IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 8, 2021, accepted October 25, 2021, date of publication October 27, 2021,
date of current version December 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3123726

Real-Time Pashto Handwritten Character Recognition Using Salient Geometric and Spectral Features

MUHAMMAD SHABIR[1], NAVEED ISLAM[1], ZAHOOR JAN[1], INAYAT KHAN[2], TAJ RAHMAN[3], ASIM ZEB[4], SHAFIQ AHMAD[5], ABDELATY EDREES ABDELGAWAD[5], AND MALI ABDOLLAHIAN[6]

[1]Department of Computer Science, Islamia College University, Peshawar 25120, Khyber Pakhtunkhwa, Pakistan
[2]Department of Computer Science, University of Buner, Buner 19290, Pakistan
[3]Department of Physical and Numerical Science, Qurtuba University of Science and Technology Peshawar, Peshawar 25000, Pakistan
[4]Department of Computer Science, Abbottabad University of Science and Technology, Abbottabad 22010, Khyber Pakhtunkhwa, Pakistan
[5]Department of Industrial Engineering, School of Science, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[6]School of Science, College of Sciences, Technology, Engineering, Mathematics, RMIT University, Melbourne, VIC 3001, Australia

Corresponding author: Asim Zeb (asim.zeb@aust.edu.pk)

This work was supported by the King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP-2021/387.

ABSTRACT Pashto scripts are cursive in nature and hard to recognize in real-time. Native speakers of the Pashto language are large in numbers and reside in different regions of the world. Due to the cursive nature of the Pashto script along with variations in character strokes, the printed, as well as handwritten characters, are difficult to be detected, classified or recognized. In real-time handwritten character recognition systems, the challenging factors that constraints the system depends on the stroke noise, geometric behavior (like rotation, scaling and shifting, etc.) of the text. In this article, we provide an efficient technique that aimed to recognize handwritten characters in real-time by first smoothing the noise components in the text and then extract shape-based invariant features from the handwritten strokes. For real-time recognition of characters, the probability-based multi-class Naïve Bayesian classifier is exploited, which determines the probabilities of geometric invariant features to predict the character with the highest likelihood. The performance of the proposed approach has been validated through extensive experiments and based on the recognition matrices, the proposed technique achieves an accuracy of 97.5% for online Pashto handwritten characters in real-time.

INDEX TERMS Invariant features, naive Bayesian, Pashto characters recognition, real-time.

I. INTRODUCTION

The writing style of a language depends on its character set. The character set is a collection of symbols from which the words are constructed. There are different character sets, i.e. Arabic, Latin, Cyrillic, Hangeul, Devanagari, Greek, Syllabaries, Kanji and Chinese logography, etc. Most of the existing known languages use the alphabet sets without any change like European languages uses Latin character set. In Latin script languages, concatenation of characters takes place to construct a valid word, i.e. word like mango in the English language is shown in Fig. 1.

Similarly, some other languages like Arabic, Urdu, Balti, Kashmiri, Dari, Farsi, Shino and Pushto etc. use Arabic and Perso-Arabic characters. In Arabic and Perso-Arabic script-based language, characters are connected with each other in different positions to form a valid word of the language, as given in Fig. 2.

M	+	A	+	N	+	G	+	O	=	MANGO
T	+	A	+	B	+	L	+	E	=	TABLE
C	+	A	+	N	+	D	+	I	=	CANDI

FIGURE 1. Concatenated characters to form words.

Pashto characters connect vertically at a different position with each other to form words, i.e., characters connected from the front, back, or from both sides [1]. Unlike most Latin scripts written from left to right, Pashto script is written from right to left. Each character is constructed from one or more

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

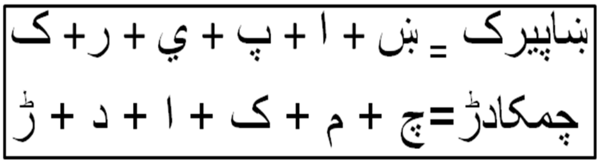

FIGURE 2. Connected characters to form words.

strokes, i.e., primary and secondary stroke [24]. A stroke is a continuous movement of an input device or fingers over the surface, such as a touchpad, touch screen, or tablet. Primary stroke is the skeleton of a word, and secondary stroke is the additional stroke to make a character different from each other. A stroke starts at the point of contact between input devices and a touch-sensitive surface. Similarly, it ends by disconnecting the input device from the surface.

The main problem investigated in this paper with real-time handwritten character recognition systems is geometric invariance i.e., rotation, scaling, and shifting. The existing systems impose constraints on users to write characters on the baseline with a specific size without any rotation. It is tough to write characters of equal size without any rotation on the exact location. In this article, we proposed a technique, which provides solutions to the mentioned problems in real-time for Arabic and Perso-Arabic script-based languages but we will test it on Pashto characters. The main contributions of this article can be summarized as follows:

1. We provide a technique for recognizing invariant handwritten Pashto characters in real-time.
2. The proposed technique extracts features from real-time handwriting characters in a novel way, which are invariant to translation, rotation, and scale. These features are based on geometric and spectral properties of handwritten characters.
3. The proposed technique achieves real-time performance by incorporating parallel execution of features via multi-threading.
4. We proposed Extended RDP, which reshapes the strokes in fewer smoothed points. Thus, reducing the time and space complexity of the proposed approach. In Extended RDP, we divided a character into two parts, both parts executed concurrently. In traditional RDP, the character is processed sequentially, which can slow down the processing speed.

The remaining paper is divided into sections as follows.

Section 2 pointed the previous related work. Section 3 elaborates the proposed approach used in this research. Section 4 consists of the experimental results and discussion. Section 5 draws the conclusion of this research.

II. RELATED WORK

Handwritten character recognition systems have been studied for the last three to four decades, and numerous researchers have developed specialized techniques. Most of these techniques are developed to recognize characters that belong to a particular language or script [23]. These techniques fail to accurately recognise slightly deviated characters in terms of rotation, shifting, and scaling. Similarly, these systems constrain a particular writing style, allowing users to diversify handwriting applications. These constraints are in terms of writing characters on baseline, ascender, and descender.

Most Arabic and Perso-Arabic character recognition systems use machine learning techniques (MLT) to classify and recognize characters with their respective character sets [6]. Artificial Neural Network (ANN) based techniques have been used in the literature for a real-time script [10], [36], [34]. A different form of ANN i.e. Feed Forward Neural Network (FNN), Back Propagation Neural Network (BNN), Convolution Neural Network (CNN), and Recurrent Neural Network (RNN), etc. are used to achieve high accuracy regardless of the space and time complexity of the system. ANN is used for the recognition of real-time handwriting scripts with statistical, spectral, and geometric features. These systems have issues with geometrically variant characters, i.e. scaling, rotating, and shifting, along with the imposition of constraints on the user to write on ascender, descender and baseline during writing. These issues make the system difficult to be used for a common user on portable devices, i.e. smartphones, tablets, personal digital assistants, etc.

Support Vector Machine (SVM) has been used in handwritten character recognition systems [9], [12], [17], [27], [30], [33]. A linear SVM differentiates two classes of objects by providing an optimal separating line. Similarly, multiclass classification of characters is achieved by modifying SVM to produce hyperplane by finding margins with the help of support vectors. These systems have the issue of time complexity and take a massive amount of time to converge. SVM-based techniques have the drawback of incapability of recognizing geometric variant characters.

Naive Bayes is a probabilistic supervised machine learning algorithm used for character classification and recognition [1], [26], [35]. Naïve Bayes classifier is very efficient for a real-time application having large data sets. The existing online or real-time handwriting recognition systems based on the Naive Bayes technique are computationally less expensive as compared with ANN and SVM-based techniques. However, these systems are unable to recognize geometric invariant characters. These techniques are also inconvenient to use for a common user due to writing constraints.

For real-time handwriting character recognition, fuzzy rule-based techniques are proposed in the literature [14], [15], [21], [22], [24]. Fuzzy rules-based techniques have the drawback of being dependent on the baseline. Invariant characters are not considered to be recognized in these systems. These techniques required a large number of rules for classification and recognition, which is difficult to manage in case of some error. Similarly, in [11], [28], the weighted linear classifier is used with local features in which weights are assigned to each extracted feature vector. This technique also has the incapability of geriatric variation.

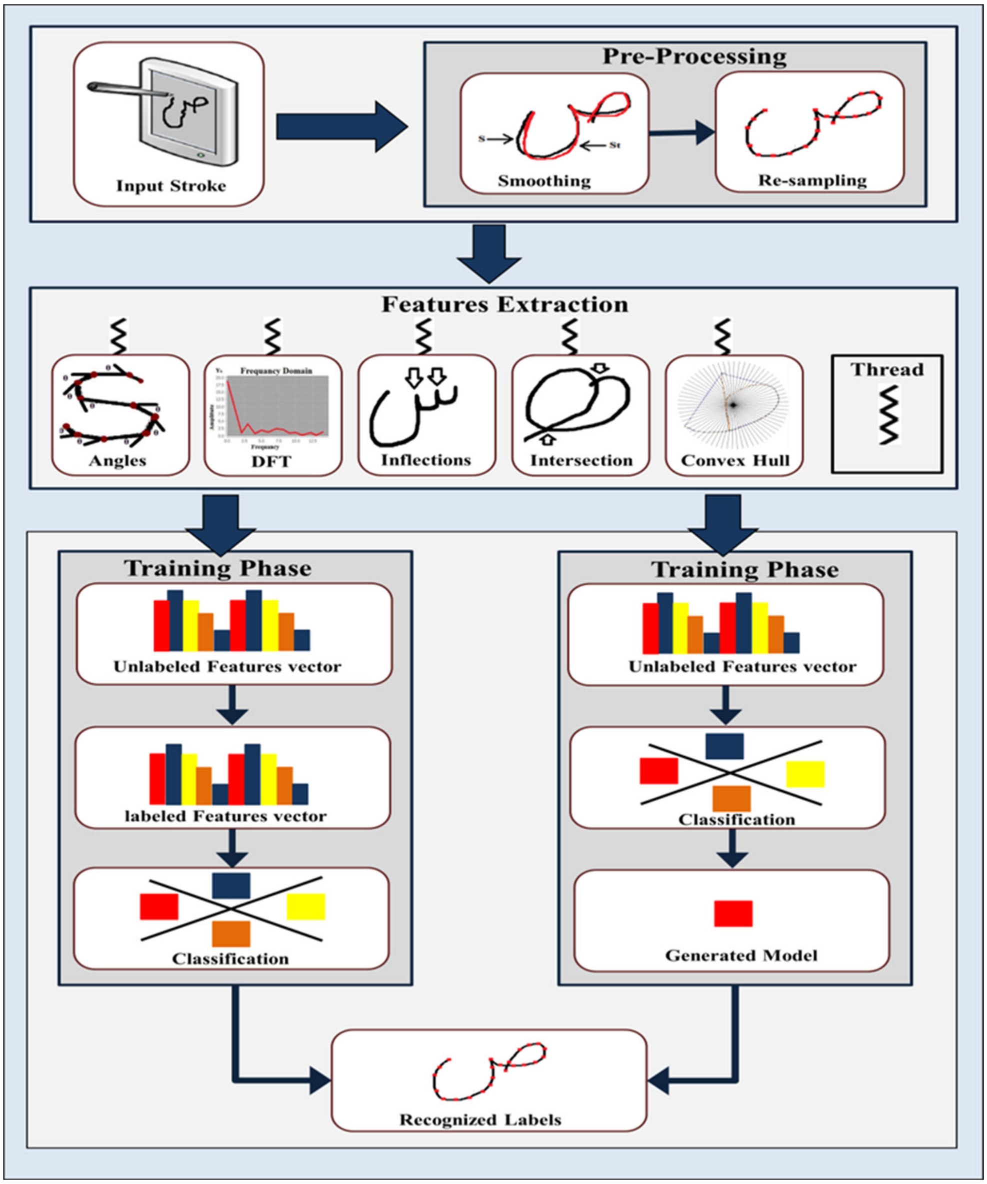

FIGURE 3. The proposed frame work.

Large number of systems use Hidden Markov Model (HMM) for real-time handwriting character recognition [2], [4], [5], [7], [13], [14], [16], [18], [19], [32]. The time complexity of these techniques is less than other state-of-the-art techniques due to the probabilistic approach for the transition from one state to the other. However, these techniques are unable to recognize rotated, scaled, and shift-invariant characters. These techniques have also inconvenient to use for a common user due to constraints on handwriting.

Template matching techniques are also used for character recognition in the literature [3], [7], [8], [25], [29], [31]. Like other techniques, template matching also suffers from geometric invariant data. Matching templates, i.e. one against all, is time-consuming and less accurate due to slight changes in data.

Recent work published as well as online or real-time recognition of Arabic characters is concerned [38]. In this research, they used LSTM and BLSTM for classification and recognition. The baseline for character alignment is considered, meaning classification and recognition will decrease if words are below or above the baseline.

In recent years, research work has published on Gurmukhi character recognition [39] in which Neural Network is used for classification. This work meets the requirement of real-time handwriting recognition. The Gurmukhi characters are very different from the characters based on Arabic and Perso-Arabic.i.e. Pashto. The algorithm design for Gurmukhi character recognition cannot be applied directly to Pashto characters. We do not know from the proposed research that either they consider the geometric variations of characters or not.

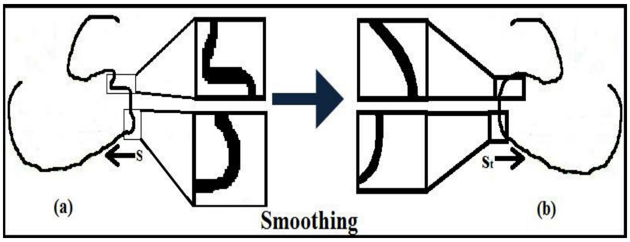

FIGURE 4. (a) Trembling noise in handwritten character (b) Result after smoothing.

III. THE PROPOSED APPROACH

User interaction with computing devices has become more convenient through touch-sensitive surfaces. Due to the massive usage of these devices, millions of applications are developed in the last couple of years. Ordinary users understand only natural languages and common symbols to communicate with the system. A large number of Latin script-based applications are developed for touch-sensitive devices to make the interaction easier. Less attention has been given to develop applications with Arabic and Perso-Arabic script-based languages. We proposed a generalized technique for recognizing online geometric invariant handwritten characters for Arabic and Perso-Arabic script-based languages. The proposed approach can be divided into three significant steps: pre-processing, feature extraction, and character classification. The flow diagram of the proposed approach is given in Fig. 3, whereas the detail about each step is provided in the following subsections.

A. PRE-PROCESSING

In pre-processing, smoothing of handwritten stroke is performed, followed by re-sampling in a sequence. Various character artifacts occur to touch-sensitive devices during real-time handwriting. These handwritten strokes consist of different vector trajectories having trembling noise. Trembling noise is produced due to hand shivering on the input device. If S represents any character stroke define by its components, i.e. $S_i = \{S_1, S_2, \ldots, S_n\}$ where S_1 is the starting point, and $S_j = n$ is the ending point in. Each point S_i in stroke trajectory is connected to neighbouring point S_j where $(i \neq j)$. In the proposed approach, the trembling noise at each point S_t in a stroke is reduce using (1).

$$S_c = S_{t-1} \times (1 - \alpha) + S_t \times \alpha \tag{1}$$

In (1) S_t is the point to be smoothed, S_{t-1} is the prior smoothed point and represents the amount of smoothing where $(\alpha \epsilon [0, 1])$, and S_c indicates the current point in stroke trajectory.

Fig. 4(a) represents a handwritten character example where trembling noise can be observed in irregular character shapes. Fig. 4(b) shows the resultant smoothed character after performing the smoothing operation.

We proposed a parallel computing approach called Enhanced Ramer Douglas Peucker (ERDP), based on Ramer Douglas Peucker (RDP) algorithm. The stroke trajectory of Pashto characters consists of a large number of data points. For low-resource devices, it is computationally expensive to process such a huge amount of data in real-time. The proposed ERDP approach performed fast computation than RDP and increased the overall speed of the system. In this approach, vertices of each character are defined by polyline P where $P = \{P_1, P_2, \ldots, P_f\}$. The step-wise procedure of ERDP is shown in Fig. 5 and explained as follows:

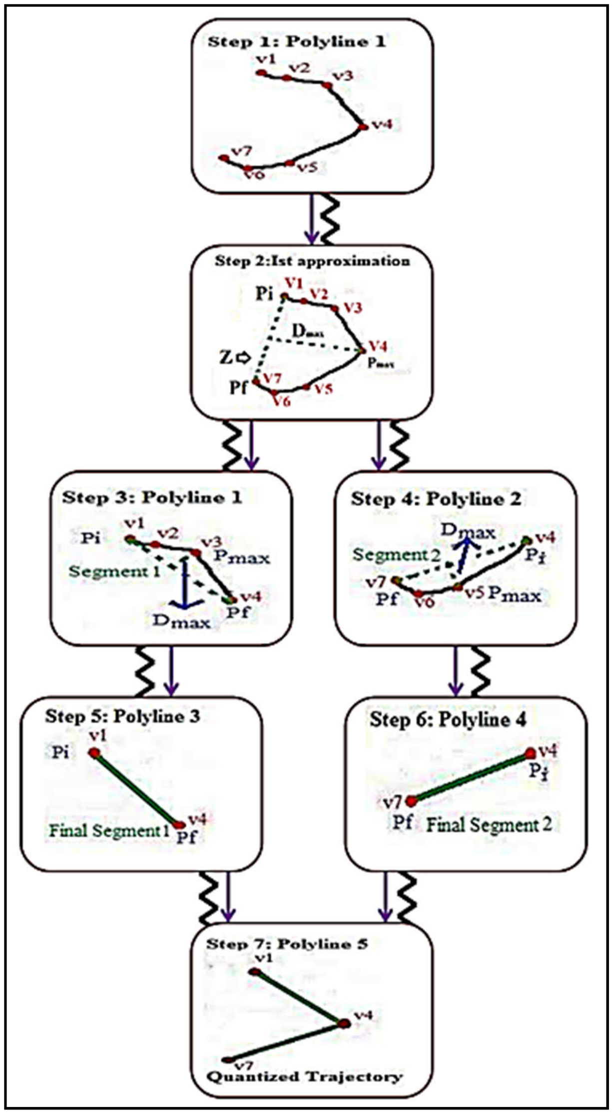

FIGURE 5. Re-sampling character ډ.

Step 1: Let Z be the line segment connecting initial point P_i and final point P_f of a polyline P. Point P_i has two coordinates, i.e., Pi1 and Pi2, point Pf has two coordinates, i.e., P_{f1} and P_{f2}. The line segment Z can be computed by finding the slope m of two points P_i and P_f. The Slope of these two points can be computed by using (2).

$$m = ((Pf1 - Pf))/((Pi1 - Pi2)) \tag{2}$$

The line segment Z or line equation can be determined using a point-slope formula (3).

$$(P_{f1} - P_{f2}) = m \times ((P_{i1} - P_{i1}) + (P_{i1} - P_{i2})) \tag{3}$$

Step 2: P_{max} represents the farthest point on polyline P from the line segment Z. The Farthest point Pmax is determined by using the Euclidean distance (4).

$$P_{max} = \sqrt{(P_i)^2 + (P_f)^2} \tag{4}$$

where D_{max} is the distance between line segment Z and farthest point P_{max}.

Step 3: Based on a threshold value ε, the polyline P is further split into two polylines i.e. polyline P_1 and P_2.

Step 4: The splitting process of polyline stops when the distance between the line segment Z and farthest point is less than ε.

B. FEATURE EXTRACTION

In the proposed method, after smoothing and re-sampling, different geometric invariant features are extracted on the separate thread. Multi-threading makes the feature extraction process fast and decreases the computational complexity. The feature data related to characters are defined in a feature space F_i with five dimensions i.e., $F_i = \{f_1, f_2, f_3, f_4, f_5\}$, where f_1 represents the cosine angle, f_2 is angles based spectral feature, f_3 is the frequency of inflection angles, f_4 represents self-intersection of character stroke and f_5 represents the angles of the convex hull. The determination of all these features are explained as follows:

1) THE ANGLE BETWEEN LINE SEGMENTS OF A CHARACTER

The first feature extracted from character stroke is the cosine of the angle between the line segments of a stroke trajectory. Character is segmented into n number of line segments collectively called l_i, i.e. $l_i = \{l_1, l_2, \ldots, l_n\}$. Cosine of angle between two consecutive line segments is determined by taking their dot product as follows:

$$l_1.l_2 = |l_1| \times |l_2| \times \cos\theta(5) \tag{5}$$

$$\theta = cos^{-1}\left(\frac{l_1.l_2}{|l_1| \times |l_2|}\right)(6) \tag{6}$$

$|l_1|$ is the length of vector l_1, $|l_2|$ is the length of vector l_2 and $(l_1.\ l_2)$ is their dot product. Since there will be n-1 angles for f_{1i} each character, can be computed efficiently and stored for further process. We are interested only in angles with a higher degree than a threshold value, i.e. $r >= 90^0$.

Fig. 6 (a) shows the graphical representation of cosine of angles between line segments l_1 and l_2 denoted by θ_j, and Fig. 6(b) shows the online handwritten character along with θ_j values projected on each point of the character.

2) ANGLE BASED SPECTRAL FEATURE

the proposed method uses the spectral feature by taking the Discrete Fourier Transform (DFT) of the cosine angles given

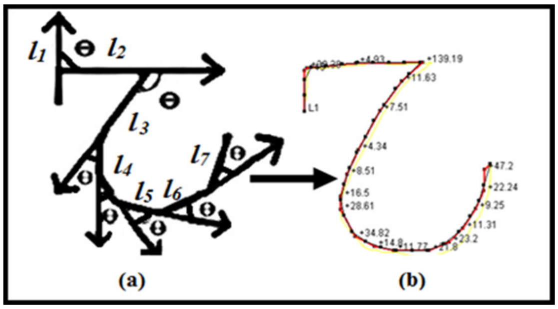

FIGURE 6. (a) Cosine angle (b) Values of trajectory.

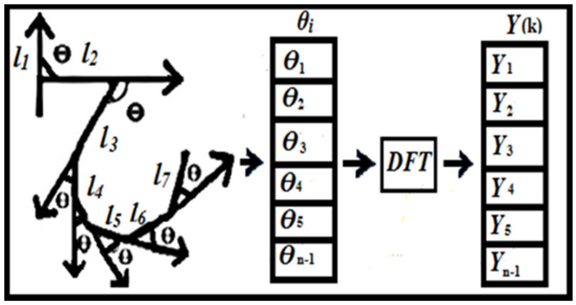

FIGURE 7. DFT of the cosines of angles between the line segments Fig 8 is the result of proposed technique in which DFT is computed for character ص in frequency domain.

in Fig.6 (b). The DFT of the one-dimensional vector θ(i) of size n-1 is given by (7).

$$Y(k) = \sum_{i=1}^{n-1} \theta(i)\, e^{-\frac{2\pi jki}{n}} \tag{7}$$

where Y(k) is DFT calculated value, θ(i) is the one-dimensional vector representing the trajectory cosine angles. n-1 is the number of angles in θ(i), θ is the current angle in θ(i) as shown in Fig.7.

3) FREQUENCY ANGLES OF INFLECTION

Pashto characters are cursive in nature and have some useful angles of inflection. Inflected angles are cosine angles of character trajectory, which suddenly have a big change in angle. A threshold value T is set to get an inflected angle. i.e. $\theta_i >= T$. The frequency of inflected angles is computed as shown in Fig. 9. Each character can be differentiate based on the frequency of the angle of inflection.

4) SELF-INTERSECTIONS OF CHARACTER

Due to the cursive nature of the pashto language, self-intersections are quite common within the characters. If the stroke trajectory intersects itself, then it is defined to be self-intersection. Fig. 10 demonstrates the intersection point p(x) and p(y) of two-line segments h_1 and h_2, where h_1 consists of points $p(x_1)$, $p(y_2)$, $p(x_2)$ and $p(y_2)$, similarly h_2 consists of points $q(x_1)$, $q(y_1)$, $q(x_2)$ and $q(y_2)$. The point of intersection p (x, y) depends on the slope of h_1 and h_2, respectively. The

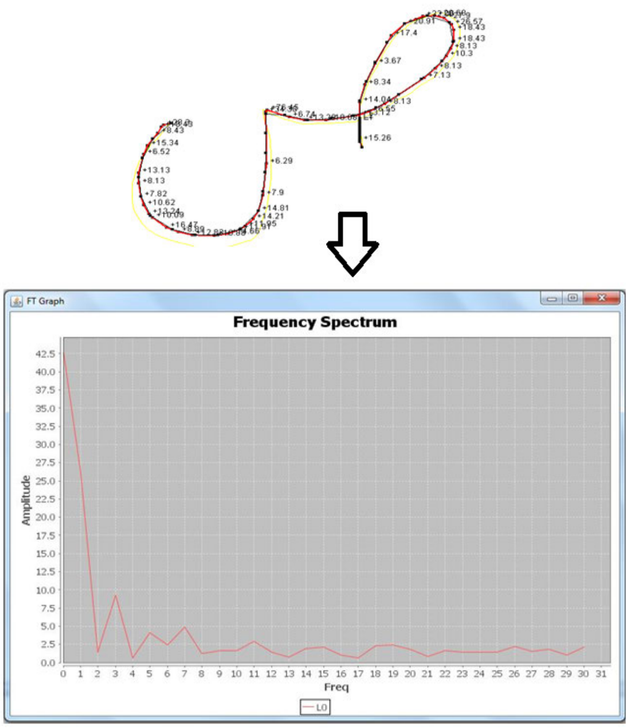

FIGURE 8. DFT generated by the *proposed* system for character ص.

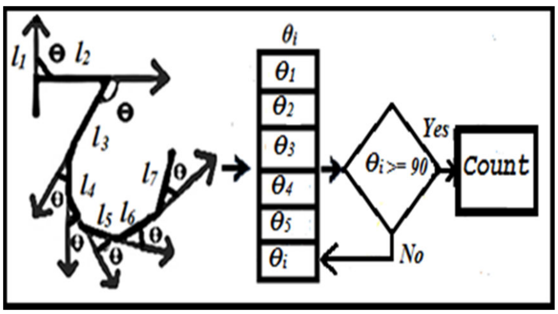

FIGURE 9. Inflection points in character.

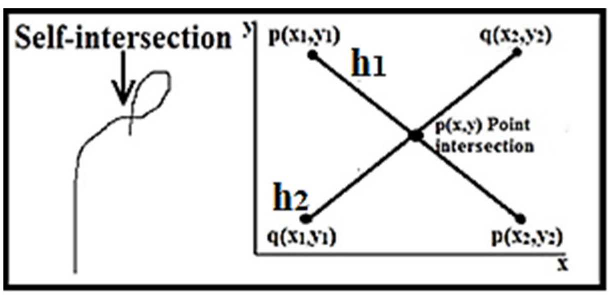

FIGURE 10. Self-intersection of character.

slope of each line segment can be computed by using (8).

$$h_1 = \frac{q(y_2)-q(y_1)}{q(x_2)-q(x_1)}, \quad h_2=\frac{p(y_2)-p(y_1)}{p(x_2)-p(x_1)}$$

$$\text{INTERSECTION} = \begin{cases} 1, & 0 \le h_1 \le 1 \text{ AND } 0 \le h_2 \le 1 \\ 0, & \text{OTHERWISE} \end{cases} \tag{8}$$

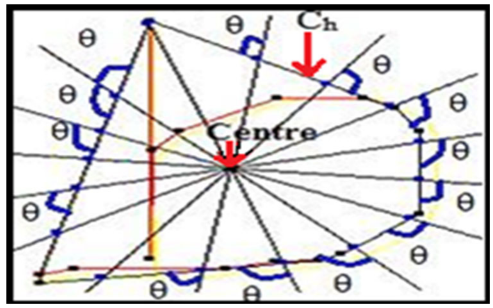

FIGURE 11. Convex hull of a character.

If the conditions $0 \le h1 \le 1$ and $0 \le h2 \le 1$ are satisfied, then the line segments intersect and we count 1 a self-intersection.

5) INTERSECTING ANGLES OF CONVEX HULL

Another geometric invariant feature used in the proposed technique is the convex hull. We draw a convex hull by using an algorithm of a finite planer set [40] as shown in Fig. 11. We draw a line in some directions with a little space, starting from the center and intersect the convex hull boundary. Angles of intersection between line segments and convex hull boundary are computed by using (8).

C. CLASSIFICATION AND RECOGNITION

Classification is a process to separate different classes of objects from each other based on features. The parallel features extraction technique is used to extract data from each handwriting stroke. A feature vector is extracted for each character using the proposed approach explained in section 2.2. Pashto characters classification and recognition is tedious work and needs an efficient and light-weighted classifier.

We used the Naïve Bayes Classification technique for classification because it is computationally efficient compared to other classification techniques like SVM or ANN. In the Naïve Bayes technique, the probability of character h i.e. P(h), called the prior probability, is computed, determined from the whole data set of handwritten characters during the training phase. P(d) is the probability of a new character being recognized, obtained from the whole data set in the recognition phase. P (d|h) is the probability of a character data d when the prior probability of h was true.

We are interested in calculating the posterior probability, i.e. P(h|d) from the prior probability P(h), new character probability P(d) and P(d|h), which is given as follows:

$$P(h|d) = \frac{(P(d|h) \times P(h))}{P(d)} \tag{9}$$

The predictive class is the one with a maximum posterior probability (MAP) which is given below:

$$MAP(h) = max(P(h|d)) \tag{10}$$

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In Pashto script, some characters are composed of a single stroke and some are multi-strokes, similarly, some single and

Single stroke characters		Multi stroke characters (All characters are Complex)
Simple characters	ل, د, ا, ں, ک, ر,	خ, ف, ي, ۏ, ث, غ, ج, ړ, پ, ۀ, گ, ۆ, ټ, ج, ز, ګ, چ, ئ, ة, ظ, ا, خ, ژ, غ, گ, ۓ, و, ل, ڤ, ز, ڈ, ٹ, ة, ګ, پ, ڕ, چ, ئ, ك, خ, ۀ, ګ, ڤ, ړ, ج, ۆ, ي, ج, ف, ۏ, ڽ, ڽ, ډ, ټ, ۆ, ز, ۆ, ڵ, ښ, ډ, ټ, ز, أ, ذ, ۆ, ل, ڦ, ژ, ډ, ب, ن, آ, و, ل, ف, ژ, ډ, ث, ذ, ڎ, ڀ, أ, ض, غ, ۆ, ڻ, ك, ڝ, ڎ, ٿ, ي, ئ, ض, ۏ, ک, ڜ, ڍ, پ, ڜ, ا, ڜ, ۏ, ڹ, ڮ, ڟ, ڏ, خ, أ, ب, ۿ, ۆ, ڽ, ڪ, ڞ,
Complex characters	ح, ہ, صد, سں, ع, ٯ, ے, ی, ص, ی, س, و, ک, م, ء, ے, ع, لا, کھ, لھ, کھ, ھ, ط,	

FIGURE 12. Single and multi-stroke syntactically simple and complex classes.

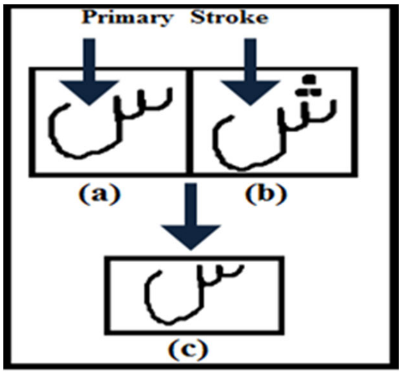

FIGURE 13. (a) Pashto Character pronounced as "seen" (b) Pashto Character pronounced as "sheen" (c) Common primary stroke of (a) and (b).

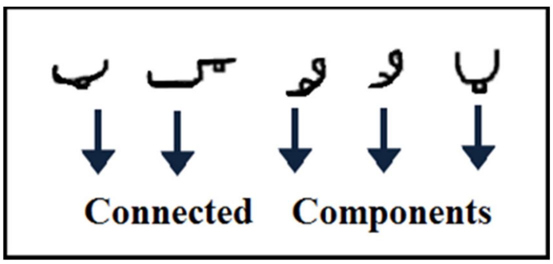

FIGURE 14. Character with additional connected component.

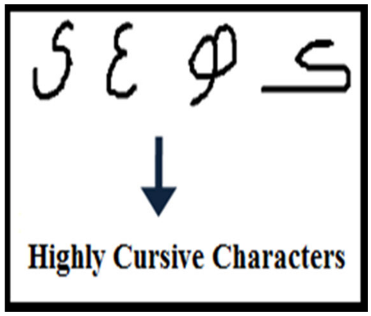

FIGURE 15. Highly cursive characters.

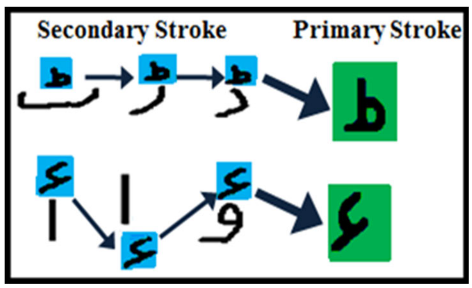

FIGURE 16. Similar primary and secondary stroke.

multi-stroke characters are syntactically simple, and some are complex as shown in Fig. 11. This figure also consist of some non-Pashto language characters. The recognition rate decreases due to the complex syntactical structure of some characters. The complexity is shown in Fig. 12.

There are some other problems with Pashto characters, which are given below:

1. Many characters shared a common primary structure with a little change. Fig. 13 (a) represents character pronounced as "Seen", Fig. 13 (b) represents character pronounced as "Sheen" and Fig. 13 (c) represents the primary stroke of the character Seen and Sheen. Due to high resemblances in a primary stroke, the feature vectors are much closed to each other.
2. Some characters have additional complex connected components i.e. Fig. 14. The additional connected component must be detected otherwise, the characters will mismatch with other characters.
3. Some characters are highly cursive in nature i.e. Fig. 15. The improper handwritten curve makes the character similar to other characters.
4. A very serious problem with Pashto characters is that some secondary strokes are similar to primary stroke i.e. Fig. 16. A small size primary stroke used as a secondary stroke with another primary stroke to make a character different from other characters.

For experimentation of the proposed approach, data set of 44 unique Pashto characters were used. These characters

Arabic and Perso-Arabic Characters with Unicode									
0600 ـ	0629 ة	0638 ظ	0673 ٳ	0682 ڂ	0691 ڑ	A006 ڠ	AF06 گ	BE06 ھ	CD06 ۍ
0601 ؁	A062 ت	0639 ع	0683 ڃ	0692 ڒ	A106 ڡ	B006 ڰ	BF06 ڿ	CE06 ێ	0602 ؂
B062 ث	A063 غ	0675 ٵ	0684 ڄ	0693 ړ	A206 ڢ	B106 ڱ	C006 ۀ	CF06 ۏ	0603 ؃
C062 ج	0641 ف	0676 ٶ	0685 څ	0694 ڔ	A306 ڣ	B206 ڲ	C106 ہ	D006 ې	B060 ؋
D062 ح	0642 ق	0677 ٷ	0686 چ	0695 ڕ	A406 ڤ	B306 ڳ	C206 ۂ	D106 ۑ	060E ؎
E062 خ	0643 ك	0678 ٸ	0687 ڇ	0696 ږ	A506 ڥ	B406 ڴ	C306 ۃ	D206 ے	060F ؏
F062 د	0644 ل	0679 ٹ	0688 ڈ	0697 ڗ	A606 ڦ	B506 ڵ	C406 ۄ	D306 ۓ	0621 ء
0630 ذ	0645 م	A067 ٺ	0689 ډ	0698 ژ	A706 ڧ	B606 ڶ	C506 ۅ	D506 ە	0622 آ
0631 ر	0646 ن	B067 ٻ	A068 ڊ	0699 ڙ	A806 ڨ	B706 ڷ	C606 ۆ	EE06 ۮ	0623 أ
0632 ز	0647 ه	C067 ټ	B068 ڋ	A069 ښ	A906 ک	B806 ڸ	C706 ۇ	EF06 ۯ	0624 ؤ
0633 س	0648 و	D067 ٽ	C068 ڌ	B069 ڛ	AA06 ڪ	B906 ڹ	C806 ۈ	FA06 ۺ	0625 إ
0634 ش	0649 ى	E067 پ	D068 ڍ	C069 ڜ	AB06 ګ	BA06 ں	C906 ۉ	FB06 ۻ	0626 ئ
0635 ص	A064 ي	F067 ٿ	E068 ڎ	D069 ڝ	AC06 ڬ	BB06 ڻ	CA06 ۊ	FC06 ۼ	0627 ا
0636 ض	0671 ٱ	0680 ڀ	F068 ڏ	E069 ڞ	AD06 ڭ	BC06 ڼ	CB06 ۋ	FF06 ۿ	0628 ب
0637 ط	0672 ٲ	0681 ځ	0690 ڐ	F069 ڟ	AE06 ڮ	BD06 ڽ	CC06 ی		

FIGURE 17. Pashto characters from arabic and perso-arabic set along with unicode.

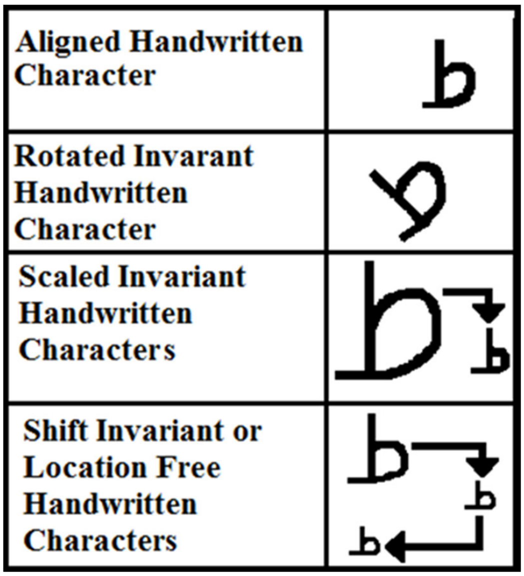

FIGURE 18. Rotated, Scaled and shift invariant characters.

are taken from Arabic and Perso-Arabic character sets, as shown in Fig. 17. The researchers often ignore the character at Unicode B906. However, we used it in the experimentation.

To recognize online or real-time rotated, scaled, and shift-invariant handwritten characters, data from different subjective sources, i.e., teachers and students, has been collected using Genius easy pen M506 tablet. They wrote characters with different variations, which are shown in Fig. 18. 2000 samples of each Pashto handwritten invariant character were taken and used as a training dataset. The total number of character samples used in the training phase was $44 \times 2000 = 88{,}000$.

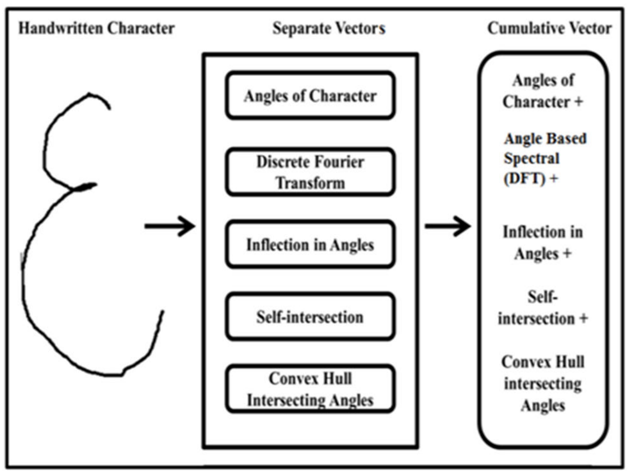

FIGURE 19. Vector construction for a character.

Features such as the angle between line segments of a character, angle-based Spectral feature (DFT), Frequency angles of inflection, Character-intersections, and convex hull intersection angles were extracted from handwritten characters from teachers and students.

To construct a cumulative vector for each character, all feature vectors have been concatenated as given in Fig. 19.

In order to find the recognition accuracy of the proposed technique, 2000 samples of characters have been taken from 200 different teachers.

We have not split the dataset into training and testing parts. The testing phase was in real-time entry by the teachers. The testing samples are new and unknown to the system. The recognition accuracy of the teacher's handwriting characters was measured using a confusion matrix, as shown in Table 1.

TABLE 1. Confusion matrix for teachers handwriting.

		Predict	
Actual	**Total (2000)**	**Yes**	**No**
	Yes	TP (974)	FP (24)
	No	FN (25)	TN (975)

TABLE 2. Confusion matrix for students handwriting.

		Predict	
Actual	**Total (2000)**	**Yes**	**No**
	Yes	TP (956)	FP (44)
	No	FN (42)	TN (958)

TABLE 3. Accuracy and error rate in percent.

Users	Accuracy	Error Rate	Accuracy in % = Accuracy × 100	Error Rate in %= Error Rate × 100
Teachers	0.975	0.025	97.5%	2.5%
Students	0.957	0.043	95.7%	4.3%

We express recognition accuracy by using the following equation:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{11}$$

where TP is True Positive 974, TN is True Negative 975, FP is False Positive 24, and FN is False Negative 25 samples, respectively. These values are put into (12), and we get 0.975 accuracies. We also calculate the error rate of our proposed technique by using (13), which is 0.025.

$$Error\ Rate = (1 - \text{Accuracy}) \tag{12}$$

Similarly, to find the recognition accuracy of the proposed system, 2000 samples were taken from 100 students.

The recognition accuracy of the student's handwriting characters was measured via a confusion matrix as given in Table 2.

The TP is 956, TN is 958, FP is 44, and FN is 42 samples. By putting these values into (12), we get 0.957 accuracies. To find the error rate for students' handwriting, we use (13), which gives us a 0.043 error rate.

We also express recognition accuracy in the present for better understanding. To change the accuracy and error rate, we multiply it by 100, as given in Table 3.

The variation in accuracy for different users depends on the usage of touch-sensitive devices.

All the comparative systems in Table 4 are either for other languages or not tested for real-time Pashto handwritten characters recognition.

TABLE 4. Comparative analysis of the proposed technique with other techniques.

Systems	Rotated, Scaled and shift Invariant	Methods	Accuracy in %
This Technique	Yes	Invariant Naïve Bayes	97.5%
[2]	No	HMM + ANN	97.6 %
[3]	No	String Matching	91.5 %
[4]	No	HMM	90 %
[5]	No	HMM	96.4 %
[7]	No	Template Matching	86.9 %
[8]	No	HMM	16.3% Error rate
[9]	No	SVM	97.2%
[10]	No	Template Matching	2.3% Error Rate
[11]	No	ANN	96%
[12]	No	ANN	Not Clear
[13]	No	SVM	80 %
[14]	No	HMM	2% Error Rate
[15]	No	RNN	74%
[16]	No	Tree based dictionary	93%
[17]	No	HMM	9% Error rate
[18]	No	SVM	92.4 %
[19]	No	HMM	95.7 %
[20]	No	HMM	89%
[21]	No	GA,HS	96 %
[22]	No	ANFC	68.6%
[25]	No	Fuzzy Rule + HMM	89.2 %
[26]	No	Template DTW	88.59%
[27]	No	MMCNN	99.7%
[28]	No	SVM	98.2%
[29]	No	Weighted linear classifier	92.8%
[30]	No	Template Matching	92%
[31]	No	SVM	83%
[32]	No	LTM Matching	87.6 %
[33]	No	HMM	9.3% Error Rate
[34]	No	SVM + HMM	96.7%
[35]	No	CNN	97.3%
[36]	No	SVM + KNN	98.6%
[37]	No	RNN	Not Clear
[39]	No	LSTM and BLSTM	Not Clear
[40]	Unknown	ANN	93.5%

V. CONCLUSION

In handwritten Pashto language, the cursive nature of the written scripts along with the connecting and non-connecting characters and words make it challenging for real-time recognition systems. There are various approaches being exploited in the literature for the recognition of characters, which basically emphasizes the preprocessing of the user interface by applying ascenders, descenders, and baselines.

In this article, an efficient technique is proposed, which recognizes the characters in real-time by first applying a smoothing technique to remove the trembling noise, followed by a feature extracting process to extract geometric invariant features from the characters and then apply a multi-class Naïve Bayesian Classifier to recognize handwritten characters. It has been observed through experiments that the proposed approach achieves recognition accuracy of 97.5% in real-time. In the future, we are aiming to exploit the same technique for multi-word Pashto characters, which are difficult to be recognized in real-time.

REFERENCES

[1] G. A. Abandah, K. Younis, and M. Z. Khedher, "Handwritten Arabic character recognition using multiple classifiers based on letter form," in *Proc. 5th Int. Conf. Signal Process., Pattern Recognit., Appl.*, 2008, pp. 128–133.

[2] N. B. Amor and N. E. B. Amara, "Multifont Arabic characters recognition using Hough transform and HMM/ANN classification," *J. Multimedia*, vol. 1, no. 2, pp. 4–50, May 2006.

[3] K. Aparna, V. Subramanian, M. Kasirajan, G. V. Prakash, V. Chakravarthy, and S. Madhvanath, "Online handwriting recognition for Tamil," in *Proc. 9th Int. Workshop Frontiers Handwriting Recognit.*, Oct. 2004, pp. 438–443.

[4] T. Artieres, S. Marukatat, and P. Gallinari, "Online handwritten shape recognition using segmental hidden Markov models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 205–217, Feb. 2007.

[5] F. Biadsy, J. El-Sana, and N. Y. Habash, "Online Arabic handwriting recognition using hidden Markov models," in *Proc. 10th Int. Workshop Frontiers Handwriting Recognit.* Colombia: Suvisoft, 2006. [Online]. Available: https://www.researchgate.net/profile/Jihad-El-Sana/publication/228614493_Online_Arabic_Handwriting_Recognition_Using_Hidden_Markov_Models/links/0deec51685aa0daf27000000/Online-Arabic-Handwriting-Recognition-Using-Hidden-Markov-Models.pdf and https://academiccommons.columbia.edu/doi/10.7916/D8RR26MR/download

[6] A. Cichocki, R. Unbehauen, and E. Rummert, "Robust learning algorithm for blind separation of signals," *Electron. Letters.*, vol. 30, no. 17, pp. 1386–1387, 1994.

[7] S. D. Connell and A. K. Jain, "Template-based online character recognition," *Pattern Recognit.*, vol. 34, no. 1, pp. 1–4, 2001.

[8] S. D. Connell and A. K. Jain, "Writer adaptation for online handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 329–346, Mar. 2002.

[9] U. Garain, B. Chaudhuri, and T. T. Pal, "Online handwritten Indian script recognition: A human motor function based framework," in *Proc. Int. Conf. Pattern Recognit.*, vol. 3, Aug. 2002, pp. 164–167.

[10] S. Izadi and C. Y. Suen, "Online writer-independent character recognition using a novel relational context representation," in *Proc. 7th Int. Conf. Mach. Learn. Appl.*, 2008, pp. 867–870.

[11] S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "Online handwriting recognition: The NPen++ recognizer," *Int. J. Document Anal. Recognit.*, vol. 3, no. 3, pp. 80–169, 2001.

[12] D. Keysers, T. Deselaers, H. A. Rowley, L.-L. Wang, and V. Carbune, "Multi-language online handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1180–1194, Jun. 2017.

[13] D. Kumar and N. Rana, "Speech synthesis system for online handwritten Punjabi word: An implementation of SVM & concatenative TTS," *Int. J. Comput. Appl.*, vol. 97, no. 5, pp. 87–88, 2011.

[14] M. Liwicki and H. Bunke, "Combining on-line and off-line systems for handwriting recognition," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2007, pp. 372–376.

[15] M. Liwicki, A. Graves, S. Fernàndez, H. Bunke, and J. Schmidhuber, "A novel approach to online handwriting recognition based on bidirectional long short-term memory networks," in *Proc. 9th Int. Conf. Document Anal. Recognit.*, 2007, 2007, pp. 1–5.

[16] S. Malik and S. A. Khan, "Urdu online handwriting recognition," in *Proc. IEEE Symp. Emerg. Technol.*, Sep. 2005, pp. 27–31.

[17] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama, "Substroke approach to HMM-based on-line kanji handwriting recognition," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 491–495.

[18] H. Nakkach, S. Hichri, S. Haboubi, and H. Amiri, "Hybrid approach to features extraction for online Arabic character recognition," in *Proc. 13th Int. Conf. Comput. Graph., Imag. Visualizat. (CGiV)*, Mar. 2016, pp. 253–258.

[19] P. Patil and S. Ansari, "Online handwritten devnagari word recognition using HMM based technique," *Int. J. Comput. Appl.*, vol. 95, no. 17, pp. 17–21, Jun. 2014.

[20] M. Pechwitz and V. Maergner, "HMM based approach for handwritten Arabic word recognition using the IFN/ENIT–database," in *Proc. 7th Int. Conf. Document Anal. Recognit.*, 2003, pp. 890–894.

[21] M. Potrus, U. Ngah, and B. Ahmed, "An evolutionary harmony search algorithm with dominant point detection for recognition-based segmentation of online Arabic text recognition," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1129–1139, 2014.

[22] P. JR and U. Kulkarni, "Gujarati character recognition using adaptive neuro fuzzy classifier with fuzzy hedges," *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 5, pp. 763–775, 2015.

[23] J. R. Prasad and U. V. Kulkarni, "Gujrati character recognition using adaptive neuro fuzzy classifier," in *Proc. Int. Conf. Electron. Syst., Signal Process. Comput. Technol.*, Jan. 2014, pp. 402–407.

[24] S. Puri and S. P. Singh, "Text recognition in bilingual machine printed image documents—Challenges and survey: A review on principal and crucial concerns of text extraction in bilingual printed images," in *Proc. 10th Int. Conf. Intell. Syst. Control (ISCO)*, Jan. 2016, pp. 1–8.

[25] M. Razzak, F. Anwar, A. Belaid, and M. Sher, "HMM and fuzzy logic: A hybrid approach for online Urdu script-based languages' character recognition," *Knowl.-Based Syst.*, vol. 23, no. 8, pp. 914–923, 2010.

[26] S. Kc and C. Nattee, "A comprehensive survey on on-line handwriting recognition technology and its real application to the Nepalese natural handwriting," *Kathmandu Univ. J. Sci., Eng. Technol.*, vol. 5, no. 1, pp. 31–55, Jan. 1970.

[27] R. Sarkhel, N. Das, A. Das, M. Kundu, and M. Nasipuri, "A multi-scale deep quad tree based feature extraction method for the recognition of isolated handwritten characters of popular indic scripts," *Pattern Recognit.*, vol. 71, pp. 78–93, Nov. 2017.

[28] S. Sen, A. Bhattacharyya, A. Das, R. Sarkar, and K. Roy, "Design of novel feature vector for recognition of online handwritten Bangla basic characters," in *Proc. 1st Int. Conf. Intell. Comput. Commun.*, 2017, pp. 485–494.

[29] N. Shahzad, B. Paulson, and T. Hammond, "Urdu Qaeda: Recognition system for isolated Urdu characters," in *Proc. IUI Workshop Sketch Recognit.*, Sanibel Island, FL, USA, 2009, pp. 1–10.

[30] J. Sternby, J. Morwing, J. Andersson, and C. Friberg, "On-line Arabic handwriting recognition with templates," *Pattern Recognit.*, vol. 42, no. 12, pp. 3278–3286, 2009.

[31] H. Swethalakshmi, A. Jayaraman, V. Chakravarthy, and C. Sekhar, "Online handwritten character recognition of Devanagari and Telugu Characters using support vector machines," in *Proc. 10th Int. Workshop Frontiers Handwriting Recognit.*, Oct. 2006, pp. 1—7.

[32] H. Tanaka, K. Nakajima, K. Ishigaki, K. Akiyama, and M. Nakagawa, "Hybrid pen-input character recognition system based on integration of online-offline recognition," in *Proc. 5th Int. Conf. Document Anal. Recognit.*, 1999, pp. 209–212.

[33] A. H. Toselli, M. Pastor, and E. Vidal, "On-line handwriting recognition system for Tamil handwritten characters," *Proc. InIberian Conf. Pattern Recognit. Image Anal.*, 2007, pp. 370–377.

[34] K. Verma and R. K. Sharma, "Comparison of HMM and SVM-based stroke classifiers for gurmukhi script," *Neural Comput. Appl.*, vol. 28, no. S1, pp. 51–63, Dec. 2017.

[35] X. Xiao, L. Jin, Y. Yang, W. Yang, J. Sun, and T. Chang, "Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition," *Pattern Recognit.*, vol. 72, pp. 72–81, Dec. 2017.

[36] S. Xu, M. Li, R.-R. Zheng, and S. Michael, "Manchu character segmentation and recognition method," *J. Discrete Math. Sci. Cryptogr.*, vol. 20, no. 1, pp. 43–53, Jan. 2017.

[37] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing Chinese characters with recurrent neural network," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 849–862, Jun. 2017.

[38] Y. Hamdi, H. Boubaker, and A. M. Alimi, "Online Arabic handwriting recognition using graphemes segmentation and deep learning recurrent neural networks," in *Enabling Machine Learning Applications in Data Science*. Singapore: Springer, 2021, pp. 281–297.

[39] S. Singh and A. Sharma, "Recognition of online handwritten Gurmukhi characters through neural networks," in *Advances in Communication and Computational Technology*. Singapore: Springer, 2021, pp. 223–234.

[40] R. L. Graham, *Information Processing Letter*. New York, NY, USA: Murray Hill, 1972, pp. 132–133. [Online]. Available: https://ci.nii.ac.jp/naid/30007731417/#cit

MUHAMMAD SHABIR received the M.S.C.S. degree in computer science from Islamia College University, Peshawar, Pakistan, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. His research interests include artificial intelligence, machine learning, computer vision, and natural language processing.

NAVEED ISLAM received the Ph.D. degree in computer science from the University of Montpellier II, France, in 2011. He is currently an Assistant Professor with the Department of Computer Science, Islamia College University, Peshawar, Pakistan. He is the author of numerous international journal articles and conference papers. His research interests include artificial intelligence, machine learning, computer vision, and data security. He is a Regular Reviewer of IEEE, Elsevier, and Springer journals.

ZAHOOR JAN received the M.S. and Ph.D. degrees from FAST University, Islamabad, in 2007 and 2011, respectively. He is currently an Associate Professor of computer science with Islamia College Peshawar, Pakistan, where he is also the Chairperson of the Department of Computer Science. His current research interests include image processing, machine learning, computer vision, artificial intelligence, and medical image processing, biologically inspired ideas, such as genetic algorithms and artificial neural networks, and their soft-computing applications, biometrics, solving image/video restoration problems using the combination of classifiers using genetic programming, and optimization of shaping functions in digital watermarking and image fusion.

INAYAT KHAN received the Ph.D. degree in computer science from the Department of Computer Science, University of Peshawar, Pakistan. His current research is based on the design and development of context-aware adaptive user interfaces for minimizing drivers' distractions. His research interests include lifelogging, ubiquitous computing, accessibility, and mobile-based assistive systems for people with special needs, and published several articles in international journals and conferences in these areas.

TAJ RAHMAN received the B.S. degree in computer science from the University of Malakand Dir (Lower), Pakistan, in 2007, the M.S. degree in computer science from Agriculture University, Peshawar, Pakistan, in 2011, and the Ph.D. degree in computer science from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. He is currently working as an Assistant Professor with the Department of Physical and Numerical Sciences, Qurtuba University of Science and Technology, Peshawar. His research interests include wireless sensor networks and the Internet of Things.

ASIM ZEB received the B.Sc. and M.Sc. degrees in computer science from the University of Peshawar (UOP), Pakistan, in 2002 and 2005, respectively, and the Ph.D. degree in computer science from University Technology Malaysia, in 2016. From 2014 to 2015, he worked as a Research Fellow with the Nagoya Institute of Technology, Japan. He has received the MJIIT-Malaysia Scholarship (2013–2014) and JASSO-Japan Scholarship (2014–2015). From February 2016 to April 2019, he worked as an Assistant Professor with the Qurtuba University of Science and IT. Since May 2019, he has been working as an Assistant Professor with the Department of Computer Science, Abbottabad University of Science and Technology, Pakistan. His research interests include self-organized networks, networks architectures and protocols, and supervised machine learning techniques.

SHAFIQ AHMAD received the Ph.D. degree from RMIT University, Melbourne, VIC, Australia. He is currently working as an Associate Professor at the Industrial Engineering Department, College of Engineering, King Saud University, Riyadh, Saudi Arabia. He has more than two decades working experience both in industry and academia in Australia, Europe, and Asia. He has published a research book and several research articles in international journals and refereed conferences. His research interests include smart manufacturing, the IIoT and data analytics, multivariate statistical quality control; process monitoring and performance analysis; and operations research models and bibliometric network analysis. He is also a Certified Practitioner in Six Sigma Business Improvement Model.

ABDELATY EDREES ABDELGAWAD received the B.Sc. degree in production engineering from Beni Suef University, Beni Suef, Egypt. He is currently a Researcher with the Industrial Engineering Department, King Saud University. He has collaborated on various industrial and research projects. He has experience in teaching wide range of IE courses, is involved with several administrative duties. His research interests include Industry 4.0, design and manufacture using reverse engineering, various manufacturing methods, study and analysis of human factors, sigma 6 applications in industrial facilities, and quality, reliability, and six sigma.

MALI ABDOLLAHIAN has designed, developed, and implemented the two-year Master of Analytics at RMIT, in 2012. She is currently a Program Manager-Master of Analytics and Master of Statistics and Operations Research. She is an internationally recognized specialist in the areas of applied statistics, statistical quality control and performance analysis in advanced manufacturing, service industry and public health. She has been researching, teaching, consulting, and presenting on these topics for over 25 years. She has coauthored over 100 refereed articles and conducted over 30 conferences/seminar presentations. The refereed publications in the area of research are listed on Google Scholar. She is a member of the Advisory Committee of the North and West Melbourne Data Analytics Hub. She is an Associate Editor of *IEEE Systems, Man and Cybernetics Magazine* (SMC).