LOCAL BINARIZATION FOR DOCUMENT IMAGES CAPTURED BY CAMERAS WITH DECISION TREE

Naser Jawas, Randy Cahya Wihandika, and Agus Zainal Arifin

Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember, Kampus ITS Sukolilo, Surabaya, Jawa Timur, 60111, Indonesia

E-mail: naser.jawas11@mhs.if.its.ac.id

Abstract

Character recognition in a document image captured by a digital camera requires a good binary image as the input for the separation the text from the background. Global binarization method does not provide such good separation because of the problem of uneven levels of lighting in images captured by cameras. Local binarization method overcomes the problem but requires a method to partition the large image into local windows properly. In this paper, we propose a local binariation method with dynamic image partitioning using integral image and decision tree for the binarization decision. The integral image is used to estimate the number of line in the document image. The number of line in the document image is used to devide the document into local windows. The decision tree makes a decision for threshold in every local window. The result shows that the proposed method can separate the text from the background better than using global thresholding with the best OCR result of the binarized image is 99.4%.

Keywords: *binarization, decision tree, document images, image partitioning, local window*

Abstrak

Pengenalan karakter pada sebuah dokumen citra yang diambil menggunakan kamera digital membutuhkan citra yang terbinerisasi dengan baik untuk memisahkan antara teks dengan *background*. Metode binarisasi global tidak memberikan hasil pemisahan yang bagus karena permasalahan tingkat pencahayaan yang tidak seimbang pada citra hasil kamera digital. Metode binarisasi lokal dapat mengatasi permasalahan tersebut namun metode tersebut membutuhkan metode untuk membagi citra ke dalam bagian-bagian *window* lokal. Pada *paper* ini diusulkan sebuah metode binarisasi lokal dengan pembagian citra secara dinamis menggunakan *integral image* dan *decision tree* untuk keputusan binarisasi lokalnya. *Integral image* digunakan untuk mengestimasi jumlah baris teks dalam dokumen citra. Jumlah baris tersebut kemudian digunakan untuk membagi citra dokumen ke dalam *window* lokal. Keputusan nilai *threshold* untuk setiap *window* lokal ditentukan dengan *decision tree*. Hasilnya menunjukkan metode yang diusulkan dapat memisahkan teks dari dokumen citra lebih baik dari binarisasi global dengan tingkat pengenalan OCR hingga 99.4%.

Kata Kunci: *binerisasi, citra dokumen, decision tree, membagi image, window lokal*

1. Introduction

Character recognition in a document image captured by a digital camera requires a good binary image as the input for the separation of the text from the image. The characteristic of a document image captured by a digital camera is the uneven levels of lighting in some areas of the image. It is a challenge in binarizing document images. Global binarization method that is commonly used to binarize image cannot well separate the text from the background. Many researchers have proposed to use local binarization for such case.

In [1], Park et al. binarized document images by partitioning them into 8×8 local windows. Each local window is then categorized into one of two classes: concentric and non-concentric window. The concentric window is the ones that contain text, and non-concentric window is the ones that contain background. The concentric window have the Otsu threshold value that is less than or equal to the absolute value of the total coefficient of the low frequency of the discrete cosine transform. The other windows are categorized into the non-concentric window. For the concentric window, the threshold value is determined using Otsu's thresholding algorithm, while for the non-concentric window, 0 is used for the threshold value. The categorization of the local windows into the two categories makes some local windows, which have a small part of text, categorized into the background window so that types of images will be white (threshold=0) and miss some text.

Another research [2] by Chou et al. tried to enhance the method in [1] by proposing local binarization with decision cascade. The rule used to take the decision was added to four rules: text window, black-background window, white-background window, and white-background window that partially contain text from the previous window. The categorization is not based on the comparison of the Otsu threshold value with the absolute value of the total coefficient of the discrete cosine transform, but the comparison with the standard deviation, mean, and the Otsu threshold value of the neighboring windows. In this paper, the image is divided into 5×5 windows. This overcomes the drawback in the method used in [1].

Chou et al. [3] improved their previous paper by using SVM training as the replacement to the decision cascade to improve the action decision. It was also concluded that the division of the image into 3×3 windows is the most satisfactory number to make the local window. Paper [1-3] did not implement dynamic image partitioning to handle large-size input image.

Dynamic image partitioning was performed in [4] that divided images recursively and in [5] that detect windows using integral projection. Paper [4] has a rather high complexity because the computation of the decision is done recursively, particularly in processing images that have uneven levels of lighting such as images captured by cameras. In [5], integral projection is used to divide the whole image into small windows that contain just 1 character each window. The advantage of this method is that it can detect the size of various character but, however, the division of the image into local window characters results in the high computational cost, while to do a good binarization as in [2][3] do not require image division into characters.

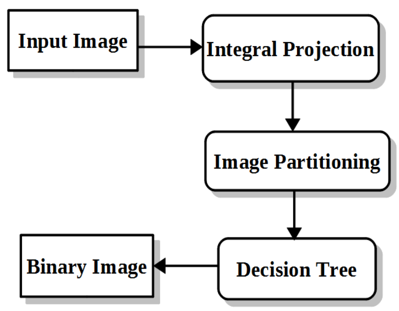
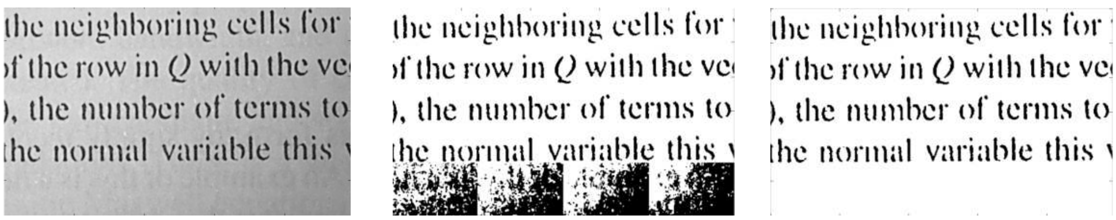


Figure 1. Block diagram of the proposed method.

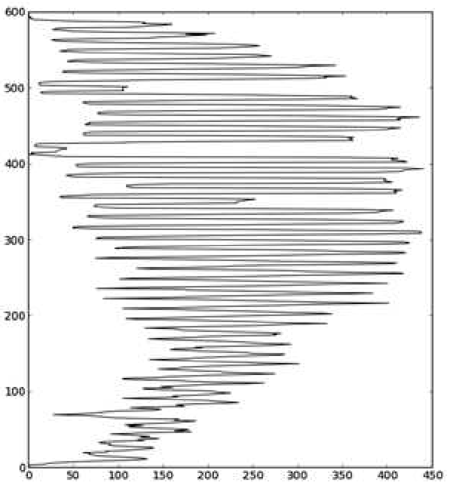


(a) (b) (c)

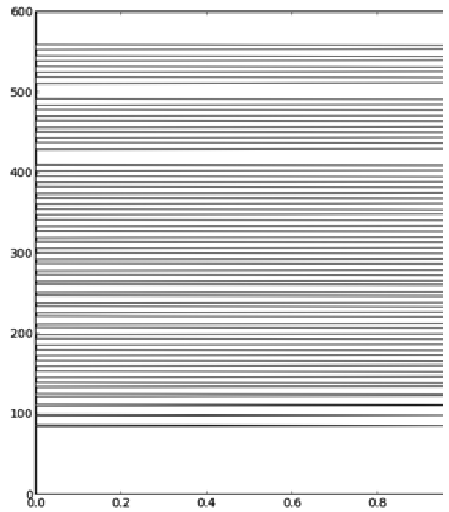
Figure 2. (a) The input image, (b) The binarized image using Otsu's method applied to local windows, and (c) The binarized image using Otsu's method applied to local windows with bottom-row local windows threshold value set to 0.

(a) (b) (c)

Figure 3. (a) Edge detection, (b) Otsu binarization, and (c) Morphology close.



(a)



(b)

Figure 4. Integral projection (a) Before normalization and (b) After normalization.

Binarization of document image needs a dynamic partitioning to make a proper size of local window. Chou algorithms [2][3] shows a good local window binarization result in a document image captured in a close distance and contains just a few words. But it cannot give good result on a full page document image. The method in [3] cannot be used properly because local window size can be different from the SVM training phase to the testing phase. In this paper, we propose a local binarization method with dynamic image partitioning using integral image and decision tree for the binarization decision.

The reminder of this paper is organized as follows. In section 2, we present the steps of the method including the integral image and decision tree. Section 3 contains the result and analysis of the method. Finally, section 4 concludes the paper.

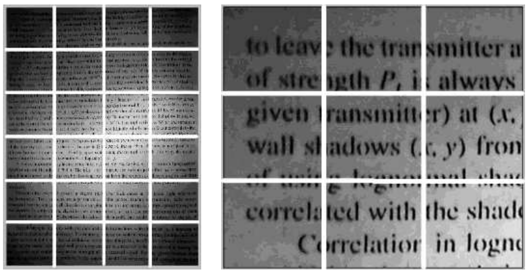
2. Methodology

We use Local Binarization Decision Tree in this experiment. The algorithm that is used to binarize the document image are consist of two general steps. The first step is estimating the number of lines in the document image. The number of line in the document image is used as a parameter to divide the whole document image into local windows. The second step is calculating the threshold value in every local window with decision tree. Every local window from the first step is divided once more into 3×3 local windows. The decision tree uses properties from the local windows to decide the threshold value. Figure 1 shows the block diagram of the proposed method.

First is number of line estimation. First, the image is resized. We set the height to a value and the width is calculated using the input height and keep the ratio of height-width from original image. This resized image is used only for this estimation step. After the input image is resized, then horizontal Prewitt is used to detect the edge of the document. The edge image is then binarized using Otsu's method. The edge detection process is performed to make the binarization result better. The edge detection gives good line but each line in the original image is represented as two lines in the binary image because the edge detection gets the top and bottom of each line. To delete the gap between the two lines, morphological close process is applied to the binary image.

After the morphological close process, the binary image will have a good line representation. The next step is to generate the integral image. Pixel values in each row are summarized and then normalized with the half value between the maximum and minimum value of the integral image as the threshold. The sum which is lower than the threshold is set to 0 and which is higher than the threshold is set to 1. From the normalization process, we get the clear integral image which represents the line number. The line number is counted from the number of 1 value sequence in integral image.

Second is decision tree based local thresholding. Decision tree is a technique that is used in data mining to classify a set of data based on specific criteria. It consists of a phase those results in a representation of the sequential decision and the possible outcomes.



(a) (b)

Figure 5. (a) Local window and (b) Local window is divided to smaller local windows.

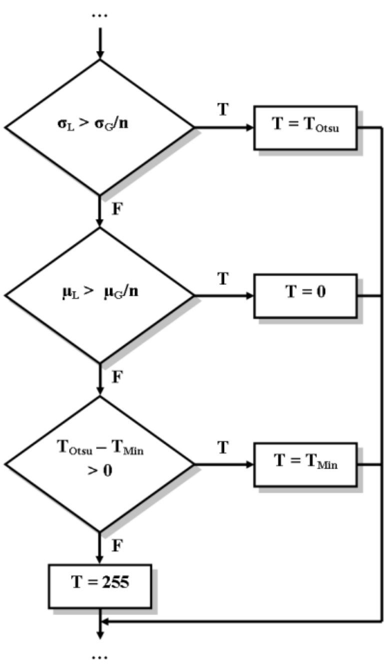


Figure 6. Decision tree rule.

The decision tree for local thresholding is adopted from decision cascade in [2]. After partitioning an image into windows, several windows may not contain any text (i.e contains background pixels). Thus, applying Otsu's method will result in a poor binary image, as shown in figure 2(b). Therefore, it is necessary to check whether or not the window contains some text by examining the standard deviation of the window. The standard deviation represents the distribution of the gray values across the window, capable of representing the presence of text in the window. If the standard deviation is high, it can be concluded that the window contains text, and therefore applying Otsu method will yield a good separation between the text and the background. But if the standard deviation is low, we can tell that it contains only background pixels.

For dark-background windows, the threshold value must be set to 255, making it completely black. To determine if the background is the dark or the bright one, the average pixel value μ is examined. Dark-background windows are denoted by the low value of the μ of that window and 255 should be applied as the threshold value of that window. But if that μ value is high (i.e bright-background windows), the Otsu threshold value of the neighboring windows must be examined. The minimum of the Otsu threshold value of the neighboring windows TMin is the most suitable value to be applied to the window if the Otsu threshold of the window TOtsu value subtracted by TMin is less than TOtsu. Otherwise, 0 is applied as the threshold value to the window to make the entire window white. The resulting binary image is shown in figure 2(c).

3. Results and Analysis

In this research, we use ten images that are captured perpendicularly using a 3.2-megapixel mobile phone camera in portrait position. Each image contains the entire area of 1 page text (A4 or B5 size paper) document. Images are taken in a normal lighting condition with camera flash activated.

The preprocessing step gives the clear line representation as shown in figure 3. The next step is to calculate the sum of each row in the image array of figure 3(c) to form the integral image. The integral image is normalized to make the integral image to contain just 0 and 1 value. After the integral image is normalized, the next step is to count the 1 value sequence in the integral image. This sequence gives the estimation of line numbers. Table I shows the estimation of line number of 10 sample images and figure 4 shows the integral image.

The number of lines in the page is used to divide the page. This method uses the number of lines per local window as the parameter. The document image height is divided with the result of dividing number of lines in page with number of lines per local window to get the local window height. Local window width is calculated with the aspect ratio of the document image but with reversed value to get landscape ratio. The local window result is shown in figure 5(a). Every local window is divided once more into 3×3 local windows for the decision tree process. The image properties (standard deviation, mean, and the Otsu threshold value) in each 3×3 local windows are compared with their global area properties in the decision tree. Figure 5(b) shows the example of

3×3 local windows. Figure 6 shows the rule of how the decision tree works.

To test the method, the result image is read by OCR software. Figure 7 shows the example of OCR result. The percentage of recognized character is calculated by comparing the resulting text with the text from the original image as the ground truth. The OCR recognition rate is the percentage of the number of true recognized characters divided by the number of all the characters in the document image. Table I shows the recognition rate of the OCR in processing ten input image samples.

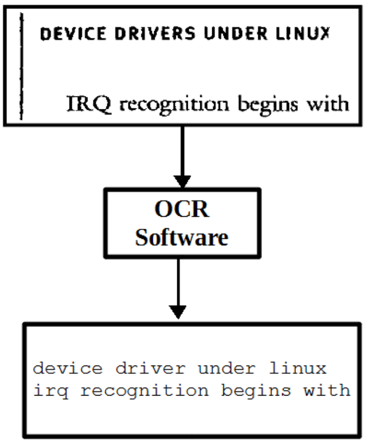


Figure 7. Optical character recognition (OCR) using OCR software.

TABLE I
NUMBER OF LINE ESTIMATION RESULTS

Sample No.	Original Number of line	Line Number Estimation	Number of Chars	Recognized Chars (%)
1	29	27	1980	**99.4**
2	40	35	2628	99.1
3	27	35	1827	96.6
4	44	32	2193	94.5
5	52	11	3100	93.2
6	27	25	1782	99.2
7	35	29	2315	97.5
8	29	**29**	1972	96.0
9	18	15	717	**99.4**
10	48	15	2694	93.3

The line number estimation give not good result if many lines are covered more than a half in dark lighting. The preprocessing successfully makes an improvement but still does not make a good result. Without the normalization process, the line estimation is far from the real line number. The Morphological close process is used with 5×5 window kernel and it successfully close the gap between top and bottom of a line.

In the normalization process of the integral projection, a line that have length less than half of the difference of maximum and minimum line is categorized as space and set to 0. This makes a line in top and bottom of the page are not recognized. Table I shows the result of number of line estimation of 10 sample image. The best result is achieved in estimating the line number of 8th sample. That image was captured without flash light and therefore the light was almost evenly spread. There is a shadow in the left area but it only covers about 1/4 of the page.

Another image that was captured with no flash light is 10th sample. Although, it has evenly spread light, but it contains many line with small space between each line. It makes the estimation of line number is not good. This also occurred in 5th sample.

The next step is partition the image into local window. The estimation line number result is divided with the number of line that is desired in one local window. The desired number of local window is set to 5. The result is used as the height of the local window and the width is calculated using image ratio.

The local window is divided once more to 3×3 as shown in figure 5(b). The size of the local window is adopted from [3]. Every local window properties is compared with their global window to decide the threshold value. Because the size of the local windows is smaller, the properties of the global window are divided by 2. Based on experiment, this value is suitable for most input images.

The document images are recognized with *Tesseract* OCR software. Of all the input images, the best results are achieved in image 1 and 9 with 99.4% of accurate character recognition. The number of lines in both images was estimated close to the original number of lines.

Figure 8 shows the result of this local threshoding method compared with the global thresholding method using Otsu's algorithm. Figure 8(a) contains big space and it can be binarized clearly with the proposed method. Figure 8(d) shows the image that has full text page and figure 8(g) has a block of black line. All of these images can be binarized clearly with this method.

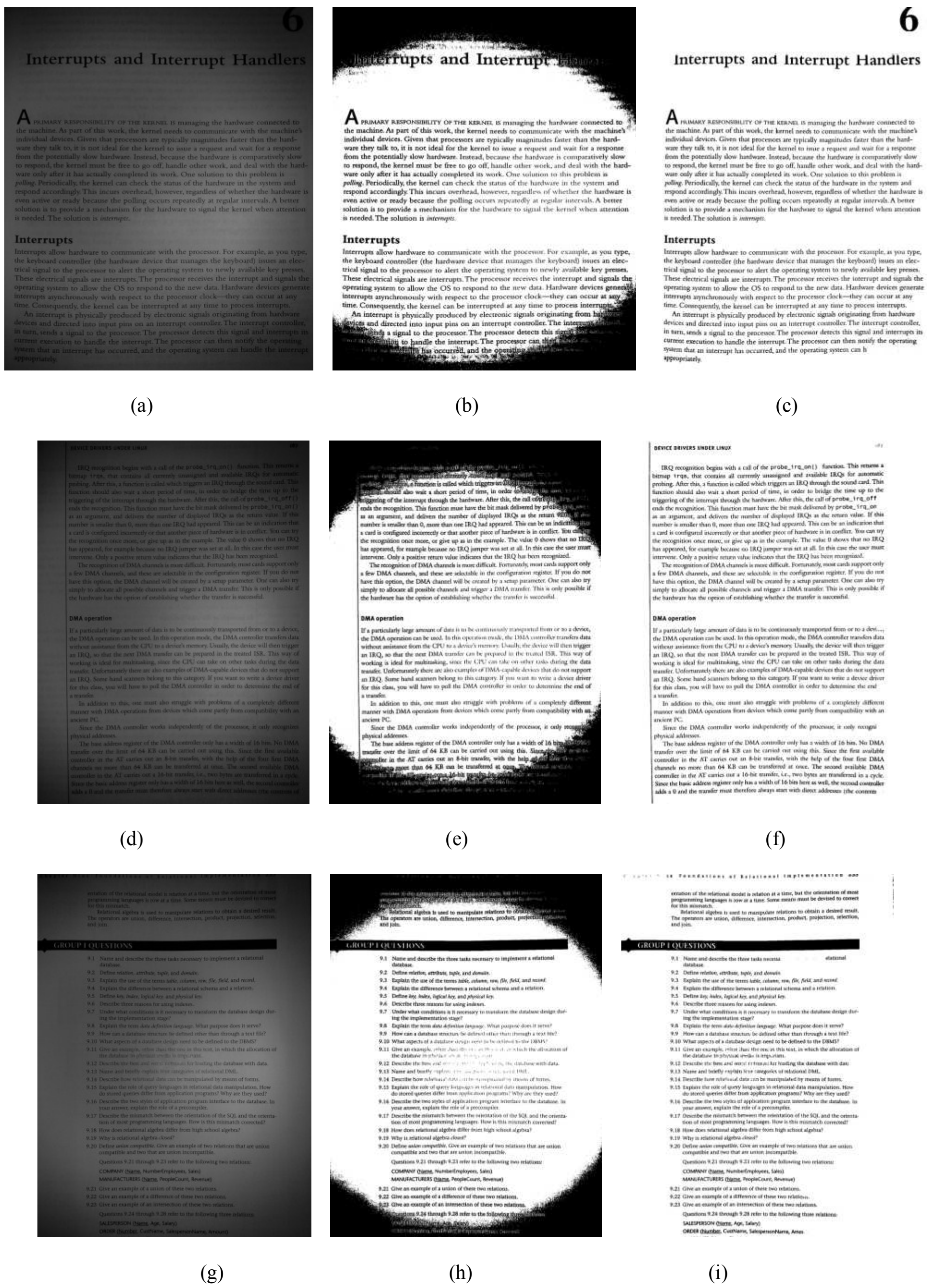


Figure 8. Input image: (a), (d), and (g); Otsu's method applied as global threshold: (b), (e), and (h); Our method (c), (h),and (i).

4. Conclusion

The proposed method can separate the text from the background form the image that has uneven lighting. The success of estimating the number of line is determinating the later process. The better result of line number estimation, the better binarization result achieved. A better line number estimation result makes good size of local window and therefore good local binarization result.

References

[1] J.H. Park, I.H. Jang, & N.C. Kim, "Skew correction of business card images in PDA" *In Proceeding of IEEE Pacific Rim Conference on Communication, Computers and Signal Processing 2003*, pp. 724-727. 2003.

[2] C.H. Chou, C.C. Huang, W.H. Lin, & F. Chang, "Learning to Binarize Document Images Using A Decision Cascade" *In Proceeding of IEEE International*

Conference on Image Processing 2005, pp. 518-521, 2005.

[3] C.H. Chou, W.H. Lin, & F. Chang, "A Binarization Method with Learning-Built Rules for Document Images Produced by Cameras," *Pattern Recognition*, vol. 43. pp. 1518-1530, 2010.

[4] J. Shi, N. Ray, & H. Zhang, "Shape Based Local Thresholding for Binarization of Document Images," *Pattern Recognition Letters*, vol. 33, pp. 24-32, 2012.

[5] Y.T. Pai, Y.F. Chang, & S.J. Ruan, "Adaptive Thresholding Algorithm: Efficient Computation Technique Based on Intelligent Block Detection for Degraded Document Images," *Pattern Recognition*, vol. 43, pp. 369-377, 2010.