

Image and Video Processing Coursework

CS3330: Image and Video Processing Coursework: Document Image Binarization

Damon Tran

Student Number: 160111190

Background

What is the aim of document image binarization?

Document image binarization is essentially a method that is used for taking an image of a text document and separating different objects within it generally foreground (text) from background, using various approaches via thresholding on a grayscale image, to separate the text from the rest of the picture to make it more readable [1].

The typical documents you would use this on, would be where the image is affected by various variables that make the document hard to read as displayed in (figure 1) below [2]. You can see that the black text although visible, it is slightly difficult to see due to the dark patches and stains. However, through applying image binarization you can solve this issue.

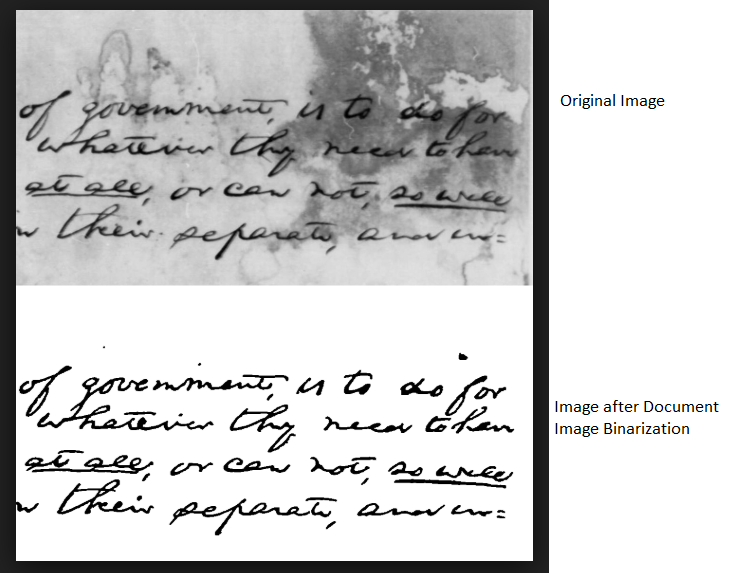


Figure 1

Why is binarizing a document image useful? (hint: binarization is a mid-level image processing technique, and the output of mid-level techniques is often fed into high-level techniques).

It is useful because mid-level processes take an image as an input, analyses and processes the image to obtain the relevant attributes which are then passed to a high-level processing technique to apply to said attributes [3]. Acting as a phase in the processing pipeline, setting the image up for other steps to achieve a specific outcome. Meaning that we can apply any high-level technique we wish to the output we have generated from the mid-level process, depending on our goal the flexibility of what we can do is quite varied.

For example, if we wanted to restore historical documents from many years ago despite the degradation of the paper and any issues that may obstruct us from reading it, If we wanted we could apply image binarization so that we could and then apply any high level technique depending on what we want to produce.

• What are the existing approaches to document image binarization? Briefly outline THREE different algorithms for document binarization found through your research and discuss the way in which the aims and performances of these algorithms differ in academic literature.

Existing approaches to document image binarization

(Note they differ because the approaches are only good depending on context, context mean that they are being applied to different images and thus different goals it is unfair to say that one algorithm is better than the other at something because they have different aims)

There are many approaches but the three I have researched and found are Parker, Eikvil and Sauvola.

Parker The Edge Level Thresholding (ELT) algorithm is based off the idea that it impossible to choose one single threshold value that can efficiently binarize the whole image [4]. Instead, it uses multiple threshold values which are derived from each pixel and applied to various sub-regions to solve the problem of thresholding an image which is affected by illumination [4]. Results show when compared to other binarizing techniques that it performs better given the image has poor or close to none illumination [4].

Eikvil Fast Adaptive Binarization (FAB) works with the concept of creating sub-images of the image to binarize and then identifying and classifying pixels to separate the image text from background [5]. Specifically, optimal for grayscale images where there are large discrepancies in the grey levels of the image (foreground and background) as well as where intensities of pixels within the locality of an area differs [5]. Results in the literature indicate that it performs well in comparison to simple binarization techniques and is efficient in terms of computation time [5].

(Sauvola)Adaptive Document Image Binarization (ADIB) is designed with the intended goal of binarizing a more general range of text document images that are affected by different types of image defects [6] , with the underlying concept that each part of the picture (text, background and picture) are intrinsic to the whole image and solves issues that many document image binarization algorithms face such as, noise and illumination by splitting up the image into equal size neighborhoods and deriving a threshold value for each region [6]. Sauvola performs well and operates to a similar level when compared to Niblack’s and Eikvil’s. Contrasting with Parker’s, which did poorly showing the algorithm suffers where illumination varies seen in figure 2 [6].

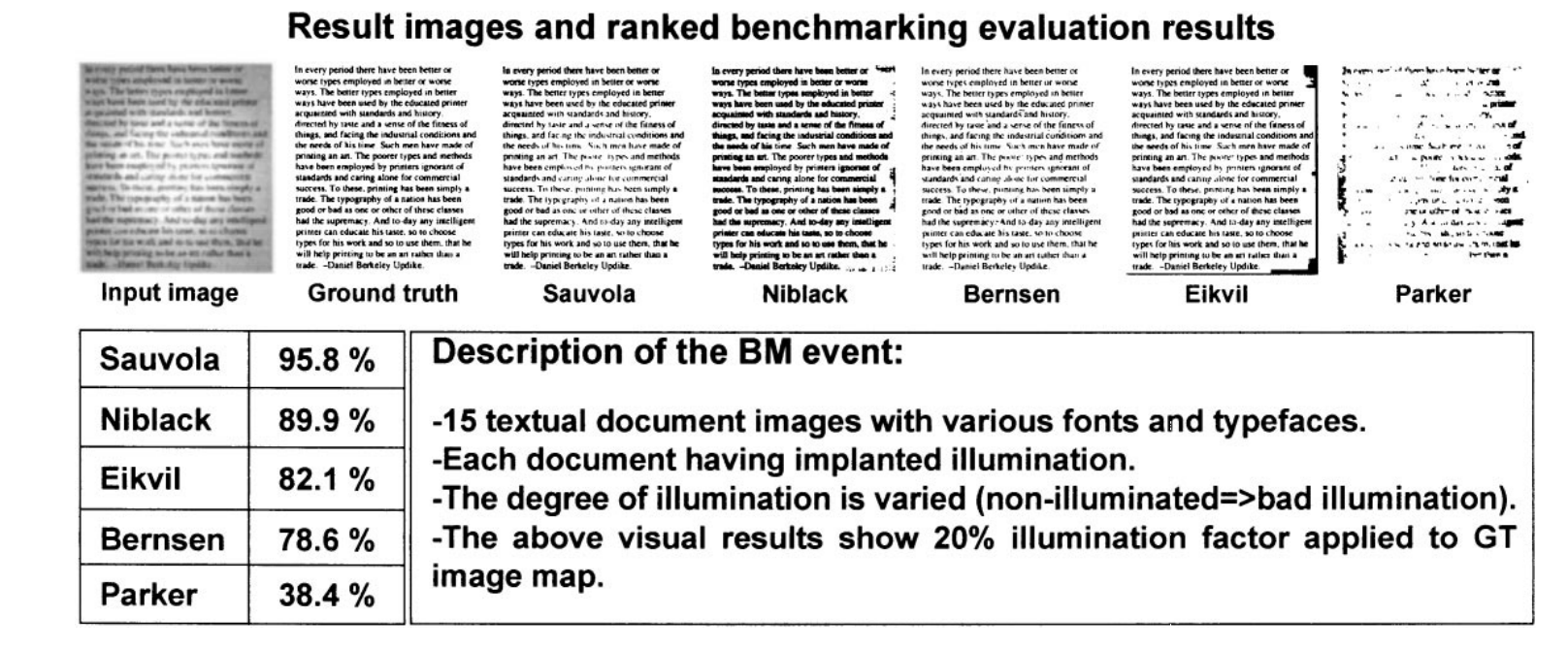


Figure 2 - Visual and numeric results on the comparison algorithms applied to illuminated, textual images. [6]

The performance of these algorithms differs in respect to their aims as they try to binarize text document images with a problem in mind, thus have been designed to cater to that problem. Each approach uses the same logic of thresholding but their methods of creating the thresholds vary which in turn will produce varying results as seen in the aforementioned figure 2 [6]. Depending on degradation type or condition of the document image, each algorithm can perform better or worse than the others.

Section2: Alrgorithm (20%)

Motivation

o Motivate why it would be suitable for document image binarization, making reference to the literature.

Sauvola is suitable for document image binarization because it can binarize images, reduce noise and illumination producing consistently good results [6]. Including the fact, it has been compared to algorithms well known for performance and even outperforming them [6]. Because of the algorithms adaptive nature, it can perform well across a variation of document types with different degradations proving it to be quite robust [6].

o Discuss any drawbacks to the algorithm, again, making reference to the literature.

Despite how well the algorithm performs, there are a few drawbacks that have been indicated in research papers. Namely, the fact that Sauvola only performs well given the text document is affected by noise or illumination [6]. But suffers when the image is quite clear (“when contrast between the foreground and background is low or if the text is in form of thin pen stroke” [8]) because there is little variation in pixels, it is harder to obtain precise threshold values to differentiate whether the pixels are part of the text or background[7] which can be seen in figure 3 in cases like this, performance of the algorithm degrades the clearer the image is. Another disadvantage is that Sauvola is requires a lot of computational power [7] because it needs lots of resources to be able to binarize a text document image.

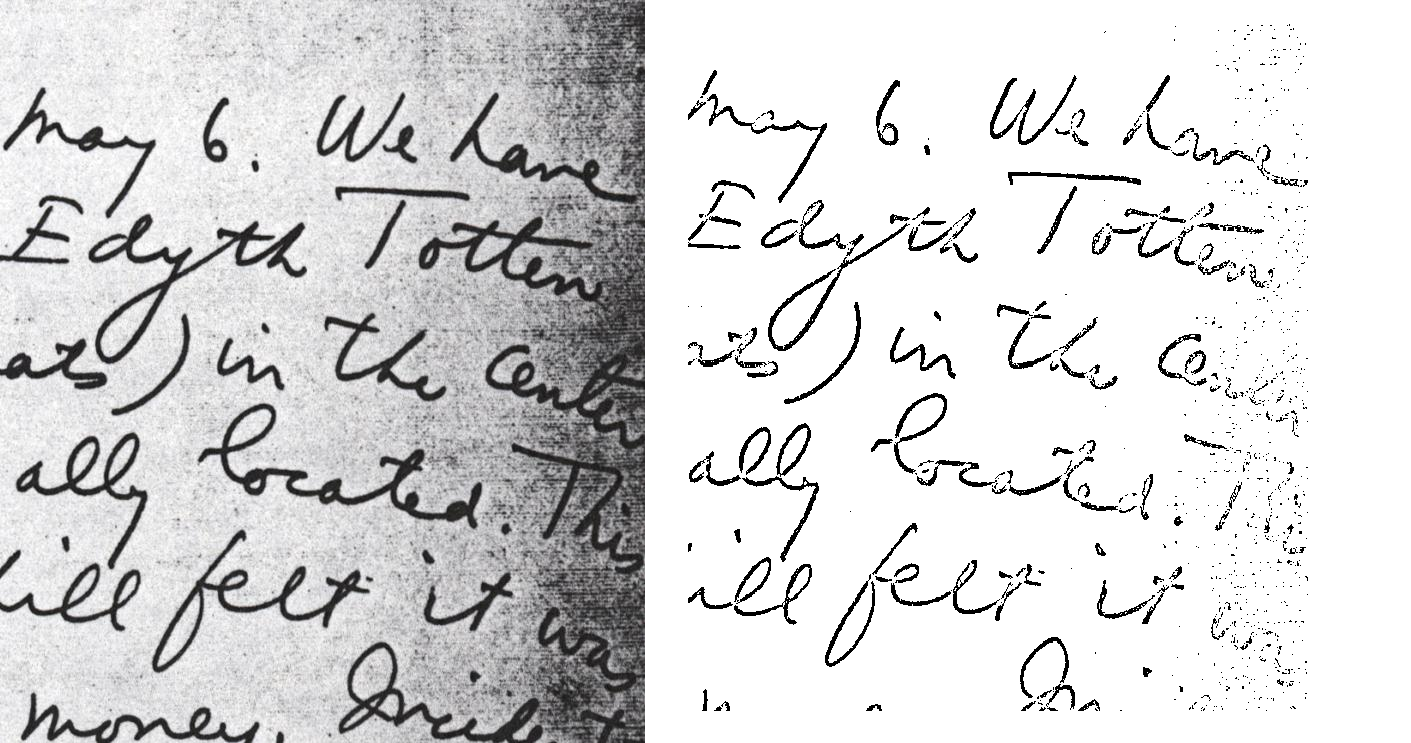


Figure 3- Original Image (Left) Sauvola applied to image (right)

o Describe how the algorithm works in your own words (e.g. using high-level pseudocode).

Sauvola takes a grayscale image and outputs a binary image which represents the segmentation, in the context of text document images the segmentation is the separation of text from background.

It does this by using a local adaptive approach to obtain the threshold values. First taking the grayscale image and separates it into equal sized neighborhoods n x n (n being the window size which is determined manually [8])

After, it finds the mean values of each neighborhood by using a filter on the greyscale image which outputs a new matrix consisting of all the average values within each neighborhood. With this we then find the standard deviation of all pixels from the original greyscale image to our mean values, which can then be used in the thresholding formula, see figure 4. Lastly, we use the threshold value to determine whether pixels in the original greyscale Image are text or background and output it.

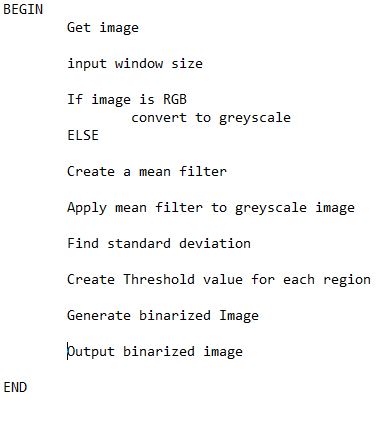




Figure 4 Thresholding formula used in Sauvola [8]

3. Results (25%)

• Describe how you will test the applicability of your method to your chosen problem application (your methodology). You may want to propose some subjective or objective analysis of your results.

I will do a combination Objective and subjective analysis of my algorithm on the test images provided, including how it compares to the ground truth images.

o At minimum, you should use Matlab to apply your method to a set of test images and comment on the results.

o You may gain extra marks for objective analysis (comparison to ground truth images is a good candidate).

• Visually demonstrate the results of applying your segmentation algorithm and discuss how well it has binarized each document image. As stated above, you should aim to binarize at least one of the test images with reasonable accuracy.

• Present and discuss any further evidence that you have gathered, as per your methodology

|  |  |  |
| --- | --- | --- |
| Test Number | Test Images | Result |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

Comments on the results:

Looking at test 1, we can see that the Sauvola algorithm has segmented the image from text and background quite clearly making it more readable for the text on the right that was illuminated in a very dark shade. The output shows the dark shade is still there due the weakness stated earlier—that the algorithm tends to suffer where there is little variation in pixel intensity [7] thus making it harder for it to segment the text from the dark shaded background in this image. Overall It has segmented the original image well and has achieved its intended design.

Test 2 –

The original image as you can see has a brown parchment type background and black ink text, suggesting that Sauvola should be able to segment this well due to the large variation in intensities which has been indicated to be a strength of the algorithm [6] in various literature.

The result shows the original image completely binarized as you can see text separated from the background making it clearly readable, even where the original image had a dark shade of bleeding coming through which made the writing obscure. Although there are still remnants of the bleeding effect in the output, it is very minimal. It is notable however, that some of the writing is very faint, this is one drawback of the algorithm where it suffers if there is little variation, in this case the faint pen stroke as shown in figure 5 which led to poor binarization of this specific word.

https://i.gyazo.com/c865d94b8eaf71ada386dcf2d88613db.png

Figure 5- Thin pen stroke on binarized Image (right) Thin pen stroke from original Image(left)

Test 3 –

Here the result is very good, as it has managed to separate the text from background including areas where text from the page below were made faintly visible on the front page (see figure 6 if unsure what is meant). In the output these areas have been completely erased and all is left is the text, which in fact is very sharp and vivid as well clear separation from the background.

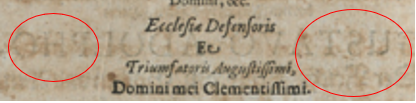


Figure 6- Original Image showing areas where text from another page is visible

Test4 –

In this result the algorithm has clearly performed worst on this document image than any other so far. The text is very faint and, in some cases,, words are not readable. It is safe to say

**References**

1. Karthika M, Ajay James , A Novel Approach for Document Image Binarization using Bit-Plane Slicing, Romania , Tirgu-Mures, 2014.
2. https://webee.technion.ac.il/labs/cgm/Computer-Graphics-Multimedia/Software/Software.html, 26th November 2018.
3. Dr. Harry Goldingay, Image and Video Processing CS3330 Introduction, Aston University.
4. J. R. Parker, C. Jennings, Thresholding Using an Illumination Model, ICDAR ’93, Japan, 1993.
5. Line Eikvil, Torfinn Taxt and Knut Moen, A Fast Adaptive Method for Binarization of Document Images, France, 1991.
6. J. Sauvola, M. PietikaKinen Adaptive document image binarization, Machine Vision and Media Processing Group, Infotech Oulu, University of Oulu, P.O. BOX 4500, FIN-90401 Oulu, Finland, 21 January 1999.
7. Mahua Nandy (Pal) , Satadal Saha, An Analytical Study of different Document Image Binarization Methods, Department of CSE, MCKV Institute of Engineering, Howrah, India.
8. Rashmi Saini, Document Image Binarization Techniques, Developments and Related Issues: A Review, Assistant Professor, Govind Ballabh Pant Engineering College Pauri Garhwal, 246001, Uttarakhand, India.