Segmentation Free Nastalique Urdu OCR

Sobia T. Javed, Sarmad Hussain, Ameera Maqbool, Samia Asloob, Sehrish Jamil and Huma Moin

***Abstract*—**Electronically available Urdu data is in image form which is very difficult to process. Printed Urdu data is the root cause of problem. So for the rapid progress of Urdu language we need an OCR system, which can help us to make Urdu data available for the common person. Research has been carried out for years to automate Arabic and Urdu script. But the biggest hurdle in the development of Urdu OCR is the challenge to recognize Nastalique Script which is taken as standard for writing Urdu language. Nastalique script is written diagonally with no fixed baseline which makes the script somewhat complex. Overlap is present not only in characters but in the ligatures as well. This paper proposes a method which allows successful recognition of Nastalique Script.

***Keywords*—** HMM, Image processing, Optical Character Recognition, Urdu OCR.

I. Introduction

THE text within the image is non-editable. Due to this non-editable nature of text, information retrieval, collection and sharing become a big issue. So we need a mechanism to convert this text image into machine editable text form. OCR (Optical character recognition) helps to convert the text image such as scanned document, electronic fax files, and picture of document taken from cameras into text file that can be opened in any word processor or text editor.

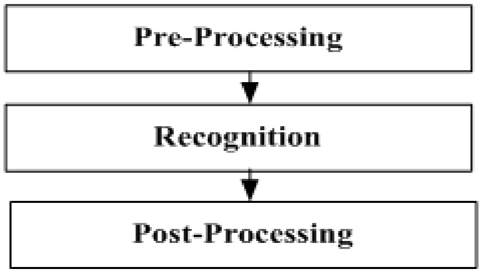


Fig. 1 OCR process [1]

The process of OCR can be divided into three main processes [1]

1. Preprocessing
2. Recognition
3. Post Processing

Different preprocessing techniques are applied to extract noise and distortion free image. The text areas are extracted and fed to different classifiers for recognition. The result of classifiers is significantly improved in the last step of post processing.

II. Urdu Writing System

Urdu is the national language and lingua franca of Pakistan. Urdu is bidirectional as words are written right to left where numbers are written left to right as shown below [2].

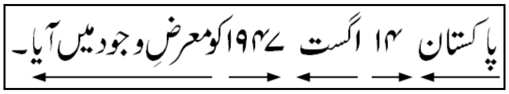
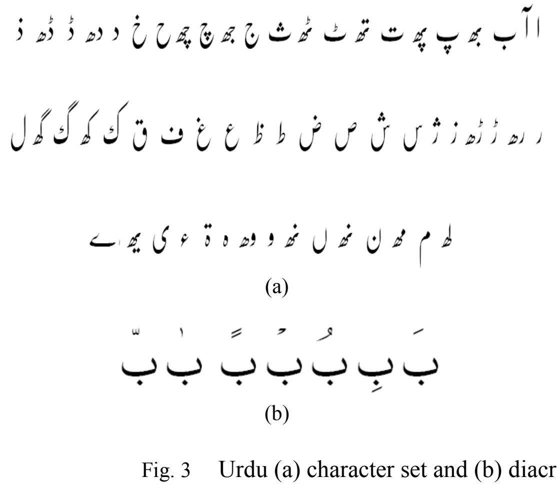


Fig. 2 Bidirectional Urdu script

The extended Arabic character set of Urdu join together to form words/ ligatures ([3], [4]). Urdu is very context sensitive, that is, characters change their shapes depending upon the context in which it is written. Due to the context sensitive nature, the shape of character may be categorized as isolated, initial, medial and final. In addition to the dots of the character, diacritical marks are used for the proper pronunciation of Urdu word [5]. The diacritics may appear above or below a character.


(a)

(b)

Fig. 3 Urdu (a) character set and (b) diacritical marks. Figure taken from [1]

Following table shows the different unique classes.

Sobia T. Javed is working as an Assistant Professor at National University of Computer & Emerging Sciences (FAST-NU). (e-mail: sobia.tariq@nu.edu.pk).

Sarmad Hussain is at Center for Language Engineering at Al-Khawarizmi Institute of Computer Science, UET, Lahore (sarmad@cantab.net).

Ameera Maqbool, Samia Asloob, Sehrish Jamil and Huma Moin were with National University of Computer & Emerging Sciences (FAST-NU)

TABLE I
CLASSIFICATION OF URDU LETTERS BASED ON THEIR SHAPES IN CURSIVE FORM

Members	Class	Members	Class	Members	Class
م	م	ص ض	ص	ا آ	ا
و	و	ط ظ.	ط	ب پ ت ٹ ث ن	ب
ء	ء	ع غ	ع	ج چ ح خ	ج
ھ	ھ	ف ق	ف	د ڈ ذ	د
ی	ی	ک گ	ک	ر ڑ ز ژ	ر
ے	ے	ل	ل	س ش	س
ق	ق	ف	ف	ن ں	ن

Urdu can be written with different styles. Nastalique, Naskh, Kofi, Thuluth, Devani and Riqa are some of the different styles of writing Urdu [6]. Today most commonly used script are Nastalique and Naskh. The figure below tells how these different styles look like.

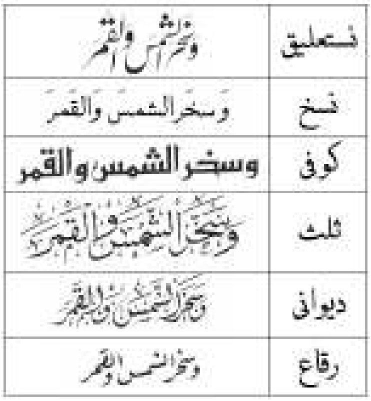


Fig. 4 Different writing styles for Arabic script. Figure taken from [6].

A. Nastalique Script

Nastalique is a combination of two different fonts, Naskh and Taleeq. It was initially created by Mir Ali Tabrezi. The calligrapher tends to increase the beauty of the script and making it a complex script. It is highly cursive and context sensitive in nature

Some of the characteristics are as follows:

1. It is written diagonally from top right to bottom left. That means all the ligatures are tilted at a certain angle towards the right side as shown below. Due to this diagonal nature the Nastalique consumes less horizontal space as compared to Nask [2].

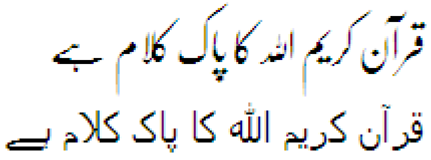


Fig. 5 Nastalique style (top) takes less horizontal space than Naskh style (bottom) for the same text [1]

2. The joins are formed by cusp-like shapes, which are concave upwards and have their initial end higher than the final end [2].

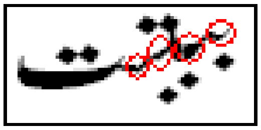


Fig. 6 Cusp like joint are marked with red

3. Overlapping problem is present in characters and ligatures [2]. For example kaf of the word "kisam" is overlapping "meem" of tamam.

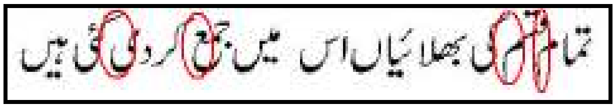


Fig. 7 Overlap between the ligatures

4. Context Sensitive nature of nastalique compels the letters to adapt different shapes [6].

Following figures are taken from [2]

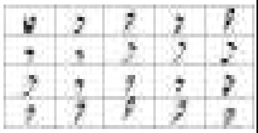
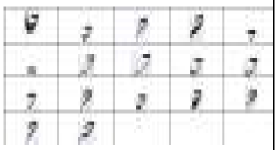
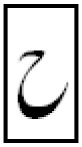
 Initial Shapes of Hay	 Medial Shapes of Hay
 Isolated form of Hay	 Final Shape

Fig. 8 Different form of letter Hay depending on the context.

III. Methodology

Mostly the work that is available today is on Arabic OCR. Little effort has been made on Urdu OCR and especially on Nastalique Urdu OCR [17, 18, 19, 20, 21, and 22]. The cursive and context sensitive nature of the script complicates the situation even further.

For recognizing cursive script, there are two main approaches to deal with connected characters in a word.

1. Segmentation based Approach
2. Segmentation free Approach

Due to highly cursive nature the segmentation of Urdu script is not a trivial task. So we will discuss segmentation free approach in which the ligature as a whole is used instead of segmenting it into smaller units. Different pattern matching techniques are used to classify the pattern. Most commonly used method in this approach is to extract features from the image and feed them to some recognizer for identification purpose. Three types of features are normally extracted which are transformational [7, 8], statistical [9, 10, 15] and structural [13].

In our research, we extracted global transformational features from a non segmented ligature and then fed them to Hidden Markov Model (HMM) recognizer. HMM has an ability to perform recognition with great ease and efficiency. Language-independent training and recognition methodology was a major reason of using HMM in developing recognition technology [11]. Hidden Markov model Tool Kit (HTK) [12] is used to implement HMM.

The Process of HMM recognition is divided into two tasks.

1. Training
2. Recognition

A. Training

Before starting the recognition process, the system is properly trained on each unique HMM (training data) [16]. Each connected body is considered as a separate HMM. Connected body may be a diacritic or a main body. A separate recognizer for diacritics and main bodies are built. Diacritic Recognizer is trained on one dot, two dots, three dots, Shad, Mad, Secondary Stroke of hey, Secondary Stroke of gaf, Khari Zabar, Do Zabar, Hamza and small toey. Main Body Recognizer is trained on 1500 high frequency ligatures which vary from one character ligature to seven characters ligature. In order to cater all sorts of noise and variations in the image we need to collect at least 30 samples of each HMM model.

The figure below gives the overall flow of the process.

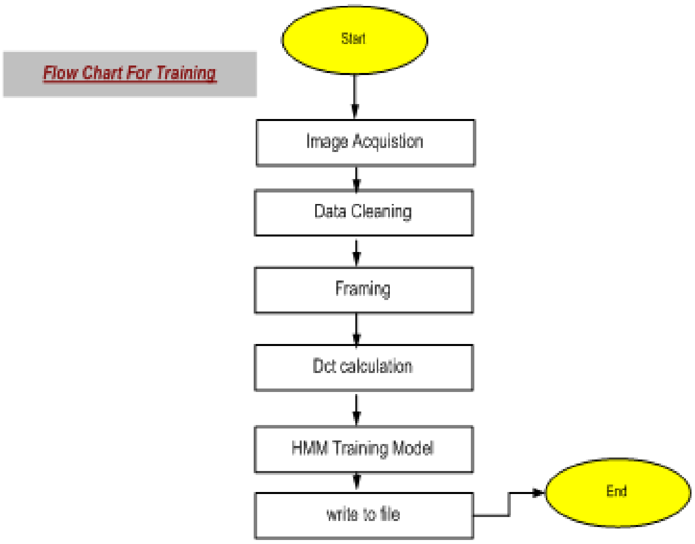
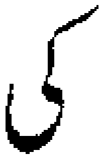
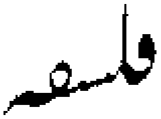


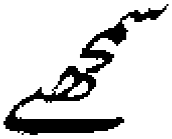
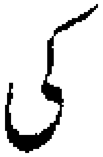
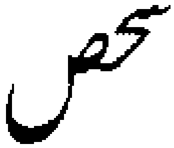
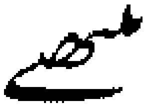
Fig. 9 Flow Chart of Training

TABLE II
HMM models for main body Recognizer

 h022	 h0366	 h0429
 h0528	 h0626	 h013
 h061	 h0620	 h0616

For main body Recognizer not only the HMM models are trained but a lexicon is also maintained at the time of training which tells about all the character classes within each model. Entry for each HMM model is incorporated in the lexicon. The following table gives some of the entries of lexicon to identify the character classes with in the ligature.

TABLE III
SOME OF THE ENTRIES OF LEXICON TO IDENTIFY THE CHARACTERS CLASSES WITHIN THE LIGATURE.

Main Body	HMM Model	Rule
	h061	(ے) class+ (ب) class+(ﮪ) class+(ج) class +(م) class+ (س) class
	h022	(ی) class + (ک) class
	h0366	(ص) class+ (ج) class + (س) class
	h0616	(ے) class+ (ب) class+(ﮪ) class+(ب) class +(ب) class + (ب) class

B. Recognition

Once all the models are properly trained, the system is ready for recognition. The overall flow of recognition process is given in the figure below. After separating and re-associating the diacritics and main body, each connected body is further divided into small chunks/windows called frames. The discrete cosine Transform (DCT) is calculated for each frame and then fed to HMM recognizer. The information regarding the location of the diacritic is also stored which will be in the recognition phase.

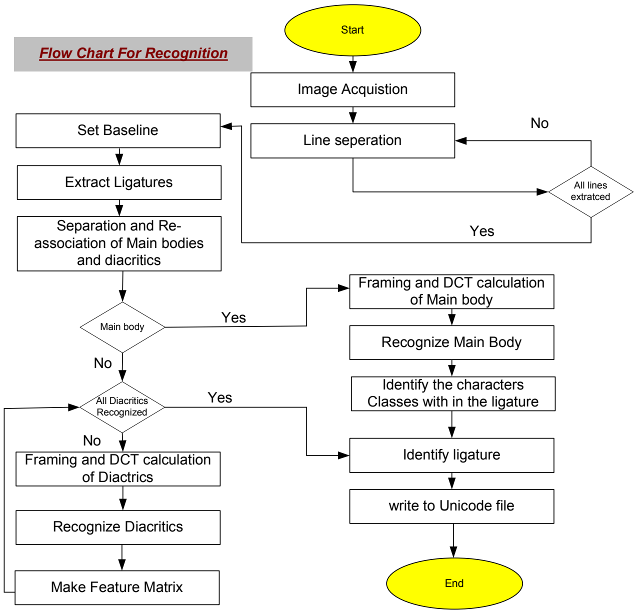

Flow Chart For Recognition
Start
Image Acquistion
Line seperation
No
All lines extracted
Yes
Set Baseline
Extract Ligatures
Separation and Re-association of Main bodies and diacritics
Main body
Yes
No
All Diacritics Recognized
Yes
No
Framing and DCT calculation of Diactrics
Recognize Diacritics
Make Feature Matrix
Framing and DCT calculation of Main body
Recognize Main Body
Identify the characters Classes with in the ligature
Identify ligature
write to Unicode file
End

Fig. 10 Flow Chart for Recognition

Diacritics are divided into vertical windows of size 9x25 whereas main body is divided into vertical windows of size 8x90. Bits are padded to complete the window. The direction of frame traversal is from right to left as shown below. Let's take an example of the Urdu letter "ب" and see how framing is done for its diacritic and main body.

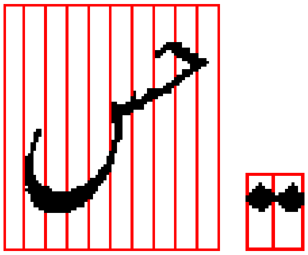


Fig. 11 Framing in main body and diacritic.

C. Feature Matrix

Once the diacritics are recognized, feature vector is prepared according to the information of diacritics. If no diacritic is present then the vector is '0'. If the diacritics are present then after recognizing them we make the vector as shown below.

We used the following codes for making feature matrix:

TABLE IV
CODES FOR MAKING FEATURE MATRIX

Main Body	Rule
1 dot above	01
2 dots above	02
3 dots above	03
1 dot below	04
2 dots below	05
3 dots below	06
Shad	07
Mad	08
Secondary Stroke of Hey	09
Secondary Stroke of Gaaf	10
Khari zabar	11
Do zabar	12
Hamza	13
Small toey	14
1 dot in the middle	16
2 dots in the middle	17
3 dots above in the middle	18
3 dots below in the middle	19
Hay in the middle	20
Hamza in middle	21
Small Toay in the middle	22

For example the feature matrix for the ligature "ٹبتـــغ" is give as below-

Feature Matrix:

14	16	02	01

D. Recognizing the ligature

After recognizing the main body, the already developed lexicon is used to identify the classes of the characters with in the ligature. Once the classes are identified, next step is to exact characters within the ligature. For this purpose rule based recognizer is used. Feature matrix along with the identified classes is fed to rule based recognizer which then correctly recognizes the ligature. Some of the rules are listed below

TABLE V
RULES FOR LIGATURE CREATION

Sr.	Identified classes	Feature Matrix	Rule
1	**(ا) class + (ب) class**	01	تا
2	**(ا) class + (ب) class**	06	با
3	**(ے) class+ (ب) class+(ھ) class+(ب) class +(ب) class + (ب) class**	04 | 05 | 14 | 01	بیٹھنے
4	**(ے) class+ (ب) class+(ھ) class+(ب) class +(ب) class + (ب) class**	04 | 05 | 14 | 02	بیٹھنے

As the characters are identified, their respective Unicode is also written in the file which gives us the required ligature.

Let's take an example of the following ligature and see how it is recognized.

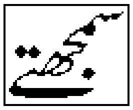


Fig. 12 Ligature to be recognized

First of all diacritics and main bodies will be separated and location of both the diacritics is store. The first single dot is marked as middle and two dots as above. Then main body and all the diacritics are sent to their respective recognizer.

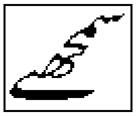
Main body	**Diacritics**	
	 (1)	 (2)
h061	H01	h02

Fig. 13 Diacritics Separated from Main body

Main body recognizer will give us h061 as the identified model; similarly output of diacritic recognizer will be h01 and h02. These HMM numbers tell us the type of the pattern recognized. h01 and h02 tell that diacritic (1) is one dot and diacritic (2) is two dots respectively. Model h061of main body tells that it is six characters word with the following classes.

(ے) class+ (ب) class+(ھ) class+(ج) class +(م) class+ (س) class

Next step is to prepare a feature vector. It has one dot in the middle and two dots above so the feature vector will look like the following.

Feature Matrix:

16	02

(ے) class+ (ب) class+(ھ) class+(ج) class +(م) class+ (س) class +	**16**	**02**	ســـمجھتے

Then the Unicode for all these characters are written in the text file such that a ligature is formed.

IV. RESULT

A total of 1282 unique ligatures are extracted from the 5000 high frequency words in a corpus-based dictionary [14]. It is also confirmed that all Urdu letters are used in these ligatures in a variety of contexts. For analysis purpose three or more samples of each ligature are taken to form the text. These pages are printed in Noori Nastalique font at font size 36. The pages are then scanned at dpi 150 and then separated back into ligatures. A total of 3655 ligatures are tested and 3375 ligatures are accurately identified, giving an accuracy of 92%.

V. DISCUSSION

Testing and further analysis shows the following reasons for the errors.

Sometimes the shapes of different ligatures are so similar that they give approximately same state transition probabilities. So when these inputs are given to HMM for recognition same state sequence of HMM is obtained. Erroneous output is thus obtained due to the slight similarity in shape. The shape of the ligatures “بسبز” and “تبسر” are similar to each other when written in Noori Nastaleeq font. Sometimes the OCR is not able to recognize ligatures like these correctly because of the similarity in shape. Another such problem can be seen in “مگ” and “نگ”. Both these ligatures are more similar than different. Sometimes they are recognized correctly and sometimes they are not. This problem can be removed by making the sliding window overlapping. By using overlapping windows finer details of ligature shape can be also be covered.

Sometimes due to noise or poor scanning the actual shape of the diacritic is distorted. In this case, the diacritic is not recognized correctly and as a result incorrect feature vector is made. Thus the ligature is not recognized. During scanning the diacritic “ ◆◆ ” is distorted. Total number of windows and DCT values differ to great extent. This results in erroneous recognition of diacritic. Increasing the number of samples, while training, can improve the accuracy.

For recognizing a ligature we first identify its shape and then recognize it by seeing the class of feature vector which it belongs to. Sometimes it so happens that 2 different ligatures have exactly the same feature vector though position and location of dots are different. In this case the result is erroneous due to the same feature vector. In Ligatures “عذ” and “غد” the shape of main body is same and the feature vector of both the ligatures is “01”. Due to the same feature vector and main body, these ligatures can’t be trained and recognized at the same time. Another such problem can be seen in the ligatures “خصر” and “حضر”. These ligatures too have same shape of main body and the feature vector in both these ligatures is “02”. Only one of them can be recognized at a time.

VI. Conclusion

The Nastalique script is complex due to its context sensitive and cursive nature. The absence of baseline and complex mark placement rules makes the situation even more worst. So the extraction of Nastalique text from the image is not a trivial task. The algorithm has been developed which successfully extracts the texts with an accuracy of 92%.

VII. References

[1] Javed, S.T., Hussain, S. “Improving Nastalique Specific Pre-Recognition Process for Urdu OCR”, In the Proceedings of 13th IEEE International Multitopic Conference 2009 (INMIC 2009), Islamabad, Pakistan, 2009 (URL: http://www.jinnah.edu.pk/inmic2009)

[2] Wali, A. and Hussain, S. “Context Sensitive Shape-Substitution in Nastaliq Writing system: Analysis and Formulation,” in the Proceedings of International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE), 2006.

[3] Hussain, S. and Durrani, N. “Urdu,” in A Study on Collation of Languages from Developing Asia, Center for Research in Urdu Language Processing, NUCES, Pakistan, 2007.

[4] Hussain, S. and Afzal, M. “Urdu Computing Standards: UZT 1.01”, in the Proceedings of the IEEE International Multi-Topic Conference, Lahore, Pakistan, 2001.

[5] Hussain, S. “Letter to Sound Rules for Urdu Text to Speech System”, In the Proceedings of Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004, Geneva, Switzerland, 2004.

[6] Hussain, S. “www.LICT4D.asia/Fonts/Nafees_Nastalique,” in the Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society, Asian Media Information Center, Singapore, 2003.

[7] Lu, Z., Bazzi, I., Kornai, A. and Makhoul, J. “A Robust, Language-Independent OCR System,” in the 27th AIPR Workshop: Advances in Computer Assisted Recognition, SPIE, 1999.

[8] El-Hajj, r., Likforman-Sulem, L. and Mokbel, C. "Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling," in the 8th International Conference on Document Analysis and Recognition (ICDAR), South Korea, 2005.

[9] Shah, Z. and Saleem, F. “Ligature Based Optical Character Recognition of Urdu, Nastaliq Font,” in the Proceedings of International Multi Topic Conference, Karachi, Pakistan, 2002.

[10] Husain, S.A. and Amin, S.H. “A Multi-tier Holistic approach for Urdu Nastaliq Recognition,” in the Proceedings of International Multi Topic Conference, Karachi, Pakistan, 2002.

[11] Rabiner, L. and Juang, B. “Theory and Implementation of Hidden Markov Models” in the book, “Fundamental of Speech Recognition”, chapter 6, published in 1993.

[12] Young,S., Evermann, G., Hain,T., Kershaw, D., Moore, G., Odell, J.,Ollason, D., Povey, D., Valtchev, V., and Woodland, P. “The HTK Book”, December 1995.

[13] Khorsheed, M. S., Clocksin, W.F. "Structural Features Of Cursive Arabic Script", in Proceeding of British Machine Vision Conference, pg.1285-1294, 1999.

[14] Ijaz, M., Hussain, S. “Corpus Based Urdu Lexicon Development”, In the Proceedings of Conference on Language Technology (CLT07), University of Peshawar, Pakistan, 2007.

[15] Pal, U. and Sarkar, A. “Recognition of Printed Urdu Text,” in the Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.

[16] Bojovic, M. and Savic, M. D. "Training of Hidden Markov Models for Cursive Handwritten Word Recognition," in the Proceedings of the15th International Conference on Pattern Recognition (ICPR) vol.1, 2000.

[17] Ahmad, Z., Orakzai, J. K. , Shamsher, I. and Adnan, A. “Urdu Nastaleeq Optical Character Recognition,” in the Proceedings of World Academy of Science, Engineering and Technology 26, 2007.

[18] Shafait, F., Hasan, A., Keysers, D. and Breuel, T. “Layout analysis of Urdu document images” in Proceedings of IEEE Multitopic Conference (INMIC 06), 2006.

[19] Safabakhsh, R. and Abidi, P. "Nastaaligh Handwritten Word Recognition Using a Continuous-Density Variable-Duration HMM", The Arabian Journal for Science and Engineering, April 2005.

[20] Shamsher, I., Ahmad, Z., Orakzai, J. K. and Adnan, A. “OCR for Printed Urdu Script Using Feed Forward Neural Network”, in the Proceedings of World Academy of Science, Engineering and Technology 23, 2007.

[21] Razzak,M., Hussain,A., Sher,M., and Khan,Z. "Combining Offline and Online Preprocessing for Online Urdu Character Recognition",Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I, IMECS 2009, March 18 - 20, 2009.

[22] Hussain, A., Anwar, F., and Sajjad, A. “Online Urdu Character Recognition System.” MVA2007 IAPR Conference on Machine Vision Applications, 2007.