

# Analysis of Image Pre-processing in Noisy Character Recognition

Priyanka Gupta

<sup>1</sup>Asst. Professor, CVR College of Engineering/ CSE Department, Hyderabad, India  
Email: prionnet@yahoo.com

**Abstract:** This paper demonstrates the importance of various pre-processing steps for the noisy character recognition system with the help of MATLAB simulated results. Image pre-processing is one of the first and foremost steps for working with images of noisy characters among handwritten characters, printed characters, online written characters or offline written character, and it can have mixed font characters, omni-font characters, boxed discrete character, spaced discrete character or pure cursive script writing. Pre-processing of such image aims at image noise reduction, image normalization, and image compression. This paper shows that there are various steps/methods of image pre-processing. The sequence and use of a particular step/method is used based on the application requirement. The simulation result presented in this paper highlights the usability of each pre-processing steps and review the commonly used methods of each preprocessing stage.

**Index Terms:** Noisy Character, Thresholding, Noise, Character Recognition, image pre-processing

## I. INTRODUCTION

Noisy character recognition is one of the classical area of pattern recognition research for many years. Researchers have used many methods for recognizing the noisy text. Considering the significance image preprocessing, in this paper we have reviewed major methods used for various application. The main purpose of this paper is to present comprehensive literature of all image pre-processing methods at one place to serve as reference to many other researchers. The significance of each step is demonstrated with the help of simulation result as per the parameters listed in Table 1.

### A. Noisy character recognition design

The stages for the noisy character recognition system from a character pixel to text recognition are as follows:

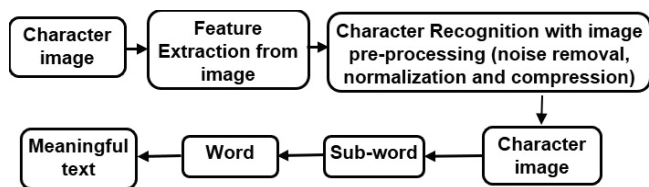


Figure 1. Various steps of the character recognition system

From the literature survey [1-4], it has been concluded that the overall neural network character recognition process can be grouped into the various stages such as: image pre-processing, image segmentation, image feature extraction, training of the network and recognition of character.

Depending upon the desired system output and application, stages in the character can be combined or omitted.

### B. Pre-processing

Out of the various stages in the noisy character recognition system, image pre-processing [3-4] of noisy character is foremost and necessary step to make the noisy character image appropriate for the further processing of image such as: segmentation and feature extraction algorithms. The main objective of image pre-processing step to make the task of the character recognition system easy and helps improve the overall accuracy by providing clean images. The main steps involved in the image pre-processing are: noise reduction, normalization, and compression.

**Noise Reduction:** Noise may get incorporated into the character image either during the process of scanning the document from optical scanning devices or during the writing of the document. The different persons have a slightly different style of writing the same character; as a result, they have varied line-disconnected segments, character-bumps, gaps between lines and filled loops, etc., Such noise can be reduced by filtering and morphological operations.

**Normalization:** Normalization is used to get the standardized data from the set of raw data acquired from various sources by reducing the variations that has been got during writing. The basic normalization techniques are: size normalization, slant normalization, and skew normalization.

**Compression:** Compression is used to improve speed of character recognition by reducing the size of the input data. To meet this image thinning and image thresholding algorithms are used apart from classical lossless image compression technique.

## II. NOISE REDUCTION

Noise reduction is the process by which we can remove noise from the acquired character image. Images may have noise from different sources such as an optical scanning device, writing instruments, etc. It is mandatory to remove the noise from these characters images in order to get a clean image for the character recognition system. Depending upon the acquired images, various types of noise removal techniques are used. In this section, it has been implemented the various commonly used noise reduction techniques and demonstrated the suitability of each technique with MATLAB simulation results.

**A. Smoothing**

Smoothing of the image character helps in acquiring the important pattern of data by removing the finer pattern. During image smoothing, pixel intensity levels are smoothed by modifying the data points in an image by some kind of local average intensity level of the surrounding data point pixel of. Smoothing may be used in two ways for the data analysis of the acquired characters.

In the case of smoothing assumption is correct, it will help to extract the better information. Also, help in identifying the data-points, which are flexible and robust. Image smoothing can be done with the help of various algorithms. Usually, image smoothing is done through density estimators like the histogram. Apart from these, smoothing also may be done using various filtering techniques.

Out of the various kinds of filters, Gaussian smoothing filters [5-6] are most widely used to reduce the noise. Gaussian smoothing filters are generally isotropic in nature and have the same standard deviation along with both axes. The scale of smoothing can be chosen by taking a suitable value of sigma. In [6], Gaussian mixture model Symmetric Smoothing Filter (GSF) has been designed to improve the performance of the Gaussian-smoothing filter further. Smoothed image using a normal Gaussian smoothing filter is given in Fig.3.



Figure 2. Input image of handwritten characters



Figure 3. Smoothen image of Fig. 2 using Gaussian smoothing filter with  $\sigma = 2$ .

**B. Sharpening**

Image sharpening is used for enhancing the character’s edges and other finer details. To improve the sharpness of the original image, a high pass filter version of the original image is added to the same images. Mathematically, image-sharpening operation in the spatial domain can be represented as

$$O(i, j) = I(i, j) + \lambda H\{I(i, j)\} \quad (1)$$

Where  $I(i, j)$  is a spatial representation of the original image at  $(i, j)$  coordinate,  $H$  is the high-pass filter operator,  $\lambda$  is a sharpness tuning parameter greater than or equal to zero, and  $O(i, j)$  is the output sharpened the image at the

same coordinate. Depending upon the sharpness required  $\lambda$  value can be suitably chosen. Generally, increasing  $\lambda$  yields a more sharpened image.

Unsharp masking [7-8] is one of the most widely used technique for image sharpening. In this technique, image enhancement is done by subtracting a blurred version of the image from itself. Fig. 4 show the sharpen image using unsharp masking.



Figure 4. Sharpened image of Fig. 2 using the unsharp-masking filter.

Sharpening of the image can be further improved by the use of adaptive unsharp masking [7] or blurriness guided unsharp masking [8].

**C. Thresholding**

Scanned character images are represented in gray level format. In the gray level representation intensity level of each pixel of the image, varying between 0 to 255. Hence, various shades of gray are possible between these two values. To remove this ambiguity gray level image is transformed to a binary image using thresholding or binarization. During this process, the level is restricted to ‘1’ or ‘0’, indicating the presence or absence of the character pixel. This conversion is done by selecting a suitable threshold intensity level, above the threshold level intensity are marked as character-background ‘1’ and below the intensity level marked as ‘0’ character image or vice versa.

Depending on the input character image various thresholding algorithms can be: global [9] or locally adaptive methods [10]. In the case of the global adaptive thresholding algorithm, a single threshold intensity is calculated for the overall scanned characters image. While, in the case of the locally adaptive algorithm, the various threshold levels are possible on the variation of pixel level in the neighborhood of the particular character. Fig 5. and Fig. 6 shows the output image using global and locally adaptive thresholding technique respectively.

**D. Contrast adjustment**

Perceptivity of the particular region in the character image can be improved by contrast adjustment. During this process, contrast is improved by increasing the pixel intensity value difference between the desired region and its background. In contrast adjustment, intensity values of the images are remapped to the full display range of the data type. A good contrast image has a large difference between the intensity values of black and white representation. Contrast adjustment can be done with intensity value remapping based on various statistical parameters.

Image intensity adjustment [11], histogram equalization and adaptive histogram equalization [12-13] are a few of

popular contrast adjustment. Fig. 7, Fig. 8 and Fig. 9 show the contrast adjusted image using intensity adjusted, histogram equalization and adaptive equalization technique.



Figure 5. Binary image of Fig. 2 gray image using global thresholding

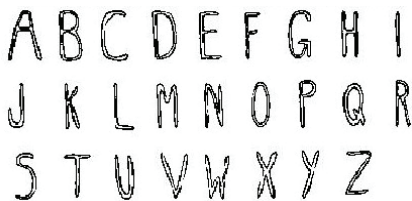


Figure 6. Binary image of Fig. 2 image using locally adaptive thresholding



Figure 7. Contrast adjusted image of Fig. 2 image using intensity adjustment

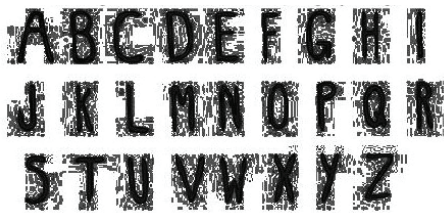


Figure 8. Contrast adjusted image of Fig. 2 gray image using histogram equalization

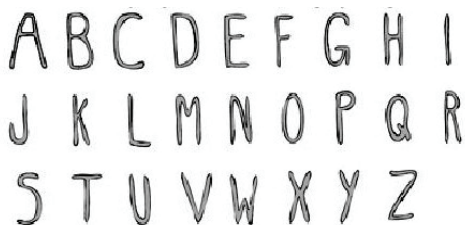


Figure 9. Contrast adjusted image of Fig. 2 gray image using adaptive histogram equalization

### E. Edge detection

Edge detection is used to detect a sudden change in the intensity level. Edge is associated with the region of sudden discontinuities. There are various methods based on the threshold value of gradient [14-15] or zero crossing for edge-detection [14-15]. The most popular methods based on the threshold value of gradient are: Sobel, Prewitt, Roberts,

as demonstrated in Fig. 10, Fig. 11 and Fig. 12, and zero-crossing are: zero crossings after Laplacian of Gaussian filter/any other filtering of the input image as shown in Fig. 13 and Fig. 14. However, both of these methods are likely to be fooled by noise in case of weak edges. In [16], the Canny method is using a gradient calculated from the derivative of Gaussian filter and based on two thresholding for weak and strong edges, hence it is difficult to be fooled by noise as shown in Fig. 15.

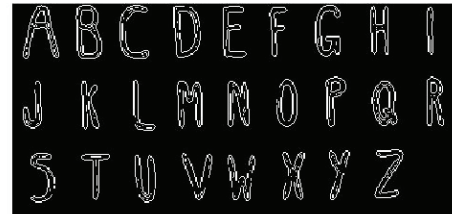


Figure 10. Edge detection of Fig. 2 gray image using the Sobel method

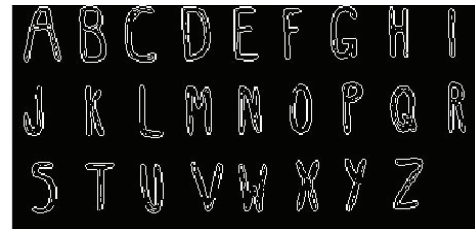


Figure 11. Edge detection of Fig. 2 gray image using Prewitt method

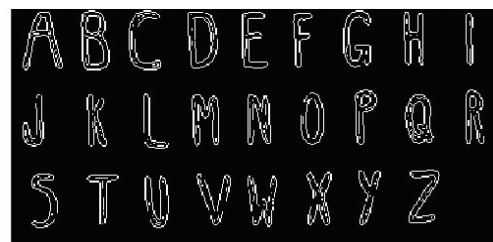


Figure 12. Edge detection of Fig. 2 gray image using the Roberts method



Figure 13. Edge detection of Fig. 2 gray image using the zero-cross method

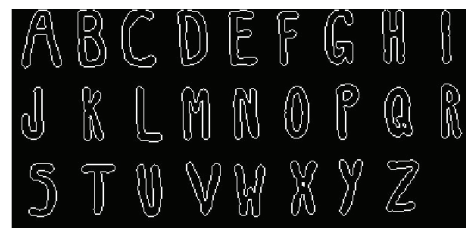


Figure 14. Edge detection of Fig. 2 gray image using zero-crossing of Laplacian of Gaussian filter

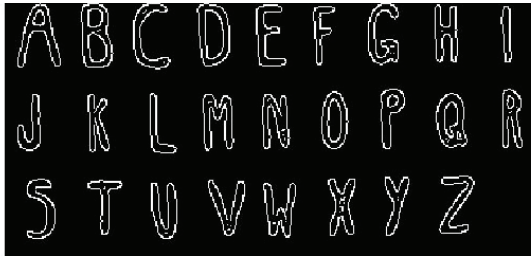


Figure 15. Edge detection of Fig. 2 gray image using Canny method

**F. Erosion**

Erosion is a morphological image processing technique [17] for a binary image I, erosion by a structuring element ‘S’, produces an output image ‘O’, with ‘1’ in all the locations of (x,y) of ‘S’ at which ‘S’ fits ‘I’.

$I(x,y) = 1$  is ‘S’ fits ‘I’ and ‘O’ otherwise, repeating for all pixel coordinates (x,y). Dilation removes a layer of pixels to both the inner and outer boundaries of the region. Fig. 16 demonstrates the erosion of the input gray image with ‘holes’ as structure element.

$$O = I \ominus S \tag{2}$$



Figure 16. Eroded image of Fig. 2 gray image

**G. Dilation**

Dilation is a morphological image processing technique [18], for a binary image ‘I’, erosion by a structuring element ‘S’, produces an output image ‘O’, with ‘1’ in all the locations of (x,y) of ‘S’ at which ‘S’ hits ‘I’.

$I(x,y) = 1$  is ‘S’ hits ‘I’ and ‘O’ otherwise, repeating for all pixel coordinates (x,y). Dilation adds a layer of pixels to both the inner and outer boundaries of the region. Fig. 17 shows the erosion of the input gray image with ‘square’ structure element of size ‘2’.

$$O = I \oplus S \tag{3}$$



Figure 17. Dilation image of Fig. 2 gray image

TABLE I.

SUMMARY OF VARIOUS SIMULATION PARAMETER

S. No.	Pre-processing step	Parameter value
1.	Smoothing	$\sigma = 2$
2.	Adaptative thresholding	sensitivity=0.4
3.	Erosion	square of size 2
4.	Dilation	square of size 2

**III. NORMALIZATION**

Normalization [19-20] refers to operations such as the estimation and correction of a character’s slant, scaling the character to a uniform size and also possibly reducing the character to a skeleton so that the line width is uniform, one unit wide. Depending upon the variation among the character image, various types of normalization methods can be used such as:

- Skew Normalization
- Slant Normalization
- Size normalization
- Curve Smoothing

**IV. COMPRESSION**

Compression of input data is desired, to improve the speed of character recognition. In addition to classical lossless image compression technique, thresholding and thinning [21-23] are the most popular techniques in the character recognition system.

*Thresholding:* It dramatically reduces the size of input grayscale image by representing it using only two intensity i.e., ‘0’ and ‘1’. Because of the reduction size, it improves the processing speed of character recognition system [21].

*Thinning:* During this process similar value intensity level represented by a single value. Hence, during this process, there is a possibility of reducing compression. It helps in extracting desired information from the shape of the stroke. Thinning can be done pixel-wise or non-pixel wise [22-23]. Pixel level thinning is done using erosion and iterative contour peeling until one-pixel wide skeleton remained. This type of thinning is sensitive to noise and may lead to deformation of the character shape. In the case of non-pixel thinning global information of the character used for thinning. Fig. 18 represents, thinning of edge detected image Fig. 15 after image filling.



Figure 18. Thin image of the edge detected image Fig. 15 after image filling

## V. CONCLUSION AND DISCUSSION

Image pre-processing is one of the most significant steps of character recognition. It is a multi-step process and the use of a particular step or not depending on the quality of input image. This paper shows the stage-by-stage analysis of each image-preprocessing stage with-respect to character recognition with the simulation result. The simulation parameters used in each stage such as smoothening, sharpening, normalization etc., can be further tuned based on the application requirement. Further work can be done by using the preprocessed image for a character recognition system that may be statistical ACR (Automatic Character Recognition) technique, Syntactic ACR technique or character recognition using neural networks.

## REFERENCES

- [1] Y. Shi, W. Fan, G. Shi, "The research of printed character recognition based on neural network", *IEEE 4th International symposium on parallel architectures, algorithm and Programming*, pp. 119 – 122, 2011.
- [2] H. Fujisawa, "Forty Year of research in character and document recognition an industrial perspective", *Elsevier-Patten Recognition*, vol. 41, no. 8, pp. 2435-2446, 2008.
- [3] M. Cheriet, N. Kharma, C. L. Liu, C. Y. Suen, "Character Recognition System –A guide for students and Practioners ", *Wiley-Interscience Press*, 2007.
- [4] W. Bieniecki, S. Grabowski, and W. Rosenberg, "Image pre-processing for improving OCR accuracy," *IEEE Int. Conf. on Perspective Technologies in MEMS Design*, pp. 75-80, 2007.
- [5] L. Sorba, J. Grman, R. Ravas, "Impact of Gaussian noise and image filtering to detected corner points positions stability", *MEASUREMENT 2017*, Proceedings of the 11th International Conference, Smolenice, Slovakia, pp. 123-126.
- [6] Stanley H. Chan, Todd Zickler, Yue M. Lu, "Understanding Symmetric Smoothing Filters: A Gaussian Mixture Model Perspective", *IEEE Trans. On Image processing*, pp. 5107 – 5121, Vol. 26(11), 2017.
- [7] A. Polesel, G. Ramponi, V.J. Mathews, "Image enhancement via adaptive unsharp masking", *IEEE Trans. On Image processing*, pp. 505 - 510, Vol. 9(3), 2000.
- [8] Wei Ye, Kai-Kuang Ma, "Blurriness-Guided Unsharp Masking", *IEEE Trans. On Image processing*, pp. 4465 - 4477, Vol. 27(9), 2018.
- [9] Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.
- [10] Bradley, D., G. Roth, "Adapting Thresholding Using the Integral Image," *Journal of Graphics Tools*. Vol. 12, No. 2, 2007, pp.13-21
- [11] A. Nandal, V. Bhaskar, A. Dhaka, "Contrast-based image enhancement algorithm using grey-scale and colour space", *IET signal processing*, Vol. 12(4), 2018, pp. 514 – 521.
- [12] Yi-Chong Zeng, "Automatic local contrast enhancement using adaptive histogram adjustment", *IEEE International Conference on Multimedia and Expo 2009*, New York, NY, USA, pp. 1318-1321.
- [13] Y. Wang, Q. Huang, J. Hu, "Adaptive Enhancement for Low-Contrast Color Images via Histogram Modification and Saturation Adjustment", *IEEE 3rd International Conference on Image, Vision and Computing (ICIVC) 2018*, Chongqing, China, pp. 405-409.
- [14] S. J. Lim, *Two-Dimensional Signal and Image Processing*, *Englewood Cliffs*, NJ, Prentice Hall, 1990, pp. 478-488.
- [15] J. R. Parker, *Algorithms for Image Processing and Computer Vision*, New York, John Wiley & Sons, Inc., 1997, pp. 23-29.
- [16] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679-698.
- [17] J. M. C. Brown, J. E. Gillam, D. M. Paganin, M. R. Dimmock, "Laplacian Erosion: An Image Deblurring Technique for Multi-Plane Gamma-Cameras", *IEEE Transactions on Nuclear Science*, Vol. 60(5), 2013, pp. 3333 – 3342.
- [18] A. N. Rahman, O. Heriana, P. Putranto, et. al., "Morphological dilation for radar image enhancement", *International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2017, pp. 68-71.
- [19] M. Kozielski, J. Forster, and H. Ney, "Moment-based image normalization for handwritten text recognition," in *International Conference on Frontiers in Handwriting Recognition*, Bari, Italy, Sep. 2012, pp. 256-261.
- [20] W. Guerfaii and R. Plamondon, "Normalizing and Restoring On-line Handwriting", *Elsevier-Patten Recognition*, vol. 26, no. 3, pp. 418-431. 1993.
- [21] H. Sung, "A Skip-line with Threshold Technique for Binary Image Compression," Master Thesis, Fu Jen Catholic University, Taipei, Taiwan 106, R.O.C, Jul. 2008
- [22] T. Abu-Ain, S. N. H. S. Abdullah, B. Bataineh, K. Omar, "A fast and efficient thinning algorithm for binary images", *J. ICT Res. Appl.*, vol. 7, no. 3, pp. 205-216, 2013
- [23] H.K. Anasuya Devi, "Thinning: A Pre-processing Technique for an OCR System for the Brahmi Script", *Ancient Asia*, vol.1, pp 167-172, 2006.