Adapting Tesseract for Complex Scripts: An Example for Urdu Nastalique

Qurat ul Ain Akram, Sarmad Hussain, Aneeta Niazi, Umair Anjum, Faheem Irfan

Center for Language Engineering, Al-Khawarizmi Institute of Computer Science
University of Engineering and Technology
Lahore, Pakistan
ainie.akram@kics.edu.pk, firstname.lastname@kics.edu.pk

***Abstract*—Tesseract engine supports multilingual text recognition. However, the recognition of cursive scripts using Tesseract is a challenging task. In this paper, Tesseract engine is analyzed and modified for the recognition of Nastalique writing style for Urdu language which is a very complex and cursive writing style of Arabic script. Original Tesseract system has 65.59% and 65.84% accuracies for 14 and 16 font sizes respectively, whereas the modified system, with reduced search space, gives 97.87% and 97.71% accuracies respectively. The efficiency is also improved from an average of 170 milliseconds (ms) to an average of 84 ms for the recognition of Nastalique document images.**

Keywords- Tesseract, Urdu, Nastalique, OCR

I. INTRODUCTION

Urdu language is written in Nastalique writing style of Arabic script which is cursive and has highly context sensitive shaping [1]. Normally, one or more characters join to form a ligature [2] which has a single main body (base stroke of a ligature) and Tashkil or diacritics. These ligatures group together to form words in Urdu. Based on the similarity of the shape of their main bodies, Urdu characters are categorized into twenty classes [1].

Nastalique is a complex writing style. The ligature is written diagonally from top right to bottom left with only the last character placed on the baseline. Ligature height grows diagonally upwards as various characters are cursively joined, varying the line height. Further character strokes and joins vary in thickness over a stroke, as shown in Fig. 1.

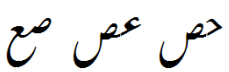


Figure 1. Thick-thin stroke variation across characters in a ligature

Space is used inconsistently for writing Urdu words [3], with more complex methods to correctly group ligatures into words [4]. In addition, due to diagonal nature of Nastalique, in flowing text ligatures overlap vertically, as shown in Fig. 2. These factors cause significant processing complexity [1].

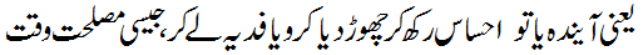


Figure 2. Vertical overlapping of characters and ligatures

Tesseract is an open source multilingual OCR engine [5], with some success for complex scripts. This paper presents challenges of using Tesseract for the recognition of Urdu text in Nastalique writing style, and suggests the necessary modifications for better recognition accuracy.

II. LITERATURE REVIEW

Tesseract uses the polygonal approximation technique [6] to extract features used for classification and recognition. Its pre-processing module extracts connected components called the blobs from the input image and computes the noisy connected components using bounding box size information of the neighboring connected components [5]. Connected components are further processed to segment lines using horizontal projection profile. The extracted blobs are initially considered as characters, and a ranked recognition list is computed for each blob. After recognition, the dictionary is consulted to form a word from a combination of recognized choices. The blobs which have low recognition confidence are chopped into smaller sub-blobs and the process is repeated. At the end, highest ranked word is selected. Chopping of the blobs improves the recognition results for Latin script based languages but increases the recognition time. This also works well for other languages with predictable character widths and regular space between characters and words.

OCRs using Tesseract have been developed for English, French, Italian, German, Spanish and Dutch languages with 99.25% character recognition accuracy. Systems using Tesseract for Cyrillic, Chinese and Devanagari scripts report accuracies of 98.67%, 84.59% and 96.23% respectively [5, 6]. Tesseract based OCR system for Bangla [7] reports 93%, 85% and 70% recognition accuracies for printed books, newspapers and screen printed images respectively. Chaulagain et al. [8] present an OCR for Nepali using Tesseract. The engine has also been used for handwriting recognition, e.g., for Roman numerals with 92.1% and 86.59% accuracies on known and unknown writers respectively [9] and for English letters, digits and alpha-numeric characters using three different Tesseract sub-classifiers [10], with 84% character and 93.89% digit recognition accuracy.

Character based recognition of Arabic text is difficult due to its cursive nature. Therefore, segmentation-free, ligature based recognition is generally used. For example, HMMs are

used for the recognition of Arabic text [11]. The synthesized data generated for five writing styles at 14 font size gives an accuracy of 97.65%. Limited work has been carried out for Urdu text images in Nastalique writing style. Examples include recognition systems using the structural features of a main body with neural networks as the classifier [12] and using the contextual features of ligature contours (of main body and diacritics) [13]. The reported accuracy of the latter system is 91% for Urdu and 86% for Arabic, tested on synthetic data for Urdu (using Nastalique writing style) and Arabic (using Naskh). Javed et al. [14] extract features through windowing ligatures and use HMMs for recognition. The reported accuracy of the system is 92% on synthesized data at 36 font size. Lehal and Rana [15] recognize Nastalique ligatures using Zoning, DCTs, Density, and Gabor features. The primary and secondary ligature strokes are separately recognized using SVM, KNN and HMM classifiers for the comparative analysis. The system recognizes 2190 unique primary strokes and 17 secondary strokes. It gives 98.01% accuracy for the primary stroke recognition tested on 4380 images and 99.91% for the secondary stroke tested on 1700 images, by using DCTs with SVM for best results. The reported computation time for recognition of 400 primary components is 77 seconds for SVM and 26 seconds for KNN. In addition, the segmentation-based OCRs for the recognition of Nastalique text are also reported with reasonable accuracy for synthesized data at 36 font size [16, 17].

III. METHODOLOGY

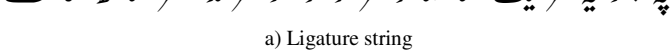
Recognition of a cursive script is a challenging task. This becomes more complicated while dealing with the complex styles such as Nastalique. Different approaches have been used for the recognition of Nastalique text, with limitations such as small synthesized data set, large font sizes, etc., whereas majority of the Urdu books are available between 14-16 font sizes. Recent attempts to use Tesseract for the recognition of Arabic text need to mature as reported accuracy of Arabic recognition is 83.8% [13]. In this paper, Tesseract engine is evaluated and analyzed for the recognition of Urdu main bodies. After a detailed analysis of Tesseract algorithm, pre-processing, chopping, dictionary and top ranked list functionalities are modified to give better recognition and efficiency results for Nastalique writing style. The details of these modifications are given below.

Urdu ligature has a main body and zero or more diacritics with complex placement rules as shown in Fig. 2. After training, it is found that Tesseract confuses some main bodies with diacritics.

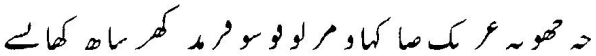
One reason for this mis-recognition is that some of the diacritics have confusingly similar shapes with the main bodies, even though they are smaller in size, as shown in Table I. Hence, to enable Tesseract for Urdu Nastalique text recognition only the main bodies are trained and recognized, and the diacritics are recognized separately. A pre-processing module is used to segregate them based on size. A document image is parsed and then recreated for training and recognition processes. Only the main bodies are placed in the image such that bottom of each is placed on the same line. Fig. 3 shows (a) the original image, and (b) the main body image, separated from the diacritics. For Tesseract training, a model is created for each main body.

TABLE I. DIACRITICS AND MAIN BODIES CONFUSION

Ligature	Main body shape	Confusing diacritic	Diacritic shape (enlarged)
		Diacritic of 	
		Diacritic of 	
		Diacritic of 	



a) Ligature string

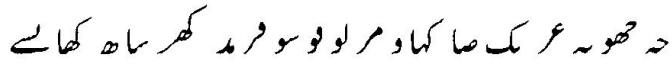


b) Main bodies of ligatures

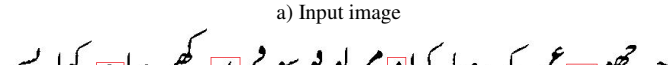
Figure 3. Ligature string and correcponding main bodies

Further, Tesseract engine is re-configured to give all the recognition results (along with the confidence measure against each option) instead of its default setting which only returns the option with the best confidence. If the desired main body is part of this result set, this main body can be shortlisted through further post-processing, and therefore the recognition in this context is considered correct.

In Nastalique, due to the diagonality of ligatures, height of main bodies can vary considerably, as seen in Fig. 4(a). In such a mix, Tesseract considers the small valid main bodies as noise and does not process them for recognition. For example, in Fig. 4(b) the marked main bodies are skipped and not processed for recognition. This functionality is removed by changing the relevant Tesseract flags in its code. Further, Tesseract's horizontal line cutting introduced to address varied line height causes additional errors for the recognition of Urdu main bodies. This functionality is blocked in the code.



a) Input image



b) Missed main bodies (marked by the rectangle)

Figure 4. Main bodies in input and original images

Tesseract chops a blob into smaller pieces based on the thin stroke and convex point analysis. Chopping criteria of Tesseract is not applicable to Nastalique ligatures because chopping of the main bodies of a single class is inconsistent for the different tokens, as can be seen in Fig. 5. In addition, this chopping introduces another error by recognizing two or more main bodies against a single one, as given in Table II. This functionality is disabled using the chop_enable flag.

Figure 5. Inconsistent Tesseract chopping of tokens of same type

TABLE II. TESSERACT CHOPPING MIS-RECOGNITION RESULTS

Main body image	Liga-ture of main body	Recog-nized shape 1	Liga-ture string of shape 1	Recog-nized shape 2	Liga-ture string of shape 2
ف	ف	ٶ	٬	ب	ب
لس	لس	س	س	ا	ا
کھکا	پھینکا	سکا	سکا	کھ	بچھ

Normally space is not used to define word boundary in Urdu. A statistical model has been developed which converts the sequence of ligatures into best sequence of words [4]. A comprehensive cleaned text corpus is required to develop the language model of ligatures and words for this word segmentation. Tesseract requires DAWG files for contextual post-processing. These DAWGs are commonly based on characters therefore Tesseract cannot easily process ligatures for the recognition of words through its dictionary [5]. Therefore, the dictionary lookup of Tesseract is also disabled using the load_system_dawg, load_freq_dawg, load_punc_ dawg, load_number_dawg, and load_fixed_length_dawgs flags and by modifying the relevant code.

Instead of having a single system trained to recognize all the main bodies, the main bodies are divided into four sub-sets to improve recognition accuracy, with a separate Tessearct based classifier for each sub-set. The height and width features of the main bodies are tested using C4.5 algorithm, which identifies the width as the more significant factor for this division. Thus, the thresholds based on the width are computed to divide the training data into four sub-sets which are overlapping at their edges. The main bodies which lie in the overlapping region of two sets (based on their width) are trained for recognition in both sets. See also [20] for an alternate method for the search space reduction for Urdu Nastalique.

Initially ligature main bodies are classified using character class analysis. However, it is found that main bodies which have very similar shape are confused by Tesseract. Some such confusing main bodies are given in Table III. These main body classes are collapsed in training and testing data to improve recognition results, and are later disambiguated based on the diacritics associated with them. If main bodies have same sequence of diacritics, they are eventually disambiguated during word formation phase, not discussed in this paper.

TABLE III. SAMPLE MAIN BODIES CONFUSED BY TESSEARCT BUT DISAMBIGUATED BY DIACRITICS ASSOCIATED WITH THEM

Main body shape	Ligature string of main body	Similar main body class shape	Ligature string of similar main body class
ملہ	بلہ	ملہ	ملہ
ملے	بلے	ملے	ملے
حصہ	حصہ	حصہ	خفیہ

For each change in the Tesseract algorithm, a separate system is developed so that the impact can be evaluated on the recognition accuracy and efficiency, as given in Table IV.

TABLE IV. MODIFIED TESSERACT SYSTEMS

System	Base version for change	Details
System-1		Tesseract code ver. 3.01
System-2	System-1	Modified code to output ranked list
System-3	System-2	Chopping disabled
System-4	System-3	Dictionary disabled
System-5	System-4	Pre-processing changed
System-6	System-5	Similar shapes merged in training data

IV. DESCRIPTION OF DATA FOR 14 AND 16 FONT SIZES

Majority of Urdu books are written in 14 and 16 font sizes, therefore image corpora for 14 and 16 font sizes [18, 19] are used for the training and testing of these systems. These image corpora are developed by identifying the high frequency ligatures from different text corpora. A total of 1,490,894 ligatures tokens are extracted, giving 7761 ligature types (covering 17,453 unique words derived from an Urdu corpus with 18 million words [21]). The ligatures reduce to 1475 main body types after removing the diacritics. Therefore, these main bodies constitute a representative subset of the larger set of main bodies found in Urdu language. Thirty five tokens of each main body type are initially printed (see [18, 19]). The images are manually cleaned to remove broken and joined instances, finally extracting 25 synthesized tokens of each main body. The breakdown of the main body classes in the final data set, according to the number of characters in the main body, is given in Table V.

TABLE V. DATA SET INFORMATION FOR MAIN BODIES OF HIGH FREQUENCY LIGATURES

Main body classes	Types	Training tokens	Testing tokens
One character	12	10	15
Two characters	61	10	15
Three characters	312	10	15
Four characters	632	10	15
Five characters	458	10	15

V. TESTING AND RESULTS

Urdu has a large number of main body classes (based on the unique shapes) as compared to Latin character classes. To handle this large set for training, an automatic training and testing system is developed. A total of 14,750 main body images are trained for 14 and 16 font sizes separately. Separate systems are trained using single training set and four overlapping sub-sets of training data, as already discussed. For the single system, Tesseract has to learn 1475 shapes, whereas for the sub-set based system, it has to learn (on average) 350 shapes for each sub-set. Although Tesseract is able to handle multiple font sizes, results still vary for 14 and 16 font sizes, as can be seen in Table VIII. Therefore, separate systems are developed for these font sizes. The image is recreated having ten samples of each type for training. The line spacing and the connected components spacing is set experimentally to achieve the best accuracy. For testing, 15 tokens of each of the 1475 classes are used (22,125 main body images for each font size). The recognition results are computed for each system discussed in Table IV, and are listed in Table VI and Table VII for 14 and 16 font sizes respectively. System-5 and System-6 give better accuracy. The overall accuracy of recognition is 97.87% for 14 font size and 97.71% for 16 font size. This is significant improvement from the baseline results using System-1, which gives 65.59% and 65.84% accuracy for 14 and 16 font sizes respectively. System-6 gives better efficiency compared to the other systems as shown in Table IX. Further, the system with four sub-sets gives better accuracy and efficiency compared to the systems with the complete data trained in a single set.

TABLE VI. TESSERACT SYSTEM ACCURACY FOR 14 FONT SIZE MAIN BODIES

	One character class		**Two characters class**		**Three characters class**		**Four characters class**		**Five characters class**	
	Single trained data file	***Four trained data files***	***Single trained data file***	***Four trained data files***	***Single trained data file***	***Four trained data files***	***Single trained data file***	***Four trained data files***	***Single trained data file***	***Four trained data files***
System-1	25.00	25.00	85.60	87.30	78.18	79.05	70.81	71.68	63.66	64.90
System-2	97.78	100.00	93.10	95.75	87.86	88.81	87.14	87.88	86.77	87.33
System-3	100.00	100.00	94.71	94.71	89.46	89.82	87.83	88.56	87.19	87.32
System-4	100.00	100.00	93.22	94.83	88.92	89.63	87.91	88.48	86.65	87.30
System-5	100.00	100.00	98.16	98.16	97.03	96.98	96.03	96.30	97.71	97.77
System-6	100.00	100.00	97.70	98.28	96.98	97.11	96.30	96.16	97.77	97.80

TABLE VII. TESSERACT SYSTEM ACCURACY FOR 16 FONT SIZE MAIN BODIES

	One characters class		**Two characters class**		**Three characters class**		**Four characters class**		**Five characters class**	
	Single trained data file	***Four trained data files***	***Single trained data file***	***Four trained data files***	***Single trained data file***	***Four trained data files***	***Single trained data file***	***Four trained data files***	***Single trained data file***	***Four trained data files***
System-1	25.00	25.00	88.49	89.06	78.95	79.62	70.45	71.61	62.49	63.83
System-2	100.00	100.00	93.64	93.64	88.33	88.87	85.56	86.23	85.61	86.07
System-3	100.00	100.00	94.33	94.56	88.25	88.96	86.24	86.84	85.76	86.23
System-4	100.00	100.00	94.33	94.56	88.25	88.94	85.87	86.19	84.84	85.41
System-5	100.00	100.00	98.70	98.81	96.62	97.35	95.68	96.01	95.94	96.42
System-6	100.00	100.00	98.59	98.81	96.66	97.39	95.93	95.91	96.14	96.47

TABLE VIII. OVERALL ACCURACY RESULTS ON SYSTEM-6 WITH FOUR SETS

Font size of testing data	**Total images**	**Font size of training data**	**Accuracy %**
14	22125	14	97.87
16	22125	16	97.71
14	22125	16	96.31
16	22125	14	96.71

TABLE IX. SYSTEM-WISE TESSERACT EFFICIENCY RESULTS

	Tested on single set (ms)	**Tested on four sub-sets (ms)**
System-1	170	122
System-2	172	134
System-3	155	105
System-4	158	102
System-5	143	91
System-6	123	84

VI. DISCUSSION

In this paper, Urdu Nastalique text recognition using the modified versions of Tesseract is presented. Tesseract's pre-processing, chopping and dictionary functionalities are disabled to improve the recognition accuracy and efficiency of Nastalique text recognition. As can be seen in Table VI and Table VII, the recognition results of each character class are improved. Further, dividing data into smaller sets considerably impacts both its accuracy and efficiency. The system-wise incremental accuracy improvements can also be observed. Cutting down the specialized components for processing Latin-type alphabetic scripts not only improves the accuracy but also reduces the computational costs, as given in Table IX, which gives the average time for recognizing an image of 15 main bodies. System-2 has more computational time as compared to the System-1 because of the output of the ranked list against each main body. From

System-2 onwards, the computational time reduces as different undesired functionalities are disabled. Overall System-6 takes 123 ms and 84 ms computation time using the single trained data file and the four sub-sets of trained data files respectively. The time for recognition using the trained data files for the sub-sets is reduced as there are fewer comparisons during the recognition process. The results clearly indicate that the classification and recognition algorithms used by Tesseract are effective. However, script specific processing (optimized for non-cursive Latin-like scripts) creates significant deterioration for cursive scripts, and needs to be blocked. It is also observed that Tesseract provides font size independent recognition, but the font specific systems still give better accuracy, as shown in Table VIII. The system shows most improvement by extracting a ranked list (and disambiguating using diacritics and a language model later) and by constraining its pre-processing, which splits the ligatures based on its height and width. The recognition results of System-6 are considerably improved from the Tesseract original system. System-6 has 97.87% and 97.71% accuracies for 14 and 16 font sizes whereas System-1 has 65.59% and 65.84% respectively. The data shows that with the modifications Tesseract can be adapted for use with cursive writing styles like Nastalique. A comprehensive system for Urdu Nastalique OCR has been developed based on this analysis.

VII. Conclusion

In this paper, Tesseract-based recognition system has been shown to work well for the recognition of Urdu Nastalique writing style. Initially, complete analysis of the Tesseract engine is performed for the recognition of Urdu. To handle different complexities of the Nastalique writing style for Arabic script, modifications are made to improve the accuracy. The system is analyzed and tested on 1475 main body types with up to five character ligatures. The modified system has 97.87 % accuracy for 14 font size and 97.71 % accuracy for 16 font size. Based on the current work, a complete system for the recognition of Urdu Nastalique has been developed using Tesseract.

VIII. Acknowledgemetns

This work has been supported by Urdu Nastalique OCR research project grant by ICTR&D Fund, Ministry of IT, Govt. of Pakistan. See www.UrduOCR.net for details.

IX. References

[1] A. Wali and S. Hussain, "Context Sensitive Shape-Substitution in Nastaliq Writing System: Analysis and Formulation," in CISSE, 2006.

[2] M. Davis, "Unicode Text Segmentation," Unicode, 2013.

[3] S. Hussain, "www.LICT4D.asia/Fonts/Nafees_ Nastalique," in 12th Annual Conference on E-Worlds, AMIC, Singapore, 2003.

[4] M. Akram and S. Hussain, "Word Segmentation for Urdu OCR System," in 8th Workshop on ALR, COLING, Beijing, China, 2010.

[5] R. Smith, D. Antonova and D.-S. Lee, "Adapting the Tesseract open source OCR engine for multilingual OCR," in International Workshop on Multilingual OCR, Barcelona, Spain, 2009.

[6] R. Smith, "An Overview of the Tesseract OCR Engine," in ICDAR, 2007.

[7] M. A. Hasnat, M. R. Chowdhury and M. Khan, "An Open Source Tesseract Based Optical Character Recognizer for Bangla Script," in ICDAR, 2009.

[8] B. Chaulagain, B. B. Rai and S. K. Raya, "Final Report on Nepali Optical Character Recognition," unpublished report, 2009.

[9] S. Rakhshit, A. Kundu, M. Maity, S. Mandal, S. Sarkar and S. Basu, "Recognition of handwritten Roman Numerals using Tesseract open source OCR engine," in Int. Conf. on Advanced in Computer Vision and Information Technology, 2009.

[10] S. Rakshit, S. S. Das, K. S. Sengupta and S. Basu, "Automatic Processing of Structured Handwritten Documents: An Application for Indian Railway Reservation System," International Journal of Computer Applications, vol. 6, September, 2010.

[11] A. Krayem, N. Sherkat, L. Evett and T. Osman, "Holistic Arabic Whole Word Recognition using HMM and Discrete Cosine Transformation," in ICDAR, 2013.

[12] Z. Shah and F. Saleem, "Ligature Based Optical Character Recognition of Urdu, Nastaleeq Font," in INMIC, Karachi, Pakistan, 2002.

[13] N. Sabbour and F. Shafait, "A Segmentation Free Approach to Arabic and Urdu OCR," in SPIE, Volume 8658, 2013.

[14] S. T. Javed, S. Hussain, A. Maqbool, S. Asloob, S. Jamil and H. Mohsin, "Segmentation Free Nastalique Urdu OCR," World Academy of Science, 2010.

[15] G. S. Lehal and A. Rana, "Recognition of Nastalique Urdu Ligatures," in 4th International Workshop on Multilingual OCR, New York, USA, 2013.

[16] S. Tariq and S. Hussain, "Segmentation Based Urdu Nastalique OCR," in CIARP, Havana, Cuba, 2013.

[17] A. Muaz, "Urdu Optical Character Recognition System," Unpublished, MS Thesis Report, NUCES, Lahore, Pakistan, 2010.

[18] CLE, "CLE Urdu HFL 14 Point Size," CLE. [Online]. Available: http://www.cle.org.pk/clestore /cleurduhfl14pt.htm.

[19] CLE, "CLE Urdu HFL 16 Point Size," CLE. [Online]. Available: http://www.cle.org.pk/clestore /cleurduhfl16pt.htm.

[20] El-Korashy and F. Shafait, "Search Space Reduction for Holistic Ligature Recognition in Urdu Nastalique Script," in ICDAR, 2013.

[21] M. Ijaz, S. Hussain, "Corpus Based Urdu Lexicon Development," in Conference on Language Technology, University of Peshawar, Pakistan, 2007.