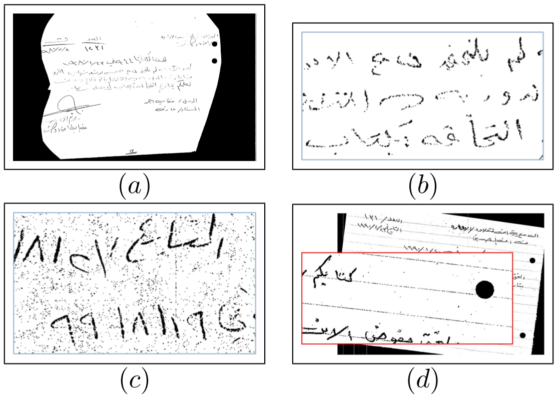
Image Enhancement for Degraded Binary Document Images

Zhixin Shi, Srirangaraj Setlur and Venu Govindaraju
Center for Unified Biometrics and Sensors
Department of Computer Science and Engineering
State University of New York at Buffalo
Buffalo, NY 14260, U.S.A.
zshi@buffalo.edu

Abstract

This paper presents a novel set of image enhancement algorithms for binary images of poorly scanned real world page documents. Problems that are targeted by the methods described include large blobs or clutter noise, salt-and-pepper noise and detection and removal of non-text objects such as form lines or rule-lines. The algorithms described are shown to be very effective in removing clutter noise and pepper noise as well as form lines and rule-lines. A region growing algorithm is also described to enhance the quality of the text and to fix the problems arising from the salt noise which leaves holes in the text and creates broken strokes. The methods were tested on 204 images from the challenge set of the DARPA MADCAT Arabic handwritten document image data. The results indicate that the methods described are robust and are capable of significantly improving the image quality for downstream OCR systems.



(a) (b) (c) (d)

Figure 1. Sample images showing (a) irregular clutter noise, (b) salt noise inside text causing broken strokes, (c) pepper noise, and (d) non-text objects such as broken rule-lines, punch holes and other types of noise.

1 Introduction

Image pre-processing algorithms for noise removal and enhancement of text quality followed by extraction of text objects such as text lines, words or characters are a necessary first step in any end-to-end OCR system. Poorly scanned images with a variety of noise artifacts such as those seen in the DARPA MADCAT challenge set present challenges that cannot be trivially overcome using existing methods. This paper describes novel approaches to address some of these problems that can be seen in the sample images from our data set (Figure 1).

Prior image enhancement algorithms for degraded document images are mostly designed to target a particular problem. Algorithms specifically designed for removing clutter noise can be found in [1, 4], Salt-and-pepper noise can be removed using filters such as modified median filters [8], kFill operator [2], optimal Boolean filters [3] and Modified Directional Morphological Filter (MDMF) [5].

In this paper, we propose a novel technique for the enhancement of degraded binary document images such as legacy scanned document images that suffer from multiple noise artifacts. A biased-downsampling approach is used to remove the clutter noise efficiently, followed by a statistical approach designed to classify different types of noise. Images identified as being pepper noise intensive are processed through a local filtering algorithm which removes the pepper noise from the text background after a pivotal text identification process. Salt noise, which leaves holes in the text causing broken strokes, is addressed by a region growing algorithm which enhances the text by interpolation. Rule-lines and form lines are also detected and removed without breaking the text strokes. The presented algorithms are tested on 204 images from the challenge set of the DARPA MADCAT Arabic handwritten document image corpus. The results indicate that the methods are robust and improve the quality of the images significantly and this is borne out by the improved performance of downstream OCR modules on the DARPA MADCAT test set.

In Section 2 we present our algorithms. In Section

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.305

3 we describe our experiments and results. Section 4 describes our conclusions.

2 Proposed algorithms

Scanned binary document images encountered in practical OCR applications often produce poor recognition results due to many problems that can be alleviated with appropriate pre-processing. Our goal for the enhancement of such images is to detect the appropriate pre-processing steps that are necessary to improve OCR performance and apply them in the right sequence. This includes detection and removal of non-text objects such as noise and rule-lines, and improving the quality of the text by fixing broken strokes. Degraded binary document images can be broadly classified into four categories: (i) Clutter noise, which includes noise in the margins and any connected components that are large solid black areas (ii) Pepper noise - These are typically small connected components produced as a binarization artifact due to over-thresholding of document images where contrast between the foreground and background is poor or uneven (iii) Form lines and rule-lines - It is important to detect and remove these lines without degrading the interfering text and (iv) Salt noise and holes within text strokes encountered due to light or uneven writing such as text written using a pencil or marker and is typically an artifact of under-thresholding.

An important consideration in the practical design of enhancement algorithms for the type of real world binary document images being targeted is to ensure that the individual algorithms designed for each type of problem do not result in the creation of new problems of another type. For example, an algorithm designed for removing rule-lines should remove pixels only from the rule-lines and should not create a new problem by removing pixels that are part of intersecting text strokes.

Also, sometimes the text is so broken that some of the strokes are made of small connected components that have component sizes similar to the surrounding pepper noise. The algorithm described is designed to remove the pepper noise while not removing the small components that are part of the strokes and enhancing the strokes by repairing the broken strokes.

2.1 Clutter noise removal

Clutter noise is typically a result of poor scanning which results in dark strips or solid areas around the document. The clutter noise is removed using a multi-resolution approach. First, we use a biased down sampling algorithm to map an input binary document image to 1/36th the original size (1/6 on each direction). The down sampling rate is based on a statistical estimate over a set of randomly selected training images. The goal of the biased down sampling is to erase the text and leave the surrounding area which is significantly larger than the width of the text strokes. To accomplish this goal, our biased down sampling algorithm is designed to favor the background, where each pixel in the down-sampled image that represents a 6x6 window is set to be black only if all the pixels in the window are black. To remove the noise, connected components are generated on the down sampled image. Large components represent noise and the pixels in the original image corresponding to the marked noise areas are assigned to white. Experimental results indicate that our noise removal algorithm is not only very efficient but also very effective. The biased down sampling is the key in separating the text from the noise.

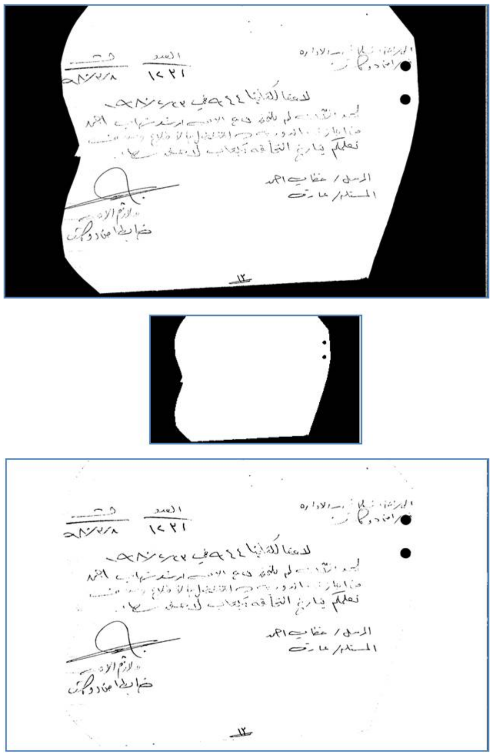


Figure 2. Removal of clutter noise. Above: Original image. Middle: Down sampled image as a mask with the text filtered out. Below: Document image with surrounding noise removed.

2.2 Noise Type Classification

Pepper noise and salt noise are different in the fact that pepper noise is removed by removing black pixels but removal of salt noise, which are holes and small gaps, is achieved by adding black pixels. To enhance the degraded document images effectively, we first have to detect if an input image is dominated with pepper noise.

One of the characteristics of pepper noise in a binary document image is that the density of connected components in the areas of pepper noise is high. To categorize a document image in terms of pepper noise, we use an estimate of the local connected component density. First, we partition an input binary document image using windows of n by n pixels. Using a size threshold s we count the number of small connected components in each window. The count gives us a distribution sampling of small connected components representing pepper noise. We then use the mean and standard deviation of the distribution to determine whether the image is pepper noise intensive. Based on our randomly selected training images, we found that the choice of parameters for window size $n = 20$ and noise threshold $s = 5$ gives us reliable classification.

2.3 Removal of Pepper Noise by Text Pivoting

Among the popular algorithms for removing pepper noise, mathematical morphology is the most used and effective method. Similarly, filter based methods have also proven to be effective for general pepper noise removal [8, 2, 5]. One of the drawbacks of these methods is that these methods are all designed for removing pepper noise without any regard for protecting potential text pixels.

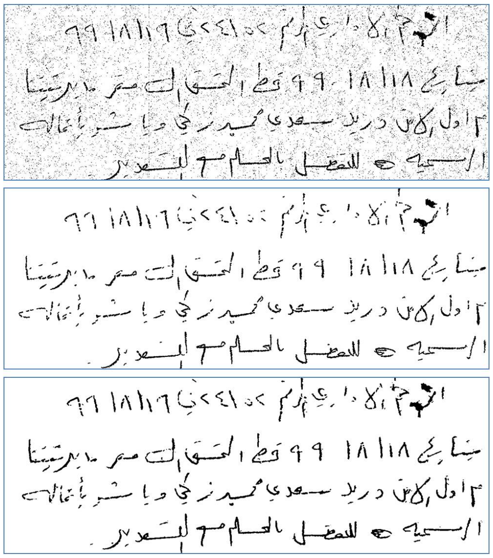


Figure 3. Removal of pepper noise by text pivoting. Above: Original image including pepper noise. Middle: Text mask image. Below: Result of pepper noise removal.

The method we propose for removing pepper noise from handwritten binary document images puts a lot emphasis on minimizing distortion of text as well as removing pepper noise with size that spans a wide range. Intuitively, we remove pepper noise away from the text region aggressively but we remove noise close to the text region conservatively. Very often, we notice that a small connected component close to text could be part of the text. Aggressive removal of small connected components near text may end up leaving gaps within the text strokes.

The first step in our noise removal algorithm is to locate text by removing most of the pepper noise. We use a mathematical morphological operator to remove the noise followed by a size based filter to remove more small connected components. The end result from the step is a mask image that consists of the majority of the text pixels.

The second step is to use the text mask generated in the first step as a pivotal image to generate a noise evaluation image. To do so, we apply a filter - a window of 10 by 10 pixels, at each noise candidate pixel, which is a black pixel in the original image but not in the mask image. Within each window location we calculate a text pixel density by counting the number of text pixels in the mask image. In the case where there is no text pixel in the window, the current noise candidate is directly considered noise and removed. Otherwise, the pixel value for the current noise candidate is set to be a gray scale value between 0 to 255, which is scaled from the text pixel density inside the window. The new gray scale values are saved in a new image buffer we call the noise evaluation image.

Finally, a simple thresholding algorithm is used to binarize the noise evaluation image to determine whether the noise candidates are to be removed or not. Combining the text mask with the result from this final step, we generate the final output image from pepper noise removal. (See Fig. 3.

2.4 Removal of Rule-lines

For removing rule-lines, we propose a technique based on a directional local profiling approach for the detection of the rule-line locations. Then a refined adaptive vertical run-length search is designed for removing the rule-line pixels without affecting text pixels.

2.4.1 Detection of Rule-lines

Ideally, a pixel on a rule-line should belong to a significantly long scan line of black color and along the direction of a rule-line, a projection profile should show a distinctive peak at the center location of the rule-line. But in reality, projection profile is very sensitive to skews in rule-lines so that it cannot produce accurate results when there are rule-lines with varying skews in the image. Also, a pixel on a rule-line may not be part of a long run, for example when a rule-line displays a rough edge. Another example is broken rule-lines where consecutive black runs may not be significantly long. This makes detection based on a simple projection profile error-prone.

To detect rule-lines, we apply an adaptive local connectivity transform [7] to a document image. We first transform the document image into its connectivity image using a new kind of runlength that we call fuzzy runlength. Intuitively, the fuzzy runlength at a pixel is a run-length (with small skip regions) at that location along horizontal(or vertical) direction. The connectivity image is a two dimensional matrix in the size of the original binary image. Each entry of the matrix is the fuzzy runlength of the pixel in its position. See Fig.4.

We take this matrix as an image and binarize it by using a modified local adaptive thresholding algorithm to reveal the locations of the rule-lines. We find that the fuzzy runs amplify the pixel intensities for the pixels on the rule-lines.

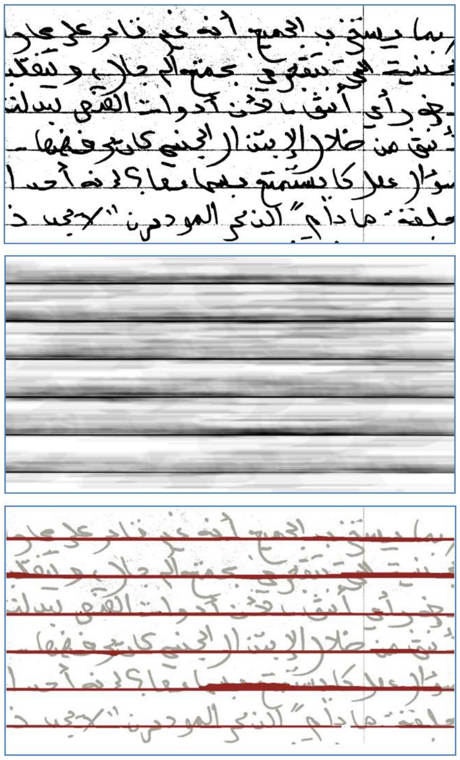


Figure 4. Rule-line detection using fuzzy runlength. Above: Original image including rule-lines. Middle: The fuzzy runlength image. The fuzzy runlength image is a gray-scale image showing the connectivity of the foreground pixels. Below: Binarized fuzzy runlength image reveals the approximate locations of the rule-lines.

2.4.2 Removal of Rule-lines

In document images from the target data set that included rule-lines, a wide variation can be observed in the thickness of the rule-lines. This is true even for rule-lines in a same image. The binary patterns in Fig. 4 show the variation. Although almost all of the pixels on the rule-lines are covered by the line patterns, these line patterns do not accurately represent the true rule-lines. In the areas where the rule-lines intersect with the handwritten text, the detected line patterns in Fig. 4 are generally thicker than the real rule-lines. For very thin or disconnected rule-lines, the corresponding line pattern may not be a single piece.

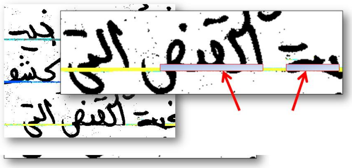


Figure 5. Using the average thickness of a rule-line estimated from its line pattern, the thick areas are marked out (pointed by arrows). The rest of the pixels in the line pattern will be used in the linear regression estimation of a best fitting line.

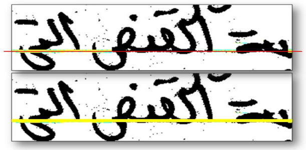


Figure 6. Using the linear regression method, a best fitting line is estimated (above). Then the entire rule-line is re-constructed by filling in pixels around the best fitting line (below).

We use the detected line patterns in Fig. 4 to re-construct the true rule-line by using linear regression. See Fig. 5 and 6. Using the re-constructed rule-lines in an image, we trace the vertical runs in the original document image. If a vertical run is entirely covered within a re-constructed rule-line, the run is removed from the original document image. If a vertical run is longer than the width of a rule-line, the run is retained, see Fig. 7. Fig. 8 shows the result of rule-line removal for the original image in Fig. 4.

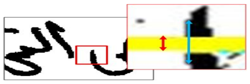


Figure 7. Removing rule-line pixels by removing vertical runs covered by the re-constructed lines.

2.5 Image Enhancement by Region-growing

In images that include salt noise, after pepper noise removal and rule-line removal, we often find holes, broken strokes and rough edges near the original intersection of rule-lines and the text. To enhance the quality of

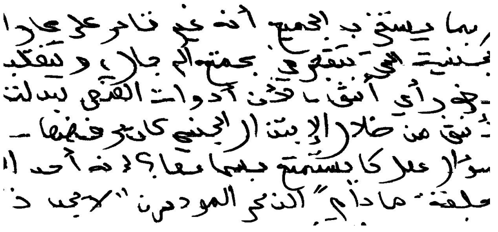


Figure 8. Rule-line removal result for image in Fig. 4.

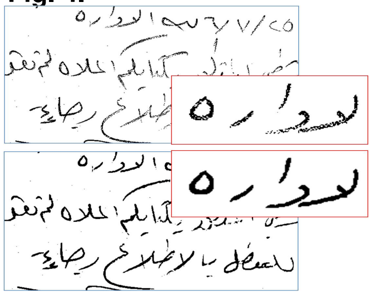


Figure 9. Binary image enhancement by region-growing for fixing salt noise and broken strokes.

these images, the binary image enhancement algorithm presented in [6] is modified to address this problem.

The algorithm uses a deformable window of 5 by 5 pixels. Starting from any black pixel on which the window is centered, by sheering to left and right, we define a shape that includes maximum number of black pixels in the window, which is deformed into a parallelogram. Within the window, gaps are filled row by row (and column by column) using the algorithm in [6]. The enhanced region also grows according to the algorithm. The improvements to the algorithm in [6] are the use of a deformable window and enhancement in two directions. Fig. 9 shows an example of the enhancement.

3 Experiment and results

The enhancement algorithms were tested on a set of 204 handwritten Arabic document images from the DARPA MADCAT data.

The pepper noise removal was evaluated by checking the result of text line separation. Without pepper noise removal, the text line separation resulted in incorrect results. Using the proposed enhancement algorithms resulted in a 95% correct line separation rate, which is a significant improvement compared to the 48% line separation rate without using the enhancement algorithms.

To evaluate the rule-line removal, we consider a rule-line as detected and removed if 90% of the pixels on the rule-line are removed. Using this metric, we obtain a correct rule-line removal rate of 91%.

As OCR results were not available, our region growing algorithm was quantitatively evaluated by visual examination of result images. The result images from our enhancement process show a significant improvement in text image quality in terms of reduced number of broken strokes, and result in smooth and even strokes.

4 Conclusion

In this paper, we present a novel method for enhancing degraded binary handwritten document images that include multiple categories of degradations. Our method was tested using the DARPA MADCAT challenge set images and our results demonstrate that the method is robust, effective and efficient.

5 Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency DARPA/IPTO (PLATO: A System for Taming MADCAT: Multilingual Automatic Document Classification Analysis and Translation) ARPA Order No. X103 Program Code No. 7M30 Issued through a subcontract from BBN Technologies Corp. under DARPA/CMO Contract # HR0011-08-C-0004. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency or the U.S. Government.

References

[1] M. Agrawal and D. S. Doermann. Clutter noise removal in binary document images. In *ICDAR*, pages 556–560, 2009.

[2] K. Chinnasarn, Y. Rangsanseri, and P. Thitimajshima. Removing salt-and-pepper noise in text/graphics images. 1998.

[3] W. Lee and K. Fan. Document image preprocessing based on optimal boolean filters. *SP*, 80(1):45–55, January 2000.

[4] W. Peerawit and A. Kawtrakul. Marginal noise removal from document images using edge density. In *Proc. Fourth Information and Computer Eng. Postgraduate Workshop*, 2004.

[5] Z. Ping and C. Lihui. Document filters using morphological and geometrical features of characters. *IVC*, 19(12):847–855, October 2001.

[6] Z. Shi and V. Govindaraju. Character image enhancement by selective region-growing. *Pattern Recogn. Lett.*, 17(5):523–527, 1996.

[7] Z. Shi, S. Setlur, and V. Govindaraju. Text extraction from gray scale historical document images using adaptive local connectivity map. In *ICDAR '05: Proceedings of the Seventh International Conference on Document Analysis and Recognition*, pages 794–798. IEEE Computer Society, 2005.

[8] K. K. V. Toh and N. A. M. Isa. Noise adaptive fuzzy switching median filter for salt-and-pepper noise reduction. *IEEE SIGNAL PROCESSING LETTERS*, 17(3):281–284, 2010.