# IMPROVING THE IMAGE BINARIZATION PROCESS BY USING A WEIGHTED-VOTING SCHEME

*Cristian-Daniel POBLEANU*[1]
*Andrei DUMITRESCU*[2]
*Ștefania CRISTEA*[3]
*Giorgiana Violeta VLĂSCEANU*[4]

**Abstract:** *An essential step in the OCR (Optical Character Recognition) flow is the stage of image binarization. There are considerable algorithms used for the stage of binarization. The significant difference between them is the method that computes the pixel threshold value. These algorithms can categorize into two groups: those that require a single threshold value for the whole image (which can recognize larger areas or objects) and those that use a threshold obtained from the pixel's local space (pixels in the neighborhood of the given pixel and that can identify smaller clusters of pixels). The objective of this paper is to present a voting-based image binarization algorithm that makes use of results from both categories of algorithms previously mentioned.*

**Keywords:** *image binarization, thresholding, voting-based processing, Otsu, Kittler, Niblack, Sauvola*

## 1. Introduction

The binarization method proposes a separation of planes - background and foreground, in order to select the relevant data in the image to investigate it. The first step is the conversion of the input image to a grayscale image, and then the proposed thresholding algorithms split the output image into two types of pixels: foreground pixels and background pixels [1].

The proposed method makes use of the qualities of multiple thresholding algorithms and combines them to achieve a better result.

### 1.1. Previous work

Image thresholding methods group into two major categories: global and local [2].

[1] Engineer, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, pobleanu.cristian@yahoo.com
[2] Engineer, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, andrei.dumitrescu95@stud.acs.upb.ro
[3] Engineer, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, stefania.cristea1708@stud.acs.upb.ro
[4] Teaching Assistant, PhD Student, Eng., Computer Science and Engineering Department, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, giorgiana. vlasceanu@cs.pub.ro

**Global thresholding methods** compute a statistic on the whole image and obtain a threshold value for every pixel. Such methods are high-speed but work best on (near) ideal image conditions (smooth background, uniform lighting, low noise), and most images are not ideal. Thus, global methods will mostly have bad results.

**Local thresholding methods** take into consideration the neighboring pixels when computing the threshold value. These methods overcome the problems presented by global methods by calculating pixel threshold values by probing information from neighboring pixels. They also achieve good results on non-ideal images, but they deal with noise more than global methods do. Thus, objects and pixel clusters can be erroneously identified.

Voting based image binarization algorithms try to combine the two previously presented approaches by using them in image areas where they work best. One of the first attempts to combine them is found in [3].

The most basic method of voting based image binarization presented in [4] is the one which democratically chooses the value of a specific pixel, based on the results of all the chosen thresholding algorithms. If most algorithms decided that a pixel should be of value 0, then the output image will contain this pixel with value 0 or value 1 otherwise. This method can be further developed by adding weights to specific algorithms before probing the pixel values. This way, the solution will have a more personalized result.

One solution presented in [5] uses a window of $N*M$ pixels and analyzes the number of objects found within that specific window. If the number is below a certain previously chosen threshold, then it is optimal to apply a local algorithm. Otherwise, they choose a global algorithm. They also save a matrix (equal to the dimension of the image) of threshold values. If the area analyzed by the window falls in the local category, then they increment that threshold. Otherwise, they decrement it. After the window moves through the entire image input, local and global algorithms apply depending on the values from the computed threshold matrix.

In [6] is presented a voting technique applied to a data set that contains only text-based documents that combine the following algorithms: Niblack, Nick, Sauvola, Wolf, and Otsu. Their approach describes a system that generates candidates for each algorithm. Based on tournaments, validation tests, and voting-based procedures, the viable candidate is selected from the pool generated.

Another approach found in academia also uses a window, but this time the window is iteratively expanding. This process stops when the standard deviation inside the window is constant. In this case, no new information given anymore. Afterward, a global thresholding method utilizes the newly generated window.

This voting-based approach is present in other fields of Image Processing and Computer Vision. For example, these procedures returned precise results in areas such as Image Segmentation [7], Layout Analysis [8] and OCR Systems [9].

## 1.2. Problem motivation

The goal of this article is to present a method that makes use of the knowledge from previously mentioned works but tries a slightly different approach to window placement. This way, the proposed method should take less time to compute, while obtaining good results overall.

## 2. Proposed method

As mentioned beforehand, the images targeted are document images that contain handwriting (Figure 1.a). After choosing the input image, this one is converting to grayscale using an algorithm called Color to Grey [10] (Figure 1.b).

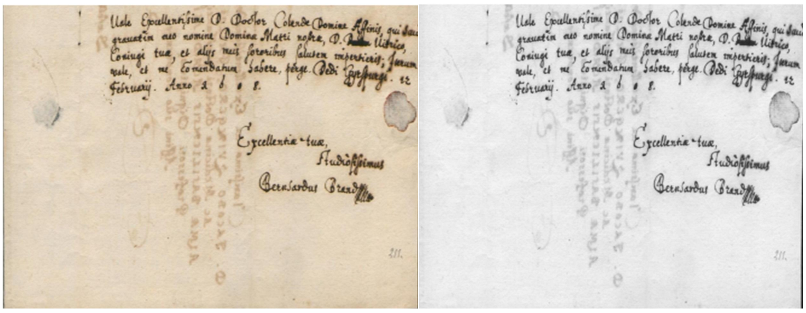

Figure 1. From left to right: a. Original document image; b. Grey scaled image

The proposed method uses both global and local thresholding methods to solve the problem.

The method relies on results from the following algorithms: ***Global Binarization***: Otsu, Kittler; ***Local Binarization***: Niblack, Sauvola.

To understand the principle of the proposed method, in the next section, are presented the algorithms of the chosen thresholding method.

### OTSU

Two classes of pixels are involved: pixels in the area of the foreground and pixels in the background. The foreground represents the layer where the objects are. The algorithm Otsu iteratively seeks for the threshold value that decreases the similarity within these classes. For the two previously described groups, the scope is to compute the weighted sum of the standard deviation [11]:

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \quad (1)$$

This method works ideally when the input image histogram has a bimodal distribution, like all global thresholding algorithms.

Otsu is proven to obtain the best results among all global thresholding algorithms (Figure 2.a), as it can handle the background noise, and identifies text correctly.

**KITTLER**

This method uses Gaussian distribution to find the threshold value. It tries to approximate the histogram as a bimodal distribution and finds the cutoff point to segment the image into either foreground or background [12].

Moreover, it obtains great results on bimodal distributions but has trouble when the difference between foreground and background pixels is too small (Figure 2.b).

**NIBLACK**

The threshold value is calculated by this method using a local window. In this case, the estimation of the threshold is computing using the local mean and standard deviation for the pixel values in a local window defined to an image [13]. The formula used to estimate the value is:

$$T_{Niblack} = m + k * s \tag{2}$$

where:

- $m$ - mean of the local window of pixels,
- $s$ - standard deviation of the local pixel window,
- $k$ - a fixed value.

In this case, $k$ has a value of 0.2.

This method always correctly identifies text regions from image documents as foreground (Figure 2.c). However, it also tends to create a lot of binarization noise in areas that do not contain text.

To summarize, Niblack is a suitable candidate for the proposed binarization algorithm, knowing that our input images consist of old document type images that contain text.

**SAUVOLA**

Sauvola is a particularized version of the previous method presented - Niblack. In some specific conditions such as variation of light in the processed document or texture [14], Sauvola might have better results than Niblack. The value of the threshold results using the equation:

$$T_{Sauvola} = m * (1 - k * \left(1 - \frac{S}{R}\right) \tag{3}$$

where:

- $m$ - mean of the pixels under the window region
- $S$ - the dynamic range of variance
- $k$ - fixed value.

The value of $k$ is chosen 0.2, like for Niblack. This method computes images with lower noise than Niblack, basically solving the problem with non-text areas. Although it also produces wrong results on images with low contrast (Figure 2.d).

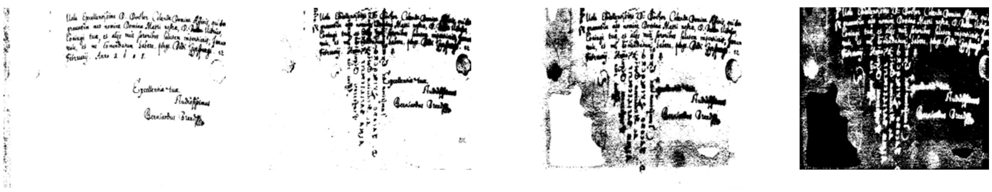

Figure 2. From left to right: a. Otsu Thresholding, b. Kittler Thresholding; c. Niblack Thresholding; d. Sauvola Thresholding

## THE VOTING METHOD

This section introduces the method used to construct the final image, starting from the presented thresholding algorithm candidates.

The first step is to apply all four algorithms presented previously onto the grayscale image. As a result, four images are created (each one for the methods Otsu, Kittler, Niblack, Sauvola). For the second step, the four images were combined into one by using majority voting as follows: each pixel in the resulting image will be black if most of the methods have the corresponding pixel with that same value. Otherwise, the color will be white, meaning that the pixel will be part of the background.

The results obtained were close to the ground-truth, but since the technique was used with equal weights set to all thresholding algorithms, there was more noise in the output than desired. Where global algorithms performed well, the local algorithms would add noise or vice-versa.

To eliminate it and obtain even better results, individual weights were assigned to each binarization algorithm. The values were selected as follows: the global algorithms received a higher value as they performed better on the data set, while local ones received a smaller number.

The weight did not exceed the value 2, which represents the total number of methods – 2 [4].

## 3. Performance measurements

In order to have significant results in section performance measurements, the proposed method was applied to three separate image type documents, which contain text and have variable light and contrast conditions.

## MEASUREMENTS

To measure how efficient a method is compared to another; the pixels were split into three categories [15]:

- *TP* – true positive; the pixels which have the same value are part of objects in both the resulting image and the ground-truth
- *FP* – false positive; the pixels which are part of an object in the resulted image only

- *FN* – false negative; the pixels which are part of an object in the ground-truth only

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + \ Precision}$$

For results closer to the ground-truth, the value of the F-measure is higher.

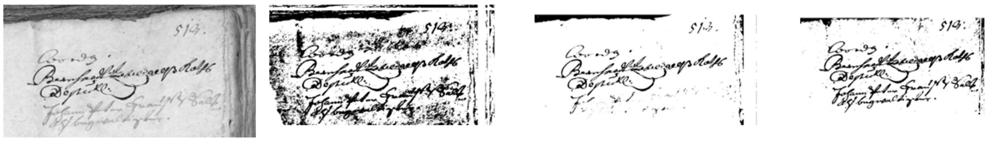

Figure 3. Image1 – From left to right: a. ColorToGray; b. Niblack; c. Otsu; d. Kittler

For the first image (Figure 3.a), the proposed method produces a very good result (Figure 4) compared to the other existing techniques (Figure 3.b, 3.c., 3.d.). The only problem seems to be the optimal block size, which is generated for the Niblack algorithm (it may be too high), which seems to generate a bit of noise in the writing sections.

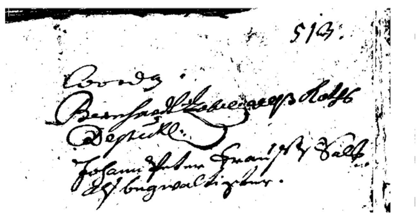

Figure 4. Image1 – Proposed algorithm

The main problem with this image is the difference in the writing's opacity. The first, third, and fourth lines are less dark than the second. In the Otsu algorithm, the last two are faded almost completely. In this aspect, our method identified all rows correctly and had less noise in the output. The differences between the algorithms results can be found in the table below (Table 1).

| Algorithm | Recall | Precision | F-measure |
|---|---|---|---|
| Otsu | 52.30 | 56.96 | 54.53 |
| Kittler | 63.12 | 65.60 | 64.34 |
| Niblack | 64.02 | 25.93 | 36.91 |
| Proposed | 94.81 | 62.41 | 75.27 |

***Table 1 – Results Image 1***

For the second image (Figure 5.a), the situation is similar to the previous one, as the proposed method (Figure 6) manages to binarize the image correctly

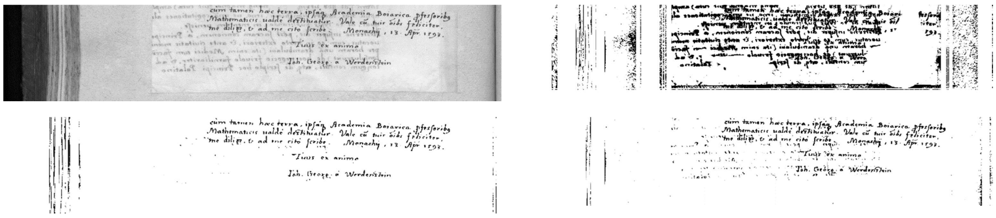

Figure 5. Image2 – From left to right, up to bottom: a. ColorToGray; b. Niblack; c. Otsu; d. Kittler

The proposed algorithm generates a similar result to the Otsu and Kittler binarizations (Figure 5.c and 5.d) but does not generate binarization noise like the Niblack thresholding algorithm (Figure 5.b). The main problem of this case was to eliminate the writing from the page above.

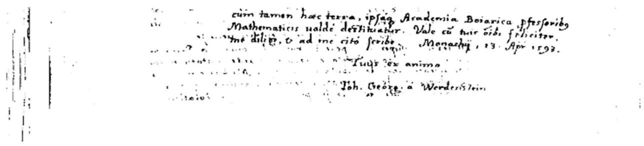

Figure 6. Image 2 – Proposed algorithm

The results obtained for the second image are presented in Table 2.

| Algorithm | Recall | Precision | F-measure |
|---|---|---|---|
| Otsu | 46.87 | 70.17 | 56.20 |
| Kittler | 46.58 | 70.52 | 56.10 |
| Niblack | 44.74 | 17.31 | 24.96 |
| Proposed | 74.68 | 74.68 | 74.68 |

***Table 2 – Results Image 2***

The third image (Figure 7.a) also obtains very good results in comparison to the other algorithms and does not segment any characters or essential data. The damage created by the paper fold is significantly minimized compared to Niblack and Otsu algorithms.

| Algorithm | Recall | Precision | F-measure |
|---|---|---|---|
| Otsu | 87.82 | 69.63 | 77.68 |
| Kittler | 70.40 | 88.36 | 78.37 |
| Niblack | 71.51 | 35.80 | 47.71 |
| Proposed | 92.61 | 85.60 | 88.97 |

***Table 3 – Results Image 3***

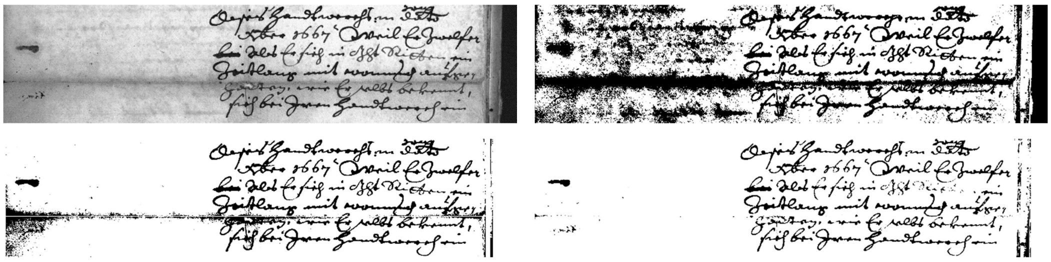

Figure 7. Image3 – From left to right, up to bottom: a. ColorToGray; b. Niblack; c. Otsu; d. Kittler

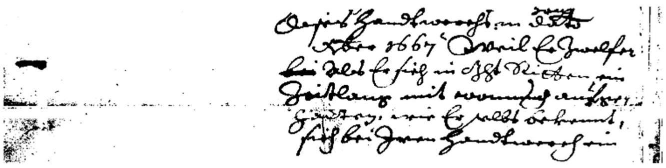

Figure 8. Image3 – Proposed algorithm

## 4. Conclusion

After analyzing and comparing the resulting images from multiple algorithms and the proposed one, the conclusion that rises is that the results are at least satisfactory. By voting between the use of the local or global algorithms, it manages to eliminate the binarization noise generally created by global algorithms when they have to deal with lots of objects or variable contrast and by local algorithms when they deal with areas with very few objects. The proposed algorithm manages to deliver a clear output image, which can be later used in the OCR process.

## Acknowledgement

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI - UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0689 / „Lib2Life- Revitalizarea bibliotecilor și a patrimoniului cultural prin tehnologii avansate” / "Revitalizing Libraries and Cultural Heritage through Advanced Technologies", within PNCDI III

## References

[1] Nabendu Chaki, Soharab Hossain Shaikh, Khalid Saeed, “Exploring Image Binarization Techniques”, Springer, New Delhi, India, 2014, DOI: 10.1007/978-81-322-1907-1.

[2] Reinhard R. Beichel, Ph.D. Medical Computer Vision & Graphics Lab – The University of Iowa, Course 55:148 Digital Image Processing, Chapter 5, Part

I, Segmentation: Thresholding, 2000, Accessed on 15 November 2019, URL: http://user.engineering.uiowa.edu/~dip/lecture/Segmentation1.html.

[3] Costin-Anton Boiangiu, Alexandra Olteanu, Alexandru Victor Stefanescu, Daniel Rosner, Alexandru Ionut Egner, "Local Thresholding Image Binarization using Variable-Window Standard Deviation Response" (2010), Annals of DAAAM for 2010, Proceedings of the 21st International DAAAM Symposium, 20-23 October 2010, Zadar, Croatia, pp. 133-134, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria 2010.

[4] Alexandru Pruncu, Cezar Ghimbas, Radu Boeru, Vlad Neculae, Costin-Anton Boiangiu; "Majority voting image binarization", The Journal of Information Systems & Operations Management, Vol.12 No.2 - 2018, pp. 422-430.

[5] Costin-Anton Boiangiu, Mihai Simion, Vlad Lionte, Zaharescu Mihai – "Voting Based Image Binarization", The Journal of Information Systems & Operations Management Vol. 8, No. 2 / December 2014 (Journal of Information Systems, Operations Management), pp. 343-351.

[6] Costin-Anton Boiangiu, Giorgiana Violeta Vlăsceanu, Alexandru Marian Atanasiu; Petrisor Alin Damian, Cristian Panaitescu, "A voting approach for image binarization of text-based documents" UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science, Vol. 81, No. 3, 2019, pp. 53-64.

[7] Costin-Anton Boiangiu, Radu Ioanitescu, "Voting-Based Image Segmentation, The Journal of Information Systems & Operations Management Vol. 7 No. 2 / December 2013 (Journal of Information Systems, Operations Management), pp. 211-220.

[8] Costin-Anton Boiangiu, Paul Boglis, Georgiana Simion, Radu Ioanitescu, "Voting- Based Layout Analysis", The Journal of Information Systems & Operations Management Vol. 8 No. 1 / June 2014 (Journal of Information Systems, Operations Management), pp. 39-47.

[9] Costin-Anton Boiangiu, Radu Ioanitescu, Razvan-Costin Dragomir, "Voting-Based OCR System", The Journal of Information Systems & Operations Management, Vol. 10 No. 2 / December 2016 (Journal of Information Systems, Operations Management), pp 470-486.

[10] Gooch, A. A., Olsen, S. C., Tumblin, J., & Gooch, B., "Color2Gray: Salience-preserving color removal", ACM Transactions on Graphics, 24(3), 634-639, 2005 https://doi.org/10.1145/1073204.1073241.

[11] H.J. Vala, Astha Baxi, "A review on Otsu image segmentation algorithm". International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 2 (2), 2013.

[12] Puneet and Naresh Garg, "Binarization Techniques used for Grey Scale Images", International Journal of Computer Applications 71(1):8-11, June 2013, DOI: 10.5120/12320-8533.

[13] S. Farid and F. Ahmed, "Application of Niblack's method on images", 2009 International Conference on Emerging Technologies, Islamabad, 2009, pp. 280-286.

[14] J. Sauvola, M Pietikainen, “Adaptive document image binarization, Pattern Recognition”, Volume 33, Issue 2, DOI: 10.1016/S0031-3203(99)00055-2, 2000, pp. 225-236.

[15] Soharab Hossain Shaikh, Asis Umar Maiti, Nabendu Chaki, “A new image binarization method using iterative partitioning”, Machine Vision and Applications(24), No. 2, pp. 337-350 February 2013.