

Study and Analysis of various preprocessing approaches to enhance Offline Handwritten Gujarati Numerals for feature extraction

Hetal R. Thaker*, Dr. C. K. Kumbharana**

* Assistant Professor, Department of MCA, Atmiya Institute of Technology & Science, Rajkot, India

** Head, Department of computer science, Saurashtra University, Rajkot, India

Abstract- Since last many years Optical character recognition has been an area attracting many researchers. Due to wide range of applications and advancement of digital technology offline and online handwritten character recognition for regional script is becoming fascinated area of research. In any character recognition system feature extraction phase requires input of image which is noise free, binary and having only region of interest. Main objective of Enhancement of image (EOI) phase is to process image in a way which gives more appropriate result than original acquired image for further steps in character recognition. This phase has high influence and hence plays a vital role in any character recognition system. Wide choices are available for digital image enhancement for enhancing visual quality of image. Choosing appropriate approach for image enhancement is a significant step. This paper discusses various image enhancement approach, analyzes them on the basis of result obtained by experimenting on sample handwritten image dataset at every step so as to provide suitable input for feature extraction for recognizing Offline Handwritten Gujarati Numerals.

Index Terms- Image Preprocessing, Preprocessing handwritten images.

I. INTRODUCTION

Pattern recognition is a branch of artificial intelligence where an objective is to recognize pattern or an identification of faces, objects, words, character etc. Character recognition is an area which can be categorized into two ways as: (1) Online and Offline (2) Machine printed and Handwritten (Figure 1). In online character recognition characters are recognized as soon as user writes using digitizer or PDA where as in offline character recognition characters are recognized from images acquired by camera or digitized using scanner. Digitized document may contain handwritten character or characters printed using computer font and is classified accordingly.



Figure 1 Classification of Character Recognition

Offline Handwritten character recognition is an area attracting many researchers. Six important steps are employed in any character recognition task these are: Preprocessing, Segmentation, Feature extraction, Classification and Post processing. Preprocessing step is a preliminary step to be performed on acquired image, which involves certain operations to be performed and hence providing a necessary base to perform further tasks of character recognition. If image is noisy or it is not in a proper format then it directly affects the performance of character recognition. Preprocessing does the task of enhancing image making it suitable input for segmentation or feature extraction. In this paper authors have presented analysis of various approaches for image enhancement. Paper is divided into various sections as Previous work, Image Enhancement approaches, Prototype sequence for image enhancement followed by conclusion.

II. PREVIOUS WORK

Hsin-Chia Fu et.al. [1] have employed series of image preprocessing steps that includes smoothing of boundary, removing noise, normalization of space, thinning of stroke.

To eliminate noise and for simplified procedure of feature extraction N.AZIZI et. al. [2] have proposed some preprocessing approach which is script independent such as Normalization, Contour smoothing, base line detection etc. in order to extract structural features.

To remove salt and pepper noise median filter is used by J. R. Prasad et.al.[3] in their work and have to reduce character to minimum one pixel thickness thinning is applied.

For converting image to binary image N.Shanthi et. al.[4] have applied Otsu's global thresholding method on image and for skeletonization hilditch algorithm is applied.

Apurva A. Desai[5] has presented his work to recognize Gujarati numerals where some preprocessing approaches employed includes Contrast limited adaptive histogram equalization for contrast adjustment, smoothing boundaries using median filter and for making all scanned images in uniform size nearest neighbor Interpolation algorithm is used. For removing skew numerals are rotated upto 100 about center point and created five patterns rotating numeral images in both direction ie. Clock wise and anti-clock wise.

R. Kannan et. al.[6] in their work of recognizing Tamil handwritten characters have applied preprocessing techniques where to extract foreground ink from background image thresholding is used, to remove noise median filtering and wiener filtering is used and for detecting skew angle, calculation of cumulative scalar product of windows of text blocks using gabor filters at different orientation and have employed on all possible 50X50 windows and skew angle was found as median of all angles obtained.

Normalization process in which slant correction, width normalization and normalizing height of three zones using vertical scaling is utilized by R.Kannan et. Al.[7]

III. ANALYSIS OF VARIOUS IMAGE ENHANCEMENT APPROACHES

This section discusses various image enhancement approaches and analyzes them on the basis of result obtained.

A. Handwritten Image Dataset

For experimental work 750 handwritten isolated Gujarati numerals were collected and digitized using Brother DCP-7030 scanner at 300 dpi in png format. Figure 2 shows variation in handwriting Gujarati numeral five (pronounced as 'panch') also variation arise as writers have used their pen.

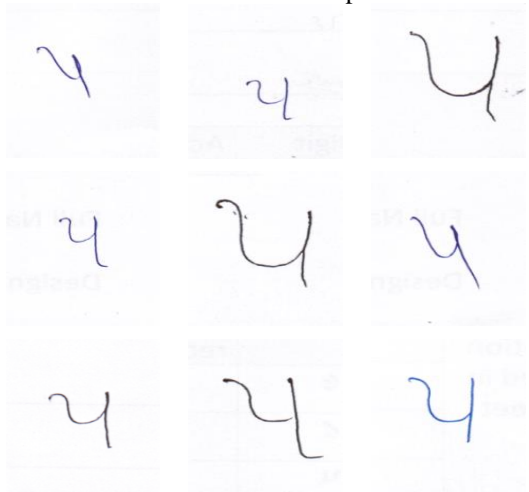


Figure 2: Variation in writing Gujarati Numeral 'five'

For demonstrating result on applying various approaches one sample image from Figure 2 is used.

B. Convert rgb image to grayscale

Color of a pixel in RGB image is determined by amount of Red, Green and Blue, which is a stack of three matrices representing color proportion of RGB. Hence for every pixel one can trace three value. Converting this image into Grayscale where every pixel will have shade of gray. In a conversion process of RGB to grayscale hue and saturation is eliminated and luminance is retained. Grayscale occupies less memory space as compare to RGB image as each pixel is representing eight bits information. $V1 * R + V2 * G + V3 * B$ equation is used to converting RGB image to Grayscale where R,G and B represents

Red, Green and Blue value of a pixel and V1, V2 and V3 are real values, variation in which yields following result as shown in figure 3. The process will be repeated for every pixel of a image to obtain Grayscale image.

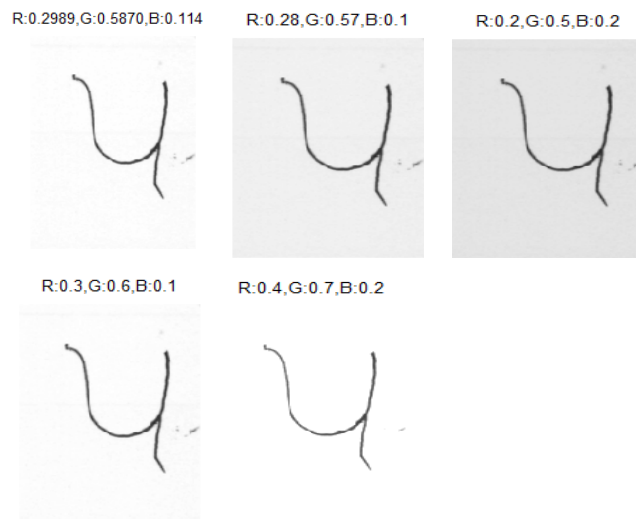


Figure 3 Grayscale Images with variation in for R,G and B

C. Contrast adjustment

Histogram Equalization is a method used to enhance contrast of an image. histeq enhances the contrast of images by transforming the values in an intensity image, or the values in the colormap of an indexed image, so that the histogram of the output image approximately matches a specified histogram. [8] as per figure 4(a)

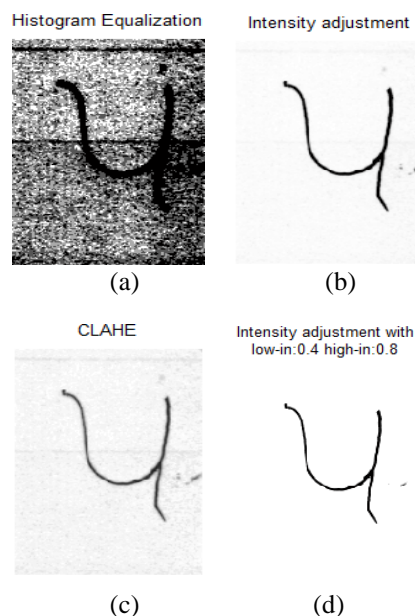


Figure 4 (a) Histogram Equalization (b) Intensity adjustment (c) Contrast Limited Adaptive Histogram Equalization (d) Intensity Adjustment with low_in:0.4 and high_in:0.8 parameter values

Another approach is to adjust image intensity values here by 4 values are define i.e. low_in, high_in, low_out, high_out and values below low_in and high_in are clipped. Resultant image

obtained will have values between low_in and high_in map to values between low_out and high_out. Variation in limits yields results as show in figure 4(b) and 4(d). Contrast limited adaptive histogram equalization (CLAHE) is a method in which entire image is divide into smaller parts and histogram equalization is applied to all small parts and then interpolates the result as shown in figure 4(c).

D. Sharpening and Reducing Noise

For sharpening image predefine 2D filter ‘unsharp’ is used which is also known as unsharp contrast enhancement filter which creates filter from the negative of Laplacian filter with default alpha value 0.2. this filter is applied to contrast adjusted image result obtained is as per figure 5(a) . To remove noise and preserve edges Median filter is applied as shown in figure 5(b).

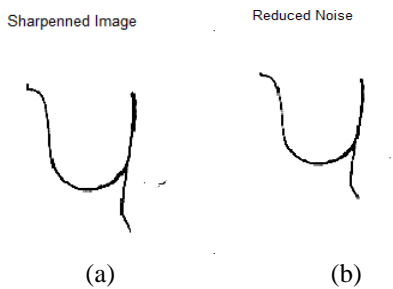


Figure 5 (a) Sharpened image (b) Reduced noise in image

E. Binarization of Image

To convert an image to binary image requires determining appropriate threshold value. In binary image pixel will have value either 0 or 1. When grayscale image is converted to Binary image luminance value above threshold value will be converted to 1 and below it will be converted to 0. Figure 6 shows result obtained as a result of variation in threshold value.

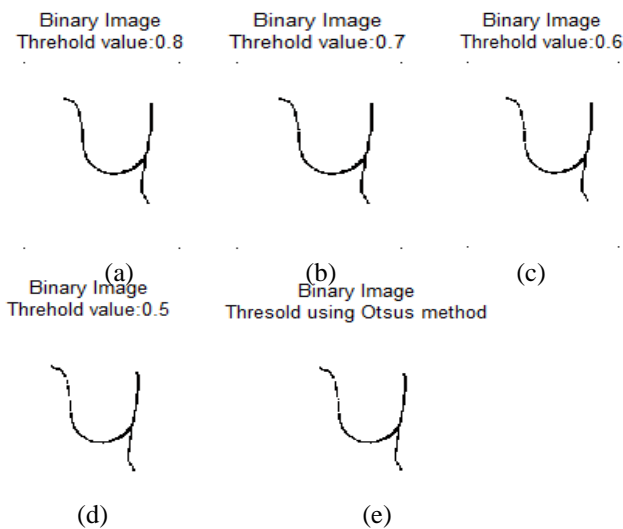


Figure 6 (a) Threshold Value:0.8 (b) Threshold value:0.7 (c) Threshold value:0.6 (d) Threshold value:0.5 (e) Global threshold using Otsu’s method.

F. Morphological Operations

To fill the gaps in binary image so features can be extracted accurately ‘line’ structuring element is created and morphological close operation is applied. Figure 7(a) shows output of the same. To remove small objects morphological open operations is used components connected less than 8 components are removed as shown in figure 7(b).

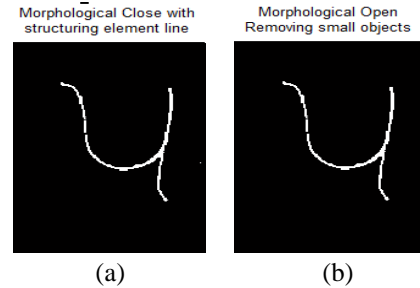


Figure 7 (a) Structuring element line and morphological close operation (b) Removing small objects with morphological open

G. Detecting Boundary

To extract region of interest boundary needs to be framed for which edges needs to be detected. To crop region top-left and bottom-right values are identifies by row wise scanning pixel values for its value 1. After identify boundaries image is clipped as per identified coordinates, as per figure 8.



Figure 8 Detecting boundary and cropping region to obtain desirable region for feature extraction

H. skeletonizing and Thinning

Skeletonizing is removing pixel on the boundaries of object without breaking object. Result of skeletonizing is shown in figure 9(a). Thinning reduces lines to single pixel thickness as shown in figure 9(b).

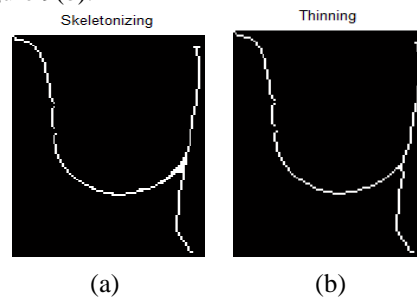


Figure 9 (a) Skeletonizing image (b) Thinning operation on image

IV. PROTOTYPE SEQUENCE OF STEPS FOR IMAGE ENHANCEMENT

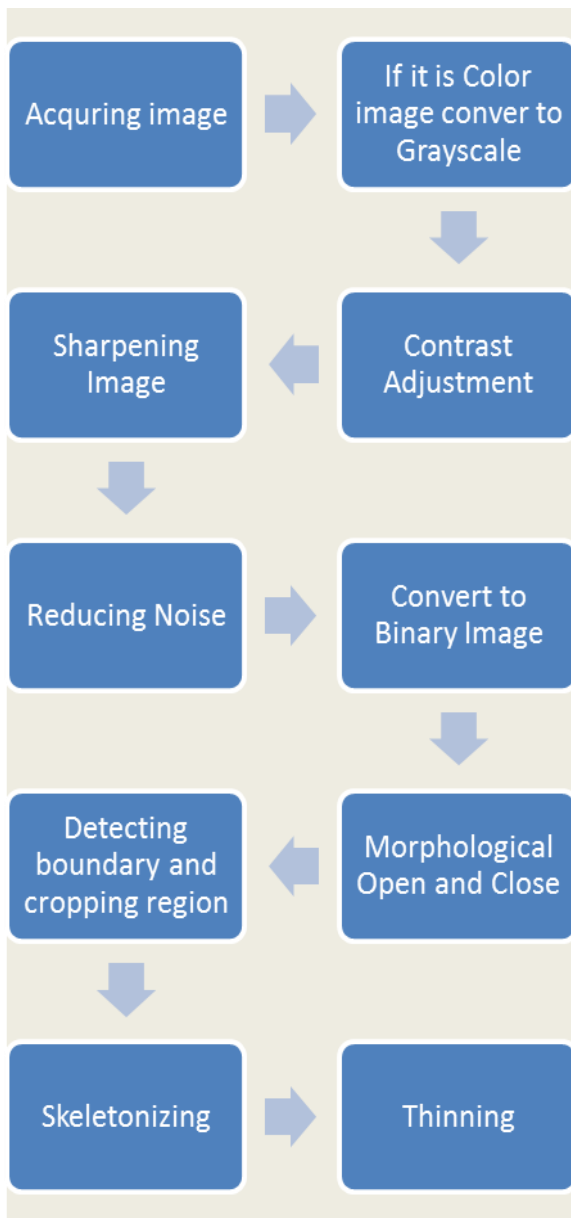


Figure 10 Series of steps for enhancing image

Figure 10 represents series of steps where input will be acquired scanned image and output will be an image suitable for feature extraction. Principal goal of this processing flow is to have an image which is highly suitable for character recognition task. Accuracy of feature extraction highly depends on the image given as input if image is noisy and clumsy it will be very difficult to obtain precisely features from character and hence this complexity is carry forward to classify it and as a result sometimes correct output cannot be obtained. It becomes mandatory to choose correct sequence so as to obtain desired image for feature extraction.

Figure 11 shows some sample handwritten numbers written in Gujarati script where series of steps from prototype modal is applied.

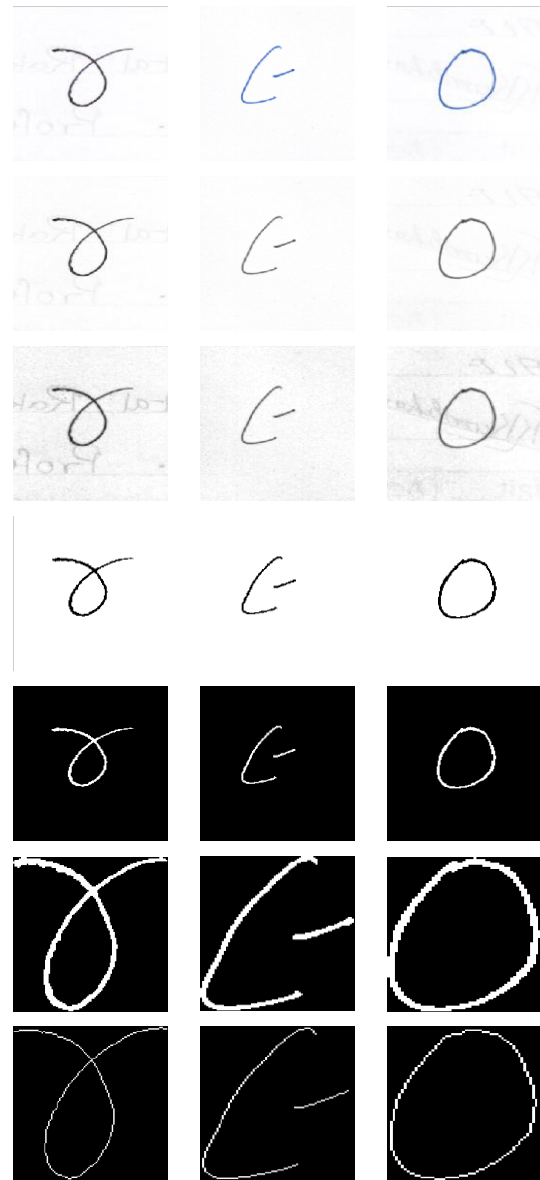


Figure 11 : Enhancement of sample handwritten images for feature extraction

V. CONCLUSION

Converting RGB image to Grayscale requires right blend of weighted sum for R, G and B in pixel. Higher value gives darker shade and lower value gives lighter shade. For contrast adjustment intensity can be set by choosing appropriate parameter values and depends on nature and source of an image. To remove salt and pepper noise median filter yields better result. If threshold value is higher than some pixels are lost. It is better to determine graythresh level of an image as a threshold value to convert image into binary image. Boundaries can be extracted accurately if image is noise free. To obtain structural features from image such as cross point, end point it requires

image to be skeletonized or thinned. Thinning yields better result than skeletonizing approach. Both operations can be applied turn by turn to achieve better result. Depending on nature of image and task at hand prototype sequence of steps presented in this paper can be included or excluded to obtain desired result.

ACKNOWLEDGMENT

Authors are very much thankful to all the writers who have contributed for providing handwritten input for proposed experimental work.

REFERENCES

- [1] H. C. Y. Y. X. H.-T. P. Hsin-Chia Fu, "User Adaptive Handwriting Recognition by Self-Growing Probabilistic Decision-Based Neural Networks," *IEEE TRANSACTIONS ON NEURAL NETWORKS*, vol. 11, no. 6, 2000.
- [2] N. F. M. S. N. AZIZI, "off-line handwritten word recognition using ensemble of classifier selection and features fusion," *Journal of Theoretical and Applied Information Technology*, pp. 141-150, 2010
- [3] J. R. Prasad, U.V. Kulkarni, 2003. Offline Handwritten Character Recognition of Gujrati Script using Pattern Matching, *Computer Engineering*
- [4] N. Shanthi, K. Duraiswamy, 2010, A novel SVM-based handwritten Tamil character recognition system, *New York*, pp. 173-180.
- [5] Apurva A. Desai, 2010 Gujarati handwritten numeral optical character reorganization through neural network, *Pattern Recognition*, vol. 43, pp. 2582-2589.
- [6] R. J. Kannan, R. Prabhakar, 2008. Off-Line Cursive Handwritten Tamil Character Recognition, *Signal Processing*, vol. 4, no. 6, pp. 351-360
- [7] R. Jagdeesh Kannan, R. Prabhakar, 2008 An Improved Handwritten Tamil Character Recognition System using Octal Graph, Department of Computer Science and Engineering , Department of Computer Science and Engineering , Coimbatore Institute of Technology , Co, *Journal of Computer Science*, vol. 4, no. 7, pp. 509-516.
- [8] "<http://nf.nci.org.au/facilities/software/Matlab/toolbox/images/histeq.html>," [Online].

AUTHORS

First Author – Hetal R. Thaker, Assistant Professor, Department of M.C.A, Atmiya Institute of Technology & Science, Rajkot, India, e-mail: hrt.research@gmail.com.

Second Author – Dr. C.K.Kumbharana, Head, Department of Computer Science, Saurashtra University, Rajkot, India, e-mail: ckkumbharana@yahoo.com

Correspondence Author – Hetal R. Thaker, hrt.research@gmail.com, hrthaker@gmail.com, Cell: +91 9726931780