified Fourier spectrum (MFS) vs, Fourier descriptors", *Cybernetics and Systems: An International Journal*, **40**(3), pp. 189–210 (2009). DOI:10.1080/01969720802714758
13. Alimi, A.M. "An evolutionary neuro-fuzzy approach to recognize on-line arabic handwriting", *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, **1**, pp. 382–386 (1997). DOI:10.1109/ICDAR.1997.619875
14. Hamdani, M., Abed, H.E., Kherallah, M., et al. "Combining multiple hmms using on-line and off-line features for off-line Arabic handwriting recognition", *2009 10th International Conference on Document Analysis and Recognition*, pp. 201–205, ISSN 1520-5363 (2009). DOI:10.1109/ICDAR.2009.40.
15. "Telecom indicators", [Online accessed, 14-June-2016], Pakistan Telecommunication Authority (2016). URL: http://www.pta.gov.pk/index.php? Itemid=599
16. Baloch, F. "Telecom sector: Pakistan to have 40 million smartphones by end of 2016", URL: http://tribune.com.pk/story/953333/telecom-sector-pakistan-to-have-40-million-smartphones-by-end-of-2016/
17. Javed, S.T., Hussain, S., Maqbool, A., et al. "Segmentation free Nastalique Urdu OCR", World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, **4**, pp. 1514-1519 (2010).
18. Satti, D.A. and Saleem, K. "Complexities and implementation challenges in offline Urdu nastaliq OCR", *Conference on Language & Technology 2012 (CLT12)*, University of Engineering & Technology (UET), Lahore, Pakistan (2012).
19. Naz, S., Hayat, K., Razzak, M.I., et al. "The optical character recognition of Urdu-like cursive scripts", *Pattern Recognition*, Elsevier, **47**, pp. 1229–1248 (2014).
20. Malik, S. and Khan, S.A. "Urdu online handwriting recognition", *IEEE International Conference on Emerging Technologies* (2005).
21. Shahzad, N., Paulson, B., and Hammond, T. "Urdu Qaeda: recognition system for isolated Urdu characters", *IUI Workshop on Sketch Recognition* (2009).
22. Haider, I. and Khan, K.U. "Online recognition of single stroke handwritten Urdu characters", *IEEE 13th International Multitopic Conference (INMIC2009)* (2009).
23. Khan, K.U. and Haider, I. "Online recognition of multi-stroke handwritten Urdu characters", *Image Analysis and Signal Processing (IASP)* (2010).
24. Shabbir, S. and Siddiqi, I. "Optical character recognition system for Urdu words in Nastaliq font", *International Journal of Advanced Computer Science and Applications*, **7**(5), pp. 567–576 (2016).

25. Hussain, M. and Khan, M.N. "Online Urdu ligature recognition using spatial temporal neural processing", *IEEE International Multitopic Conference (INMIC05)* (2005).

26. Husain, S.A., Sajjad, A., and Anwar, F. "Online Urdu character recognition system", *IAPR Machine Vision Applications (MVA2007)* Conference on (2007).

27. Razzak, M.I., Anwar, F., Hussain, S.A., et al. "HMM and fuzzy logic-A hybrid approach for online Urdu script-based languages' character recognition", *Knowledge-Based Systems*, Elsevier, **23**(8), pp. 914-923 (2010).

28. Razzak, M.I., Hussain, S.A., Abdulrahman, A.M., et al. "Bio-inspired multilayered and multilanguage arabic script character recognition system", *International Journal of Innovative Computing Information and Control*, **8**(4), pp. 2681–2691 (2012).

29. Pal, U. and Sarkar, A. "Recognition of printed Urdu script", *7th International Conference on Document Analysis and Recognition (ICDAR' 03)* (2003). DOI:0-7695-1960-1/03.

30. Ahmad, Z., Orakzai, J.K., Shamsher, I., et al. "Urdu Nastaleeq optical character recognition", *Internation Journal of Computer, Information, Systems and Control Engineering*, **1**, pp. 2374–2377 (2007).

31. Lehal, G.S. "Choice of recognizable unit for Urdu OCR", *Workshop on Document Analysis and Recognition (DAR12)* (2012). DOI:10.1145/2432553.2432569

32. Zaman, S., Slany, W., and Saahito, F. "Recognition of segmented Arabic/Urdu characters using pixel values as their features", *ICCIT* (2012).

33. Javed, S.T. and Hussain, S. "Segmentation based Urdu Nastalique OCR", *18th Iberoamerican Congress (CIARP2013)*, pp. 41-49 (2013). DOI:10.1007/978-3-642-41827-3_6

34. Naz, S., Umar, A.I., Bin Ahmed, S., et al. "An OCR system for printed Nasta'liq script: A segmentation based approach", *IEEE 17th International, Multi-Topic Conference (INMIC'2014)*, pp. 255–259 (2014). DOI:10.1109/INMIC. 2014.7097347

35. Naz, S., Arif, I.U., Ahmad, R., et al. "Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks", *Neurocomputing*, **177**, pp. 228–241 (2016).

36. Razzak, M.I., Sher, M., and Hussain, S.A. "Locally baseline detection for online arabic script based languages character recognition", *International Journal of the Physical Sciences*, **5**(7), pp. 955–959 (2010).

37. Razzak, M.I., Hussain, S.A., Khan, M.K., et al. "Handling diacritical marks for online arabic script based languages character recognition using fuzzy c-mean clustering and relative position", *Information-An International Interdisciplinary Journal*, **14**(1), pp. 157–165 (2011).

38. Razzak, M.I., Husain, S.A., Mirza, A.A., et al. "Fuzzy based preprocessing using fusion of online and offline trait for online Urdu script based languages character recognition", *International Journal of Innovative Computing, Information and Control*, **8**(5(A)), pp. 3149–3161 (2012).

39. Safdar, Q. and Khan, K.U. "Online Urdu handwritten character recognition: initial half form single stroke characters", *12th International Conference on Frontiers of Information Technology*, pp. 292–297 (2014). DOI:10.1109/FIT.2014.61

40. Patel, D.K., Som, T., Yadav, S.K., et al. "Handwritten character recognition using multiresolution technique and Euclidean distance metric", *Journal of Signal and Information Processing*, **3**, pp. 208–214 (2012).

41. Wei, W., Ming, L., Weina, G., et al. "A new mind of wavelet wransform for handwritten Chinese character recognition", *Second International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC)* (2012).

42. Aburas, A. and Rehiel, S.M.A. "Off-line omni-style handwriting Arabic character recognition system based on wavelet compression", *Arab Research Institute in Sciences & Engineering ARISER*, **3**(4), pp. 123–135, ISSN 1994-3253 (2007).

43. Mowlaei, A., Faez, K., and Haghighat, A.T. "Feature extraction with wavelet transform for recognition of isolated handwritten Farsi/Arabic characters and numerals", *IEEE 13th Workshop on Neural Networks for Signal Processing, NNSP'03*, pp. 547–554, ISSN 1089-3555 (2003). DOI:10.1109/NNSP.2003.1318054

44. Jenabzade, M.R., Azmi, R.B.P., and Shirazi, S. "Two methods for recognition of handwritten Farsi characters", *International Journal of Image Processing (IJIP)*, **5**(4), pp. 512–520 (2011).

45. Singh, P. and Budhiraja, S. "Handwritten Gurmukhi character recognition using wavelet transform", *International Journal of Electronics, Communication & Instrumentation Engineering Research and Development*, **2**(3), pp. 27–37 (2012).

46. Primekumar, K.P. and Idiculla, S.M. "On-line Malayalam handwritten character recognition using wavelet transform and SFAM", *3rd International Conference on Electronics Computer Technology (ICECT)*, **1** (2011).

47. Abed, H.E., Märgner, V., Kherallah, M., et al. "Icdar 2009 online Arabic handwriting recognition competition", *2009 10th International Conference on Document Analysis and Recognition*, pp. 1388–1392, ISSN 1520-5363 (2009). DOI:10.1109/ICDAR.2009.284

48. Zhang, X.Y., Yin, F., Zhang, Y.M., et al. "Drawing and recognizing Chinese characters with recurrent neural network", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**(4), pp. 849–862, ISSN 0162-8828 (2018). DOI:10.1109/TPAMI.2017.2695539

49. Graves, A. "Supervised sequence labelling with recurrent neural networks", Ph.D. Thesis, Technical University Munich (2008). URL: http://d-nb.info/99115827X

50. Javed, S.T. and Hussain, S. "Improving Nastaliq specific prerecognition process for Urdu OCR", *13th IEEE International Multitopic Conference* (2009).
51. Abandah, G.A. and Jamour, F.T. "Recognizing handwritten Arabic script through efficient skeleton-based grapheme segmentation algorithm", *10th International Conference on Intelligent Systems Design and Applications* (2010).
52. Wahi, A., Sundaramurthy, S., and Poovizhi, P. "Recognition of handwritten Tamil characters using wavelet", *International Journal of Computer Science & Engineering Technology (IJCSET)*, **5**, pp. 335–340 (2014).
53. Jaeger, S., Manke, S., Reichert, J., et al. "Online handwriting recognition: the NPen++ Recognizer", *International Journal of Document Analysis and Recognition, IJDAR*, **3**(3), pp. 169–180 (2001).
54. Al-Hassani, M.D. "Optical character recognition system for multifont English texts using DCT and wavelet transform", *International Journal of Computer Engineering and Technology (IJCET)*, **4**(6), pp. 48–61 (2013).
55. Mallat, S., *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press Elsevier Inc., San Dieago (2008).
56. Gonzalez, R.C. and Woods, R.E., *Digital Image Processing*, 3rd Ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, ISBN 013168728X (2006).
57. Amar, C.B., Zaied, M., and Alimi, A. "Beta wavelets. synthesis and application to lossy image compression", *Advances in Engineering Software*, **36**(7), pp. 459–474, ISSN 0965-9978 (2005). URL: http://www.sciencedirect.com/science/article/pii/S0965997805000116
58. Murru, N. and Rossini, R. "A Bayesian approach for initialization of weights in backpropagation neural net with application to character recognition", *Neurocomputing*, **193**, pp. 92–105, ISSN0925-2312 (2016). DOI:https://doi.org/10.1016/j.neucom.2016.01.063 URL: http://www.sciencedirect.com/science/article/pii/S0925231216001624
59. Prieto, A., Prieto, B., Ortigosa, E.M., et al. "Neural networks: An overview of early research, current frameworks and new challenges", *Neurocomputing*, **214**, pp. 242–268, ISSN 0925-2312 (2016). DOI: https://doi.org/10.1016/j.neucom.2016.06.014 URL: http://www.sciencedirect.com/science/article/pii/S0925231216305550
60. Shamsher, I., Ahmad, Z., Orakzai, J.K., et al. "OCR for printed Urdu script using feed forward neural network", *World Academy of Science, Engineering and Technology*, **1** (2007).
61. Salameh, W.A. and Otair, M.A. "Online handwritten character recognition using an optical backpropagation neural network", *Issues in Informing Science and Information Technology*, **2**, pp. 787–795 (2005).
62. Theodoridis, S. and Koutroumbas, K., *Pattern Recognition*, 4th Ed., Academic Press, ISBN 1597492728, 9781597492720 (2008).
63. John, S.T. and Nello, C., *Kernel Methods for Pattern Analysis*, Cambridge University Press, ISBN 0521813972 (2004).
64. Chang, C.C. and Lin, C.J. "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, **2**, pp. 1–27 (2011).
65. Lipton, Z.C. "A critical review of recurrent neural networks for sequence learning", CoRR, ArXiv (2015). URL: http://arxiv.org/abs/1506.00019
66. Graves, A. "Rnnlib: A recurrent neural network library for sequence learning problems", http://sourceforge.net/projects/rnnl/.
67. Jannoud, I.A. "Automatic Arabic handwritten text recognition system", *American Journal of Applied Sciences*, **4**, pp. 857–864 (2007).
68. Asiri, A. and Khorsheed, M.S., *Automatic Processing of Handwritten Arabic Forms Using Neural Networks*, World Academy of Science, Engineering and Technology, ISSN 1307-6884 (2005).
69. Broumandnia, A., Shanbehzadeh, J., and Varnoosfaderani, M.R. "Persian/Arabic handwritten word recognition using M-band packet wavelet transform", *Image and Vision Computing*, **26**, pp. 829–842 (2008).

Biographies

Quaratul Ain Safdar is a PhD scholar at PIEAS, Islamabad, IR Pakistan. She received her MS degree in Computer Science from University of Central Punjab, Lahore, IR Pakistan in 2005. Her research interests include pattern recognition and Urdu handwriting recognition.

Kamran Ullah Khan is with the Department of Electrical Engineering at Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, IR Pakistan. He received his BSc from UET Peshawar in 1998, MSc from Quaid-e-Azam University, Islamabad, in 2000, and PhD from Tsinghua University, Beijing, in 2008, all in Electrical Engineering.

Liangrui Peng is currently an associate professor in the Department of Electronic Engineering, Tsinghua University, Beijing, China. Her research interests include multilingual document image recognition and understanding, machine learning, and pattern recognition. She is a member of the IEEE and the ACM.