Optical Character Recognition System for Urdu Words in Nastaliq Font

Safia Shabbir* and Imran Siddiqi*
*Bahria University, Islamabad, Pakistan

***Abstract*—Optical Character Recognition (OCR) has been an attractive research area for the last three decades and mature OCR systems reporting near to 100% recognition rates are available for many scripts/languages today. Despite these developments, research on recognition of text in many languages is still in its early days, Urdu being one of them. The limited existing literature on Urdu OCR is either limited to isolated characters or considers limited vocabularies in fixed font sizes. This research presents a segmentation free and size invariant technique for recognition of Urdu words in Nastaliq font using ligatures as units of recognition. Ligatures, separated into primary ligatures and diacritics, are recognized using right-to-left HMMs. Diacritics are then associated with the main body using position information and the resulting ligatures are validated using a dictionary. The system evaluated on Urdu words realized promising recognition rates at ligature and word levels.**

Keywords*—Optical Character Recognition; Urdu Text; Ligatures; Hidden Markov Models; Clustering

I. Introduction

Today, most of the information is available in digital form and can be accessed within a span of few clicks. This has resulted in a tendency to digitize the existing paper documents like books, newspapers, official notes etc. and make them available online. A number of libraries across the world have also scanned their books providing online access to the readers. These scanned versions of printed documents can be consulted more easily and efficiently as opposed to the paper format. However, mere scanning of huge amounts of printed papers is not sufficient. The job is only half done if these scanned documents are not searchable. Manual transcription of these huge collections of paper documents is naturally a tiresome, time consuming and very inefficient solution. This attracted researchers to develop automatic Optical Character Recognition systems which take scanned document images as input, apply image processing and pattern classification techniques and convert the image into text which is not only searchable and editable but also requires significantly lesser storage space as opposed to their image format.

Formally, Optical Character Recognition (OCR) is defined as a technique in which scanned image of (handwritten or printed) document is processed through machine, characters are recognized and extracted and rendered in a word editor [1], [2].

The most useful application of an OCR system is the concept of digital libraries where huge collections of books could be converted to text and made available online. Other typical applications include automatic processing of bank cheques, computerized validation of identification documents (passports, driving licenses etc.), processing of utility bills, text to speech applications, vehicle registration number recognition and text guided navigation of autonomous vehicles or visually impaired individuals.

Research in Optical Character Recognition is as old as the computerized document analysis and recognition itself. The ultimate objective of most of the document recognition problems is indeed a comprehensive OCR system which could eventually convert the huge collections of existing paper documents with all sorts of variations to digital form. The last few decades have witnessed extensive research on Optical Character Recognition systems for the Roman script. Today, commercial OCR software are available reporting near to 100% recognition rates for languages based on the Roman script. Research on Chinese and Arabic OCRs is also quite mature with acceptably good recognition rates. Few Multilingual OCRs have also been developed with the aim to propose techniques which are general in nature [3]. Despite these tremendous developments, OCRs for many languages around the globe are either non-existent or are witnessing early days of research, Urdu being one of them and makes the subject of our study. Research on Urdu and similar cursive scripts like Pashto, Farsi etc. is still in its early days with limited literature available till date.

This paper presents a segmentation free optical character recognition system for printed Urdu text in Nastaliq font. The proposed methodology relies on ligatures as units of recognition and is based on a semi-automatic clustering scheme which extracts ligatures from a given data set and clusters different instances of the same ligature into classes. Recognition is carried out using Hidden Markov Models (HMM) where a separate HMM is trained for each ligature (cluster). The recognition of ligatures is first carried out without dots which are later associated with the ligatures to recognize the complete word. Unlike most of the traditional approaches which either work on isolated Urdu characters or a fixed font size, the proposed technique works on complete Urdu words and is scale invariant. The sequential clustering employed for generation of training data makes the framework scalable which allows extension of the system to consider an increased number of ligatures.

The next section discusses the recognition techniques proposed for Urdu and similar scripts in the recent years along with a comparative analysis of these methods. Section III details the proposed methodology including training and recognition of Urdu words. Experimental evaluations carried out to validate the proposed methodology are presented in Section IV while Section V concludes the paper with a discussion on future research directions on this subject.

II. Background

This section presents an overview of notable contributions towards the development of an OCR for cursive scripts like Arabic, Farsi and Urdu etc. OCR systems for such scripts generally follow one of the two approaches: segmentation-based or segmentation-free. Another categorization of these OCRs is based on units of text used for recognition. Some of the techniques work on isolated characters only [2], [4], [5], [6] while others work on complete words or ligatures [7], [8], [9], [10], [11]. The systems developed to work with isolated characters naturally report much better results as opposed to those working on words or ligatures. Since the segmentation of Urdu and Urdu-like text images into its basic units (words or characters) is itself a challenging task, a significant research dedicated to segmentation of text has also been reported in the literature [12]. With the increase in the usage of tablets and other hand held devices, a new categorization of OCRs as offline or online OCR has also emerged. Online OCRs recognize the text on the fly as it is input by a user while offline OCRs work on the scanned images of text. Online OCRs have the advantage of having additional information on the sequence of strokes while offline OCRs have only the shape information in the form of text pixels making them more challenging.

Segmentation-based approaches for OCR work on individual characters which are extracted by segmenting the text into ligatures and further into characters. The main advantage of these approaches is that the number of classes to be recognized is the same as number of characters (and their different shapes). This number is much smaller when compared to the number of ligatures or words which are units of recognition in segmentation-free approaches. The segmentation of text into characters, however, is a complex and challenging problem in itslef [13], [7], [14].

Among well-known segmentation-based approaches, morphological processing followed by an analysis of contours is applied for character extraction in [15]. Chain codes computed from the character contours are employed as features while classification is carried out through feedback loop. In another study [16], moment invariant descriptors computed from Arabic characters are fed to a multilayer perceptron network for recognition. In [17], authors segment Arabic words into characters and use matching of edge points to recognize characters. Sarfraz et al. [18] employ horizontal and vertical projection profiles for segmentation of text into lines and characters respectively. Features based on moment invariant descriptors are extracted from segmented characters and are used to train a radial-basis function network which learns the different character classes. Authors in [13] present an effective methodology for segmentation of Urdu text into characters. A set of structural features is used for segmentation and many of the common problems causing over and under segmentation have been addressed in this study. Zaheer et al. [7] apply three levels of segmentation to Urdu text, line, word and character segmentation. The segmented characters are recognized using a neural network. Another segmentation based Urdu OCR is proposed in [19] where the authors exploit pixel strength to segment the text into words and subsequently into characters. Using neural network as classifier, the system reports an average recognition rate of 70% on 56 character classes.

Pal et al. [14] present an Urdu OCR which segments lines of text using projection profiles while characters are extracted using heuristics on runs of text and non-text pixels. Features including topological features, water reservoir features and contour features are used to recognize characters using a tree classifier. In another study [20], Arabic characters are segmented using a dynamic window sizing and are recognized using cross correlation. A segmentation based Urdu OCR for Nastaliq font is presented in [8] where the skeletonized text is segmented at branching points and each segment is framed. These segments are then used to train HMMs which are subsequently used for recognition. A similar approach for recognition of Noori Nastalique Urdu text in font size 36 is proposed in [11]. For segmentation, the authors first extract the baseline and categorize the components attached with the baseline as primary components (main body). Thinning of these primary components is then carried out and the stroke junctions in the thinned image are used a segmentation points. Each segment is then framed and DCT based features extracted from these frames are fed to HMMs for training/recognition.

In contrast to segmentaiton-based approaches, the segmentaiton-free approaches perform recognition at ligature or word levels. Segmentation-free techniques tend to be less complex than segmentation-based approaches in the sense that they do not require segmentation of text into individual characters. These methods are relatively easier to implement but the major challenge with these approaches is the larger number of classes to be recognized [21], [22], [4], [23], [24], [25]. This number is the same as the number of unique words (ligatures) in the vocabulary under study.

A segmentation-free approach for recognition of Urdu handwritten words is proposed in [9]. The authors extract gradient and structural features from Urdu words which are recognized using Support Vector Machine with radial basis function (RBF) as kernel. A relatively simple recognition system is presented in [21] where template matching using cross correlation is used to match ligatures. A font independent approach for offline and online Urdu OCR is presented in [10]. Each word is considered a composition of compound components and baseline information is used to associate the secondary components with the respective primary components. Features based on Hu moments are extracted by using four windows of different sizes while recognition is carried out using K-nearest neighbor (KNN)

classifier [10].

In [26], the authors propose a multi-font Arabic and Persian OCR. The technique relies on a series of preprocessing steps including global thresholding, connected component extraction and skew detection and correction. Features extracted from the contours of connected components are used for recognition. A system for recognition of Arabic literal amounts is presented in [27]. The features investigated in this study include number of ascenders, number of descenders and number of loops etc. These structural features are fed to a number of classifiers for recognition.

A system for recognition of isolated Urdu characters is presented in [4]. The character images are binarized and the chain code of each character is saved in an xml file along with the respective class name and character code. For recognition, diacritics are removed and the chain code of query sample is compared with those in the database to find the nearest match [4]. Another effective Urdu OCR system based on Hidden Markov Models (HMM) is presented in [23]. Main body ligatures are separated from diacrtics and a separate HMM is trained on each ligature. Features based on Discrete Cosine Transform (DCT) computed from a sliding window on each ligature are fed to the HMMs for training. During recognition, the query word is separated into diacritics and main body and the features computed from each are fed to the recognizers. A set of rules is defined to associate the diacritics with the main body and recognize the complete ligature. Another segmentation-free recognition system for Urdu ligatures is presented in [24] where a set of shape descriptors is used to characterize the ligatures. A total of 10,000 Urdu ligatures in Nastaliq font and 20,000 Arabic ligatures in the Naskh font are used as training data. Recognition is carried out using k-nearest neighbor scheme to find the best match for the query ligature. The system evaluated on a custom developed UPTI (Urdu Printed Text Image Database) database realized promising recognition rates.

A font size independent Urdu OCR is proposed in [25]. Ligatures are extracted using connected component labeling and the stroke width information is exploited to distinguish primary and secondary (diacritics) ligatures. The secondary ligatures are categorized into four classes while a fifth class comprises all primary ligatures. Recognition is carried out using structural features mainly including end points, turning points, junction points, cross points stroke width and height etc. Shehzad et al. [28] developed a system for recognition of isolated Urdu characters. The technique rests on a set of primary and secondary stroke features. The primary stroke features include length of bounding box diagonal, angle of bounding box diagonal, and total length of primary stroke etc. while the secondary stroke features comprise the number of secondary strokes, total length of secondary strokes, positioning of the secondary strokes, and number of dots in secondary strokes. Among other ligature based recognition systems template matching [29] and Fourier descriptors have also been investigated [30].

The above paragraphs discussed few of the notable contributions towards development of an OCR for Urdu and Urdu-like cursive scripts. This discussion by no means is exhaustive but is intended to provide an overview of the wide variety of approaches that have been proposed in the recent years. Interested readers may consult the detailed survey papers on this subject [31].

After having discussed few of the well-known OCR techniques proposed for Urdu and similar cusrive scripts, we present the proposed recognition methodology in the next section.

III. Proposed Methodology

This section presents in detail the proposed approach for recognition of Urdu text in Nastaliq script. The technique relies on a segmentation-free method which employs ligatures as the basic units of recognition where each ligature comprises one or more characters (connected together using the joiner rules). As discussed earlier, segmentation-free approaches reduce the complexity of the system as segmentation at character level is not required but, on the other hand, it increases the number of classes to be recognized which is same as the number of unique ligatures. We extract the connected components in an image and separate the components into primary and secondary liguatres which correspond to main body and diacritics respectively. Each ligature is represented by a set of features and a clustering is carried out to group ligatures into clusters. These clusters serve as training data and a separate hidden Markov model (HMM) is trained on each ligature. Once the individual ligatures are recognized using these HMMs, the secondary ligatures are associted with the primary ligatures to recognize the complete word. These steps, distinguished into training and recognition phases, are presented in detail in the following.

A. Training

Training involves making the model(s) learn to discriminate between different (ligature) classes. The different steps invovled in this training phases are detailed in the following.

1) Preprocesing: Preprocessing is the first step in the development of any OCR system which prepares images for the subsequent phases. Depending upon the application and the type of input images, preprocessing may involve binarization [32], noise removal and, skew and slant detection and correction [33]. In our study, we intend to work on contemporary images of Urdu text which are not likely to suffer from noise or degradations. The preprocessing in our case, therefore, comprises binarization of image to segment text from the background. In our implementation, we have employed the well-known Otsu‘s global thresholding to binarize the text image. An example grayscale image and its binarized version are illustrated in Figure 1.

2) Extraction of Ligatures: Urdu, being a highly cursive language, makes the segmentation of Urdu text into characters not only challenging but also prone to errors. Urdu words are a composition of ligatures and diacritics. A ligature can be an isolated character or a combination of characters joined together while diacritics are the secondary components. In our study, we use ligatures as the basic recognizable units. Ligatures and diacritics are extracted from binarized words

using connected component labeling. Figure 2 illustrates an example image and the corresponding connected components which are then fed to the next stage of feature extraction.

3) Feature Extraction: Feature extraction is the pivotal stage in any recognition/classification task. Representing shapes (ligatures in our case) by features not only allows reducing the dimension but also allows effective comparison of these ligatures as opposed to the pixel representation. Most of the features previously employed for Urdu OCR, however, work on fixed font sizes. In our methodology, we have chosen scale invariant global transformational features to represent ligatures and diacritics. These features include horizontal projection, vertical projection, upper profile and lower profile. These features have been effectively employed in a number of shape matching (word spotting) problems [34] and have shown promising performances. These features and their computational details are presented in the following.

a) Horizontal Projection: : is the sum of pixel values in each row of the input image. The projection is normalized to the range $[0-1]$ by dividing each value by the width (number of columns) of the input ligature image.

b) Vertical Projection: : is the sum of pixel values in each column of the input image. The sequence values are normalized between 0 and 1 by dividing them by the height (number of rows) of input ligature.

c) Upper Profile: : is calculated by finding, in each column, the distance of the first text pixel from the top of the bounding box of the ligature. Upper profile is then normalized by dividing it by the height (number of rows) of the input ligature.

d) Lower Profile: : is computed by finding, for each column, the distance of the last text pixel from the top of the bounding box of the ligature. Like upper profile, lower profile is also normalized by dividing it by the height of the ligature.

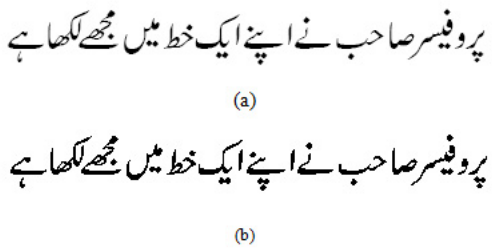


Fig. 1: a) Original image b) Binarized image with global thresholding

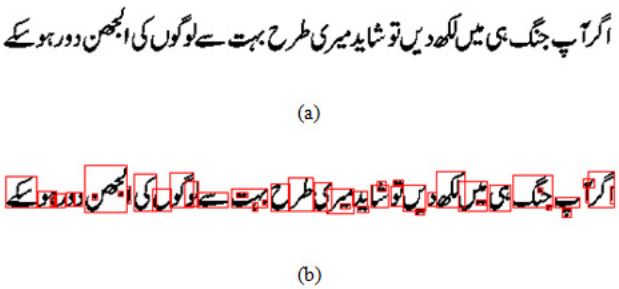


Fig. 2: a) Binarized image b) Connected component labeling

The projection and profile features extracted from an example ligature are illustrated in Figure 3.

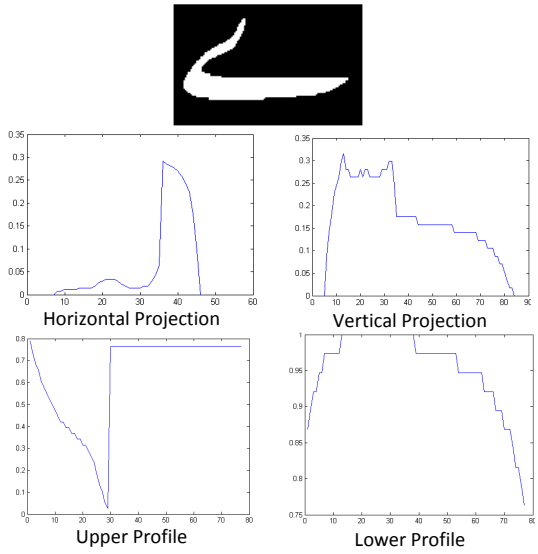


Fig. 3: Projection and Profile features extracted from a ligature 'Bari Yay'

4) Clustering of Ligatures: In order to train the classifiers to recognize ligatures, labeled ligature classes need to be established. Manual generation and labeling of training data, naturally, is a tedious task. We, therefore, chose to employ a semi-automatic scheme where clusters of ligatures are generated from a given set of document images. Each cluster is then labeled and the errors in clustering process are removed manually to ensure that the training data does not contain erroneous clusters. These clusters or classes serve as training data to train the recognizers.

To generate training data for recognizers, we take samples of 30 document images and extract the ligatures (main body as well as dots) as discussed earlier. In order to have scale invariance, some of the documents are resized using scale factors of 0.5, 0.75, 1.25, 1.5 and 2.0. The extracted ligatures from these images are grouped into clusters eventually to be used as training data. For clustering, we have employed a sequential clustering algorithm [35], [36] which does not require a priori the number of clusters. We start by randomly picking a ligature and assuming it to be the mean (representative) of the first class (cluster). For each subsequent ligature, we compute its distance (using Dynamic Time Warping) with the center of each cluster and chose the nearest cluster as a potential candidate. If the distance of the ligature in question to the nearest cluster is below an empirically determined threshold, the ligature is assigned to this cluster and the cluster mean is updated. In case the distance does not fall below the predefined threshold, a new cluster is created with the ligature in question as the mean of

the newly generated cluster. This process is repeated until all the ligatures have been assigned to a cluster.

Naturally, the clustering algorithm used in our study has certain short comings. The most significant of these is that the generated clusters are sensitive to the order in which the ligtaures are presented to the algorithm. However, it should be noted that the objective of clustering is to generate an approximate set of ligature classes which are manually corrected prior to training of recognizers. Hence, the performance of the recognition system is not sensitive to this clustering step.

Executing the mentioned clustering algorithm on the sample images, we get a total of 246 clusters. These clusters, naturally, contain some errors which are corrected manually. After refinement, we come up with a total of 250 clusters. The idea of using a sequential clustering instead of traditional k-means clustering and similar algorithms is not to fix the number of clusters a priori. The 250 ligature clusters, representing the frequent Urdu ligatures have been generated using random Urdu texts. Using a larger collection of text images to generate the clusters will naturally produce a larger number of clusters.

It is to be noted that the total number of valid Urdu ligatures is more than twenty thousand. However, most of these ligatures occur very rarely in the text. Studies carried out by the Center of Language Engineering (CLE) at Lahore, Pakistan, have concluded that most of the Urdu words can be generated using the frequent Urdu ligatures. CLE has also compiled the frequencies of occurrence of Urdu ligatures from a huge corpus of text. In our study, we have extracted the frequencies of occurrence of our 250 ligatures from the statistics compiled by the CLE. These frequencies are sorted in descending order and are illustrated in Figure 4. It can be noticed that using only 250 frequent ligatures can allow recognition of a large number of frequent Urdu words.

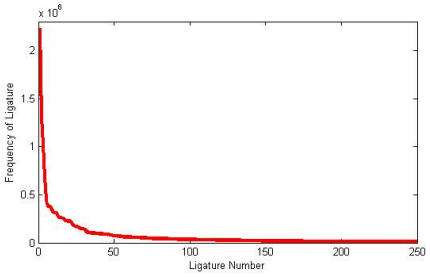


Fig. 4: Frequencies of 250 ligatures

Once the ligature clusters are generated, each cluster needs to be assigned its respective Unicode. Each ligature comprises one or more characters. Once the characters appear as a part of ligature, a number of characters may exhibit the same shape and can only be differentiated by the position and number of dots. Since in our implementation, the ligature classes do not include diacritics (which are treated separately), the number of character classes in the ligatures is less than the actual number of characters. To elaborate this idea Figure 5 illustrates two Urdu ligatures 'Ba' and 'Na'. In the absence of dots, both these ligatures are exactly the same so the characters 'Bay' and 'Noon' belong to the same class as they have exactly the same shape (for this combination). Analyzing different combinations of characters to form ligatures, a total of 20 character classes, listed with their respective codes in Figure 6, are identified. Each ligature has its unique Unicode string depending upon the classes of characters it comprises of. Referring again to the example of ligatures 'Ba' and 'Na', in the absence of dots, both these ligatures are in the same cluster and the Unicode associated with this cluster is the Unicode of 'Bay' + 'Alif' as both 'Bay' and 'Noon' belong to the character class 'Bay'. On the other hand, in isolated form, 'Noon' belongs to its own class.

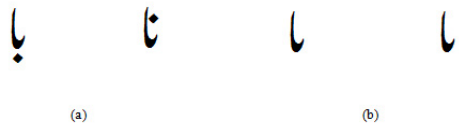


Fig. 5: Images of two ligatures 'ba' and 'na' (a): With dots (b): Without dots

Once the ligatures have been grouped into clusters, we need to choose a classifier which could be trained to recognize these ligatures. Classifiers like artificial neural networks, support vector machine or hidden Markov models (HMM) could be effectively employed for this purpose. In our study, we have used HMMs which have been successfully applied to problems like gesture recognition [37], [38], [39], speech recognition [40], handwriting recognition [41], [42], [43], [44], musical score recognition [45] and optical character recognition [23]. The training of HMMs to model the ligatures is discussed in the following.

5) HMM Training: A separate HMM is trained for each ligature (main body as well as diacritics) considered in our study. All ligatures are resized to a predefined height of 64 pixels and each ligature is scanned from right to left using a frame (sliding window) of size 64×7 with an overlap of 4 pixels (Figure 7). The sensitivity of overall system performance to these parameters is dicussed later in the paper. For each position of window, we extract a set of features which includes the total number of text pixels in the window, sum of horizontal edges, sum of vertical edges, horizontal projection and vertical projection. For edge based features, Sobel edge detection (horizontal/vertical) is applied to the pixels in the window and the numbers of 1s in the output image are counted. The horizontal (vertical) projection is computed by counting the number of pixels in each row (column) of the image (window). These features have been effectively applied to a number of recognition problems including character recognition [46], word spotting [47] and identification of writers [36]; consequently, they are likely to perform well for characterization of ligatures as well.

To train the models, the feature space is quantized to a 75 symbol codebook (since the HMMs are discrete). Right-to-left HMMs with 9 states (Figure 8) are used in our study and a separate HMM is trained for each ligature using

Class	Class Members	Class Unicode
ا	ا	0627
ب	ب, پ, ت, ٹ, ث, ں , ی	0628
ج	ج, چ, ح, خ	062C
د	د, ڈ, ذ	062F
ر	ر, ڑ, ز, ژ	0631
س	س, ش	0633
ص	ص, ض	0635
ط	ط, ظ	0637
ع	ع, غ	0639
ف	ف, ق	0641
ق	ق	0642
ک	ک, گ	06A9
ل	ل	0644
م	م	0645
ن	ن, ں	0646
و	و	0648
ہ	ہ	06C1
ھ	ھ	06BA
ی	ی	06CC
ے	ے	06D2

Fig. 6: Groups of character classes in the absence of diacrtics and the respective Unicodes

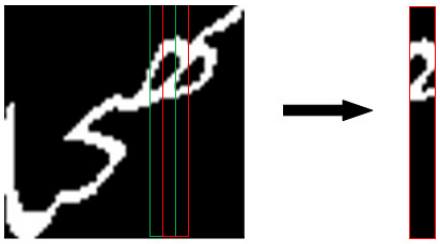


Fig. 7: Sliding Windows a) Overlapping in Sliding Windows b) A single window

the standard Baum-Welch algorithm. Once the models are trained, the Unicode string of each ligature is associated with its respective HMM.

B. Recognition of Words

The basic unit of recognition in or study is ligature. However, since a word represents a semantically meaningful unit, the system is presented with words for recognition. Once a query word is presented to the system, we first segment it into ligatures using connected component labeling. Ligatures are then separated into primary ligatures and diacritics and the position information of diacritics with respect to the primary ligatures is also stored. Each ligature is recognized by finding the HMM which reports the maximum probability when the features of the respective ligature are fed to the trained HMMs. Diacritic information is then associated with the recognized primary ligature to find the true ligature. The Unicode of the complete word is then written to an output file after dictionary matching. The steps involved in the recognition phase are summarized in Figure 9 while each of these steps is discussed in detail in the following sections.

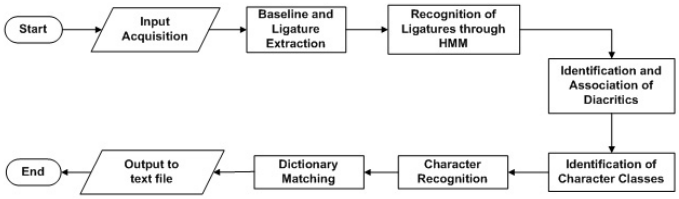


Fig. 9: Flow chart for Recognition of Urdu Words

1) Baseline and Ligature Extraction: The query word presented to the system is binarized using global thresholding and the baseline of the word is determined by finding the row with the maximum number of text pixels. Figure 10 illustrates a sample input word and its baseline. To extract the ligatures of a given word, connected component labeling is employed similar to the training phase. The extracted ligatures are then recognized using the trained models.

Fig. 10: a) Input word 'Aiwan' b) Baseline detection

2) Recognition of Ligatures: Ligatures extracted from the query word are fed to the trained HMMs. Each ligature is recognized by the HMM which produces the maximum probability, the respective Unicode being the output. It should be noted that since the main body of the ligature is separate from the diacritics in the clusters of ligatures, the output of the HMM is actually the Unicode of the respective character class (as per Figure 6) rather than the true Unicode.

3) Association of Diacritics with Ligatures: Diacritics are differentiated from ligatures through their Unicodes. Diacritics are categorized depending upon their position with respect to baseline of word. When a diacritic is above the baseline it is categorized as 'above' and, if it is below the base line it is identified as a 'below' diacritic. In cases where the diacritic resides on the baseline we use the height information

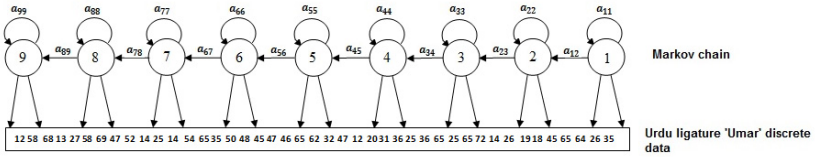


Fig. 8: Markov chain model generated for an example ligature

TABLE I: Diacritics with corresponding integer values and descriptions

Diacritic	Value	Description
No dots	-1	No dots
One dot above	1	one dot is above the baseline
One dot below	2	one dot is below the baseline
Two dots above	3	two dots are above the baseline
Two dots below	4	two dots are below the baseline
Three dots above	5	three dots are above the baseline
Three dots below	6	three dots are below the baseline

to determine whether more part of the diacritic is above or below the baseline. Based on the number and position of diacritics, an integer value is assigned to it as per Table I.

Within a word, diacritics are associated with ligatures depending upon their position information with respect to the ligature. Each diacritic is analyzed for each ligature and, if it lies within the width of ligature then it is associated to the respective ligature. In case of more than one candidate ligature, percentage of diacritic width overlap with each ligature is computed and the diacritic is associated with ligature which has maximum overlap.

Once the diacritics are associated with a main body ligature, they have to be further associated with the characters within a ligature. In some cases, multiple diacritics within a single main ligature can generate a large number of possible associations. This is illustrated in Figure 11 where we have ligature classes 'Bay' + 'Bay' + 'Alif'. The diacritic information associated with this ligature is 'one dot above' and 'one dot below'. This could result in two different ligatures 'Bana' and 'Naba'. In all such cases, the possible combinations of ligatures are matched with a valid ligature dictionary and the first valid instance is picked as the recognized ligature.

Fig. 11: (a) Ligature 'Bay' + 'Bay' + 'Alif' with 'one dot above' and 'one dot below' (b) possible words 'Bana' and 'Naba'

Once the individual ligatures in a word are recognized, the Unicodes of these ligatures are combined to generate the Unicode string for the query word. The Unicode string is written to a text file using UTF-8 encoding to save the output.

This concludes our discussion on the proposed recognition methodology. In the next section we present the experiments carried out to validate the proposed technique.

IV. Experiments and Results

This section presents the results of the experiments carried out to evaluate the effectiveness of the proposed methodology. We first present the data set used in different experiments followed by a discussion on the clustering of ligatures into classes. We then present the recognition rates reported by the system at ligature and word levels followed by some interesting analytical experiments.

A. Dataset

For generating the clusters of ligatures we used 30 full page length scanned images of Urdu documents from the CLE database (http://www.cle.org.pk). In order to have scale invariance, the original images were scaled by different factors as discussed earlier. The system was evaluated on 100 query words with a total of 351 ligatures. For comparison purposes, ligature recognition rate was also reported on 2,017 high frequency ligatures from the CLE database.

B. Performance of Ligature Clustering

As stated earlier, we employ a semi-automatic sequential clustering to generate ligature classes. The 30 document images used for clustering comprised a total of 10,364 ligatures out of which 8,800 were correctly categorized into 246 clusters realizing an accuracy of 85%. Since the clusters had to be used as training data, the errors in clusters were manually corrected after visual inspection making a total of 250 clusters.

C. Performance of Recognition

To evaluate the recognition performance we performed experiments with 100 query words. Recognition rate is computed at ligature as well as word level. Each of these is discussed in the following.

1) Ligature Recognition Rate: The 100 query words comprise a total of 351 ligatures which include isolated characters, two, three, four, and five character ligatures and diacritics. Figure 12 illustrates some example query words used in our study. For recognition, features extracted by sliding window on the query ligature are fed to all the (trained) HMMs. The HMM reporting the maximum probability identifies the ligature. Table II summarizes the recognition rates on each of these categories of ligatures. Over all the system correctly

recognizes 335 out of 351 ligatures realizing an overall ligature recognition rate of 95%.

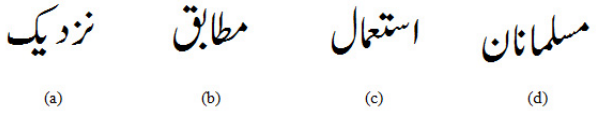


Fig. 12: Examples of query words

2) Word Recognition Rate: Once the ligatures are recognized, we apply the post processing steps to associate diacritics with their respective ligatures and recognize the complete word. This step naturally is error prone as association of diacritics with words is based on a set of heuristics and, in some cases, this association is not very straight forward. In addition, error in any one of the ligatures of a query word is considered an error at word level. Using the diacrtic association rules discussed earlier, the system is able to correctly recognize 89 out of 100 query words.

Comparing a word recognition rate of 89% with the results reported in the literature, it is important to mention that the high recognition rates reported in the literature are either on isolated characters or on ligatures. Some studies, which work at word level, totally ignore the diacritic information resulting in relatively high recognition rates. In our study, we consider the query words without any constraints to resemble the real world scenarios as closely as possible. A recognition rate of 89% at word leve, therefore, is very promising.

In order to compare the performance of our system with existing Urdu OCR techniques, we also perform a series of experiments on the CLE ligature database as discussed in the next section.

D. Performance on CLE High Frequency Ligature (HFL) Dataset

The experimental results discussed in the above sections are based on a total of 250 clusters of ligatures (independent of the font size). For comparison purposes and in order to study how the performance of the recognizer varies as the number of ligatures increases, we also train HMMs on the frequent ligature dataset of CLE. We compute the ligature recognition rate as a function of the number of ligatures by varying the number of ligatures from 50 to 2,017. The results of these evaluations are summarized in Figure 13 where it can be seen that there is a natural gradual decrease in the recognition rates as the number of ligatures rises. A ligature recognition rate of 92% on 2,017 ligatures is not only promising but it also demonstrates the scalability of the proposed recognition scheme. A comparison of our recognition rates with some notable studies on Urdu OCR is summarized in Table III. It can be seen that our results are comparable with those obtained by [23] on the same data set. The major drawback of this dataset, however, is the fixed font size of all the ligatures; a constraint which is hard to meet in real world problems. Other studies listed in Table III either work on isolated characters or ignore the diacritic information hence the high recognition rates realized in these studies may not be very signficiant.

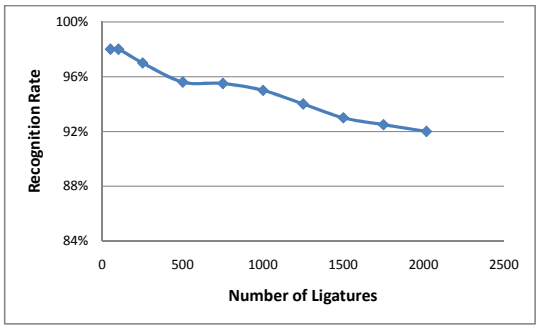


Fig. 13: Recognition rate as a function of number of ligatures (CLE Database)

In addition to the number of ligatures, we also study the sensitivity of the recognition rate to other parameters of the system. These include the number of states in the HMMs and the cell size for framing of ligatures. These experiments are carried out on the first 500 ligatures of the CLE high frequency ligature (HFL) database and the realized recognition rates are summarized in Figure 14. It can be observed that the recognition rates increase with the increase in number of states in the HMM and begin to stabilize from 9 states onwards. The recognition rates seem to be more sensitive to the frame size employed during feature extraction. For all experiments, for a frame width of n pixels, there is an overlap of $(n+1)/2$ pixels. Smaller frame sizes yield higher recognition rates which drop as the frame width increases. This observation is natural as larger windows include larger proportions of ligatures which may not be common across multiple samples of the same ligature. Smaller frame widths result in a larger number of windows per ligature and the extracted features are more effective in characterizing these ligatures.

V. Conclusion

This work presented a segmentation free Urdu OCR for printed text in Nastaliq font. The proposed technique considers ligatures as the basic units of recognition. Ligatures are extracted by performing connected component labeling on the binarized document images of Urdu text. A total of 250 ligature clusters are generated which serve as training data for ligature modeling. A separate right-to-left Hidden Markov Model (HMM) is trained for each of the ligatures (main body as well as diacritics) using features extracted by a sliding window. For recognition, ligatures in the query word are extracted and each ligature is recognized using the trained HMMs. Diacritics and main body ligatures are separately recognized. Diacritics are then associated with respective ligatures using the position information and the complete ligature is recognized using dictionary validation. Finally, the Unicode string of the word is written to a text file.

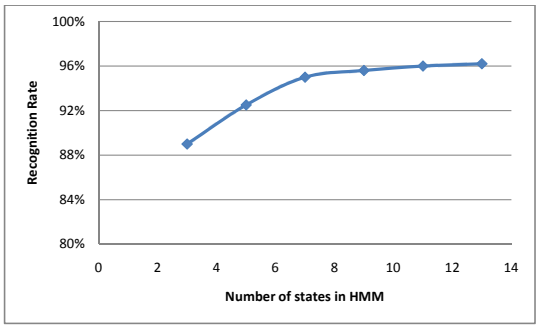
The proposed system realizes very promising ligature recognition rates. The relatively low word recognition rate is

TABLE II: Types of ligatures and corresponding recognition rates

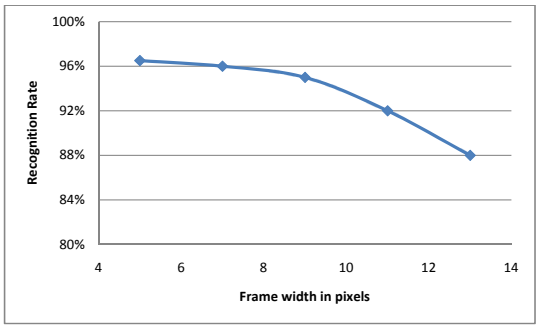
Type of ligature	Total Ligatures	Correctly Recognized	Recog. Rate
Isolated characters	117	112	96%
Two character ligatures	99	92	93%
Three character ligatures	22	20	91%
Four character ligatures	6	5	83%
Five character ligatures	4	4	100%
Diacritics	103	102	99%

TABLE III: Notable studies on Urdu OCR with recognition rates and limitations

Study	Dataset	Recognition Rate	Limitations
Z. Ahmad et al. [7]	Synthetic and real-world images of Urdu	93.4%	Assumes ligatures to be diacritic free
T. Nawaz et al. [4]	Isolated characters in different font sizes	89%	Tested only on isolated characters
N. Sabbour & F. Shafait [24]	UPTI Online Arabic e-book	Urdu:99% Arabic: 86%	Ignores diacritic information
D. Satti [25]	29,517 ligatures	97.10%	Ignores secondary components and ligature arrangement
Javed et al. [23]	1282 unique ligatures	92%	Fixed size ligatures only
Proposed Method	100 words 2,017 ligatures	89% 92%	Complete words, scale invariant Same dataset as in [23]



(a)



(b)

Fig. 14: Recognition rates on 500 ligatures as a function of: (a) Number of states in the HMM (Frame width fixed to 7) (b) Frame width for feature extraction (Number of states fixed to 9)

understandable due to the complexities involved in associating diacritics with the main body ligatures. It should also be noted that contrary to most of the existing Urdu OCRs, our approach is scale invariant and is not tuned to recognize words in a particular font size. The most obvious extension of the proposed system is to increase the number of ligatures to cover a major proportion of all Urdu words. The diacritics handled in our study include different numbers and positions of dots. Other diacritics like 'choti toye', 'shad', 'mad', 'zabar', 'zair' etc. can also be incorporated in the system. Real word Urdu documents may suffer from problems like noise, degradation and skew and consequently will require a preprocessing step prior to recognition.

REFERENCES

[1] F. Iqbal, A. latif, N. Kanwal, and T. Altaf, "Conversion of urdu nasstliq to roman urdu using ocr," in *4th International Conference on Interaction Sciences (ICIS)*, 2011, pp. 19–22.

[2] I. Shamsher, Z. Ahmad, J. K. Orakzai, and A. Adnan, "Ocr for printed urdu script using feed forward neural network," in *Proceedings of World Academy of Science, Engineering and Technology*, 2007, pp. 172–175.

[3] P. Natarjan, Z. Lu, R. Schwartz, I. Bazzi, and J. Makhou, "Multilingual machine printed ocr," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 43, 2001.

[4] T. Nawaz, S. A. H. S. Naqvi, H. ur Rehman, and A. Faiz, "Optical character recognition system for urdu (naskh font) using pattern matching technique," *International Journal of Image Processing*, vol. 3, pp. 92–104, 2009.

[5] N. B. Amor and N. E. B. Amara, "Multifont arabic character recognition using hough transform and hidden markov models," in *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 2005, pp. 285–288.

[6] K. Khan, R. Ullah, N. A. Khan, and K. Naveed, "Urdu character recognition using principal component analysis," *International Journal of Computer Applications*, vol. 60, no. 11, pp. 1–4, 2012.

[7] Z. Ahmad, J. K. Orakzai, I. Shamsher, and A. Adnan, "Urdu nastaleeq optical character recognition," in *Proceedings of world academy of science, engineering and technology*, 2007, pp. 249–252.

[8] S. T. Javed and S. Hussain, "Segmentation based urdu nastalique ocr," in *Proceedings of 18th Iberoamerican Congress on Pattern Recognition*, 2013, pp. 41–49.

[9] M. W. Sagheer, C. L. He, N. Nobile, and C. Y. Suen, "Holistic urdu handwritten word recognition using support vector machine," in *Proceedings of 20th International Conference on Pattern Recognition*, 2010, pp. 1900–1903.

[10] S. Sardar and A. Wahab, "Optical character recognition system for urdu," in *Proceedings of International Conference on Information and Emerging Technologies*, 2010, pp. 1–5.

[11] A. Muaz, "Urdu optical character recognition system," 2010.

[12] M. A. U. Rehman, "A new scale invariant optimized chain code for nastaliq character representation," in *Proceedings of 2nd International Conference on Computer Modeling and Simulation*, vol. 4, 2010, pp. 400–403.

[13] H. Malik and M. A. Fahiem, "Segmentation of printed urdu scripts using structural features," in *Proceedings of 2nd International Conference in Visualisation*, 2009, pp. 191–195.

[14] U. Pal and A. Sarkar, "Recognition of printed urdu script," in *Proceedings of 12th International Conference on Document Analysis and Recognition*, vol. 2, 2003, pp. 1183–1187.

[15] A. Cheung, M. Bennamoun, and N. W. Bergmann, "An arabic optical character recognition system using recognition-based segmentation," *Pattern Recognition*, vol. 34, no. 2, pp. 215–233, 2001.

[16] M. M. Altuwaijri and M. A. Bayoumi, "Arabic text recognition using neural networks," in *IEEE International Symposium on Circuits and Systems*, vol. 6, 1994, pp. 415–418.

[17] M. A. Abdullah, L. M. Al-Harigy, and H. H. Al-Fraidi, "Off-line arabic handwriting character recognition using word segmentation," *Computing Research Repository*, vol. 4, no. 3, pp. 40–44, 2012.

[18] M. Sarfraz, S. N. Nawaz, and A. Al-Khuraidly, "Offline arabic text recognition system," in *Proceedings on International Conference on Geometric Modeling and Graphics*, 2003, pp. 30–35.

[19] Z. Ahmed, J. K. Orakzai, and I. Shamsher, "Urdu compound character recognition using feed forward neural networks," in *Proceedings of 2nd IEEE International Conference on Computer Science and Information Technology*, 2009, pp. 457–462.

[20] A. Mesleh, A. Sharadqh, J. Al-Azzeh, Z. MazenAbu, N. Al-Zabin, T. Jaber, A. Odeh, and M. Hasn, "An optical character recognition," *Contemporary Engineering Sciences*, vol. 5, pp. 521 – 529, 2012.

[21] S. A. Sattar, S. Haque, and M. K. Pathan, "Nastaliq optical character recognition," in *Proceedings of the 46th Annual Southeast Regional Conference*, 2008, pp. 329–331.

[22] M. Akram and S. Hussain, "Word segmentation for urdu ocr system," in *Proceedings of the 8th Workshop on Asian Language Resouces*, 2010, pp. 88–94.

[23] S. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil, and H. Moin, "Segmentation free nastalique urdu ocr," in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 46, 2010, pp. 456–461.

[24] N. Sabbour and F. Shafait, "A segmentation-free approach to arabic and urdu ocr," in *Proceedings of Document Recognition and Retrieval*, 2013.

[25] D. Satti, "Offline urdu nastaliq ocr for printed text using analytical approach," 2013.

[26] M. Kavianifar and A. Amin, "Preprocessing and structural feature extraction for a multi-fonts arabic/persian ocr," in *Proceedings of the 5th International Conference on Document Analysis and Recognition*, 1999, pp. 213–216.

[27] M. T. El-Melegy and A. A. Abdelbaset, "Global features for offline recognition of handwritten arabic literal amounts," in *Proceedings of the 5th International Conference on Information and Communications Technology*, 2007, pp. 125–129.

[28] N. Shahzad, B. Paulsonn, and T. Hammond, "Urdu qaeda: Recognition system for isolated urdu characters," in *Proceedings of IUI Workshop on Sketch Recognition*, 2009.

[29] Z. Shah and F. Saleem, "Ligature based optical character recognition of urdu nastaleeq font," in *Proceedings of International Multi Topic Conference*, 2002.

[30] S. M. Lodhi and M. A. Matin, "Urdu character recognition using fourier descriptors for optical networks," in *Photonic Devices and Algorithms for Computing VII, Proc. of SPIE*, 2005.

[31] 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 Recognition*, vol. 47, no. 3, pp. 1229–1248, 2013.

[32] M. Michael and N. Papamarkos, "An adaptive layer-based local binarization technique for degraded documents," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 24, no. 245, 2010.

[33] P. Sanjoy, P. Bhowmick, S. Sural, and J. Mukhopadhyay, "Skew correction of document images by rank analysis in farey sequence," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 27, no. 1353004, October 2013.

[34] K. Khurshid, "Analysis and retrieval of historical document images," Ph.D. dissertation, 2009.

[35] A. Bensefia, T. Paquet, and L. Heutte, "A writer identification and verification system," *Pattern Recognition Letters*, vol. 26, no. 13, pp. 2080–2092, 2005.

[36] I. Siddiqi and N. Vincent, "Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features," *Pattern Recognition*, vol. 43, no. 11, pp. 3853–3865, 2010.

[37] C. W. Ng and S. Ranganath, "Real-time gesture recognition system and application," *Image and Vision Computing*, vol. 20, no. 13-14, pp. 993 – 1007, 2002.

[38] J. Triesch and C. von der Malsburg, "Classification of hand postures against complex backgrounds using elastic graph matching," *Image Vision Computing*, vol. 20, pp. 937–943, 2002.

[39] H.-S. Yoon, J. Soh, Y. J. Bae, and H. S. Yang, "Hand gesture recognition using combined features of location, angle and velocity," *Pattern Recognition*, vol. 34, no. 7, pp. 1491–1501, 2001.

[40] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov models for speech recognition*. Edinburgh university press Edinburgh, 1990, vol. 2004.

[41] Thomas and G. A. Fink, "Markov models for offline handwriting recognition: A survey," *International Journal of Document Analysis and Recognition*, vol. 12, no. 4, pp. 269–298, 2009.

[42] E. Kavallieratou, E. Stamatatos, N. Fakotakis, and G. Kokkinakis, "Handwritten character segmentation using transformation-based learning," in *Proceedings of International Conference on Pattern Recognition*, vol. 2, 2000, pp. 2634–2634.

[43] H. Yasuda, K. Takahashi, and T. Matsumoto, "A discrete hmm for online handwriting recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 675, 2000.

[44] J. J. Lee, J. Kim, and J. H. Kim, "Data-driven design of hmm topology for online handwriting recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, no. 107, 2001.

[45] B. Pardo and W. Birmingham, "Modeling form for on-line following of musical performances," in *Proceedings of 20th National Conference on Artificial Intelligence*, vol. 2, 2005, pp. 1018–1023.

[46] S. Al-Qahtani, M. Khorsheed, and M. AlSuliman, "Recognising cursive arabic script using hmms," in *NCC*, vol. 17, 2004, pp. 631–637.

[47] K. Khurshid, C. Faure, and N. Vincent, "Word spotting in historical printed documents using shape and sequence comparisons," *Pattern Recognition*, vol. 45, no. 7, pp. 2598–2609, 2012.