# How to Choose a Binarization Algorithm for a Document Image?

Rafael Dueire Lins
*UFRPE/UFPE*
Recife, PE, Brazil
rdl@cin.ufpe.br

Gustavo P. Chaves
*UEA*
Manaus, AM, Brazil
gpssc.eng20@uea.edu.br

Thaylor Vieira
*UEA*
Manaus, AM, Brazil
tvm.pds24@uea.edu.br

Ricardo da Silva Barboza
*UEA*
Manaus, AM, Brazil
rsbarboza@uea.edu.br

***Abstract*—This document is a tutorial on how to choose a binarization algorithm for a document image, considering different analysis criterias. This tutorial focuses on some main challenges one can face when dealing with binarization algorithms.**

***Index Terms*—component, formatting, style, styling, insert**

## I. Introduction

The operation of converting a grayscale picture into a binary picture is crucial in various tasks related to image processing and computer vision. As early as in 1979 Nobuyuki Otsu had stressed that "in picture processing binarization is important to select an adequate threshold of gray level for extracting objects" [4]. This technique simplifies the image by reducing it to two colors, typically black and white, which enhances the contrast between the foreground and background. Binarization is important for various tasks: OCR, object detection, edge detection—all of them require clear distinction of objects from the background. By throwing away inefficient color information, binarization makes the computational and storage complexity of analysis and processing much lower—helping engineers work on speed and accuracy in critical applications.

Ongoing research is committed to developing new algorithms for binarizing images. This is because the very many challenges of varying lighting conditions, noise, and complex backgrounds in images make current researchers look for more accurate and efficient binarization techniques. He improves upon his precision since the varying lighting conditions, noise, and complex backgrounds in images make binarization a bit challenging. The improvements provide the quality of binary images produced, so that useful information is conserved and irrelevant information is minimized. The other hand, by continually improving the methods of binarization, such innovations contribute to applications that are much more stable and efficient in computer vision, document analysis, and pattern recognition.

Thus, this tutorial aims to address the steps of selecting the most suitable binarization algorithm for document images, since it is important to evaluate both classical and newly developed ones, as each has unique strengths and limitations. By carefully analyzing these options, this tutorial provides guidance on making informed decisions to achieve optimal results in document image binarization.

## II. Challenges in Database Creation

Creating a comprehensive and large dataset is crucial for the success of any analysis, particularly in tasks like binarization algorithm evaluation. A large dataset provides a diverse range of samples that better represent the variability in real-world scenarios. This diversity is essential for training and testing algorithms, as it ensures that the solutions developed are robust and applicable to a wide range of conditions. Without a sufficient amount of data, the analysis may become biased, leading to inaccurate conclusions and suboptimal algorithm performance. A large dataset also facilitates more reliable statistical analysis, allowing researchers to draw meaningful insights and make data-driven decisions with greater confidence. In summary, investing time and resources in building a substantial dataset is a critical step in ensuring the accuracy and effectiveness of any image processing analysis.

Document images can be categorized as photographed or scanned, and understanding the differences between these types is important for proper document processing and analysis. Photographed documents are captured using digital cameras or smartphones, resulting in images that often vary in quality due to factors like lighting conditions, perspective distortion, and camera resolution. These images may also include background elements or shadows, making them more challenging to process. On the other hand, scanned documents are produced using scanners, which typically provide more consistent image quality, being usually flat, well-lit, and free from perspective distortion, resulting in clearer and more uniform documents. However, scanned images might still face issues such as low resolution or noise, depending on the scanner's quality and settings.

Acquiring the images for processing is one challenge; having available corresponding ground truth data is another. Ground truth is very important for carrying out accurate and meaningful analysis—especially of algorithms developed for binarization or under evaluation, being the ideal binary version of an image, with all text and pertinent features of interest perfectly identified and well separated from the background. This serves as a critical benchmark in assessing the performance of automated processes (allowing a clear comparison of algorithm outputs with what one wants to achieve).

Creating a new database of photographed documents is a process that involves complexity and at the same time meticulousness. It is necessary to acquire images using various devices and configurations to ensure comprehensive coverage. For each device, multiple photos of documents printed with laser, inkjet, and book offset printers were taken. Additionally, the lighting conditions were also varied, including artificial, natural, and low-light environments. The backgrounds were also diversified, featuring both white and black surfaces. Finally, different flash settings—on, off, and auto—were tested. This extensive approach is necessary to build a robust dataset that accurately represents a wide range of real-world conditions. Related to the scanned documents, there is a large amount of data at the (https://dib.cin.ufpe.br/) that can be used for researches, including the ground-trouth for each image.

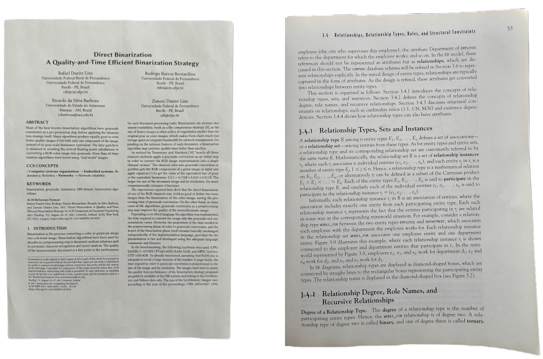


Fig. 1. Photographed Dataset Examples.

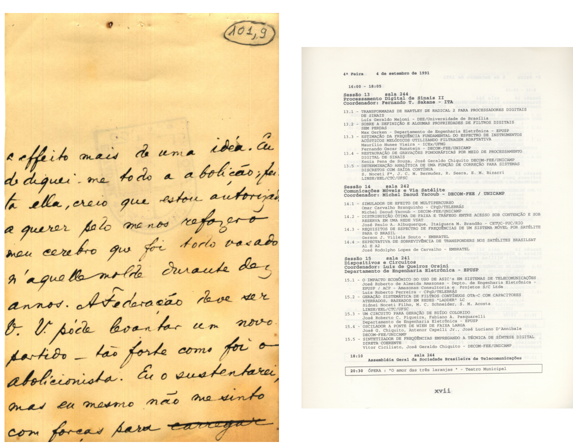


Fig. 2. Scanned Dataset Examples.

## A. Preprocessing

Preprocessing images before binarization is another crucial step in image processing, particularly when the goal is to convert an image into a binary format. This step is important because it enhances the quality and accuracy of the binarization process, which can significantly affect subsequent analysis tasks. One way to do so is to cropping images, that involves selecting and extracting a specific portion of an image while discarding the rest, and so highlighting the most important elements from it. This process can be done manually, but is also possible by using online softwares that automate this task.

For improved analysis, it is also possible to extract each color channel (red, green, and blue) from an image and generate three new images, each containing only one of the color components: R (red), G (green), and B (blue), while the value on the others are reseted. This way, new results can be obtained when binarizing these images. New ways to include different versions from the same image is generating new ones with only the biggest or smallest components from each pixel, resulting into a grey scale document. At least, it is also viable to do the most common approach: use a weighted combination of the primary color values (red, green, and blue), taking into account the sensitivity of the human eye to these colors. The formula to calculate the intensity in grayscale from a colored image is:

$$\text{Gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

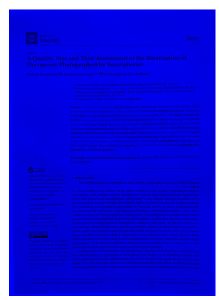


Fig. 3. Image with only the blue RGB component

Fig. 4. Image with only the green component

Fig. 5. Image with only the red component

# III. BINARIZATION

As presented, there are numerous algorithms available for binarizing images, each with its own unique methods and characteristics, as demonstrated in recent DocEng competitions [1] and [2]. These algorithms are often developed using a variety of programming languages and dependencies, thus, preparing the environments for each one is a complicated task, but an important one to allow its testing and usage. All binarization algorithms that are recommended for use and testing are: ACM [10], AugraphyD [1], AugraphyF [1], Akbari_1 [11], Akbari_2 [11], Akbari_3 [11], Bataineh [12], Bernsen [13], Bhowmik [14], Bradley [15], Calvo-Zaragoza [16], dSLR [17], DeepOtsu (SL) [18], DE-GAN [19], DiegoPavan (DP) [19], DilatedUNet [8], DocBin1 [1], DocBin2 [1], DocBin3 [1], DocDLinkNet [20], DocUNet (WX) [21], DocEnTR [1], ElisaTV [22], Ergina-Global [23], Ergina-Local [24], FourBi [1], Gattal [25], Gosh [26], HandwritteNet [21], Howe [27], Huang ($AH_1$) [28], Huali [21] iNICK [29], Intermodes [44], ISauvola [31], IsoData [32], Jia-Shi [33], Johannsen-Bille [59], Kapur-SW [35], Li-Tam [36], LS-HDIB [1], Lu-Su [37], Mean [38], Mello-Lins [39], Michalak [40], MO1 (Michalak-Okarma) [41] MO2 (Michalak-Okarma) [40] MO3 (Michalak-Okarma)

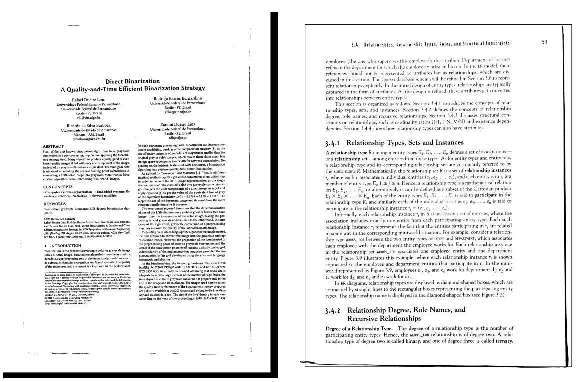
[42] MinError [43], Minimum [44], Moments [45], Niblack [46], Nick [47], Otsu [4], Percentile [48], Pun [49], Robin [1], Sauvola [50], Shanbhag [51], Singh [52], Su-Lu [53], Texture Based [6], Tensmeyer [61] Triangle [54], Vahid [10], WAN [55], Wolf [56], Wu-Lu [57]. Yasin [21] Yen [58], YinYang [8], YinYang21 (JB) [8]. YinYang22 [60], YinYang23a [7], YinYang23b [7], YinYang23c [7], Yuleny [21] and ZigZag [5].

They were implemented by different authors and can be found on public repositories, being some of these classical ones and some new recently published. A short description of these can be found on a table on the last page of this tutorial.

Another challenge is the computational resources necessary to execute all these tasks, since each one was tested on different architectures and so it functionality can be unpredictable, which happened in the following algorithms:

- The algorithms Akbari_1 [11], Akbari_2 [11], and Bhowmik [14] implemented for Windows for some reason were not able to run in the more recent architectures and versions of the operating system used in this competition. Thus, they were excluded from this assessment.
- DocDLinkNet [20], HandwritteNet [21], Huali [21] and Yuleny [21] were very slow and did not produce any output for some images, thus they were discarded from the global assessment.
- Gosh [26] has problems in the installation set-up and is unable to be called on a batch of images, having to be invoked to process each image one-by-one. That unstable "behavior" was also dependent on the smartphone device model, and has not been observed whenever applied to scanned document images.

Malfunctioning algorithms are part of the process when working in this kind of research.


Direct Binarization
A Quality-and-Time Efficient Binarization Strategy

3.4 Relationships, Relationship Types, Roles, and Structural Constraints

3.4.1 Relationship Types, Sets and Instances

3.4.2 Relationship Degree, Role Names, and Recursive Relationships

Fig. 6. Image binarization results examples.

## IV. Analysis

During the analysis phase, the quality of the binarized images is the most important factor to consider, and the processing time will only been taken in account if two or more algorithms presents the same quality. Although, if it doesn't happen, it will only be used primarily to provide an order of magnitude of the elapsed time, since even if an algorithm is faster than another, it won't be preferred to if its results are inferior.

The first metric to be considered while analyzing the quality of an image is called "Perr". It analyzes the proportion of black and white pixels in the binarized image, compared to the ground truth image as a reference. The ground truth is selected by choosing the best algorithm from other researches, such as [1] and [3], as a standard for the other images. As a result, the "Perr" value is obtained, and the smaller it's value is, implies a higher quality of the algorithm, since it means a minor error when comparing those two images.

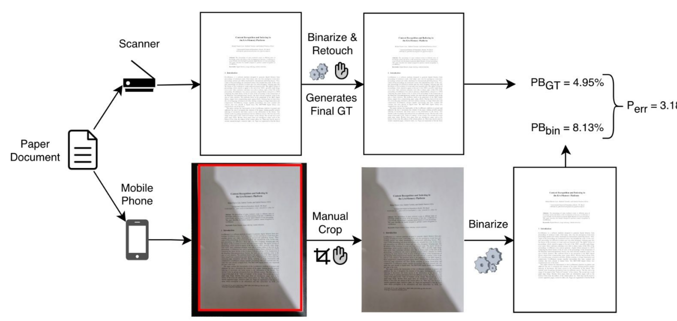



Fig. 7. Perr measure example (GT: ground-truth, bin:binary).

The second metric may be using the Google Vision to perform Optical Character Recognition (OCR). To use Google OCR, first, access the Google Cloud Console (https://console.cloud.google.com/), create a new project (or select an existing one), and complete the necessary configurations. It is worth to note that Google Cloud provides an initial credit of $300 USD for new free accounts, a value that was barely used while analyzing all the binarized images.

After perfoming OCR on the images, the Levenshtein distance of the results is calculated using the ground truth image previously generated as a reference. This metric is used to measure the difference between two sequences of characters, being defined as the minimum number of operations required to transform one sequence of character into another, where operations allowed include insertion, deletion, and/or substitution of a character. Some words may have characters that are easily confused, especially in the context of binarization and OCR. For example, after an image is binarized, during the OCR process, the character sequence "rn" might be read as "m", or "5" can be mistaken for "s", among other examples that can hinder this task. In summary, the fewer operations required to make a given character sequence match the ground truth, the lower the Levenshtein distance will be, indicating that the extracted text is similar to the original and that the algorithm is performing well. On the other hand, a high distance indicates that the algorithm needs improvements.

Another evaluation metric that can be used is the compression ratio of each algorithm. This metric is not directly related to the algorithm's quality, since algorithms with a good

[1] https://github.com/masyagin1998/robin

compression ratio are not necessarily the ones that produce the best results. To calculate it, the binarized images are first compressed from the .PNG format using the Tag Image File Format Group 4 (TIFF_G4). The reason for this choice is that the TIFF_G4 file format is possibly the most efficient lossless compression scheme for binary images [62]. Additionally, evaluating an algorithm's compression is important because a high or average ratio greatly helps to reduce the size of the output data and save storage, especially when dealing with large volumes of data being processed.

It is important to emphasize that, for each variation presented in the dataset creation challenges section, the performance results of the algorithms should be analyzed separately for these variants. Different devices, printer type, and lighting conditions can also lead to varying results from the algorithms, with one algorithm potentially outperforming another in a specific variant. The same applies to preprocessing when using RGB channels to generate new input images, when an algorithm might have outstanding performance on the red channel, but not be as effective on the green channel.

## V. Conclusions

Document image thresholding is a fundamental step in several document processing pipelines, but it is far from being a trivial task to choose a binarization algorithm that meets the quality, time, and space performance needed by different applications and devices, presenting the necessity of this tutorial.

Considering all the information presented on this tutorial, it is feasible to infer that are different criterias to analyze an algorithm and its results. Besides, it is also possible to conclude that several challenges exists when dealing with them, such as computational power and environment preparation. This tutorial plays a roll to show how this topic is still important and demands more research, and also tries to bring new interested individuals to do so.

At the Table 1 presented on the last page of this tutorial there is a short description of the algorithms mentioned, and also an analysis example of the results of the $L_{dist}$ metrics for the S24 device of a inkjet printed document with the strobe flash on.

For future works, new algorithms and methods are always appearing to improve one of the analysis metrics, and for that it is necessary to test them. Also, new ways of analyzing a binarized image can also be developed, such as a new way to generate a ground-truth as a base for the evaluations.


## Acknowledgements

The research reported in this paper was sponsored by the RDI project Callidus Academy signed between the Universidade do Estado do Amazonas (UEA) and Callidus Indústria through the Lei de Informática/SUFRAMA. Rafael Dueire Lins was also partly sponsored by CNPq - Brazil.

Also, the authors of this paper are very grateful to all those who made the code for their algorithms available, for their cooperation and good spirit.


## References


[1] R. D. Lins, G. P. de F. P. Silva, G. P. Chaves, R. da S. Barboza, R. B. Bernardino, and S. J. Simske, "Quality, Space and Time Competition on Binarizing Photographed Document Images," in *Proceedings of the ACM Symposium on Document Engineering 2023*, New York, NY, USA: Association for Computing Machinery, 2023, pp. 1-10. doi: 10.1145/3573128.3604903.

[2] R. D. Lins, G. P. de F. P. Silva, G. P. Chaves, R. da S. Barboza, T. Vieira, and S. J. Simske, "Competition on Binarizing Photographed Document Images
2024 Quality, Time and Space Report," in *Proceedings of the ACM Symposium on Document Engineering 2024*, New York, NY, USA: Association for Computing Machinery, 2023, pp. 1-10. doi: 10.1145/3573128.3604903.

[3] R. D. Lins, R. B. Bernardino, R. Barboza, and R. Oliveira, "The Winner Takes It All: Choosing the "best" Binarization Algorithm for Photographed Documents," in *Document Analysis Systems*, S. Uchida, E. Barney, and V. Eglin, Eds. Cham: Springer International Publishing, 2022, pp. 48-64.

[4] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE T. on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.

[5] J.-L. Bloechle, J. Hennebert, and C. Gisler, "ZigZag: A Robust Adaptive Approach to Non-Uniformly Illuminated Document Image Binarization," in *Proceedings of ACM Symposium on Document Engineering 2024 (DocEng '24)*, New York, NY, USA: ACM, 2024.

[6] R. Bernardino, R. D. Lins, and R. Barboza, "Texture-based Document Binarization," in *Proceedings of ACM Symposium on Document Engineering 2024 (DocEng '24)*, New York, NY, USA: ACM, 2024.

[7] J.-L. Bloechle, J. Hennebert, and C. Gisler, "YinYang, a Fast and Robust Adaptive Document Image Binarization for Optical Character Recognition," in *Proceedings of the ACM Symposium on Document Engineering 2023*, Limerick, Ireland: Association for Computing Machinery, 2023, pp. 1-4. doi: 10.1145/3573128.3609354.

[8] R. D. Lins, S. J. Simske, and R. B. Bernardino, "DocEng'2020 Time-Quality Competition on Binarizing Photographed Documents," in *DocEng '20: ACM Symposium on Document Engineering 2020*, Virtual Event, CA, USA: ACM, 2020. doi: 10.1145/3395027.3419578. Available: https://doi.org/10.1145/3395027.3419578.

[9] R. D. Lins, R. Bernardino, and D. M. Jesus, "A Quality and Time Assessment of Binarization Algorithms," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 1444-1450. doi: 10.1109/ICDAR.2019.00232.

[10] R. D. Lins, R. B. Bernardino, and S. J. Simske, "DocEng'2021 Time-Quality Competition on Binarizing Photographed Documents," in *DocEng'2021*, New York, NY, USA: ACM, 2021, pp. 1-4. doi: 10.1145/3395027.3419578.

[11] Y. Akbari and others, "Binarization of Degraded Document Images using Convolutional Neural Networks based on predicted Two-Channel Images," in *ICDAR*, 2019.

[12] B. Bataineh and others, "An adaptive local bin. method for doc. images based on a novel thresh. method and dynamic windows," *Pattern Recognition Letters*, vol. 32, no. 14, pp. 1821-1831, 2011. doi: 10.1016/j.patrec.2011.06.016.

[13] J. Bernsen, "Dynamic thresholding of gray-level images," in *International Conference on Pattern Recognition*, 1986, pp. 1251-1255.

[14] S. Bhowmik, R. Sarkar, B. Das, and D. Doermann, "GiB: a Game theory Inspired Binarization technique for degraded document images," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1443-1455, 2018.

[15] D. Bradley and G. Roth, "Adaptive Thresholding using the Integral Image," *Journal of Graphics Tools*, vol. 12, no. 2, pp. 13-21, 2007.

[16] J. Calvo-Zaragoza and A.-J. Gallego, "A selectional auto-encoder approach for document image binarization," *Pattern Recognition*, vol. 86, pp. 37-47, 2019. doi: 10.1016/j.patcog.2018.08.002.

[17] J. M. M. Silva, R. D. Lins, and V. C. Rocha, "Binarizing and Filtering Historical Documents with Back-to-Front Interference," in *ACM SAC 2006*, 2006, pp. 853-858. doi: 10.1145/1141277.1141471.

[18] S. He and L. Schomaker, "DeepOtsu: Document Enhancement and Binarization using Iterative Deep Learning," *Pattern Recognition*, vol. 91, pp. 379-390, Jan. 2019.

[19] M. A. Souibgui and Y. Kessentini, "DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement, vol. 44, no. 3, pp. 1180-1191, 2022. doi: 10.1109/TPAMI.2020.3022406.

[20] L. Zhou and others, "D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction," in *IEEE CS CCVPR Workshops*, 2018.

[21] R. D. Lins, E. Kavallieratou, E. Barney Smith, R. B. Bernardino, and D. M. Jesus, "ICDAR 2019 Time-Quality Binarization Competition," in *ICDAR*, 2019, pp. 1539-1546. doi: 10.1109/ICDAR.2019.00248.

[22] E. H. Barney Smith, L. Likforman-Sulem, and J. Darbon, "Effect of Pre-processing on Binarization," in *Document Recognition and Retrieval XVII*, 2010, pp. 75340H.

[23] E. Kavallieratou, "A binarization algorithm specialized on document images and photos," *ICDAR*, vol. 2005, no. 1, pp. 463-467, 2005.

[24] E. Kavallieratou and S. Stathis, "Adaptive binarization of historical document images," in *Proceedings - International Conference on Pattern Recognition*, vol. 3, 200

[25] A. Gattal, F. Abbas, and M. R. Laouar, "Automatic Parameter Tuning of K-Means Algorithm for Document Binarization," in *7th ICSENT*, ACM Press, 2018, pp. 1-4. *Annote: kmeansBin — Gattal*.

[26] S. K. Bera and others, "A non-parametric binarization method based on ensemble of clustering algorithms," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 7653-7673, 2021.

[27] N. R. Howe, "Document binarization with automatic parameter tuning," *IJDAR*, vol. 16, 2013.

[28] L. K. Huang and M. J. Wang, "Image thresholding by minimizing the measures of fuzziness," *Pattern Recognition*, vol. 28, no. 1, pp. 41-51, 1995.

[29] K. Saddami, K. Munadi, S. Muchallil, and F. Arnia, "Improved Thresholding Method for Enhancing Jawi Binarization Performance," in *ICDAR*, vol. 1, IEEE, 2017. *Annote: iNICK — Saddami 1*.

[30] J. M. S. Prewitt and M. L. Mendelsohn, "The Analysis of Cell Images," *Annals of the New York Academy of Sciences*, vol. 128, no. 3, pp. 1035-1053, 2006. *Annote: Intermodes - original Minimum - versão modificada*.

[31] Z. Hadjadj et al., "ISauvola: Improved Sauvola's Algorithm for Document Image Binarization," in *Lecture Notes in Computer Science*, vol. 3212, A. Campilho and M. Kamel, Eds. Springer Berlin Heidelberg, 2016.

[32] F. R. Velasco, "Thresholding Using the Isodata Clustering Algorithm," Tech. Rep., OSD or Non-Service DoD Agency, 1979.

[33] F. Jia, C. Shi, K. He, C. Wang, and B. Xiao, "Degraded document image binarization using structural symmetry of strokes," *Pattern Recognition*, vol. 74, pp. 225-240, 2018.

[34] G. Johannsen and J. Bille, "A threshold selection method using information measures," in *International Conference on Pattern Recognition*, 1982, pp. 140-143.

[35] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 1, pp. 140, 1985.

[36] C. H. Li and P. K. S. Tam, "An iterative algorithm for minimum cross entropy thresholding," *Pattern Recognition Letters*, vol. 19, no. 8, pp. 771-776, 1998.

[37] S. Lu, B. Su, and C. L. Tan, "Document image binarization using background estimation and stroke edges," *IJDAR*, vol. 13, no. 4, pp. 303-314, 2010.

[38] C. Glasbey, "An Analysis of Histogram-Based Thresholding Algorithms," *Graphical Models and Image Processing*, vol. 55, no. 6, pp. 532-537, 1993.

[39] C. A. B. Mello and R. D. Lins, "Image segmentation of historical documents," *Visual 2000*, 2000. *Annote: Mello-Lins binarization*.

[40] H. Michalak and K. Okarma, "Fast Binarization of Unevenly Illuminated Document Images Based on Background Estimation for Optical Character Recognition Purposes," *Journal of Universal Computer Science*, vol. 25, no. 6, pp. 627-646, 2019.

[41] H. Michalak and K. Okarma, "Adaptive image binarization based on multi-layered stack of regions," in *International Conference on Computer Analysis of Images and Patterns*, Springer, 2019, pp. 281-293.

[42] H. Michalak and K. Okarma, "Improvement of image binarization methods using image preprocessing with local entropy filtering for alphanumerical character recognition purposes," *Entropy*, vol. 21, no. 6, p. 562, 2019. MDPI.

[43] J. Kittler et al., "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, 1986. *Annote: MinError algorithm*.

[44] J. M. S. Prewitt and M. L. Mendelsohn, "The Analysis of Cell Images," *Annals of the New York Academy of Sciences*, vol. 128, no. 3, pp. 1035-1053, 2006. *Annote: Intermodes - original Minimum - versão modificada*.

[45] W.-H. Tsai, "Moment-preserving thresholding: A new approach," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 377-393, 1985.

[46] W. Niblack, *An introduction to digital image processing*, Strandberg, 1985.

[47] K. Khurshid, I. Siddiqi, C. Faure, and N. Vincent, "Comparison of Niblack inspired binarization methods for ancient documents," in *SPIE*, 2009, pp. 72470U.

[48] W. Doyle, "Operations Useful for Similarity-Invariant Pattern Recognition," *Journal of the ACM*, vol. 9, no. 2, pp. 259-267, 1962. *Annote: Percentile*.

[49] T. Pun, "Entropic thresholding, a new approach," *Computer Graphics and Image Processing*, vol. 16, no. 3, pp. 210-239, 1981.

[50] J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen, "Adaptive document binarization," in *ICDAR*, IEEE Comput. Soc, 1997, pp. 147-152.

[51] A. G. Shanbhag, "Utilization of Information Measure as a Means of Image Thresholding," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 5, pp. 414-419, 1994.

[52] T. Romen Singh, S. Roy, O. I. Singh, et al., "A New Local Adaptive Thresholding Technique in Binarization," *IJCSI*, vol. 08, no. 6, 2011.

[53] B. Su, S. Lu, and C. L. Tan, "Robust Document Image Binarization Technique for Degraded Document Images," *IEEE Transactions on Image Processing*, vol. 22, no. 4, pp. 1408-1417, 2013.

[54] G. W. Zack, W. E. Rogers, and S. A. Latt, "Automatic measurement of sister chromatid exchange frequency," *Journal of Histochemistry and Cytochemistry*, vol. 25, no. 7, pp. 741-753, 1977.

[55] W. A. Mustafa and A. M. Kader, "Binarization of Document Image Using Optimum Threshold Modification," *Journal of Physics: Conference Series*, vol. 1019, no. 1, 2018.

[56] C. Wolf and D. Doermann, "Binarization of low quality text using a Markov random field model," in *Object recognition supported by user interaction for service robots*, IEEE Comput. Soc, 2002, pp. 160-163.

[57] W. Lu, S. Ma, and H. Lu, "An effective entropic thresholding for ultrasonic images," in *14th ICPR*, vol. 2, pp. 1552-1554, 1998.

[58] J. C. Yen, F. J. Chang, S. Chang, J. C. Yen, F. J. Chang, and S. Chang, "A New Criterion for Automatic Multilevel Thresholding," *IEEE Transactions on Image Processing*, vol. 4, no. 3, pp. 370-378, 1995.

[59] G. Johannsen and J. Bille, "A threshold selection method using information measures," in *International Conference on Pattern Recognition*, 1982, pp. 140-143.

[60] R. D. Lins, R. B. Bernardino, R. Barboza, and S. J. Simske, "DocEng'2022 Quality, Space, and Time Competition on Binarizing Photographed Documents," in *DocEng'22*, ACM, 2022, pp. 1-4.

[61] C. Tensmeyer and T. Martinez, "Document Image Binarization with Fully Convolutional Neural Networks," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, Nov. 2017, pp. 99-104. doi: 10.1109/ICDAR.2017.25.

[62] Lins R.D., Avila B.T. A New Algorithm for Skew Detection in Images of Documents. Int. Conf. Image Anal. Recognit. 2004;3212:234–240. doi: 10.1007/978-3-540-30126-4_29.

TABLE I

BINARIZATION ALGORITHMS SHORT DESCRIPTION

| **Method** | **Category** | **Description** | $\mathbf{L}_{dist}$ |
|---|---|---|---|
| DiegoPavan [19] | Deep Learning | Downscale image to feed a DE-GAN network | 1,46 |
| dSLR [17] | Global threshold | Uses Shannon entropy to find a global threshold | 5,37 |
| DeepOtsu [18] | Deep Learning | Neural networks learn degradations and global Otsu generates binarization map | 7,71 |
| Niblack [46] | Local threshold | Based on window mean and the standard deviation | 8,08 |
| Percentile [48] | Global threshold | Based on partial sums of the histogram levels | 8,08 |
| Pun [49] | Global threshold | Defines an anisotropy coefficient related to the asymmetry of the histogram | 8,46 |
| YinYang [8] | Image Processing | Detect background , extract it and apply Otsu | 8,76 |
| Robin | Edge based | U-net model trained with several datasets | 8,78 |
| Michalak [40] | Image Processing | Downsample image to remove low-frequency information and apply Otsu | 8,87 |
| DocUNet [21] | Deep Learning | Hybrid U-Net network with bottom-hat enhanced document images | 8,92 |
| ZigZag [5] | Image Processing | Improvement of YinYang23, having five different processing modes. | 8,94 |
| Bernsen [13] | Local threshold | Uses local image contrast to choose threshold | 8,97 |
| Tennsmeyer [61] | Deep Learning | Makes use of a fully convolutional neural network | 8,97 |
| Michalak21$_a$ [41] | Image Processing | Downsample image to remove low-frequency information and apply Otsu | 9,00 |
| Vahid [10] | Deep Learning | A pixel-wise segmentation model based on Resnet50-Unet | 9,00 |
| Akbari_3 [11] | Deep Learning | Variation of Akibari_1 where fewer channels are used | 9,01 |
| Calvo-Zaragoza [16] | Deep learning | Fully convolutional Encoder–decoder FCN with residual blocks | 9,02 |
| Gattal [25] | Clustering | Automatic Parameter Tuning of K-Means Algorithm | 9,02 |
| ElisaTV [22] | Local threshold | Background estimation and subtraction | 9,02 |
| YinYang23a,b,c [7] | Image Processing | Fasters and more efficient versions of YinYang22 algorithm | 9,02 |
| YinYang21 [8] | Image Processing | A faster and more effective version of YinYang algorithm | 9,02 |
| YinYang22 [60] | Image Processing | Uses maximum color occurrence to detect and subtract background, then apply Otsu | 9,03 |
| WAN [55] | Global threshold | Improves Sauvola's method by shifting up the threshold | 9,03 |
| Huang [28] | Global threshold | Minimizes the measures of fuzzines | 9,33 |
| Mello-Lins [39] | Global threshold | Global threshold determined by shannon entropy. | 9,42 |
| MinError [43] | Global threshold | Minimum error threshold | 9,60 |
| Mean [38] | Global threshold | Mean of the grayscale levels | 9,63 |
| Li-Tam [36] | Global threshold | Minimum cross entropy | 9,68 |
| Ergina-Local [24] | Local threshold | Detects where to apply local thresholding after a applying a global one | 9,68 |
| Bataineh [12] | Local threshold | Based on local and global statistics | 9,70 |
| Yen [58] | Global threshold | Multilevel threshold based on maximum correlation criterion | 9,71 |
| DE-GAN [19] | Deep Learning | Uses a conditional generative adversarial network | 9,72 |
| Shanbhag [51] | Global threshold | Improves Kapur-SW by viewing the two pixel classes as fuzzy sets | 9,74 |
| Moments [45] | Global threshold | Aims to preserve the moment of the input picture | 9,79 |
| DilatedUNet [8] | Deep Learning | Downsample to smooth image and uses a dilated convolutional layer | 9,80 |
| Su-Lu [53] | Edge based | Canny edges using local contrast | 9,80 |
| Johannsen-Bille [59] | Global threshold | Minimizes formula based on the image entropy | 9,81 |
| Intermodes [44] | Global threshold | Smooth histogram until only two local maxima | 9,81 |
| Bradley [15] | Local threshold | Adaptive thresholding using the integral image of the input | 9,81 |
| Sauvola [50] | Local threshold | Improvement on Niblack | 9,82 |
| Wu-Lu [57] | Global threshold | Minimizes the difference between the entropy of the object and the background | 9,83 |
| IsoData [32] | Global threshold | IsoData clulstering algorithm applied to image histogram | 9,85 |
| Minimum [44] | Global threshold | Variation of Intermodes algorithm | 9,85 |
| Kapur-SW [35] | Global threshold | Maximizes formula based on the image entropy | 9,85 |
| Singh [52] | Global threshold | Uses integral sum image prior to local mean calculation | 9,85 |
| Triangle [54] | Global threshold | Based on most and least frequent gray level | 9,86 |
| Wolf [56] | Local threshold | Improvement on Sauvola with global normalization | 9,86 |
| Otsu [4] | Global threshold | Maximize between-cluster variance of pixel intensity | 9,86 |
| ISauvola [31] | Local threshold | Uses image contrast in combination with Sauvola's binarization | 9,86 |
| Nick [47] | Local threshold | Adapts Niblack based on global mean | 9,87 |