Concordia University

COMP 6771 Image Processing

Fall 2017

Project Report

Student Name:
Qingbo Kang

Student NO.:
40058122

1 Part I

1.1 Main Ideas

1.1.1 Motivations and Contributions

There are a vast number of historical and badly degraded document images (DI) in libraries and archives. Because these images has poor quality such as bleeding-through, large black border, ink-fading, uneven illumination, contrast variation, smear, various pattern background and so on, these images are hard to read and process. Image binarization is an important task in document analysis system, how to perform image binarization in these badly degraded DI is an unsolved problem.

Image binarizatrion can be divided into 2 types which is global binarization and local binarization. Global binarization uses just one value to separate an entire image, while local binarization uses multiple value to do this task. This paper propose a novel nanparametric local adaptive binarization approach for poorly degraded DI which is based on ternary entropy and morphological operation (MO). Experiments show that this proposed approach outperforms other methods in 3 databases.

1.1.2 Flowchart of The Main Steps

Figure 1 shows the flowchart of the main steps of the proposed approach.

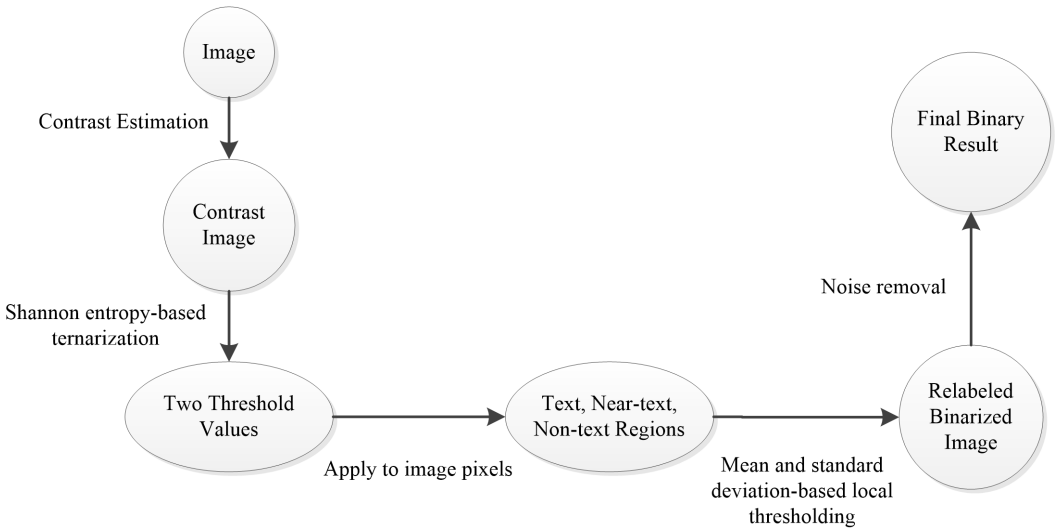


Figure 1: Flowchart of the main steps

1.1.3 Short Explanation of Each Step

The approach contains main three steps: pre-processing (contrast estimation), double threshold binarization and post-processing (noise removal). Below is a short explanation of each step.

1) Pre-processing(Contrast Estimation) This step uses the grayscale morphological closing operation to extract the contrast intensity image. Given an input grayscale DI image, it produces a grayscale contrast image.

2) Double Threshold Binarization This step generate the binarized image using the ternarization thresholding and the local mean and the standard deviation. Firstly, it uses a Shannon entropy-based ternarization technique to obtain two threshold values. Then text, near-text, non-text regions are separated. At last, the near-text regions are relabeled by applying a mean and standard deviation-based local threshold method. Given an input grayscale DI image and its contrast estimation image, it will produce a binary image.

3) Post-processing(Noise Removal) This step uses binary MO, shrink and swell filters, and a graph searching strategy to remove two kinds of noise: salt-pepper and block noise. Its input and output both are binary images.

1.2 The Pre-processing (Contrast Estimation)

The steps of pre-processing are illustrated in Algorithm 1.

Algorithm 1 Contrast estimation

Input: A grayscale image;
Output: A grayscale contrast image;
1: Remap all gray-levels to a new range [0, 255] to get adjusted image;
2: Extract background image using gray-level morphological closing operation on adjusted image;
3: Subtract the original from the background image to obtain the contrast image;
4: **return** Contrast image;

1.3 The Post-processing (Noise Removal)

The steps of post-processing are illustrated in Algorithm 2.

Algorithm 2 Post-processing (noise removal)

Input: A binary image;
Output: A noise removed image;
1: Apply binary morphological operators;
2: Correct the areas that are larger than one pixel and smaller than the character stroke width by using shrink and swell filters;
3: Utilizes a graph searching strategy to remove block noise;
4: **return** Noise removed binary image;

1.4 The Double Threshold Binarization Method Implementation

Figure 2 shows the binarization result image. It can be seen from the result image that the text region is segmented, although it is not very precious and contains a lot of noise, but it is much better than the original image.

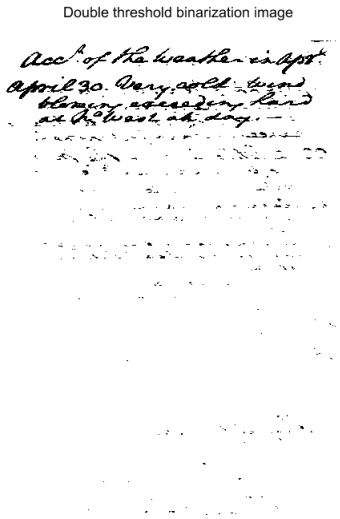
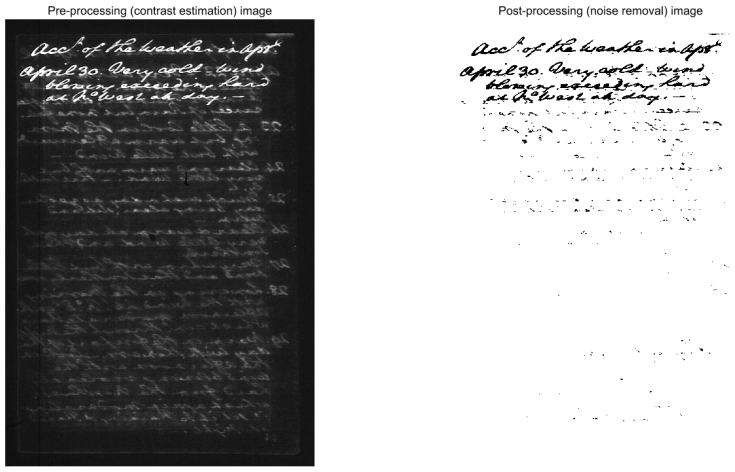


Figure 2: Double threshold binarization image

1.5 The Pre-processing Implementation

Figure 3 (a) shows the pre-processing result image.



(a) (b)

Figure 3: (a) Pre-processing result image. (b) Post-processing result image.

1.6 The Post-processing Implementation

Figure 3 (b) shows the post-processing result image. It can be seen that after this operation, salt-and-pepper and block noise are mainly be removed.

2 Part II

For this problem, I choose a photo which was taken by me at Chongqing, China. It's a night scene, so the image brightness is a little low and I think it requires some form of histogram correction. The original image is a color image with resolution 5472×3648. Figure 4 shows the original image and Figure 5 shows the histogram of each channel.

2.1 Histogram Equalization

Figure 6 shows the histogram equalization result image and its histogram is shown in Figure 7. As we can see, the result is not good which means the visual effect is a little bad, for example, the sky region even has some yellow color pixels.

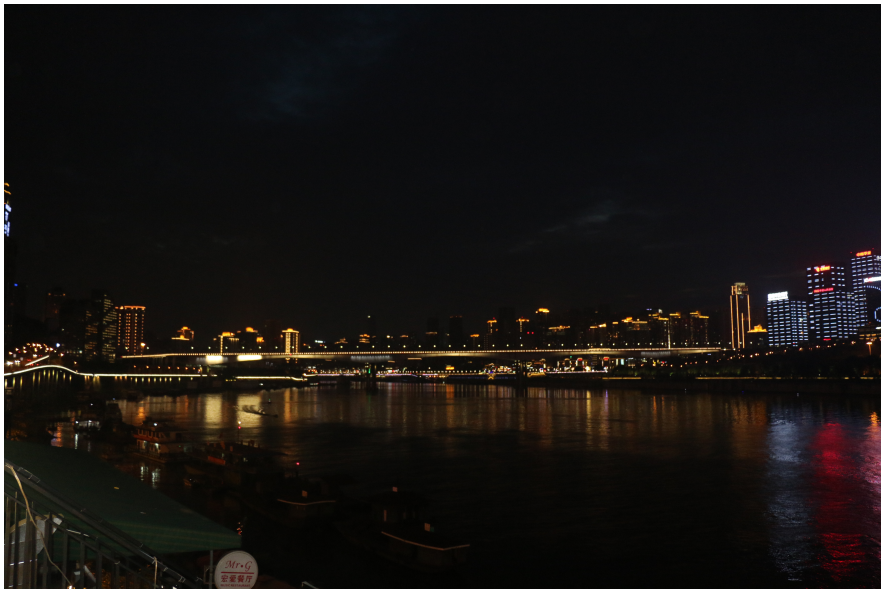


Figure 4: Original image

2.2 Histogram Matching

For the target histogram, I choose a night scene image from the Internet which is much similar to my original image. Figure 8 shows the target image for histogram matching and Figure 9 shows its histogram.
Figure 10 shows the histogram matching result image and Figure 11 shows its histogram. And we can see its visual effect is much better than the previous one.
From the above results, obviously, histogram matching result is much better than the histogram equalization result if you choose right target histogram, actually, histogram equalization is just a special kind of histogram matching.

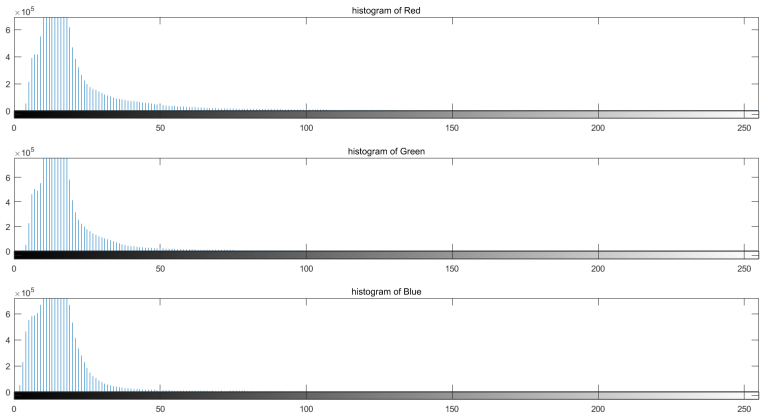


Figure 5: Histogram of the original image

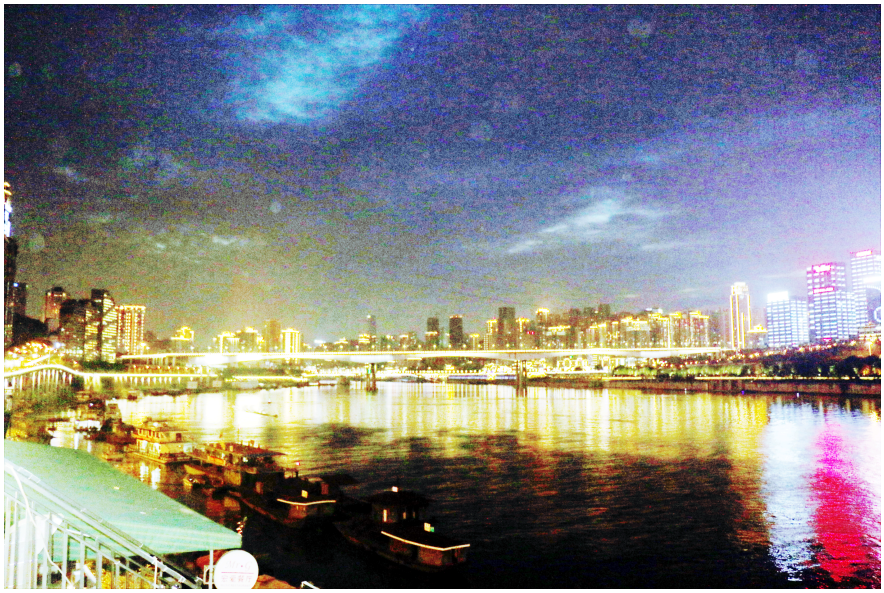


Figure 6: Histogram equalization result image

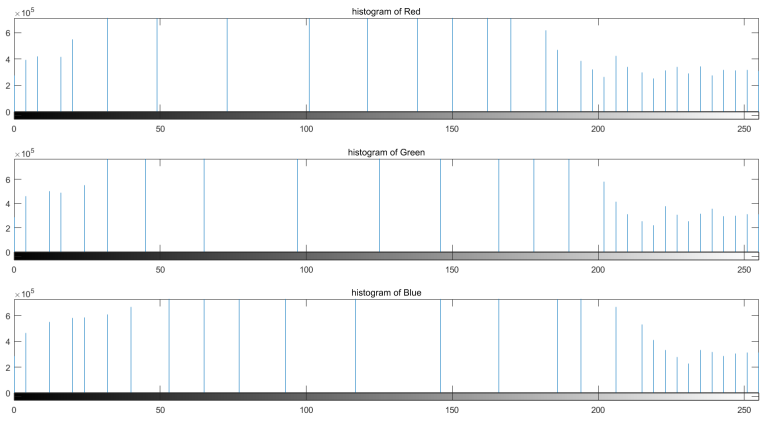


Figure 7: Histogram of the histogram equalization result

Figure 8: Target image for histogram matching

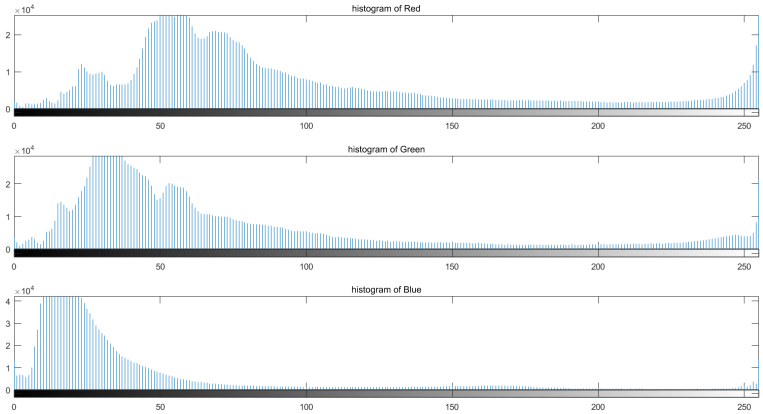


Figure 9: Histogram of the target image

Figure 10: Histogram matching result image

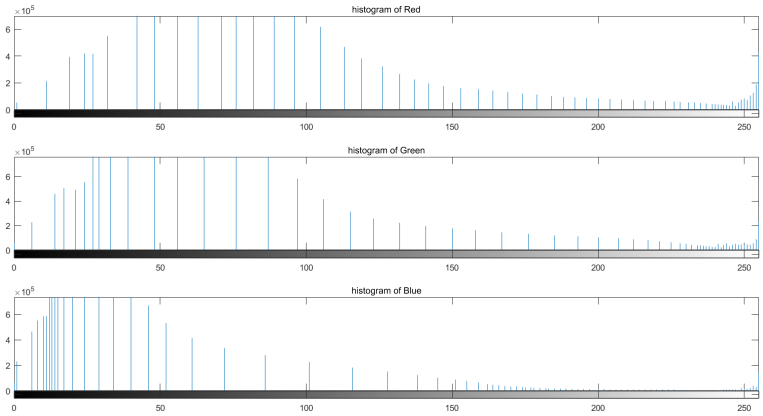


Figure 11: Histogram of the histogram matching result image