ROBUST LOCAL THRESHOLDING METHOD FOR SEGMENTATION OF NON-UNIFORM COLOURED DOCUMENTS

THIDAR AUNG

M.C.Sc. **AUGUST 2019**

ROBUST LOCAL THRESHOLDING METHOD FOR SEGMENTATION OF NON-UNIFORM COLOURED DOCUMENTS

BY

THIDAR AUNG

B.C.Sc.(Hons:)

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Computer Science

(M.C.Sc.)

University of Computer Studies, Yangon

AUGUST 2019

ACKNOWLEDGEMENTS

I would like to express my respectful gratitude to **Dr. Mie Mie Thet Thwin**, Rector, University of Computer Studies, Yangon, for allowing me to develop this thesis and giving me general guidance during the period of study.

And also I would like to express my gratefulness to **Dr. Myint Myint Sein**, Professor, Head of Geographical Information System (GIS) Lab, University of Computer Studies, Yangon**,** for her suggestions, patience and support while I studied.

I would like to express my special thanks to my supervisor, **Dr. Thin Lai Lai Thein**, Professor, Faculty of Information Science, and Geographical Information System Lab, University of Computer Studies, Yangon, for her kind and continued guidance, motivation and support for my thesis.

I would like to thank and respect go to **Dr. Thi Thi Soe Nyunt,** Professor and Head, Faculty of Computer Science, University of Computer Studies, Yangon, for her invaluable guidance and administrative support, as Dean of Master Course, throughout the development of the thesis.

I would like to thank **Daw Hnin Yu Mon**, Lecturer, Department of Language, University of Computer Studies, Yangon, for editing my thesis from language point of view.

Thanks are also extended especially to all my teachers, my parents, seniors, friends and staff members of the University of Computer Studies, Yangon for the contribution towards the completion of this thesis.

STATEMENT OF ORIGINALITY

I hereby certify that the work embodies in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

--------------------------------- -------------------------------

Date Thidar Aung

ABSTRACT

Text document segmentation is one of the essential steps in text document recognition and extraction systems. The existing image segmentation methods are not much reliable for text with colour gradient and texture background. Also, long processing time of existing methods is unfavorable. Thus, aiming to have reliable segmentation and less processing time, a new local thresholding method is proposed and its performance is tested in this study.

The proposed method is based on pixel intensity magnification concept. In proposed algorithm, the input image is enhanced by edge sharpening. Then, the image is divided into multiple local windows by using non-overlapping. The magnification factor for each non-overlapping local window is calculated based on minimum intensity, maximum intensity, range and the number of dominant intensities in the corresponding local window. For segmenting pixels in each local window, magnification factor is used.

The performance of proposed algorithm is measured in terms of segmenting accuracy and processing time. The tested images include different types such as text in uniform colour, text in multicolour, text in gradient background, text in national ID, text with watermark, highlight text, text in light illumination, text in passport, text in bank card. Also, the performances of Otus' and Niblack's methods are tested with the same images.

The proposed method gives the better results for most images with maximum accuracy of 100% and lowest accuracy of 80%. The highest efficiency of Otsu's method is 100% and its lowest accuracy is 0%. Since Otsu's method loses all data sometimes. For Niblack's method, it gives 100% accuracy only for text with simple colour background while its lowest accuracy is 10%. The proposed system's average accuracy is higher than Otus' and Niblack's average accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTS OF EQUATIONS

CHAPTER 1

INTRODUCTION

Today, OCR (Optical Character Reorganization) becomes popular for automatic retrieval information from image version of text documents, vehicle license plates, business cards, invoices, ID cards, driver licenses, etc. Generally, there are three main steps to perform optical character recognition. These are preprocessing of image that contains targeted characters, segmentation of characters and character recognition. In the first step, some processes such as image quality enhancement, skewness correction are performed if required. In the second step, the characters are segmented from the background and fragmented as individual character. Finally, in the last step, each character is recognized by matching with pre-known features of each character. It can be understood that the outcome of every step is very critical for consequential steps and to obtain reliable recognition of an OCR system.

Therefore, it has been an essential thesis area and gained a great attention of thesis society in order to develop more and more powerful OCR systems. Until now, many different approaches have been proposed to cope with difficult conditions in each individual step or entire process of OCR [1].

One can realize that character segmentation (the second step) in OCR process in a key step to further recognize each character or to analyze the information in the image. The failure in the character segmentation is the failure of OCR system. Thus, the approach for character segmentation must be flexible at any quality level and illumination conditions.

As stated above, the character segmentation involves segmenting all contained characters from the background and fragmenting all characters as individual one. There are many methods to segment the characters from the background. Most frequently used methods are global thresholding method and colour based character detection. However, the drawback of these methods is the failure of segmenting characters when the gradient between character colour and background colour is small. Moreover, these methods are also not applicable if there is non-uniform illumination in the image. For this reason, in these days, many systems started to propose local thresholding method for segmenting characters from the background colour. The local thresholding perform better than global thresholding but it is even not perfect for some conditions.

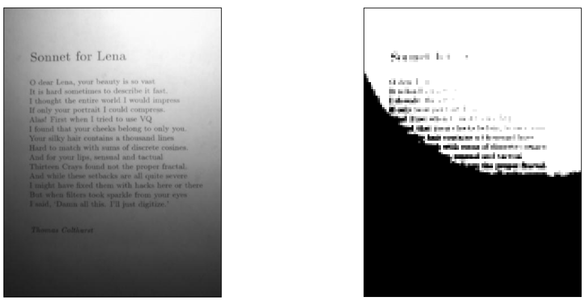
In this regard, the objective of propose method is robust local thresholding method for segmentation of non-uniform coloured documents. The advantages of proposed method are mathematically simplicity, short processing time and reliable segmentation for text in non-uniform colour background.

1.1 Problem Statements

Segmentation (i.e. binarizing) is commonly implemented by means of colour based detection, global thresholding, or local thresholding methods. The drawback of global thresholding can be seen in figure 1.1(a), which shows the segmentation result for non-uniform illumination text document by using global thersholding method (Otsu's method). Figure 1.1(b), it can be noticed that some text in document are lost during binarizing.

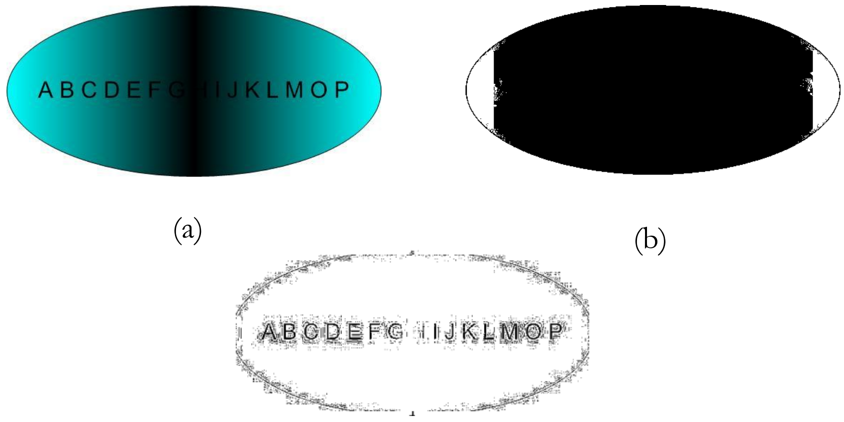
Local thresholding method is more efficient for the text image with non-uniform illumination or non-uniform colour [14]. In local thresholding method, intensity threshold for binarizing is locally searched by setting a window. Thus, threshold value varies from region to region. However, local thresholding method is also imperfect for some conditions. Figure 1.2 shows the original image, and segmented result using global thresholding (Otsu's method) and local thresholding (Niblack’s method).

It can be seen that local thresholding shows the better performance than global thresholding. However the result has not reached to a satisfactory level. There is some noise around each character. This noise can interrupt in further processes of OCR (Optical Character Reorganization).



(a) (b)

Figure 1.1 Text Image Segmentation (a) Original Image
(b) Result Using Global Thresholding



(a) (b)

(c)

Figure 1.2 Text Image Segmentation (a) Original Image (b) Result Using Global Thresholding (c) Result Using Local Thresholding

1.2 Motivation

Most local thresholding methods are based on local mean and a bias constant (k). In most methods, the required bias constant is described with a range, which necessitates the condition-dependent adjustment of k's value. The selected value of bias constant (k) can affect the performance of the method. For example, the effect of k value can be seen in Figure1.2(a), Figure 1.2(b) and Figure 1.2(c) show the results using Niblack's method with k=-0.1 and k=-0.2. Due to selected k values, there is a significant difference between the results. Thus, local mean and standard deviation based methods are not appropriate for the document images with large colour gradient [6]. In addition, a longer processing time is required if the threshold is calculated for each pixel. Thus, a simpler, more robust, more time-efficient local thresholding method is still in demand. This produces a motivation for this method.

1.3 Objectives of Propose Local Thresholding Method

The aims of this study are to improve segmentation for non-uniform coloured documents and to reduce the processing time. Thus, to achieve these aim, the specific objectives are to propose a new local thresholding method which is based on intensity magnification which is derived from local maximum, local minimum and local contrast range, to experimentally verify the performance of proposed thresholding method by testing colourful document images and to compare the performance of proposed method with that of other existing methods (Otsu’s method and Niblack method). The performance parameters are segmentation result and processing time.

1.4 Scope of Study

The main focus of proposed method is an image processing algorithm for segmentation (binarization) of non-uniform coloured text documents. Different non-uniform coloured text documents such as text in gradient background, text in national ID, text with watermark, highlight text, text in light illumination, text in passport, text in bank card. Text fragmentation and recognition are out of scope. Some text images are created and some are taken from image source.

1.5 Organization of the thesis

This thesis is composed of five chapters. The first chapter is introduction of thesis which explains about background, problem statement, motivation, objectives of propose local thresholding method, scope of study and organization of the thesis. In the following chapter, review on related theories (conversion to gray image, edge sharpening, dilation, erosion, etc.,) and many existing segmentation methods (Otsu’s method, Niblack’s method, Sauvola’s method, Singh’s method, LAAB (Local Adaptive Automatic Binarization) method is conducted. Then, proposed image processing algorithm is presented and explained in Chapter 3. The concept of intensity magnification is explained in detail. In Chapter 4, the results for different types of text images are depicted and discussed. Finally, conclusion is drawn based on major findings in Chapter 5.

CHAPTER 2

THEORETICAL BACKGROUND

Binarization is a key step for successful text document recognition. The performance of document image analysis directly depends on the output of binarization. Nowadays, there are many global and local thresholding methods for binarization although they are not specifically aimed for text document binarization. Also, there have been many works which proposed both modified and innovative thresholding methods. Most are local thresholding methods. Thus, theoretical background and updated thresholding methods are recalled and discussed in this chapter.

2.1 Binarization of Image

Binarization is segmentation of image into only two colour's foreground (white) and background (black). Thus, the output is binary image. The binarization can be illustrated by using the system block diagram is in figure 2.1.

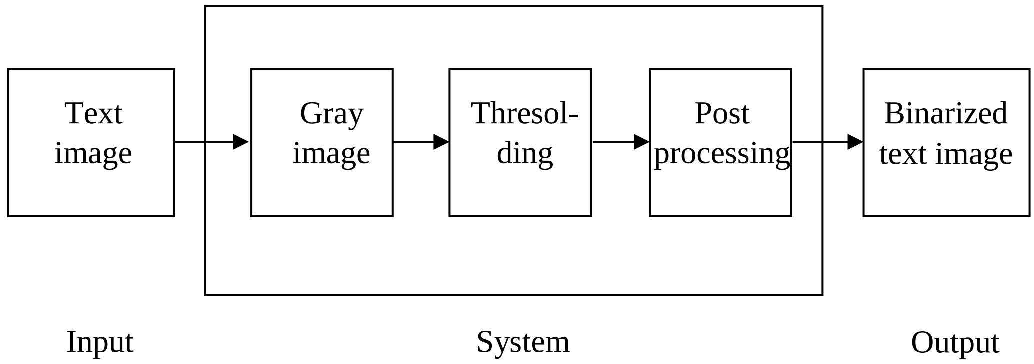


Figure 2.1 Block Diagram of System Binarization

In figure 2.1, the input image is text document image and then it is converted to gray scale image. Next, an optimum threshold value is calculated and used to convert the various intensity levels as binary values, one and zero. Afterwards, the binary image sometimes undergoes post processing if it is necessary. The post process could be morphological operation. Sometimes, the post process is not included in the binarization system.

There have been many methods for thresholding in both global and local perspectives. It has been shown.

2.2 Preprocessing

2.2.1 Conversion of Colour Image to Gray Scale Image

For binarizing a colour image based on thresholding, one essential process is conversion of colour image to gray scale image. When the true colour in the image is represented by RGB values, the colour image can be converted to gray scale image is shown in equation (2.1).

$$I_g = \alpha_1 R + \alpha_2 G + \alpha_3 B \tag{2.1}$$

where α_1, α_2, α_3 are coefficient of Red (R), Green (G) and Blue (B) intensities. The values of α_1, α_2, α_3 are dependent of the gray model. In average gray scale model, α_1=0.33, α_2=0.33 and α_3=0.33 are used. In popularly used gray scale model, α_1=0.3, α_2=0.6 and α_3=0.1 are used.

2.2.2 Edge Sharpening

When it is required, the edges of objects in an image are blurring, need to be sharpened. The edge sharpening process can be applied in both colour image and gray scale image. The edge sharpening can be shown in equation (2.2).

$$I_{gs} = \omega\left(I_g - \kappa I_{gf}\right) \tag{2.2}$$

where I_{gs} is sharpened edge, I_g is original gray image, I_{gf} is smoothened gray image, κ is scale factor and ω is multiplication factor. The value of κ is normally 0.5 and the value of ω is 2.

2.2.3 Filtering

To smooth an image, it must undergo a filtering process [17]. In edge sharpening process, to produce a smoothen image, filtering process is required. Average (mean) filtering is the most straightforward and mostly used method for smoothing image. The concept of average filtering is as follows.

From original gray image, a sliding local window is set at every pixel. It is called “mask”. Any size of window can be applied, but different results obtain.

Then, the filtered value for center pixel is simply the average value of all intensities of pixels in the windows. Local window (3x3 Mask) is shown in figure 2.2.

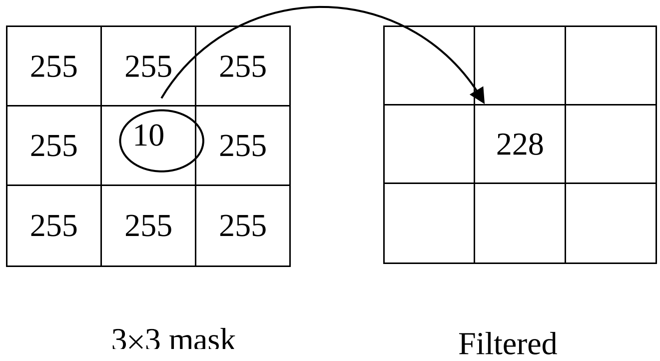


Figure 2.2 A Local Window (3×3 Mask) and Filtered Value

2.3 Otsu's Thresholding Method

It is called global threshold because only a single threshold value is used for binarizing the whole image. The input image must be gray scale image [11, 13]. The mathematical expression for the threshold-based binarization equation is (2.3).

$$I_{bw}(i, j) = \begin{cases} 1 & if \ \ I_g(i, j) \geq T \\ 0 & Otherwise \end{cases} \tag{2.3}$$

where I_{bw} is a binary image, I_g is a gray image, T is an optimum threshold value, i is a row number and j is a column number. The optimum threshold value is freely selected or theoretical calculated. Otsu's method is the most popular method for calculating optimum threshold.

This method was proposed by a Japanese professor, Nobuyuki Otsu [18]. It is based on histogram and statistics of pixel intensities in the image [23]. The input image must be gray scale image. The concept of Otsu's method is described below.

The histogram of gray image is firstly created. Thus, the histogram has 256 bins for the whole since the gray intensity has the rage of 0~255. From these bins, every level is set as a threshold value. Then, the total variance, called "*between-class-variance*" is calculated and the optimum threshold is defined as the value that gives minimum "*between-class-variance*". The "*between-class-variance*" is calculated as follows.

The background properties are calculated equation (2.4) and (2.5) using all pixels values which are smaller than selected threshold value.

$$w_b = \frac{\sum_{i=1}^{T} n_i}{n_{total}} \tag{2.4}$$

$$\mu_b = \frac{\sum_{i=1}^{T} p_i n_i}{\sum_{i=1}^{T} n_i} \tag{2.5}$$

The foreground properties are calculated equation (2.6) and (2.7) using all pixels values which are greater than selected threshold value.

$$w_f = \frac{\sum_{i=T+1}^{I_{max}} n_i}{n_{total}} \tag{2.6}$$

$$\mu_f = \frac{\sum_{i=T+1}^{I_{max}} p_i n_i}{\sum_{i=T+1}^{I_{max}} n_i} \tag{2.7}$$

where w is weight factor, μ mean value, n is the number of pixel, T is selected threshold, p is pixel intensity value.

Then, the *within-class-variances* for background and foreground are calculated using corresponding properties are equation (2.8) and (2.9).

$$\sigma_b^2 = \frac{\sum_{i=1}^{T} n_i(\mu_b - p_i)}{\sum_{i=1}^{T} n_i} \tag{2.8}$$

$$\sigma_f^2 = \frac{\sum_{i=T+1}^{I_{max}} n_i(\mu_f - p_i)}{\sum_{i=T+1}^{I_{max}} n_i} \tag{2.9}$$

where σ_b is the *within-class-variance* for background and σ_f is the *within-class-variance* for foreground and I_{max} is maximum intensity in the image. Then, the total variance (*between-class-variance*) for selected threshold (T) can be calculated equation (2.10).

$$\sigma^2 = w_b \sigma_b^2 + w_f \sigma_f^2 \qquad (2.10)$$

Every value between 0 and I_{max} is set as threshold value and the corresponding total variance must be calculated for each threshold. Then, the value that gives minimum total variance is defined as optimum threshold value and it is used for binarization.

The advantages of Otsu's thresholding method are simplicity and less time-consuming. On the other hand, the weakness of Otsu's thresholding is that it cannot give even acceptable level when the image has non-uniform illumination. For this condition, locally calculated threshold is required. Thus, local thresholding methods have been proposed.

2.4 Local Thresholding

To overcome the drawback of global thresholding method, many local thresholding methods have been developed. The optimum threshold value for binarization is calculated locally for every single pixel [19]. The statistics of a group of pixels in a window that centers a pixel is used to calculate the locally optimum threshold. Thus, it gives more flexibility than global thresholding but at cost of long processing time. A number of local thresholding methods have been developed in previous works. Some popular methods are discussed.

2.4.1 Niblack's Method

This method was developed by Maney Niblack [12, 15]. Thus, it is called Niblack's method. In this approach, pixel intensities in a sliding window are used to calculate the threshold value for the center pixel. It can be illustrated as shown in Figure 2.3. In original approach, the window size (m×n) is 15×15 and every pixel must be taken as center pixel to be binarized.

Using all 225 pixels in the window, the local means and the local standard deviation are calculated in equation (2.11) and (2.12).

$$\mu = \frac{1}{m \times n} \sum_{i=1}^{m \times n} p_i \qquad (2.11)$$

$$\sigma = \sqrt{\frac{1}{m \times n} \sum_{i=1}^{m \times n} (\mu - p_i)} \qquad (2.12)$$

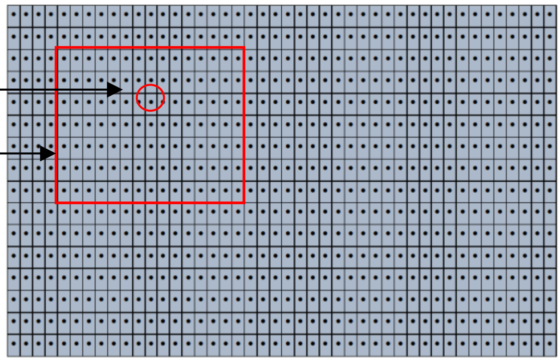


Figure 2.3 Local Window and Center Pixel

where μ is local mean, σ is the local standard deviation, p pixel intensity in the window. Then, the optimum threshold for the center pixel can be calculated using local mean value and standard in equation (2.13).

$$T_w = \mu + k\sigma \tag{2.13}$$

where T_w is the local optimum threshold value, k is a bias (a constant value) and it is taken as (-0.2) in the original approach [24]. Then, the pixel at the i^{th} row and the j^{th} column is binarizing using T_w.

2.4.2 Sauvola's Method

This method was developed by Jaakko Sauvola and M. Pietikainen [8, 20]. Thus, it is called Sauvola's method. It is modified version of Niblack's method. It is also based on local mean and standard deviation in a 15×15 sliding window. Then, the optimum threshold for the center pixel can be calculated equation (2.14).

$$T_W = \mu\left(1 + k\left(\frac{\sigma}{R} - 1\right)\right) \tag{2.14}$$

where T_W is the local optimum threshold value, k is a bias constant and it is taken as 0.2~0.5 and R is also a constant and 128 for gray scale image [2].

Then, the pixel at the i^{th} row and the j^{th} column is binarizing using T_w. The local mean (μ) and the local standard deviation (σ) can be calculated in the same way in Niblack's method. The value of k is the quality of the segmented result.

2.4.3 Singh's Method

This method was developed by Romen Singh [22]. Thus, it is called Singh's method. In this approach, only local mean is applied, but not the standard deviation. To calculate the local mean, the integral sum image is used for more efficient in time.

Then, the optimum threshold for the center pixel can be calculated in equation (2.15) and (2.16) [22].

$$T_w = \mu\left(1 + k\left(\frac{\Delta}{1-\Delta} - 1\right)\right) \tag{2.15}$$

$$\Delta = I(i, j) - \mu \tag{2.16}$$

where T_w is the local optimum threshold value, Δ is the mean deviation, k is the constant and it is in the range of 0~1.

2.4.4 LAAB Method

This method, called local adaptive automatic binarization was also developed by Romen Singh [22]. Since, the gray scale image is directly changed to binary image in this approach. It does not need a threshold value. It also needs the mean value in every local window are calualated the equation (2.17), (2.18) and (2.19).

$$I_{bw} = \frac{|1-2\nu| - (1-2\nu)}{2|1-2\nu|} \tag{2.17}$$

$$\nu = \frac{k(1+\partial)}{(1-\partial)} \tag{2.18}$$

$$\partial = \{I(i, j) - \mu\}\{(1-\mu)\} \tag{2.19}$$

where, the value of k is in the range of 0.5~0.6 and ∂ is the mean deviation and I_{bw} is a binary value.

2.4.5 Bernsen's Method

This method was proposed by Bernsen [3, 17]. This approach is a combination of global and local thresholding. The optimum threshold value is decided depending on the L that is a contrast threshold. Bernsen's method equations are (2.20) and (2.21).

$$I_m = \frac{I_{max} + I_{min}}{2} \tag{2.20}$$

$$T_w = \begin{cases} I_m & If \quad I_{max} - I_{min} \geq L \\ T & If \quad I_{max} - I_{min} < L \end{cases} \tag{2.21}$$

where I_{max} and I_{min} maximum and minimum intensities in local window, the value of k is in the range of 0.5~0.6 and ∂ is the mean deviation and I_{bw} is the binary value. The value of L is 15.

2.4.6 Imocha Singh's Method

This method was proposed by Imocha Singh [21]. This approach is based on the local mean and the local contrast. The window size that gives the best result is 3×3. In equation (2.22) are shown.

$$T_w = k[\mu + (I_{\max} - I_{\min})(1 - I(i, j))] \tag{2.22}$$

where μ is the local mean, I_{max} and I_{min} maximum and minimum intensities in the local window, k is the bias constant and it is in the range of 0~1 and I (I, j) is the pixel intensity at the i^{th} row and the j^{th} column.

2.5. Post Processing

Post processing is sometimes necessary to make the resulted binary image more perfect. The erosion and dilation are discussed as the post processing processes.

2.5.1 Erosion

The erosion is the process of removing foreground pixels if they have no complete foreground structure as the structure element. This process can be explained using an illustration as shown in figure 2.4.

The structure element can be any size and any shape. Mostly, it is rectangle and circle (disk) shape. The result of eroded binary image depends on the shape and size of structuring element.

In illustrated example shown in figure 2.4, the structuring element has (3×3) size and rectangle shape. When it is superimposed on each pixel in the binary image, only highlighted pixels have same complete structure. Thus, the other pixels are eroded, i.e, change as background pixel, and the resulted binary image is also shown in figure 2.4. It is called the erosion process.

0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	1	1	1	1	0	1	1
0	1	1	1	0	0	1	1
1	1	1	1	0	0	1	1
1	1	1	1	0	0	1	1
1	1	1	1	0	0	1	1

Original binary image

1	1	1
1	1	1
1	1	1

Structuring element

0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0

Eroded binary image

Figure 2.4 Binary Image and Structuring Element

2.5.2 Dilation

The dilation is the process of adding foreground pixels so that the boundary pixel has complete foreground structure as the structure element. This process can be explained using figure 2.5.

As in the erosion process, the structure element can be any size and any shape. Mostly, 3×3 rectangle shape is used. The result of dilation by using (3×3) structuring element shown in figure 2.5. It can be seen that the object becomes bigger than before. Thus, a larger structuring element produces a more extreme dilation effect. Very similar effect can be achieved by repeated dilations using a smaller but similarly shaped structuring element.

Dilation process is used to fill the holes in the objects and to combine disconnect objects by making them larger.

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0

Original binary image

1	1	1
1	1	1
1	1	1

Structuring element

0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0

Dilated binary image

Figure 2.5 Binary Image and Structuring Element

CHAPTER 3

THE PROPOSED LOCAL THRESHOLDING METHOD

The advantage of global thresholding is time-efficient since it uses a single threshold value reducing much processing time. On the other hand, the local thresholding methods are obviously efficient for binarization at the cost of more processing time. It is because of searching threshold value for each pixel. Thus, it is worth to have a method that has both features.

The proposed method takes the concept of local thresholding. It considers for local small region but not every pixel. Thus, the window is not non-overlapping. Thus, the details of proposed method are discussed a follows.

3.1 Flow Chart of Proposed Method

The flow chart of proposed method is shown in figure 3.1. The algorithm starts with acquiring image. The image undergoes preprocessing. Preprocessing includes conversion of colour image and edge sharpening. Then, the image is divided into multiple non-overlapped small regions.

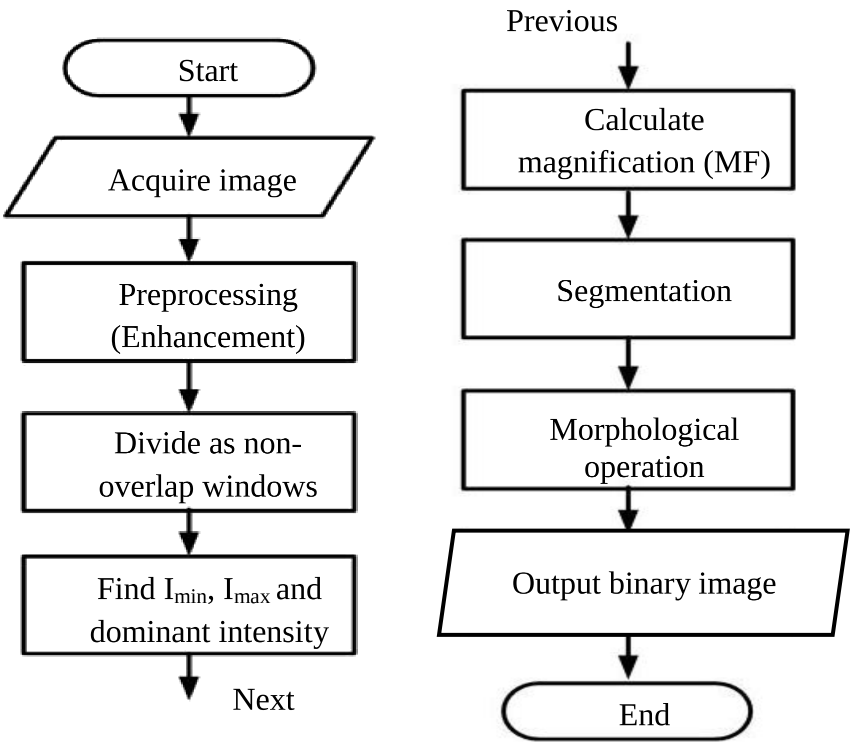


Figure 3.1 Flow Chart of Proposed Method

After that, the dominant pixel intensities, maximum intensity and minimum intensity are extracted from each region. Depending on these data, calculate magnification factor. The calculation process of magnification factor is explained in details. Then magnification factor is used to multiply the corresponding region so that it can be easily segmented.

Afterwards, morphological operation, specifically dilation process, is used to get better segmented image (binary image).

3.2 Acquiring Image

The images used are offline images in JPEG format. Some images are captured, some images are created and some are extracted from internet sources. The images have different sizes. The small image size is 83×173 and the largest image size is 902×1612. The true colour in images is represented by RGB model. The resolution of the images is 96 dpi. For the image of text document is shown in figure 3.2.

Figure 3.2 Colour Text Image

3.3 Preprocessing

In preprocessing, the image is firstly converted to gray scale image. Then, the edges in image are sharpened.

3.3.1 Gray Scale Conversion

The true colour image is firstly converted to gray scale image by using gray scale model with α_1=0.3, α_2=0.6 and α_3=0.1. For gray scale image is shown in figure 3.3.

Figure 3.3 Gray Scale Image

3.3.2 Edge Sharpening

Edge sharpening is performed to improve the interface difference especially in text image with gradient colour. Equation 2.2 with scale factor, 0.5 and multiplication factor is 2, is used for edge sharpening the gray image. The resulted image from edge sharpening process is shown in figure 3.4.

Figure 3.4 Gray Image after Edge Sharpening

3.4 Dividing into Multiple Windows

In this step, the enhanced image is subdivided into multiple small square windows. The window size is $w_r \times w_c$. In this work, w_r and w_c are set as 30. The windows are not overlapping to each other. Since the image has $m \times n$ dimensions, the numbers of windows in horizontal and in vertical direction are calculated by using equation (3.1) and (3.2).

$$m_r = \frac{(m - s_r)}{w_r} \tag{3.1}$$

$$n_c = \frac{(n - s_c)}{w_c} \tag{3.2}$$

where, the number of windows in vertical direction is m_r+1, the number of windows in horizontal direction is n_c+1, s_r is the size of the last window in vertical direction and s_c is the size of the last window in horizontal direction, s_r and s_c can be zero or smaller than normal window size $w_r \times w_c$. To be more obvious, the chopping of the image into multiple sub windows is illustrated in figure 3.5.

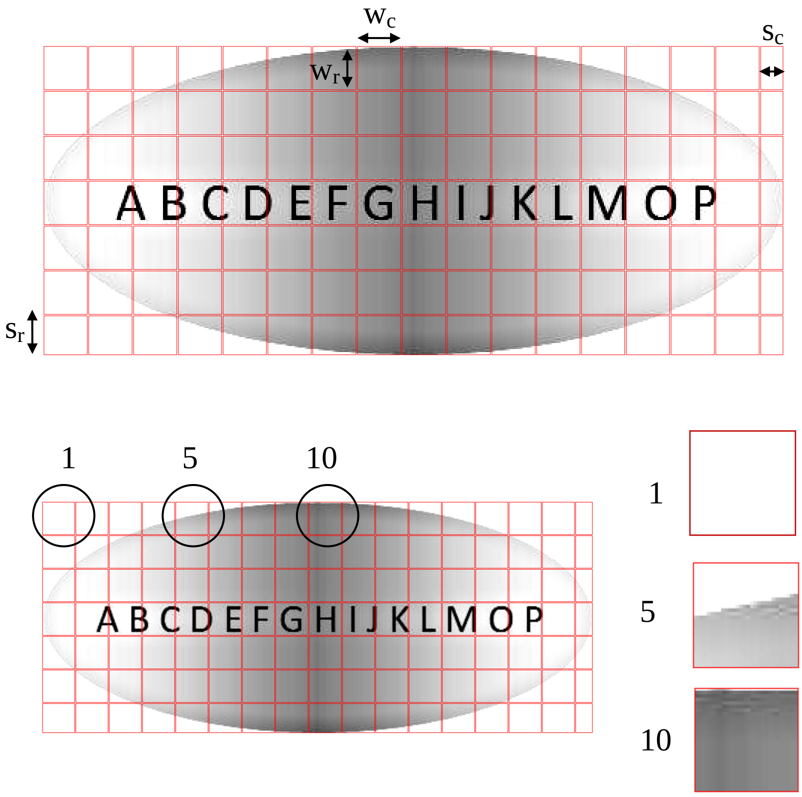
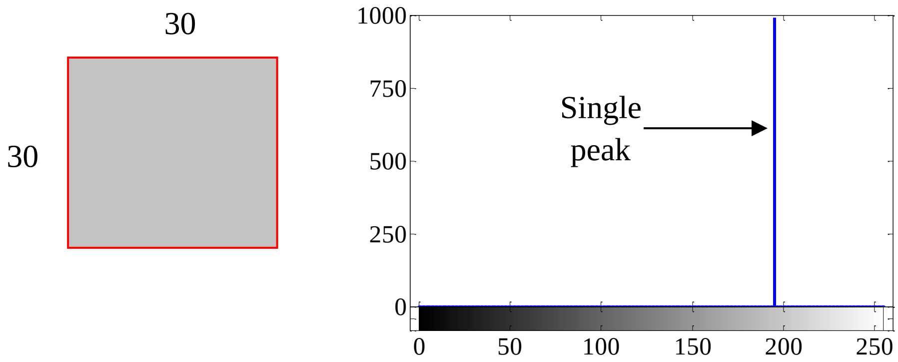


Figure 3.5 Multiple Subwindows of Image

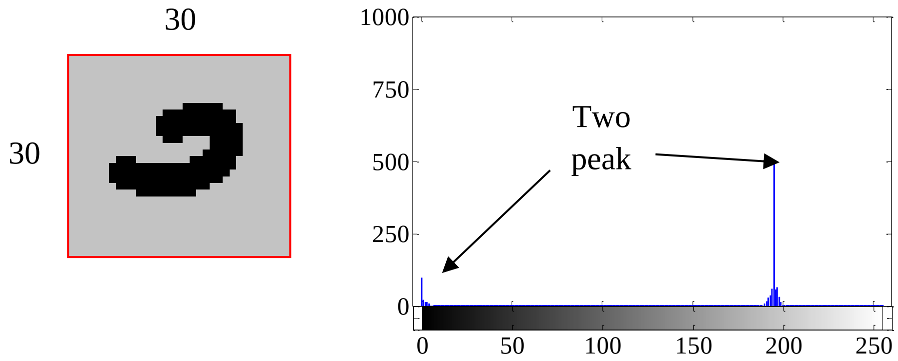
3.5 Extraction of Dominant Intensities

In proposed method, magnification factor is based on range of dominant intensities in each subwindow. Thus, dominant intensities in each subwindow are firstly extracted. Thus, it needs to develop histogram for each window. In image processing, histogram is the plot of intensity levels versus the number of corresponding pixel intensities. In figure 3.6, close-up views of three different types of subwindows (no.1, no.5 and no.10) are shown.

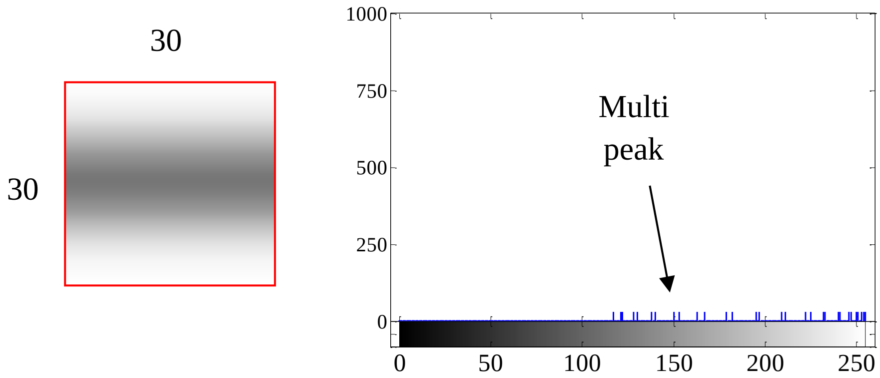
As shown in figure 3.6, generally there are three different types of subwindows. These are single-colour window, two-colour window and multicolour window. The histogram of each type of subwindow can be explained as follows.



(a)



(b)



(c)

Figure 3.6 Different Subwindows and Corresponding Histogram

(a) Single-Colour (b) Two-Colour (c) Multicolour

From the histogram, the gray tone vector for the subwindow is developed as follows.

$$G_t = \lfloor G_{p,min} : G_p, max \rfloor \tag{3.3}$$

where G_t is gray tone vector, $G_{p,min}$ is minimum of gray peak and $G_{p,max}$ is maximum of gray peak.

As shown in figure 3.6(a), in single-colour subwindow, there is only one dominant intensity value when histogram is created for it. Thus, it finds only single peak. For 30×30 window, there are totally 900 pixel. All pixels have the same intensity level. Therefore, for this case gray tone vector is [195 195].

In figure 3.6(b), there are two dominant in the second type window. When histogram is created, two dominant intensity values can be found. The first peak is around 10 and the second peak is 195. Thus, for this case gray tone is [10 195].

For the last type shown in figure 3.6(c), there are multiple dominant in the window. When its histogram is created, there are many dominant intensity values. Thus, multiple peaks can be seen. The first peak is around 110, the second peak is 115, the third peak is 125 and so on. Thus, gray tone for this type of window is [110 115 125…255].

To find the peak, a threshold must be defined. If the number of pixel is at least 10, it should be defined as dominant pixel intensity. In each window, there will be a total of 900 pixels. Thus, 1% is defined as the threshold value to define peak intensity.

The difference between previous method and our proposed method is that only dominant gray levels are considered in this approach. Although there is minimum intensity level in the window, if the number of that intensity level is small, it will not be much important in threshold computation.

Then, the minimum and maximum intensity in the window is calculated by using Equation (3.4), (3.5) and (3.6).

$$G_{min} = \min(\lfloor G_{p,min} : G_p, max \rfloor) \tag{3.4}$$

$$G_{max} = \max(\lfloor G_{p,min} : G_p, max \rfloor) \tag{3.5}$$

$$R_g = G_{max} - G_{min} \tag{3.6}$$

3.6 Magnification Factor

3.6.1 Concept of Intensity Magnification

In this proposed method, the basic idea is magnification of intensities in each subwindow with a suitable magnification factor. The objective of magnification is to make intensity level difference more significant. It makes binarization easier than original condition. The concept of magnification can be explained as follows.

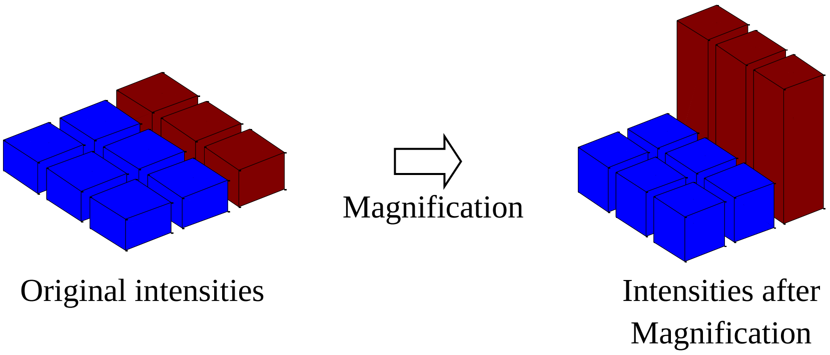


Figure 3.7 Intensity Magnification

As depicted in figure 3.7, the original intensity level in the window is not much different and it is not easy to apply a threshold to binarize the pixels. After magnification the intensities with a suitable factor, the difference of intensities becomes significant. It makes the window easier for binrization. Here, the challenge is to how to find suitable magnification factor for each window. It is discussed below.

3.6.2 Double Base Lines

The suitable magnification for a subwindow depends on the number of dominant intensities, the minimum gray tone and maximum gray tone and the range of gray tone. To calculate the magnification factor, these minimum gray tone and maximum gray tone must be compared with preset thresholds.

In this work, from gray levels range of 0~255, double base lines are defined as

B_1=1	For single pixel dominant window
B_1=1, B_2=30	For two and multiple pixel dominant window

Condition I ,II and III (Single , Double and Multi Intensity Domaint Window)

For single intensity dominant dominant window, double intensity dominant window and multi intensity dominant window the magnification factor (MF) is calculated by using algorithm.

Algorithm for calculating MF (Magnification Factor)

Begin

Inputs→ B1, B2, R_g, G_{min}, G_{max}, k

Outputs→ MF

```
if k==1
    if   Gmin < B1
        MF=0
    else
        MF=ceil (255/Gmin)
    end
elseif k==2
    if (Gmax-Gmin)<Rg & Gmin<B1
        MF=0
    elseif (Gmax-Gmin)<Rg & Gmin≥B1
        MF=ceil(255/Gmin)
    elseif (Gmax-Gmin)≥Rg & Gmin< B1
        MF=ceil(255/Gmax)
    elseif (Gmax-Gmin)≥Rg & Gmin≥B1 & Gmin<B2
        MF=ceil(255/Gmax)
    elseif (Gmax-Gmin)≥Rg & Gmin>=B2
        MF=ceil(255/Gmin)
    end
elseif k≥3
    if (Gmax-Gmin)<Rg & Gmin<B1
            MF=0
     elseif (Gmax-Gmin)<Rg & Gmin≥B1
            MF=ceil(255/Gmin)
     elseif (Gmax-Gmin)≥Rg & Gmin<B1& (Gk-1 -Gmin)<10
            MF=ceil(255/Gk >B1)
```

```
        elseif (Gmax-Gmin)≥Rg & Gmin<B1& (Gk-1 -Gmin)<10
                    MF=ceil(255/(Gk >B1))
            elseif (Gmax-Gmin)≥Rg & Gmin<B1 & (Gk-1-Gmin)≥10
                    MF=ceil(255/Gk-1)
            elseif (Gmax-Gmin)≥Rg & Gmin≥B1 & Gmin<B2 & (Gk-1-Gmin)<10
                     MF=ceil(255/(Gk>B2))
            elseif (Gmax-Gmin)≥Rg & Gmin≥B1 & Gmin<B2 & (Gk-1-Gmin)≥10
                    MF=ceil(255/Gk-1);
            elseif (Gmax-Gmin)≥Rg & Gmin≥B2
                    MF=ceil(255/Gmin);
            end
end
```

Figure 3.8 Proposed the Algorithm

After achieving the magnification factor, the corresponding window is multiplied by magnification factor. To have integer value, ceil value is used for rounding off magnification factor.

3.6.3 Calculation of Magnification Factor

The sample calculations of magnification factor at three different conditions.

Example for Condition I

G_t=[55 55] $\rightarrow$ G_{min}=55, G_{max}=55, $(G_{max}-G_{min})=0$

$(G_{max}-G_{min}) < Rg$ & $G_{min} > B_1$,

Thus, MF=ceil (255/55)=5

Example for Condition II

G_t=[30 125] $\rightarrow$ G_{min}=30, G_{max}=125, $(G_{max}-G_{min})=95$

$(G_{max}-G_{min}) > R_g$ & $G_{min} > B_1$,

Thus, MF=ceil (255/30)=9

Example for Condition III

G_t=[0 5 22 30 125] $\rightarrow$ G_{min}=1, G_{max}=125, $(G_{max}-G_{min})=125$

$(G_{max}-G_{min}) > R_g$ & $G_{min} < B_1$ & $(G_2-G_{min}) < 10$

Thus, MF=ceil (255/5)=51

Figure 3.9 shows the resulted MF for each window in a given image. Only one MF is used for each window. Thus, it saves processing time compared to existing local thresholding method.

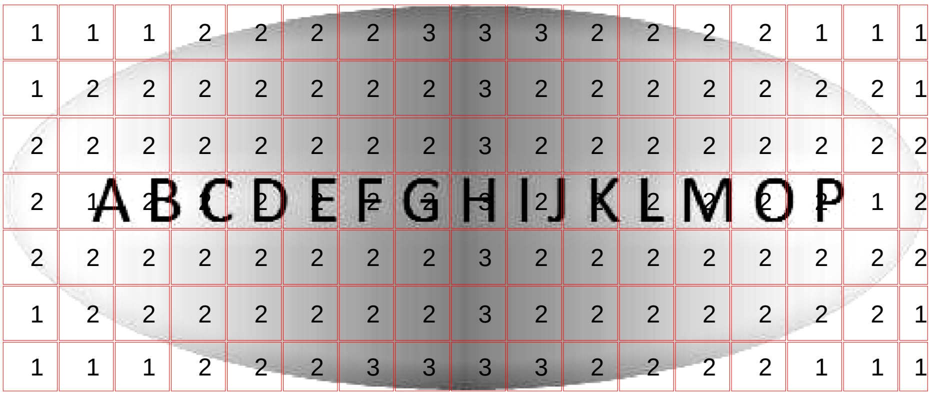


Figure 3.9 Magnification Factor

3.7 Dilation

Dilation is performed to get a better quality of text in the process. A 3×3 square shape structural element is used in morphological operation. It give a better quality for character recognition. Figure 3.10 shows the segmented result before and after.

(a)

(b)

Figure 3.10 Dilation (a) before Dilation (b) after Dilation

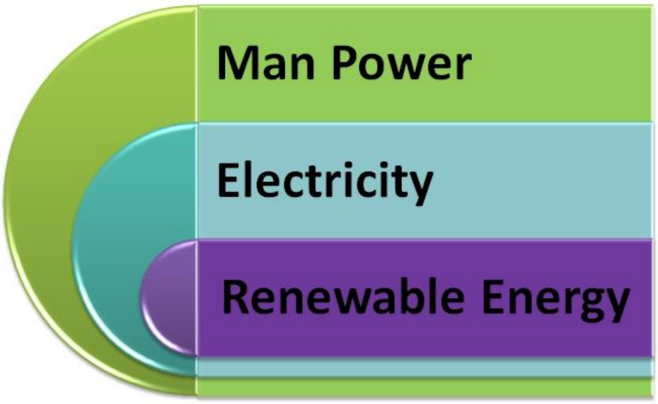
3.8 Experimental Tests

To know the performance of proposed method, it is tested with different text documents. Text documents include text in uniform colour, text in multicolour, text in gradient background, text in national ID, text with watermark, highlight text, text in light illumination, text in passport, text in bank card. Some images are created and some are taken from image source. A total images are used in tests Figure 3.11(a)-(i) shows the different text images.

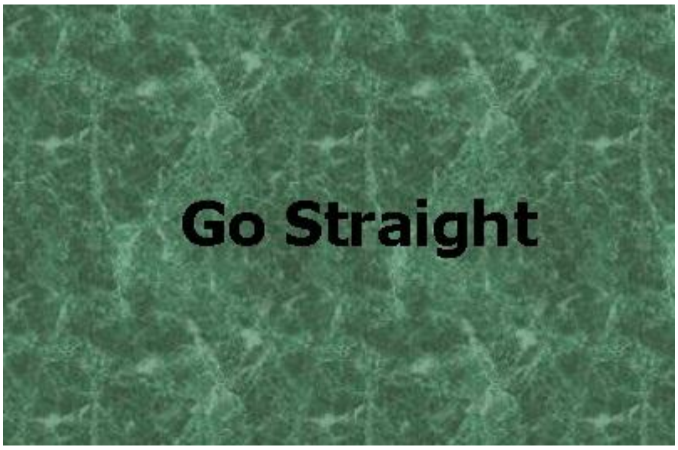
(a)

(b)

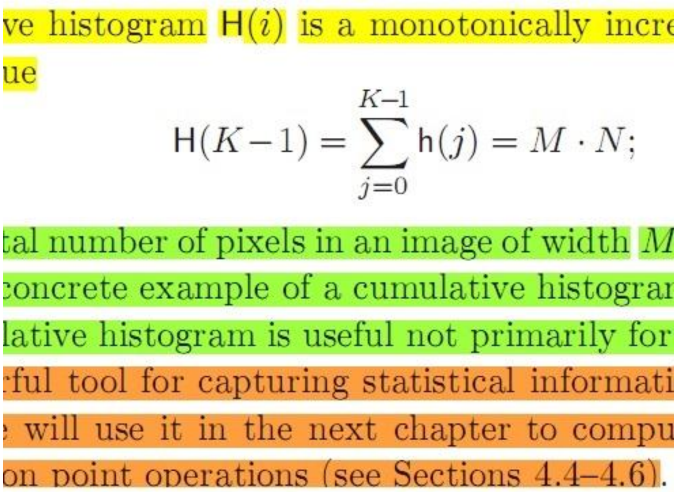
(c)



(d)



(e)

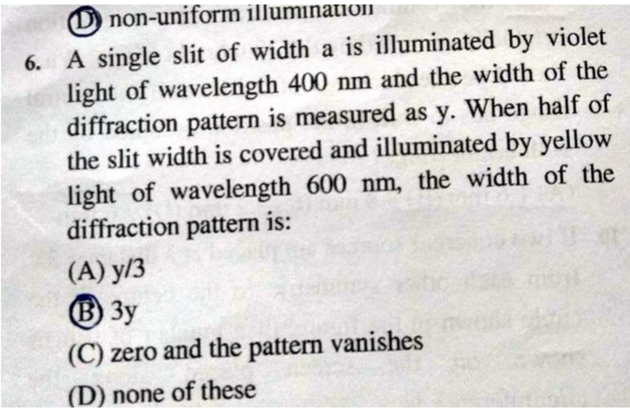


ve histogram $\mathsf{H}(i)$ is a monotonically incre
ue

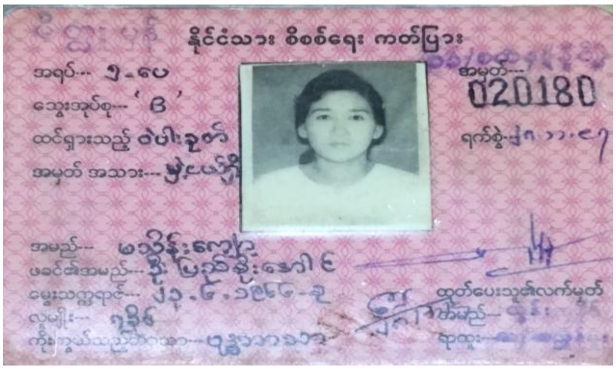
$$\mathsf{H}(K-1) = \sum_{j=0}^{K-1} \mathsf{h}(j) = M \cdot N;$$

tal number of pixels in an image of width M
concrete example of a cumulative histogra
lative histogram is useful not primarily for
ful tool for capturing statistical informati
will use it in the next chapter to compu
on point operations (see Sections 4.4–4.6).

(f)



(g)



(h)

(i)

Figure 3.11 (a)-(i) Test Images

3.9 Performance Evaluation

The performance of proposed method is evaluated in terms of processing time and data loss after segmenting the image comparing with that of Otsu's method and Niblack's method. The algorithms are implemented in MATLAB software (R2016a).

3.9.1 Processing Time

The processing time is measured in second (s). Only CPU time is used in comparison. The processor of the computer is Intel (R) Core (TM) i7-7500U and processor speed is 2.9 GHz. The processing time is automatically recorded by the program.

3.9.2 Accuracy

The accuracy of segmentation is calculated in terms of F-mean [3] by using equation (3.7) as follows.

$$\mathrm{Fmean} = \frac{2 \times \mathrm{recall} \times \mathrm{precision}}{\mathrm{recall} + \mathrm{precision}} \tag{3.7}$$

where, F-mean is accuracy percentage and recall and precision are defined in terms of true positive (TP), false positive (FP) and false native (FN).

$$\mathrm{recall} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})} \tag{3.8}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})} \tag{3.9}$$

where, a **true positive** test result (TP) is one that detects the condition when the condition is present. A **true negative** test result (TN) is one that does not detect the condition when the condition is absent. A **false positive** test result (FP) is one that detects the condition when the condition is absent. A **false negative** test result (FN) is one that does not detect the condition when the condition is present.

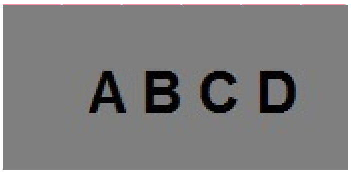
CHAPTER 4

RESULTS DISCUSSION

The performance of proposed method is verified in terms of processing time and data lost percentage after segmentation compare with the performance of Otsu's method and Niblack's method. The tested image are categorized as text in uniform colour, text in multicolour, text in gradient background, text in national ID, text with watermark, highlight text, text in light illumination, text in passport, text in bank card. The results of tested images from Otsu's method, Niblack's method and proposed method are presented and discussed.

4.1 Text in Uniform Colour

First, the performance of proposed method is tested with a simple text image. The size of image is 83×173. The image has uniform gray colour as background and black text colour. There are four characters in the image. The pairs of acquired images and output images for different algorithm are shown in Figure 4.1 (a)-(d). In output image, the text is converted to foreground colour and dilated to have a better condition.



(a)

(b)

(c)

(d)

.

Figure 4.1 Results for Text in Uniform Colour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

One can observe in Figure 4.1 that all methods including proposed method give the optimum segmentation performance for the image. Since the image background colour simple and it has only two colours, all the characters contained in the image segmented well. Thus, no data loss exists. However, the processing times are different. For this case, Otsu's method takes the shortest processing time, the proposed method takes second shorted processing time and Niblack's method takes the longest time.

4.2 Text in Multicolour

Figure 4.2 shows the comparison of results from different algorithms for the texts in multicolour. The images have four different colour rows. The background colours are gray, light blue, red and green. The colours are very different from each other. According to the segmented results, one row is lost in the result from Otsu's method. Then, Niblack's method cannot give a perfect segmentation for the third row. Meanwhile, the result from proposed method shows very perfect segmentation for every row. Thus, it has zero percent data loss. The processing time of Otsu's method is 0.09 s and that of Niblack's method is 5.5 s. The processing time of proposed method is 1.1 s.

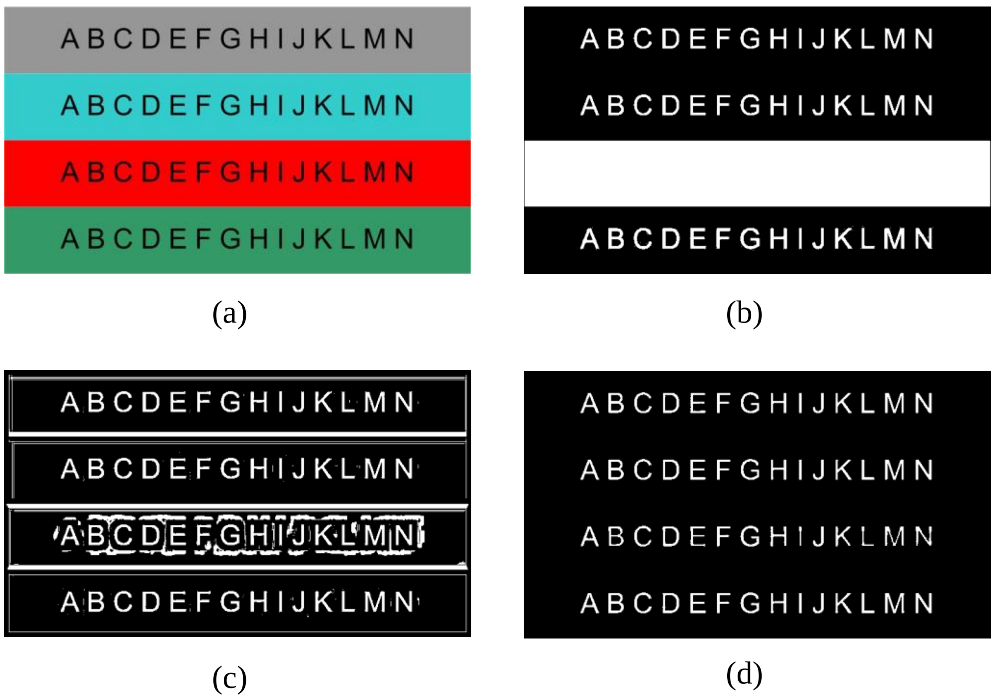


Figure 4.2 Results for Text in Row-wise Multicolour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

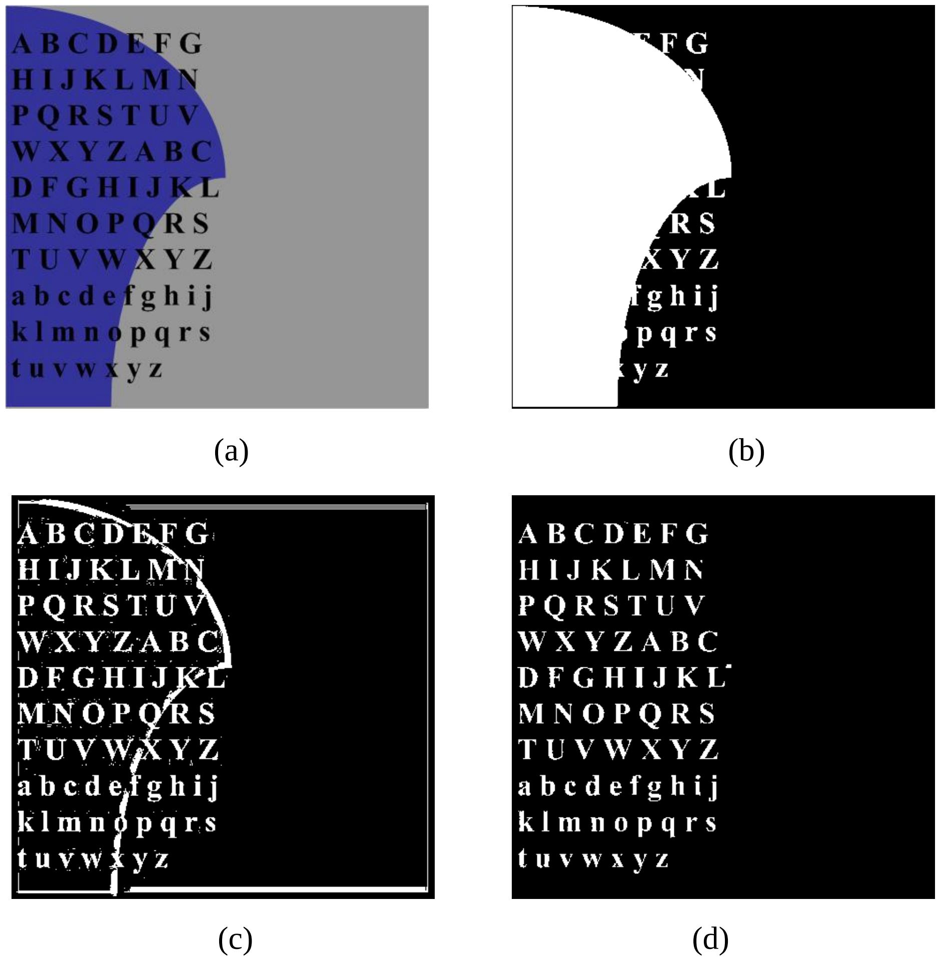


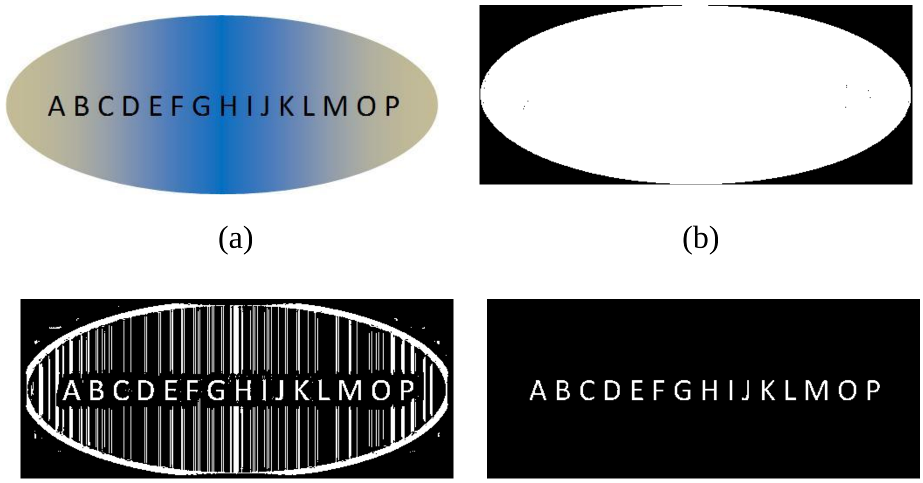
Figure 4.3 Results for Text in Irregular Region Multicolour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

The same condition is observed in the results shown in Figure 4.3. In the original image, the background has two uniform colours, blue and gray. The interface of these two colours is very sharp. The image size is 430×451. In the result from Otsu's method, the texts in the blue regions are totally lost. In the result from Niblack's method, not only the interface line between two background colours but also the spot noise around character. In contrast, the proposed method gives a better segmentation result.

Although the processing time is the shortest, Otsu's method has 70% data loss. In the result from Niblack's method, some characters are crossed through by the interface line. Thus, the original shapes of characters are lost. Although there is no data loss in Niblack's method, the appearance of interface line and the noise in segmentation will interfere in recognition process. Only proposed method can provide the perfect segmentation.

4.3 Text in Gradient Colour

The proposed method is also tested for segmenting text in gradient colour. Figures 4.4, Figures 4.5, Figures 4.6, Figures 4.7 and Figures 4.8 show the resulted images using three different algorithms. First, in Figure 4.4, the text exists in light blue colour gradient. The gradient direction is horizontal. Using Otsu's method, all the texts in the image are lost after segmentation. When Niblack's method is used, the texts in the result are surrounded by gradient lines. The proposed method gives a clear segmentation.



(a) (b)

(c) (d)

Figure 4.4 Results for Text in Vertical-Centre Gradient Colour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

In Figure 4.5, the combination of red-yellow-blue-pink colours exists as the gradient colour background. The gradient direction is vertical. The image size is 720×960. From the results, one can notice that Otsu's method is not compatible with text segmentation from gradient background. There is about 50% data loss.

The result from Niblack's method contains straight lines due to colour gradient. It is also a kind of data loss. The best result is obtained using proposed method. Moreover, the processing time taken by proposed method the second is shorter than that of Niblack's method. Although Otsu' method has the fastest processing time, the data loss is very high. The processing time of Niblack method for the image size of 720×960 is 12 second. It is 6 times larger than that of proposed method.

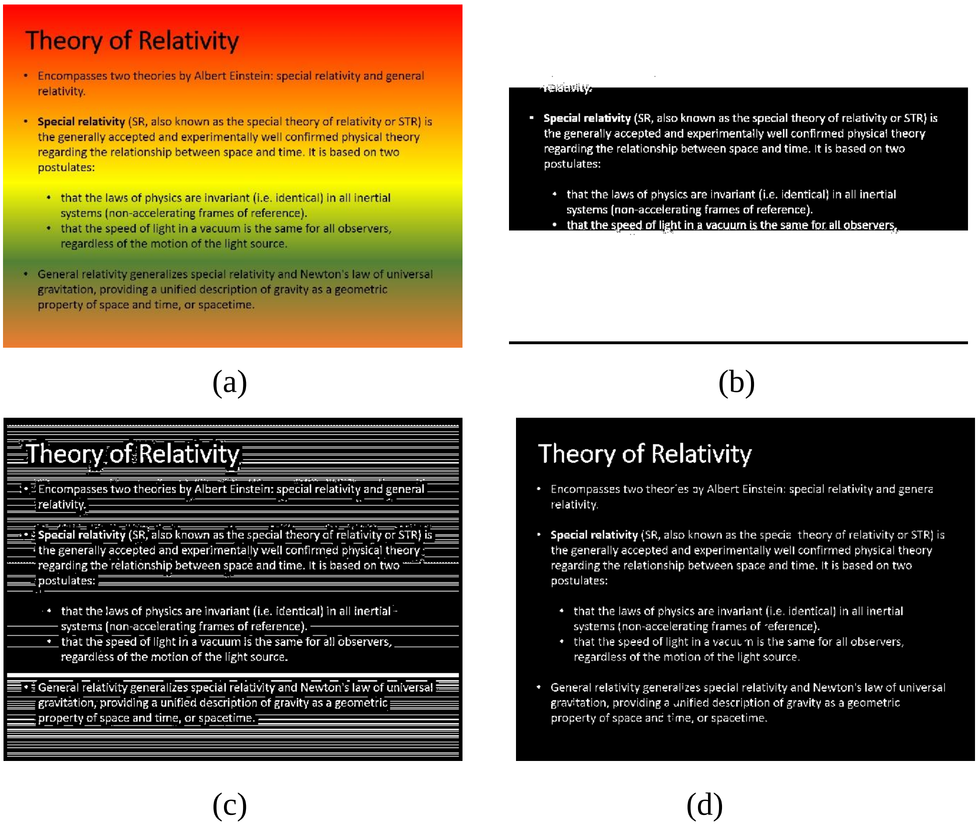


Figure 4.5 Results for Text in Gradient Colour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

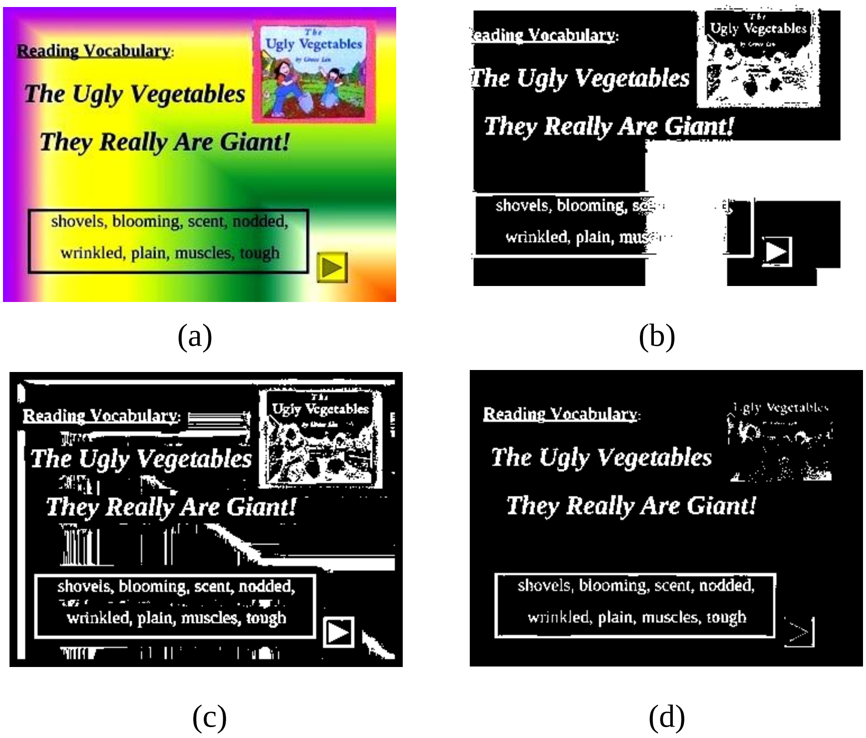
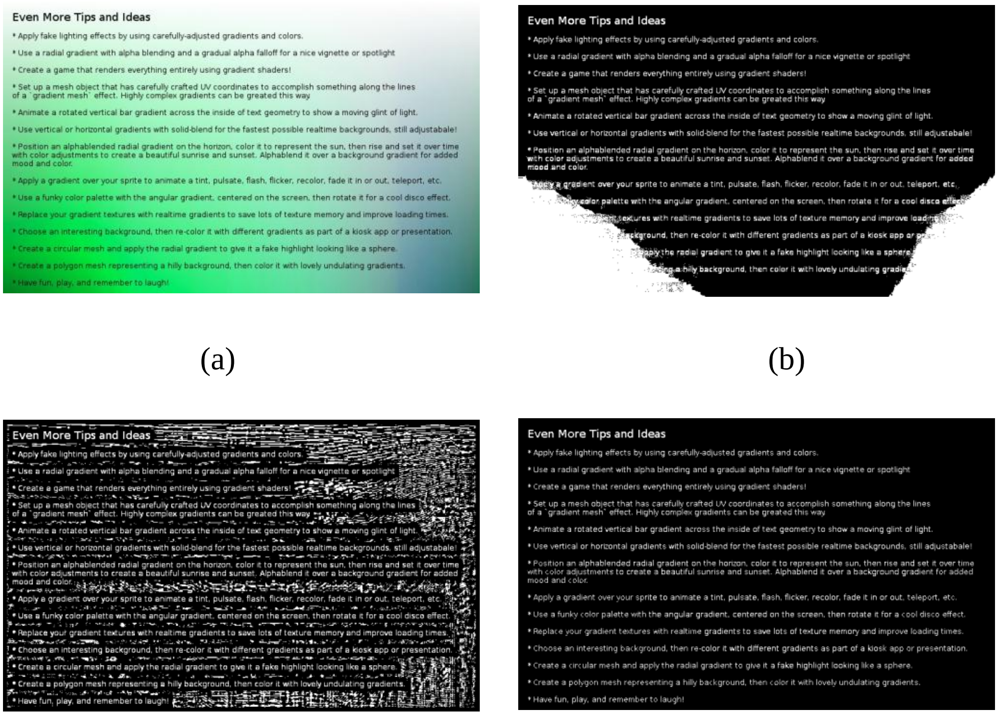


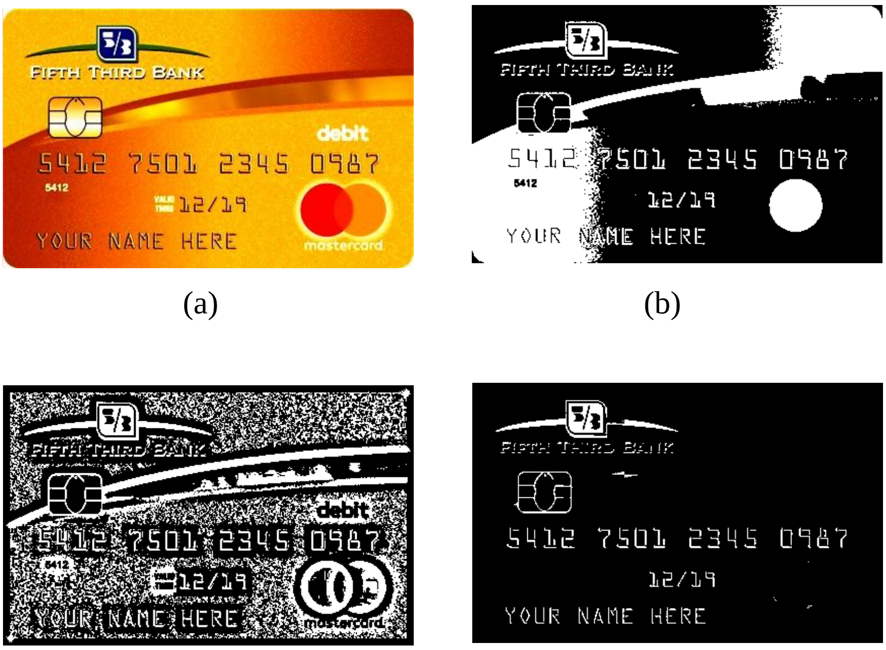
Figure 4. 6 Results for Text in Non-Uniform Gradient Colour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method



(a) (b)

(c) (d)

Figure 4.7 Results for Text in Corner Gradient Colour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method



(a) (b)

(c) (d)

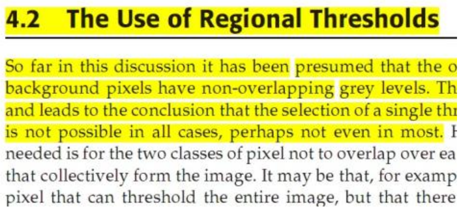
Figure 4.8 Results for Text in Multicolour Corner Gradient Colour (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

The results in Figure 4.6, Figure 4.7 and Figure 4.8 also prove that Otsu's method and Niblack's method are not applicable for segmentation text in colour gradient background. Otsu's method has data loss while Niblack's method results in noisy segmentation. For segmenting texts in colour gradient, the proposed method is the most suitable algorithm.

4.4 Highlighted Text

The segmentation results for highlighted texts are shown in Figure 4.9, Figure 4.10 and Figure 4.11. In different images, the texts are highlighted with yellow, green, brown, blue and gray. Some images have only one highlighted colour and some images have many colour. It can be seen that Otsu's method can give good segmentation for yellow and green colour highlighted text. However, it is not capable for brown and blue colour highlighted which are close to black colour.

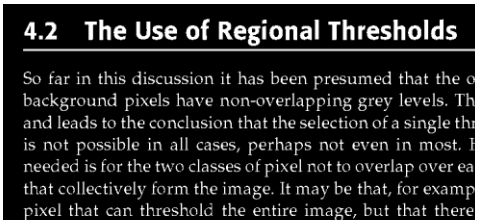
For Niblack's method, the results come up with noise for every colour. It is clear that Niblack's method is not compatible with segmenting highlighted text. The proposed method can give the best segmentation for every highlighted colour. For every result, no data is lost and no noise is found. Thus, it is effective in terms of both segmentation performance and processing time.



4.2 The Use of Regional Thresholds

So far in this discussion it has been presumed that the o
background pixels have non-overlapping grey levels. Th
and leads to the conclusion that the selection of a single thr
is not possible in all cases, perhaps not even in most. H
needed is for the two classes of pixel not to overlap over ea
that collectively form the image. It may be that, for examp
pixel that can threshold the entire image, but that there

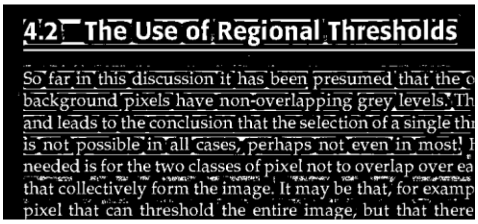
(a)



4.2 The Use of Regional Thresholds

So far in this discussion it has been presumed that the o
background pixels have non-overlapping grey levels. Th
and leads to the conclusion that the selection of a single thr
is not possible in all cases, perhaps not even in most. H
needed is for the two classes of pixel not to overlap over ea
that collectively form the image. It may be that, for examp
pixel that can threshold the entire image, but that there

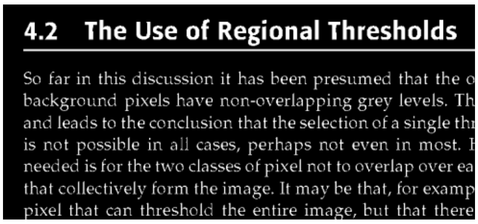
(b)



4.2 The Use of Regional Thresholds

So far in this discussion it has been presumed that the o
background pixels have non-overlapping grey levels. Th
and leads to the conclusion that the selection of a single thr
is not possible in all cases, perhaps not even in most. H
needed is for the two classes of pixel not to overlap over ea
that collectively form the image. It may be that, for examp
pixel that can threshold the entire image, but that there

(c)

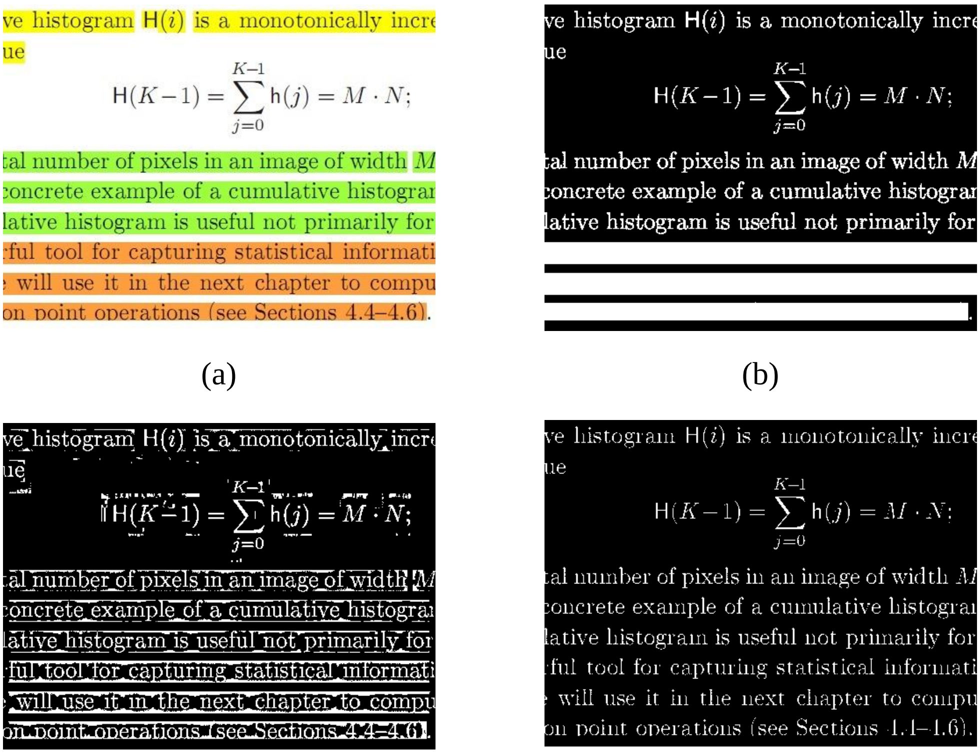


4.2 The Use of Regional Thresholds

So far in this discussion it has been presumed that the o
background pixels have non-overlapping grey levels. Th
and leads to the conclusion that the selection of a single thr
is not possible in all cases, perhaps not even in most. H
needed is for the two classes of pixel not to overlap over ea
that collectively form the image. It may be that, for examp
pixel that can threshold the entire image, but that there

(d)

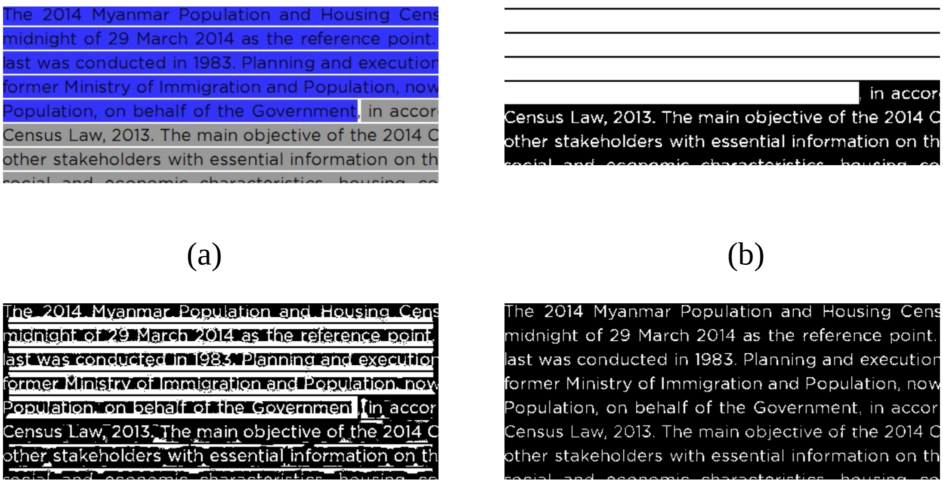
Figure 4.9 Results for Yellow Colour Highlighted Texts (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method



(a) (b)

(c) (d)

Figure 4.10 Results for Yellow-Green-Pink Colour Highlighted Texts (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method



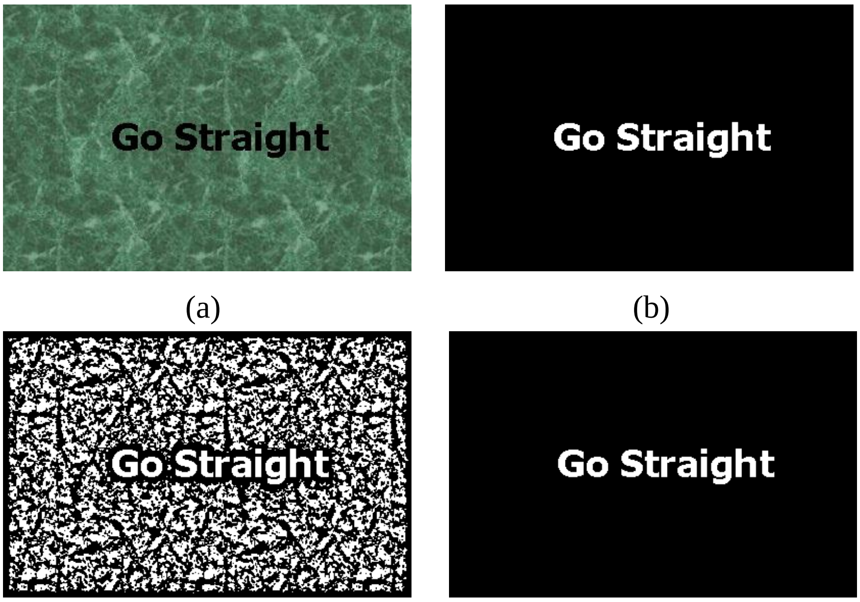
(a) (b)

(c) (d)

Figure 4.11 Results for Blue-Gray Highlighted Texts (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

4.5 Text in Texture

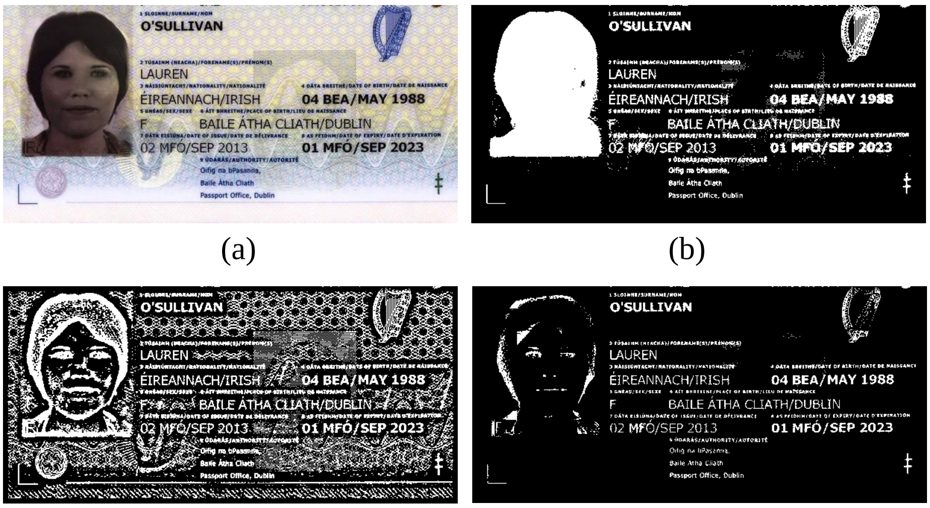
Texts are often found in texture background. Thus, the performance of proposed method is also tested with texts in different texture backgrounds. As shown in Figure 4.12, Figure 4.13 and Figure 4.14, the background textures are green marble, national ID card and passport.



(a) (b)

(c) (d)

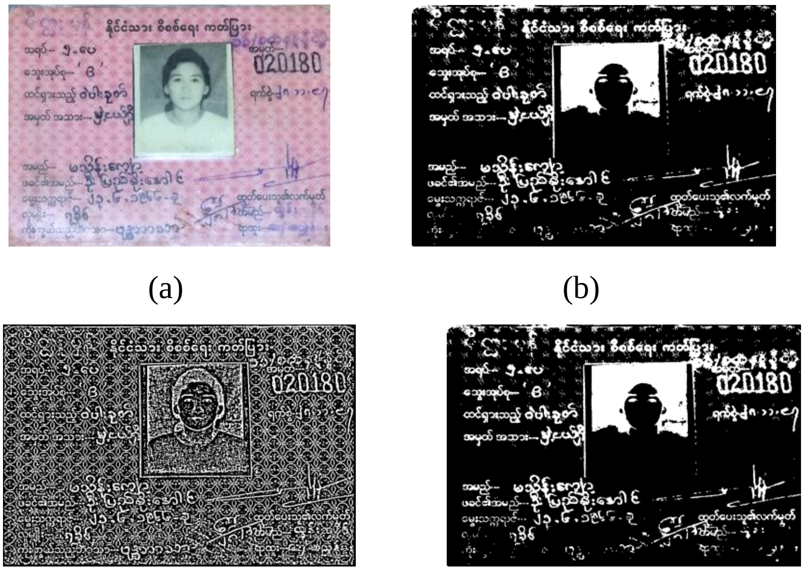
Figure 4.12 Results for Text in Green-Marble Texture (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method



(a) (b)

(c) (d)

Figure 4.13 Results for Text in Passport Texture (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method



(a) (b)

(c) (d)

Figure 4.14 Results for Text in Myanmar ID Card Texture (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

The results show that Otsu's method and the proposed method have the same performance for segmenting text in texture background, while the Niblack's method is giving poor results for all images. The segmented results from Niblack's method still contain the texture.

For passport, the background watermark cannot be removed using Otsu's method. In contrast, the watermark clearly removed by proposed method. For national ID card, the background has pink spot texture and for the passport, the background yellow sports.

4.6 Text in Non-Uniform Illumination

Non-uniform illumination pictures often encountered printed text books, for example old library books, are digitalized. Illumination effect in images appears due lighting condition during capturing. The proposed method is also tested for segmentation of text in non-uniform illumination. Some samples of non-uniform illumination images and the results from three different algorithms are shown in Figures 4.15, Figures 4.16, Figures 4.17, Figures 4.18 and Figures 4.19. Some portions of the images are in dark region and the rest are in bright regions.

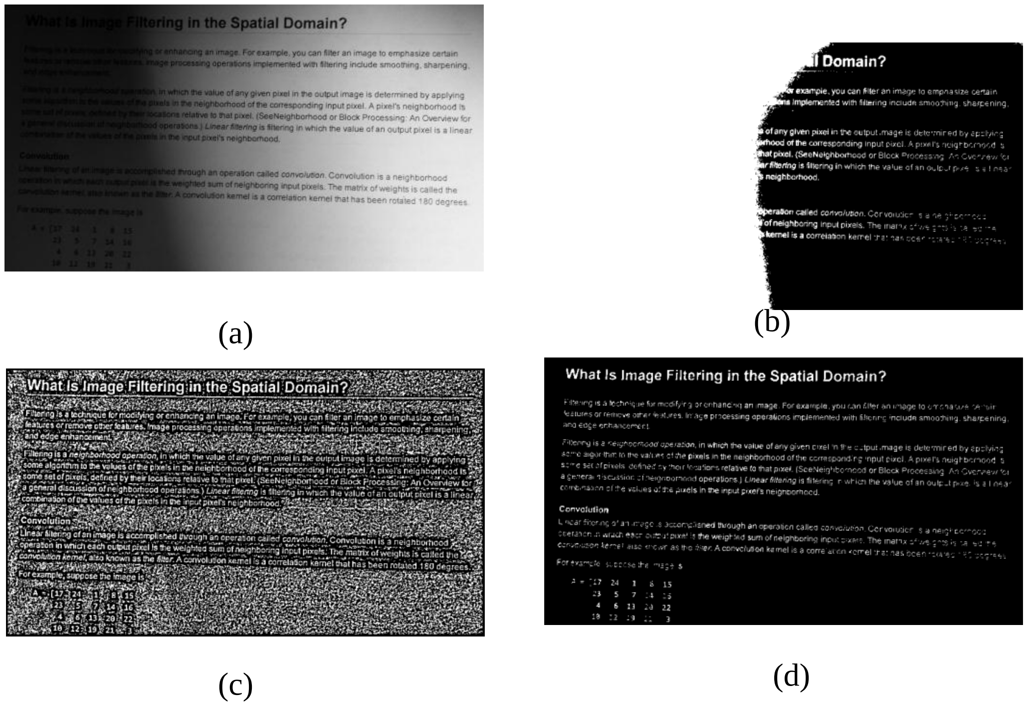


Figure 4.15 Results for Text in Corner Illumination (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

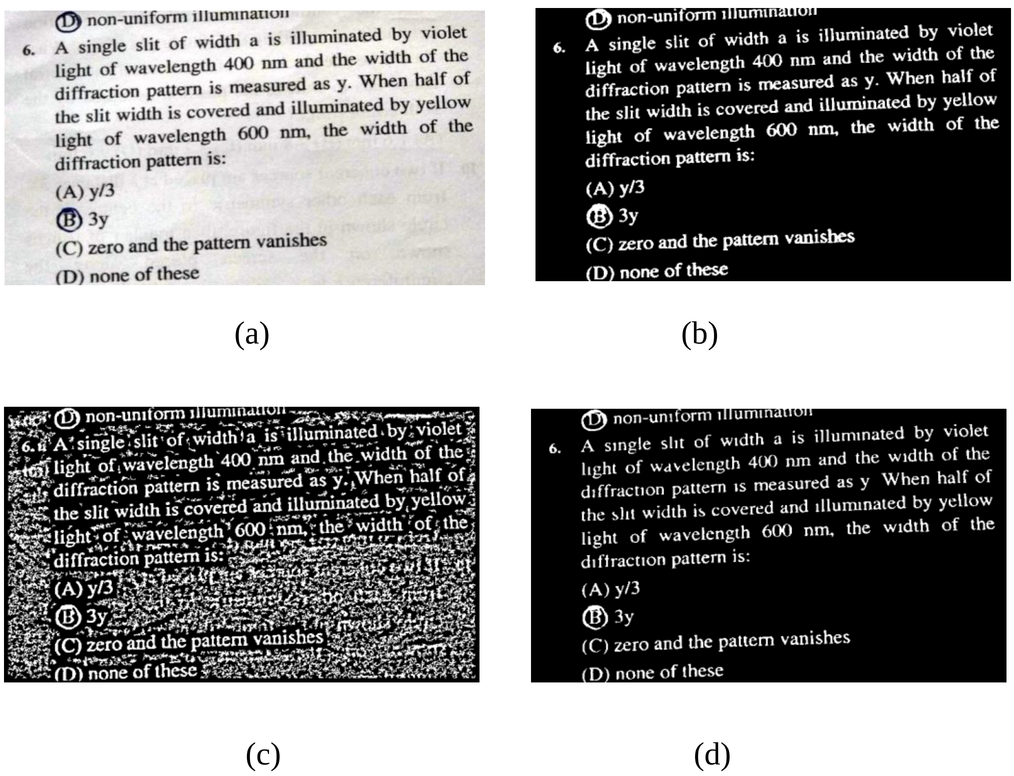


Figure 4.16 Results for Text in Non-Uniform Illumination (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

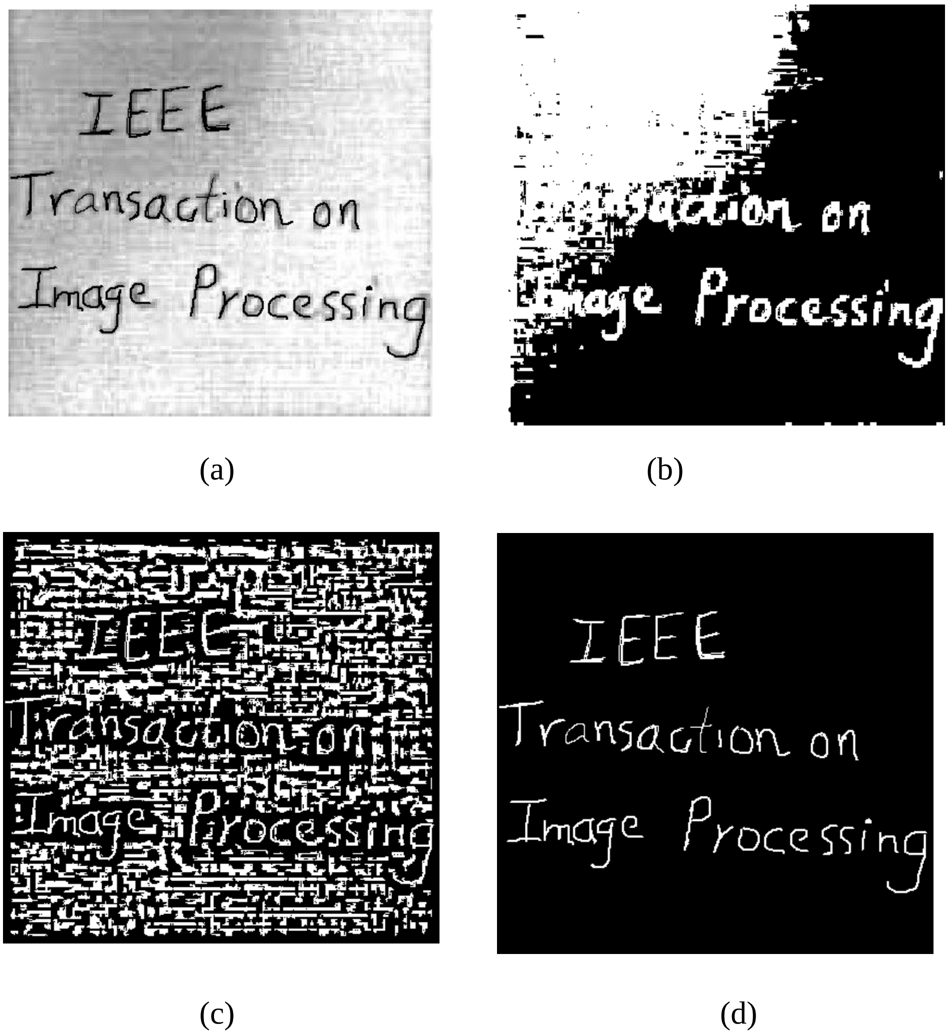


Figure 4.17 Results for Text in Mirror-like Illumination (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

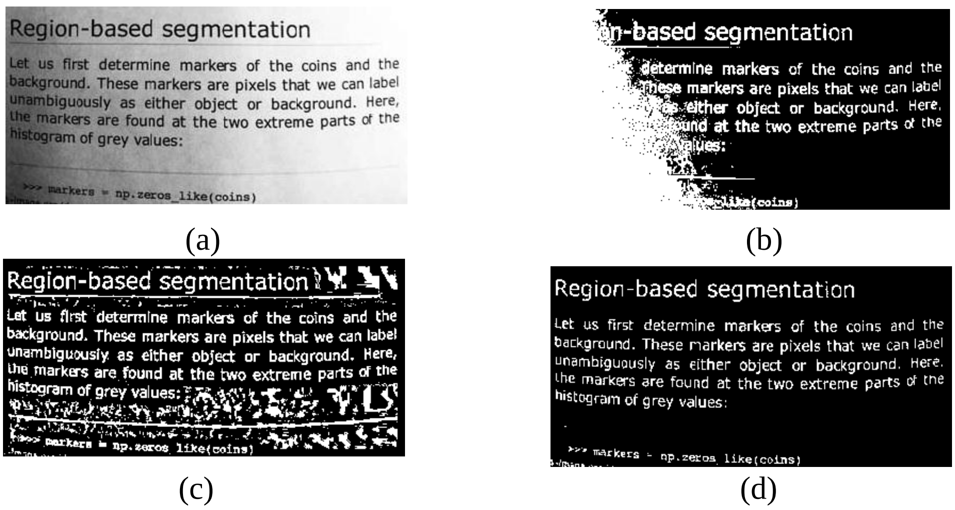
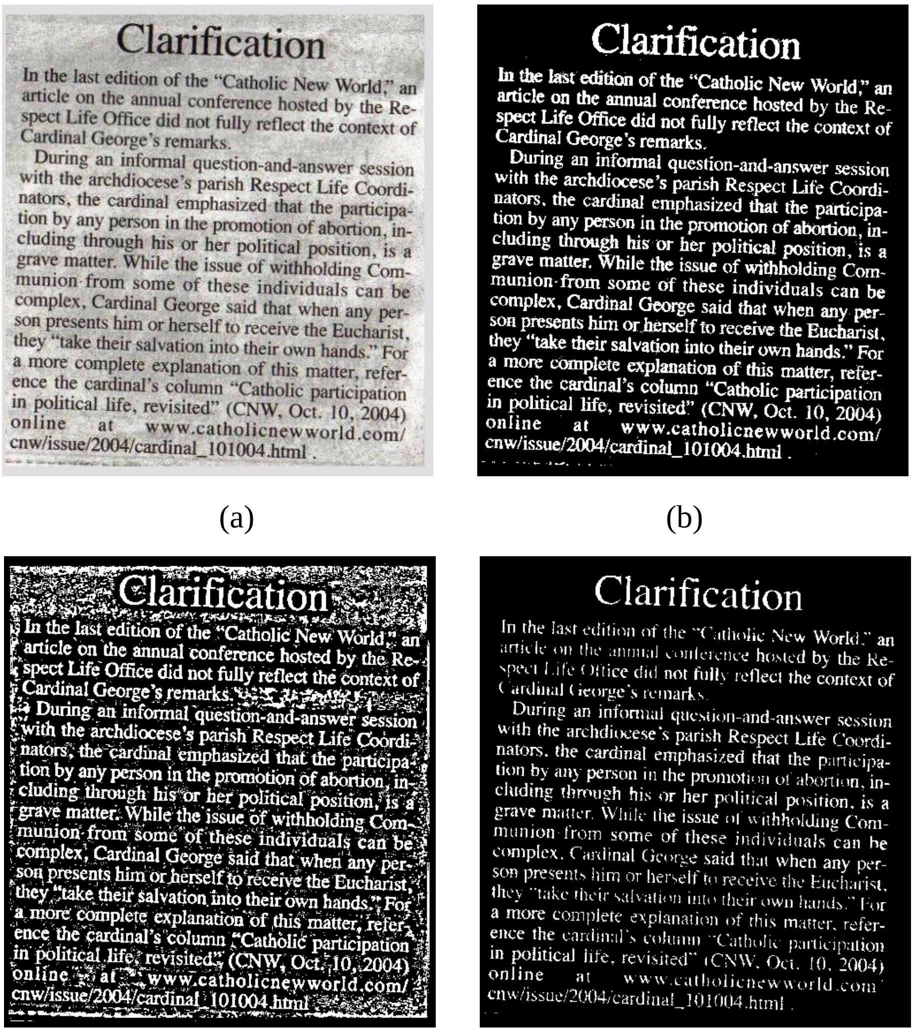


Figure 4.18 Results for Text in Ligth Illumination (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method



Clarification

In the last edition of the "Catholic New World," an article on the annual conference hosted by the Respect Life Office did not fully reflect the context of Cardinal George's remarks.

During an informal question-and-answer session with the archdiocese's parish Respect Life Coordinators, the cardinal emphasized that the participation by any person in the promotion of abortion, including through his or her political position, is a grave matter. While the issue of withholding Communion from some of these individuals can be complex, Cardinal George said that when any person presents him or herself to receive the Eucharist, they "take their salvation into their own hands." For a more complete explanation of this matter, reference the cardinal's column "Catholic participation in political life, revisited" (CNW, Oct. 10, 2004) online at www.catholicnewworld.com/cnw/issue/2004/cardinal_101004.html.

(a) (b)

(c) (d)

Figure 4.19 Results for Text in Dark Illumination (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

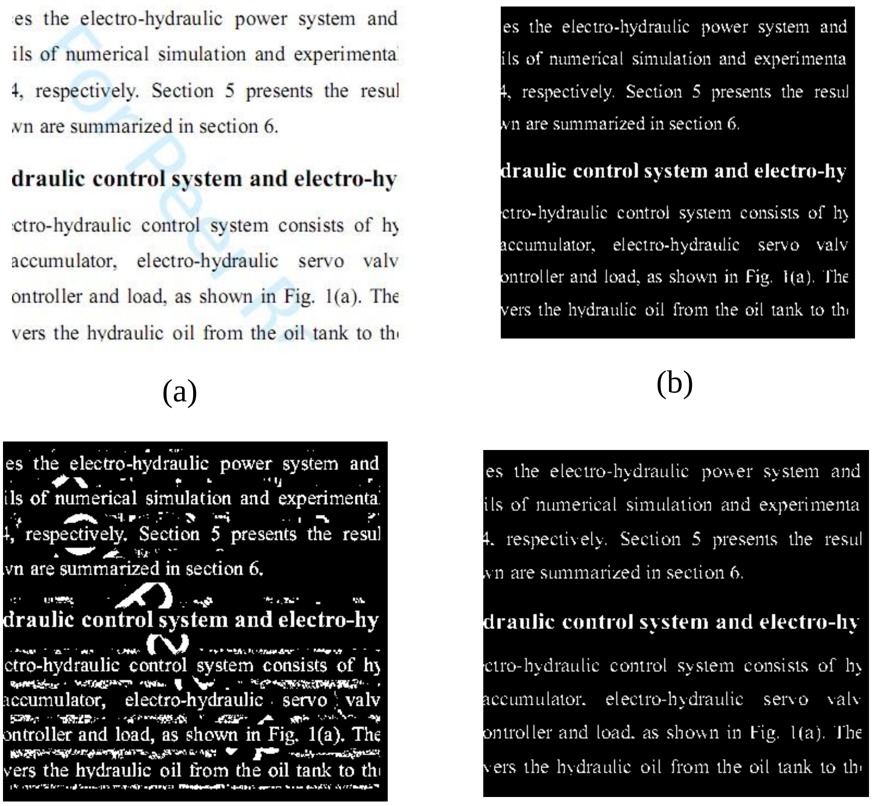
Otsu's method has a large data loss. The results in Figures 4.15, Figures 4.16, Figures 4.17, Figures 4.18 and Figures 4.19 shows that mostly data in the dark region of image are lost in Otsu's method. The results from Niblack's method have the noise appearing around texts. It is also kind of data loss. However, the proposed method gives very acceptable segmentation results compared to that from other methods.

For the case in Figure 4.16, Otsu's method gives acceptable segmentation compared to Niblack's method and proposed method. However, the result of Niblack's method cannot reach to acceptable level because there is still noise in segmented noise.

4.7 Text with Watermark

Some text documents have watermark. Thus, it is also important to test the proposed method for segmenting text with watermark. The comparisons of segmented results from three different algorithms are shown in Figures 4.20, Figures 4.21, Figures 4.22 and Figures 4.23. Some images have text watermark and some images logo watermark.

In Figure 4.20, the watermark colour is light blue. It can be seen that both Otsu's method and proposed method can give a perfect segmentation while the watermark appears in the result from Niblack's method. Besides, there is noise in the result from Niblack's method. In Figure 4.21, the image has both text watermark and logo watermark. All watermarks can be removed by both Otsu's method and proposed method. But Niblack's fails to remove both watermarks. The same condition is attained in the image shown in Figure 4.22, which has green and pink square watermark behind the text. Otsu's method and proposed method work well for segmenting text with watermark.



(a) (b)

(c) (d)

Figure 4.20 Results for Text with Letter Watermark (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

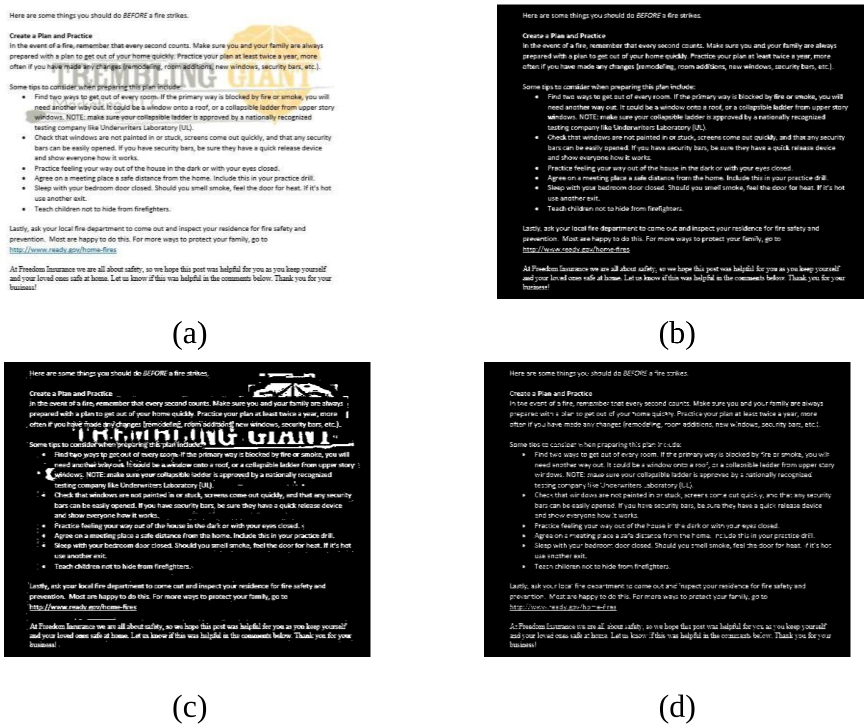


Figure 4.21 Results for Text with Logo Watermark (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

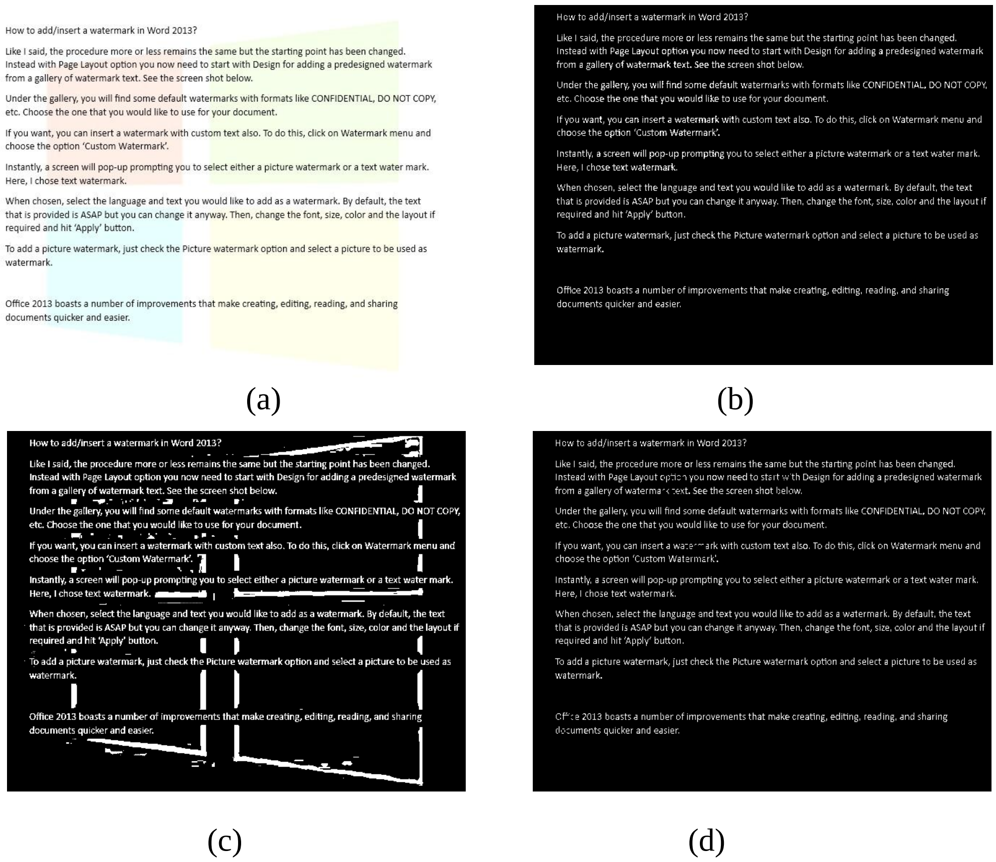
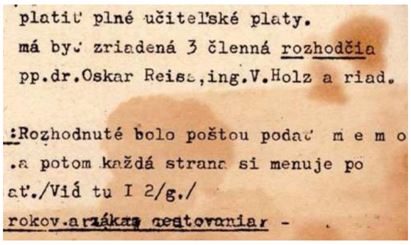


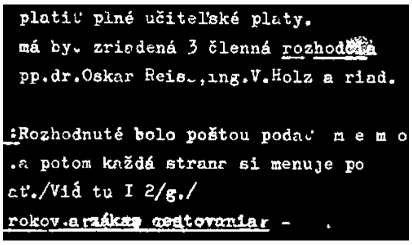
Figure 4.22 Results for Text with Shape Watermark (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

4.8 Degraded Text

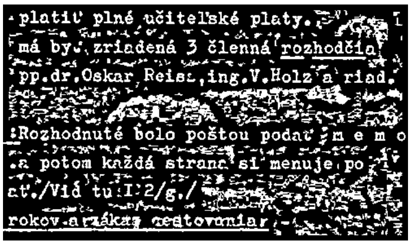
A sample of degraded text segmented using three different algorithms is shown in Figure 4.23. As one can be seen, some data is lost by Otsu's method and the result by Niblack's method is full of noise. Distinctly, the segmented result by proposed method is better one than two other results. Thus, it is a proof for that proposed method is very feasible method for different kinds of text images.



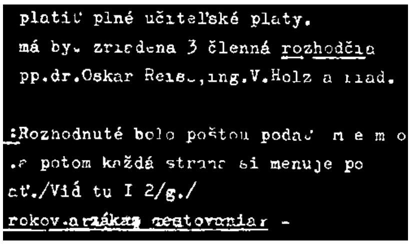
(a)



(b)



(c)



(d)

Figure 4.23 Results for Degraded Text (a) Original Image (b) Otsu's Method (c) Niblack's Method (d) Proposed Method

4.9 Performance Comparison

4.9.1 Processing Time

Table 4.1 shows the processing time for each image shown in previous figures. Processing times are CPU time and the unit is second. It is run by Intel (R) Core (TM) i7-7500U and processor speed is 2.9 GHz. It should be noted that if the processor type and speed are different, the processing time can be different.

It can be seen that the processing time totally depends on image size. Otsu's method has the shortest processing time. Niblack's method has the largest processing time while the proposed method gives comparative processing time as Otsu's method. For image size of 401x701 (281101) pixels, Otsu's method takes 0.0938, Niblack's method takes 5.5313 s and the proposed method takes 1.1875 s. The processing time of proposed method is 12 times longer than that of Out’s method and 5 times shorter than that of Niblack’s method.

Table 4.1 Comparison of Processing Time

Figure No.	**Image Size**	**Otsu’s Method (s)**	**Niblack’s Method (s)**	**Proposed Method (s)**
4.1	83×173 (14359)	0.1875	0.4688	0.4688
4.2	401×701 (281101)	0.0938	5.3281	1.1875
4.3	430×451 (193930)	0.1288	3.5938	0.8906
4.4	207×496 (102672)	0.2344	1.9219	0.5625
4.5	720×960 (691200)	0.1406	12.7813	2.0469
4.6	263×350 (92050)	0.1871	1.8125	0.6563
4.7	480×785 (376800)	0.2813	6.9219	1.3594
4.8	255×405 (103275)	0.2188	1.9219	0.6875
4.9	337×743 (250391)	0.0938	4.6094	1.0781
4.10	371×493 (182903)	0.2500	3.7188	0.8281
4.11	205×506 (103730)	0.1250	2.2909	0.7031
4.12	261×401 (104661)	0.0938	2.2909	0.6250
4.13	292×610 (178120)	0.0938	3.5913	0.8594
4.14	212×325 (68900)	0.1094	1.2031	0.5000
4.15	902×1612 (1454024)	0.3906	18.6875	2.5938
4.16	240×261 (62640)	0.5313	18.9344	1.4688
4.17	651×524 (341124)	0.1875	9.2813	0.9063
4.18	186×377 (70122)	0.2188	1.3438	0.6094
4.19	534×488 (260592)	0.0938	4.8125	1.2031
4.20	332×354 (117528)	0.0938	2.1875	0.8281
4.21	421×528 (222288)	0.2500	4.1094	1.0469
4.22	498×636 (316728)	0.9384	5.7656	1.1094
4.23	121×193 (23353)	0.1718	8.8125	1.1563

4.9.2 Accuracy

Table 4.2 shows the number of result characters for each image shown in previous figures. Sometimes, the number of characters in segmented image is larger than that in original image because noise appears due to segmentation method. Sometimes, the number of characters in segmented image is smaller than that in original image because some characters are lost due to segmentation method. Mostly, Niblack's method has the lowest accuracy compared to Otsu's method and the proposed method.

Table 4.2 Comparison of Number of Characters

Figure No.	Original number of characters in images	Otsu's Method of characters in images	Niblack's Method of characters in images	Proposed Method of characters in images
4.1	4	4	4	4
4.2	56	43	98	56
4.3	82	17	76	82
4.4	15	1	83	15
4.5	698	354	996	705
4.6	75	81	136	70
4.7	1082	1065	2014	1082
4.8	73	58	265	71
4.9	386	373	507	386
4.10	370	156	305	356
4.11	325	132	153	325
4.12	10	11	258	11
4.13	204	204	404	206
4.14	89	84	195	37
4.15	1000	440	3656	1000
4.16	271	271	1056	298
4.17	34	62	408	34
4.18	226	124	285	226
4.19	850	850	817	845
4.20	379	379	413	370
4.21	938	938	758	938
4.22	871	871	881	868
4.23	193	193	511	193

Table 4.3 shows the comparison of accuracy for each image shown in previous figures. Accuracy is calculated by using equation (3.7).

Table 4.3 Comparison of Accuracy

Figure No.	Otsu's Method Accuracy (%)	Niblack's Method Accuracy (%)	Proposed Method Accuracy (%)
4.1	100	100	100
4.2	87	74	100
4.3	37	100	100
4.4	13	31	100
4.5	67	81	98
4.6	96	71	96
4.7	92	76	100
4.8	88	43	98
4.9	96	88	100
4.10	64	89	100
4.11	10	39	100
4.12	95	7	95
4.13	100	67	99
4.14	97	63	90
4.15	61	42	100
4.16	100	40	99
4.17	70	16	100
4.18	71	88	100
4.19	100	90	98
4.20	100	88	98
4.21	100	89	100
4.22	100	99	100
4.23	100	54	100

Table 4.4 shows comparison of average accuracy for previous figures.

Table 4.4 Comparison of Average Accuracy

Otsu's Method Average Accuracy (%)	Niblack's Method Average Accuracy (%)	Proposed Method Average Accuracy (%)
80.1739	66.7391	98.7391

Figure 4.24 shows comparison of average accuracy by segmentation methods.

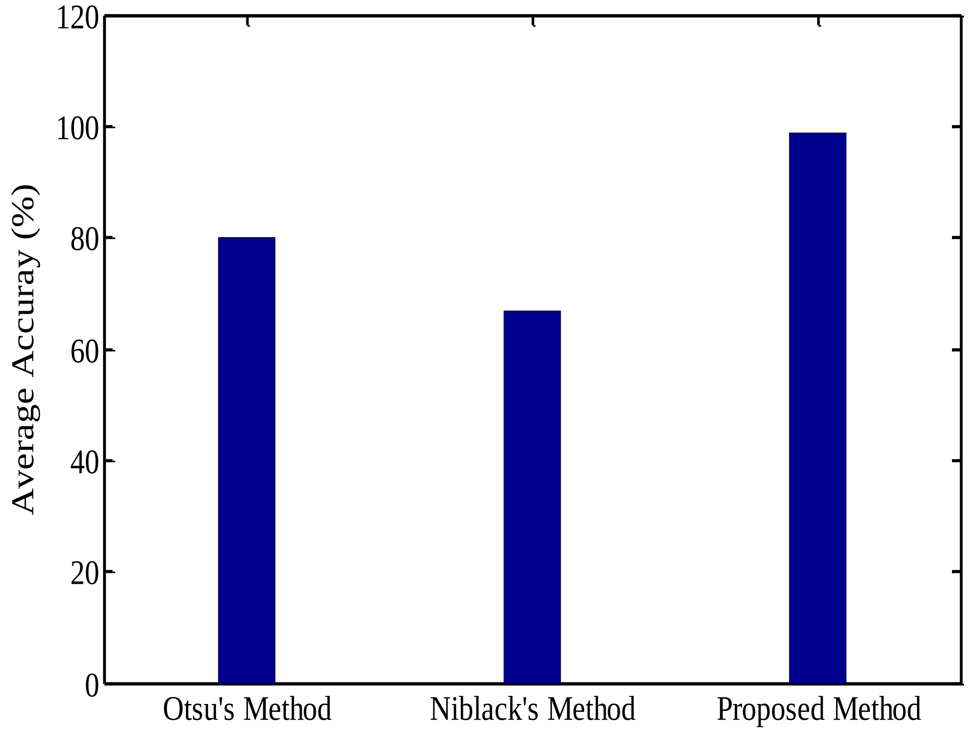


Figure 4.24 Comparison of Average Accuracy by Segmentation Methods

CHAPTER 5
CONCLUSION, LIMITATION AND FURTHER EXTENSION

5.1 Conclusion

Segmentation process is a key step in text recognition. There are many global and local thresholding methods for segmentation. However, local thresholding method is not compatible for text images with non-uniform illumination. Local thresholding methods are more effective for segmenting images with non-uniform illumination. On the other hand, the main drawback of local thresholding methods is very long processing time and these are not compatible for segmenting text with colour gradient background.

In this work, a new local thresholding method is proposed for segmenting text images with non-uniform colour. It is based on intensity magnification. Being different from local thresholding methods, the images are divided as non-overlapped small local windows. The magnification factor for each local window is calculated according to the dominant intensity levels, maximum intensity, minimum intensity and intensity range.

The proposed method is tested with different kinds of text images such as text in uniform colour, text in multicolour, text in gradient background, highlighted text, text in texture, text in non-uniform illumination, text with watermark and degraded text. The performance of proposed method is measured in terms of processing time and accuracy in segmentation. The performance of proposed method is compared with that of Otsu's thresholding method and Niblack's method.

For text images with simple colour background, all methods including proposed method give perfect results. For text images with multicolour background, Otsu's method has some data loss when the background colour is close to text colour. Niblack's method also gives noise around characters. However, the proposed method can give perfect results. For text with colour gradient background, Otsu's method has data loss depending on gradient colour and gradient condition. Niblack's method has noise for every gradient colour background.

For highlighted text, Otsu's method works well with yellow colour and green colour highlighted text. But it fails to segment brown colour and blue colour highlighted text because highlighted colour is close to text colour. Niblack's method gives noise for every highlighted colour. For text with texture background, Otsu' method works well but Niblack's method still have noise. For text in non-uniform illumination, Otsu' method has data loss in darker regions, the weakness is the existence of noise in the result. For text with watermark, Otsu's method gives perfect segmented results. However, the watermark appears as noise in Niblack's method.

Compared to Otsu's method and Niblack's method, the proposed method gives the better results. Mostly, the results from proposed method are perfect with 100% accuracy. The lowest accuracy of proposed method is 80%. Sometimes, Otsu's method losses all data. Thus, its lowest accuracy is 0% and the highest efficiency is 100%. For Niblack's method, the lowest accuracy is 10%, it gives 100% accuracy only for text with simple colour background.

In terms of processing time, Otsu's method has the shortest processing time while Niblack method has the longest processing time. Processing time is dependent of image size. The processing time of proposed method is 12 times larger than that of Otus's method and 5 times smaller than that of Otsu's method.

5.2 Limitation

The limitation of proposed method is that it has low performance for text with stone inscriptions and text in periodic table. Also, the proposed method is compatible only for segmenting black colour character.

5.3 Further Extension

The performance of proposed method also depends on the double base lines, window size and threshold for dominant pixel. Thus, the effect of those parameters should be investigated as the further works.

AUTHOR'S PUBLICATION

[1] Thidar Aung, Thin Lai Lai Thein, "A Robust Local Thresholding Method for Segmentation of Non-Uniform Coloured Documents", Proceedings of National Journal of Parallel and Soft Computing, Volume 01, Issue 01, NJPSC, University of Computer Studies, Yangon, Myanmar, March, 2019, pp.193-198.

[2] Thidar Aung, Thin Lai Lai Thein, "Local Thresholding Method for Image Segmentation Based on Intensity Magnification", Proceedings of the 2nd International Conference on Engineering Education and Innovation, Technological University (Hmawbi), Hmawbi Township, Yangon, Myanmar, November 7, 2019, pp.472-477.

REFERENCE

[1] A. Alaei, P. Pratim Roy and Umapada, Pal. “Logo and seal based administrative document image retrieval: a survey”. Computer Science Review, 22, pp. 47-63, 2016.

[2] A. P. Giotis, G. Sfikas, B. Gatos and C. Nikou. “A survey of document image word spotting techniques”. Pattern Recognition, 68, pp. 310-332, 2017.

[3] B. Bataineh, S. Norul Huda Sheikh Abdullah and K. Omar. “An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows”. Pattern Recognition Letters, 32(14), pp.1805-1813, 2011.

[4] C. Eyupoglu. “Implementation of Bernsen’s Locally Adaptive Binarization Method for Gray Scale Images”.The Online Journal of Science and Technology,7(2), p.68, April 2017.

[5] C. Wolf, J.M. Jolion, J.M. and F. Chassaing. “Text localization, enhancement and binarization in multimedia documents”. In Object recognition supported by user interaction for service robots, IEEE, Vol. 2, pp. 1037-1040, 2002.

[6] Er. N, Kaur and Er. R, Kaur, R. “A review on various methods of image thresholding”. International Journal on Computer Science and Engineering, 3(10), p.3441, Oct 2011.

[7] H. Om and M. Biswas. “A new image denoising scheme using soft-thresholding”. Journal of Signal and Information Processing, 3(03), p.360, 2012.

[8] J. Bernsen. “Dynamic thresholding of gray-level images”. Proc.8th , International Conference, on Pattern Recognition, Paris, 1986, pp. 1251-1255.

[9] J. Sauvola and M. Pietikäinen. “Adaptive document image binarization”. Pattern recognition, 33(2), pp.225-236, Feb 2000.

[10] K. Khurshid, I. Siddiqi, C. Faure and N. Vincent. “Comparison of Niblack inspired binarization methods for ancient documents”. In Document Recognition and Retrieval XVI (Vol. 7247, pp. 72470U). International Society for Optics and Photonics,2009.

[11] K. Mieloch, M. Preda, and A. Munk. “Dynamic threshold using polynomial surface regression with application to the binarization of fingerprints”. In Biometric Technology for Human Identification II (Vol. 5779, pp. 94-105, Mar 2005).

[12] L. O'Gorman, and R. Katsuri. Guest Editor's Introduction: “Document Image Analysis Systems”. Computer, 25(7), pp.5-8, July 1992.

[13] N. George. “Twenty years of document image analysis in PAMI”. IEEE Transactions on Pattern Analysis & Machine Intelligence, (1), pp.38-62, 2000.

[14] N. Otsu. “A threshold selection method from gray-level histograms”. IEEE transactions on systems, man, and cybernetics, 9(1), pp.62-66, 1979.

[15] P. Puneet and N. Kumar Garg. “Binarization Techniques used for grey scale images”. International Journal of Computer Applications, 71(1), 2013.

[16] R. Chaudhari. and D. Patil. “Document image binarization using threshold segmentation”. International Journal of Innovative Research in Computer and Communication Engineering, 3(3), pp.1873-1876, March 2015.

[17] R. Firdousi. and S. Parveen. “Local thresholding techniques in image binarization”. International Journal of Engineering and Computer Science, 3(3), pp.4062-4065, Mar 2014.

[18] R. Saini.“Document image binarization techniques, developments and related issues: a review”. International Journal of Computer Applications, 116(7), pp.41-44, April 2015.

[19] S. Eskenazi, Gomez-Krämer, and JM Ogier. “A comprehensive survey of mostly textual document segmentation algorithms since 2008”. Pattern Recognition, 64, pp.1-14, April 2017.

[20] S. Wu and A. Amin. “Automatic thresholding of gray-level using multistage approach”. In Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. IEEE, pp. 493-497, Aug 2003.

[21] T. Romen Singh, S. Roy, O.I .Singh, T. Sinam and Kh. M. Singh. “A new local adaptive thresholding technique in binarization”. arXiv preprint arXiv:1201.5227, Jan 2012.

[22] T. Romen Singh, S. Roy and Kh. M. Singh. “Local adaptive automatic binarisation (LAAB) ”. International Journal of Computer Applications, 40(6), pp.27-30, Feb 2012.

[23] W. Niblack. “An introduction to digital image processing” Strandberg Publishing Company, Oct 1986.

[24] Y. Chen. and L.Wang. “Broken and degraded document images binarization”. Neurocomputing, 237, pp.272-280, May 2017.