IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 18, 2018, accepted August 6, 2018, date of publication August 16, 2018, date of current version September 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2865532

Urdu Optical Character Recognition Systems: Present Contributions and Future Directions

NAILA HABIB KHAN AND AWAIS ADNAN
Department of Computer Science, Institute of Management Sciences, Peshawar 25000, Pakistan
Corresponding author: Naila Habib Khan (naila.khancs@yahoo.com)

ABSTRACT This paper gives an across-the-board comprehensive review and survey of the most prominent studies in the field of Urdu optical character recognition (OCR). This paper introduces the OCR technology and presents a historical review of the OCR systems, providing comparisons between the English, Arabic, and Urdu systems. Detailed background and literature have also been provided for Urdu script, discussing the script's past, OCR categories, and phases. This paper further reports all state-of-the-art studies for different phases, namely, image acquisition, pre-processing, segmentation, feature extraction, classification/recognition, and post-processing for an Urdu OCR system. In the segmentation section, the analytical and holistic approaches for Urdu text have been emphasized. In the feature extraction section, a comparison has been provided between the feature learning and feature engineering approaches. Deep learning and traditional machine learning approaches have been discussed. The Urdu numeral recognition systems have also been deliberated concisely. The research paper concludes by identifying some open problems and suggesting some future directions.

INDEX TERMS Cursive, optical character recognition, Urdu text recognition.

I. INTRODUCTION

The OCR (Optical Character Recognition) technology has a rich history, standard procedures and types, it also presents numerous benefits as well as challenges [1]. The recognition for Urdu text has been approached recently as compared to the script recognition systems for other cursive and non-cursive scripts [2]. The Urdu language has worldwide appeal, it is written and understood in numerous countries around the world, however, little to no progress or achievements have been reported for recognition of its script. This huge lag of research is mainly due to the inefficiencies in different fields, such as dictionaries, research funding's, equipment's, benchmark datasets and other necessary utilities.

Our goal with this review paper is threefold: (1) For computer vision, pattern recognition and artificial intelligence readers, we expect this paper to serve as an introduction to the OCR technology and its related concepts and believe that our work will be a substantial contribution to all of these three domains. (2) For Urdu OCR researchers, we believe this review paper will provide an in-depth study of all the previous, existing and future technologies, methods and algorithms that can be used in different scenarios for recognition of Urdu text. (3) For general readers, we hope this review to contribute and extend the significance of optical character recognition technologies. We also hope it will provide a means of identifying knowledge gaps and the possibilities of carrying out future research and development in this field.

The paper is structured as follows: Section 2 gives an in-depth insight into the general history of OCR systems, comparing various scripts, such as Latin, Arabic and Urdu. Then, in Section 3, Urdu script and its history are discussed. Next, Section 4 discusses some of the prominent Urdu datasets that are frequently used by the research community. In Section 5, a comprehensive review is given for the existing studies using different modes and categories of Urdu optical character recognition. Following, in Section 6 notable contributions in the field of Urdu OCR are mentioned covering the different phases such as image acquisition, pre-processing, image segmentation, feature extraction, classification and post-processing. There haven't been any remarkable procedures or results reported by the research community for post-processing phase, hence, no related work has been being provided. Section 7 briefly deliberate the Urdu numeral recognition systems. In Section 8, the research paper concludes by summarizing the entire research article, identifying some open problems and suggesting some future directions in the field of Urdu OCR.

II. HISTORY OF OCR

The early optical character recognition ideas date back to the technologies that were developed to help the visually

impaired people. Two famous devices the Tauschek's reading machine and Fournier Optophone were developed during 1870 to 1931 to help the blind read [3]. In the 1950s, the invention of Gismo, a machine that was capable of translating printed text messages into machine codes was used for computer processing. These devices were also capable of reading text aloud. The device was developed by the efforts of David. H. Shepard, a cryptanalyst and Harvey Cook. Intelligent Machines Research Corporation was the first company to sell out these OCR devices. Following the success, the world's first OCR system was developed by David H. Shepard. Standard Oil Company of California used the OCR system for making credit card imprint. Other consumers for the OCR system included the Readers Digest and the Telephone Company. During the era of 1954 and 1974, first, portable OCR devices hit the market such as Optacon. These devices were used to scan and digitize postal addresses. Initially, the postal number recognition was very weak but with the advancement of technology it succeeded. The OCR technology progressed immensely by developing passport scanners and price tag scanners during the 1980's. In late years of 1980's and early years 1990's, some of the most famous companies today in the field of OCR were developed, such as Caere Corporation, Kurzweil Computer Products Inc and ABBYY. During the period 2000 to 2017, the OCR technology has developed immensely. Technologies have been introduced that allows online services through the web OCR, as well as certain applications enabling real-time translation of foreign languages on smartphones are developed. Tesseract a famous OCR engine was also published by Hewlett Packard and the University of Nevada, Las Vegas. Different OCR software's have also been made available online for free by Adobe and Google Drive. Over the past decade's most of OCR research has been directed toward non-cursive scripts such as Latin. The research for cursive scripts such as Arabic and Urdu started decades later than that for Latin script (see Figure 1).

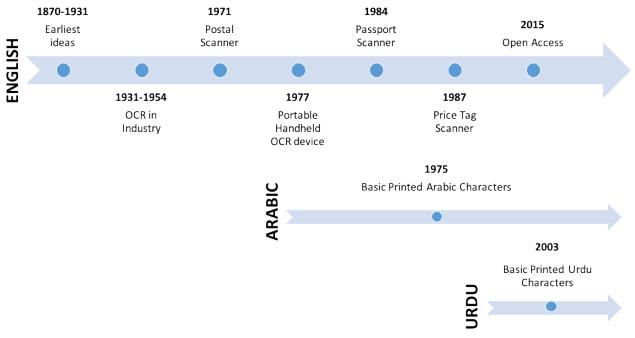


FIGURE 1. History of OCR.

Some of the earliest research towards Arabic OCR can be dated back to 1970, where an OCR was patented to read the basic printed Arabic numerals from a sheet [4]. Nowadays, the technologies and research for Arabic OCR have progressed to more intelligent character recognition systems that support handwritten and cursive scripts [5]. Currently, much commercial software's such as ABBYY provides support for Arabic script. However, the accuracy rates in comparison to the Latin text are low.

Similarly, OCR for Urdu scripts is far behind than that of Latin script as well as Arabic script. The early systems for Urdu OCR can be traced back to 2003, where a system was developed to recognize the basic isolated printed Urdu characters [6]. Over the past decade, research interest towards the Urdu OCR has increased immensely. However, due to the cursive and context-sensitive nature of its script, it's still lagging behind in printed OCR and very few developments have been reported for handwritten OCR. Some of the most famous software's such as OmniPage, Adobe Acrobat, ABBYY FineReader, Readiris, Power PDF Advanced, Soda PDF and other commercial OCR's have none or extremely little support for Urdu script. Most of these software's are multilanguage, but they provide higher accuracies for English, some of the Asian languages such as Urdu are mostly not well supported because their fonts are missing.

III. URDU SCRIPT, ITS HISTORY AND RELATION TO OTHER CURSIVE SCRIPTS

The Arabic alphabet has influenced several languages including Persian, Urdu and Pashto [7], [8]. Each of the mentioned languages has some dissimilarity in the characters, but they share the same underlying foundation. Urdu has similarities to Arabic alphabet, due to the history it shares with it [9].

Urdu is basically derived from a Turkish word "Ordu" meaning "army" or "camp." The history of Urdu language is vibrant and vivid. It is believed that Urdu can into existence during the Mughal Empire. After the 11th century, the Persian and Turkish invasions of the subcontinent cause the development of Urdu as a source of communication. During the Mughal Empire, Persian was the official language, whereas, Arabic was the language of religion, Turkish was spoken mostly by the high profile or the Sultans. Therefore, Urdu was highly under influence of these three languages. During the early years, it was just used for communication and was known as "Hindvi." As years progressed its vocabulary expanded and several names were associated with it during this period, like Dehalvi and Zaban-e-Urdu. After independence Urdu was declared the national language of Pakistan.

Arabic and Urdu both are written in Perso-Arabic script; therefore, they share similarities at the written level. Arabic and Persian writing styles have great influence on Urdu script. Hence, Urdu uses a modified and extended set of Arabic alphabets and Persian alphabets. Urdu uses Nastalique calligraphic style of the Perso-Arabic script for writing. The history of Nastalique dates back to the Islamic conquest of Persia. The Persian art of calligraphy was adopted by the Iranians. Mir-Ali Heravi Tabrizi famous Iranian calligrapher developed the Nastalique calligraphic style during the 14th century. Nastalique was formed by the combination of two scripts "Nash" and "Taliq." In the early years, it was

called ''Nashtaliq'' but later on, it was more formally known as Nastalique. In South Asia, Persian was the official language of the Mughal Empire. Nastalique emerged during these days and left a great influence on South Asia including Bangladesh, India and Pakistan. In Bangladesh, Nastalique was greatly used before 1971. In India, Nastalique is still observed widely. Nastalique is found to be the standard calligraphic style for writing in Pakistan. Nastalique is extremely beautiful and more artistic as compared to the Naskh writing style of Arabic. There are several calligraphic styles for writing Arabic script such as Naskh, Nastalique, Koufi, Thuluthi, Diwani and Rouq'i style (see Figure 2). Naskh is the most common writing style that is used for Arabic, Persian as well as Pashto script [2].

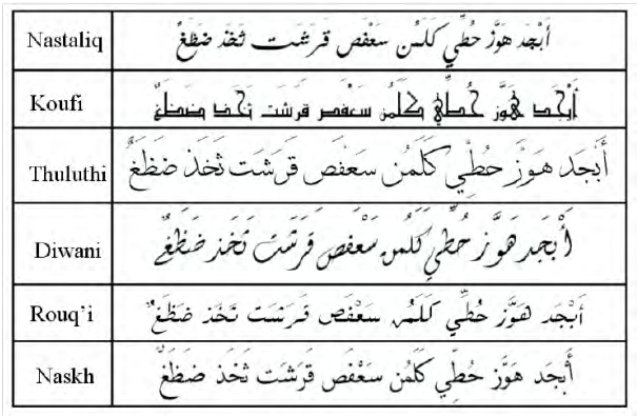


FIGURE 2. Different writing styles for Arabic script [10].

TABLE 1. Comparison between Arabic, Urdu, Persian and Pashto writing styles.

Characteristic	Urdu	Arabic	Persian	Pashto
Total No of letters	38	28	32	45
Order of Writing	Right to left	Right to left	Right to left	Right to left
Cursive	Yes	Yes	Yes	Yes
Dots and Diacritics	Yes	Yes	Yes	Yes

Arabic, Persian, Urdu and Pashto, all four alphabet systems are more or less the same, the only difference is the total number of characters (see Table 1). Arabic has the smallest number of characters in its alphabet. Persian uses the Arabic characters along with more number of characters. Urdu and Pashto both extend further from the Persian alphabet.

The Arabic alphabet is also known as an abjad. It is written from right to left and has a total of 28 characters [2], as given in Table 2. Arabic alphabet does not possess any distinct upper-case and lower-case forms. There are several characters that may have a similar appearance but they are given their own distinction by the use of dots that are placed above or below their central part. For example, the Arabic letters خ(khā'), ح (hā'), ج(jīm) have the same base shape, however, they have one dot below, no dot and one dot above.

The Persian alphabet and script share many similarities to that of the Arabic script. It is also written right-to-left and is an abjad, meaning the vowels are under-represented in the writing system. The Persian alphabet consisting of 32 characters is given in Table 2. The Persian script is cursive in nature, hence, the characters change their shape depending on its position: isolated, initial, middle and final of a word.

Pashto is the official language of Afghanistan and is also widely spoken in the Khyber Pakhtunkhwa province of Pakistan. It's used by 50 million people as a source for oral and written communication [11]. There is a total of 45 characters in Pashto alphabet (see Table 2). The characters in the alphabet may have 0 to 4 diacritic marks. There have been no significant efforts devoted to the recognition of the Pashto script.

There is a total of 38 characters in Urdu alphabet [12]. In Urdu, the text lines are read from top to bottom, whereas, the characters are read from right to left. The characters can be grouped into similar classes based on the similarities of their base forms; the characters in the same class differ only by their dots or retroflex mark. In Table 2, the character shape for basic isolated Urdu characters has been given.

A. JOINER AND NON-JOINER CHARACTERS

There are two types of characters in Urdu; joiners and non-joiners [13]. Joiner characters are written cursively. The shape of character changes depending on its neighboring character to which its connected, as well as its position within the word. Hence, all connectors in principle have four basic shape forms i.e. the ''isolated,'' ''start,'' ''middle'' and ''end.'' There are 27 joiner characters in the Urdu alphabet as shown in Figure 3.

ش	س	خ	ح	چ	ج	ث	ٹ	ت	پ	ب
11	10	9	8	7	6	5	4	3	2	1
ل	گ	ک	ق	ف	غ	ع	ظ	ط	ض	ص
22	21	20	19	18	17	16	15	14	13	12
						ی	ھ	ہ	ن	م
						27	26	25	24	23

FIGURE 3. Joiner Urdu characters.

Alternatively, non-joiner characters are those characters that have no special start or middle forms, because they don't connect to other characters. Hence, the non-joiner characters only take two basic shape forms i.e. the ''isolated'' and ''end.'' There is a total of 10 characters in Urdu alphabet that are non-joiner as shown in Figure 4.

If a word ends with a joiner character then space must be inserted after the word, else it will result in merging the current word to the following word, resulting in a visually incorrect and meaningless word. For example, two words ''چیف جسٹس'' without space will become ''چیفجسٹس,'' making it incorrect. Another example considers two words

TABLE 2. Alphabets for Pashto, Urdu, Persian and Arabic languages.

No.	Pashto	Urdu	Persian	Arabic
1	ا	ا	ا	ا
2	ب	ب	ب	ب
3	پ	پ	پ	
4	ت	ت	ت	ت
5	ټ	ٹ		
6	ث	ٹ	ث	ث
7	ج	ج	ج	ج
8	چ	چ	چ	
9	ح	ح	ح	ح
10	خ	خ	خ	خ
11	څ			
12	ځ			
13	د	د	د	د
14	ډ	ڈ		
15	ذ	ذ	ذ	ذ
16	ر	ر	ر	ر
17	ړ	ڑ		
18	ز	ز	ز	ز
19	ژ	ژ	ژ	
20	ږ			
21	س	س	س	س
22	ش	ش	ش	ش
23	ښ			
24	ص	ص	ص	ص
25	ض	ض	ض	ض
26	ط	ط	ط	ط
27	ظ	ظ	ظ	ظ
28	ع	ع	ع	ع
29	غ	غ	غ	غ
30	ف	ف	ف	ف
31	ق	ق	ق	ق
32	ک	ک	ک	ك
33	ګ	گ	گ	
34	ل	ل	ل	ل
35	م	م	م	م
36	ن	ن	ن	ن
37	ڼ			
38	و	و	و	و
39	ه	ہ		ه
40		ھ	ھ	
41	ۀ	ء		
42	ي			ي
43	ې			
44	ی	ی	ی	
45	ۍ			
46	ئ			
47		ے		

"محنت بھی," without space separation between the joiner words will become "محنتبھی," completely making it visually meaningless. However, words ending with non-joiner

FIGURE 4. Non-joiner Urdu characters.

usually don't have space from its next word, and therefore are ligatures within the same word. For example, the text "نظام" seems like two words but there is no space between them. However, these are actually two ligatures "م" and "نظا" belonging to the same word. Segmentation of ligatures is extremely complex since there is no separation or space between the ligatures.

B. DOTS AND DIACRITICS

Urdu characters are surrounded by a special type of marks known as diacritics. The diacritics surrounds the characters main body and lie above or below it [14]. There are three types of diacritics i.e. Nuqta (Dot), Aerab and 'ط' superscript. The dots(s) placement and the number is used to distinguish several characters in the Urdu alphabet. The dots(s) can be placed below or above the associated character. The dots(s) can range from one to maximum three in number. Total 17 characters in the Urdu alphabet are accompanied by the dots(s) as shown in Figure 5.

FIGURE 5. Characters associated with dots.

The consonants are represented by characters. Diacritics which serve as vowel marks are also sometimes known as aerab. Aerab helps in the pronunciation of the Urdu characters. Aerab are optional and written with the Urdu script when there is need to remove any confusion in the pronunciation. The aerab helps in changing the sound of the letter. Some of the common aerab are shown in Figure 6.

FIGURE 6. Some of the common Aerab used with Urdu characters.

There are three characters in Urdu that are known as retroflex consonants, see Figure 7. A retroflex consonant is spoken when the tongue has a curled, flat or concave shape. These were not present in the Persian or Arabic alphabet. The retroflex consonants are created by placing the ط superscript on three Urdu characters. These Urdu characters are known

FIGURE 7. Retroflex consonants and ط superscript.

as dental consonants. A dental consonant is spoken when the tongue is pressed against the upper teeth.

IV. DATASETS FOR URDU OCR

There has been relatively little research in Urdu OCR because of the paucity of corpora and image datasets. A benchmark dataset is an essential part for the efficient and robust development of a character recognition system. However, there are not enough datasets available to do the basic research and development tasks in Urdu NLP applications such as Urdu OCR [15]. The commonly used datasets for various Urdu NLP tasks have been mentioned below and shown in Figure 8.

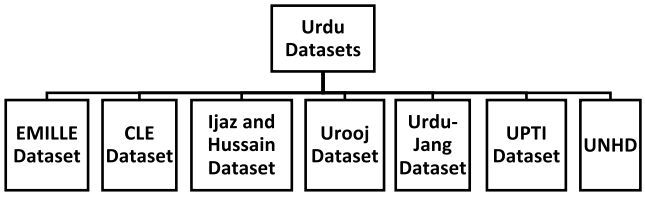


FIGURE 8. Datasets for Urdu optical character recognition.

EMILLE (Enabling Minority Language Engineering) project [16] that was initiated by Lancaster University in 2003, is the first ever initiative for making Urdu corpus available. The main objective of the project was to develop a dataset for South Asian languages. Over the years, now the dataset has been extended to include more than 96 million words. It encompasses three types of data i.e. monolingual, parallel and annotated data. The dataset consists of a total of 512,000 spoken Urdu words and 1,640,000 words of Urdu text. Besides Urdu, the EMILLE project also includes thirteen other South Asian languages.

Corpora and associated tools for Urdu text processing have also been developed by the Centre for Language Engineering (CLE) in Pakistan. It has been conducting research and development for computational aspects of Pakistan's various languages. The main aim of the center is to enable the public to access information and communicate in their local languages using information and communication technology. They have provided 19.3 million ligature corpus that has been collected from a wide range of domain, namely, sports/games, news, finance, culture/entertainment, consumer information and personal communication. CLE also provided A high-frequency ligature corpus [17].

Another small dataset for Urdu was given by Shafait *et al.* [18]. The dataset is publicly available and consists of only 25 documents and had Ground-truth for both OCR and layout analysis.

A huge word dictionary was developed by Ijaz and Hussain [19], in 2007. A total of 50,000 distinct words were collected from documents of different domains, namely, finance, sports, culture, entertainment, news, personal communication and consumer information.

A dataset was provided to the public by Urooj *et al.* [20]. It consists of Urdu Nastalique font of various font sizes. There are more than 100, 000 words in the developed dataset. The data for the dataset has been collected from various domains, namely, interviews, press, novels, letters, translations, religion, short stories, sports, science, culture, health care and book reviews.

A synthetic dataset named 'Urdu-Jang Dataset', containing a total of 26,925 UTF encoded text lines was generated by Ahmed, et al. [12]. The synthetic dataset was generated from renowned Urdu newspaper named Jang, that is printed in Alvi Nastalique script. The dataset covers various social, political and religious issues.

A synthetic dataset was proposed by Sabbour and Shafait [14], called UPTI (Urdu Printed Text Image Database) dataset, it consists of 10,063 synthetically generated text lines and ligature images. It was developed as an analogy to the dataset, APTI (Arabic Printed Text Image), proposed by Slimane *et al.* [21]. The size of the dataset was increased by applying different degradation procedures. The UPTI dataset consists of both of text-line version and ligature version to measure the accuracy of the recognition system. It consists of 12 sets of by varying four parameters for the images, namely jitter, elastic elongation, threshold and sensitivity. The synthetic dataset comprises different political, social and religious issues, and was collected from the Jang newspaper. The dataset also contains ground-truth information for both text-line version and ligature version.

An offline handwritten dataset for Urdu Nastalique text named 'Urdu Nastalique Handwritten Dataset (UNHD)' was provided by Ahmed, et al. [22]. The generated UNHD dataset covers some of the most commonly used Urdu characters, ligatures and numerals with different variations that were collected from 500 writers (both male and female) on an A4 size paper. The dataset is provided publicly on the website 'https://sites.google.com/site/researchonurdulanguage'. Overall the dataset consists of 312,000 words written by 500 candidates with a total of 10,000 text lines.

V. CATEGORIES OF OCR SYSTEM

Typically, Optical Character Recognition Systems can be divided into different types based on its characteristics i.e. input acquisition mode (online or offline), writing mode (printed or handwritten), character connectivity (isolated or cursive) and lastly font constraints (single font or omni font) [23]. The categorization of OCR system is shown in Figure 9.

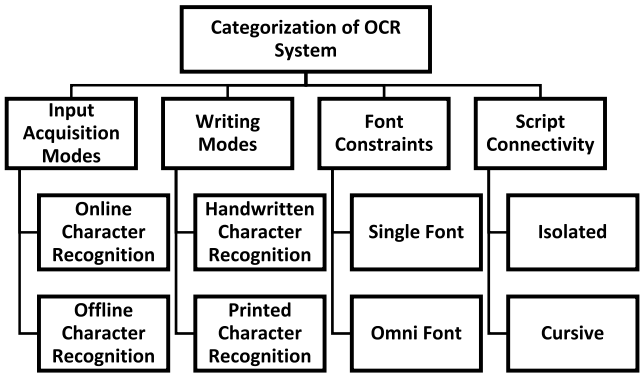


FIGURE 9. Categorization of OCR system.

A. INPUT ACQUISITION MODES

The mode in which input is given to an Optical Character Recognition system can be divided into two types i.e. online recognition and offline recognition [2], [24], [25].

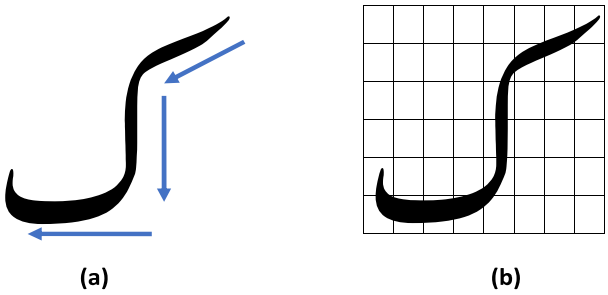


FIGURE 10. (a) Online character recognition, (b) Offline character recognition.

Online recognition deals with real-time recognition of characters, characters are recognized as the movements of the pen are received when writing something (see Figure 10 (a)). Online recognition requires specialized hardware such as pen and tablet to obtain the text input [9], [26]. A concept of digital ink is used, in which a sensor is used to analyze pen tip movements like pen up/down. In comparison to offline recognition, its less complex since the temporal information such as writing order, pen lifts, velocity, speed are readily available. Online OCR systems for Latin font are available in PDA's, Handheld PC's, and also available in some latest touchscreen mobile phones.

Offline recognition deals with recognition of text that has already been converted into a digital image (see Figure 10 (b)). The input image is usually a product of scanning through some digital device such as a scanner or a digital camera [2], [24], [25], [27]. Offline character recognition is also sometimes referred to as static recognition. Offline character recognition is a complex process in comparison to the online recognition, since, in offline, the characters first needs to be located. Offline recognition can be associated with both handwritten and printed scripts. While the online recognition can only be associated with the handwritten text [24].

B. WRITING MODES

A major categorization for OCR system is based on the mode of text the OCR system will be handling. The form of text i.e. printed or handwritten is known as the mode of text when developing an optical character recognition system [23]. The resources available for OCR can be in different formats such as typewritten text containing tables, headers, footers, borders, page numbers etc. or handwritten format having variations of writing styles for different people. Text recognition may seem a minor task for human beings, however, for computational machines, it tends to be extremely challenging for both handwritten and printed script. It poses to be challenging for printed text due to the availability of a large number of fonts, while, for handwritten text, there are numerous variations possible.

An optical character recognition that deals with printed text is sometimes known as printed character recognition system. In case of printed text, its uses different font styles such as for the English language, Times New Roman, Arial, Calibri, Courier etc. are used. Similarly, for Urdu, the most famous style of writing is the Nastalique. Printed character recognition systems are simpler as compared to the handwritten character recognition systems. However, printed text may pose to be complex for recognition based upon the quality of font, document, and writing rules of the language under consideration. Printed text can only be offline [28].

An OCR system that deals with handwritten text is sometimes known as handwritten character recognition system. Handwritten Character Recognition is extremely challenging research area in the field of image processing and pattern recognition. Recognition of handwritten text is exceptionally difficult as compared to printed/typewritten text. Handwritten text possesses a lot of variations not only due to the different writing styles of different people but also due to the varying pen movements of the same writer. Even with the latest recognition methods and systems, the recognition of handwritten text still remains an extremely challenging task even for Latin script. Similarly, for the recognition of Urdu handwritten text, there is still room for a lot of improvement and progress. Few researchers have focused on using handwritten text for Urdu optical character recognition. Handwritten character recognition systems can be further divided into two types, i.e., offline handwritten character recognition and online handwritten character recognition [28].

C. FONT CONSTRAINTS

The geometric features of the characters written in one font style may vary to a great extent from character written in another font. Therefore, the OCR process is highly dependent on font style. An OCR system that has been developed for one font style may completely fail to process or may partially succeed to recognize same text written in another font style. If an OCR system is capable of processing only a single font

style it is known as a single font recognition system. Systems capable of processing and recognizing multiple fonts are called Omni-font character recognition systems [29]. Usually, generalized algorithms are used with Omni-font systems. To use a different font style, only the training process is required to be performed out again. Most of the OCR systems available for Urdu only uses the Nastalique calligraphic font.

Nastalique is basically a fusion of Naskh and Taliq writing styles. Mirza Ahmed Jameel, in 1980, computerized 20,000 Nastalique ligatures for the first time, ready to be used in computers. The font was named Noori Nastalique. Over the years many researchers have created their own version of the Nastalique calligraphic style such as Alvi Nastalique, Jameel Noori Nastalique and Faiz Lahori Nastalique. All the Nastalique fonts fulfill the basic characteristics of Nastalique writing style. However, Nastalique is far more complex than the fonts given for Arabic script [13], [30].

D. SCRIPT CONNECTIVITY

Another categorization for an OCR system is based on the use of the isolated or cursive script. Isolated scripts have characters that do not join with each other when they are written, contrarily, in cursive, the neighboring characters in words may join each other and may also affect the character to change its shape based on the characteristics and position of the character within the word. An optical character recognition using the cursive script is sometimes known as intelligent character recognition. If the system operates at word level its known as intelligent word recognition.

Recognition of cursive text is an active area of research [2]. A new level of complexity is introduced when using cursive scripts with optical character recognition systems. This complexity adds an extra level of segmentation in the recognition process in order to isolate the characters within each word. Due to this added complexity, some of the languages are introducing segmentation free approaches. Segmentation free approach is more commonly known as the holistic approach and attempts to recognize the whole word or sub-word (ligature) without breaking it into subsequent characters [14], [31]. Currently, the OCR systems for cursive scripts are suffering due to segmentation complexities. To achieve higher recognition for general cursive scripts like Urdu, the use of contextual and grammatical information is required. For example, recognizing whole words or sub-words (ligatures) from a dictionary is easier than segmenting individual characters from the text. Words, ligatures and isolated characters for Urdu script are shown in Figure 11.

Ligatures	Isolated Characters	Word
ا + نٹر + نیشنل	ا + ن + ٹ + ر + ن + ی + ش + ن + ل	انٹرنیشنل
آ + ر + ٹیکل	آ + ر + ٹ + ی + ک + ل	آرٹیکل

FIGURE 11. Examples of Urdu isolated characters, ligatures and words.

E. RELATED WORK FOR DIFFERENT CATEGORIES OF URDU OCR

As mentioned earlier, the modes for an OCR system can be divided into different categories based on input acquisition that can be online or offline, writing mode that can be handwritten or printed, font constraints that may include the font variations handled by a recognition system and finally the script connectivity that may include the segmentation process for character or ligature or may completely avoid the segmentation when developing a recognition system.

Compared to online, most of the research has been conducted towards offline isolated and cursive analytical segmentation based character recognition systems. Several authors opted to use offline recognition for development of Urdu OCR systems for isolated characters [32]–[38]. Shamsher *et al.* [32] developed a system for printed Urdu script used for recognizing individual Urdu characters using their proposed algorithm. Similarly, a font size independent technique was proposed for Noori Nastalique font of Urdu, the technique was tested on single Urdu character ligatures [33]. The system proposed by Nawaz *et al.* [34], also aimed for recognition of isolated Urdu characters. The overall system provided the basic pre-recognition techniques such as image pre-processing, segmentation (line and character), and finally the creation of XML files for the training purpose. Similarly, [37] recommended an offline recognition system but for both handwritten and printed Urdu text, and also highlighted the pre-processing, features extraction and classification of Urdu language text. An OCR named ''soft converter,'' was presented by [35], that could recognize the isolated characters of Urdu language utilizing a database. Khan *et al.* [36] developed two databases of images, namely, a 'TrainDatabase' and a 'TestDatabase' for offline recognition of Urdu text. Likewise, [38] also proposed the representation and recognition for offline printed isolated Urdu characters.

Cursive texts are extremely complex to deal with. However, several authors also focused towards developing cursive analytical Urdu text OCR systems [6], [30], [39]–[44]. Ahmad, et al. [39] developed an OCR system that comprised of two main components, i.e. segmentation and classification. Both, segmentation and classification of the compound characters were studied. In another study, Ahmad *et al.* [40] discussed different characteristics of Urdu script and presented a novel and robust method to recognize printed Urdu script without a lexicon. Likewise, [42] proposed an offline recognition system for Naskh script, it operated on segmented characters and classified it into 33 groups for recognition purposes. In a study done by [30], the differences between Naskh and Nastalique fonts were discussed, an analytical segmentation-based model was proposed for pre-processing and recognition of Urdu script. Pal and Sarkar [6] proposed an OCR system for printed Urdu script. Characters were segmented and recognized on basis of multiple features and achieved promising accuracy. Similarly, [41] and [43]

trained an OCR system for on offline printed segmented characters, however, the font utilized has not been mentioned. Relatedly, Iqbal, et al. [44], proposed a system that dealt with recognition of Urdu Nastalique font for character based segmentation of printed Urdu names. The recognized characters were converted into Roman Urdu, handling various complexities during the conversion. Likewise, Khan and Nagar [45] proposed a study for printed and handwritten cursive Urdu text written in Nastalique and Naskh font style. Ul-Hasan *et al.* [9] used deep learning for recognition of printed Urdu text in the Nastalique script. Whereas, Patel and Thakkar [46] also used a deep learning approach but for recognition of cursive handwritten documents, both in Arabic and English scripts. The studies in [47] and [48] proposed an implicit segmentation based recognition system for Urdu text lines in the Nastalique script and a hybrid approach combining the convolution and deep learning for classification of cursive Urdu Nastalique script respectively. In another similar study, Naz *et al.* [49] achieved high accuracy for sentence based implicit recognition of Urdu Nastalique font using human extracted features and deep learning.

Due to the added complexity of cursive script specifically for analytical systems, segmenting text at the character level, several authors are shifting towards recognition of words and ligatures [12], [14], [17], [50]–[60]. Working with ligature based systems seems more feasible since an extra level of character segmentation is omitted that is usually prone to errors. Ahmed *et al.* [12] developed an offline ligature based Urdu recognition system. Another offline OCR system was proposed by Sabbour and Shafait [14] called Nabocr. The system was trained to recognize both the Urdu Nastalique and the Arabic Naskh font. An alternative multi-script OCR system was proposed by Chanda and Pal [50] for word level recognition of multiple scripts i.e. English, Devanagari and Urdu from a single document. The characteristics of different scripts are taken into consideration for development of the final recognition system. Sattar *et al.* [51] presented a novel offline segmentation-free approach for Nastalique based Optical Character Recognition called NOCR. In [61], Khan and Khan developed a technique which was capable of extracting the Urdu font ''Jameel Noori Nastalique'' from images and converted it into editable textual Unicode's. The approach comprised of pre-processing techniques, label connected components, feature extraction, and image comparison. Javed and Hussain [57], focused on the development of a ligature based OCR system that input the main body without the diacritics and recognizes the related ligatures. Similarly, [58] proposed a technique for recognizing font invariant cursive Urdu Nastalique ligatures. Rana and Lehal [59], presented a ligature based OCR system for Urdu Nastalique script, describing its various challenges and using k-NN and SVM classifiers. In [60], a new method was presented for offline recognition of cursive Urdu text written in Noori Nastalique style, the system aimed at ligature based identification. Khattak *et al.* [17] used a segmentation-free and scale-invariant technique for recognition of printed Urdu ligatures in Nastalique font.

A system was presented to recognize printed Nastalique pre-segmented ligatures in [31] and [55]. Similarly, Hussain, et al. [52] analyzed and modified the Tesseract engine to be used for the recognition of offline printed Nastalique calligraphic style of Urdu language. Similarly, Khan and Khan [53], also developed a technique for recognizing Jameel Noori Nastalique font from Urdu newspaper clippings. The clippings were converted into editable textual Unicode's. Mukhtar *et al.* [54], claims to be the first study reported for handwritten Urdu words. The system proposed was an offline OCR system for Urdu. Similarly, two strategies were investigated for improving the classification of Urdu printed ligatures using nearest neighbor by [56].

Online character recognition systems are less complex compared to that of offline recognition systems. Few authors have opted to use online Urdu character recognition systems, using handwritten writing mode by default. Shahzad *et al.* [62] proposed an online Urdu character recognition system capable of recognizing isolated, hand-sketched characters. Similarly, an online segmentation–free character recognition system was suggested by Razzak *et al.* [63]. Khan [64] studied the online recognition of Urdu characters considering only their initial half forms, whereas, Jan *et al.* [65] proposed a system to recognize both the Urdu characters and words. In studies implemented by Husain *et al.* [66] and Sardar and Wahab [67], both online and offline recognition system were combined that could operate independently of fonts and scripts.

The summary of several notable contributions for different categories of Urdu OCR systems is given in Table 3.

VI. OCR PROCESS

A typical offline character recognition system consists of some or all of the six phases, namely the: image acquisition, pre-processing, segmentation, feature extraction, and recognition or classification and post-processing. Figure 12 shows the block diagram of a typical character recognition system. The different phases of an OCR system have been explained in the sections below.

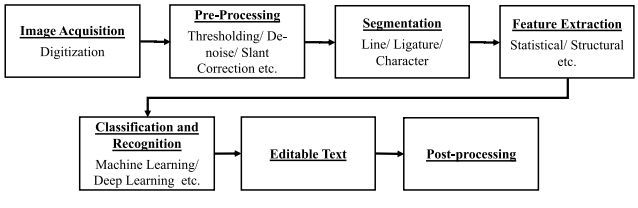


FIGURE 12. Generic optical character recognition process.

A. IMAGE ACQUISITION

Image acquisition is commonly the first stage of any computer vision system. Image acquisition is the process of

TABLE 3. Summary of different categories of Urdu OCR.

Related Work	Input Acquisition			Writing Mode			Font Constraints				Script Connectivity	
	O	Off	OOff	P	H	PH	NQ	NK	NQK	FI	IS	CU
Ahmed, et al. [12]		✓										✓
Sabbour and Shafait [14]		✓		✓					✓			✓
Husain, et al. [66]			✓		✓					✓		✓
Shahzad, et al. [62]	✓				✓						✓	
Razzak, et al. [63]	✓				✓				✓			✓
Khan [64]	✓				✓		✓				✓	
Jan, et al. [65]	✓				✓						✓	
Megherbi, et al. [38]		✓		✓							✓	
Husain [60]		✓		✓								✓
Pal and Sarkar [6]		✓		✓								✓
Chanda and Pal [50]		✓		✓						✓		✓
Javed [43]		✓		✓								✓
Shamsher, et al. [32]		✓		✓							✓	
Ahmad, et al. [40]		✓		✓			✓					✓
Sattar, et al. [51]		✓		✓								✓
Haque and Pathan [41]		✓		✓								✓
Hussain, et al. [42]		✓		✓				✓				✓
Mukhtar, et al. [54]		✓			✓							✓
Akram, et al. [33]		✓		✓			✓				✓	
Nawaz, et al. [34]		✓		✓				✓			✓	
Ahmad, et al. [39]		✓		✓								✓
Javed, et al. [31]		✓		✓			✓					✓
Tariq, et al. [35]		✓		✓							✓	
Khan and Nagar [45]		✓				✓			✓			✓
Khan, et al. [36]		✓									✓	
Hussain, et al. [52]		✓		✓			✓					✓
Ul-Hasan, et al. [9]		✓		✓			✓					✓
Lehal and Rana [55]		✓		✓			✓					✓
El-Korashy and Shafait [56]		✓		✓			✓					✓
Javed and Hussain [57]		✓		✓			✓					✓
Nazir and Javed [58]		✓		✓			✓					✓
Patel and Thakkar [46]		✓		✓			✓					✓
Khan and Khan [61]		✓		✓								✓
Khan and Khan [53]		✓		✓							✓	
Khan, et al. [37]		✓				✓					✓	
Rana and Lehal [59]		✓		✓			✓				✓	
Naz, et al. [49]		✓		✓			✓					✓
Hussain and Ali [30]		✓		✓			✓					✓
Naz, et al. [47]		✓		✓			✓					✓
Naz, et al. [48]		✓		✓			✓					✓
Sardar and Wahab [67]			✓	✓		✓						✓
Khattak, et al. [17]		✓		✓			✓					✓
Iqbal, et al. [44]		✓		✓			✓					✓

O- Online, Off- Offline, OOff-Online and Offline
P- Printed, H-Handwritten, PH-Printed and Handwritten
NQ- Nastalique, NK-Naskh, NQK-Nastalique and Naskh, FI-Font Independent
IS- Isolated, CU-Cursive

acquiring an image into the digital form for manipulation by the digital computers [24], [42]. There are numerous resources for acquiring images into the computer. The text may also be entered into the computer using a tablet and a pen using online recognition input mode. In offline recognition, the source images can be obtained by scanning printed documents, typewritten documents, handwritten documents and by capturing a photograph through an attached camera, digital camera or image scanner. Further for offline recognition, the source images might also be synthetic i.e. generated without the scanning process. The image can be stored in any of the specific formats for example jpeg, bmp, png etc.

Regardless of the source, the quality of the input image plays a vital role in the recognition accuracy [68]. If an image has not been acquired properly then all the later tasks may be affected and the final goal of the recognition system might not be achievable. There are numerous reasons that may affect the overall quality of the input image such as having multiple subsequent copies generated for an original document. Poor printing quality may also make the scanned

document to be noisy. Another major reason that might affect the image quality is the font style and its size. Extremely small fonts are more likely to be considered noise and go unrecognized by the OCR system. Punctuation marks, subscripts and superscripts may also introduce complexities in recognition and may be treated as noise in the image, if its size is extremely small. The recognition quality may also be affected by the quality of paper that was used for printing. Heavyweight and smooth papers are relatively easier to process than lightweight and transparent papers. High quality, smooth and noise-free images are more likely to result in better recognition rates. On the other hand, noise affected images are more prone to errors during the recognition process.

1) RELATED WORK FOR IMAGE ACQUISITION

Image acquisition is the first phase of any recognition system. Over the years, researchers have used numerous sources to input text, either directly by scanning text images, by generating synthetic images or by using a digitizing tablet. Usually, the input method proposed by authors is mostly dependent on the input acquisition mode i.e. offline or online.

Computer generated images i.e. synthetic images have been used by several authors for optical character recognition systems [34], [35], [39], [41], [61]. Different text images, that contained isolated printed Urdu characters was used for training by [34]. Likewise, [35] presented a soft converter that read images in form of a matrix. Haque and Pathan [41] performed a number of experiments on a subset of word images, the font size was kept consistent for all words in the acquired images. In [39], Urdu text line images were given as input and segmentation process was carried out next. Likewise, [61] read images having file extensions tiff, jpeg or png.

Real world images are far more complex to deal with, since the images are scanned. During the scanning process, the images might be affected by noise, skew, slant and other degradations. Several authors opted to develop Urdu recognition systems using the scanned images as input [6], [30], [31], [43], [50], [53], [57], [58]. Hussain and Ali [30] scanned a total of 25 pages from 22 different number of books having varying paper variety, print and page transparencies. Similarly, Pal and Sarkar [6] tested the proposed OCR system on a variety of printed documents that were scanned. Some of the documents had good-quality and were printed on clean paper, while others had inferior printing and paper quality such as children's alphabet books. Javed [43] used scanned images containing Urdu text as input and it was assumed that the images were kept proper during the scanning process to avoid skewness, noise and other distortions in the image. Khan and Khan [53] considered newspaper clippings as input images, that were converted into an editable form to be used on the notepad application. Chanda and Pal [50] digitized the images by an HP scanner at 300 DPI, whereas, Nazir and Javed [58] scanned Urdu document pages at a resolution of 200 dpi. In [58], the pages were set to have fixed size text and had all the letter as well as diacritics were distributed among a variety of context. The document images were composed of single as well as multiple lines. Javed *et al.* [31] extracted three or more samples of each ligature from the text for the analysis purpose. 36 font sizes and Noori Nastalique font was used for printing the pages. After printing these pages were scanned at 150 dpi, and later segmented back into ligatures. Similarly, in another study, Javed and Hussain [57] used printed scanned document images at 36 font sizes.

Some authors used a combination of synthetic and real-world images with the OCR system. Shamsher *et al.* [32] and Ahmad *et al.* [40] collected a corpus of two set of images the computer-generated (synthetic) images and real-world images consisting of scanned Urdu documents. The documents did not contain any other language other than Urdu. Likewise, [33] used manually generated data and data scanned from different books and magazines. Rana and Lehal [59], generated synthetic images of different font sizes i.e. 35, 38, 40, 50 and 55 for the primary components of the ligatures, having being formatted as bold or regular. A total of 1200 primary components were also scanned from books and the rest of primary components were generated synthetically. Khan and Nagar [45] proposed a system that was trained on samples that contained different sized documents having text from different writers. All the input data was resized to 250×250 and contained a single character or a single word. The overall scanning process was expected to be of high quality to avoid any skew correction.

When developing online character recognition systems, the input source is directly shifted towards usage of such equipment that can take input in real time such as a digitizing tablet. Similarly, for Urdu such digitizing tablets have been used for input strokes acquisition [62], [64]–[67]. Shahzad *et al.* [62] extracted hand-sketched Urdu characters drawn on a Tablet PC. Correspondingly, [64] also used a combination of pen-tablet to collect the data. The data signal was stored as a binary file containing the coordinates information and the pressure point values, reducing the complexity of the recognition system. Jan, et al. [65] applied several pre-processing techniques before taking an input of the stroke trajectory to avoid noise due to the input device i.e. pen-tablet and hand movements. Similarly, Husain *et al.* [66] used a digitizing tablet for ligature acquisition. Each of the input strokes represents a ligature in its full form. The ligature is not broken into characters to avoid the errors associated with segmentation. Sardar and Wahab [67] proposed a methodology that works on both online and offline recognition, hence the input could be an image or given through an input device such as digitizing tablet. The different acquisition methods used by researchers is summarized in Table 4.

B. PRE-PROCESSING

Pre-processing involves a series of operations that are carried out on the input image to make it more effective for later stages of the recognition and to improve the overall performance [2], [69]. Pre-processing is used to remove any kind of distortions, quality breakdown, orientation issues that are

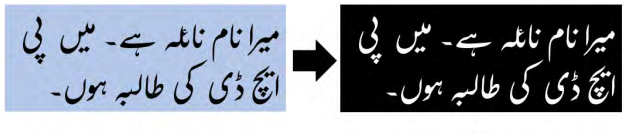
TABLE 4. Summary of different image acquisition sources used for Urdu OCR.

Related Work	Synthetic	Scanned	Digitizing Tablet
Husain, et al. [66]			✓
Shahzad, et al. [62]			✓
Khan [64]			✓
Jan, et al. [65]			✓
Pal and Sarkar [6]		✓	
Chanda and Pal [50]		✓	
Javed [43]		✓	
Shamsher, et al. [32]	✓	✓	
Ahmad, et al. [40]	✓	✓	
Haque and Pathan [41]	✓		
Akram, et al. [33]	✓	✓	
Nawaz, et al. [34]	✓		
Ahmad, et al. [39]	✓		
Javed, et al. [31]		✓	
Tariq, et al. [35]	✓		
Khan and Nagar [45]	✓	✓	
Javed and Hussain [57]		✓	
Nazir and Javed [58]		✓	
Khan and Khan [61]	✓		
Khan and Khan [53]		✓	
Rana and Lehal [59]	✓	✓	
Hussain and Ali [30]		✓	
Sardar and Wahab [67]		✓	✓

introduced during the image acquisition phase. The decline in image quality introduces several problems in the text analysis. Therefore, the pre-processing phase is extremely significant and plays a huge role in the development of a successful recognition system. There are number techniques, such as, image thresholding, noise removal, smoothing, de-skewing, skeletonization, image dilation, normalization etc. that can be used for pre-processing [2]. The selection of the techniques depends on the nature and source of the images. The final outcome of the pre-processing phase is a quality image that is suitable for the segmentation phase. Some of the pre-processing techniques that are frequently used in an OCR system are discussed in the sub-sections below.

1) THRESHOLDING

The process of converting an RGB or Grey image to a bi-level image is known as thresholding [67] (see Figure 13). It is one of the simplest forms of image segmentation, that separates the foreground (actual text) from the background [70]. Thresholding makes the acquired image small, fast and easy to analyze by removing all the unnecessary color information. The acquired image may be in an RGB or indexed format, where each pixel holds certain color information. However, for an OCR system, this color information is not needed and therefore must be removed. If the image is converted into a grey scale image, some color information is removed but still, the image has unnecessary information. The grey scale image is therefore converted into a bi-level image, where each pixel can hold a value of 1 or 0 [43]. The thresholding algorithms can be further divided into two main groups i.e. global thresholding and local adaptive thresholding.



(a) (b)

FIGURE 13. (a) Original image. (b) Thresholded image.

In global thresholding, a single threshold is generated for the entire image [71]. The global thresholding is computed by exploiting the grey level intensity of the image histogram. Images that have non-varying backgrounds are considered more feasible for global thresholding. The implementation of global thresholding is easier than the local adaptive thresholding. On the other hand, local adaptive thresholding is a far more complex but an intelligent technique as compared to the global image thresholding. Instead of selecting a single threshold for the entire image, it classifies every single pixel into the foreground and the background. It works well for images having a varying background. The classification for each pixel is performed by taking into consideration several properties such as the pixel neighborhood [71]. If the pixel in question is darker than its adjacent neighbors, the pixel is converted into black and vice-versa. The results for local adaptive thresholding are far more accurate as compared to the global thresholding algorithm.

2) NOISE REMOVAL

The acquired images are usually distorted with unwanted elements. The external disturbance that leads to the degradation of an image signal is known as noise. There are many sources of noise such as bad photocopying or scanning etc. One of the most popular types of noise is the salt and pepper noise.

3) SMOOTHING

Smoothing is a procedure in which unwanted noise is removed from the edges of the image. Mostly, morphological operations are used for the purpose of smoothing [24]. The morphological operation of dilation and erosion can be used for the purpose of smoothing. Other than erosion, opening and closing can also be applied for smoothing. The opening morphological operation opens small gaps between an object in an image. The closing morphological operation, on the other hand, works by filling all the small gaps between an object's edges in an image.

4) DE-SKEWING

Skewness of the document is when the lines of text become tilted. Skewness can be introduced a result of bad photocopying or scanning. Skewness leads to numerous problems in segmentation. Hence, the de-skewing process is applied to remove the skewness from an image [24].

5) THINNING

Thinning, also known as skeletonization is a process of deleting the dark points along the edges of an object in an image.

Thinning is performed till the object in an image is reduced to a thin line. The final thinned object is 1 pixel wide and henceforth known as the skeleton [27]. Thinning is a very important step of a recognition system and has many advantages. The skeleton of the text can be used to extract features like loops, holes, branch points etc. Thinning also reduces the amount of data to be handled.

6) RELATED WORK FOR PRE-PROCESSING

There are a number of techniques that can be used for pre-processing, such as, image thresholding, noise removal, smoothing, de-skewing, skeletonization, image dilation, normalization etc. Megherbi *et al.* [38] applied filtering, smoothing and thinning algorithms to remove noise from the input images and to reduce the overall variation in the thickness and slanginess of the different Urdu characters. Nawaz *et al.* [34] performed the conversion of a grayscale image into a binary image as well as removing noise i.e. salt and pepper noise from the images. The noise must be removed in order to avoid erroneous system classification and recognition. In [66], smoothing process was used to remove hooks from the data. Usually, the data obtained contained irregularity such as hooks and erratic handwriting. These hooks occurred due to the pen up/down inaccuracies by inexperienced users when using the tablet. 2 to 3-pixel smoothing was applied to remove these issues. Khan *et al.* [36] removed noise from the images of 'TrainDatabase' and 'TestDatabase'. Usually, the salt and pepper or speckle noise were present in the scanned images. Jan *et al.* [65], used different filters to remove unwanted data and to extract the data of interest. The proposed OCR system also used Ramer Douglas Peucker algorithm to reduce the total number of data points. Khan *et al.* [37] used a filtering technique for noise removal. Whereas, Global image thresholding was used for the binarization. The acquired images were also normalized, edges detected, skew was detected and corrected. For normalization, thinning process was applied and correlation approach was used for skew detection and correction.

Skew detection and correction were also used by [6], for skew angle detection Hough transform algorithm was used. The image was rotated according to the detected skew angle. It was observed that the skew estimation method was font style and size invariant. The proposed skew estimation method could handle documents with angles between $+45°$ to $-45°$ and could compute skew angles with a tolerance of $\pm$ 0.5 degrees. Haque and Pathan [41] observed the input bitmap image to analyze skew or slant and remove it. Several other pre-processing processes were applied to the text images such as image binarization, noise removal, blur removal, thinning, skeletonization and edge detection. Likewise, [60] used smoothing, skew detection and correction, document decomposition, slant normalization etc. Tariq, et al. [35] performed skew detection and correction on the acquired image. This operation ensured that the image came into its original position when the user rotated the image. As a pre-processing, the soft converter also performs image binarization operation. In a study by Javed and Hussain [57], document images were considered to be un-skewed and had less to no noise or distortion. First, the main bodies were extracted from the text lines within the image, next the baseline was detected. Once the main bodies were extracted, the skeletonization process was performed using the Jang Chin Algorithm. In Ahmed *et al.* [12], primarily the pre-processing was performed by skew correction, slant correction, de-noising and finally the text normalization.

Chanda and Pal [50] used a histogram based method for thresholding and converting the input images into two tones binary image. The digitized image was further pre-processed to remove noise pixels and irregularities on the boundary of the characters. The images were smoothed out to remove the noise, since it may lead to undesired effects for the OCR system. Similarly, [42] performed image binarization, de-hooking and image smoothing operations. The input image obtained was converted into a binary image appropriate for the feature extraction phase. The input image in RGB format was first converted to a grey level image having pixel values between '0' to '255'. This grey level image was later converted to a binary image containing only two-pixel values i.e. '0' and '1'. Thinning or skeletonization was also applied to the binary image to make it one pixel wide and appropriate for feature extraction. Mukhtar *et al.* [54] and Sardar and Wahab [67] used a thresholding algorithm to perform binarization procedure, [54] also used a moment based algorithm was used for slant normalization whereas [67] used smoothing and noise removal. Iqbal *et al.* [44], the pre-processing was divided into two phases: Binarization and Target Image Detection (TID). The first phase converted the image into binary form, while the second phase removed the unnecessary white pixels that surrounded the image.

Javed [43] proposed skeletonization and another pre-processing technique to detect the baseline for Urdu script that can be used to differentiate between the main bodies and the diacritics. The horizontal projection profile was used for baseline estimation, rows having the maximum number of pixels was set as baseline initially. However, it was observed that this may lead to problems such as the baseline was sometimes set towards the top of the line. To avoid this issue the maximum number of pixels were just computed from the lower half of the line. Also, instead of using a single row, 5 to 10 lines of the row were used as a band of baseline. Likewise, [63] proposed a unique baseline detection rule for handwritten input for Urdu script written in both Naskh and Nastalique writing styles. Different pre-processing steps are required to minimize the unnecessary components in handwritten strokes, and to work with a vast variety of writing styles. Locally Minimum Enclosing Rectangle (MER) was used for baseline detection, using three primary strokes. Ul-Hasan *et al.* [9], each text line image was resized to a fixed height to extract the baseline information. In a study by Nazir and Javed [58], document image binarization was performed as well as baseline detection. The row containing the maximum number of pixels

TABLE 5. Summary of notable contributions for different pre-processing techniques used for OCR.

Study	B	F	T	S	SDC	SN	NR	SK	ED	SZN	Other
Megherbi, et al. [38]		✓		✓				✓			CM*
Husain [60]				✓	✓	✓					DD*
Pal and Sarkar [6]					✓						
Chanda and Pal [50]			✓				✓				
Javed [43]	✓							✓			
Husain, et al. [66]				✓							
Ahmad, et al. [40]											WS*
Haque and Pathan [41]			✓		✓		✓	✓	✓		BM*
Hussain, et al. [42]			✓	✓				✓			D*
Mukhtar, et al. [54]			✓			✓					
Nawaz, et al. [34]			✓				✓				
Razzak, et al. [63]	✓										
Sardar and Wahab [67]	✓		✓	✓			✓				
Tariq, et al. [35]			✓		✓						
Iqbal, et al. [44]			✓								
Khan, et al. [36]							✓				
Ul-Hasan, et al. [9]	✓									✓	
Khan [64]				✓							R*
Javed and Hussain [57]	✓							✓			
Nazir and Javed [58]	✓		✓								
Patel and Thakkar [46]	✓									✓	
Khan, et al. [37]			✓		✓		✓	✓	✓		
Jan, et al. [65]		✓		✓							R*
Ahmed, et al. [12]					✓	✓	✓			✓	

B-Baseline F-Filtering T-Thresholding S-Smoothing
SDC-Skew Detection & Correction SN-Slant Normalization NR-Noise Removal
SK-Skeletonization ED-Edge Detection SZN-Size Normalization
CM*: Component Marking and Framing DD*: Document Decomposition WS*: Word Stretching BM*: Blur Removal/Morphological
D*: De-hooking R*: Re-sampling

was marked as the baseline after line segmentation phases. Patel and Thakkar [46] rescaled each text line image to a fixed height to obtain baseline information. This baseline information is used for distinguishing characters.

To eliminate the possibility of overlapping, characters in words and sub-words, [40] stretched the words images horizontally to make space between two connected characters (see Figure 14).

FIGURE 14. Horizontal word stretching to avoid overlapping [40].

Khan [64] used downsampling and discarding to remove the repeated sample records for each character. Due to the wide variations in the writing speeds of the uses, the sample rate is not constant. However, a tablet has a constant temporal data rate. This leads a large number of samples to be processed at specific locations that may have already been used to process multiple samples, leading to erroneous values for feature extraction. The different pre-processing techniques used by researchers is summarized in Table 5.

C. SEGMENTATION

Dividing a source image into sub-components is known as segmentation [25]. Segmentation is used to segment and locate the text when used in Optical Character Recognition system [27]. Segmentation process for a page image can be divided into three levels as shown in Figure 15.

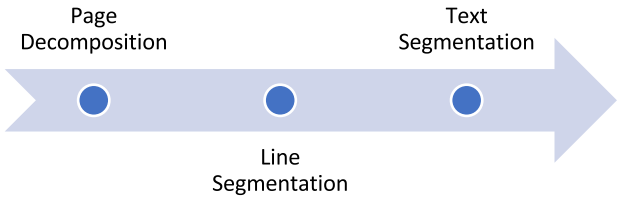


FIGURE 15. Segmentation process for a document image.

The page decomposition is known as level 1 segmentation. Page decomposition refers to the separation of text components from other elements within the source image. A source image may contain different types of elements such as tables, figures, header, footers etc. The initial step in page decomposition is identifying the different elements within the source image and dividing it into rectangular blocks. Next, each block is given a label such as a table, text, figure etc. Once the text has been extracted, further processing can be carried out only on the text portions [24].

The next process after page decomposition is the line level 2 segmentation. One of the most common methods for line segmentation is horizontal projection profile. For horizontal projection profile, the pixel values along the horizontal direction of text image are summed up. Each value of the horizontal projection profile corresponds to the total number of foreground images in that row of the text image. One of the most natural choice for line segmentation is the horizontal projection profile, however, there are some other methods too such as grouping, stochastic, smearing and Hough transform. Zero height valleys in the horizontal projection profile correspond to the whitespaces between the text lines.

Once a text document is segmented into lines, next, text segmentation is carried out into subsequent characters, ligatures or words, known as level 3 segmentation. Text segmentation for an OCR system can be divided into two main types i.e. the holistic approach and the analytical approach [2] (see Figure 16).

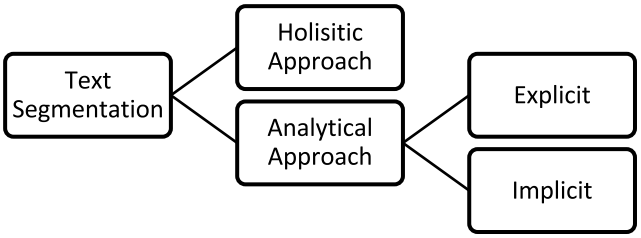


FIGURE 16. Approaches for text segmentation.

1) ANALYTICAL APPROACH

Analytical approach refers to the recognition of text by splitting it into characters. Further, the analytical approach can be divided into two main types i.e. explicit segmentation and implicit segmentation.

Explicit segmentation explicitly divides handwritten or printed text into characters. Great success has been achieved when using explicit approach for character segmentation [6], [39]–[41], [43], [72]. However, this method is prone to errors and requires extensive knowledge of a characters start and end points. Using this start and end information the characters are recognized and isolated, following the segmentation procedure is carried out. Detection of the start and end points is prone to error due to size variation, complexity and placement of characters. There are also numerous techniques to perform implicit segmentation, this may lead to over-segmentation and under-segmentation.

Text is segmented into a very small number of segments that are based on the component classes of the alphabet in implicit segmentation. Hence, the implicit approach uses a concept of over-segmentation. Implicit segmentation is also referred to as recognition based segmentation and has been used successfully in several studies [12], [46]–[49]. Segmentation and recognition, both processes of OCR are done in parallel in implicit segmentation. Implicit segmentation may pose challenging when deciding the total number of segments. Fewer segments leads to efficient computation but widely written words will not be covered. More segments means more computationally expensive, increasing the junk segments that may also be modelled by the OCR recognizer [73].

2) HOLISTIC APPROACH

When an OCR system recognizes text at word or ligature-level it is known to be using the holistic approach [17]. Over the years the holistic approach has gained immense popularity due to its upfront solution by avoiding any character-level segmentation [14], [17], [74]–[76]. Document image segmentation is one of the most significant task in document recognition (printed and handwritten). The overall accuracy of an OCR system is immensely dependent on the correct segmentation of the recognition units (character, ligature or word). As stated earlier, Urdu Nastalique script is highly cursive and context-sensitive in nature, having a lot of overlapping issues. Therefore, a proper segmentation algorithm is required that is robust enough to handle the complexities associated with Urdu script. Two of the most popular methods among researchers for ligature segmentation is the projection profile based method and connected component analysis.

3) RELATED WORK FOR SEGMENTATION

Segmentation is one of the most crucial stages in the optical character recognition process. Though considerable research has been carried out using both the holistic and analytical approaches, it's still not mature enough. Limited datasets have been used for holistic approaches. To evaluate and compare the existing algorithms and techniques, benchmarks and large datasets should be used. The segmentation based approach including the implicit [9], [12], [46]–[49], as well as explicit [6], [39]–[41], [43] segmentation, have been proposed by many researchers. However, over the years the researchers are shifting towards a more realist option of using a holistic approach for segmentation. Some of the popular studies for both explicit and implicit segmentation strategies are given in Table 6.

TABLE 6. Summary of notable contributions employing explicit and implicit segmentation strategies.

Study	Explicit Segmentation	Implicit Segmentation
Pal and Sarkar [6]	✓	
Javed [43]	✓	
Ahmad, et al. [40]	✓	
Haque and Pathan [41]	✓	
Ahmad, et al. [39]	✓	
Khan and Nagar [45]	✓	
Patel and Thakkar [46]		✓
Naz, et al. [49]		✓
Naz, et al. [47]		✓
Ahmed, et al. [12]		✓
Naz, et al. [48]		✓
Ul-Hasan, et al. [9]		✓

Numerous researchers focused on using connected component approach for holistic recognition of Urdu text [14], [17], [43], [58], [60], [67], [75]–[78]. Husain [60] presented a method for recognition of cursive Urdu Nastalique script using connected component analysis. The overall, connected component labeling was divided into two steps. First, the secondary components (Dots, Tay, Hamza and Mad) were separated from the base ligatures. Several features; solidity, number of holes, axis ratio, moments, eccentricity, curvature, normalized segment length, ratio of height and width of bounding box were analyzed to separate the special ligatures from the base ligatures. Second, the special ligatures were associated to the most feasible neighbouring base ligatures. No segmentation accuracy was reported for the proposed segmentation. Identically, [43], acknowledged those connected components that lie on the baseline as main bodies of the ligature and the rest were considered as dots or diacritics. Sardar and Wahab [67] suggested an algorithm that extracted the connected components by reading text from right to left. Moreover, all connected components were analyzed and certain rules were applied to form the final ligature. The proposed algorithm achieved an accuracy of 98.86% for extraction and association. In [14], first extracted the baseline position using the maximum horizontal projection row. Following, all connected components were extracted and separated into primary and secondary. All those components that didn't touch the baseline were considered as dots or diacritics. Dots and diacritics were then associated to their respective ligature using horizontal span of each secondary component on the baseline. Nazir and Javed [58] used horizontal projection profile for baseline identification. All components that lie on the baseline are considered as ligatures. A size threshold was set to separate the ligatures from the diacritics. Vertical projection of start and end point of diacritics was studied for ligature association. Whereas, [75] proposed a line and ligature segmentation algorithm for Urdu printed Nastalique text. For ligature segmentation, connected components were analyzed, extracted and associated. The proposed ligature segmentation algorithm examined height, width, coordinates, centroids and baseline information, leading to 99.80% accuracy. Similarly, [76], segmented text line into ligatures using primary (main body) and secondary (dots and diacritics) ligatures. Ali *et al.* [77], proposed segmentation phase for an OCR system that had two main steps, dividing the text into lines and further lines into ligatures. The first step was achieved using horizontal projection profile and for second step connected component analysis was used. The system was trained on 23204 ligatures. In [78], a ligature based OCR system was suggested by first separating the text into primary ligatures and diacritics. Once segmented, right-to-left HMMs were used for recognition purposes. The system was tested on 2017 high frequency ligatures. Similarly, [17], separated the secondary components from the main body and used HMM for training and recognition. The system achieved an accuracy of 97.93% for a total of 2000 ligatures.

Projection profile based textual image segmentation is also one of the most famous methods among researchers. In 2016, [79] presented a novel segmentation approach for Urdu Nastalique ligature recognition using projection profile. The proposed method was tested on a total of 300 ligature samples achieving segmentation accuracy of 91.3% and diacritic association accuracy of 78%. Similarly, [50] also used vertical projection profile for word segmentation followed by feature extraction.

Husain *et al.* [66] identified a total of 250 base ligatures and 6 secondary stroke ligatures used a novel algorithm for online ligature segmentation. Whereas in [63], a segmentation free hybrid approach was proposed for online Urdu handwritten script using HMM and fuzzy logic. Javed and Hussain [57] used the baseline to separate the diacritics from the main bodies and later extracted the main bodies. Some of the notable contributions used for holistic text segmentation are given in Table 7.

TABLE 7. Summary of some notable contributions using holistic approach for text segmentation.

Study	Algorithm	Accuracy Reported
Husain [60]	Connected Component Labelling	100%
Javed [43]	Connected Component Labeling	-
Husain, et al. [66]	Online ligature recognition	-
Razzak, et al. [63]	Online ligature recognition	-
Sardar and Wahab [67]	Connected Component Labelling	98.86%
Sabbour and Shafait [14]	Connected Component Labelling	-
Javed and Hussain [57]	Baseline Information	-
Nazir and Javed [58]	Connected Component Labelling	-
Ahmad, et al. [75]	Connected Component Labelling	99.80%
Din, et al. [76]	Connected Component Labelling	-
Ali, et al. [77]	Connected Component Labelling	95%
Shabbir and Siddiqi [78]	Connected Component Labelling	-
Khattak, et al. [17]	Connected Component Labelling	97.93%
Ganai and Koul [79]	Projection Profile	91.3%
Chanda and Pal [50]	Projection Profile	-

D. FEATURE EXTRACTION

When the input to an algorithm is extremely large and redundant for processing, then it can be transformed into a reduced set of parameters known as features. The features collectively are known as a feature vector. After pre-processing and segmentation, a feature extraction technique is required to extract distinct features, followed by classification and an optional post-processing phase. The primary goal of feature extraction phase is to capture the necessary characteristics of all the text

elements i.e. characters or words [27]. Features hold great significance, since, it may directly affect the efficiency and recognition rate of an OCR system [80]. The total number of features, quality of features, dataset along with the classification method are said to contribute towards an effective OCR system (see Figure 17) [80].

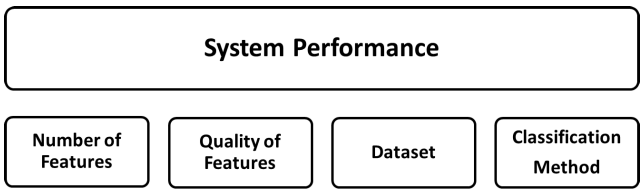


FIGURE 17. Factors affecting OCR system's performance.

Feature extraction is broadly divided into two main categories i.e. the feature learning as used by [48] and feature engineering approach as used by [81]. If features are automatically identified and extracted by it is known as feature learning approach. When hand-crafted features are identified and extracted it is known as feature engineering approach. Following feature extraction, sometimes we may need to get a reduced or subset of the initial features, a task achieved through feature selection process.

1) FEATURE LEARNING APPROACH

Feature learning enables generation of a large number of features from the data itself [82], it may improve the overall performance of the classification algorithms. Such systems are particularly valuable when such specialized features are needed that cannot be created by hand. Mostly with feature learning, unsupervised machine learning algorithms are used that trains on several layers of features to learn multi-level representations [83]. Feature learning can be supervised or unsupervised. Supervised feature learning deals with labeled input data, for e.g. supervised dictionary learning and neural networks. The labeled data allows the system to learn when the system fails to produce the correct label. Whereas, unsupervised feature learning deals with unlabeled input data, for e.g. matrix factorization [84], clustering [82] and auto-encoders [85].

2) FEATURE ENGINEERING APPROACH

When using hand-crafted features, some parameters need to be considered such as its quality, quantity, usefulness, distinctiveness and effectiveness. There are numerous features associated with each character in an Urdu alphabet. For optical character recognition system, it is necessary to observe techniques that achieve maximum recognition using simplest and minimum features. The hand-crafted features can be classified into three types [24], [86].

3) STRUCTURAL FEATURES

Structural features are based on topological and/or geometric properties of characters such as loops, wedges, start point, end point, branches, crossing points, horizontal lines, vertical lines, number of endpoints., horizontal curves at top or bottom etc. [24], [25], [40], [87]. The structural feature requires knowledge about the structure of character or the knowledge about the strokes and the associated dots that make up the character. In case of Urdu character recognition, structural feature extraction is extremely difficult since the shape of characters varies according to its neighborhood.

4) STATISTICAL FEATURES

Statistical features are related to the distribution of pixels in an image [69]. Few popular methods to extract statistical features are zoning, projection profiles and crossing and distances. Statistical features are easy to detect and are not affected by noise or distortions as compared to structural features. Statistical features provide low complexity and high speed to some extent, they also provide some level of font invariance. Statistical features may also be used for dimension reduction of the feature set.

There are different methods to extract statistical features such as zoning, crossing and distances, and projections [87]. The statistical feature can be further classified into first-order, second-order and higher order statistical features. First-order features compute properties of only individual features such as average and variance. Second-order and higher-order statistical features compute interactions between two or more pixel values that are occurring at specific locations relative to each other.

Zoning feature extraction is a popular statistical feature extraction method. The character image is divided into a pre-defined number of zones and from each zone, a feature is selected [69]. The zones might be overlapping or non-overlapping, the character strokes in different zones are analyzed. The image is usually divided into 2×2, 3×3, 4×4 etc. zones.

Counting the number of transitions from the background to the foreground pixels in a character image is known as crossing. In crossing the transitions are computed along the vertical and the horizontal lines. Along the horizontal lines of an image, the distance calculated the distance of the first pixel detected from the lower and upper boundaries in an image.

For each character image, vertical and horizontal vectors are generated for each pixel in the background. The total number of times a character stroke is intersected by any of the vectors is used as a feature.

5) GLOBAL TRANSFORMATION AND SERIES EXPANSION

These features present the image as continuous signals that contain more information and its features can be used for classification. Some of the famous features extracted are Fourier transforms, Gabor filter and transform, wavelets, Zernike moments and Karhunen-Loeve (KL) Expansion [69], [87]. Global transformation first transforms the image representation into such a form i.e. a signal so that relevant features can easily be extracted. There are numerous ways to represent a signal, such as a linear combination of a series of simpler

smaller signals, known as series expansion. Some of the most common global transform and series expansion are discussed below.

Fourier transform feature uses a magnitude spectrum of measurement vector in an n-dimensional Euclidean space as a feature vector. Fourier transform holds great significance due to its ability to recognize characters that have shifted its position by observing the magnitude spectrum. On the other hand, Hough transform is used to find parameter curve of characters. It is also used as a technique for baseline detection in text documents. Gabor transform is a special form of Fourier Transform. In Gabor transform a windowed Fourier transform is applied to a character image. The window size is not discrete and is stated by a Gaussian function [69]. Wavelets allow the representation of a signal at different levels of resolutions, hence, a series expansion technique. Finally, an Eigenvector analysis technique also known as Karhunen Loeve Expansion is used to reduce feature dimension by created new features from linear combinations of the original features. Moment features such as Zernike moments are image size, rotation and translation independent. Zernike moments are invariant descriptors for an image. The overall shape of an object is described in a compact way using only a small subset of value.

6) RELATED WORK FOR FEATURE EXTRACTION

Over the years researchers have experimented vastly with the feature engineering approach, suggesting handcrafted features for text recognition. The recent feature learning approaches have also been utilized by several researchers, and are usually used with datasets that are extremely large and complex.

Ahmad *et al.* [40] extracted simple topological features from characters. These topological features included the width, number of holes, height of the holes and the direction of holes. In [41], a unique set of features were extracted from the Nastalique characters and named it as the Nastalique feature set FS = Height, Thickness, Angle, Rotation. Hussain, et al. [42] extracted more than twenty-five different features such as height, width, loops, curves, cross, endpoints, and joints during the extraction phase. These features were processed to be used for the final character recognition. Contrarily, [35] calculated three different features from each character i.e. the height, width and the checksum. The width of the character was calculated by counting the total number of black pixels in the image from left to right direction. Similarly, the height of character was calculated by counting the black pixels from top to bottom. Finally, the X checksum calculated the total number of black pixels making up the character body. Mukhtar *et al.* [54] extracted structural features, concavity features and GSC features from normalized character images. Structural features were computed from the image, examining the gradient direction. The concavity features extracted captured the image density and the stroke information. GSC feature was represented by a 512-bit binary vector.

The feature detection methods of contour and topological are claimed to be robust but simple. Chanda and Pal [50] proposed contour and water reservoir based features. For identification of English characters, it was observed that the upper portion of the top reservoir was not open, however, for Urdu characters it was open. Similarly, the reservoir was also computed from the right direction and the water flow levels were noted. Also, the normal distance was computed between the water flow line and the right side of the bounding box. It was observed that for Urdu character this distance was three times greater than the stroke width, however, for English, it was not true. Likewise, [6] extracted topological, contour and water reservoir based features from individual characters. The topological features used were holes, total number and position of holes, the ratio of hole height to the character height, number of different components in the character. Contour features included the different profiles obtained from a portion of a character's contour. There was numerous water reservoir based features extracted such as numbers, position, height, water flow and direction of water low, and the ratio of reservoir height to the component height. Similarly, [14] extracted contour from each ligature sample. Further, each ligature shape was described using a descriptor that was termed as the 'shape context'. To extract the contour a logical grid was applied to the ligature image. Next, the transition points from black to white and/or from white to black were considered as the contour points. El-Korashy and Shafait [56] used the shape features as given by [14] as shown in Figure 18. However, some more features were added to the feature vector, such as size and the location of the dots, size features like width, height and aspect ratio were also used.

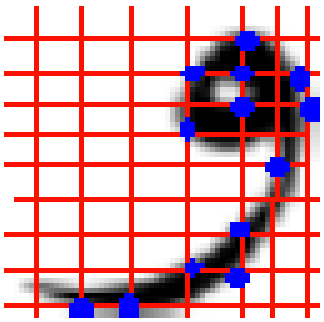


FIGURE 18. Contour (boundary) extracted for a ligature [14].

Naz *et al.* [47] extracted 12 statistical handcrafted features from the text lines by sliding a window over it. These features were not language or script dependent and were extremely simple to compute and work with. The features extracted were vertical and horizontal edges, foreground pixels, intensity features, projection features and GLCM Features. In [49] statistical features extracted were vertical edges intensities, horizontal edges intensities, foreground distribution, density function, intensity features, mean and variance of horizontal projections, mean and variance of vertical projections, GLCM features and center of gravity X and Y. The features

were extracted by sliding a window of size 4×48 from right-to-left on the text line. The results showed that the proposed features significantly reduced the labeling errors. In [63], a total of twenty-six time-variant structural and statistical features were extracted such as cusps, lines and loops based on the fuzzy logic from the base strokes.

Shamsher *et al.* [32] extracted simple features from all the characters. Once the characters were detected their pixels were copied to a simple matrix and 4 extreme points were detected. In the first pass the top, right and left extreme points were detected. In second pass, only the bottom extreme point was detected. While, Nawaz *et al.* [34] extracted chain code from each input image. The chain code was calculated by scanning the input image and calculating the string of on and off pixels. This chain code is used in the recognition phase to match column by column with every character calculated chain code for class identification. Khan, et al. [36] proposed to extract a matrix L (MxM) for each image of the 'TrainDatabase' as well as the 'TestDatabase'. Eigenvectors and Eigenvalues were calculated for each image of these databases. M' Eigenvector was selected such that it had the highest Eigenvalues. Khan *et al.* [37] used three different feature extraction techniques, namely, the Hu moments, Zernike moments and the principal component analysis. Hu moments of order 2 to 9 were used. Whereas, Zernike moments were used to calculate the Euclidean distance between the character to be recognized and the training image. Megherbi *et al.* [38] proposed and defined a unique set of seven fuzzy features to recognize the Urdu characters. Javed [43] Proposed an OCR system that first applied the process of skeletonization on the main bodies of the objects and then extracted region-based features from the objects.

Handcrafting features can be an extensive task for an extremely diverse dataset. In such situation, the entire image pixels can be taken as features. In studies [9], [12], and [46] raw pixel values were extracted as used as features, no other sophisticated features were tested. Similarly, in another study Naz *et al.* [48] employed a five-layered CNN model for extracting abstract and generic features from 60,000 handwritten digit images from the MNIST database. The first layer of the CNN extracted information from the raw pixels of the image such as the edges, lines and corner information. Features were selected from the first convolution layer in form of convolution kernels (K1-K6).

There are numerous features that can be extracted from the text images. Such as, a cross-correlation was used by Sattar *et al.* [51] for recognition of the character shapes. The character codes were written in a sequence into a text file as characters were found during the recognition phase. In [52], simple height and width features were extracted from the main bodies of the ligatures. C 4.5 algorithm was used to evaluate the significance of both the height and width. It was found that the width was more significant, it was used for dividing the training data into four subclasses. Khan and Khan [53] proposed a novel technique that utilized the point feature matching, SURF features. Point feature matching uses point correspondence between the target and the reference image for detecting objects. A total of 100 strongest SURF feature points were found in the reference image and 300 strongest points were found in the target image. However, Lehal and Rana [55] experimented with different feature extraction techniques. They used a combination of DCT, Gabor filters and zoning features. For Gabor filter, the word image was normalized to 32×32 pixels and partitioned into 16 sub-regions of 8×8 size. The images were convolved with symmetric and odd-symmetric Gabor filters. Higher value DCT coefficients were extracted in a zigzag fashion and stored a feature vector of size 100. For zoning features, the image was divided into 3×3, 4×4 and 7×7 zones. For each zone, the percentage of black pixels were calculated and various experiments were done. Nazir and Javed [58] extracted a feature vector that contained the code of the mark, base and the diacritics associated with them. Whereas, [59] used DCT, Gabor, gradient and directional features for the classification of the primary components. Husain [60] extracted features in two stages, first features were extracted only from the special ligatures i.e. solidity, number of holes, axis ratio, eccentricity, moments, normalized segment length, curvature, the ratio of bounding box width and height. In the second stage, twenty new features were added to the feature vector of the base ligature, which was done to associated special ligatures to the base ligatures. In [17], a model was proposed that used features capturing projection, concavity and curvature information. Right-to-left sliding windows were used to extract this information from the ligatures and feed it for training. Javed *et al.* [31] extracted diacritic dependent feature vector. If a diacritic was detected then the vector was '0', else if there was no vector present then after detecting it the vector was set to '1' Whereas, [62] extracted a feature vector after careful analysis of the Urdu characters. They extracted a set of Rubine features i.e. length of the bounding box diagonal, angle of the bounding box diagonal, distance between the first and last point, cosine of the angle between the first and last point, sine of the angle between the first and last point, total length of the primary stroke, total angle traversed, sum of absolute value of angle at each point, sum of the squared value of those angles from the primary components. Also, separate features were extracted from the secondary strokes that included the number, length, positioning and Number of dots in secondary strokes. Khan [64] used a multilevel one-dimensional wavelet analysis with Daubechies wavelet (db2) at level 2 to form the feature vector. In [65] geometric invariant features that were font, scale, rotation and shift invariant were extracted i.e. cosine angles of trajectory, discrete Fourier transform of trajectory, inflection points, self-intersections, convex hull, radial feature, grid (orthogonal and perspective), and retina feature. A feature vector was extracted by [66], the stroke (x, y) coordinates, chain codes and unique features for every stroke were detected. Total 20 features were extracted from the base strokes that included start vertical, end vertical, Horizontal R2L, Horizontal L2R, Hedge, Curve L2R, CurveR2L,

loop flag etc. 6 features were extracted from the secondary strokes. Sardar and Wahab [67] extracted five features from single ligature and characters. Four features were extracted using 3 types of sliding windows. These sliding windows computed the ratio between the white and black pixels. One feature was extracted using the Hu Invariant Moment. While [45] extracted and created vectors from the binary images, SOM model was used to captures the invariant features of Urdu script. Some of the notable contributions using different features are given in Table 8.

TABLE 8. Notable contributions using different features.

Related Work	Features		
Shamsher, et al. [32]	Extreme Points		
Ahmad, et al. [40]	Topological Features		
Haque and Pathan [41]	Height, Thickness, Angle, Rotation		
Hussain, et al. [42]	Height, Width, Loops, Curves, Cross, End Points, and Joints		
Nawaz, et al. [34]	Chain Code		
Tariq, et al. [35]	Height, Width and Checksum		
Khan, et al. [36]	Eigen Vector and Eigen Values		Principal Component Analysis
Khan, et al. [37]	Hu Moments	Zernike Moments	Principal Component Analysis
Megherbi, et al. [38]	Fuzzy Features		
Pal and Sarkar [6]	Topological	Contour	Water Reservoir
Javed [43]	Region-Based		
Ahmed, et al. [12]	Raw Pixels		
Sabbour and Shafait [14]	Contour		
Chanda and Pal [50]	Contour	Water Reservoir	
Sattar, et al. [51]	Cross Correlation		
Hussain, et al. [52]	Height and Width Features		
Khan and Khan [53]	SURF Features		
Mukhtar, et al. [54]	Structural	Concavity	GSU
Lehal and Rana [55]	DCT	Gabor Filters	Zoning Based Features
El-Korashy and Shafait [56]	Contour	Size, Location of The Dots	Size Features Like Width, Height and Aspect Ratio
Nazir and Javed [58]	Code of The Mark, Base and The Diacritics		
Husain [60]	Solidity, Number of Holes, Eccentricity, Moments, Normalized Segment Length, Curvature, Ratio of Bounding Box Width and Height, Axis, Ratio		
Khattak, et al. [17]	Projection, Concavity and Curvature Information		
Javed, et al. [31]	'0' & '1'		
Ul-Hasan, et al. [9]	Raw Pixels		
Patel and Thakkar [46]	Raw Pixels		
Naz, et al. [47]	Statistical Features		
Naz, et al. [48]	CNN Features		
Naz, et al. [49]	Statistical Features		
Shahzad, et al. [62]	Rubine Features		
Razzak, et al. [63]	Statistical		Structural
Khan [64]	Wavelet Analysis		
Jan, et al. [65]	Geometric Features		
Husain, et al. [66]	Stroke Coordinates, Chain Code and Unique Features		
Sardar and Wahab [67]	Ratio Between White/Black Pixels and Hu Invariant Moment		
Khan and Nagar [45]	SOM model used to capture invariant features		

E. CLASSIFICATION AND RECOGNITION

Classification can be defined as a computational process that sorts images into groups/classes according to their similarities. Classification is a significant application of image retrieval. Classification simplifies searching through an image dataset to retrieve those images with particular visual content. All classification algorithms assume that the image in question depicts one or more features (e.g., geometric) and that each of these features belongs to one of several distinct and exclusive classes.

Classification algorithms typically employ two phases of processing: training and testing [88]. In the training phase, the characteristics of typical image features are isolated. Based on the image features a unique description of each classification category, i.e. training class, is created. In the subsequent testing phase, these feature-space partitions are used to classify image features.

Two similar concepts i.e. machine learning and deep learning have been on the rage in the research communities during the past several years. Deep learning is not new, but recently have gained hype from the research community and is getting more attention. Below both, the concepts have been explained, along with the related studies in Urdu Optical character recognition systems.

1) TRADITIONAL MACHINE LEARNING

Machine learning is a field of artificial intelligence that aims to mimic intelligent abilities of the humans by machines [89]. Machine learning involves the important queries and procedures needed to make the machines capable of learning. It is difficult to define learning precisely since it covers a broad range of processes. In most dictionaries, the phrases used for its definition are "to gain knowledge," "understanding of" and "to gain some skill by study, instruction, or experience". Some of the famous traditional machine learning techniques used with the character recognition systems are Neural Network, Support Vector Machine, K-Nearest Neighbors, Bayesian Classification, and Decision Tree Classification. Machine learning can be divided into two major types. However, there are some other types of machine learning techniques also available, such as, reinforcement learning, semi-supervised learning and learning to learn.

Supervised machine learning involves inferring a function from labeled training data [90]. In supervised learning, we work with a pair of input and its desired output. Supervised learning algorithms analyze the training data and produce a function. If the function has been inferred correctly it can then be used to correctly determine the future unseen input instances. Supervised learning is concerned with classification or regression. The primary goal is to enable the computer to learn a classification system that has been created by users. Character and digit recognition systems are famous examples of classification based learning.

Classification based learning is applied to any problem where classification is useful and easier to determine. However, classification is not always supervised, unsupervised learning may also be used for classification problems. Some of the renowned supervised learning algorithms are Neural Networks, Naive Bayes, Nearest Neighbor, Regression models, Support Vector Machines (SVMs) and Decision Trees [91].

Comparatively, in unsupervised machine learning, the major goal is to enable the computer to learn how to do something, without telling it how to do it. Unsupervised learning is much harder than the supervised learning. Unsupervised learning involves finding structures in unlabeled data [90]. There are two approaches to unsupervised learning, first, clustering which includes k-means, mixture models and hierarchical clustering. Second, feature extraction techniques for e.g. independent component analysis, non-negative matrix factorization, and singular value decomposition. Some of the unsupervised learning algorithms are Neural network based approaches for meeting a threshold, Partial based clustering, Hierarchical clustering, Probabilistic based clustering, and Gaussian Mixture Models (GMM).

2) DEEP LEARNING

Widely accepted, the deep learning is taken as a recently developed but important part of a broader family of machine learning methods [92]. Deep learning architecture are also used for performing classification tasks. Some differences between the deep learning approach and the traditional machine learning approach are discussed in the subparagraphs given below in this section.

The performance of a classification algorithm highly depends on the features that have been identified and extracted. Traditional machine learning methods use the feature engineering approach to extract features for classification [93]. Whereas, deep learning automatically finds out and extracts the raw pixels as features for classification. When hand-crafted features are used for classification, it is known as feature engineering. In feature engineering, the domain knowledge is put into use for feature creation and extraction [93]. Feature engineering may pose to be difficult, expensive and time-consuming in terms of knowledge. Once the features have been identified they are hand-coded as per data type and domain. Hand-engineered features can be of different types, such as shape, pixel values, position, textures and orientation. Deep learning on the other hand automatically extracts high-level features from the data [94]. Hence, the task of finding and extracting unique features for each problem are subsided. For example, the famous, Convolution Neural Network learns different low-level features such as lines and edges from an object in its early layers, it then learns medium-level features and later the final high-level representation is learned.

The performance of deep learning increases as the scale of data increases [94]. For smaller datasets, deep learning algorithms don't perform that well. A large amount of data is required by deep learning algorithms to understand the data perfectly. Contrarily, traditional machine learning methods perform well for smaller datasets, its performance is not affected by larger datasets.

The traditional machine learning algorithms can work well on low-end machines. On the other hand, deep learning algorithm requires high-end-machines to process and classify the data. GPU (Graphical Processing Unit) is capable of carrying out a large amount of matrix multiplication operations, it is an essential requirement for the working of deep learning [95].

The traditional machine learning algorithms usually break down a problem into sub-parts before solving it. There are usually two steps involved, object detection and recognition. Deep learning, on the other hand, solves the problem without sub-dividing it into sub-parts. It provides an end-to-end solution for problems.

The traditional machine learning algorithms take less time to train, probably, from few seconds to maybe few hours. Whereas, deep learning algorithms comparatively take a long time to train due to its large number of parameters. However, the traditional machine learning algorithms such as K-NN may take more test time.

Deep learning architectures give excellent results, having near human perfection. But with deep learning, it's not possible to understand why has it given this much accuracy. The information about the nodes that were activated is known and can be mathematically found but the work of neurons and their modeling strategy is unrevealed. Hence, the results can't be interpreted. Traditional machine learning algorithms, however, allows you to easily interpret the results and also gives clear rules, suggesting what was chosen and why did it choose it.

3) RELATED WORK FOR CLASSIFICATION AND RECOGNITION

Machine learning and deep learning both are very similar, but more recently, deep learning has gained immense hype among the research community. Traditional machine learning methods like SVM, k-NN, Template matching, Decision Tree etc. is being overtaken by the not so traditional high computation deep learning, such as the LSTM, Auto-encoders, Deep Belief Networks, Deep Neural Networks etc. Here, different classifiers are discussed in perspective of the script connectivity used by the recognition system i.e. isolated or cursive (analytical/holistic).

Neural Networks are extremely complex structures but performance and recognition wise they are remarkable. Several authors opted to develop recognition systems for isolated Urdu handwritten or printed characters using the neural network. Shamsher *et al.* [32] used supervised learning and Feed Forward Neural Network for training and testing. The prototype of the system was tested on printed Urdu characters and achieved an accuracy of 98.3% on an average. Likewise, Tariq *et al.* [35] used Neural Network for constructing an OCR, named Soft converter. The prototype of the system had an accuracy rate of 97.43%. In [38] a two-stage neural network was used to classify 36 Urdu characters and reported

a high recognition accuracy. The classifier consisted of a multi-hidden-layers Back Propagation Neural Network architecture for classification of Urdu characters within a subclass. Khan [64] used a Back Propagation Neural Network classifier was used for single stroke Urdu characters written in their initial half form. It was tested on 3600 instances of Urdu letters written in their initial half forms, achieving an overall accuracy of 91.3%.

Contrary to the complex neural network architectures, other traditional machine learners like HMM, k-NN, SVM, principal component analysis, decision tree, K-SOM, linear classifiers and template matching has also been reported to give good results. Akram *et al.* [33] presented an HMM technique to be tested on single Urdu character ligatures, the system achieved a recognition rate of 96% for scanned data and 98% for manually generated data. Khan *et al.* [36] proposed an OCR system for Urdu script using the Principal Component Analysis. The recognition accuracy reported for noise-free images was superior, while for noisy images the recognition rate dropped. Overall 96.2% accuracy was reported for character recognition, the system was implemented using the MATLAB tool. Khan, et al. [37] used decision tree from classification, specifically the decision tree J-48 algorithm. The system was tested on 441 handwritten and machine written Urdu characters and accomplished a recognition rate of 92.06%. Hussain, et al. [42] recognized the segmented characters in two steps, first by categorizing the different shapes for the same character into 33 classes using the auto-clustering technique, Kohonen Self-Organizing Map. Next, in feature extraction, the 25 extracted features were once again processed for the final recognition. The system was tested for 104 segmented characters and was fully capable to recognize it. In [54] OCR system was evaluated on a dataset of about 1300 handwritten words. For classification k-NN and SVM was used, achieving an accuracy on an average of 70% for top choice and 82% for top three choices. Likewise, Lehal and Rana [55] used SVM, k-NN and HMM classifiers for training. The proposed system was capable of recognizing 9262 Urdu ligatures with more than 98% recognition accuracy. In [65] linear support vector machine was used for training and testing, giving a 97% classification accuracy and a very low false rejection rate on the test data. Nawaz, et al. [34] implemented a system and tested it for different types of fonts using pattern matching and it achieved an accuracy of 89% for the isolated character with 15 char/sec recognition rate. Similarly, [61] applied template matching technique to different sentences of Urdu script having 72 font sizes. All identified objects were saved as templates on basis of comparison to other templates already available in the dataset. Whereas, [53] used a point feature matching technique i.e. SURF for feature extraction, classification and recognition. The system was tested on 20 newspaper clippings that had 177 objects and achieved a recognition accuracy of 93%. Hussain *et al.* [52] analyzed the tesseract engine for the recognition of Nastalique Urdu language. The original tesseract system was reported to have 65.59% accuracy for 14 font sizes and 65.84% accuracy for 16 font sizes. While the modified system outperformed the original system and improved the recognition from an average of 170 milliseconds to 84 milliseconds. The system was trained using 14,750 main body images for 14 and 16 font sizes separately achieving an accuracy of 97.87% for 14 font sizes and 97.71% for 16 font sizes. Some of the notable contributions for isolated character Urdu OCR are given in Table 9.

Analytical recognition system for Urdu printed text has using Neural Networks has obtained phenomenal results on standard Urdu datasets like UPTI. Ahmad *et al.* [40] used Neural Networks for training different forms of a character. The system was tested on synthetic images as well as real-world images obtaining an accuracy of 93.4%. In [39] a Feed Forward Neural Network was used to train 56 different classes of the 41 characters, each having a total of 100 samples. The prototype of the system was implemented in MATLAB, having 70% reported accuracy on the average. Whereas, [12] investigated the performance of recurrent neural network (RNN) for cursive and non-cursive scripts. Bidirectional Long Short-Term Memory (BLSTM), a variant of RNN was used for evaluation on Latin as well as Urdu script. A special layer Connectionist Temporal Classification (CTC) was used for sequence alignment. The character recognition accuracy for non-cursive Latin script was 99.17% for the UNLV-ISRI dataset. For cursive Urdu, without position information, the accuracy reported was 88.94% and for the position, accuracy was 88.79%, evaluated on unconstrained datasets. Similarly, in [9] a variant of LSTM, i.e. a Bi-directional LSTM was evaluated online images from the UPTI dataset. The system was evaluated for characters by ignoring its shape and by considering the shape. An error rate of 5.15% for the first case and 13.6% for the second case was evaluated. Patel and Thakkar [46] used multidimensional BLSTM and ANFIS method. The ANFIS method learned different membership functions and rules from the data. This adaptive network was composed of nodes and directional links, providing a relationship between the inputs and the outputs. It was evaluated on the UPTI dataset, achieving recognition error rate of 5.4%. Likewise, [47] proposed a system that extracted statistical features and fed it to multi-dimensional long short-term memory recurrent neural network (MDLSTM RNN) with a connectionist temporal classification (CTC) output layer. The CTC layer was used to label the character sequences. The system was evaluated on the standard UPTI dataset having 10,000 lines written in Nastalique font. Overall the system gave promising recognition rate of 96.40%. Naz *et al.* [48] extracted invariant features using the Convolution Neural Network and then fed these features to the Multi-dimensional LSTM (Long Short-Term Memory) for learning. Experiments were carried out of UPTI dataset and achieved an accuracy of 98.12%. In [49] high accuracy was achieved for Urdu Nastalique

TABLE 9. Summary of contributions for isolated character Urdu OCR.

Study	Features	Classifier*	Dataset	Accuracy
Shamsher, et al. [32]	Extreme Points	FFNN	100 Characters	98.3%
Akram, et al. [33]	Extreme Points	HMM	Manual Data and Scanned Data	98% and 96%
Nawaz, et al. [34]	Chain Code	PM	Urdu Characters	89%
Tariq, et al. [35]	Height, Width, Checksum	Neural Network	Soft Matching DB and Hard Matching DB	97.43% and 100%
Khan, et al. [36]	Principal Component Analysis, Eigen Space	Threshold Euclidean Distance	Train and Test Database	96.2 %
Khan, et al. [37]	Hu Moments Zernike Moments Principal Component Analysis	Decision Tree	441 Characters	92.06%
Megherbi, et al. [38]	Fuzzy Features	BPNN	36 Characters	--
Hussain, et al. [42]	Height, Width, Loop, Joint, Cross, Curves, End Points	K-SOM	104 Characters	80 %
Hussain, et al. [52]	Height and Width	Character Class Analysis	14,750 14 Font Size Images and 16 Font Size Images	97.87% and 97.71%
Khan and Khan [61]	White Pixel Length	Template Matching	Different Sentences	--
Khan and Khan [53]	SURF	Point Feature Matching	20 Newspaper Clippings	93%
Mukhtar, et al. [54]	Structural Features	k-NN SVM	1300 Handwritten Words	70% and 82%
Lehal and Rana [55]	DCT Gabor Filters Zoning Features	k-NN HMM SVM	9262 Ligatures	98%
Khan [64]	Wavelet Analysis	BPNN	3600 Letters	91.30%
Jan, et al. [65]	Geometric	SVM	Primary Ligature of each Alphabet Class having 128 Samples	97%

Classifier*
FFNN: Feed Forward Neural Network HMM: Hidden Markov Model PM: Pattern Matching
K-SOM: Kohonen Self-Organizing Map (SOM) Algorithm BPNN: Back Propagation Neural Network
k-NN: k- Neural Network SVM: Support Vector Machine

using statistical features and multi-dimensional long short-term memory. The system was evaluated on the Urdu Printed Text Images dataset and achieved a good recognition accuracy of 94.97%. In another study by Naz *et al.* [96], zoning features were used in combination with a 2-Dimensional LSTM, achieving an accuracy of 93.39%.

Several other methods have also been used for character level recognition system. Haque and Pathan [41] proposed a finite state Nastalique text recognizer had the following two components: Character Shape Recognizer and Next-State Function. When the recognition process is carried out the character codes were stored in a text file. These text files were later concatenated in the sequence as the ligatures were found. In [30], a segmentation based technique was used for recognition of Urdu Nastalique text using the HMM classifier. The OCR system was tested on 79093 instances of 5249 main body classes and achieved an overall recognition accuracy of 97.11%. The system was also tested on document images extracted from different books and achieved main body accuracy of 87.44%. Whereas, [6] implemented a feature based tree classifier. The system was tested on 3050 printed Urdu characters and numerals, it was reported to achieve an accuracy of 97.8%. Javed [43] used HMM technique to make ligature independent OCR system for Urdu Nastalique script using the HMM technique. For testing and analyzing different words were extracted from the Nokia dictionary. Three or more samples of each word were taken, that was written in Noori Nastalique font with the font size of 36. For each letter accuracy was calculated separately, overall on average an accuracy of 95.76% was attained for a total of 2898 characters. Similarly, the system was also analyzed for a total of 1692 ligatures achieving an accuracy of 92.73%. In [44] template matching was used for identification of character within the image. After identification, the Roman Urdu character corresponding to the Urdu Nastalique character in the image was selected. Khan and Nagar [45] proposed an algorithm based on the Kohonen SOM algorithm for recognition of Urdu characters. The training set was composed of 200 samples while the testing set was composed of 800 samples. The recognition rate of 79.9% was reported for the first choice and about 98.5% for the top three choices. Some of the notable contributions for cursive character Urdu OCR are given in Table 10.

Ligature based OCR systems have gained immense popularity recently. Javed and Hussain [57] proposed a Hidden Markov Model and a rule-based post-processor system that achieved an accuracy of 92.73% for printed and then scanned Urdu document images having 36 font sizes. In [17], Hidden Markov Model (HMM) was used to train the system. It was evaluated on 2000 frequently occurring Urdu ligatures and got a recognition rate of 97.93%. Equally, [31] used HMM for recognition of Urdu ligatures. A total of 3655 ligatures were tested and 3375 ligatures were accurately identified, giving an overall accuracy of 92%. Razzak *et al.* [63] used a hybrid HMM and fuzzy logic classification technique for large and complex data recognition. The proposed OCR system was evaluated on 1800 ligatures and obtained an accuracy of 87.6% and 74.1% for Nastalique and Naskh, respectively. Whereas, [76] presented

TABLE 10. Summary of contributions for cursive character Urdu OCR.

Study	Features	Classifier*	Dataset	Accuracy
Ahmad, et al. [40]	Topological features	Neural Networks	Synthetic Images and Real-world Images	93.4%
Haque and Pathan [41]	Height, Thickness, Angle, Rotation	Finite State Model	--	--
Ahmad, et al. [39]	--	FFNN	56 Classes Of 100 Samples	70%
Hussain and Ali [30]	--	HMM	79093 Instances Of 5249 Main Body Classes Document Images from Books	97.11%. 87.44%.
Pal and Sarkar [6]	Topological Contour Water Reservoir	Tree Classifier	3050 Characters	97.8%
Javed [43]	Region-Based	HMM	2898 Characters and 1692 Ligatures	95.76% and 92.73%
Iqbal, et al. [44]	--	TM	--	--
Ahmed, et al. [12]	Raw Pixels	BLSTM	UNLV-ISRI for Latin and Urdu-Jang and UCOM for Urdu	99.17% and 88.94% and 88.79%
Ul-Hasan, et al. [9]	Raw Pixels	BLSTM	UPTI	94.85%
Patel and Thakkar [46]	Raw Pixels	Multi-dimensional BLSTM and ANFIS	UPTI	94.6%
Naz, et al. [47]	Statistical Features	MDLSTM	UPTI	96.40%
Naz, et al. [48]	CNN Features	MDLSTM	UPTI	98.12%
Naz, et al. [49]	Statistical Features	MDLSTM	UPTI	94.97%
Naz, et al. [96]	Zoning Feature	2DLSTM	UPTI	93.39%
Khan and Nagar [45]	SOM Model Based Invariant Features	K-SOM	Testing 200 samples	79.9% (first choice) 98.5% (top three choices)

Classifier*
FFNN: Feed Forward Neural Network
HMM: Hidden Markov Model TM: Template Matching
K-SOM: Kohonen Self-Organizing Map (SOM) Algorithm
BPNN: Back Propagation Neural Network
k-NN: k- Neural Network SVM: Support Vector Machine

an OCR system that relied heavily on the statistical features and employed Hidden Markov Models for classification. A total of 1525 unique high-frequency Urdu ligatures from the standard Urdu Printed Text Images (UPTI) database were considered in the study. Hidden Markov Models were trained separately for each ligature. The system gave an overall ligature recognition rate of 92%.

Tree-based classifiers have also been tested for word and ligature based recognition systems. Chanda and Pal [50] applied a binary tree classifier on 8011 words, out of which Urdu words were 3210, English 2738 and Devanagari were 2063. Overall the accuracy of the system was 97.51%. The highest accuracy was reported for Urdu script of 98.09%. The confusion rate for the proposed system was 0.78%. In [56] Nearest Neighbour and Random Forrest classifier were used to evaluate the system. First, spectral hashing was carried out and the accuracy of the system was measured and accuracy of 81.5% was achieved. Second, Random forest classifier was used for identification of different ligatures, such as one-character, two-character, and three-or-more character ligatures. An accuracy of 98% was achieved using the Random forest for single character ligatures. Likewise, [62] used a weighted linear classifier for training and classification. The proposed concept was integrated into an application used for aiding people in learning the Urdu language. Five Samples Of 38 Urdu characters were used for training the system. An accuracy of 92% was obtained for Urdu native writers. While an accuracy of 73% was reported for non-native Urdu writers.

K-NN, SVM and correlation methods have also given high accuracies for text recognition. In [59] SVM and k-NN classifier were used to train and recognize 11,000 Urdu ligatures. An overall accuracy of 90.29% was reported for Urdu text images. Sabbour and Shafait [14] evaluated the performance of the system for both Urdu and Arabic script. After segmentation, the unknown ligatures from the dataset were classified in the training phase using the k-NN, k-Nearest Neighbor. The performance of the system for Urdu clean text was 91% and for Arabic text was 86%. However, [67] used K-Nearest Neighbors (k-NN) algorithm for features matching and applied Euclidean distance with 10 nearest neighbors. A total of five features were extracted independently by the k-NN. Overall the system gave a recognition accuracy of 97.12% different printed and handwritten documents of different fonts and script. Whereas, [51] proposed an algorithm based on correlation for printed Nastalique text. Experiments were carried out on a small subset of text and overall the recognition results obtained were very encouraging. Nazir and Javed [58] proposed a methodology for the processing and recognition of diacritics based Nastalique Urdu script. The system was capable of recognizing invariant cursive texts of 48 font size. Overall an accuracy of 97.40% was reported for the proposed new technique, correlation method, based recognition of 6728 main Urdu ligatures.

Husain [60] used 200 carefully selected ligatures for training a back propagation neural network and got an accuracy of 100%. A total of 34 features were fed into the neural network, having 34 inputs, 65 hidden neurons and 45 output neurons. Correspondingly, [66] presented the design of an online Urdu

handwriting recognition system that used a BPNN for the classification of every stroke to its respective class. A total of 850 single characters, 2 characters and 3 character ligatures were fed to the BPNN for classification. All these ligatures formed approximately 50000 words. A recognition rate of 93% was reported for base ligatures and a recognition rate of 98% was reported for secondary strokes. A stacked denoising autoencoder for automatic feature extraction from raw pixel values of ligature images [97]. Different stacked denoising autoencoders were trained on 178573 ligatures having a total of 3732 classes. For training and testing the un-degraded (noise free) UPTI (Urdu Printed Text Image), data set was used. Overall the recognition accuracy for the system was in the range of 93% to 96%. Ahmad *et al.* [98] proposed a Bidirectional long short-term memory (BLSTM) architecture for recognition of Urdu Nastalique sentence images. A gated BLSTM (GBLSTM) model for recognition of printed Urdu Nastalique that incorporated raw pixel values as input was used. The model was trained and tested on the un-degraded version of UPTI dataset achieving an accuracy of 96.71%. Some of the notable contributions for ligature based Urdu OCR are given in Table 11.

F. POST-PROCESSING

Post-processing is the final stage of an OCR process, it includes tasks which aims towards the improvement or correctness of classification and recognition of the system [99], [100]. The chosen classifier might not produce accurate results for an image. Hence post-processing might be required. It may include different processes such as grammar correction, spell-checking, text-to-speech conversion and improving the overall recognition rate and output.

VII. URDU DIGIT RECOGNITION SYSTEMS

A lot of research work has been done for recognition of numerals from different languages like English and Chinese [101]. However, very limited research has been carried out for the recognition of numerals for alphabets of Farsi, Arabic and Urdu. Urdu numerals are composed of different curves and line segments and written using old Arabic script. The numerals for Urdu are similar to that of the Farsi script, however, commonly old Arabic numeral form is used for writing it instead of Urdu numeral form. Ansari and Borse [102] proposed a research for OCR of handwritten Urdu Digits. In the pre-processing stage, different processes were applied to the Urdu digit images, such as gray level image conversion, image thresholding, median filtering, and image normalization (64 × 64). For feature extraction, different types of Daubechies Wavelet transform and zonal densities from different zones of images were used. Back Propagation Neural Network was used for classification of the digits. The proposed system was tested on 200 samples of each digit, a total of 2150 samples, and achieved recognition accuracy of 92.07% on average. A digit training and testing sample is given in Figure 19.

TABLE 11. Summary of contributions for ligature based Urdu OCR.

Study	Features	Classifier*	Dataset	Accuracy
Rana and Lehal [59]	DCT Gabor Filters Directional Gradient	SVM k-NN	11,000 Ligatures	90.29%
Khattak, et al. [17]	Projection Concavity Curvature Information	HMM	2,000 Ligatures	97.93%
Shahzad , et al. [62]	Rubine Features	Weighted Linear Classifier	5 Samples of 38 Characters	92.80%
Sabbour and Shafait [14]	Contour	K-NN	UPTI-10,000 Ligature and Arabic 20,000 Ligatures	91% and 86%.
Chanda and Pal [50]	Contour Water Reservoir	Binary Tree Classifier	3210 Urdu Words	98.09%
Sattar, et al. [51]	--	Cross Correlation	Small Sub-Set	--
El-Korashy and Shafait [56]	Contour Size and Location of The Dots width, height, and aspect ratio	Nearest Neighbour Classificatio n Random Forest	Training-20,000 Testing-18000	81.5% 98%
Javed and Hussain [57]	--	HMM	1692 Ligatures	92.73%
Nazir and Javed [58]	Code of The Mark, Base and The Diacritics	Correlation Method	6728 Ligatures	97.40%
Husain [60]	Solidity, Number of Holes, Eccentricity , Moments, Normalized Segment Length, Curvature, Ratio of Bounding Box Width and Height, Axis, Ratio	Feed Forward Back Propagation neural network	200 Ligatures	100%
Javed, et al. [31]	'0' & '1'	HMM	3655 Ligatures	92%
Razzak, et al. [63]	Statistical Structural	HMM Fuzzy Logic	1800 Ligatures	87.6% Nastaliqu e and 74.1% Naskh
Husain, et al. [66]	Stroke Coordinates , Chain Code and Unique Features	BPNN	850 Ligatures	93%
Sardar and Wahab [67]	Ratio Between White/Blac k Pixels, Hu Invariant Moment	k-NN	Handwritte n and Printed Text	97.12%
Ahmad, et al. [97]	Stacked Autoencode r Features from Raw Pixels	Softmax	UPTI	93% to 96%
Ahmad, et al. [98]	Raw Pixels	GBLSTM	UPTI	96.71%
Din, et al. [76]	Statistical	HMM	UPTI Ligatures	92%

Classifier*
FFNN: Feed Forward Neural Network
HMM: Hidden Markov Model TM: Template Matching
K-SOM: Kohonen Self-Organizing Map (SOM) Algorithm
BPNN: Back Propagation Neural Network
k-NN: k- Neural Network SVM: Support Vector Machine

In [103] a handwritten Urdu Character recognition technique was presented based on Zernike invariants and SVM as the classifier. Zernike moment invariants were used for

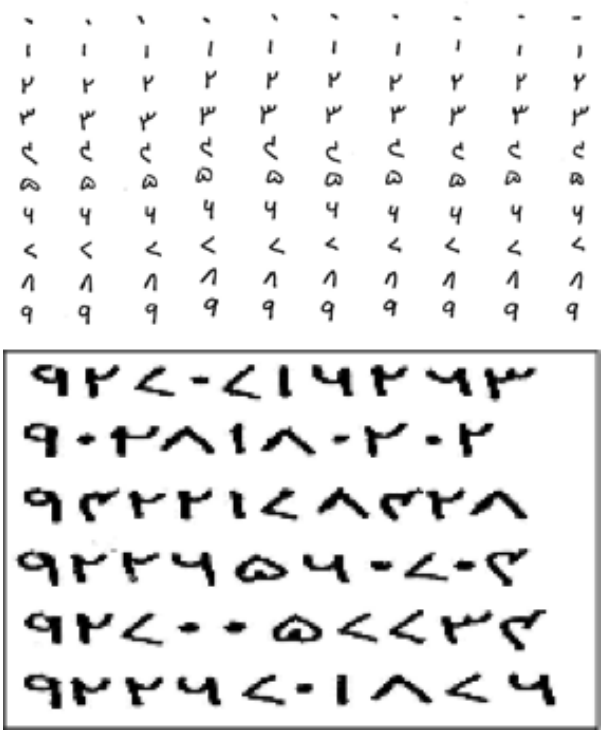


FIGURE 19. Digit training and testing sample [102].

feature extraction, overall 22 features were extracted from each numeral. Support Vector Machine (SVM) was used for classification and got a success rate of 96.29%. The problem of handwritten offline numerals was addressed by Uddin *et al.* [104]. A novel approach of Non-Negative Matrix Factorization (NMF) was proposed in the research. A two-page form was developed to get the input of Urdu numerals from a variety of people. Numerals from the first page of the form were used for training. Numerals the second page were used for testing. The pre-processing stage involved various steps, noise removal, locating the rectangular boxes, and numeral isolation from the form images, padding the numerals with appropriate margins to preserve its orientation and resizing the numeral images to 175×175. Overall the system achieved an accuracy of 86% for nearly 1600 pages. Some of the notable contributions for Urdu numeral recognition systems are given in Table 12.

TABLE 12. Summary of Urdu numeral recognition systems.

Study	Features	Classification	Dataset	Accuracy
Ansari and Borse [102]	Daubechies Wavelet Transforms and Zonal Densities	Back Propagation Neural Network	2150 Samples	92.07%
Kaushal, et al. [103]	Zernike Moments	SVM	700 Samples	96%
Uddin, et al. [104]	Non-Negative Matrix Factorization (NMF)	L2-Norm	1600 Pages	86%

VIII. CONCLUSION AND FUTURE DIRECTIONS

In this extensive literature study, it has been found that this research area is relatively new and wide open for research and development. Its effective implementations can lead to numerous applications in the future. The literature for Urdu optical character recognition is reviewed thoroughly to conclude by identifying the knowledge gaps and figuring out the future research and development in the field. Overall the literature reviewed is based on different categories and phases of the OCR system i.e. image acquisition, pre-processing, segmentation, feature extraction, classification and recognition. Significant conclusions are drawn from the addressed literature. The image acquisition phase is highly related to the type of OCR system being used. Since there are very few online recognition systems for Urdu script, very few digitizing tablets have been used for input in image acquisition phase. Pre-processing step of thresholding is most frequently used when developing OCR systems for Urdu script. Offline Urdu character recognition systems are extremely complex, most of which have been used for recognizing isolated non-cursive Urdu text. Comparatively, the segmentation-free systems i.e. holistic systems are very scarce. If the recognition system deals with ligatures, it's more appropriate to extract features that are not concerned with the structure of the text, since, there are a lot of variations. Due to large number ligatures in Urdu text, it's more appropriate to do clustering to identify the appropriate number of textual classes.

Looking in depth into the literature several suggestions can be drawn for the future research. The extensive review of the literature has led to identifying several open problems. The possible solutions and future directions to tackle each problem for Urdu optical character recognition have also been given in this section. Some of the open problems are given below.

- Test Datasets: The existing studies for Urdu Nastalique have been trained and tested on small datasets.
- Labeling: How to label characters when it possesses different shapes at different positions and due to neighboring characters?
- Segmentation: Urdu script is highly cursive in nature, how to make sure the text recognition system is capable of performing segmentation without affecting the original text?
- Feature Extraction: What type of features can be used for effective ligature recognition?
- Classification: How to decide which algorithm works best for ligature based recognition systems?

Most of the existing Urdu Nastalique based recognition systems have been trained and tested on small datasets. Standard datasets need to be created, maintained and followed to evaluate the all the research studies in future. Labeling of data is an important pre-requisite for supervised learning.

It's extremely intensive job to manually label characters for large dataset due to the variations of character's shape within the ligature or word. As a solution, labeling any dataset with ligatures is more accurate and easier. Another problem is that of text segmentation, Urdu text written in Nastalique is extremely cursive, this gives rise to several issues during the development of Urdu recognition system dealing with analytical segmentation. This problem can be tackled by shifting the focus towards segmentation-free/holistic ligature based recognition systems. Automated Feature learning is an extremely time-consuming process, instead, hand-crafted manual features can be used since it delivers the same results but with less processing. Many of the prevalent studies for character-based recognition systems also heavily rely on hand-engineered feature extraction methods. In hand engineered features, the structural features deal with the overall structure of the character and may seem inappropriate for ligature based recognition system due to being large in number and huge variations in its character shapes. In its place, statistical features can be easily extracted easy without having to know the structural information of characters within the ligature. In future, experiments should be carried out using traditional machine learning and the recent deep learning methods to improve the overall recognition rate for Urdu Nastalique calligraphic style.

REFERENCES

[1] S. Mori, C. Y. Suen, and K. Yamamoto, "Historical review of OCR research and development," *Proc. IEEE*, vol. 80, no. 7, pp. 1029–1058, Jul. 1992.
[2] S. Naz, K. Hayat, M. I. Razzak, M. W. Anwar, S. A. Madani, and S. U. Khan, "The optical character recognition of Urdu-like cursive scripts," *Pattern Recognit.*, vol. 47, no. 3, pp. 1229–1248, 2014.
[3] H. F. Schantz, *History of OCR, Optical Character Recognition*. Manchester, VT, USA: Recognition Technologies Users Association, 1982.
[4] W. J. Bijleveld and A. J. Van De Toorn, "Process and apparatus for producing and reading Arabic numbers on a record sheet," U.S. Patent 3 527 927, Sep. 8, 1970.
[5] M. Rabi, M. Amrouch, and Z. Mahani, "Recognition of cursive Arabic handwritten text using embedded training based on hidden Markov models," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 32, no. 1, 2018, Art. no. 1860007.
[6] U. Pal and A. Sarkar, "Recognition of printed Urdu script," in *Proc. 7th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2003, pp. 1183–1187.
[7] R. Ahmad, S. Naz, M. Z. Afzal, S. F. Rashid, M. Liwicki, and A. Dengel, "The impact of visual similarities of Arabic-like scripts regarding learning in an OCR system," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 7, Nov. 2017, pp. 15–19.
[8] S. Naz, A. I. Umar, R. Ahmed, M. I. Razzak, S. F. Rashid, and F. Shafait, "Urdu Nasta'liq text recognition using implicit segmentation based on multi-dimensional long short term memory neural networks," *SpringerPlus*, vol. 5, no. 1, pp. 1–16, 2016.
[9] A. Ul-Hasan, S. B. Ahmed, F. Rashid, F. Shafait, and T. M. Breuel, "Offline printed Urdu Nastaleeq script recognition with bidirectional LSTM networks," in *Proc. 12th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2013, pp. 1061–1065.
[10] S. Naz, K. Hayat, M. I. Razzak, M. W. Anwar, and H. Akbar, "Arabic script based language character recognition: Nasta'liq vs Naskh analysis," in *Proc. World Congr. Comput. Inf. Technol. (WCCIT)*, Jun. 2013, pp. 1–7.
[11] R. Ahmad, M. Z. Afzal, S. F. Rashid, M. Liwicki, T. Breuel, and A. Dengel, "KPTI: Katib's pashto text imagebase and deep learning benchmark," in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Shenzhen, China, Oct. 2016, pp. 453–458.
[12] S. B. Ahmed, S. Naz, M. I. Razzak, S. F. Rashid, M. Z. Afzal, and T. M. Breuel, "Evaluation of cursive and non-cursive scripts using recurrent neural networks," *Neural Comput. Appl.*, vol. 27, no. 3, pp. 603–613, 2016.
[13] S. Naz, A. I. Umar, S. H. Shirazi, S. B. Ahmed, M. I. Razzak, and I. Siddiqi, "Segmentation techniques for recognition of Arabic-like scripts: A comprehensive survey," *Educ. Inf. Technol.*, vol. 21, no. 5, pp. 1225–1241, 2016.
[14] N. Sabbour and F. Shafait, "A segmentation-free approach to Arabic and Urdu OCR," *Proc. SPIE*, vol. 8658, p. 86580N, Feb. 2013.
[15] A. Daud, W. Khan, and D. Che, "Urdu language processing: A survey," *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 279–311, 2017.
[16] P. Baker, A. Hardie, T. McEnery, H. Cunningham, and R. Gaizauskas, "EMILLE, A 67-million word corpus of indic languages: Data collection, mark-up and harmonisation," in *Proc. LREC*, 2002, pp. 819–825.
[17] I. U. Khattak, I. Siddiqi, S. Khalid, and C. Djeddi, "Recognition of Urdu ligatures—A holistic approach," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 71–75.
[18] F. Shafait, A. ul-Hasan, D. Keysers, and T. M. Breuel, "Layout analysis of Urdu document images," in *Proc. 10th Int. Multitopic Conf. (INMIC)*, Dec. 2006, pp. 293–298.
[19] M. Ijaz and S. Hussain, "Corpus based Urdu lexicon development," in *Proc. Conf. Lang. Technol. (CLT)*, vol. 73. Peshawar, Pakistan: Univ. Peshawar, 2007.
[20] S. Urooj, S. Hussain, F. Adeeba, F. Jabeen, and R. Parveen, "CLE Urdu digest corpus," in *Proc. Conf. Lang. Technol.*, vol. 47, pp. 47–53, 2012.
[21] F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, and J. Hennebert, "A new Arabic printed text image database and evaluation protocols," in *Proc. 10th Int. Conf. Document Anal. Recognit. (ICDAR)*, Jul. 2009, pp. 946–950.
[22] S. B. Ahmed, S. Naz, S. Swati, and M. I. Razzak, "Handwritten Urdu character recognition using one-dimensional BLSTM classifier," *Neural Comput. Appl.*, 2017, doi: 10.1007/s00521-017-3146-x.
[23] D. A. Satti, "Offline Urdu Nastaliq OCR for printed text using analytical approach," Quaid-i-Azam Univ., Islamabad, Pakistan, 2013.
[24] M. S. Khorsheed, "Off-line Arabic character recognition—A review," *Pattern Anal. Appl.*, vol. 5, no. 1, pp. 31–45, 2002.
[25] L. M. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 712–724, May 2006.
[26] M. Kumar, M. K. Jindal, and R. K. Sharma, "Review on OCR for handwritten Indian scripts character recognition," in *Advances in Digital Image Processing and Information Technology*. Springer, 2011, pp. 268–276.
[27] S. Impedovo, L. Ottaviano, and S. Occhinegro, "Optical character recognition—A survey," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 5, no. 2, pp. 1–24, 1991.
[28] N. Islam, Z. Islam, and N. Noor, "A survey on optical character recognition system," *J. Inf. Commun. Technol.*, vol. 10, no. 2, pp. 1–4. 2016.
[29] R. Mehran, H. Pirsiavash, and F. Razzazi, "A front-end OCR for omnifont Persian/Arabic cursive printed documents," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2005, p. 56.
[30] S. Hussain, S. Ali, and Q. ul Ain Akram, "Nastalique segmentation-based approach for Urdu OCR," *Int. J. Document Anal. Recognit.*, vol. 18, no. 4, pp. 357–374, 2015.
[31] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, "Segmentation free Nastalique Urdu OCR," *World Acad. Sci., Eng. Technol.*, vol. 46, pp. 456–461, Oct. 2010.
[32] I. Shamsher, Z. Ahmad, J. K. Orakzai, and A. Adnan, "OCR for printed Urdu script using feed forward neural network," in *Proc. World Acad. Sci., Eng. Technol.*, vol. 34, 2007, pp. 172–175.
[33] Q. U. A. Akram, S. Hussain, and Z. Habib, "Font size independent OCR for Noori Nastaleeq," in *Proc. Graduate Colloq. Comput. Sci. (GCCS)*, Lahore, Pakistan, vol. 1, 2010.
[34] T. Nawaz, S. Naqvi, H. ur Rehman, and A. Faiz, "Optical character recognition system for Urdu (Naskh font) using pattern matching technique," *Int. J. Image Process.*, vol. 3, no. 3, pp. 92–103, 2009.
[35] J. Tariq, U. Nauman, and M. U. Naru, "Softconverter: A novel approach to construct OCR for printed Urdu isolated characters," in *Proc. 2nd Int. Conf. Comput. Eng. Technol. (ICCET)*, Apr. 2010, pp. V3-495–V3-498.
[36] K. Khan, R. Ullah, N. A. Khan, and K. Naveed, "Urdu character recognition using principal component analysis," *Int. J. Comput. Appl.*, vol. 60, no. 11, pp. 1–4, 2012.

[37] K. Khan, R. U. Khan, A. Alkhalifah, and N. Ahmad, "Urdu text classification using decision trees," in *Proc. 12th Int. Conf. High-Capacity Opt. Netw. Enabling/Emerg. Technol. (HONET)*, Dec. 2015, pp. 1–4.

[38] D. B. Megherbi, S. M. Lodhi, and A. J. Boulenouar, "Two-stage neural-network-based technique for Urdu character two-dimensional shape representation, classification, and recognition," *Proc. SPIE*, vol. 4390, pp. 84–96, Mar. 2001.

[39] Z. Ahmad, J. K. Orakzai, and I. Shamsher, "Urdu compound character recognition using feed forward neural networks," in *Proc. 2nd Int. Conf. Comput. Sci. Inf. Technol. (ICCSIT)*, Aug. 2009, pp. 457–462.

[40] Z. Ahmad, J. K. Orakzai, I. Shamsher, and A. Adnan, "Urdu Nastaleeq optical character recognition," in *Proc. World Acad. Sci., Eng. Technol.*, vol. 26, 2007, pp. 249–252.

[41] S. Abdul, S. Shams-ul, H. Mahmood, and K. Pathan, "A finite state model for Urdu Nastalique optical character recognition," *Int. J. Comput. Sci. Netw. Secur.*, vol. 9, no. 9, p. 116, 2009.

[42] S. A. Hussain, S. Zaman, and M. Ayub, "A self organizing map based Urdu Nasakh character recognition," in *Proc. Int. Conf. Emerg. Technol. (ICET)*, Oct. 2009, pp. 267–273.

[43] S. T. Javed, "Investigation into a segmentation based OCR for the Nastaleeq writing system," M.S. thesis, Nat. Univ. Comput. Emerg. Sci., Lahore, Pakistan, 2007.

[44] F. Iqbal, A. Latif, N. Kanwal, and T. Altaf, "Conversion of Urdu Nastaliq to roman Urdu using OCR," in *Proc. 4th Int. Conf. Interact. Sci. (ICIS)*, Aug. 2011, pp. 19–22.

[45] Y. Khan and C. Nagar, "Recognize handwritten Urdu script using kohenen SOM algorithm," *Int. J. Ocean Syst. Eng.*, vol. 2, no. 1, pp. 57–61, 2012.

[46] R. Patel and M. Thakkar, "Handwritten Nastaleeq script recognition with BLSTM-CTC and ANFIS method," *Int. J. Comput. Trends Technol.*, vol. 11, no. 3, pp. 131–136, 2014.

[47] S. Naz *et al.*, "Offline cursive Urdu-Nastaliq script recognition using multidimensional recurrent neural networks," *Neurocomputing*, vol. 177, pp. 228–241, Feb. 2016.

[48] S. Naz *et al.*, "Urdu Nastaliq recognition using convolutional–recursive deep learning," *Neurocomputing*, vol. 243, pp. 80–87, Jun. 2017.

[49] S. Naz, A. I. Umar, R. Ahmad, S. B. Ahmed, S. H. Shirazi, and M. I. Razzak, "Urdu Nasta'liq text recognition system based on multi-dimensional recurrent neural network and statistical features," *Neural Comput. Appl.*, vol. 28, no. 2, pp. 219–231, 2017.

[50] S. Chanda and U. Pal, "English, devnagari and Urdu text identification," in *Proc. Int. Conf. Document Anal. Recognit.*, 2005, pp. 538–545.

[51] S. A. Sattar, S. Haque, and M. K. Pathan, "Nastaliq optical character recognition," in *Proc. 46th Annu. Southeast Regional Conf.*, 2008, pp. 329–331.

[52] Q. ul Ain Akram, S. Hussain, A. Niazi, U. Anjum, and F. Irfan, "Adapting Tesseract for complex scripts: An example for Urdu Nastalique," in *Proc. 11th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2014, pp. 191–195.

[53] W. Q. Khan and R. Q. Khan, "Urdu optical character recognition technique using point feature matching; A generic approach," in *Proc. Int. Conf. Inf. Commun. Technol. (ICICT)*, Dec. 2015, pp. 1–7.

[54] O. Mukhtar, S. Setlur, and V. Govindaraju, "Experiments on Urdu text recognition," in *Guide to OCR for Indic Scripts*. London, U.K.: Springer, 2009, pp. 163–171.

[55] G. S. Lehal and A. Rana, "Recognition of Nastalique Urdu ligatures," in *Proc. 4th Int. Workshop Multilingual OCR*, 2013, Art. no. 7.

[56] A. El-Korashy and F. Shafait, "Search space reduction for holistic ligature recognition in Urdu Nastalique script," in *Proc. 12th Int. Conf. Document Anal. Recognit. (ICDAR)*, Washington, DC, USA, Aug. 2013, pp. 1125–1129.

[57] S. T. Javed and S. Hussain, "Segmentation based Urdu Nastalique OCR," in *Proc. Iberoamerican Congr. Pattern Recognit.* Berlin, Germany: Springer, 2013, pp. 41–49.

[58] S. Nazir and A. Javed, "Diacritics recognition based Urdu Nastalique OCR system," *Nucleus*, vol. 51, no. 3, pp. 361–367, 2014.

[59] A. Rana and G. S. Lehal, "Offline Urdu OCR using ligature based segmentation for Nastaliq script," *Indian J. Sci. Technol.*, vol. 8, no. 35, pp. 1–9, 2015.

[60] S. A. Husain, "A multi-tier holistic approach for Urdu Nastaliq recognition," in *Proc. 6th Int. Multitopic Conf. (INMIC)*, Feb. 2002, p. 84.

[61] E. R. Q. Khan and E. W. Q. Khan, "Urdu optical character recognition technique for Jameel Noori Nastaleeq script," *J. Independent Stud. Res.*, vol. 13, no. 1, pp. 81–86, 2015.

[62] N. Shahzad, B. Paulson, and T. Hammond, "Urdu Qaeda: Recognition system for isolated Urdu characters," in *Proc. IUI Workshop Sketch Recognit.*, Sanibel, FL, USA, 2009, pp. 1–5.

[63] M. I. Razzak, F. Anwar, S. A. Husain, 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.

[64] Q.-T.-A. Safdar and K. U. Khan, "Online Urdu handwritten character recognition: Initial half form single stroke characters," in *Proc. 12th Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2014, pp. 292–297.

[65] Z. Jan, M. Shabir, M. A. Khan, A. Ali, and M. Muzammal, "Online Urdu handwriting recognition system using geometric invariant features," *Nucleus*, vol. 53, no. 2, pp. 89–98, 2016.

[66] S. A. Husain, A. Sajjad, and F. Anwar, "Online Urdu character recognition system," in *Proc. IAPR Conf. Mach. Vis. Appl. (MVA)*, Tokyo, Japan, 2007, pp. 98–101.

[67] S. Sardar and A. Wahab, "Optical character recognition system for Urdu," in *Proc. Int. Conf. Inf. Emerg. Technol. (ICIET)*, Jun. 2010, pp. 1–5.

[68] D. J. Ittner, D. D. Lewis, and D. D. Ahn, "Text categorization of low quality images," in *Proc. Symp. Document Anal. Inf. Retr.*, 1995, pp. 301–315.

[69] N. Arica and F. T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 2, pp. 216–233, May 2001.

[70] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *J. Electron. Imag.*, vol. 13, no. 1, pp. 146–166, 2004.

[71] F. Shafait, D. Keysers, and T. M. Breuel, "Efficient implementation of local adaptive thresholding techniques using integral images," *Proc. SPIE*, vol. 6815, p. 681510, Jan. 2008.

[72] S. Mir, S. Zaman, and M. W. Anwar, "Printed Urdu Nastalique script recognition using analytical approach," in *Proc. 13th Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2015, pp. 334–340.

[73] A. Rehman, D. Mohamad, and G. Sulong, "Implicit vs explicit based script segmentation and recognition: A performance comparison on benchmark database," *Int. J. Open Problems Compt. Math*, vol. 2, no. 3, pp. 352–364, 2009.

[74] G. S. Lehal, "Ligature segmentation for Urdu OCR," in *Proc. 12th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2013, pp. 1130–1134.

[75] I. Ahmad, X. Wang, R. Li, M. Ahmed, and R. Ullah, "Line and ligature segmentation of Urdu Nastaleeq text," *IEEE Access*, vol. 5, pp. 10924–10940, 2017.

[76] I. U. Din, I. Siddiqi, S. Khalid, and T. Azam, "Segmentation-free optical character recognition for printed Urdu text," *EURASIP J. Image Video Process.*, vol. 2017, p. 62, Dec. 2017.

[77] T. Ali, T. Ahmad, and M. Imran, "UOCR: A ligature based approach for an Urdu OCR system," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2016, pp. 388–394.

[78] S. Shabbir and I. Siddiqi, "Optical character recognition system for Urdu words in Nastaliq font," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 5, pp. 567–576, 2016.

[79] A. F. Ganai and A. Koul, "Projection profile based ligature segmentation of Nastaleeq Urdu OCR," in *Proc. 4th Int. Symp. Comput. Bus. Intell. (ISCBI)*, Sep. 2016, pp. 170–175.

[80] Ø. D. Trier, A. K. Jain, and T. Taxt, "Feature extraction methods for character recognition—A survey," *Pattern Recognit.*, vol. 29, no. 4, pp. 641–662, 1996.

[81] N. H. Khan, A. Adnan, and S. Basar, "Urdu ligature recognition using multi-level agglomerative hierarchical clustering," *Cluster Comput.*, 2017, doi: 10.1007/s10586-017-0916-2.

[82] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, pp. 1–9.

[83] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[84] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Adv. Inf. Process. Syst.*, 2005, pp. 1329–1336.

[85] H.-C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[86] V. K. Govindan and A. P. Shivaprasad, "Character recognition—A review," *Pattern Recognit.*, vol. 23, no. 7, pp. 671–683, 1990.
[87] N. Sharma, T. Patnaik, and B. Kumar, "Recognition for handwritten English letters: A review," *Int. J. Eng. Innov. Technol.*, vol. 2, no. 7, pp. 318–321, 2013.
[88] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Dept. Comput. Sci., Nat. Taiwan Univ., Taipei, Taiwan, Tech. Rep., 2003.
[89] J. McCarthy, "What is artificial intelligence?" Dept. Comput. Sci., Stanford Univ., Tech. Rep., 1998.
[90] S. Lloyd, M. Mohseni, and P. Rebentrost. (2013). "Quantum algorithms for supervised and unsupervised machine learning." [Online]. Available: https://arxiv.org/abs/1307.0411
[91] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, pp. 3–24, 2007.
[92] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.
[93] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[94] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.
[95] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.
[96] S. Naz, S. B. Ahmed, R. Ahmad, and M. I. Razzak, "Zoning features and 2DLSTM for urdu text-line recognition," *Procedia Comput. Sci.*, vol. 96, pp. 16–22, Oct. 2016.
[97] I. Ahmad, X. Wang, R. Li, and S. Rasheed, "Offline Urdu Nastaleeq optical character recognition based on stacked denoising autoencoder," *China Commun.*, vol. 14, no. 1, pp. 146–157, 2017.
[98] I. Ahmad, X. Wang, Y. H. Mao, G. Liu, H. Ahmad, and R. Ullah, "Ligature based Urdu Nastaleeq sentence recognition using gated bidirectional long short term memory," *Cluster Comput.*, 2017, doi: 10.1007/s10586-017-0990-5.
[99] S. T. Javed and S. Hussain, "Improving Nastalique specific pre-recognition process for Urdu OCR," in *Proc. 13th Int. Multitopic Conf. (INMIC)*, Dec. 2009, pp. 1–6.
[100] S. Naz, A. I. Umar, S. B. Ahmed, S. H. Shirazi, M. I. Razzak, and I. Siddiqi, "An OCR system for printed Nasta'liq script: A segmentation based approach," in *Proc. 17th Int. Multi-Topic Conf. (INMIC)*, Dec. 2014, pp. 255–259.
[101] S. Naz, S. B. Ahmed, R. Ahmad, and M. I. Razzak, "Arabic script based digit recognition systems," in *Proc. Int. Conf. Recent Adv. Comput. Syst. (RACS)*, 2016, pp. 67–73.
[102] I. A. Ansari and R. Y. Borse, "Automatic recognition of offline handwritten Urdu digits In unconstrained environment using daubechies wavelet transforms," *IOSR J. Eng.*, vol. 3, no. 9, pp. 50–56, 2013.
[103] D. S. Kaushal, Y. Khan, and S. Varma, "Handwritten Urdu character recognition using Zernike MI's feature extraction and support vector machine classifier," *Int. J. Res.*, vol. 1, no. 7, pp. 1084–1089, 2014.
[104] S. Uddin, M. Sarim, A. B. Shaikh, and S. K. Raffat, "Offline Urdu numeral recognition using non-negative matrix factorization," *Res. J. Recent Sci.*, vol. 3, no. 11, pp. 98–102, 2014.

NAILA HABIB KHAN received the B.S. degree in computer science from the Institute of Management Sciences, Peshawar, Pakistan, in 2011, and the M.S. degree in information technology in 2014. She is a double Gold Medalist and has been awarded numerous merit scholarships during her academic career. She is currently a Ph.D. Scholar with the Institute of Management Sciences, where she is also a Research Assistant under the supervision of Dr. A. Adnan. Her areas of interest are document image understanding, pattern recognition, image segmentation, and multimedia.

AWAIS ADNAN is currently an Assistant Professor and the Director ORIC (Office for Research Innovation and Commercialization) with the Institute of Management Sciences, Peshawar. He teaches different courses at undergraduate, graduate, and post-graduate level. He also supervise students at MS-IT, MS-CS, and BS level. He has also been a Trainer at HRDC where he gives training on computer packages and data analysis tools to professionals from various government and public-sector organizations. Major areas of his research are multimedia, digital image processing, and network-on-chip.