

Font Size Independent OCR for Noori Nastaleeq

Qurat ul Ain Akram, Sarmad Hussain, Zulfiqar Habib

National University of Computer and Emerging Sciences
B Block, Faisal Town, Lahore, Pakistan

ainie.akram@nu.edu.pk, sarmad.hussain@nu.edu.pk, zulfiqar.habib@nu.edu.pk

Abstract— *This paper presents a technique for font size independent OCR of Noori Nastaleeq. Most of the existing OCRs for Noori Nastaleeq support only a single font size. Urdu government documents, news papers, magazines and books written in Noori Nastaleeq font style, has varying range of font sizes. The presented technique in this paper gives support for the font size independence for Noori Nastaleeq OCR, which makes the existing OCR [5] to recognize the Noori Nastaleeq text of different font sizes. The presented technique resizes the input ligature using Splines. Outline of the input ligature is extracted and then scaling factor is applied according to the font size to make the ligature outline to the size at which the existing OCR is trained. The scaled outline is then converted into the image form so that the OCR can recognize it. The presented technique is tested on the Urdu single character ligatures and the recognition rate is 98% for the manually generated data and 96% for the data scanned from different books and magazines.*

Keywords— **OCR, Noori Nastaleeq, Splines.**

I. INTRODUCTION

Optical character recognition (OCR) is the process of converting the document image into the editable text. Many OCRs related to the Latin script are proposed which give very reasonable accuracy.

In Pakistan, the use of Noori Nastaleeq script is most common. It is usually used in government documents, news papers, books etc. There is a need for an OCR system for Noori Nastaleeq font. Not much work has been done for the recognition of Noori Nastaleeq font.

Urdu character based technique for the optical character recognition is presented which uses MLP classifier for the training and recognition [23]. This system extracts the features which are extreme points such as left, right, top and bottom points of each character. Another OCR for Urdu is also proposed which supports the single character ligature [1]. It uses the combination of a topological, contour based and water reservoir features for the training and recognition.

A segmentation free technique of template matching for the optical character recognition of Noori Nastaleeq is also existed in which both main bodies and diacritics are recognized separately [2]. Another segmentation free approach is also presented which uses the Hidden Markov model for the training and recognition [3]. The accuracy of the system is 92.3%. A neural network based segmentation free approach is also presented which extracts the features such as

solidity, number of holes, axis ratio, moments, normalized segment length, curvature, and ratio of bounding box width and height for the training and recognition [4]. This system is tested on 3050 characters and its accuracy rate is 97.8%.

The segmentation based technique is also presented for the recognition of Noori Nastaleeq script [5]. This system supports the single font size which is 36 for the multiple characters ligatures. At the start the diacritics and main bodies are separated. After thinning, the main body is traversed. The segmentation of the ligature is performed at the point where more than one outgoing directions are found. The segmented primitives of the ligature are classified using Hidden Markov model by calculating the DCTs as features.

All of the above mentioned OCRs for the Urdu support only a single font size. Font size independent OCRs provide supports for the multiple font sizes images to be recognized by the OCR. There are many ways to do this. One way is to train the OCRs for multiple font sizes. Some OCRs for multiple font sizes are trained for different font sizes [6, 7, 8, 9, 10 and 11]. This means that the training corpus is increased with multiple font sizes to make the OCR font size independent.

Another way for the font size independent OCRs is to extract the features set which contains all those features which are invariant to the multiple font sizes [12, 13]. Some researchers also presented the technique of image normalization for font size independent OCRs [14, 15, 16].

II. METHODOLOGY

The proposed technique of the font size independent OCR for Noori Nastaleeq uses the idea for resizing using Splines. The presented technique captures the outline of the ligature and then scales the outline to the size at which the existing OCR [5] is trained. The figure given below shows the flow of the OCR along with the presented technique. Before the segmentation of the ligature, first the size normalization is performed which normalizes the size of the input ligature equivalent to the size of the trained ligature.

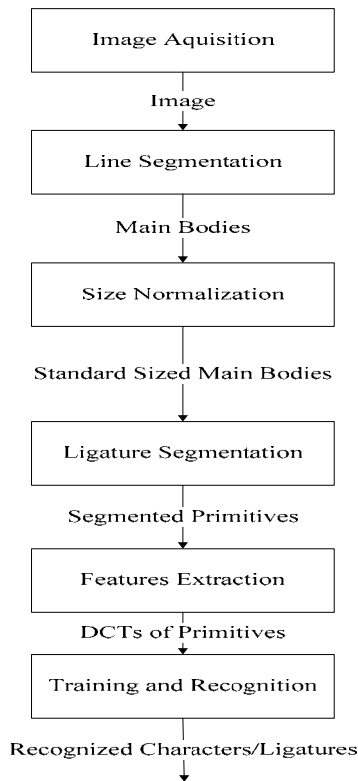


FIGURE 1. FLOW WITH RESPECT TO EXISTING OCR

The size normalization has two phases. In first phase, scaling factor is calculated according to the respective font size called scaling factor computation phase. In the second phase, simply the calculated scaling factor is applied according to the respective font size.

III. SIZE NORMALIZATION

At the start font size of the input ligature is computed. Font size normalization is not applied if the computed font size is standard one otherwise outline capturing is applied to normalize the size of the input ligature for the size normalization. After outline approximation, according to the font size, respective scaling factor is applied to the control points. These scaled control points are re-approximated. The re-approximated outline is then converted into the image form. The ligature discontinuity is resolved if it exists in the image boundary. After discontinuity resolving, the ligature's boundary is filled with black colour. The OCR [5] is used for the recognition of the scaled ligature. The Figure 2 shows the flow of this process. The detail of each of the process is given in the following sections.

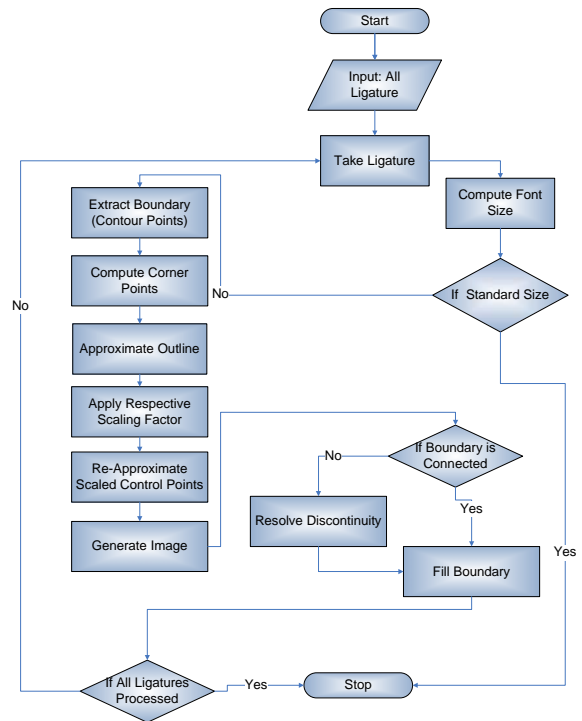


FIGURE 2. FLOW CHART OF TESTING PHASE

III.A.1 IMAGE ACQUISITION AND BINARIZATION

The first step of the system is to obtain the image form of the document which needs to be converted into the editable text. For this, the document needs to be scanned. The presented technique requires image to be in the binary form. The image is converted into the binary form using technique mentioned in [18] if the scanned image is in gray scale or colour.

III.A.2 PAGE SEGMENTATION

The main focus of the system is on developing a font size independent technique, so this system assumes that image is clean and no skew is present in the page. Page segmentation is performed to extract the ligatures. Using the horizontal projection profiles, the page is divided into lines. The ligatures can overlap therefore ligature segmentation cannot be performed simply by applying the vertical projection profiles. For the ligature segmentation, the technique presented in [5] is used. The baseline is computed with the help of horizontal projection profile. The maximum number of black pixels of a row indicates that almost all the ligatures exist or pass through that row termed as the baseline. For the font size independent technique, the main bodies of the ligatures are required. All those connected components which pass through baselines are considered as the main bodies and the remaining are the diacritics which are ignored.

III.A.3 FONT SIZE COMPUTATION

For the methodology of the font size independent OCR, the important step is to find the font size of the ligature image so that respective scaling factor can be applied.

For Latin script the font size computation is relatively an easy task. The Latin script has one character in the connected component body. Usually font size is computed with the help of x-height computation of the connected component. Due to the cursive nature of Noori Nastaleeq script, x-height cannot give clue for the font size. This is because the x-height of ligature ب at 48 font size may be confused with the x-height of the ligature حب at 24 font size. To cope with this problem, the suitable way is to define the font size detection technique based on the Qat analysis at different font sizes. Qat is a length of width of the flat nib of the pen, used for writing Urdu scripts such as Nastaleeq by calligraphers. [17]. The Qat value for the ligatures of same font size is same. In Noori Nastaleeq not all characters have full pen tip in their outlines which are highlighted in red rectangle in the Figure 4. There are different Qat values for different font sizes as can be seen in Figure 3. In this figure the rectangle in red shows the Qat size for 24 and 36 font size ligatures.

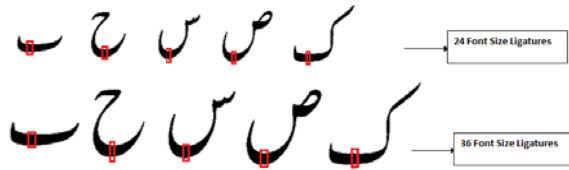


FIGURE 3. QATINDICATION AT 24 AND 36 FONT SIZE

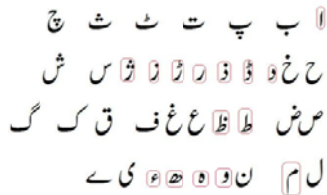


FIGURE 4. MISSING QAT INFORMATION

Different newspapers, books etc have lines which have more than one ligature which has full pen tip in its body and by processing it font size can be determined.

To find the Qat of the ligature, a window based Qat finding technique is presented. At the start 7 x 29 mask size is slid over the whole ligature body. Before traversing the window, centerline of the ligature image is computed. The values of the window are the pixels' values of the covered ligature body by the window during each traverse. The window is slid on the ligature image using the centerline such that the center of the window lies on the centerline. Actually the mask is traversed through the centerline and whole ligature body is used for the computation of the Qat size. While traversing only those image areas are stored on which following condition for the window image is met.

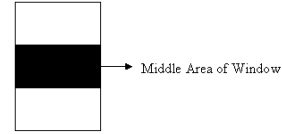


FIGURE 5. RESULTANT WINDOW SAMPLE

The Figure 6 shows the simulation of the algorithm. The Qat indication of the Urdu character Ain is represented by circle and specified by arrow.

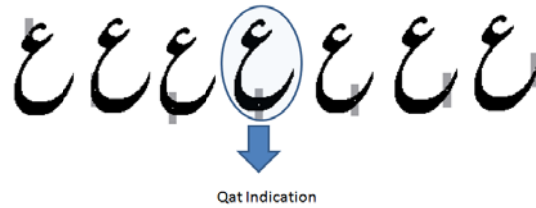


FIGURE 6. QAT INDICATION OF AIN

III.A.4 OUTLINE CAPTURING

Outline capturing is an essential step for ligature resizing. In outline capturing, ligature image is converted into the outline using splines. Many algorithms are proposed for the outline capturing in [19,31-33]. The scaling factor is applied to control points of the splines. It resizes the image without loss of shape information. Therefore a respective scaling factor can be applied to obtain the desired image size. The outline capturing has following phases.

In first phase a vectorized form of single pixel boundary (in a sequence) of the image is extracted. The chain code method is [20] is used for the boundary detection. The main reason behind the selection of this algorithm is that it can handle multiple boundary loops of image. This algorithm sorts all the boundary points according to the boundary loops. The Figure 7 shows the boundary loops for the Urdu character ligature Do-Chashmy-Hay.



FIGURE 7. DO-CHASHMY-HAY WITH THREE BOUNDARY LOOPS

Boundary detection gives the vectorized form of the image in sequence. The next step is to find the corner points of the boundary. For curve approximation, these corner points play an important role. For the proposed technique, [21] algorithm is used for corner points detection.

This algorithm has two phases. In the first phase, all those points are marked as candidate corner points which have their angle value greater than the threshold. The second phase is the elimination of all those candidate corner points which have sharper candidate corner points in their neighbours. At the end of second phase all remaining candidate corner points are detected corner points. The Figure 8 shows the detected corner points of Do-Chashmy-Hay where filled holes with blue indicate detected corner points.

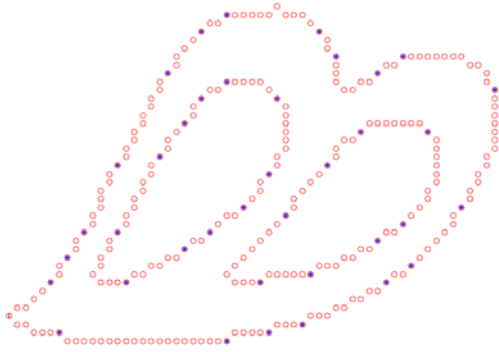


FIGURE 8. DO-CHASHMY-HAY'S CORNER POINTS

The detected corner points arrange the boundary of the ligature into the segments in order. Each segment has two corner points. The last segment has first and last corner points. The number of segments is equal to the number of corner points in the image boundary.

Now for the outline approximation each segment is taken one at a time and the curve is approximated. For the curve approximation cubic Bezier curve is used [22]. It has four control points; two of which are the end control points (or the corner point of the segment) and two are the intermediate/tangent on the end control points. The equation of the cubic Bezier curve for corner points (F_i) is as follows:

$$P_i(t) = (1-t)^3 F_i + 3t(1-t)^2 V_i + 3t^2(1-t) W_i + t^3 F_{i+1} \quad (1)$$

$$i = 1, 2, \dots, n$$

For closed curve $F_{n+1} = F_1$

$$F_0 = F_n$$

$$F_{n+1} = F_1$$

For the range of values of t the chord length parameterization is used. All the unknowns W_i and V_i parameters of eq(1) are computed using [31]. A curve of segment can be approximated using eq(1) with following steps

1. Approximate the curve for F_i and F_{i+1} using equation (1)
2. Calculate error
 - a. If Error is within the Error_Threshold value ,then
 - i. Increment ith value and repeat steps 1 and 2. for ($i=0,1,2,\dots,n$)
 - b. otherwise
 - i. Break the curve into two sub-curves at the t where highest error is found. Re-compute derivatives. Repeat step 1 and 2.

The *Error_Threshold* is 3 in this algorithm. Less error threshold introduces the more control points but tries to approximate according to the original boundary, where as reasonably large error threshold value decreases the number of control points but the approximated boundary is far from the original boundary. The Figure 9 shows the outlines approximation of Do-Chashmy-Hay. The filled holes show the corner points which are used to approximate the outline. The black line between two corner points shows that approximation error is above error threshold, indicates the scenario for the segment subdivision. The red holes show the original boundary whereas the green holes represent the approximated boundary. For multiple boundary loops, first the major boundary loop labelled "Loop1" is taken for approximation. All remaining loops are then approximated one by one. The reason behind this selection is mentioned in the section of boundary fill.



FIGURE 9. OUTLINE APPROXIMATION OF DO-CHASHMY-HAY

III.A.5 SCALING AND RE-APPROXIMATION

Once the whole curve has been approximated, the next step is to filter the control points. All those control points are removed which have no or less effect on the curve as compared to the original one. Usually there are two control points in a line but due to less error-threshold more control points are introduced in a line. For this, all those extra corner points are removed which lie in a line of the approximated curve.

III.A.6 CONTROL POINTS SCALING AND RE-APPROXIMATION

The control point filtering decreases the number of control points. The next step is the scaling and re-approximation of these control points. The scaling factor, computed in the scaling computation phase, is used. In scaling factor computation, first the whole pass of the outline capturing is applied for the computation of the scaling factor. The tentative scaling factor is applied to the outline such that the dimension of the resultant image should match the dimension of the respective ligature at 36 font size. A little tweaking is performed in the scaling factor such that the dimensions of all ligatures having the same font size match in dimensions with all the respective ligatures at 36 font size with less error. To resize the outline to the standard size, scaling factor is applied on the control points. These control points are re-approximated to get the desired size of the ligature body.

III.A.7 IMAGE GENERATION

Once the scaled outline is approximated, the next step is to perform the recognition from the Urdu Nastaleeq OCR [5]. The input of this OCR is in image form so the outline is converted into the image form. The method for the image generation is given below:

The approximated outlines are in continuous form and images are in discrete forms. To convert continuous outline into the discrete form, first maximum values of the horizontal and vertical coordinates are calculated from the outline and are stored in the *MaxHorizontal* and *MaxVertical* variables. Then the matrix of size *MaxHorizontal* x *MaxVertical* is generated. Actually this matrix is used for image generation after setting the values of its respective coordinates/dimensions. For this, the following strategy is used.

All the approximated points of boundary outline are stored in the array as $P_{xi}P_{yi}$, $i=1,2,3,\dots,n$, where n is the total number of points approximated for whole boundary. The whole arrays of the $P_{xi}P_{yi}$ are traversed and respective matrix coordinates are computed as:

1. Start from $i=1$ to n .
2. Take point at x_i and y_i positions of the arrays and store in point form as $(P_{xi}P_{yi})$. Compute upper bound of P_{xi} and P_{yi} and store in px and py respectively.
3. The 0 value is assigned at (px, py) index of the matrix.
4. The value of i is incremented by 1 and steps 2 to 4 are repeated until whole array is traversed.

Value 1 is assigned to all those indices which are unassigned after the above procedure. This matrix is then saved as an image form. The zero values of the matrix indices indicate the black pixels whereas the indices having one value indicate white pixels.

After this process, when image is generated sometimes there is discontinuity in the image outline. To cope with disconnected boundary, the following algorithm is used.

III.A.8 IMAGE BOUNDARY DISCONTINUITY RESOLVING

To resolve the discontinuity, the image is traversed starting from the upper left pixel. Each black pixel is checked, whether it is connected or not. The black pixel is tried to find which has only one black pixel in its 4-connected neighbours. If the pixel is found then the following method is applied to connect it. If the masking can be applied then remaining step is eliminated otherwise second step of Horizontal and Vertical lines are checked to apply. This step is repeated for all those black pixels which are disconnected.

Most of the discontinuity samples have single pixel discontinuity. The horizontal or vertical line scenario is checked if discontinuity is not resolved with the help of masks. This is because other than the single pixel discontinuity the remaining can be solved with the help of vertical and horizontal line. By looking at all the discontinuity samples the specific length of the vertical or horizontal line is defined. From the disconnected pixel, once the horizontal traversal is performed till a black pixel is found. The length of the horizontal traversal is computed. Horizontal line is drawn to connect that pixel if computed length of the line is not greater than the specified length, otherwise vertical scanning is performed. In the vertical scanning, onward from the disconnected pixel, the vertical traversing is performed until the black pixel is found. The length of the traversing is compared with the specified length. The vertical line is plotted if the computed vertical length is not greater than the specified length. The Figure 10 shows the boundaries of Bari-Yey which can only be solved by the horizontal and vertical lines.



FIGURE 10. HORIZONTAL AND VERTICAL LINES DISCONTINUITIES



FIGURE 11. RESOLVED DISCONTINUITIES

III.A.9 BOUNDARY FILLING

After connected boundaries are achieved, the next and last step is to fill the boundary of the image with black colour. Usually to fill the boundary of the ligature, inner and outer boundary areas are identified, which helps especially for the multiple boundary loops. The inner area is then filled with the black colour. This research presents another technique to fill

the multiple boundary loops ligature image. After the boundaries extraction of all loops, first the largest boundary loop (named ‘Loop1’) is approximated and converted into the image form. The dimensions of the ‘Loop1’ are stored, if there are more than one boundary loops. During image conversion of remaining boundary loops, the stored dimensions are used for setting the dimension of these approximated loops. Now after all the loops of the ligature image are converted into the image form, the dimensions of each loop are same. For the single boundary loop of the ligature image, the closed boundary is simply assigned black colour, whereas for the multiple boundary loops the technique is little bit different. For the multiple boundary loops, initially the major boundary loop is approximated and converted into the image. The remaining loops of the ligature are processed later. The filling algorithm is applied to each loop separately, and at the end final image is generated which represents the final ligature shape. Following procedure is applied for all the boundary loops. The XOR operator is used for this purpose. The first generated image (of Loop1) is XORed with the next generated loop image. The resultant image is again XORed with the next generated loop image (if existed). This process is repeated unless all the loops’ images are XORed. The resultant image shows the final ligature image. Before storing this image as the final, image is inverted and then saved. The following figure shows this procedure.



FIGURE 12. BOUNDARY FILLING OF MULTIPLE LOOPS

III.A.10 NOORI NASTALEEQ OCR RECOGNITION

Finally when the image form of the scaled ligature outline is obtained, the next step is to recognize this image from the OCR. For the recognition Noori Nastaleeq OCR1 is used which is still in progress. The main methodology is almost the same as [5] with little modification. In order to improve the accuracy of the [5], the segmented ligatures are thickened with the help of original shape of the ligature. The resultant segments provide more detailed features as compared to the thin segments. In addition to this all 21 character classes are also handled. It also handles the Nuqtas placement by applying the heuristics according to the Nuqtas in the original ligature body.

B. EXPERIMENTAL RESULTS AND ANALYSIS

To test the font size computation technique, total 34 unique ligature shapes which have full Qat value in their bodies are selected. Each shape has 10 samples at each font size.

Therefore for each font size, total of 340 shapes are tested. The Table 1 shows the summary of the testing results at each font size.

TABLE 1 RESULTS OF FONT SIZE COMPUTATION TECHNIQUE

Font Size	Total Samples	Correctly Recognized	Percentage Accuracy
24	340	299	87.94%
28	340	307	90.29%
32	340	316	92.94%
36	340	296	87.06%
40	340	292	85.88%
44	340	228	67.06%
44	340	303	89.11%

The overall accuracy of the font size computation technique is given in Table 2.

TABLE 2 OVERALL ACCURACY OF FONT SIZE COMPUTATION

Total Samples Tested	Correctly Identified	Accuracy
2380	2041	85.76%

III.B.1 OCR RECOGNITION

For the testing of font size normalization technique, all the single character ligatures of all the character classes are used. Two different types of testing are performed. One is the internal testing in which testing samples are generated manually and other is the external testing in which testing samples are collected from different books, magazines etc. There are some errors which are caused by the OCR itself. The Accuracy Rate is given with the two parameters. With confusion which means if the error of the OCR is considered as the error of the presented technique. Without confusion indicates the accuracy rate of only presented technique by ignoring the errors of the OCR. The detailed testing of both two different testing phases is given below.

Internal Testing

Six different font sizes which are 24, 28,32,40,44 and 48 are scaled to 36 (standard size) and then are recognized by the OCR [5]. Additionally 16, 20, 52 and 56 and 60 font sizes are also tested to see the error flow of the technique. Ten samples

¹ Internally developed at CRULP

of each class are taken and total of twenty one classes are tested. Following table shows the overall results of recognition for each font after the scaling to 36 font size.

TABLE 3. OVERALL ACCURACY OF INTERNAL TESTING

	Total Samples	Accurate	Accuracy Rate
With Confusion	1260	1212	96%
Without Confusion	1260	1240	98%

External Testing

In external testing, samples of the isolated character ligatures are obtained from different books and magazines [24-30]. The most used font sizes for the isolated character ligatures are 24, 28, 32 and 40. The overall accuracy is given in Table 4.

TABLE 4. OVERALL TESTING RESULTS OF EXTERNAL TEST SAMPLES

	Total Samples	Accurate	Accuracy Rate
With Confusion	569	496	87%
Without Confusion	569	548	96%

C. DISCUSSION AND FUTURE DIRECTIONS

Testing of two different techniques is performed. The font size computation and rescaled ligature recognition. Results of both techniques are looking promising. The error analysis of each is given below:

Font Size Computation

As a pioneer effort to compute the font size of the Noori Nastaleeq ligature, the results are reasonably good. The cause of errors is due to the inconsistency of the font. The Qat value at one font size is not consistent. This type of error can be overcome with the help of some other heuristics, such as the height value of character Alif(l) because it is the most occurring character in the Urdu text. Almost each line must have at least one occurrence of l. The second way is to use an additional parameter which is the Nuqta. It can clearly indicate the font size, but for the small font size, these Nuqtas are distorted. In testing, each character ligature's font size is tried to compute. This can be further improved, if the focus is on font size computation of a line. The Qat value is selected, which is most occurring in a line. This will definitely indicate correct font size of ligatures in a line.

OCR Recognition

The testing results for both internally developed and externally gathered samples are convincing. Following are the major problems which caused the errors in the recognition.

Sometimes especially for the small font sizes, the original image boundary is distorted. This is because for the small size, the shape of the ligature image does not preserve all the shape features. The distorted image is due to the poor scanning which caused error. The small noise is prominent when image is enlarged by applying the scaling factor.

The noise due to jagged boundary can be solved by applying boundary smoothing techniques such that sharp corners of the ligature can be reserved. There is another way to solve this problem. After applying some heuristics on the approximated control points, this type of error can be reduced. For this, all the consecutive two control points should be removed such that the shape of the image remains the same.

The majority of the errors are due to the last phase, which is the image generation from the scaled outline. As mentioned above during the conversion to the image, some discontinuity issues occurred. These issues are solved by applying some rules. These rules do not properly solve the issues of discontinuity. At some places, some over connected pixels are inserted which distort the shape of the ligature. Moreover sometimes the converted boundary still remains disconnected.

By looking at the results of the discontinuities in the images, there is a need for the more refinements of the masks, horizontal and vertical lines techniques such that discontinuity issues can be resolved. Second way to solve this issue is to devise other rasterization algorithm which can produce the mirror image of scaled outline.

Another reason for the errors in the OCR recognition is the distance of the font sizes from the standard 36 font size. As far as we move from 36 font size to above or below, the accuracy rate decreases. For further verification, samples of 16,20,52,56 and 60 font sizes are tested. The results of all the font sizes ranging from 16 to 60 are plotted in a graph which is given in Figure 13.

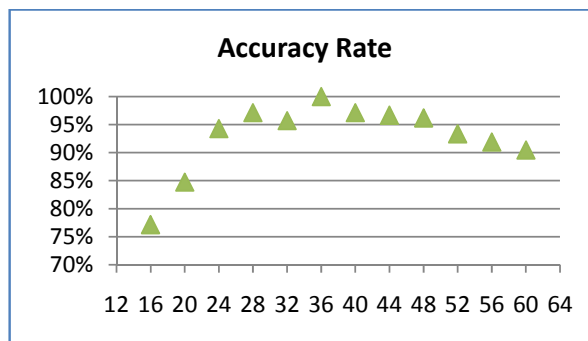


FIGURE 13. FONTS RECOGNITION GRAPH

As can be seen from the above figure that the recognition rate ranges from 94% to 97% for font sizes 24, 28 and 32 when they are scaled up to 36 font size. For the large font sizes such as 40, 44, 48 and 52, the recognition rate lies between 93% and 97% when these are scaled down to 36 font size. The possible solution for this issue is to train the OCR at different 2 to 3 font sizes so that this type of error can be reduced. It means that train the OCR at 24, 36 and 48 font sizes and then resizing is applied to the respective nearest trained font size. For example if the 16 font size ligature is input for the recognition then after outline approximation, it is resized to the 24 font size ligature. In the same way if the ligature of 44 font size is needed to be recognized, then it is rescaled to nearest font size which is 48.

The presented technique is tested for single character ligatures. This technique can be extended for multiple character ligatures as well. Testing is performed for some 2 character ligatures and 89% accuracy rate is achieved. To improve the accuracy for multiple character ligatures a little tweaking can be performed.

This technique can also be applied for other languages' scripts for font size independent OCR. A little effort will be required for this because the outline capturing process will be same. Only scaling factor is needed to be changed with respect to font size for different scripts.

REFERENCES

- [1] Pal and Anirban Sarkar, "Recognition of Printed Urdu Script", "Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)"
- [2] Zahra Shah and Farah Saleem, "Ligature Based Optical Character Recognition of Urdu, Nastaleeq Font", Multi Topic Conference, 2002. INMIC 2002. International, 2002.
- [3] Sobia Tariq Javed, Ameera Maqbool, Sehrish Jameel and Samia Asloob Qureshi, "Urdu Nastaleeq OCR", BS final year report, 2005
- [4] Syed. Afaq Husain and Syed. Hassan Amin, "A Multi-tier Holistic approach for Urdu Nastaliq Recognition"
- [5] Sobia Tariq Javed. Investigation Into A Segmentation Based OCR For The Nastaleeq Writing Style. Master's Thesis, NUCES, August 2007
- [6] L. Pratap Reddy and L. Satyaprasad and A. S. C. S. Sastry, "Middle Zone Component Extract and Recognition of Telugu Document Image", ICDAR, IEEE Computer Society, page 584-588. (2007).
- [7] Aparna.K.G and A.G.Ramakrishnan, "Tamil Gnani - an OCR on Windows," Proc. Tamil Internet 2001, Kuala Lumpur, August 26-28, 2001, pp. 60-63.
- [8] MNSSK Pavan Kumar, S. S. Ravikiran, Abhishek Nayani, C. V. Jawahar and P.J. Narayanan, "Tools for Developing OCRs for Indian Scripts", Proceedings of the Workshop on Document Image Analysis and Retrieval (DIAR:CVPR'03) Jun. 2003, Madison, WI
- [9] Tapas Kanungo , Philip Resnik , Song Mao , Doe-Wan Kim , Qigong Zheng, The Bible and multilingual optical character recognition, Communications of the ACM, v.48 n.6, p.124-130, June 2005
- [10] Mehran, R., Pirsiavash, H., and Razzazi, F. 2005. A Front-End OCR for Omni-Font Persian/Arabic Cursive Printed Documents. In Proceedings of the Digital Image Computing on Techniques and Applications (December 06 - 08, 2005). DICTA. IEEE Computer Society, Washington, DC, 56.
- [11] Lakshmi C V, C Patvardhan, 2002 A multi-font OCR system for printed Telugu text. Proc. Of Language engineering conference LEC, Hyderabad.pgs.7-17.
- [12] Lehal G S and Chandan Singh 2000 A Gurmukhi Script Recognition System 15th International Conference on Pattern Recognition (ICPR'00) - Volume 2 p. 2557.
- [13] Di Zenzo, S., Del Buono, M., Meucci, M., and Spirito, A. 1992. Optical recognition of hand-printed characters of any size, position, and orientation. IBM J. Res. Dev. 36, 3 (May. 1992), 487-501.
- [14] T V Ashwin and P S Sastry "A font and size-independent OCR system for printed Kannada documents using support vector machines", Department of Electrical Engineering, Indian Institute of Science, Bangalore 560 012, India.
- [15] Pujari Arun K , C Dhanunjaya Naidu & B C Jinaga 2002 An Adaptive Character Recognizer for Telugu Scripts using Multiresolution Analysis and Associative Memory. ICVGIP, Ahmedabad.
- [16] R.Smith, "An Overview of the Tesseract OCR Engine", Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR), 2007, pp. 629-633.
- [17] 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.
- [18] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Systems, Man, and Cybernetics 9(1), pp. 62-66, 1979.
- [19] Sarfraz, M. and Khan, M. A. 2004. An automatic algorithm for approximating boundary of bitmap characters. Future Gener. Comput. Syst. 20, 8 (Nov. 2004), 1327-1336.Sonka, M., Hlavac, V., and Boyle, R. (1993) Image Processing, Analysis and Machine Vision, Chapman and Hall.
- [20] Chetverikov D, Szabo Z. A simple and efficient algorithm for detection of high curvature points in planner curves. In: Proceedings of 23rd workshop of Australian pattern recognition group. Steyr, 1999, p. 175-84.
- [21] Farin Gerald., 2002. Curves and Surfaces for CAGD, A Practical Guide. Morgan-Kaufmann, 5th edition, 2002, ISBN 1-55860-737-4.
- [22] Inam Shamsher et.al, OCR For Printed Urdu Script Using Feed Forward Neural Network, Proceedings of World Academy of Science, Engineering and Technology. Vol 23, Aug 2007 ISSN 1307-6884
- [23] "عبدالملکیری"، اکرم شیخ/فہمیدہ کوثر
- [24] ساگر پلیشرز، "مذہبی سیاست کے تقاضات"، سبیل وراچ
- [25] اے عشق جنوں پیٹھ، احمد فراز، دوست پبلیکیشنز
- [26] سانجھ پبلیکیشنز، "قدیم ہندوستان"، ڈاکٹر مبارک علی
- [27] نومبر ۲۰۰۷، مقتدرہ قومی زبان اسلام آباد، "ماہنامہ اخبار اردو"
- [28] سنگ میل پبلیکیشنز، "نکر شہاب"
- [29] قائداعظم پیپرز پروجیکٹ کینٹ ڈویژن، حکومت پاکستان، "تمہید پاکستان"
- [30] Sarfraz, M. and Khan, M. A. 2002. Automatic outline capture of Arabic fonts. Inf. Sci. Inf. Comput. Sci. 140, 3 (Jan. 2002), 269-281.
- [31] Sarfraz, M. and Razzak, F. A. 2003. A web based system to capture outlines of Arabic fonts. Inf. Sci. Inf. Comput. Sci. 150, 3-4 (Apr. 2003), 177-193.
- [32] Muhammad Sarfraz , Aiman Rashid, A randomized knot insertion algorithm for outline capture of planar images using cubic spline, Proceedings of the 2007 ACM symposium on Applied computing, March 11-15, 2007, Seoul, Korea
- [33] Masood, A. and Sarfraz, M. 2009. Capturing outlines of 2D objects with Bézier cubic approximation. Image Vision Comput. 27, 6 (May. 2009), 704-712.