



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

RODRIGO BARROS BERNARDINO

Assessing Binarization Algorithms for Document Images

Recife

2025

RODRIGO BARROS BERNARDINO

Assessing Binarization Algorithms for Document Images

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Mídia e Interação

Orientador: Prof. Dr. Rafael Dueire Lins

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Bernardino, Rodrigo Barros.

Assessing binarization algorithms for document images /
Rodrigo Barros Bernardino. - Recife, 2024.
125 f.: il.

Tese (Doutorado) - Universidade Federal de Pernambuco, Centro
de Informática, Pós-Graduação em Ciência da Computação, 2024.
Orientação: Rafael Dueire Lins.
Inclui referências.

1. Algoritmos de binarização; 2. Documentos históricos; 3.
Documentos escaneados; 4. Documentos fotografados; 5.
Smartphones; 6. Avaliação de desempenho. I. Lins, Rafael Dueire.
II. Título.

UFPE-Biblioteca Central

Rodrigo Barros Bernardino

“Assessing Binarization Algorithms for Document Images”

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação. Área de Concentração: Mídia e Interação

Aprovada em: 09/09/2024.

Orientador: Prof. Dr. Rafael Dueire Lins

BANCA EXAMINADORA

Prof. Dr. Silvio de Barros Melo
Centro de Informática/UFPE

Prof. Dr. Cleber Zanchettin
Centro de Informática/UFPE

Prof. Dr. Steve John Simske
Systems Engineering / Colorado State University

Prof. Dr. Valdemar Cardoso da Rocha Junior
Departamento de Eletrônica e Sistemas / UFPE

Prof.Dr. Gabriel de França Pereira e Silva
Unidade Acadêmica do Cabo de Santo Agostinho / UFRPE

Dedico a todos aqueles que lutam para encontrar sua própria voz. Que não se deixem levar pelas aparências ou opiniões e saibam escolher seu próprio caminho e percorrê-lo com toda a força.

ACKNOWLEDGEMENTS

I would like to begin by expressing my gratitude to my father, Salvador Ramos (in memoriam), who did everything within and beyond his power to support me in all my major life decisions, without ever directly interfering in them, but instead pushing me forward with love and strength. Countless were his messages, phone calls, words of encouragement, and, above all, his examples of nobility, courage, and kindness. After his passing, he seemed to become even closer, now as an internal reference of peace e positiveness that was fundamental for completing this academic journey.

To my mother, Suely Barros, who has nurtured me emotionally, spiritually, and even physically from my birth to the present day. Her affection and unconditional support have shaped me into a man who strives to be strong and fight to the end, yet without ever losing sight of the other's perspective, and always striving to be good and just. With her support, I not only managed to complete my doctorate but also found true love and happiness.

To my wife, Marília Dália, my lady and inspiration. By her side, I found what I was missing in life. I will be forever grateful for her patience, temperance, and authentic love.

I also extend my thanks to my beloved brother Hugo, for his example of integrity and perseverance.

To my friends and family members who, whenever possible, shared positive words of love and encouragement.

To my first mentor in "computational arts", Prof. Jucimar Jr., for his many technical teachings, but above all, for the life lessons that led me to Japan, then to Recife, and finally to the completion of this journey.

Finally, to my advisor, Prof. Rafael Dueire Lins, who spared no effort to, quite literally, push me forward and ensure I never took a step back. There were many years of intense discussions, laughter, and learning. I will carry his example of dedication, professionalism, and commitment as a guide whenever I face challenges. Thank you for being this steadfast mountain that has supported my journey for so many years.

ABSTRACT

Binarization algorithms are essential for document processing, analysis, compression, and recognition, with their performance heavily influenced by document characteristics such as paper texture and noise. This thesis introduces novel algorithms and evaluation methodologies for assessing binarization performance, focusing on image quality, processing time, and file size. Nearly 70 binarization schemes were tested on 39 historical documents and 376 mobile-captured images. To expand the analysis, the Direct Binarization approach was proposed, analysing the RGB channels of input images separately. This generated hundreds of additional images, which were used to train an automatic binarization algorithm selection tool, the Image Matcher, based solely on paper texture and the strength of the back-to-front interference. The tool demonstrated significant improvements in binarization results across various cases. Recognizing the growing prevalence of smartphone-captured documents, the thesis also investigated such type of documents by proposing and extensively testing three new evaluation measures: the proportion of black pixels in the binary image, a normalized Levenshtein distance, and a combined metric incorporating both. These measures facilitated a comprehensive assessment of mobile-captured images using six widely used mobile devices under varying conditions, including strobe flash settings, illumination, and positional changes. Additionally, the compressed image size (using the TIFF Group 4 compression scheme) proved to be a valuable metric for evaluating the algorithms efficiency. It has been shown that if processing time is a priority, the Michalak21a algorithm with the red channel would be preferred for this type of image, but if compression rate is a priority, Yinyang22 is a better choice. Choosing the best algorithm for a given setup using the PL measure provided a better choice when compared to using only the OCR accuracy. The thesis also significantly expanded existing datasets for document image binarization by adding 24 new historical document images with manually generated ground truth and 296 mobile-captured images.

Keywords: Binarization algorithms; historical documents; scanned documents; photographed documents; smartphones; performance evaluation

RESUMO

Os algoritmos de binarização são essenciais para o processamento, análise, compressão e reconhecimento de documentos, sendo seu desempenho fortemente influenciado por características do documento, como textura do papel e ruído. Esta tese apresenta algoritmos e metodologias de avaliação inovadoras para analisar o desempenho de binarização, com foco na qualidade da imagem, tempo de processamento e tamanho do arquivo. Cerca de 70 esquemas de binarização foram testados em 39 documentos históricos e 376 imagens capturadas por dispositivos móveis. Para expandir a análise, foi proposta a abordagem Binarização Direta, que analisa separadamente os canais RGB das imagens de entrada. Isso gerou centenas de imagens adicionais, utilizadas para treinar uma ferramenta automática de seleção de algoritmos de binarização, chamada Image Matcher, baseada exclusivamente na textura do papel e na intensidade da interferência frente-verso. A ferramenta demonstrou melhorias significativas nos resultados de binarização em diversos casos. Reconhecendo a crescente prevalência de documentos capturados por smartphones, a tese também investigou esse tipo de documento, propondo e testando extensivamente três novas medidas de avaliação: a proporção de pixels pretos na imagem binária, uma distância de Levenshtein normalizada e uma métrica combinada que incorpora ambas. Essas medidas possibilitaram uma avaliação abrangente de imagens capturadas por dispositivos móveis, utilizando seis dispositivos amplamente usados em condições variadas, incluindo configurações de flash, iluminação e mudanças de posição. Além disso, o tamanho da imagem comprimida (usando o esquema de compressão TIFF Group 4) provou ser uma métrica valiosa para avaliar a eficiência dos algoritmos. Demonstrou-se que, se o tempo de processamento for uma prioridade, o algoritmo Michalak²¹a com o canal vermelho é preferível para esse tipo de imagem, enquanto, se a taxa de compressão for o foco, o algoritmo Yinyang²² apresenta melhores resultados. A escolha do melhor algoritmo para uma configuração específica usando a métrica PL mostrou-se superior em comparação ao uso exclusivo da acurácia do OCR. A tese também expandiu significativamente os conjuntos de dados existentes para binarização de imagens de documentos, adicionando 24 novas imagens de documentos históricos com ground truth gerado manualmente e 296 novas imagens capturadas por dispositivos móveis.

Palavras-chaves: Algoritmos de binarização; documentos históricos; documentos escaneados; documentos fotografados; smartphones; avaliação de desempenho

LIST OF FIGURES

Figure 1 – Example of binarization using grayscale hue thresholding	24
Figure 2 – Nabuco Light Handwritten Example With strong back-to-front interference	47
Figure 3 – Nabuco Dark Handwritten and Mid Typewritten Example Images	47
Figure 5 – Livememory Example Image	47
Figure 6 – DIB Mobile sample images clustered by device (Samsung Note 10+, Samsung S21) and set-up of the strobe flash “off”.	49
Figure 7 – DIB Mobile sample images clustered by device (Samsung Note 10+, Samsung S21) and set-up of the strobe flash bottom-line “on”.	50
Figure 8 – PRImA dataset example images	51
Figure 9 – DIBCO example images	52
Figure 4 – The full Nabuco dataset with pixel-level ground-truth.	53
Figure 10 – DIBCO Dataset Example Images (Small)	66
Figure 11 – DIB image matcher.	67
Figure 12 – Direct binarization example	74
Figure 13 – Binarization results summary	75
Figure 14 – Texture Matcher Step 1: Binarize all training images with each algorithm and rank to find the best ones.	77
Figure 15 – Texture Matcher Step 2: Compare the input image paper texture with each training image to find the most similar.	77
Figure 16 – Texture Matcher Step 3: Find the most recommended algorithm for the input image.	78
Figure 17 – Texture Matcher Step 4: Binarize with the recommended algorithm. . .	78
Figure 18 – Rank Diff: texture matching quality measure.	79
Figure 19 – Results for image matching with image HW 05 and TW 06 with grouping.	84
Figure 20 – Results for image matching with image HW 15 and TW 10 without grouping.	84
Figure 21 – Example of mobile-captured document images. Strobe flash noise (left); Strong shadow with natural light (middle); Skew due to capture angle (right).	86
Figure 22 – P_{err} measure example (GT: ground-truth, bin: binary).	90

Figure 23 – Comparison between different measures: PL , $[L_{dist}]$, P_{err} . For each case, the full image is shown on the top and an example region bellow, where the red boxes indicates the crop position for the example region. (a) Original image; (b) Ranking by P_{err} only, DiegoPavan-C binarized image; (c) Ranking by $[L_{dist}]$ only, dSLR-C binarized image; (d) Ranking by PL measure, Yasin-R binarized image.	92
Figure 24 – Example of ranking by the quality-time criteria	95
Figure 25 – Example of sorting by the ranking summation criterion	95
Figure 26 – Dataset 1 example images. (a) Samsung Note 10+, book offset page, strong natural light, flash <i>off</i> with strong shadow, binarized by HuangUNet-B; (b) Samsung S21, laser printed, artificial light, medium shadow, flash <i>off</i> , binarized by Wolf-R; (c) Same as (b), but with flash <i>on</i> and binarized by YinYang22-R.	104
Figure 27 – Dataset 2 example images. (a) iPhone SE 2, book offset page, artificial light, flash <i>off</i> with medium shadow; (b) Samsung S20, deskjet printed, artificial light, medium shadow, flash <i>off</i> ; (c) Same as (b), but with flash <i>on</i> , note that on deskjet printed pages no flash reflex interfere on the photo	105

LIST OF TABLES

Table 1 – Tested binarization algorithms	40
Table 1 – <i>Cont.</i>	41
Table 1 – <i>Cont.</i>	42
Table 2 – Nabuco 39-dataset images dimentions in pixels.	48
Table 3 – Summary of device camera specifications	49
Table 4 – Quality-time Results for Nabuco, Light Texture, Handwritten Documents . .	63
Table 5 – Quality-time Results for Nabuco, Dark Texture, Handwritten Documents . .	64
Table 6 – Quality-time Results for Nabuco, Mid Texture, Typewritten Documents . . .	64
Table 7 – Quality-time Results for LiveMemory Test Set	65
Table 8 – Quality-time Results for PRLmA Data Set	65
Table 9 – Results of binarizing DIBCO dataset	66
Table 10 – Texture features used in this study	70
Table 11 – Example of Score for a descriptor and distance combination.	80
Table 12 – Assessment of the combination of feature and distance measure either sep- arating in groups or not	81
Table 13 – Texture Matching Considering Image Features – with Groups	83
Table 14 – Texture Matching for Best three Features without grouping	85
Table 15 – Summary of device camera specifications	89
Table 16 – Example of the choice of a channel with some of the best algorithms	96
Table 17 – Mobile captured overall results by device sorted according to the ranking summation criterion.	99
Table 18 – Mobile captured summary of results - PL measure and flash OFF (quality- time criteria).	100
Table 19 – Mobile captured summary of results - PL measure and flash ON (quality-time criteria).	101
Table 20 – Mobile captured summary of results - L_{dist} measure and flash OFF (quality- time criteria).	102
Table 21 – Mobile captured summary of results - L_{dist} measure and flash ON (quality- time criteria).	103

CONTENTS

1	INTRODUCTION	13
1.1	RESEARCH QUESTIONS	17
1.2	SCIENTIFIC OUTCOMES	17
1.3	RELATED PUBLICATIONS	18
1.4	THESIS ORGANIZATION	21
2	BINARIZATION ALGORITHMS	23
2.1	CATEGORIES OF BINARIZATION ALGORITHMS	27
2.1.1	Threshold Based Binarization	27
2.1.2	Edge Detection Based Binarization	32
2.1.3	Optimization Based Binarization	33
2.1.4	Image Processing Based Binarization	34
2.1.5	Pixel Classification Based Binarization	36
3	ASSESSING DOCUMENT IMAGE BINARIZATION ALGORITHMS	43
3.1	DATASETS FOR DOCUMENT IMAGE BINARIZATION	46
3.2	CLASSICAL EVALUATION METHODS	54
3.3	NEW EVALUATION METHODS	57
3.3.1	Cohen's Kappa applied to document binarization	57
3.3.2	New Measures for Mobile-Captured Document Images	58
3.4	PROCESSING TIME EVALUATION	58
3.5	ASSESSMENT OF SCANNED DOCUMENT IMAGES	59
3.6	MATERIALS AND METHODS	60
3.7	RESULTS AND DISCUSSION	62
3.8	CONCLUSIONS	63
4	TEXTURE BASED BINARIZATION	67
4.1	TEXTURE DESCRIPTORS	68
4.2	MATERIALS AND METHODS	73
4.3	DIRECT BINARIZATION	73
4.4	BINARIZATION RESULTS	75
4.5	TEXTURE MATCHING	76

4.5.1	Matching Process	76
4.5.2	Choosing the best feature descriptor and distance	79
4.6	RESULTS AND CONCLUSIONS	81
5	NEW EVALUATION MEASURES FOR PHOTOGRAPHED DOCUMENT BINARIZATION EVALUATION	86
5.1	MATERIALS AND METHODS	88
5.1.1	The Quality Measure of the Proportion of Pixels (P_{err})	89
5.1.2	Normalized Levenshtein Distance ($[L_{dist}]$)	91
5.1.3	Pixel Proportion and Levenshtein Measure (PL)	93
5.1.4	Evaluation by Compressed Image File Size	93
5.1.5	Quality, Space and Time Evaluation	94
5.2	CHOOSING THE BEST CHANNEL	95
5.3	RESULTS	96
5.4	CONCLUSIONS	106
6	CONCLUSIONS AND FUTURE WORK	108
6.1	FUTURE WORKS	112
	REFERENCES	114

1 INTRODUCTION

The popularization of computers in the last few decades has generated a growing interest in converting paper documents into digital forms. Digital documents are not vulnerable to some of the problems of paper documents, as they need far less physical storage space, are easily copied and distributed through computer networks, keeping the same quality as the source file. Furthermore, a wide number of automatic processing strategies may be applied, ranging from transcription, language and author identification, information extraction and summarization, classification, indexing, and many others. Before applying such analyses, a legated paper document needs to be converted into a digital form. The first device developed specifically for this purpose was the flatbed scanner, which was the result of a long technological evolution dating back to 1957, when Russell Kirsch based on primitive FAX machines envisaged the possibility of capturing a document image and automatically transcribing it ¹. That was the dawn of document engineering, when the focus became on how to offer the information physically stored in the paper in a digital form, not only by transcribing the text, but also by identifying its layout and logical components [1].

Image binarization is a process of identifying regions of interest in a given image, mapping the color of the pixels into two classes: foreground (black) and background (white). Possibly the first binarization algorithm was proposed by Nobuyuki Otsu, published in 1979 [2], in biomedical images as a preprocessing step to calculate the dimensions of a baby in ultrasound images. Otsu algorithm globally analyzes the grayscale histogram of an image and returns a single intensity threshold that separates pixels into two classes, foreground and background. Since then, image binarization has become a key part of many image processing systems and has been extensively studied over the years [3, 4, 5].

Document binarization, as well as many other image processing systems, is one of the most important steps in the document processing pipeline, as many algorithms and platforms such as image OCR, de-skew, compression, and enhancement, among several others, work on binary images, including content recovery [6]. In 1983, White and Rohrer [7] used binarization as an OCR preprocessing step. At first, document image processing made use of general binarization algorithms such as Otsu [2], Niblack [8], and Bernsen [9]. The first binarization algorithm focused on documents is possibly the one by Eikvil, Taxt, and Moen from 1991, published at

¹ <https://history-computer.com/computer-scanner/>, visited on 2022/01/25

the 1st International Conference on Document Analysis and Recognition [10].

Those algorithms face many challenges, as paper documents can have some physical noises [11, 12], such as stains due to fungi, inadequate handling or storage, aging or folding marks, which degrade the quality of the document image and can cause loss of information and bring errors into the binarization process. The kind of printing (typed, offset, laser, inkjet, etc.), handwriting, kind of pen, ink, and color may also influence the quality of the final black-and-white document. One particular complication arises in the binarization of document images, the *back-to-front* interference, which appears when a document is printed or handwritten on both sides of the page and some of the *verso* information is visible in the front image [13]. The use of binarization algorithms such as Otsu in documents with back-to-front interference causes an image overlap, yielding an unreadable document. The first solution to such a problem was also proposed in [13] and consisted of scanning the document scanned on both sides, horizontally mirroring one of the images, aligning both images, and comparing the intensity of each pixel. Such an approach works fine, but has the drawback of having to align both images, which can be made extremely hard if the document has been folded as a letter. The first binarization algorithm to overcome such a difficulty, looking only at one side of the document with back-to-front interference, was [14].

William Thompson (b. 1824, d. 1907), the first Lord Kelvin, the famous British mathematician and engineer, said:

“When you can measure what you are speaking about, and express it in numbers, you know something about it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts advanced to the stage of science.”

Thus, being able to somehow quantify what one is talking about is a fundamental part of any research in any area, but it also poses significant challenges. Assessing image quality is no exception. The first attempts to evaluate such algorithms date back to 1995, when Trier [15] developed a system to recognize digits on hydrographic maps and the number of correctly identified digits was the quality measure. Following the growing interest in binarizing document images, more precise measures for textual documents were necessary and [16] presented an image quality evaluation measure, based on which Stathis, Kavallieratou and Papamarkos proposed a new evaluation measure to evaluate the quality of 30 binarization algorithms [17]. It was based on the principle of comparing the number of correctly mapped pixels compared

to a reference, *ground truth* image, an image that would be considered “perfect” under visual inspection by several people. Such a reference image generated from a real document could be used either as a reference to directly compare the performance of binarization algorithms or to generate synthetic noisy document images, which could be used for the same purposes.

Ntirogiannis and Gatos [18] proposed a new quality assessment methodology, which has been used since 2009 in the series of algorithm competitions called DIBCO – Digital Image Binarization Contest [19, 20]. The DIBCO assessment methodology has been widely used to evaluate the quality of binarization algorithms. It is based on some statistical measures that compare a high-dpi small part of a real document image with their respective ground-truth (GT) image, which is generated by either a fully manual or a semi-automated process.

Although this methodology became popular, it does not take into account the situation of processing a full document page. This is an important issue, as documents may have uneven texture such as background and stains, fungi marks, etc. may affect parts of a document. The first binarization competition and one of the first assessments to test full-size documents was organized in 2010 at ICFHR - International Conference on the Frontiers of Handwritten Recognition [21]. Six competitors had their algorithms assessed using as quality measure the geometric-mean accuracy, the square root of the product of the proportion correctly classified to the total number of black and white pixels.

DIBCO was not explicit about document scanning resolution, and that is relevant in the tuning and the choice of several binarization algorithms. Besides that, DIBCO only assessed the enrolled competitors with a very small test set of only around ten historical documents with very little variation amongst themselves. DIBCO every year pointed out one algorithm as the competition “winner”. Another fundamental characteristic not considered by the DIBCO evaluation method is the processing time. The first assessment on binarization algorithms that took into account the average processing time of real-world 200 and 300 dpi text document images was the ICDAR 2019 Time-Quality Document Binarization Competition (TQDIB) [22], which is part of this thesis. In addition, different from previous studies, that assessment also clustered the input images by its main features and evaluated the algorithms within the context of each type of document, as the assessments have shown that no binarization algorithm is good for all kinds of document and that their time performance vary widely. It compared the quality and processing time of seventeen new algorithms together with thirty classical binarization algorithms, reporting the results of the ten best algorithms in each category. They were scanned (200/300 dpi, printed, typed, handwritten) and photographed documents with

six different models of cameras embedded in mobile cell phones with the integrated strobe flash on and off.

The TQDIB'2019 was the first competition to include photographed documents, highlighting the importance that such a type of document gained in the last few years. In addition to all those paper-related issues, when the document is captured using mobile devices, several other complications arise. The resolution and illumination are uneven, there are perspective distortions and often interference from external light sources [23]. Even the in-built strobe flash may add further difficulties if activated by the user or automatically. In addition to all that, the standard file format used by smartphone cameras to save images is jpeg, which inserts jpeg noise [24]. Finally, those cameras and the capture software are usually set to take family photos, which is not always the best setup for document image capturing.

All of these challenges make the evaluation of binarization algorithms applied to photographed images even more difficult than the scanned ones. The most common type of photographed documents are photos of printed books, articles, and office documents; therefore, initial efforts to binarize such images focused on the transcription precision of primarily printed textual images [25, 26]. In 2017, reference [27] proposed a new quality measure for photographed documents that used the proportion of correctly mapped pixels compared to the scanned version of the same document.

Given the large diversity and complexity of the challenges in binarizing document images, it has been shown that there is no single binarization algorithm good enough for all types of documents [28], as already said. Thus, in order to provide the best results for a given image, it is necessary to find the most suitable binarization algorithm for each type of document (i.e. historical handwritten, printed, offset printed, laser printed, inkjet printed, etc.). There have been a few attempts to provide a framework or even automatic selection of binarization algorithms based on the features of a document and also nearly no assessment which considered the specific document characteristic. [29] presents a machine learning approach for choosing among five binarization algorithms to binarize parts of a document image. [30] proposes a way of combining several binarization algorithms to provide the monochromatic image.

The DIB platform ² has developed to address this problem by providing an accessible way to generate more than 5 million different synthetic document images and to apply the highest possible number of binarization algorithms. Its ultimate goal is, given a real image, to find the most similar synthetic image and to indicate the rank of best quality binarization

² <https://dib.cin.ufpe.br/>

algorithms, with their average processing time and the tiff-G4 size of the final image, allowing the user to find the most suitable algorithm to binarize a specific document. The DIB website (<https://dib.cin.ufpe.br>) also contains the most important binarization data sets, pointers to competitions, and attempts to gather the most important information related to the area of document binarization in one place. As a result of this thesis, the DIB platform has been expanded with several new data sets and assessment results. The DIB platform had its relevance acknowledged and was included in the International Association for Pattern Recognition (IAPR), Technical Committee Number 11 (TC11) datasets ³.

Given the large diversity of algorithms and intense publications in the area, it is fundamental to establish a proper evaluation methodology if one wants to choose an algorithm for a specific application. This thesis proposes evaluating binarization algorithms in a more concise, precise, and realistic way. The purpose of this project is to establish several new perspectives when evaluating scanned historical and photographed modern document images that may serve as a reference for future research in the area.

1.1 Research Questions

The following research questions motivated this thesis:

1. What are the main features of the document image that affect binarization algorithms?
2. What is the best binarization algorithm for a given document image?
3. Is it possible to choose the best binarization algorithm based only on analyzing the texture of the document paper?
4. Is it possible to obtain better binarization results with one single RGB channel instead of combining them into the luminance grayscale representation?

1.2 Scientific Outcomes

The aim of this thesis is to propose a new perspective on binarizing document images by developing new binarization methods, evaluation methodology, and contexts of applications for existing methods. Decades of image processing development have been evaluated not only

³ <http://www.iapr-tc11.org/>

in terms of traditional quality measures, but also new ones: processing time, input image version (luminance or one of the RGB channels) and size of the compressed binary image. The expected results of this thesis are summarized in the following points.

1. Expanding the DIB platform with new data sets, algorithms, and results.
2. Proposing a new evaluation methodology considering the specific image characteristic of the images and conducting extensive performance assessments, both for scanned and photographed documents.
3. Presenting the processing time and size of the compressed binary images as relevant measures when evaluating binarization schemes.
4. Proposing a new application for the Cohen's Kappa as a quality measure for scanned document binarization.
5. Introducing a new quality measure for photographed documents which takes into account not only the OCR transcription quality but also the overall visual quality.
6. Providing a new perspective when evaluating binarization algorithms by feeding not only the color or grayscale image as input, but also each of the RGB channel separately and studying its impact on the final quality of the binary images.
7. Proposing a binarization algorithm selection methodology based on the texture of the paper.
8. Providing new insights on the impact of diverse documents' characteristics on processing time, such as image resolution and type of noise present in the image.

1.3 Related Publications

- **2019:** ICDAR 2019 Time-Quality Binarization Competition [22].
 - **Conference Paper** at 2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR)
 - Assessed 17 new and 30 classical algorithms using historical scanned, modern scanned, synthetic and photographed document images. Introduces processing time evaluation and the use of Cohen's Kappa measure.

-
- 20 historical images from Nabuco bequest; 100 synthetic; 72 mobile captured;
 - My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis)
- **2020:** DocEng'2020 Time-Quality Competition on Binarizing Photographed Documents [31]
 - **Conference Paper** at ACM Symposium on Document Engineering, DocEng 2020
 - Assessed eight new and 41 classical and modern binarization algorithms. Focused on photographed documents, provides more detailed assessment on such kind of documents with many different setups. The normalized Levenshtein distance measure was introduced.
 - 32 mobile captured images
 - My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis)
- **2021:** Direct binarization a quality-and-time efficient binarization strategy [32]
 - **Conference Paper** at ACM Symposium on Document Engineering, DocEng 2021
 - Introduces a new perspective on binarization algorithms analysis by providing each RGB channel individually. The results show that some channels might provide equally good or even better quality than the full-color image or the usual luminance grayscale version.
 - My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis), and writing the paper.
- **2021:** ICDAR 2021 Competition on Time-Quality Document Image Binarization [33]
 - **Conference Paper** at 2021 16th IAPR International Conference on Document Analysis and Recognition (ICDAR)
 - Assessed 12 new and 49 other previously published binarization algorithms. The project focused on historical scanned images, with more images from previous datasets and a new image source (PRImA library). Having been conceded a special authorization for more pages, a more detailed evaluation with more than double amount of data has been provided along with a more detailed discussion with valuable insights.

-
- My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis)
 - **2021:** Binarisation of photographed documents image quality and processing time assessment [34]
 - **Conference Paper** at ACM Symposium on Document Engineering, DocEng 2021
 - A sequel to the previous binarization competition on the same conference on the previous year. Assessed 13 new and 50 existing algorithms. Four newer smartphones have been used and a more challenging dataset has been proposed.
 - 192 mobile captured images (four devices, two external illumination positions, two flash conditions, three types of printing)
 - My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis)
 - **2022:** Using Paper Texture for Choosing a Suitable Algorithm for Scanned Document Image Binarization [35]
 - **Journal Paper** at Journal of Imaging
 - Proposes an automatic binarization algorithm selection method using a sample of the texture of scanned historical documents as the main document feature. Sixty-three widely used algorithms, using five different versions of the input images (Direct Binarization), have been used in the experiments.
 - My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis) and writing the paper
 - **2022:** The Winner Takes It All: Choosing the “best” Binarization Algorithm for Photographed Documents [36]
 - **Conference Paper** at DAS 2022: 15th IAPR International Workshop on Document Analysis Systems
 - It is proposed a new methodology to choose the best binarization algorithm applied to binarize documents photographed using smartphone cameras. Instead of choosing in the usual way, which is by determining an overall best in terms of OCR precision only, in this paper two other criteria are considered: for printing and

distributing; for OCR applications. The time-quality best, as opposed to the usual quality-best.

- My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis)
- **2023:** A Quality, Size and Time Assessment of the Binarization of Documents Photographed by Smartphones [37]
 - **Journal Paper** at Journal of Imaging
 - This paper assesses the quality, file size and time performance of sixty-eight binarization algorithms using five different versions of the input images. It expands the discussion of the previously published binarization competitions with two new recent smartphones, a new and even more challenging dataset, a new evaluation measure combining the two previously published ones, and the compression rate of TIFF Group 4 as another novel quality measure. With a longer evaluation, new insights are presented which advance the area of binarization analysis applied to photographed images.
 - My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis) and writing the paper
- **2024:** Texture-based Document Binarization [38]
 - **Conference Paper** at ACM Symposium on Document Engineering, DocEng 2024
 - This paper extends the analysis on texture based binarization with 12 texture descriptors and three distance measures. It provides solid evidence that it is possible to choose which binarization algorithm to use based solely on the paper background texture.
 - My contributions: executing the algorithms, collecting the results, organizing them, pointing out the main conclusions (analysis) and writing the paper

1.4 Thesis Organization

This thesis is organized as follows.

Chapter 2 introduces the challenges and most important solutions of the binarization algorithms. It also includes an adapted version of the latest binarization contest on scanned document images, first published in the ICDAR 2021 proceedings.

Chapter 3 focus on presenting the history of previous assessments, the datasets used in this thesis, traditional evaluation measures and comments on the new ones introduced in this thesis.

Chapter 4 presents the novel texture-based binarization method, where excerpts of the paper texture are used to choose the best binarization algorithm for a given image. The Direct Binarization approach is also introduced, where the RGB channels are used individually as input to the binarization algorithms.

Chapter 5 contains an adapted version of the latest journal publication “A Quality, Size and Time Assessment of the Binarization of Documents Photographed by Smartphones”, which proposes the new evaluation measures proportion of black-and-white pixels (P_{err}), normalized Levenshtein distance ($[L_{dist}]$), PL (combination of P_{err} and $[L_{dist}]$) and the evaluation by file size (CR_{G4}).

Chapter 6 presents some final considerations and appointments for future work.

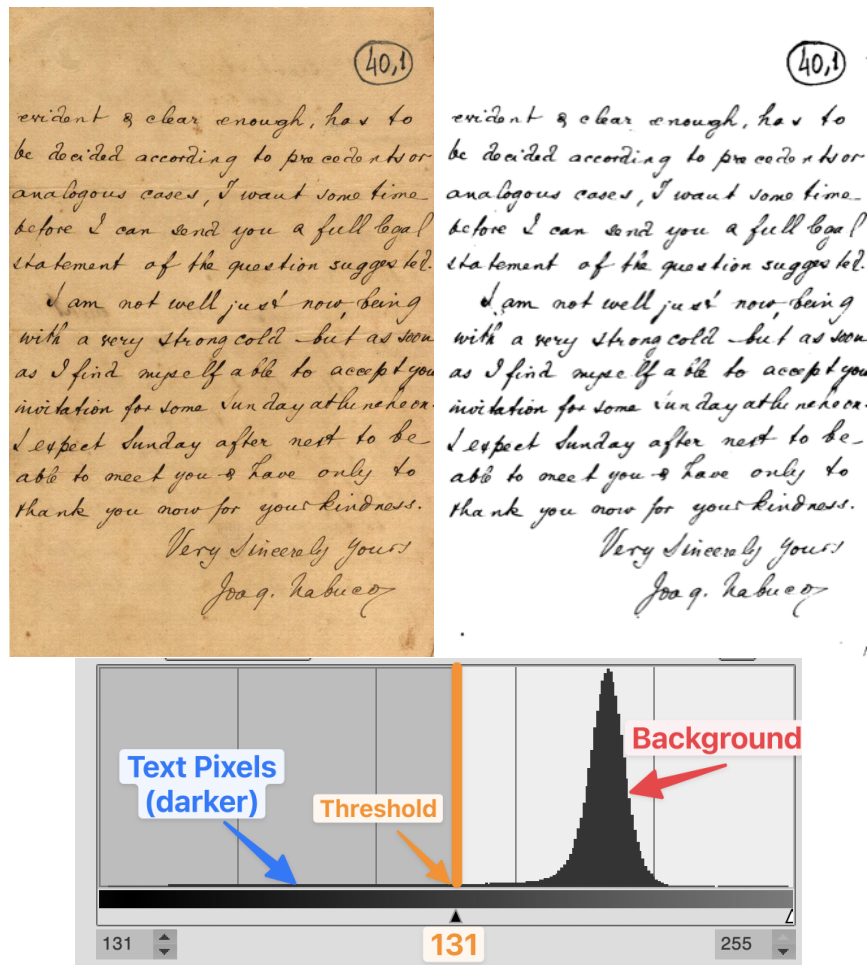
2 BINARIZATION ALGORITHMS

Image binarization is a process of identifying regions of interest in a given image, mapping the color of the pixels into two classes: foreground (black) and background (white). When applied to document images, the text pixels correspond to the foreground, while the paper texture and any noise correspond to the background. There are mainly three situations in which one would want to apply binarization to a document image. The first is to further process it in a Optical Character Recognition (OCR) system, where the highlighted pixels will be used to identify exactly which characters are encoded in an image format to generate a digital, editable, text. The binary image takes much less space when compressed, given the repetition of black and white pixels, thus the second application would be to store and transmit over the Internet large amounts of documents, which can be done much more efficiently if the image size is reduced. The third application would be for further printing, where black and white images save ink and generate much more readable images.

Since the development of the first binarization algorithms, the problem has been approached in a wide variety of ways, from signal processing to classical machine learning and, more recently, applying deep learning. The initial proposals were based on scanning the grayscale histogram of the document image and splitting it into two regions by choosing a threshold. The hues of gray to the left of the threshold are mapped onto black pixels (foreground), while the others are mapped onto white pixels (background). An example of such a process is illustrated in Figure 1. The calculation of the threshold value for the classical methods is usually done by calculating some statistics, entropy, or other measures over the pixel values.

Otsu [2] method (1979) is one of the first global thresholding methods and is still used in many cases due to its simplicity, speed, and effectiveness for images with uniform background. The threshold is calculated by iterating over all the 255 possible threshold values and choosing the one that splits the histogram into two regions and minimizes the within-cluster variance and maximizes the between-cluster variance. Many methods proposed later used either the same histogram analysis principle or directly the Otsu algorithm as part of their binarization approaches, given its simplicity and effectiveness [39, 40, 41]. The Kapur-Sahoo-Wong (KSW) [42] method (1985) treats the foreground and background images as two distinct sources and calculates their entropy by a formula based on Pun's [43] method. The global threshold will be the one that maximizes the entropy of both distributions.

Figure 1 – Example of binarization using grayscale hue thresholding



Source: The author (2024)

After some years, a new type of binarization algorithm was created: local methods. Instead of choosing a global threshold for the whole image, a pixel-wise threshold is calculated by sliding a rectangular window over the gray level image. One of the first and most famous algorithms of this kind was Niblack's algorithm [8] (1985) and is still used by several recent methods as part of their pipeline and served as inspiration for several variations [44]. The local window threshold is determined using the local mean and standard deviation. Usually, it effectively identifies the text regions, but also tends to generate a large amount of background noise in regions without foreground pixels.

Sauvola [45] (2000) significantly improves Niblack's algorithm by computing the threshold using the dynamic range of image gray-value standard deviation. The improvement is more evident on images with a light light background (near 255 gray-level value) and dark foreground (near 0 gray-level pixels). However, in images where the gray values of the text and non-text pixels are close to each other, the results degrade significantly. Wolf [46] (2003) method

further improves it by normalizing the contrast and the mean gray value of the image, using the minimum gray value and the maximum gray-value standard deviation obtained from all sliding windows. In most cases, it outperforms all its predecessors; however, if there is a region with a very abrupt change (sharp noise), it will degrade its performance due to the global statistics involved in the formula. Nick [47] (2009) algorithm improves further by taking care of the issue of black noise in Niblack's method, a low-contrast issue when using Sauvola's algorithm. Alters the formula for the local threshold by moving the threshold downward.

Another important class of binarization algorithms is the one based on energy optimization. The most successful algorithm of such kind is Howe's [48] (2013) binarization method, which uses the Laplacian operator to assess and minimize a global energy function. This function penalizes labelings that do not conform with the image's Laplacian, e.g. foreground pixels mapped as background and vice versa. Additionally, labeling discontinuities are penalized, unless they take place at an edge, which is determined by the Canny edge detection algorithm. In order to minimize the energy function, Howe's algorithm finds the minimal cut to separate foreground and background pixels with the help of a graph cut algorithm [49]. All this process makes Howe a time-costly method, but offers a high binarization quality for many different types of image, especially historical document images.

More recently those important algorithms, among several others, were either reimplemented with faster versions or combined intelligently to create more accurate binary images. iNICK [50] (2017) proposes a new approach to calculate the k values for the Nick binarization method based on the global standard deviation of the image, which increased the quality of the binarization. Westphal reduced the execution time of Howe's method [49] (2018) by correctly mapping its algorithm to be executed taking advantage of a GPU combined with the CPU. Chan [51] (2019) applied integral images to increase Sauvola's method speed by computing the sliding window statistics with integral images. Yuleny's method [22] (2019) applies a XG-Boost classifier trained with features generated from the Otsu, Niblack, Sauvola, Su [52] and Howe algorithms.

The current trend for binarization algorithms follows the overall trend in the scientific community of applying deep learning to improve older and effective methods, combine different types or generating whole new approaches for binarization. Most of the Deep Learning methods rely on traditional algorithms either as their building blocks or as a final step in their execution pipeline. DocDLinkNet [22] (2019) was one of the first successful applications of such an architecture, which first crops the input image into 256×256 patches, applies data

augmentation strategies such as shape and color shift, and trains a D-LinkNet [53] network using the document image patches as input and the corresponding binary maps as ground truths.

DeepOtsu [41] (2019) is a neural network trained to learn the degradation in document images and produce uniform images. The stacked refinement (SR) is applied, which uses a stack of different neural networks for iterative output refinement. The final binarization map is generated by applying Otsu's method. This method provided one of the best binarization output in recent editions of binarization competitions [22, 31] (part of this thesis) and can effectively be used for small images; however, it requires a large amount of memory to process and processing time to process full-sized document images.

DPLinkNet [54] (2021) is a recent proposal that offers state-of-the-art quality binary images for most cases at the cost of processing time and the availability of the GPU in the processing unit. It uses a new fully dilated convolutional network, named FD-Net, using atrous convolutions instead of downsampling or upsampling, which differs from most approaches that uses fully convolutional networks methods.

Deep learning methods often produce high-quality images but, on the other hand, require a powerful setup (which usually includes a GPU unit), are time-costly, and sometimes highly dependent on the training sets. Due to these limitations, traditional image processing algorithms are still being used to create time and quality efficient binarization methods. Michalak21_a [55] (2019), for example, offers one of the best quality and time performances when applied to photographed images, as demonstrated in [56], a study conducted as part of this thesis. The input image is downsampled with bilinear method and then the simple nearest neighbour algorithm, then it is expanded back to its original size with the same kernel, obtaining the image containing only the low frequency information. Next, this image is subtracted from the original, followed by a simple contrast increase and logical negation and the final image is obtained by applying the Otsu method. Even with its simplicity, it performs equally and sometimes even better than several other classical and deep learning methods.

Another state-of-the-art image processing-based algorithm is the YinYang22 [56] method, which is also among the best for most cases of photographed images in the same assessment, reference [56]. It proceeds in 5 main steps. First, the image background is detected by keeping the maximum color occurrence for each pixel close to the neighborhood. In the second step, the background is subtracted from the original image. In the third step, the resulting foreground image is normalized and converted to a gray-level image. In the fourth step, a threshold

image is computed from the foreground image by applying the Otsu method to each loose neighborhood of pixels. In the fifth step, the image is first upsampled and then thresholded thanks to the Otsu threshold image.

During the last decades, several researchers have tried to compare and summarize the area, providing important insights into how this challenging problem evolved [3, 57, 4, 58, 12, 44]. Those articles mostly focus on individual binarization methods, combining them into groups based on the main calculation approach. Recently, Tensmeyer and Martinez [5] brought a new perspective to the analysis, focusing on the individual steps, which cover the preprocessing, actual binarization and post-processing. On Table 1 a timeline of the binarization algorithms tested is presented. The criteria for choosing the algorithms was the source code or executable availability. As the author organized a series of binarization competitions, the algorithms creators sent their code and allowed the execution of this research. In the following section, a more detailed description of the most important algorithms of each type is provided.

2.1 Categories of Binarization Algorithms

Given the large diversity of noises and the complexity of the task, many different approaches to binarization have been proposed. In the following sections, the most important algorithms of each category is briefly discussed.

2.1.1 Threshold Based Binarization

Traditional threshold-based binarization algorithms scan the grayscale histogram of the document image and split it into two regions. The hues of gray (I) that are to the left of the threshold t are mapped onto black pixels (foreground), while the other are mapped onto white pixels (background), as described in Equation 3.8, where $B(i, j)$ is the pixel in the binary image and $I(i, j)$ is the pixel in original image at position (i, j) and t is the threshold value.

$$B(i, j) = \begin{cases} 0 & , \text{if } I(i, j) < t \\ 255 & , \text{if } I(i, j) \geq t \end{cases} \quad (2.1)$$

In the case of local methods, the image is split into regions and the threshold value is determined for each section of the image, as described in Equation 3.9.

$$B(i, j) = \begin{cases} 0 & , \text{if } I(i, j) < T_N \\ 255 & , \text{if } I(i, j) \geq T_N \end{cases} \quad (2.2)$$

The calculation of the threshold value for the classical methods is usually done by calculating some statistics, entropy, or other measures over the pixel values. Even though they have been published one or more decades ago, several of those classical methods are still used either in isolation or as subroutines of other more modern algorithms.

Image Statistics Based Thresholding

Otsu

Otsu [2] method is one of the first global thresholding methods and is still used in many cases due to its simplicity, speed, and effectiveness for images with uniform background. The threshold T_{otsu} is calculated by iterating over all possible threshold values T , which vary in the interval $0 \leq T \leq L$, when applied to the grayscale histogram h of the image and choosing the one that minimizes the within-cluster variance and maximizes the between-cluster variance. The number of pixels w , mean intensity μ , and variance σ of both groups are given, respectively, by

$$w_0(T) = \sum_{i=0}^{T-1} h(i), \quad w_1(T) = \sum_{i=T}^{L-1} h(i) \quad (2.3)$$

$$\mu_0(T) = \frac{1}{w_0} \sum_{i=0}^{T-1} ih(i), \quad \mu_1(T) = \frac{1}{w_1} \sum_{i=T}^{L-1} ih(i) \quad (2.4)$$

$$\sigma_0^2(T) = \frac{1}{w_0} \sum_{i=0}^{T-1} h(i)(i - \mu_0(T))^2, \quad \sigma_1^2(T) = \frac{1}{w_1} \sum_{i=T}^{L-1} h(i)(i - \mu_1(T))^2, \quad (2.5)$$

The Otsu threshold is then defined as the threshold that minimizes within-cluster variance:

$$T_{otsu} = \operatorname{argmin}_T w_0(T)\sigma_0^2(T) + w_1(T)\sigma_1^2(T) \quad (2.6)$$

or equivalently maximizes the between-cluster variance, which reduces to

$$T_{otsu} = \operatorname{argmax}_T w_0(T)w_1(T)(\mu_1(T) - \mu_0(T))^2 \quad (2.7)$$

The threshold is defined by trying all values of T and choosing the one that minimizes Eq. 2.6 or maximizes Eq. 2.7. One disadvantage of Otsu's method is that if there are many peaks in the histogram, which happen, for example, when the image has non-uniform illumination, some of the darker background pixels might be mistaken for foreground ones. This disadvantage applies to all other global methods.

Niblack

Niblack's algorithm [8] was one of the first to apply the concept of local binarization. Instead of choosing a global threshold for the whole image, a pixel-wise threshold is calculated by sliding a rectangular window over the gray level image. Specifically, Niblack's algorithm is still used by several recent methods as part of their pipeline and served as inspiration for several variations [44]. The local window threshold is determined using the local mean and standard deviation:

$$\mu(i, j) = \frac{1}{w^2} \sum_{i'=i-w}^{i+w} \sum_{j'=j-w}^{j+w} I(i', j') \quad (2.8)$$

$$\sigma(i, j) = \sqrt{\frac{\sum_{i'=i-w}^{i+w} \sum_{j'=j-w}^{j+w} (I(i', j') - \mu(i, j))^2}{w^2}}, \quad (2.9)$$

where w is the window size around the pixel (i, j) . The threshold of each pixel is then calculated by

$$T_{niblack}(i, j) = \mu(i, j) + k\sigma(i, j) \quad (2.10)$$

where k is a user-set parameter that controls the trade-off between foreground detection precision and recall. The author recommends $k = -0.2$, however, the optimal threshold will depend on the chosen window size. Usually, it effectively identifies the text regions, but also tends to generate a large amount of background noise in regions without foreground pixels.

Sauvola

Sauvola [45] (2000) significantly improves Niblack's algorithm by computing the threshold using the dynamic range of image gray-value standard deviation. The local binarization problem is solved using sliding windows only in the background, where, for each (i, j) pixel, the threshold is calculated as follows:

$$T_S(i, j) = \mu(i, j) \left[1 + k \left(\frac{\sigma(i, j)}{R} - 1 \right) \right], \quad (2.11)$$

where $\mu(i, j)$ and $\sigma(i, j)$ are computed as in the Niblack method. The authors recommend setting $k = 0.5$ as a user-set parameter and R as the maximum possible standard deviation, which, for 8-bit grayscale images, means $R = 128$.

Unlike Niblack, which adjusts the threshold drop from the mean value $\mu(i, j)$ and takes $\sigma(i, j)$ as a reference, Sauvola bases its adjustment on $\mu(i, j)\sigma(i, j)$. It has inspired many subsequent algorithms [46, 47, 59] and given its effectiveness in many types of images but high computational cost, some efforts were made to improve its efficiency, as in [51].

The improvement is more evident on images with a light light background (near 255 gray-level value) and dark foreground (near 0 gray-level pixels). However, in images where the gray values of the text and non-text pixels are close to each other, the results degrade significantly.

Wolf

Wolf's algorithm [46] is an extension of Sauvola, where the local statistics are normalized based on global statistics:

$$T_{wolf}(i, j) = \mu(i, j) - k \left(1 - \frac{\sigma(i, j)}{S} \right) (\mu(i, j) - M), \quad (2.12)$$

where $S = \max_{ij} \sigma(i, j)$, i.e., the maximum gray value standard deviation value from all windows and $M = \min_{ij} \mu(i, j)$, i.e., the minimum mean gray value from all windows. The k is fixed to 0.5, as recommended by the author. Sauvola expects foreground gray pixels to be close to 0 and background ones close to 255, but if the text is lighter and the contrast is smaller, it will not properly binarize the image. Thus, including the minimum mean and maximum standard deviation in the calculations allows for better handling of images like that, where there is a limited contrast and limited range of grayscale intensity. In most cases, this method outperforms its predecessors.

From the experiments conducted in this work, it works exceptionally well for historical document images, especially those with a darker background (smaller contrast).

CNW

This method is a combination of Niblack and Sauvola [60], calculated as the mean between both thresholds. The final formula for the local threshold is then:

$$T = \frac{2m + mk((\sigma/m) - (\sigma/S) - 1)}{2} \quad (2.13)$$

where σ is the standard deviation of the image, m is the mean of the local window, S is the maximum standard deviation, $k = 0.35$. This algorithm, although very simple, managed to appear as the top-rated algorithm in most datasets tested, specially if one uses only the red channel of the image to binarize. It is more indicated for photographed documents with uneven illumination.

Entropy Based Thresholding

Mello-Lins

The Mello-Lins algorithm [14] was possibly the first binarization algorithm capable of removing the back-to-front interference. The algorithm scans the histogram of the converted grayscale image in search of the most frequent hue of gray, which is supposed to belong to the background of the image (the paper). Such a hue of gray is used as a threshold value, t , to evaluate the entropy [1] of the black ($H_b < t$) and white ($H_b \geq t$) pixels. Three Shannon entropies are calculated using the following equations:

$$\begin{cases} H = \sum_{i=0}^{255} p_i \log_{X.Y}(p_i) \\ H_b = \sum_{i=0}^t p_i \log_{X.Y}(p_i) \\ H_w = \sum_{i=t+1}^{255} p_i \log_{X.Y}(p_i) \end{cases} \quad (2.14)$$

where $p[i]$ is the probability of the hue of gray i in the histogram. The logarithm is taken on the basis $X.Y$, where X and Y are the dimensions of the complete image. The value of H is used to define two multiplicative factors, m_w and m_b , whose values were experimentally determined by the rules:

If $H \geq 0.25$, then $m_w = 2.0$ and $m_b = 3.0$

If $0.25 < H < 0.30$, then $m_w = 1$ and $m_b = 2.6$

If $0.30 < H < 0.305$, then $m_w = 1$ and $m_b = 2.0$

If $H \geq 0.305$, then $m_w = m_b = 0.8$

The hue i in the grayscale image is turned white if $hue[i] \geq (m_w * H_w + m_b * H_b)$,
Otherwise, it is made black.

In the experiments of this thesis, it appeared as one of the top algorithms in a few datasets, even being a global approach. It works especially well for historical handwritten images with a lighter background.

Kapur-Sahoo-Wong

The Kapur-Sahoo-Wang (KSW) method [42] is an extension of Pun's method [43] and is based on entropy calculation in a global context. Consider the object likelihood distribution P_t and the background likelihood distribution $(1 - P_t)$ in determining the division entropy. The binarization threshold T_{KSW} is chosen by testing each possible value and selecting the one that maximizes the combination of the object and background entropy ($H = H_{object} + H_{background}$), where:

$$H_{object} = - \sum_{i=0}^t \frac{P_i}{P_t} \times \log \frac{P_i}{P_t}, \quad H_{background} = - \sum_{i=t+1}^{255} \frac{P_i}{1 - P_t} \times \log \frac{P_i}{1 - P_t} \quad (2.15)$$

and P_i is the likelihood of occurrence of the gray-level i in the image, and $P_t = \sum_{i=0}^t P_i$

Although it did not appear among the top algorithms very often, in some experiments, as in [56], it appeared as the top algorithm for photographed images using two different devices, which highlights that classical global algorithms can still provide good results at a cost of very small amount of time.

2.1.2 Edge Detection Based Binarization

Edge detection is the process of estimating the boundary of foreground objects present in an image, which has been extensively used to compose document image binarization methods. One of the most popular methods is the Canny edge detection, which uses the gradient magnitude image [5, 61]. Several algorithms use edge detection as a fundamental part of their binarization process, such as Jia-Shi [62], Akbari [63] or Su-Lu [64].

Su-Lu

Generates an adaptive image contrast map as a combination of the local image contrast and the local image gradient. First, a contrast map is built, then binarized and combined with the Canny edge map to calculate the text stroke edge pixels. The document text is further segmented by a local threshold that is estimated using the intensities of the detected text stroke edge pixels within a local window. Finally, some post-processing is applied to improve the final binarization. The contrast calculation is based on the Bernsen [9] method, but with a normalization factor to compensate for the image variation within the document background. It involves minimal parameter tuning and has a relatively small computational cost.

In the tests conducted, the Su-Lu algorithm frequently appears among the top ranked for historical images of all kinds, but not for photographed images. Given its low computational cost and the quality of the images produced, it is a highly recommended algorithm to be included in any historical document processing pipeline.

2.1.3 Optimization Based Binarization

Another category of binarization algorithms is the optimization-based algorithms [48, 65, 66, 67, 68, 69], which, in most cases, do not rely on parameter tuning and binarize with a soft decision process, as opposed to the sharp decision taken by thresholding. One remarkable example of such algorithms is the one proposed by Howe [48], which uses a Laplacian operator, Canny edge detection, and graph-cut method to find the threshold minimizing the energy. It has been used as a step of several other newer algorithms [70, 71]. The main drawback of optimization-based approaches is its computational complexity, which can be even prohibitive for applications with processing time constraints, as reported on [72].

Howe

Howe's method [48] minimizes a global energy function, formulated as a graph-cut problem for efficient exact computation. First, it defines the target binarization as a labeling on pixels that minimizes a global energy function inspired by a Markov random field model. Second, in formulating the data-fidelity term of this energy, it relies on the Laplacian of the image intensity to distinguish ink from background. This grants a crucial invariance to differences in contrast and overall intensity. Third, it incorporates edge discontinuities into the smoothness

term of the global energy function, biasing ink boundaries to align with edges and allowing a stronger smoothness incentive over the rest of the image.

Howe is another algorithm that often appears among the best in several datasets, however, in the latest experiments conducted for this thesis, it did not. It is considered as one of the best binarization algorithms in terms of quality, however, its required processing time can be prohibitive for some applications when binarizing a full-sized document image.

2.1.4 Image Processing Based Binarization

The algorithms gathered in this category use several classical image processing techniques in a way that generates clear binary images in a timely-efficient way.

Michalak21_a or MO₁

The first step of the algorithm is related to image downsampling where one of the well-known interpolation methods may be applied. For this purpose the MATLAB function `imresize` was used with bilinear and the simple nearest-neighbor method. The application of a relatively large kernel during the downsizing of the image results in the loss of details related to the shapes of individual characters. Therefore, only the low-frequency image data is preserved representing the overall distribution of the image brightness, being in fact mainly the downsampled background information. After resizing back the downsampled image to the original size using the same kernel, the image containing only the low-frequency information is obtained, representing the approximated high-resolution background. In the next step of the proposed method, the subtraction of this image from the original is made to enhance the text data, followed by simple contrast increase and logical negation. The image obtained is subjected to fast global thresholding using the Otsu method [55].

In the experimentation conducted for the development of this thesis, this algorithm very often appears among the best, especially for photographed images. It has been developed for uneven illuminated images, exactly the type of image that is most often generated when capturing documents with mobile cameras. In the paper [36], the winner in all categories was this algorithm or other algorithms proposed by the same authors for the same purpose.

Michalak21_b or MO₂

This method is based on the equalization of the illumination of an image, which also increases its contrast, making it easier to conduct the proper binarization. It is based on an analysis of the local entropy, assuming its noticeably higher values in the neighborhood of the characters. Hence, only the relatively high-entropy regions should be further analyzed as potentially containing some characters, whereas low-entropy regions may be considered as the background. The additional steps of the morphological dilation, increase of contrast, and final binarization using Bradley's method are made during the final stage [73].

Michalak21_c or MO₃

The initial idea of the application of the region-based binarization for text recognition was presented assuming the application of document images containing predefined text. The proposed improved method assumes the division of the image into regions of $N \times N$ pixels. For each of the regions, the local threshold can be determined as $T = a * \text{mean}(X) - b$, where $\text{mean}(X)$ is the average brightness of the image region and the parameters a and b are subjected to optimization. The algorithm is based on the same idea of calculation of the local thresholds as the average brightness corrected by two parameters; however, the number of regions is higher than would result from the resolution of the image, and therefore they partially overlap each other. In this case for each subregion several threshold values are calculated depending on the number of overlapping blocks covering the subregion. The resulting local threshold is determined as the average of the threshold values calculated for the number of regions dependent on the assumed number of layers and the overlapping factor. The rationale for such an approach is a better tolerance of rapid illumination changes with the ability to correct the binarization of the image [74].

YinYang21

The "YinYang21" binarization algorithm detects the background of the original image using small overlapping windows. First, each window calculates its median color using a quantized color palette. Then, the estimated background image is generated by interpolating the computed median pixels of the overlapping windows. Next, the background image is subtracted

from the original image and the resulting difference image is transformed into grayscale, keeping only the lowest RGB component. The binarization is performed using the Otsu algorithm. Detection and removal of small isolated connected components is made. The algorithm submitted in this competition is a faster and more accurate version of the one previously submitted in the DocEng 2020 binarization competition [31].

YinYang22

The “YinYang22” binarization algorithm detects the background of the original image using small overlapping windows. First, each window calculates its median color using a quantized color palette. Then, the estimated background image is generated by interpolating the computed median pixels of the overlapping windows. Next, the background image is subtracted from the original image and the resulting difference image is transformed into grayscale, keeping only the lowest RGB component. The binarization is performed using the Otsu algorithm. Detection and removal of small isolated connected components is made. It has been only published in the series of DocEng Time-Quality Binarization Competition, which were organized as part of this thesis [56].

Similar to Michalak’s variants, YinYang algorithms also generates high quality binary images, with almost perfect binarization for most photographed document images. It has been developed for this kind of image and has good performance, comparable to some classical local binarization methods.

2.1.5 Pixel Classification Based Binarization

More recently, the document image binarization problem has been mostly solved by machine learning models, as demonstrated by the large number of new algorithms that use this approach [20, 22, 75, 41, 76]. This type of algorithm encompasses a whole new category, where, in general, a neural network is trained to learn when a pixel is more likely to be mapped onto black or white based on a set of training images with their corresponding ground truth. They often generate good quality binary images, but also require a much higher processing time to generate the binary image [22, 77], besides the algorithm training time. Another issue is that they require a representative dataset, with a large number of example images, hence the efficacy is highly dependent on the quality and size of the training set.

Huang [33] has proposed two deep learning approaches that split the input image into small patches that are later combined, training a BDC-Unet based model another Unet based model [78]. Vahid's [33] method trains a Resnet50-Unet network that combines data sets from binarization competitions and a private one from the Berlin State Library. DocUNet [22] also uses a variation of UNet [79] to perform pixel classification, but applying morphological operations to enhance the input image and stroke width transform (SWT) to determine the size of the structural element used in the network. DeepOtsu [41] also uses deep learning, but instead of predicting the label of each pixel, it learns the degradation and removes it iteratively.

Classical Machine Learning

Gosh

It consists of three sequential steps, each of which consists of further sub-steps. The first step includes pre-processing activities which comprise of background separation and image normalization steps. The second section deals with the thresholding applying an ensemble of three classical clustering algorithms: Fuzzy C-means, K-medoids and K-means++ to group the pixels as foreground or background. The final section discusses the post-processing steps.

This algorithm was first published at the ICDAR 2019 Time-Quality Binarization Competition [22] and was later published at [80]. It was one of the best algorithms in terms of quality at the later occurrence of this competition, at ICDAR 2021, being either the best of among the five best ranked algorithms for nearly all datasets. It surpassed several modern deep learning approaches and even other important algorithms also referenced in this sections. However, it has a poor time-performance, being slower than most of the other top-ranked algorithms.

Deep Neural Networks

Akbari Algorithms

This binarization methodology relies on a Segnet network architecture that is fed by multi-channel images that correspond to the original image and the image approximations based on the coefficients of three sub-bands [81] and the image binarized using the structural symmetric pixels (SSPs) method [63]. Multichannel images were implemented and used as network inputs. Three versions of the method have been proposed:

- **Method (1):** The original image is decomposed into wavelet sub bands, the original image binarized by the structural symmetric pixels (SSPs) method (single network).
- **Method (2):** Variation of (a) with multiple networks.
- **Method (3):** Variation of (a) where fewer channels are used to reduce computational cost.

This algorithm frequently appears among the best ranked for historical and photographed document images. It does not have a high computational cost but is not comparable to classical methods in terms of time, being one order of magnitude slower.

DocDLinkNet

This method consists of three main steps. First, the original image is cropped into 256×256 patches. Data augmentation strategies such as shape shift and color shift are applied. Second, a D-LinkNet architecture [53] is adopted and trained by using document image patches as input and the corresponding binary maps as ground truths. D-LinkNet is a semantic segmentation neural network, which involves dilated convolution and pretrained encoder. Finally, the Principal Component Analysis (PCA) method is used to perform image dimensionality reduction and feature extraction, and then generates the final results according to the optimal parameters learned from the training procedure.

It was first published in the series of competitions which made up part of this thesis [22].

It has high computational costs, but very often appears among the best when processing scanned or photographed images. It is definitely one of the best algorithms in terms of quality, but in many cases it is possible to find another much faster algorithm with similar quality performance.

HuangBCD and HuangUnet

First appeared at ICDAR 2021 time-quality binarization competition [33]. A combination of binary cross-entropy and dice loss is chosen as the loss function of a deep-learning algorithm. Data augmentation is performed in the training process to improve the scores. The original colored or gray images are divided into patches with the same dimension (e.g. 128×128). For each colored patch, a trained Unet model is utilized to obtain a binarized patch. A binarized large image with the same size as the original image can be obtained with the combination of

those binarized patches. In this method, the model stacking technique is performed via two Unet models with patch dimensions of 128×128 and 256×256 . In addition, a global view with a patch dimension of 512×512 is also combined to obtain the final results. The model with a global view is trained aiming to capture the global context and the character locations.

There are two variations of the algorithm:

- **Method (1) - HuangBCD:** The segmentation model is BCD-Unet based[78]
- **Method (2) - HuangUnet:** The segmentation model is Unet based.

Huang's algorithms are another example of an algorithm which very often appears among the best, many times even better than DocDLink, but at the cost of processing time, which is one order of magnitude higher.

DiegoPavan

The "DiegoPavan" binarization method chooses to downscale the input image, rather than using patching, and then rescaling the network output to the input original size. The network architecture used is based on DE-GAN [82], where the input image is changed to HSV representation, the hyperparameters, and the training process were adjusted, including image augmentation.

Vahid

The "Vahid" algorithm is based on machine learning and is in fact a pixel-wise segmentation model. The dataset used for training is a combination of training sets for binarization competitions in different years with pseudo-labeled images from their dataset in the Berlin State Library. A specific dataset has been produced for very dark or bright images. The model is based on a Resnet50-Unet [83].

DocUNet

The DocUNet method comprises three main steps. Firstly, a bottom-hat morphological transform is performed to enhance the document image contrast, and the size of a disk-shaped structural element is determined by the stroke width transform (SWT). Secondly, a hybrid pyramid U-Net convolutional network [79] is performed on the enhanced document

images for accurate pixel classification. Finally, the Otsu algorithm is applied as an image post-processing step to yield the final image.

Table 1 – Tested binarization algorithms

Method	Year	Category	Description
Percentile [84]	1962	Global threshold	Based on partial sums of the histogram levels
Triangle [85]	1977	Global threshold	Based on most and least frequent gray level
Otsu [2]	1979	Global threshold	Maximize between-cluster variance of pixel intensity
IsoData [86]	1980	Global threshold	IsoData clustering algorithm applied to image histogram
Pun [43]	1981	Global threshold	Defines an anisotropy coefficient related to the asymmetry of the histogram
Johannsen-Bille [87]	1982	Global threshold	Minimizes formula based on the image entropy
Kapur-SW [42]	1985	Global threshold	Maximizes formula based on the image entropy
Moments [88]	1985	Global threshold	Aims to preserve the moment of the input picture
Niblack [8]	1985	Local threshold	Based on window mean and the standard deviation
Bernsen [9]	1986	Local threshold	Uses local image contrast to choose threshold
MinError [89]	1986	Global threshold	Minimum error threshold
Mean [90]	1993	Global threshold	Mean of the grayscale levels
Shanbhag [91]	1994	Global threshold	Improves Kapur-SW by viewing the two pixel classes as fuzzy sets
Huang [92]	1995	Global threshold	Minimizes the measures of fuzziness
Yen [93]	1995	Global threshold	Multilevel threshold based on maximum correlation criterion
RenyEntropy [94]	1997	Global threshold	Uses Renyi's entropy similarly as Kapur-SW method
Sauvola [95]	1997	Local threshold	Improvement on Niblack
Li-Tam [96]	1998	Global threshold	Minimum cross entropy
Wu-Lu [97]	1998	Global threshold	Minimizes the difference between the entropy of the object and the background
Mello-Lins [14]	2000	Global threshold	Uses Shannon Entropy to determine the global threshold. Possibly the first to properly handle back-to-front interference
Wolf [69]	2002	Local threshold	Improvement on Sauvola with global normalization
ISauvola [98]	2004	Local threshold	Uses image contrast in combination with Sauvola's binarization
Ergina-Global [99]	2005	Global threshold	Average color value and histogram equalization
Ergina-Local [100]	2006	Local threshold	Detects where to apply local thresholding after a applying a global one
Intermodes [101]	2006	Global threshold	Smooth histogram until only two local maxima
Minimum [101]	2006	Global threshold	Variation of Intermodes algorithm
dSLR [102]	2006	Global threshold	Uses Shannon entropy to find a global threshold
Bradley [103]	2007	Local threshold	Adaptive thresholding using the integral image of the input
Nick [47]	2009	Local threshold	Adapts Niblack based on global mean
ElisaTV [104]	2010	Local threshold	Background estimation and subtraction
Lu-Su [105]	2010	Edge based	Local thresholding near edges after background removal
Bataineh [106]	2011	Local threshold	Based on local and global statistics
Singh [107]	2011	Global threshold	Uses integral sum image prior to local mean calculation

Table 1 – *Cont.*

Method	Year	Category	Description
Howe [48]	2013	CRF Laplacian	Unary term and pairwise Canny-based term
Su-Lu [64]	2013	Edge based	Canny edges using local contrast
iNICK [50]	2017	Local threshold	Adaptively sets k in Nick method based on the global standard deviation
CNW [60]	2018	Local threshold	Combination of Niblack and Wolf's algorithm
DocDLinkNet [53]	2018	Deep Learning	D-LinkNet architecture with document image patches
Gattal [108]	2018	Clustering	Automatic Parameter Tuning of K-Means Algorithm
Jia-Shi [62]	2018	Edge based	Detecting symmetry of stroke edges
Robin	2018	Edge based	U-net model trained with several datasets (https://github.com/masyagin1998/robin , accessed on 19 January 2023)
WAN [109]	2018	Global threshold	Improves Sauvola's method by shifting up the threshold
Akbari_1 [63]	2019	Deep Learning	Segnet network architecture fed by multichannel images (wavelet sub bands)
Akbari_2 [63]	2019	Deep Learning	Variation of Akibari_1 with multiple networks
Akbari_3 [63]	2019	Deep Learning	Variation of Akibari_1 where fewer channels are used
CLD [110]	2019	Local threshold	Contrast enhancement followed by adaptive thresholding and artifact removal
Calvo-Zaragoza [75]	2019	Deep learning	Fully convolutional Encoder-decoder FCN with residual blocks
DeepOtsu [41]	2019	Deep Learning	Neural networks learn degradations and global Otsu generates binarization map
DocUNet [22]	2019	Deep Learning	Hybrid pyramid U-Net convolutional network fed with morphological bottom-hat transform enhanced document images
Michalak21 _a [55]	2019	Image Processing	Downsample image to remove low-frequency information and apply Otsu
Michalak21 _b [73]	2019	Image Processing	Equalize illumination and contrast, apply morphological dilatation and Bradley's method
Michalak21 _c [74]	2019	Local threshold	Average brightness corrected by two parameters to apply local threshold
Michalak [55]	2019	Image Processing	Downsample image to remove low-frequency information and apply Otsu
Yasin [22]	2019	Image Processing	Gradient descent optimization followed by Otsu thresholding
Yuleny [22]	2019	Shallow ML	A XGBoost classifier is trained with features generated from Otsu, Niblack, Sauvola, Su and Howe algorithms
DiegoPavan [82]	2020	Deep Learning	Downscale image to feed a DE-GAN network
DilatedUNet [31]	2020	Deep Learning	Downsample to smooth image and use a dilated convolutional layer to correct the feature map spatial resolution
YinYang [31]	2020	Image Processing	Detect background with median of small overlapping windows, extract it and apply Otsu

Table 1 – *Cont.*

Method	Year	Category	Description
YinYang21 [31]	2020	Image Processing	A faster and more effective version of YinYang algorithm
DE-GAN [82]	2020	Deep Learning	Uses a conditional generative adversarial network
Gosh [80]	2021	Clustering	Clustering applied to a superset of foreground estimated by Niblack's algorithm
HuangBCD [33]	2021	Deep Learning	BCD-Unet based model to binarize and combine image patches
HuangUNet [33]	2021	Deep Learning	Unet based model binarize and combine image patches
Vahid [33]	2021	Deep Learning	A pixel-wise segmentation model based on Resnet50-Unet
HBUT [111]	2021	Image Processing	Morphological operations using minimum entropy-based stroke width transform and Laplacian energy-based segmentation
DPLinkNet [54]	2021	Deep Learning	Fully dilated convolutional network using atrous convolutions
Vahid22 [56]	2022	Deep Learning	Pixel-wise segmentation combining a CNN with a transformer model
YinYang22 [56]	2022	Image Processing	Uses maximum color occurrence to detect and subtract background, then normalize and apply Otsu

Source: The author (2024)

3 ASSESSING DOCUMENT IMAGE BINARIZATION ALGORITHMS

Document image binarization serves three primary purposes: converting an image into digital, editable text; archiving large volumes of documents efficiently; or preparing images for high-quality printing. However, document images often contain diverse types of noise that can complicate the binarization process, and different algorithms handle these challenges with varying levels of success. Given the hundreds of binarization algorithms available in the literature, selecting the most suitable one for a specific application remains a complex and challenging task.

For decades, researchers attempted to evaluate the most prominent algorithms in order to find the advantages and drawbacks of each method and type of method. The first studies in this area did not focus on documents, and empirical criteria were often used to determine the effectiveness of the methods. For example, Lee [112], in 1990, used images of shapes and photos and shape similarity to evaluate the performance of five binarization algorithms. Possibly, the first objective evaluation that conducted a comprehensive assessment of binarization algorithms was that by Trier [15], in 1995, which analyzed the performance of 11 binarization algorithms using an experimental Optical Character Recognition (OCR) system to recognize digits on hydrographic maps. The values of recognition, reject, and error rates are used to compare the different methods. The processing time is also registered and reported. Leedham [113] compared five binarization algorithms using precision and recall analysis of the resultant words in the foreground.

Later, in 2004 Sezgin and Sankur [3] published the largest assessment at that time, evaluating 40 algorithms, which were mostly global methods, with a detailed analysis on synthetic data of shapes, circuits, characters and many other. For each image, a ground truth was generated and several statistical measures are applied. However, it does not measure the processing time and does not focus on documents.

So far, there has been no quality measure to evaluate document image binarization. Unlike other types of image, even a couple of wrongly mapped pixels may affect the characters readability and further processing. Observing this fact, Lu et al. [114] proposed a new quality measure that takes into account the distance between character strokes and the wrongly mapped pixels close to it. It was called Distance-Reciprocal Distortion Measure (DRD), and it has been shown to better quantify the visual distortion perceived by human readers when

compared with peak signal-to-noise ratio (PSNR).

In most of the studies so far, the whole image processing is evaluated, without focusing on the binarization separately. Ideally, the evaluation method should be isolated in the evaluation and, for that, new evaluation metrics should be used. Stathis et al. [17] used synthetic documents to assess the effect of back-to-front interference in the binarization process. They proposed an overall measure to quantify the number of correctly mapped pixels: pixel error rate (PERR), which counts the proportion of correctly mapped pixels in relation to the total number of pixels. They also used traditional measures for the signal-to-noise ratio (SNR) of video quality and the peak signal-to-noise ratio (PSNR) and showed that PERR is enough to measure quality. This was also possibly the first study to evaluate full-size images and one of the first to focus entirely on document images.

In 2009, Gatos et al. [19] proposed the first binarization algorithms applied specifically to historical document images, evaluating 43 binarization methods. Some statistical measures traditionally used in image processing evaluation were used along with some classification measures applied to the two classes of pixels. However, they have conducted a blind evaluation that does not take into account the specific document characteristics and did not include any of the previously proposed classical binarization algorithms. However, the quality measures used by them have become a standard in binarization evaluation and have been used by most binarization studies ever since.

Kefali et al. [115], in 2010, implemented 12 binarization algorithms to evaluate old Arabic documents. They proposed a new evaluation method that, instead of comparing the images with a ground truth and applying OCR, they manually extract the text features and measure the edit distance to convert the binarized image into the original image.

In 2013, Ntirogiannis et al [116] proposed a new binarization measure called pseudo-FMeasure (F_{ps}), which modifies the traditional FMeasure applying a weight matrix. Penalize pixels that break the character stroke or add noise around the characters. This means that if a binarization algorithm degrades regions far from text it will not penalize as much as close to text.

Later, Sekeroglu et al [117] conducted an evaluation with 13 methods and 174 images, being one of the largest databases evaluated at its time. Only a few global and local methods were evaluated. a new evaluation criteria was proposed that was a combination of visual inspection and computer-computed measures derived from PSNR was proposed.

Ismail et al. [58] performed, in 2018, one of the largest evaluations to date, where 29

thresholding algorithms were analyzed. It was focused only on statistical methods, and thus the global, local, and hybrid thresholding was excluded, having excluded several other important categories of methods. The methods were divided into a taxonomy based on the characteristics used to calculate the threshold. The methods were also briefly described, along with their formulas and the most relevant features. The evaluation criteria were the same as those proposed by Gatos [19]. The specific results were not presented, only an overall idea for each algorithm is discussed. Only DIBCO images were used.

In 2019, Sulaiman et al. [12] published a review on the area that did not assess any algorithm but indicated the most important algorithms and how the area evolved. It also presented a summary list that included many of the most important algorithms, from the initial global and local thresholding method until the modern deep learning-based ones. It also highlighted the challenges, evaluation metrics, and pointed out some direction to where the area is evolving.

In 2020, Tensmeyer [5] presented a new perspective on the area by organizing the algorithms by topic instead of focusing on individual methods. This was motivated by the fact that binarization algorithms are usually composed of many steps, which might include a pre-processing or post-processing step. Instead of presenting the algorithms and what they do, Tensmeyer discussed the techniques and which algorithms use them. In doing so, the contributions of the individual operations are highlighted and help future researchers decide what to include in their binarization processing pipeline.

Since 2019, several binarization competitions have been organized by the author of this thesis in cooperation with the DIB team, which is part of this thesis. These competitions highlighted the importance of taking into account the processing time, using a full document instead of just a portion of it (as in DIBCO competitions), and grouping the datasets by the document image characteristics. More recently, the different versions of the input image (red, green, and blue channels), in addition only the grayscale version and the resulting compressed file size of the binary image, were also added. The fact that the competition attracted many competitors and repeated every year since its first edition shows how relevant such analysis is for this area.

This chapter explores the challenges of evaluating binarization algorithms and presents the innovative solutions developed in this research. A detailed discussion is provided on the most widely used document binarization datasets with available ground truth, emphasizing their key characteristics and relevance. The chapter also reviews and critiques the most commonly employed evaluation measures, shedding light on their strengths and limitations. Furthermore, the

chapter examines binarization competitions, which represent the most prominent assessments in the field, and introduces a novel approach proposed in this work. These foundations establish the basis for the methodology described here to be applied in subsequent chapters, where a new binarization framework is proposed. This framework automatically selects the optimal algorithm for a given document image based on its characteristics, advancing the state of the art in document image binarization.

3.1 Datasets for Document Image Binarization

Nabuco Bequest

The letters of Joaquim Nabuco (b. 1849/d. 1910), a Brazilian statesman who was the first Brazilian ambassador to the USA and one of the most expressive figures in freeing black slaves in the Americas, are of great historical importance, and some of them are available in the DIB dataset. The images were generated as part of the Nabuco Project [118] in which a flatbed scanner was used to scan all letters from him. The scanner resolution was 200 dpi, saved in JPEG format with 1% compression rate. The final images have resolutions of 900×1400 , 1500×1800 , 1600×2000 and 1100×1800 pixels. The most common types of noise present in document images are found in those letters, making this dataset a good representative of such kinds of documents.

The specific subset of those images used in this research is composed of 39 images representative of the whole dataset, with dark and light background textures, handwritten and typewritten text, stain, folding marks, smudges, and several levels of back-to-front interference.

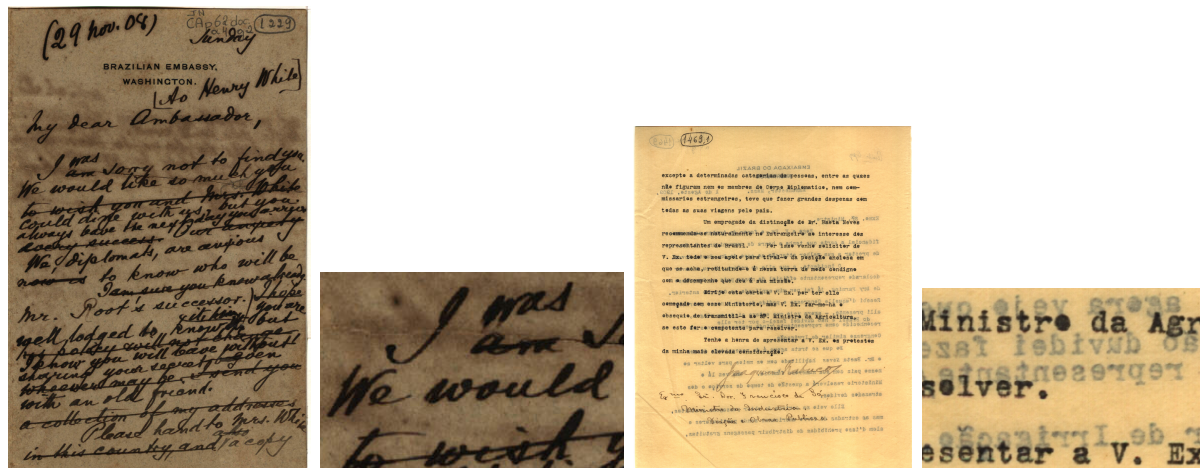
In Figures 2 and 3, some details of example images with these noises are presented, and in Figure 4 the whole dataset to which ground-truth images were generated is presented. One may zoom in to see a greater level of detail on each image. On Table 2 the dimensions of the dataset are presented.

Figure 2 – Nabuco Light Handwritten Example With strong back-to-front interference



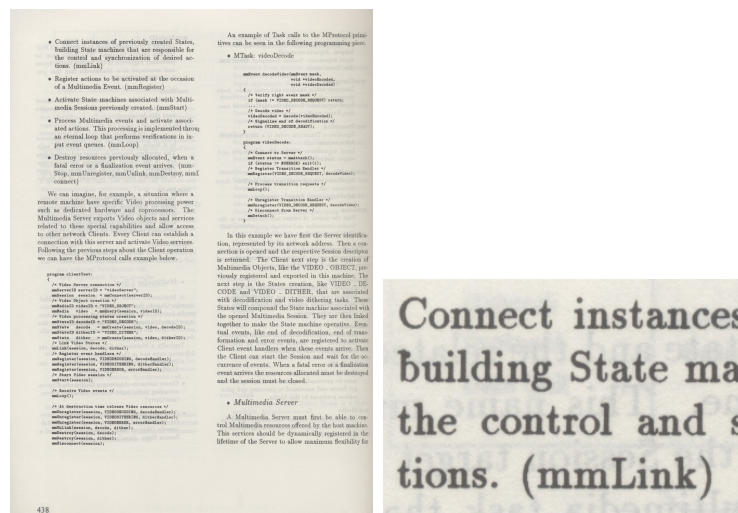
Source: The author (2024)

Figure 3 – Nabuco Dark Handwritten and Mid Typewritten Example Images



Source: The author (2024)

Figure 5 – Livememory Example Image



Source: The author (2024)

Table 2 – Nabuco 39-dataset images dimentions in pixels.

Image	Size	Image	Size	Image	Size	Image	Size
HW01	888 × 1361	HW11	907 × 1383	HW21	1077 × 1345	TW05	1602 × 2035
HW02	915 × 1358	HW12	937 × 1372	HW22	894 × 1387	TW06	1551 × 1947
HW03	920 × 1374	HW13	924 × 1381	HW23	925 × 1376	TW07	1212 × 1692
HW04	911 × 1426	HW14	895 × 1373	HW24	992 × 1552	TW07	1212 × 1692
HW05	1021 × 1586	HW15	999 × 1557	HW25	912 × 1375	TW09	1619 × 1961
HW06	1024 × 1550	HW16	890 × 1380	HW26	891 × 1381	TW10	1599 × 2067
HW07	898 × 1389	HW17	954 × 1401	TW01	1645 × 2140	TW11	1701 × 1957
HW08	1016 × 1570	HW18	1049 × 1670	TW02	1660 × 2186	TW12	1677 × 2179
HW09	866 × 1354	HW19	917 × 1372	TW03	1581 × 2119	TW13	1692 × 2193
HW10	1021 × 1579	HW20	1050 × 1326	TW04	1575 × 1989	TW14	1671 × 2165

Source: The author (2024)

The Nabuco and LiveMemory datasets used in the experiments here are part of the DIB - Document Image Binarization data set (<https://dib.cin.ufpe.br/>), which is part of the IAPR-TC10/TC11 open repository of document images [28].

LiveMemory

The LiveMemory Project [119] was a pioneering initiative to build a digital library of the entire collection of proceedings of the Brazilian Telecommunications Society (SBrT) technical events back in 2007. The real challenge was to scan all the printed-only volumes, semi-automatically index all the papers, enhance image quality, and to binarize the images in way such as to allow all the volumes to be stored in a single DVD, which was handed to all members of the SBrT. The documents were scanned in 200 dpi, true-color and stored using the jpeg file-format with standard (1% loss). The LiveMemory dataset is clearly the one with a smaller variation among images, as they are all “modern” documents, offset printed and have a uniform background with some back-to-front interference.

DIB Mobile

As a result of this thesis, the photographed document images dataset in the DIB platform has been greatly expanded with 296 new images and 7 new devices. The dataset is composed of modern documents photographed from different positions and illumination conditions. The first was created to compose synthetic documents on the platform. The other four were

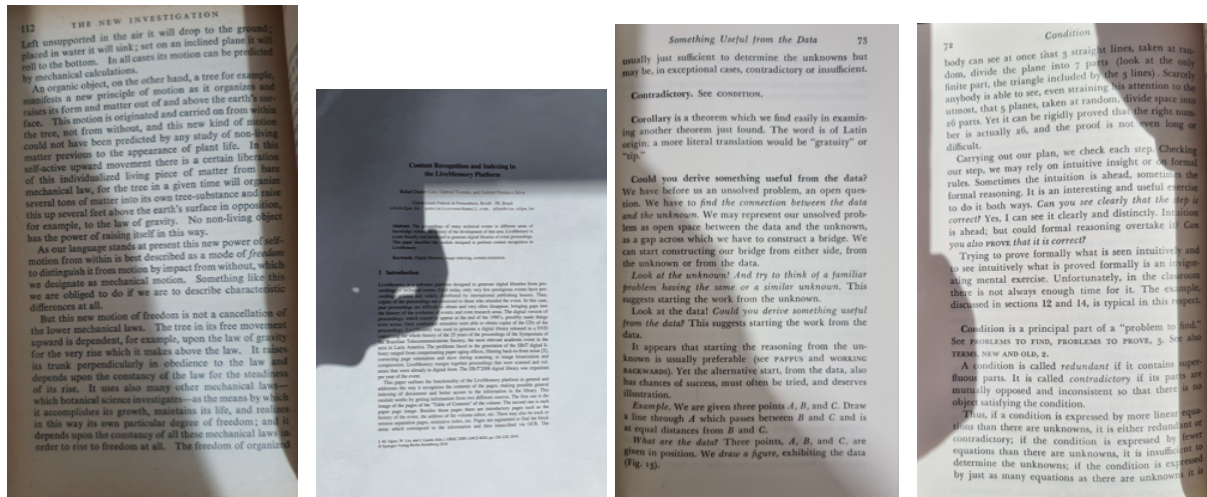
proposed in several binarization competitions in recent years, which were organized as part of this research. On Table 3, the camera specifications of the new devices are presented. In Figures 6 and 7 some example images are presented.

Table 3 – Summary of device camera specifications

	Samsung N10	Samsung S21U	Moto. G9 Plus
Megapixels	16	12	12
Aperture	F 1.5-2.4	F/1.5	F 1.8
Sensor size	1/2.55 inch	F 1.8 inch	1/1.73 inch
Pixel size	-	1.4 μm	1.4 μm
Release yr.	2019	2021	2020
	Samsung A10S	Samsung S20	iPhone SE2
Megapixels	13	12	12
Aperture	F 1.9	F 1.8	F 1.8
Sensor size	-	1/2.55 inch	1/3 inch
Pixel size	-	1.4 μm	1.4 μm
Release yr.	2020	2020	2020

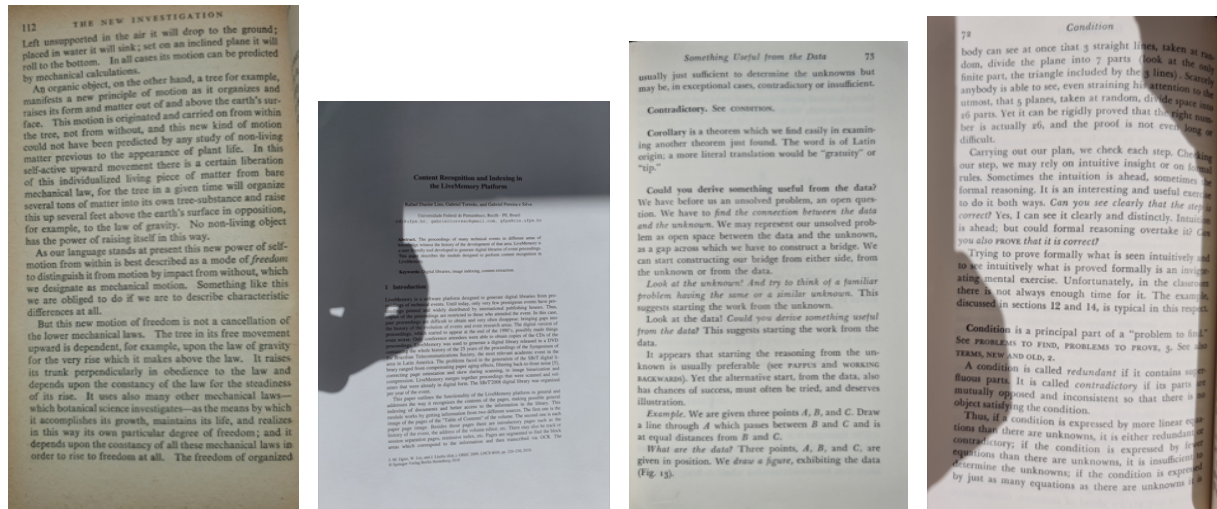
Source: The author (2024)

Figure 6 – DIB Mobile sample images clustered by device (Samsung Note 10+, Samsung S21) and set-up of the strobe flash “off”.



Source: The author (2024)

Figure 7 – DIB Mobile sample images clustered by device (Samsung Note 10+, Samsung S21) and set-up of the strobe flash bottom-line “on”.



Source: The author (2024)

DIB Synthetic

An online platform (<https://cin.ufpe.br/>) has been developed (part of a previous research by this author and the team at that time) to provide a tool to generate over 5 million of different synthetic document images. The user can choose from 231 real-world documents, 200 paper textures, 2 color types (colored or grayscale), 10 levels of back-to-front interference, 3 levels of blur and 3 lengths of shift of the back-to-front interference with the foreground. Once assembled, it is possible to retrieve the binarization results for 30 binarization algorithms, download the synthetic image, and the binary image for each algorithm. Several analysis has been performed with a selected set of images extracted from this dataset, including the ICDAR 2019 competition [22], they have been useful to show that no binarization algorithm is an all-time winner.

PRImA

The PRImA database used in this research is mainly composed of Europeana Newspapers [120]. Its main goal is to provide a representative collection of all the types of newspapers which are and/or might be subject of ongoing or future digitization activities. As such, it is hosting scanned images, metadata, and ground truth (a representation of the ideal result of a processing step like OCR or layout analysis) on the level of individual newspaper pages. On

Figure 8 is presented some images that have been used in the experimentation of the work of this thesis.

Figure 8 – PRImA dataset example images



Source: The author (2024)

DIBCO

The DIBCO dataset used in this research is composed of documents from several different libraries across the globe, but mainly from Europe. Its main goal is to provide small cropped portions of document images with the most difficult to filter noises. It has been developed as part of the DIBCO competition series [20]. On Figure 9 is presented some images that have been used in the experimentation of the work of this thesis.

Figure 9 – DIBCO example images



Source: The author (2024)



3.2 Classical Evaluation Methods

Analyzing the quality of the images produced by binarization algorithms is not a trivial task. One of the first methods to evaluate the binarization performance was to count the number of correctly detected digits, as in [15]. In several studies, human perception of individual images was used to evaluate the results of a few images tested [95]. Later, with the increase in computer power and binarization proposals, a more objective evaluation approach was used: to generate a clear human-touched binary image and use it to compare with the results of the algorithms [18]. The comparison is made by applying several statistics between the images.

In this section, the most common evaluation measures are described: DRD, PSNR, F-Measure (FM) and pseudo-FMeasure for scanned documents. The mobile captured measures are left for Chapter 5, where this kind of image is discussed in more detail.

PSNR

The Peak to Noise Signal Ratio, or PSNR, is one of the most popular measures to compare the similarity between two images. It has been extensively used on image processing studies ranging from encryption to document image binarization. It is defined as in (3.1).

$$PSNR = 10 \log \frac{C^2}{MSE} \quad (3.1)$$

where C is the difference between the intensity values of the foreground and background pixels and the MSE is the mean squared error, defined as in (3.2).

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N I'(x, y) - I(x, y) \quad (3.2)$$

where, M and N are the number of columns and rows of the image, while $I(x, y)$ and $I'(x, y)$ are, the value of the pixel (x, y) in the ground truth and the binarized image, respectively.

This measure does not make any difference whether the missed pixels are close or not to letters, which could cause the document readability to decrease. It is not too appropriate for document image binarization, however nearly all studies in this area use it since its first use in the series of document image binarization DIBCO[19].

The smaller the PSNR, the better, but its magnitude is proportional to the size of the images, thus one cannot know how close two images are just by the number, but it is possible

to determine, for instance, which binarization algorithm produced the most similar image when compared to the ground truth.

DRD

The Distance Reciprocal Distortion Measure [114] was developed specially to measure the quality of a binary document image. It aims to measure the distortion of the wrongly mapped pixels in the same way as human perception. It has been noticed that the distance between pixels plays a major role in the perception of distortion. It is defined as in (3.3):

$$DRD = \frac{1}{NUBN(GT)} \sum_{k=1}^S DRD_{ij} \times |B(i, j) - GT(i, j)|, \quad (3.3)$$

where $NUBN(GT)$ is the number of non-uniform 8×8 binary blocks in the ground-truth (GT) image, S is the number of flipped pixels and DRD_{ij} is the distortion of the pixel at position (i, j) in relation to the binary image (B) and is calculated by (3.4):

$$DRD_{ij} = \sum_{x=-2}^2 \sum_{y=-2}^2 W_{xy} \times |B(i+x, j+y) - G(i+x, j+y)|, \quad (3.4)$$

using a 5×5 normalized weight matrix W_{xy} , as defined in [114]. DRD_{ij} equals to the weighted sum of pixels in the 5×5 block of the GT that differs from the centered k th flipped pixel at (x, y) in the image of the binarization result B .

The smaller the DRD, the better.

NRM

The negative rate metric (NRM) is based on the pixel-wise mismatches between the GT and prediction. It combines the false negative rate NR_{FN} and the false positive rate NR_{FP} . It is denoted as follows:

$$NRM = \frac{NR_{FN} + NR_{FP}}{2}, \quad (3.5)$$

where $NR_{FN} = \frac{N_{FN}}{N_{FN} + N_{TP}}$, $NR_{FP} = \frac{N_{FP}}{N_{FP} + N_{TN}}$ and N_{TP} denotes the number of true positives, N_{FP} denotes the number of false positives, N_{TN} denotes the number of true negatives, N_{FN} denotes the number of false negatives.

The lower the NRM the better.

Possibly the first use of this metric was in the series of DIBCO binarization competitions. It has been used in several other studies since then.

MPM

The Misclassification Penalty Metric (MPM) evaluates the prediction against the Ground Truth (GT) on an object-by-object basis. Misclassification pixels are penalized for their distance from the ground-truth object border.

$$MPM = \frac{MP_{FN} + MP_{FP}}{2}, \quad (3.6)$$

where $MP_{FN} = \sum_{i=1}^{N_{FN}} d_{FN}^i D$, $MP_{FP} = \sum_{j=1}^{N_{FP}} d_{FP}^j D$, d_{FN}^i and d_{FP}^j denote the distance of the i^{th} false negative and the j^{th} false positive pixel from the contour of the GT segmentation. The normalization factor D is the sum over all the pixel-to-contour distances of the GT object. A lower the MPM score denotes that the algorithm is good at identifying an object's boundary.

F-Measure

Another widely used measure of "error" in the literature when evaluating the performance of a binary classification task is the F-Measure (FM) [121]. It is a score of classification correctness calculated by considering the *precision* and the *recall*. In the context of binarization algorithms, *precision* is the fraction of correctly mapped text pixels among the pixels mapped as text and *recall* is the proportion of correctly mapped text pixels among all text pixels in the original image and. The FM is calculated as in (3.7).

$$FM = 2 \times \frac{precision \times recall}{recall + precision} \quad (3.7)$$

where *precision* and *recall* are calculated as:

$$precision = \frac{TP}{TP + FP}, \quad recall = \frac{TP}{TP + FN}$$

where TP , FP , FN are, respectively, the true-positive, false-positive and false-negative mapped pixels.

Pseudo-FMeasure (Fps)

Introduced in reference [116], the *pseudo-FMeasure* is an improvement over the *F-Measure*. It uses the same formula to combine the precision and recall measures. However, the distance between the strokes and the contour of the GT is considered applying weights to generate the *pseudo-Recall* and *pseudo-Precision* measures. Those measures are combined as in (3.7) to generate the final value. *Fps* has been used mostly by DIBCO when evaluating the competitors' algorithms, but is rarely seen in other studies.

3.3 New Evaluation Methods

3.3.1 Cohen's Kappa applied to document binarization

Extensively used as a performance measure for classification tasks in remote sensing applications, Cohen's Kappa has recently been used as an evaluation measure of binarization algorithms [22], as it shows a strong correlation of image quality by visual inspection. Furthermore, as indicated by [121], the kappa coefficient is recommended over PSNR and other classical measures when evaluating the performance of binary classifiers.

The Kappa coefficient can be interpreted as a weighted sum of the error (or confusion) matrix of the number of correctly mapped foreground and background pixels, taking the GT image as reference. It compares the observed accuracy with the expected accuracy, an indication of how well a given classifier performs. Cohen's Kappa [122] is defined as:

$$k = \frac{P_O - P_C}{1 - P_C}, \quad (3.8)$$

compares the observed accuracy with an expected accuracy, indicating how well a given classifier performs. P_O is the number of correctly mapped pixels (accuracy) and P_C is calculated by using:

$$P_C = \frac{n_{bf} \times n_{gf} + n_{bb} \times n_{gb}}{N^2}, \quad (3.9)$$

where n_{bf} and n_{bb} are the number of pixels mapped as foreground and background on the binary image, respectively, while n_{gf} and n_{gb} are the number of foreground and background pixels on the GT image and N is the total number of pixels.

The Kappa coefficient has an excellent correspondence with the image-quality perception by human visual inspection of the resulting images. As indicated by Powers [121], κ may be a

good and easy-to-interpret image-quality evaluation measure for binary classifiers [72].

It was first applied as a quality measure for document image binarization as part of this thesis.

The higher the kappa the better.

3.3.2 New Measures for Mobile-Captured Document Images

They are based on the proportions of black pixels in the image and the normalized Levenshtein distance. They are discussed in detail in Chapter 5.

3.4 Processing Time Evaluation

The viability of using a binarization algorithm in a document processing pipeline depends not only on the quality of the final image but also on the processing time elapsed by the algorithm and the maximum amount of memory claimed during the process. To the best knowledge of the authors, the first assessment of binarization algorithms to take into account the average processing time was [22]. Along this thesis, the results of several assessments are presented taking into account the processing time and in this section the details about this measurement is described.

The algorithms assessed were implemented by their authors using several programming languages and operating systems, running on different platforms; thus the processing time figures presented provide **the order of magnitude** of the time elapsed for binarizing the whole dataset. All are mean values of a set of executions. The training times for the AI-based algorithms were not computed. Two processing devices were used:

- **Device 1 (CPU algorithms):** Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, with 32GB RAM and a GPU GeForce GTX 1650 4GB
- **Device 2 (GPU algorithms):** Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz, with 64GB RAM and a GPU NVIDIA GeForce RTX 2080 Ti 12GB

The algorithms were implemented using two operating systems and different programming languages for specific hardware platforms such as GPUs:

- **Device 1, Windows 10 (version 1909), Matlab:** Akbari_1, Akbari_2, Akbari_3, CLD, CNW, ElisaTV, Ergina-Global, Ergina-Local, Gattal, Ghosh, HBUT, Howe, iNICK, Jia-Shi, Lu-Su, Michalak, MO₁, MO₂, MO₃, Yasin;
- **Device 1, Linux Pop!_OS 20.10:** Bataineh, Bernsen, Bradley, Calvo-Zaragoza, daSilva-Lins-Rocha, DiegoPavan, Huang, Intermodes, ISauvola, IsoData, Johannsen-Bille, Kapur-SW, Li-Tam, Mean, Mello-Lins, MinError, Minimum, Moments, Niblack, Nick, Otsu, Percentile, Pun, RenyEntropy, Sauvola, Shanbhag, Singh, Su-Lu, Triangle, Vahid22, WAN, Wolf, Wu-Lu, Yen, YinYang, YinYang21, YinYang22;
- **Device 2, Linux Pop!_OS 22.04:** DE-GAN, DeepOtsu, DilatedUNet, Doc-DLinkNet, Doc-UNet, DPLinkNet, HuangBCD, HuangUnet, Robin, Vahid, Yuleny.

The algorithms were executed on different operating systems (OS), but on the same hardware. For those that could be executed on both OS types, the processing times for each OS was measured, and no significant differences were noticed. This is expected based on previous experiments [31]. It is important to note that each processing time number is a result of a mean of the whole dataset for each case, thus they represent an average on several repeated executions. Finally, as already mentioned, the primary purpose is to provide the order of magnitude time of the processing time elapsed, not an absolute value, providing then an overall idea on how computer intensive is a given binarization method.

3.5 Assessment of Scanned Document Images

Recent proposals on binarization algorithms applied to scanned textual document images focus on historical documents, as modern printed documents offer no significant challenge and even the simplest algorithms can successfully binarize them [5]. Binarization of historical scanned document images is far from a simple task as physical noises [12, 11], such as aging of the paper, stains, fungi, folding marks, etc., and interference from the back to the front [13] increase the complexity of the task. Some recent document binarization competitions [22, 33] show that no single binarization algorithm is efficient for all types of document images.

Their performance depends on a wide number of factors, from the digitalization device, image resolution, the kind of physical noises in the document, the way the document was printed, typed or handwritten, the age of the document, etc. In addition to that, those competitions showed that the time complexity of the algorithms also varies widely, making some of

them impossible to use in any document processing pipeline. Thus, instead of having an overall best, those competitions pointed out the top quality-time algorithms in several categories of documents.

In addition to that, most studies only compare the new algorithms with some older ones [20, 111], while it is important to raise this number to make sure the new proposal is not reinventing the wheel. As a result of this thesis, a new series of document binarization contests was created not only comparing the enrolled participants among themselves, but also comparing the quality-time performance of the new with classical algorithms. In addition to that, other studies usually evaluate the algorithms using a small portion scanned at a high dpi, while here full-sized document images are used.

Five competitions were organized, but only two included scanned document images. In the next sections, the results of the last competition are presented.

3.6 Materials and Methods

This was the third competition of the series, it was the first that focused exclusively on scanned documents. It assessed the performance of 12 new and 49 other previously published binarization algorithms for scanned document images. Four test sets were used, and for each one, the top 20 algorithms in the quality of the resulting binary images had their average processing time presented. Its results and discussion are reproduced here but have been first published at ICDAR 2021. A total of 20 documents from the Nabuco bequest, five from the LiveMemory project and four from the PRImA project (see section 3.1) was used. In Chapter 2 a description of the most important methods is presented.

To evaluate binarization algorithms relative to image quality, the scanned documents were clustered according to their characteristics (print type and paper texture luminosity). This produced five set of documents. The quality of the binary images was compared using the PSNR, DRDM, F-Measure (FM) and pseudo-FMeasure (Fps) [116], and Cohen's Kappa [72, 122]. The final ranking is defined by sorting the ranking summation in ascending order, following the methodology introduced by [20], which is explained in more detail in Section 5.1.5, page 94. The consistency of the global ranking with a carefully performed visual inspection was also checked to ensure consistency.

The top twenty algorithms in image quality, ranked after [20], will have their κ coefficient and standard deviation (shown in parentheses), together with the mean processing time and

its standard deviation (also shown in parentheses) presented in the tables of the results.

The evaluation of the processing time followed the protocol defined in Section 3.4. The training times for the AI-based algorithms were not considered. The 12 competing algorithms were implemented using different programming languages and operating systems, and even for specific hardware platforms such as GPUs. They are compared against the other 49 algorithms in the literature, most of which were implemented by their authors or are available in image processing environments such as MatLab or ImageJ, but many are also exclusive to this assessment, as their code were shared with us directly by the authors.

From the Nabuco bequest of historical documents from the late XIX century, 20 images were selected, which were subdivided into three clusters according to the average luminosity level of the background texture. Dark textures have an average luminosity of 147, a mid texture of 193, and a light texture of 220. A total of seven dark, seven light texture handwritten, and six mid-dark texture typewritten documents were selected. From the LiveMemory project, five images with various configurations were selected. From the PRImA project, four images that belong to the Europeana Newspapers Project dataset were used. The images were selected in order to provide some variability between the datasets, but similar images within the datasets. The chosen datasets are representative of a large number of "real-world" documents of interest.

The ground truth images used here were obtained by binarizing the original images with the ten best quality algorithms from previous competitions [22, 31] in images similar to the ones chosen for this competition. Such images were subjected to a careful visual inspection. The three best binary images were merged by applying the AND logical operator. The resulting image was subjected to salt and pepper filtering. The resulting image was visually reinspected and underwent a manual cleaning.

In order to understand how the algorithms would perform with standard datasets, the DIBCO dataset has been chosen and tested. Once the images vary significantly in shape, resolution and type, it is impractical to do a detailed analysis as for our dataset. Given the large variation in resolution, the processing time vary too much and thus only the quality was measured. Also, most algorithms are trained or their parameters are fine tuned with one of those datasets and thus the comparison between them is not necessarily fair. The results were included for completeness and to understand the algorithms behavior in such scenario.

3.7 Results and Discussion

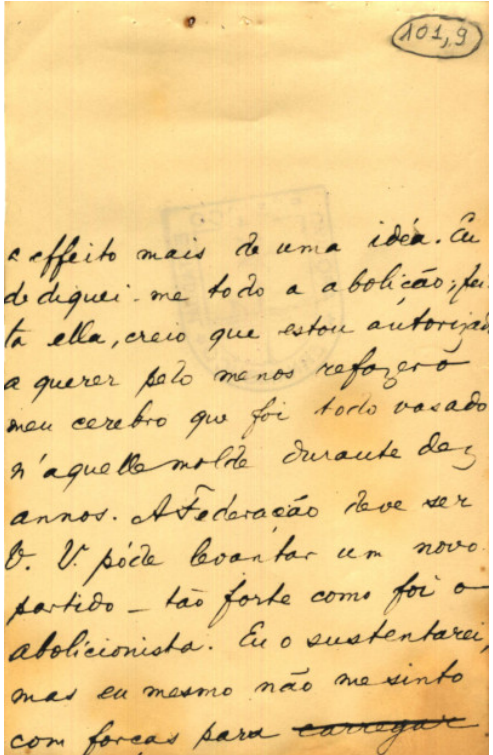
This analysis was thought to look at the trade-off between binarization performance and computational time. There is no single best algorithm. The 20 best performing algorithms are reported by dataset in Tables 4-8. Yasin and HuangBCD appeared in the top ranked algorithms for all five datasets, and the sister algorithm HuangUnet appeared in the top ranked for four of the datasets. Michalak21's first and third algorithms appeared three times in the rankings. YinYang21 appeared 3 times and Vahid appeared twice.

The average kappa values for the top 20 reported for each dataset fell in a narrow range from 0.75 to 0.94. The binarized images produced using the best quality algorithm for the test images, as one may expect, had very high visual quality. The 10 best quality images for each of the sample images were made available on the DIB website (<https://dib.cin.ufpe.br/>).

The execution times varied more significantly than the performance as measured by the kappa value. The median run time of the best performing algorithms was 1 second, with 21% of the algorithms taking less than 0.1 seconds. Michalak21_a was the fastest of the new competing algorithms in this year. Nine of the algorithms took more than a minute on average to process the page, which for most applications will not be practical due to the small performance benefit the algorithm may offer. HuangBCD, which appeared in the top rankings for all datasets, was also the algorithm that had the longest run time of all the algorithms ranked. The median run time for the algorithms published before 2010 was 0.11 seconds, while the median of the algorithms published 2010-2019 increased to 4.95 seconds and the median of those published in 2020 and 2021 is 7.39. Performance does not vary significantly between these groups.

In Table 9 it is presented the results for the DIBCO evaluation. The newer machine-learning algorithms outperform all the others in all cases, as they have been fine-tuned for binarizing DIBCO images. Even on this scenario, the global algorithm Li-Tam still appeared among the best (see DIBCO 2016 results). When comparing with the full documents dataset tests, it comes to be clear that in order to have good results with modern algorithms, it has to be retrained for each new dataset, however either the traditional algorithms or theory-based ones can be used in both situations.

Table 4 – Quality-time Results for Nabuco, Light Texture, Handwritten Documents

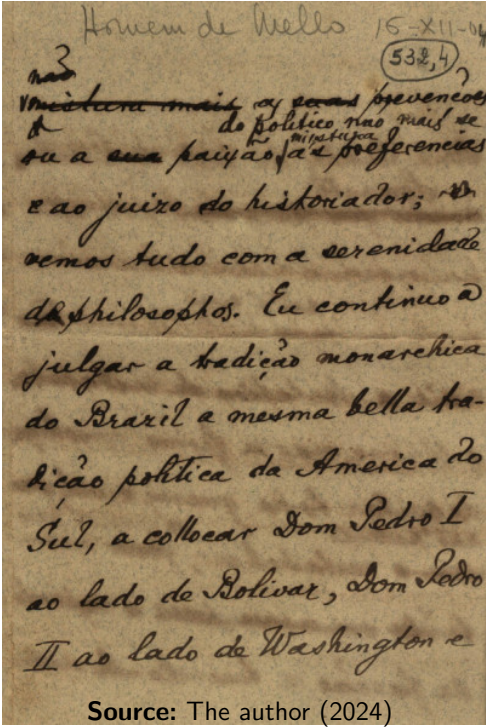
#	Team	Kappa (SD)	Time (SD)	Example Image
1	Vahid	0.89 (0.06)	10.18 (4.49)	
2	HuangUnet	0.87 (0.13)	24.91 (7.91)	
3	Akbari_1 [63]	0.84 (0.21)	4.91 (1.98)	
4	HuangBCD	0.87 (0.10)	113.29 (35.16)	
5	Akbari_2 [63]	0.84 (0.21)	4.95 (2.12)	
6	Akbari_3 [63]	0.84 (0.21)	4.89 (1.99)	
7	Jia-Shi [62]	0.84 (0.21)	4.87 (1.99)	
8	Wolf [69]	0.86 (0.05)	0.06 (0.03)	
9	Sauvola [95]	0.86 (0.06)	0.04 (0.02)	
10	DocDLink [22]	0.81 (0.18)	55.60 (26.86)	
11	Yasin	0.83 (0.10)	1.18 (0.99)	
12	Gosh [80]	0.81 (0.15)	31.84 (16.58)	
13	Su-Lu [64]	0.85 (0.06)	0.41 (0.18)	
14	Lu-Su [105]	0.81 (0.12)	16.15 (7.06)	
15	Minimum [101]	0.84 (0.10)	0.01 (0.01)	
16	iNICK [50]	0.81 (0.11)	5.32 (4.09)	
17	DilatedUNet [31]	0.80 (0.12)	44.43 (15.47)	
18	Intermodes [101]	0.80 (0.11)	0.01 (0.00)	
19	Mello-Lins [14]	0.79 (0.21)	0.01 (0.00)	
20	ElisaTV [104]	0.76 (0.20)	2.41 (1.06)	

Source: The author (2024)

3.8 Conclusions

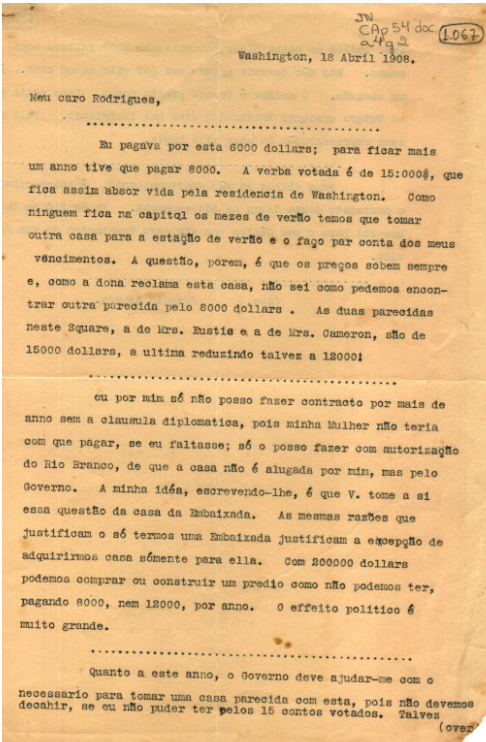
This analysis shows that document image binarization is still a challenging task. The number of ways the problem can be made more difficult leads to demand to develop a new algorithm that can handle that one outlier case that others could not properly binarize. Machine-learning binarization algorithms are rising in providing better quality images, but some of the classic algorithms like IsoData [86] and Savoula [95] continued to appear in the top ranked algorithm list and they still provide very good, if not the best quality bitonal image at a much lower time complexity. It is important to remark that the training-time for the machine-learning based algorithms was not computed. Another point worth remarking is that some of those ML algorithms require computational resources that may be considered prohibitive, as some of the competing algorithms in the ICDAR 2019 Competition on Time-Quality Document Image Binarization [22] were unable to run to all test images of the test sets used here.

Table 5 – Quality-time Results for Nabuco, Dark Texture, Handwritten Documents

#	Team	Kappa (SD)	Time (SD)	Example Image
1	Sauvola [95]	0.91 (0.04)	0.03 (0.00)	
2	Gosh [80]	0.89 (0.03)	20.97 (2.09)	
3	Wolf [69]	0.89 (0.03)	0.04 (0.00)	
4	DocDLink [22]	0.88 (0.05)	42.12 (2.31)	
5	HuangBCD	0.89 (0.02)	89.30 (7.21)	
6	Su-Lu [64]	0.90 (0.06)	0.32 (0.04)	
7	HuangUnet	0.89 (0.03)	19.81 (1.54)	
8	Yasin	0.89 (0.04)	0.82 (0.24)	
9	iNICK [50]	0.89 (0.03)	3.19 (0.51)	
10	Nick [47]	0.89 (0.03)	0.03 (0.00)	
11	Singh [107]	0.89 (0.03)	0.03 (0.00)	
12	YinYang21	0.86 (0.07)	0.51 (0.07)	
13	DocUNet [22]	0.85 (0.07)	37.33 (4.53)	
14	Li-Tam [96]	0.86 (0.05)	0.01 (0.00)	
15	Vahid	0.86 (0.06)	7.39 (0.49)	
16	Shanbhag [91]	0.85 (0.10)	0.01 (0.00)	
17	Howe [48]	0.85 (0.07)	15.59 (7.72)	
18	DilatedUNet [31]	0.85 (0.07)	31.96 (3.14)	
19	Ergina_L [100]	0.86 (0.07)	0.12 (0.02)	
20	Ergina_G [99]	0.85 (0.08)	0.08 (0.01)	

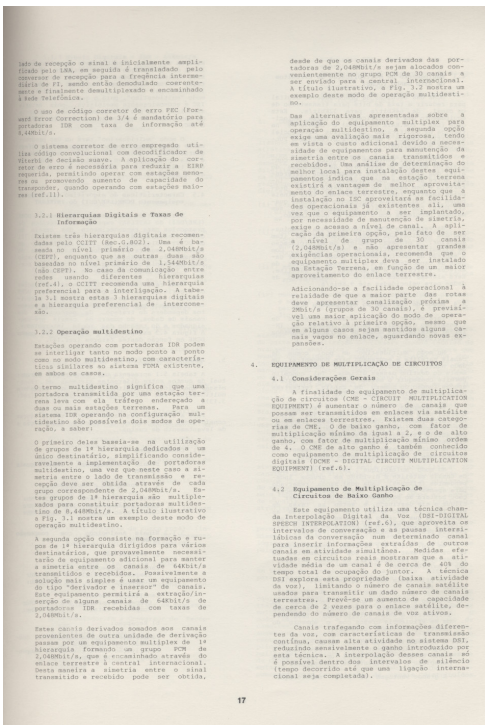
Source: The author (2024)

Table 6 – Quality-time Results for Nabuco, Mid Texture, Typewritten Documents

#	Team	Kappa (SD)	Time (SD)	Example Image
1	Gosh [80]	0.92 (0.07)	51.82 (6.28)	
2	HuangUnet	0.91 (0.05)	37.67 (1.81)	
3	Yasin	0.90 (0.06)	1.03 (0.14)	
4	HuangBCD	0.91 (0.04)	167.59 (7.49)	
5	iNICK [50]	0.89 (0.07)	3.70 (0.52)	
6	Wolf [69]	0.92 (0.03)	0.10 (0.01)	
7	Singh [107]	0.92 (0.04)	0.13 (0.01)	
8	Michalak21a	0.87 (0.10)	0.02 (0.00)	
9	Li-Tam [96]	0.88 (0.08)	0.02 (0.00)	
10	Minimum [101]	0.90 (0.03)	0.02 (0.00)	
11	Nick [47]	0.91 (0.05)	0.08 (0.00)	
12	Su-Lu [64]	0.91 (0.02)	0.71 (0.07)	
13	Intermodes [101]	0.87 (0.06)	0.02 (0.00)	
14	Michalak21c	0.85 (0.10)	0.47 (0.04)	
15	ElisaTV [104]	0.86 (0.08)	4.27 (0.20)	
16	Akbari_1 [63]	0.86 (0.06)	8.45 (0.85)	
17	Akbari_2 [63]	0.86 (0.06)	8.45 (0.87)	
18	Bradley [103]	0.84 (0.09)	0.14 (0.01)	
19	Akbari_3 [63]	0.86 (0.06)	8.46 (0.87)	
20	Jia-Shi [62]	0.86 (0.06)	8.46 (0.88)	

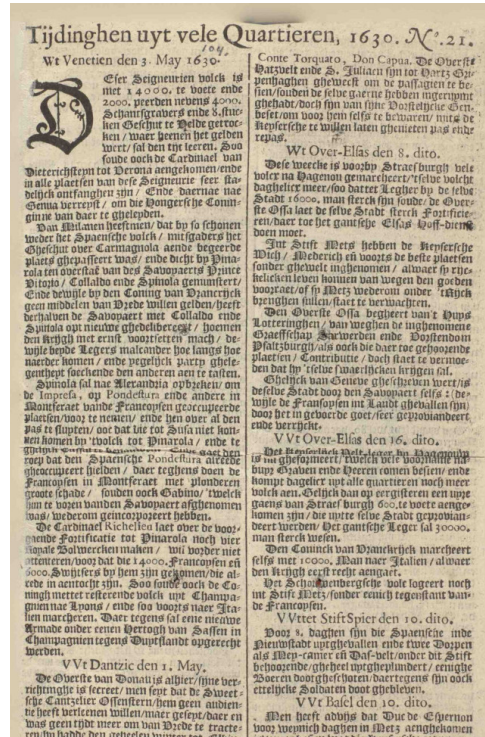
Source: The author (2024)

Table 7 – Quality-time Results for LiveMemory Test Set

#	Team	Kappa (SD)	Time (SD)	Example Image
1	Michalak [31]	0.94 (0.04)	0.08 (0.05)	
2	Bradley [103]	0.94 (0.05)	0.29 (0.01)	
3	Wolf [69]	0.94 (0.05)	0.22 (0.02)	
4	ElisaTV [104]	0.93 (0.06)	9.55 (1.15)	
5	Gosh [80]	0.94 (0.03)	111.80 (21.29)	
6	IsoData [86]	0.90 (0.12)	0.14 (0.02)	
7	Gattal [108]	0.91 (0.11)	54.40 (1.48)	
8	Otsu [2]	0.90 (0.13)	0.02 (0.00)	
9	Li-Tam [96]	0.91 (0.10)	0.14 (0.01)	
10	Yasin	0.93 (0.05)	2.05 (0.99)	
11	iNICK [50]	0.93 (0.03)	3.48 (0.35)	
12	Michalak21 _a	0.94 (0.03)	0.08 (0.05)	
13	Intermodes [101]	0.92 (0.07)	0.14 (0.02)	
14	Michalak21 _c	0.92 (0.06)	1.32 (0.67)	
15	Johannsen [87]	0.92 (0.05)	0.14 (0.02)	
16	Su-Lu [64]	0.93 (0.02)	1.67 (0.10)	
17	YinYang21	0.91 (0.07)	1.60 (0.13)	
18	HuangBCD	0.92 (0.07)	316.87 (25.66)	
19	HuangUnet	0.92 (0.07)	316.78 (26.17)	
20	WAN [109]	0.92 (0.07)	1.01 (0.09)	

Source: The author (2024)

Table 8 – Quality-time Results for PRLmA Data Set

#	Team	Kappa (SD)	Time (SD)	Example Image (cropped)
1	Gosh [80]	0.90 (0.09)	159.77 (92.16)	
2	Bradley [103]	0.90 (0.08)	0.43 (0.35)	
3	Michalak21 _a	0.89 (0.08)	0.10 (0.06)	
4	Intermodes [101]	0.89 (0.14)	0.19 (0.23)	
5	Michalak [31]	0.91 (0.08)	0.10 (0.06)	
6	Li-Tam [96]	0.87 (0.17)	0.19 (0.23)	
7	DocDLink [22]	0.92 (0.06)	292.46 (223.60)	
8	ElisaTV [104]	0.88 (0.04)	13.56 (10.63)	
9	IsoData [86]	0.87 (0.14)	0.19 (0.23)	
10	Su-Lu [64]	0.87 (0.10)	2.93 (1.95)	
11	Moments [88]	0.85 (0.16)	0.19 (0.23)	
12	Michalak21 _c	0.89 (0.05)	1.83 (1.05)	
13	Yasin	0.87 (0.08)	2.42 (1.24)	
14	Ergina_L [100]	0.87 (0.10)	1.28 (0.68)	
15	Gattal [108]	0.87 (0.13)	56.94 (3.80)	
16	Akbari_1 [63]	0.86 (0.06)	32.40 (20.71)	
17	Ergina_G [99]	0.86 (0.14)	0.85 (0.62)	
18	Huang [92]	0.86 (0.10)	0.19 (0.22)	
19	Akbari_2 [63]	0.86 (0.06)	32.39 (20.68)	
20	HuangBCD	0.86 (0.08)	445.08 (301.37)	

Source: The author (2024)

Table 9 – Results of binarizing DIBCO dataset

DIBCO 2011			DIBCO 2012		DIBCO 2013	
#	Team	Kappa (SD)	Team	Kappa (SD)	Team	Kappa (SD)
1	Vahid22	0.96 (0.01)	DilatedUNet	0.97 (0.01)	Vahid22	0.97 (0.01)
2	Vahid	0.96 (0.01)	Vahid22	0.97 (0.01)	DilatedUNet	0.97 (0.01)
3	DilatedUNet	0.96 (0.01)	Vahid	0.97 (0.01)	Vahid	0.97 (0.01)
4	DPLinkNet	0.96 (0.01)	DPLinkNet	0.97 (0.01)	DPLinkNet	0.97 (0.01)
5	Calvo-Zaragoza	0.95 (0.02)	Calvo-Zaragoza	0.96 (0.01)	Calvo-Zaragoza	0.96 (0.01)
6	Jia-Shi	0.91 (0.03)	robin	0.95 (0.01)	robin	0.94 (0.02)
7	Huali	0.90 (0.02)	Jia-Shi	0.92 (0.02)	Jia-Shi	0.93 (0.02)
8	robin	0.90 (0.06)	CLD	0.86 (0.11)	Huali	0.81 (0.24)
9	CLD	0.83 (0.09)	Huali	0.75 (0.35)	CLD	0.85 (0.07)
10	CNW	0.81 (0.12)	ISauvola	0.83 (0.13)	Michalak	0.86 (0.07)

#	DIBCO 2014		DIBCO 2016		DIBCO 2017	
1	DilatedUNet	0.98 (0.00)	Vahid	0.93 (0.01)	DPLinkNet	0.95 (0.01)
2	Vahid	0.97 (0.01)	DPLinkNet	0.93 (0.02)	DilatedUNet	0.95 (0.02)
3	Vahid22	0.97 (0.01)	Vahid22	0.93 (0.01)	robin	0.91 (0.03)
4	DPLinkNet	0.97 (0.01)	DilatedUNet	0.89 (0.02)	Vahid22	0.92 (0.03)
5	Calvo-Zaragoza	0.97 (0.01)	Michalak	0.85 (0.06)	Vahid	0.91 (0.04)
6	robin	0.96 (0.01)	Michalak21a	0.85 (0.06)	Calvo-Zaragoza	0.84 (0.11)
7	Jia-Shi	0.94 (0.02)	CLD	0.85 (0.03)	Jia-Shi	0.84 (0.13)
8	WAN	0.87 (0.18)	robin	0.85 (0.02)	CLD	0.81 (0.09)
9	Huali	0.86 (0.19)	KSW	0.84 (0.04)	Huali	0.70 (0.29)
10	CNW	0.88 (0.13)	Li-Tam	0.81 (0.10)	Michalak21a	0.81 (0.08)

Source: The author (2024)

Figure 10 – DIBCO Dataset Example Images (Small)



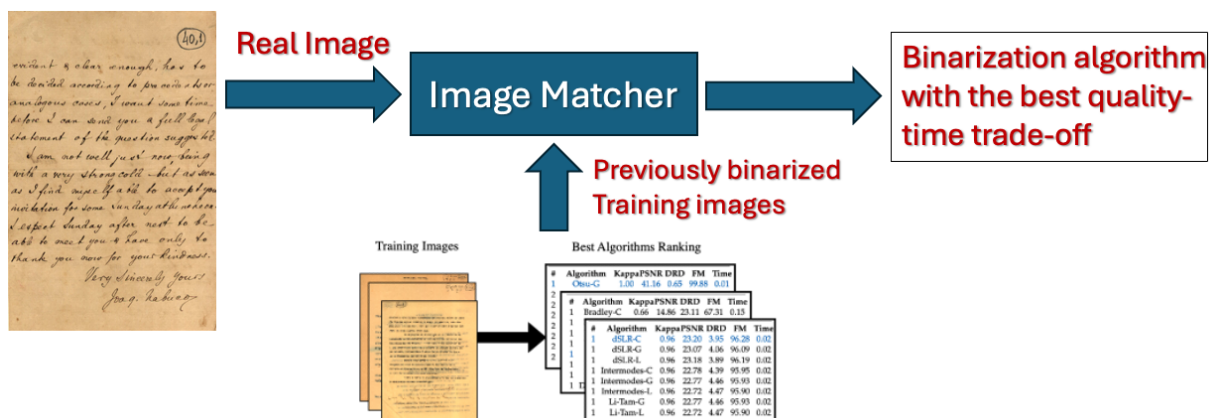
Source: The author (2024)

4 TEXTURE BASED BINARIZATION

As previously discussed, image binarization is a crucial step in converting physical documents into digital, editable formats, archiving them in databases, or preparing them for printing. However, no single binarization algorithm is universally effective for all types of document images, as evidenced by recent Quality-Time Binarization Competitions [22, 33, 31, 34, 56] and highlighted in the latest literature reviews [5]. The quality of the resulting binary image is influenced by numerous factors, including the digitization device and its setup, as well as the document's intrinsic properties, such as paper color, texture, and the method of handwriting or printing. Additionally, the time required for binarization varies significantly between algorithms, depending on the document's characteristics. This raises a fundamental question: if document features are key determinants of binary image quality, and there is substantial variability in time-performance across algorithms, how can one select the optimal algorithm to achieve the best quality-time trade-off for diverse and heterogeneous documents?

A solution to this issue would be the development of a “image matcher”, which compares a given input image with a large set of previously binarized images to find the most similar one. Once the previously binarized image has similar features and has already been tested with dozens of algorithms, the most recommended binarization algorithm for the input image can be inferred from that. This idea first appeared in a paper by Lins et. al. [28], which originated the DIB platform (<https://dib.cin.ufpe.br>) and is part of a previous research by the author of this thesis too. The final goal of such platform is to use the more than 5.5 million synthetic documents as reference to the image matcher and it's idea is depicted in Figure 11.

Figure 11 – DIB image matcher.



Source: The author (2024)

As a first attempt to implement such image matcher, the paper texture has been chosen as main feature from the document and used as reference to compare between images. Most of the document image is made of background paper texture, and several authors [123, 124] show that texture analysis plays an important role in document image processing. Barboza et al. [125], show that the analysis of paper texture may be used to determine the age of documents for forensic purposes, avoiding document forgeries. Alaei et al. [126] used twenty-six different texture feature extraction methods, divided into nine sets, to find the best way to segment the document image according to the region to which it belongs.

It is fundamental that the image matcher is a very lightweight process not to overload the binarization processing time, thus only fast feature extraction methods have been chosen. Texture extraction was initially performed manually [35], but later an automatic feature extraction method was developed and used to submit a complete binarization solution to the recent DocEng 2024 Binarization Competition. Furthermore, initially only the EFOS feature (described in the next sections) has been used [35], but recently it was expanded to use 12 other manually picked feature descriptors and this work has already been accepted for publication at DocEng'24 Conference [38]. In the end, EFOS did prove to be the overall most recommended, but if the image features is known beforehand and provided to the algorithm, other features are rated as more effective.

This chapter describes the whole process in detail, including the intermediate steps to determine the parameters and algorithms developed, implemented, and used. The dataset is composed of 39 images selected from the Nabuco bequest and is described in Section 3.1. The evaluation measures used are common in the document analysis community and have been described in Chapter 3. Specifically for this part of the study, the Cohen's Kappa, PSNR, DRD and F-Measure have been used along with the processing time.

4.1 Texture Descriptors

Texture descriptors are widely applied to document image retrieval applications, as they allow to properly identify the overall appearance of documents, specially the background texture, which often have repeated patterns. As shown on [127], there are two main approaches to take: theory-driven, where mathematical formulations are applied in order to derive a general rule on identifying patterns, and data-driven approaches, based on deep learning, which highly depend on the training data and require large and diverse datasets.

The first studies on this area began as early as 1973 and even with recent advances, classical methods can still be applied very effectively in many cases.

Mehri et. al. [128] applied several different texture features based on theory to measure the effectiveness of them on the context of document image retrieval systems. Nine sets of features were used, which, from each, several variations were derived. A dataset of 1000 real scanned historical document images was collected from many sources and categorized as containing graphics, text, one or two fonts of writing or typing. Performance analysis was also conducted.

Alaei et. al. [126] assessed twenty-six different texture feature extraction methods when applied to document image retrieval. Three document image datasets were used, and the goal was to identify, for example, whether the document was a newspaper article or a magazine sheet. Most of the features were implemented by [129], which proposed the “histogram of equivalent patterns” as a new way of generating the feature textures, by converting the output matrices into a single vector with all the dimensions of the matrix.

For the purpose of this study, 12 feature descriptors have been chosen based on previous research on this area, by Alaei, Mehri and Bianconi [130, 126, 128, 127]. The most prominent features and distances were selected and are briefly described in this section and a summary of them is presented on Table 10

First Order Statistics – FOS

The mean, standard deviation, median, mode, and several other first-order statistics (FOS) measures are grouped in a vector, where the grayscale image is used as input. The feature vector comprises a set of essential statistical measures (mean, standard deviation, median, mode, skewness, kurtosis), texture attributes (energy, entropy), extrema statistics (minimum and maximum gray level), variability assessment (coefficient of variation), percentiles (10th, 25th, 50th, 75th, 90th), and a metric for histogram width. These elements collectively provide a comprehensive representation of the data’s statistical, structural, and distributional characteristics, making it valuable for various analytical and pattern recognition tasks.

Table 10 – Texture features used in this study

Texture Descriptor	Description
First Order Statistics – FOS	A set of essential statistical measures (mean, standard deviation, median, mode, skewness, kurtosis), texture attributes (energy, entropy), extrema statistics, variability assessment, percentiles, and a metric for histogram width
Expanded First Order Statistics – EFOS	The mean, standard deviation, median, mode, minimum, maximum and kurtosis of the RGB channels of the image
HEP Gray Level Co-occurrence Matrix – GLCM	It records how often pairs of pixel values at specified offsets occur together within an image. Concatenates the matrix rows into a single array
GLCM range	Calculates several features of the GLCM matrix and uses the ranges of each one as a feature vector
GLCM mean	Calculates the means of the features extracted from the matrix
Local Binary Patterns – LBP	Compares the gray-level values of a central pixel to those of its neighboring pixels; converts these comparisons into binary codes and constructs a histogram with it
Improved Local Binary Patterns – ILBP	An extension of the LBP using circular patterns to enhance the discriminative power and robustness
Improved Binary Gradient Contours – IBGC	An extension of the Binary gradient contours (BGC) [131]. It includes the central pixel and can be easily derived from the original formulation by comparing the central pixel value with the average grey-scale value.
Statistical Feature Matrix – SFM	Is constructed from a combination of four statistical attributes: coarseness, contrast, periodicity and roughness
Gray Level Texture Co-occurrence Spectrum – GLTC+	Variation of the GLTC spectrum (GLCTS+) [129] evaluating the likelihood of different arrangements occurring when pixels within a specified neighborhood.
Gray Level Difference Statistics – GLDS	Calculates the differences between pairs of graylevel pixels
Neighborhood Gray Tone Difference Matrix – NGTDM	Correspond to the visual properties of the texture, calculating the coarseness, contrast, busyness, complexity and strength.

Source: The author (2024)

Expanded First Order Statistics – EFOS

The first order statistics are expanded (EFOS) to include the other versions of the images (RGB channels), but with fewer measures: mean, standard deviation, median, mode, minimum, maximum, and kurtosis.

Histogram of Equivalent Patterns – HEP

Also referred to as HEP, the Histogram of Equivalent Patterns defines a class of texture descriptors which partition the pattern space into classes of equivalent patterns. A histogram of found patterns is created, and the histogram bins of equivalent patterns are merged. Several texture descriptor methods generate matrices and are converted to the HEP format by concatenating the rows into a single vector.

Gray Level Co-occurrence Matrix – GLCM

A Gray Level Co-occurrence Matrix (GLCM), as proposed by Haralick, is a quantitative representation of the spatial relationship between pixel values in a grayscale image. It records how often pairs of pixel values at specified offsets occur together within an image. The following features are computed: angular second moment, contrast, correlation, sum of squares: variance, inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, information measures of correlation. For each feature, the range and mean are used as feature generating two feature vectors: GLCM range and GLCM mean.

Local Binary Patterns – LBP

It characterizes textures by comparing the gray-level values of a central pixel to those of its neighboring pixels within a specified local neighborhood. LBP converts these comparisons into binary codes, creating a unique binary pattern for each pixel. These patterns are then used to construct histograms that capture the distribution of texture features in the image.

Improved Local Binary Patterns – ILBP

The Improved Local Binary Pattern (ILBP) is an extension of the Local Binary Pattern (LBP) and is designed to enhance the discriminative power and robustness of LBP for texture classification tasks. It works by comparing the intensity value of a central pixel to the values of its neighboring pixels in a circular pattern.

Improved Binary Gradient Contours – IBGC

An extension of the binary gradient contours (BGC). It includes the central pixel and can be easily derived from the original formulation by comparing the central pixel value with the average grayscale value. It was proposed by Antonio Fernández [129].

Statistical Feature Matrix – SFM

The Statistical Feature Matrix (SFM) is composed of coarseness, contrast, periodicity, and roughness. It is constructed from a combination of four statistical attributes: coarseness, contrast, periodicity, and roughness. These elements collectively capture various aspects of the texture and patterns of the data, enabling a holistic representation suitable for tasks involving texture analysis, image processing, and feature extraction.

Gray Level Texture Co-occurrence Spectrum – GLTC+

This method is a variation of the gray-level texture co-occurrence spectrum (GLCTS+) [129] is employed to analyze texture features. This technique is based on evaluating the likelihood of different arrangements occurring when pixels within a specified neighborhood are sorted according to their grayscale intensities in descending order. The neighborhood is defined by its size and shape, and the total number of these arrangements corresponds to the number of possible permutations.

4.2 Materials and Methods

The used dataset is an excerpt of Nabuco bequest images (Section 3.1) which have ground-truth and offer a wide enough variety to do a proper evaluation. It is composed of 39 images representative of the whole dataset, with dark and light background textures, handwritten and typewritten text, stain, folding marks, smudges, and several levels of back-to-front interference. The complete dataset of the images tested has already been presented at Section 3.1, on Figure 4. They have been binarized with 63 binarization algorithms and fed five versions of the input image, totaling 315 different binarization schemes. From the Table 1, all algorithms except the ones marked have been used in this part of the study.

Twelve different texture descriptors were used (as described in Section 4.1). The FOS and EFOS are basic statistical measures. The HEP was applied to generate the HEP versions of GLCM, ILBP, IBGC and GLTC+. The default implementation of the GLCM, GLDS, NGTM and SFM features was extracted using the PyFeats library¹. The default parameters were used for both the binarization algorithms and texture feature extraction methods.

Given that real-world full-sized historical images with binarization ground-truth are rare and only 39 images are available for testing, the Leave-One-Out Cross-Validation (LOOCV) method is used. Each image is extracted from the original dataset and the rest of the dataset is used as training images. The texture in the training set with the smallest distance from the test image is chosen and its source document image is used to determine the best binarization algorithm.

4.3 Direct Binarization

As highlighted by Tensmeyer and Martinez [5] “nearly all (binarization) methods apply a grayscale conversion as an initial step in order to convert the RGB image representation into a single channel version.” The classical color into grayscale conversion algorithm gets the RGB components of a given image as input and apply equation 4.1 to get the value of the equivalent hue of gray, or the equivalent luminance:

$$L(C) = 0.176R + 0.81G + 0.011B. \quad (4.1)$$

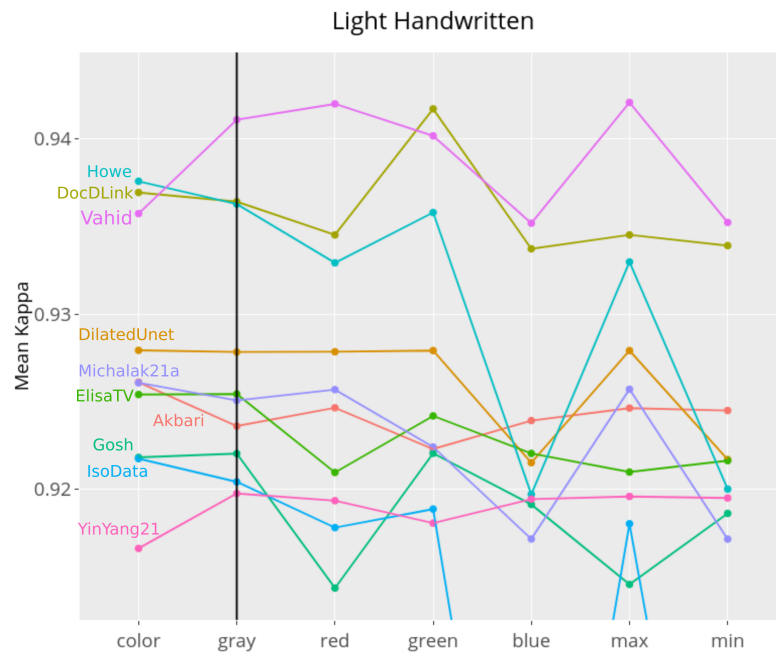
¹ <https://pypi.org/project/pyfeats/>

The larger the size of the document image and its resolution, the more computationally intensive it becomes.

In [32] (which was part of the current work), it has been shown that the direct binarization of one of the RGB channels may yield two-tone images as good or better than the binarization of the color image, saving the processing time of grayscale conversion. However, in some cases of ML algorithms, grayscale conversion as a pre-processing step may improve the quality of the monochromatic image.

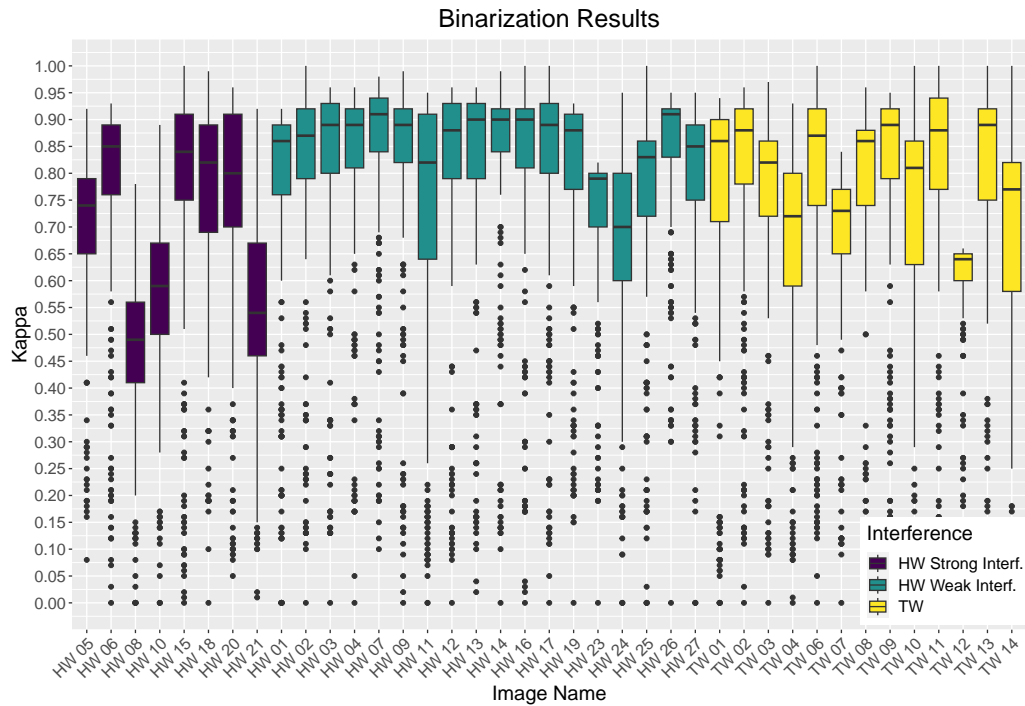
An excerpt of the results is shown in Figure 12, where one can see that DocDLink, which is a deep learning-based algorithm, had slightly better results with only the green channel, while Vahid had slightly better results with the grayscale version than with the original color image. Then, to expand the possibilities to get better results, in this research, each combination of algorithm and input version is considered a whole new algorithm, thus “Wolf-R” stands for applying the Wolf algorithm with the red channel of the input image. The image versions are: “R”, “G”, “B”, “C”, “L”, which correspond, respectively, to the red, green, blue channel, original color image, and converted luminance grayscale image.

Figure 12 – Direct binarization example



Source: The author (2024)

Figure 13 – Binarization results summary



Source: The author (2024)

4.4 Binarization Results

Figure 13 presents the results obtained for all binarization methods. This extensive exploration provides a comprehensive overview of how different algorithms perform under varying conditions of back-to-back interference and text type. Each data-point represent the quality result of a binarization scheme (algorithm + channel). Values under 0.85 are generally too noisy and means a bad binarization result. As expected, the results indicate that handwritten documents with strong back-to-front interference present a wide variance in the performance of different methods, reflecting the difficulty in achieving consistent binarization results. Notably for HW 08, 10, and 21, the best binarization algorithms are below 0.9 and sometimes even below 0.8, which means that none of the tested binarization algorithms could fully binarize them. In contrast, handwritten documents with weak interference generally perform better, with most exhibiting medians above 0.85, except HW 23, 24 and 25. The performance on typewritten documents shows a broader variance, indicating that the nature of the text—whether typewritten or handwritten—along with the interference level, plays a crucial role in determining the success of the binarization process. While the results are mixed, this variability underscores the need for a nuanced approach in selecting binarization algorithms,

taking into account not only the type of document, but also the specific characteristics of the interference and the document's intrinsic properties.

The best binarization scheme for each image is the one at the upper end of the upper segment of the boxplot. For each image, the top algorithm is not always the same and the goal of the texture-based image matcher is, for a given input image, to predict either the top algorithm or another one that produces equally high quality binary images.

4.5 Texture Matching

In summary, the texture matching process consists of:

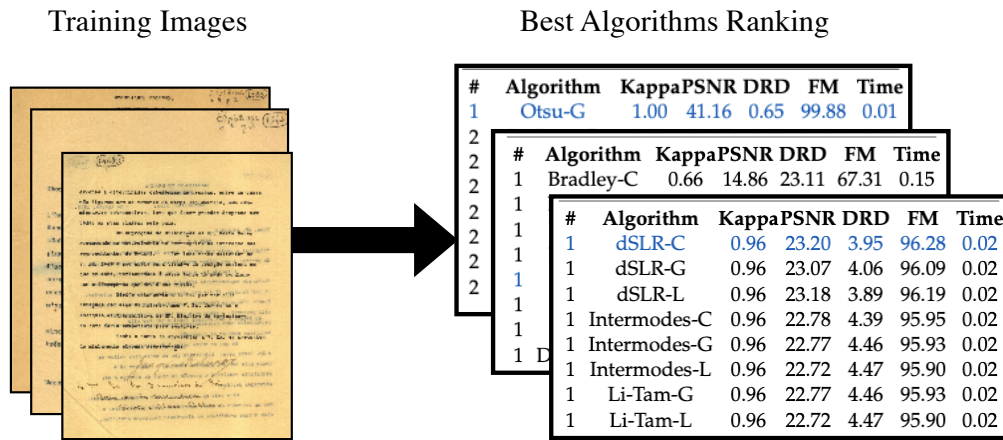
- Step 1** A set of training images is selected that is representative of historical documents and all binarization approaches are applied to each of them. The algorithms are ranked in terms of visual quality and processing time using the Kappa as a measure of quality;
- Step 2** The input image is compared with all the training set utilizing a portion of the background paper texture, which was initially extracted manually. The 12 different texture descriptors are tested with 3 distance measure in order to find the best combination for each case and it is used to find the best match;
- Step 3** The algorithm ranking for the matched image is recovered from the previously binarized results of Step 1;
- Step 4** The input (target) image is binarized with the found algorithm at Step 3;

Several tests were performed to determine the best feature descriptor and distance measure. In the next sessions, a detailed explanation of each step is presented.

4.5.1 Matching Process

Given an input image, the goal is to find the most similar image in the training set to apply the image matching process and find the most recommended binarization algorithm. So, before applying any image matching, one needs to binarize all the training images with all the available binarization algorithms (Figure 14).

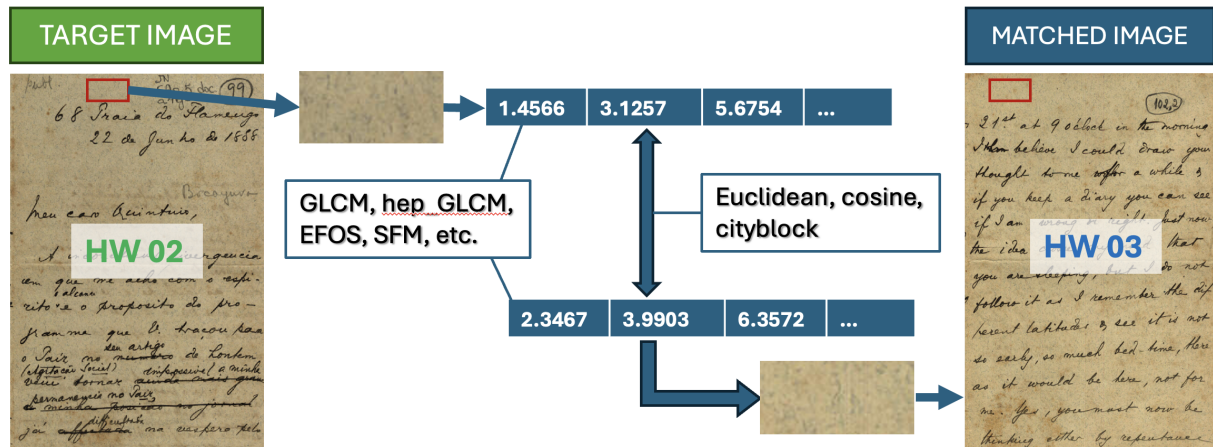
Figure 14 – **Texture Matcher Step 1:** Binarize all training images with each algorithm and rank to find the best ones.



Source: The author (2024)

The comparison (illustrated on Figure 15), consists of first extracting a portion of the background textures from both images, calculating the texture features utilizing one of the descriptors and applying a distance measure.

Figure 15 – **Texture Matcher Step 2:** Compare the input image paper texture with each training image to find the most similar.

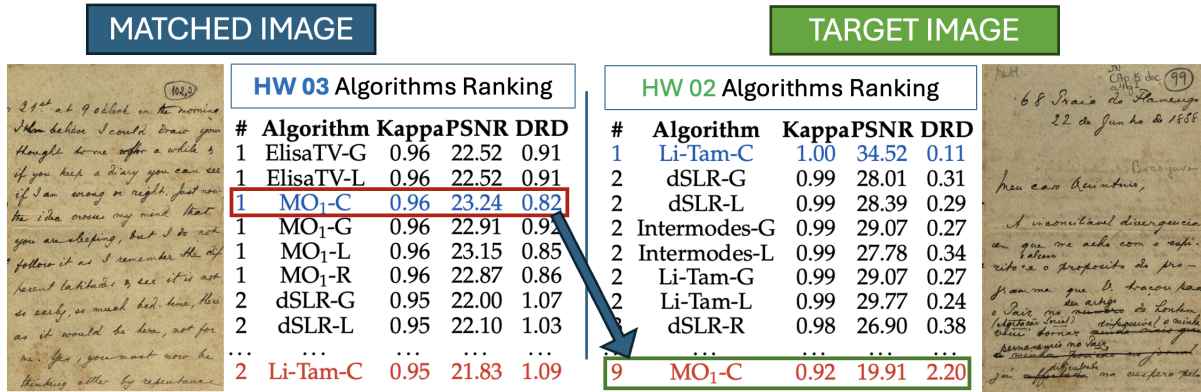


Source: The author (2024)

Once the algorithms ranking is found and the most similar image is determined, the best quality-time algorithm from the matched image is used to binarize the target image (Figure 16). Considering that the ground-truth of the target image is not known, it would be impractical to test on dozens of algorithms in order to find the one who can better binarize it. Note that in this example, the best algorithm for the matched image, MO₁ with the color channel, performed as 9th on the target image ranking, however, the Kappa is still above 0.9, which

means the final binary image has few noise remaining. The goal is not necessarily to find the top-1, but an algorithm that performs sufficiently well to be considered acceptable, with readable text and only small to imperceptible noise artifacts.

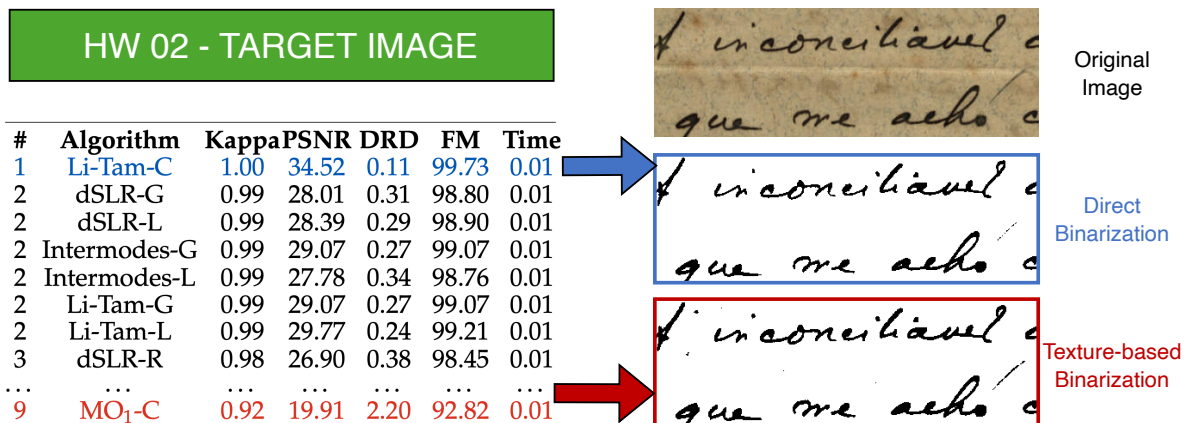
Figure 16 – **Texture Matcher Step 3:** Find the most recommended algorithm for the input image.



Source: The author (2024)

An example of binarization by this method is shown in Figure 17. If the ground truth is known and the most recommended algorithm is found, Li-Tam with the color channel would be recommended. Checking the binary output confirms that this is a good option. Now, if the ground truth is not known, binarizing with the texture-based image matcher recommended algorithm, which is MO₁ with the color channel, even though its ranking in the target image (HW 02) is as low as 9, the Kappa is still good, which can be confirmed by visually inspecting the binary result. Indeed, it has some slight points of noise more than the actual best (Li-Tam-C), but it is still a reasonable choice.

Figure 17 – **Texture Matcher Step 4:** Binarize with the recommended algorithm.



Source: The author (2024)

4.5.2 Choosing the best feature descriptor and distance

In order to find good results as shown in Section 4.5.1, it is necessary to choose wisely the best combination of texture descriptor. As described in Section 4.1, 12 feature descriptors have been tested. Several different distance measures have been proposed, each one with different advantages. The three most common ones applied to images are the Euclidean, cosine, and cityblock distances. To choose the best texture descriptor and distance measure combination, a goal-directed approach is taken. The goal of this stage is not necessarily to find the most similar texture in terms of visual perception, but rather to find the best algorithm to binarize a given input document image. Thus, the feature is chosen in terms of the quality of the binarization result.

The first step is to determine a measure of quality for the matching. On Figure 17, the selected algorithm MO_1-C had been ranked higher, it means the Kappa would also be higher, thus the higher the selected algorithm ranking, the higher is the quality. In this case, the ranking difference is of 8. This measure of quality is called “Rank Diff” (R_{diff}) and is used to classify the several combinations of texture descriptor and distance measure (see Figure 18). Applying this calculation to the whole dataset, the final Score (quality measure) for a given descriptor and distance will be given by the summation of the R_{diff} for all images (Table 11).

Figure 18 – Rank Diff: texture matching quality measure.

TARGET IMAGE						
#	Algorithm	Kappa	PSNR	DRD	FM	Time
1	Li-Tam-C	1.00	34.52	0.11	99.73	0.01
2	dSLR-G	0.99	28.01	0.31	98.80	0.01
2	dSLR-L	0.99	28.39	0.29	98.90	0.01
2	Intermodes-G	0.99	29.07	0.27	99.07	0.01
2	Intermodes-L	0.99	27.78	0.34	98.76	0.01
2	Li-Tam-G	0.99	29.07	0.27	99.07	0.01
2	Li-Tam-L	0.99	29.77	0.24	99.21	0.01
3	dSLR-R	0.98	26.90	0.38	98.45	0.01
...
9	MO_1-C	0.92	19.91	2.20	92.82	0.01

Actual Best

Rank Diff (R_{diff}) = 4

Texture-based best

Source: The author (2024)

Note that for this example, TW 10 image, which is typewritten, matched to HW 14, which is handwritten and there were some few bad matches as for HW 15, which is an image with light back-to-front interference that matched to HW 21, which has strong back-to-front

Table 11 – Example of Score for a descriptor and distance combination.

Image		Algorithm		Kappa		Chosen Rank	R_{diff}
Target	Matched	Chosen	Best	Best	Error		
HW 03	HW 12	MO ₁ -C	MO ₁ -R	0.96	0.00	1	0
HW 08	HW 10	dSLR-B	Minimum-G	0.78	0.00	1	0
HW 16	HW 02	dSLR-C	Li-Tam-C	1.00	0.00	1	0
...
TW 10	HW 14	Minimum-C	Howe-C	1.00	0.15	13	12
TW 06	HW 10	IsoData-G	Minimum-G	1.00	0.16	17	16
HW 09	HW 21	Howe-C	Wu-Lu-G	0.99	0.38	30	29
HW 15	HW 21	Minimum-C	Wu-Lu-G	1.00	0.47	39	38
Score for EFOS, Euclidean Distance without grouping						$\sum R_{diff}$	292

Source: The author (2024)

interference (Figure 20). This happened because the images were matched without taking into account other document features and possibly due to an inappropriate descriptor and distance combination. In order to mitigate this issue, the images were clustered according to the back-to-front interference strength and type of printing and the combination of feature and distance was found.

All possible combinations (24 in total) were tested and a summary of the results is presented on Table 12. The combinations are sorted by the summation of all Scores (Rank Diff) and the RMSE is also presented. The final choice for each situation is the top-1 combination: EFOS with Euclidean distance for a global evaluation; EFOS with cityblock if the printing type is known to be typewritten; GLCM-range and FOS with cityblock if the document was handwritten and the back-to-front interference is weak and strong, respectively. Note that the cosine distance was discarded, as it has offered mostly bad results.

Table 12 – Assessment of the combination of feature and distance measure either separating in groups or not

#	Feature	Distance	RMSE	Score	Time
No Groups (Global Evaluation)					
1	efos	euclidean	0.119	292	0.017
2	hep_GLCM_3x3	cityblock	0.114	296	0.006
3	hep_ILBP_3x3	cityblock	0.104	301	0.001
4	hep_IBGC1_3x3	cityblock	0.104	308	0.001
5	hep_LBP_3x3	cityblock	0.110	313	0.001
Handwritten Weak Interference Only					
1	glcm-range	cityblock	0.047	75	0.005
2	sfm	euclidean	0.047	76	0.011
3	glcm-range	euclidean	0.049	82	0.005
4	sfm	cityblock	0.050	84	0.011
5	hep_ILBP_3x3	cityblock	0.055	94	0.001
Handwritten Strong Interference Only					
1	fos	cityblock	0.056	41	0.001
2	fos	euclidean	0.065	45	0.001
3	sfm	cityblock	0.085	55	0.010
4	glcm-mean	cityblock	0.093	63	0.006
5	glcm-mean	euclidean	0.093	63	0.006
Typewritten Only					
1	efos	cityblock	0.063	72	0.017
2	efos	euclidean	0.063	72	0.017
3	sfm	cityblock	0.094	93	0.010
4	glcm-mean	euclidean	0.095	97	0.005
5	glcm-mean	cityblock	0.095	101	0.005

Source: The author (2024)

4.6 Results and Conclusions

The experiments performed confirm that the analysis of the texture of a document may provide a fast and quality-reasonable choice of a binarization algorithm. In total, 12 texture descriptors were used with three different distance measures. The dataset is composed of 39 images with several degrees of back-to-front interference and noises. It has been divided in three subsets: handwritten with strong and weak back-to-front interference and typewritten. For each subset, a combination of texture feature and distance measure was chosen and applied to compare the images. A Leave-One-Out Cross-Validation (LOOCV) approach was adopted to test the efficacy of the method.

The cosine measure did not provide good matching results for this type of application, but the cityblock works best for most texture descriptors and in many cases offer similar results to using the euclidean distance with the same texture descriptor.

The EFOS with euclidean distance, used in previous research, still presented better results in many cases, but it eventually fails when matched with an image with much stronger back-to-front-interference or a large noise difference in some region. If the subset strategy is applied, the results improve significantly and only two images were not properly binarized due to a smudge and large difference in stroke width.

If no information is known about the input image, most of the best feature and distance combinations are the HEP variations of the classical descriptors. In general, EFOS with euclidean distance would be the best choice, but if processing capabilities are limited, the HEP ILBP would be the best, as it is 10 times faster to calculate. A visual representation of the matching, showing the efficacy of the best one without grouping is shown on Figure 14

Now if the document features can be specified, the best choice varies slightly. For handwritten documents with weak interference, the GLCM-range with cityblock would be the best; for handwritten with strong interference, the FOS with cityblock; for typewritten documents in general, EFOS with cityblock. If processing time is a serious constraint, GLCM-mean with cityblock would be a better choice for typewritten documents and the other combinations remain.

If one specifies the type of writing and the strength of the interference, it is possible to look for images with closer characteristics, which improves the results. With this new approach, only image HW 05 and TW 06 had poor results, as depicted on Figure 19. The back-to-front interference of image HW 05 is similar to the one present at HW 06, however the strokes of HW 05 are much ticker, thus the Wolf algorithm with blue channel could not properly binarize it. As for TW 06, the smudge present in part of the text lead to the wrong choice of the Minimum algorithm with all channels (color image). The results with grouping for all images are presented on Table 13. Except for those two images, the chosen algorithm did provide good binarization results and proved the efficacy of the method.

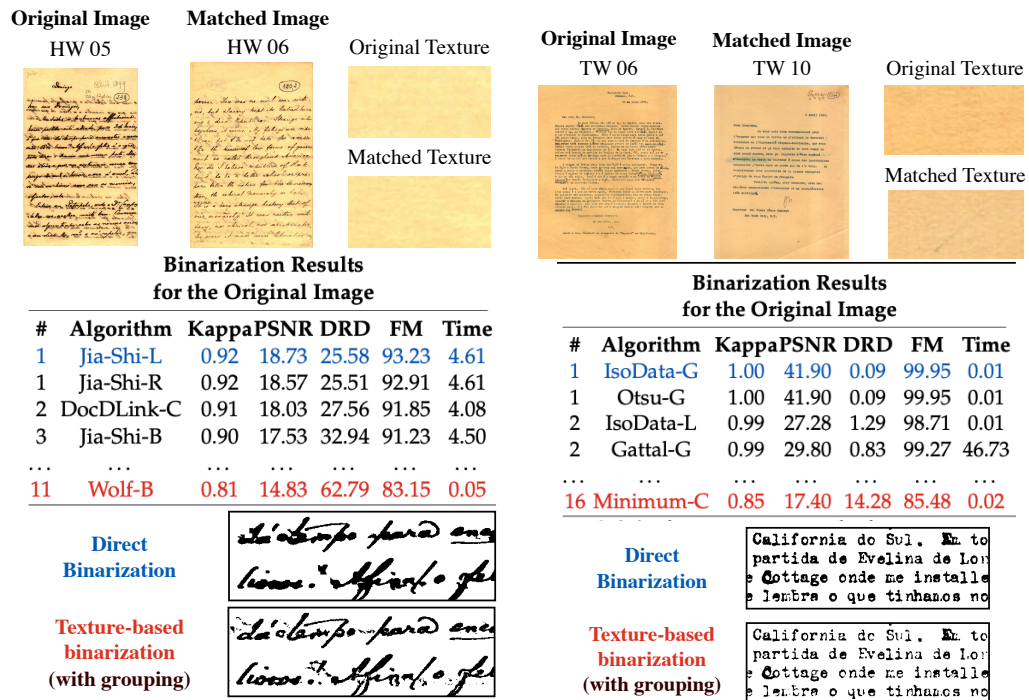
It can be concluded that the paper's texture plays a critical role in the effectiveness of the binarization process and that it can be effectively used to make a good choice of binarization algorithm. In general, the EFOS descriptor with euclidean distance is a good option to apply such matching, but if some features of the images are known beforehand other features should be used.

Table 13 – Texture Matching Considering Image Features – with Groups

Image		Best Algorithm		Kappa		Chosen Rank
Original	Matched	Chosen	Original	Original	Difference	
Handwritten Weak Interference Only - GLCM-Range with Cityblock						
HW 02	HW 03	Li-Tam-C	MO ₁ -C	1.00	0.08	9
HW 03	HW 02	MO ₁ -C	Li-Tam-C	0.96	0.01	2
HW 04	HW 14	dSLR-C	Howe-C	0.96	0.07	8
HW 07	HW 01	Otsu-C	Otsu-C	0.98	0.00	1
HW 09	HW 14	Howe-C	Howe-C	0.99	0.00	1
HW 12	HW 11	MO ₁ -R	JB-L	0.96	0.01	2
HW 13	HW 11	JB-L	JB-L	0.96	0.00	1
HW 14	HW 09	Howe-C	Howe-C	0.99	0.00	1
HW 16	HW 24	dSLR-C	Sauvola-C	1.00	0.06	7
HW 17	HW 01	Intermodes-L	Otsu-C	1.00	0.06	6
HW 19	HW 03	Minimum-C	MO ₁ -C	0.93	0.02	3
HW 23	HW 24	Mello-Lins-R	Sauvola-C	0.82	0.00	1
HW 24	HW 23	Sauvola-C	Mello-Lins-R	0.95	0.08	9
HW 25	HW 16	Su-Lu-C	dSLR-C	1.00	0.08	7
HW 26	HW 27	ISauvola-C	Sauvola-C	0.95	0.06	7
Handwritten Strong Interference Only - FOS with Cityblock						
HW 05	HW 06	Jia-Shi-L	Wolf-B	0.92	0.11	11
HW 06	HW 05	Wolf-B	Jia-Shi-L	0.93	0.03	4
HW 08	HW 10	dSLR-B	Minimum-G	0.78	0.00	1
HW 10	HW 08	Minimum-G	dSLR-B	0.89	0.02	3
HW 15	HW 06	Minimum-C	Wolf-B	1.00	0.08	9
HW 18	HW 08	Sauvola-C	dSLR-B	0.99	0.06	6
HW 20	HW 05	Li-Tam-B	Jia-Shi-L	0.96	0.04	5
HW 21	HW 15	Wu-Lu-G	Minimum-C	0.92	0.02	2
Typewritten Only - EFOS with Cityblock						
TW 01	TW 07	Li-Tam-L	Su-Lu-L	0.94	0.05	6
TW 02	TW 11	dSLR-C	Otsu-G	0.96	0.04	5
TW 03	TW 04	Minimum-C	Su-Lu-L	0.97	0.01	2
TW 04	TW 03	Su-Lu-L	Minimum-C	0.93	0.06	6
TW 06	TW 10	IsoData-G	Minimum-C	1.00	0.15	16
TW 07	TW 01	Su-Lu-L	Li-Tam-L	0.84	0.07	8
TW 08	TW 04	dSLR-C	Su-Lu-L	0.96	0.04	4
TW 09	TW 11	Intermodes-C	Otsu-G	0.95	0.01	2
TW 10	TW 03	Minimum-C	Minimum-C	1.00	0.00	1
TW 11	TW 09	Otsu-G	Intermodes-C	1.00	0.05	5
TW 12	TW 01	dSLR-G	Li-Tam-L	0.66	0.00	1
TW 13	TW 14	Minimum-C	Nick-C	1.00	0.06	7
TW 14	TW 13	Nick-C	Minimum-C	1.00	0.09	9

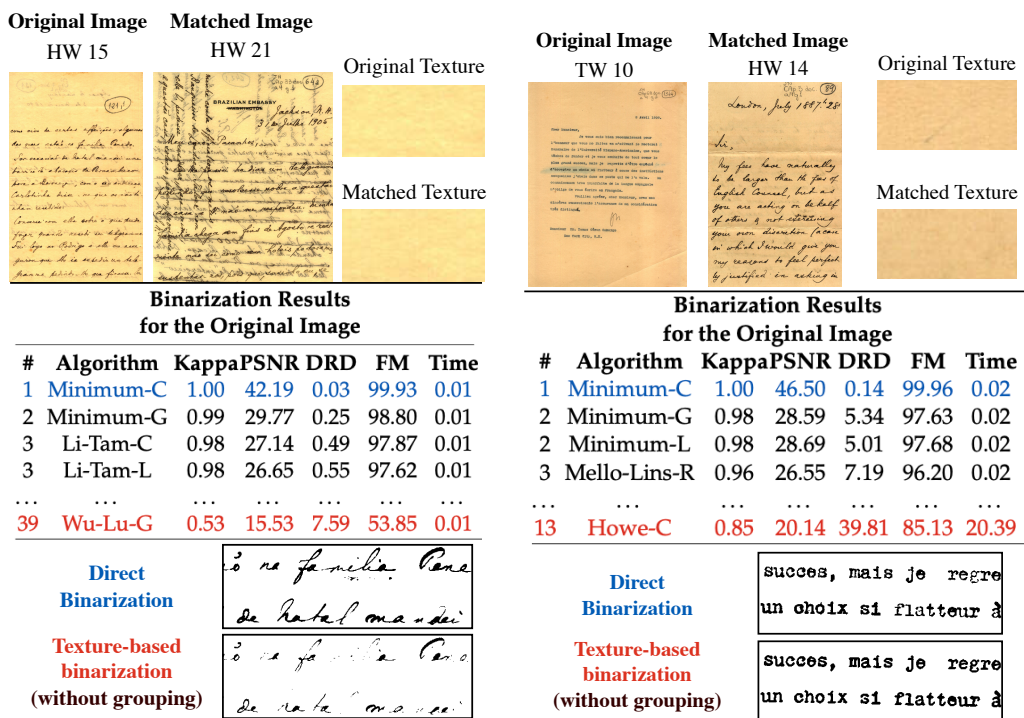
Source: The author (2024)

Figure 19 – Results for image matching with image HW 05 and TW 06 with grouping.











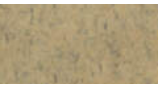








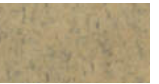


























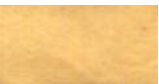










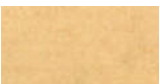
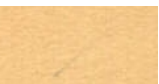


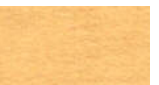












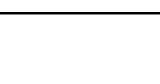
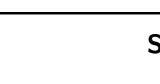
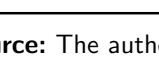
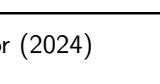
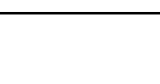
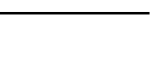






























Source: The author (2024)

Figure 20 – Results for image matching with image HW 15 and TW 10 without grouping.



Source: The author (2024)

Table 14 – Texture Matching for Best three Features without grouping

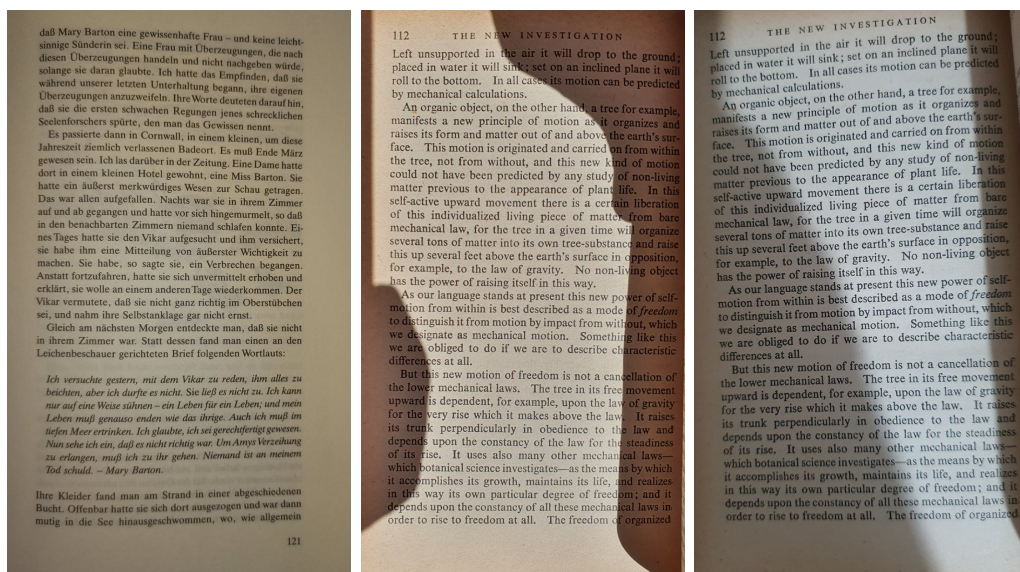
	EFOS - Euclidean		HEP GLCM - Cityblock		HEP ILBP - Cityblock	
Image	Original	Matched	Original	Matched	Original	Matched
HW 01						
						
HW 02						
						
HW 03						
						
HW 04						
						
HW 05						
						
HW 06						
						
HW 07						
						
HW 08						
						
TW 09						
						

Source: The author (2024)

5 NEW EVALUATION MEASURES FOR PHOTOGRAPHED DOCUMENT BINARIZATION EVALUATION

Currently the most common way of capturing document images is by using smartphone cameras, thus this thesis also contributes with assessment solutions to this type of image. Taking photos of documents with smartphone cameras is an attitude that started almost two decades ago [25, 132, 26, 23]. It is extremely simple and saves photocopying costs, allowing the document image to be easily stored and shared using computer networks. However, smartphone cameras were made to take family and landscape photos or make videos of such subjects and were not targeted at document image acquisition. Smartphone document images have several problems that bring challenges to processing them. The resolution and illumination are uneven, there are perspective distortions and often interference from external light sources [23]. Even the in-built strobe flash may add further difficulties if activated by the user or automatically. In addition to all that, the standard file format used by smartphone cameras to save images is jpeg, which inserts jpeg noise [24], a light white noise added to prevent two pixels of the same color from appearing next to each other. This noise makes the final image more pleasant to the human eye glancing at a landscape or family photo, but it also means a loss of sharpness in a document image, bringing difficulties to any further processing. In Figure 21 one can see in better detail three images with some of the usual noises.

Figure 21 – Example of mobile-captured document images. Strobe flash noise (left); Strong shadow with natural light (middle); Skew due to capture angle (right).



Source: The author (2024)

As with scanned documents, the binarization is an essential step in several applications usually applied to such documents, as image enhancement, binarization for compression purposes, deskewing, OCR, etc. However, the binarization of smartphone photographed documents is much more complex than doing the same with scanned ones by the aforementioned factors. In addition to that, each smartphone model has different camera features and there is enormous variation in manufacturers and models. To the best knowledge of the author of this thesis, no other study performed an extensive evaluation with real-sized natural scene-captured images. The first assessments started with a series of competitions that originated from the research of this thesis at the DocEng symposium in 2020 [31]. The whole experimentation and analysis was conducted and written by the author of this thesis. In 2021, that same competition occurred with several new competitors and devices [34]. The third venue [56] of the ACM DocEng Competition on the binarization of photographed documents assessed five new algorithms and 64 algorithms and it was possibly the first time the size of the monochromatic image was considered in the evaluation of binarization algorithms.

Assessing the quality quality of photographed documents is particularly hard to evaluate, as the image resolution is uneven, it strongly depends on the features of the device, the distance between the document and the camera and it even suffers from perspective distortion [25, 133]. One of the first studies on this subject was the creation of PhotoDoc [23], which is a toolbox for correcting the distortions of photographed documents and enhancing its quality, which involves applying binarization. The performance here is measured as the precision of the optical character recognition (OCR) process applied to the final processed image.

In [134], a similar processing pipeline is proposed, but using mobile phones embedded with cameras instead of standalone digital cameras. Fan proposed a web service to receive processing requests and save the improved version of the image. Sergey [135] was one of the first studies to focus specifically on the binarization process applied to photographed textual images. It also compared the results using the OCR recognition rates using the natural image text recognition benchmarks from ICDAR 2003 and 2011.

In [136], a smartphone is used to take pictures of documents and generate the Moiré pattern and specular noise. The accuracy of the OCR is used as a quality measurement; however, the entire image enhancement process was considered. Singh et. al [137] performed an assessment focusing on binarization algorithms only, but using both scanned and camera-captured documents. The capturing device was not specified and the evaluation was conducted with traditional quality measures (PSNR, NRM, F-measure) and, for some images, using OCR

accuracy by measuring Levenshtein [138] distance. In a recent study by Michalak et al. [74, 55, 139], several printed documents were captured with varying illumination conditions and a standalone camera. The illumination was manipulated to generate hard-to-binarize images with strong shadows and spots with a strong concentration of light.

While OCR accuracy is sufficient for applications focused solely on generating a digital transcription of text, it can fall short in scenarios requiring visual quality preservation, such as when preparing images for printing. Noise in the image, though insignificant for transcription, can negatively affect the printed result, increasing ink consumption and producing visually unappealing images. Additionally, the Levenshtein distance, while useful for small-scale comparisons, becomes impractical for large-scale experimentation as it cannot consistently compare results across different algorithms.

This chapter presents two recently proposed and two new evaluation measures to assess document image binarization of smartphone photographed document images. The first was proposed in a previous research [27] and is based on the proportion of black pixels in the resulting binary image (P_{err}), comparing the photo taken with varying resolution to the scanned version. The second, developed during this research in collaboration with other members of the DocEng20 Binarization Competition [31], is a normalized version of the Levenshtein distance ($[L_{dist}]$), comparing the OCR transcription utilizing the Google Vision API with manual transcription. The third considers the TIFF G4 compression format as a measure of quality. It first appeared in the sequence of DocEng competitions in 2022 [56] and is part of this thesis contribution. The fourth is a combination of the normalized Levenshtein distance with the proportion of pixels and is an early publication of the results of this thesis [37]. This chapter is mostly a reproduction of the last publication.

5.1 Materials and Methods

In this assessment, six different models of smartphones from three different manufacturers, widely used today, were used. Their built-in strobe flash was set *on* and *off* to acquire images of offset, laser and deskjet printed text documents photographed at four shots with small variations in position and moments, to allow for different interfering light sources. The document images captured with the six devices were grouped into two separate datasets:

- **Dataset 1:** created for the 2022 DocEng contest [56], the photos were taken with

devices Samsung N10+ (Note 10+) and Samsung S21U (Ultra 5G). It has challenging images with natural and artificial light sources and with strong shadows;

- **Dataset 2:** created for 2021 DocEng contest [34], the photos were taken with devices Motorola G9, Samsung A10S, Samsung S20 and iPhone SE. It also has challenging images, but they are less complex than Dataset 1.

The test images were incorporated to the IAPR (International Association for Pattern Recognition) DIB - Document image binarization platform (<https://dib.cin.ufpe.br>), which focuses on document binarization and had its development started with the author of this thesis in a previous research. The same strategy of Direct Binarization, as explained in Section 4.3, was used here. The binarization algorithms were fed with the color, grayscale converted, and R, G, and B channels of the RGB representation. Here, 68 classical and recently published binarization algorithms are fed with the five versions of the input image, totaling 340 different binarization schemes. The complete list of the algorithms used is presented in Table 1 (page 42), along with a short description and the approach followed in each of them. The details of the camera of each device are described in Table 15. The processing time evaluation details are the same for the whole thesis and some important remarks regarding this are described in Section 3.4 of Chapter 3 (page 58).

Table 15 – Summary of device camera specifications

	Samsung N10+	Samsung S21U	Moto. G9 Plus	Samsung A10	Samsung S20	iPhone SE2
Megapixels	16	12	12	13	12	12
Aperture	F 1.5-2.4	F/1.5	F 1.8	F 1.9	F 1.8	F 1.8
Sensor size	1/2.55 inch	F 1.8 inch	1/1.73 inch	-	1/2.55 inch	1/3 inch
Pixel size	-	1.4 μm	1.4 μm	-	1.4 μm	1.4 μm
Release year	2019	2021	2020	2020	2020	2020
Camera Count	3	4	4	2	3	1

Source: The author (2024)

5.1.1 The Quality Measure of the Proportion of Pixels (P_{err})

An alternative way to measure the quality of binarizing photographed documents is the one proposed at [27], part of a previous research, which is the proportion of black pixels in relation to a reference image. The paper sheet or book page that one wants to binarize is scanned at 300 dpi, binarized with several algorithms, visually inspected, and manually selected and

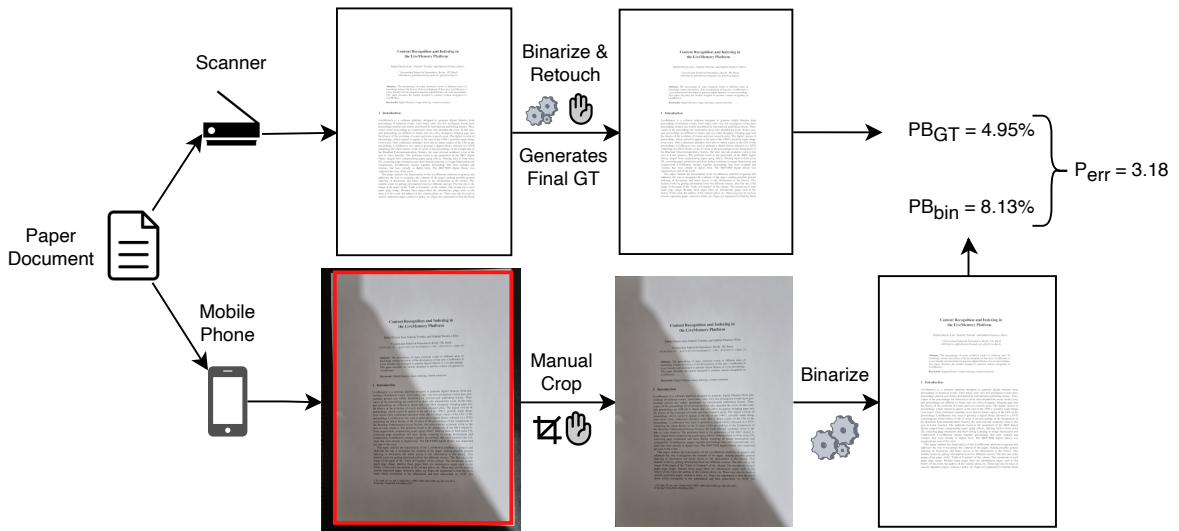
retouched to provide the best possible binary image of that scanned document, which will generate the reference proportion of black pixels for that document image. The P_{err} measure compares the proportion between the black-to-white pixels in the scanned and photographed binary documents, as described in Eq. 5.1:

$$P_{err} = abs(PB_{bin} - PB_{GT}), \quad (5.1)$$

where $PB = 100 \times (B/N)$ is the proportion of black pixels in the image, B is the total number of black pixels and N is the total number of pixels in the image. Thus, PB_{bin} is the proportion of black pixels in the binary image and PB_{GT} is the proportion of black pixels in the scanned ground-truth image.

In order to provide a fair assessment, the photographed image must meet several requirements. The resolution of the output document photo must be close to 300 dpi (which corresponds to the scanned one). To meet such a requirement, the camera should have around 12 Mpixel resolution and the document should cover almost all of the photographed image; the photo must be cropped to remove any reminding border. Here, the cropping is done manually, as the focus is to assess specifically the binarization algorithms. Figure 22 describes the preparation of the images and an example of P_{err} calculation. The P_{err} was used by the last DocEng contests [31, 34, 56] to evaluate the quality of binary images for printing and human reading.

Figure 22 – P_{err} measure example (GT: ground-truth, bin: binary).



Source: The author (2024)

5.1.2 Normalized Levenshtein Distance ($[L_{dist}]$)

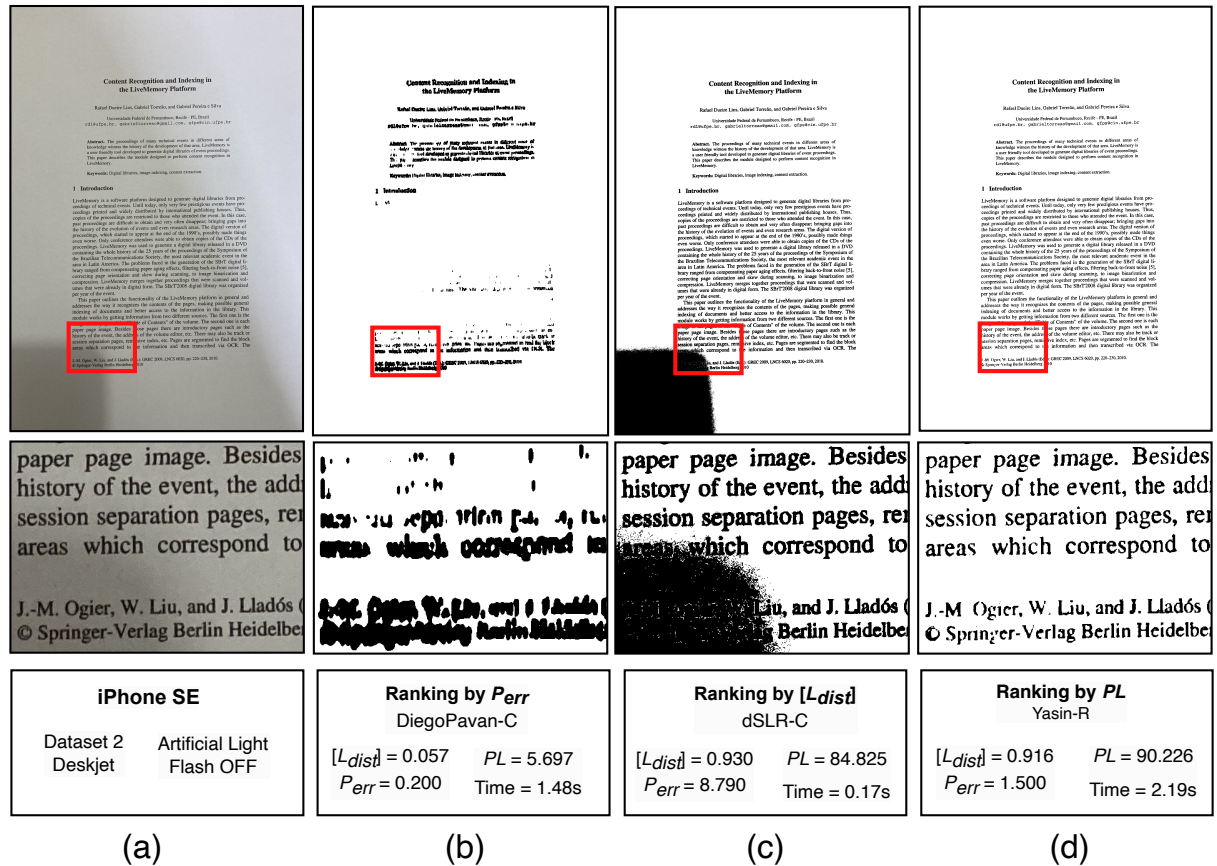
Another alternative quality measure is the Optical Character Recognition (OCR) correctness rate measured by $[L_{dist}]$ [31], which is the Levenshtein [138] distance normalized by the number of characters in the text. The Levenshtein distance, here denoted by L_{dist} , expresses the number of character insertion, deletion, and replacements that would be necessary to convert the recognized text into the manually transcribed reference text for each image. Thus, the L_{dist} depends on the length of the text and cannot be used to measure performance in different documents as an absolute value. In [31], part of this thesis, a normalized version of the L_{dist} was proposed, calculated as:

$$[L_{dist}] = \frac{\#char - L_{dist}}{\#char}, \quad (5.2)$$

where $\#char$ is the number of characters in the reference text.

The DocEng 2022 binarization competition for photographed documents presented a new challenging dataset in which complex shaded areas were introduced. Although the P_{err} quality measure worked well whenever the shaded area was more uniformly distributed, in those more complex multi-shaded documents, some algorithms may concentrate the pixels around some characters (e.g., by dilatation) while completely removing other parts of the document. This could generate an image that has the same proportion of black pixels as the ground truth, a clear background with no evident noise, but its text is unreadable. Taking, for instance, an example image taken with an Apple iPhone SE2 of a deskjet printed document with the strobe flash off (Figure 23a), the algorithm with the closest black pixel proportion would be DiegoPavan provided the original color image. The result is presented in Figure 23b. Note that even the remaining dilated letters are nearly unreadable, giving a $[L_{dist}]$ of nearly zero, which means that almost no text was transcribed. The P_{err} close to zero means that the proportion of black pixels is very close to the ground-truth.

Figure 23 – Comparison between different measures: PL , $[L_{dist}]$, P_{err} . For each case, the full image is shown on the top and an example region bellow, where the red boxes indicates the crop position for the example region. (a) Original image; (b) Ranking by P_{err} only, DiegoPavan-C binarized image; (c) Ranking by $[L_{dist}]$ only, dSLR-C binarized image; (d) Ranking by PL measure, Yasin-R binarized image.



Source: The author (2024)

If we ignore P_{err} and only sort the results by $[L_{dist}]$, the most recommended algorithm to use for this image would be dSLR, having the original color image as input. The result of such binarization is presented on Figure 23c for the same image. Almost all of the text was successfully transcribed ($[L_{dist}]$ close to 1.0), however, there is a large noisy area in the lower left corner, which only did not significantly affect the transcription due to the large margins of the document. This noise was generated by a shadow of the mobile phone and could not be detected by $[L_{dist}]$ measure, but checking P_{err} it is clear that there is a large amount of noise. A printed document usually has almost 5% text pixels (in this image, it was 3.77%), so a difference of 8.79 from the ground truth is a large one. If one would like to just transcribe the text, it could be enough to use such an algorithm for that image; however, if the margins were smaller or the binarized document was printed, such a large noise blurb would be unacceptable.

5.1.3 Pixel Proportion and Levenshtein Measure (PL)

In order to obtain the best OCR quality while providing visually pleasant human-readable binary document images, a new quality measure is proposed here:

$$PL = [L_{dist}] \times (100 - P_{err}). \quad (5.3)$$

Applying such a new measure to the already presented examples of document images would yield $PL = 5.69$ for DiegoPavan-C and $PL = 84.82$ for dSLR-C, while the best algorithm, according to the proposed quality measure, Yasin-R, would yield $PL = 90.22$. The corresponding image is presented in Figure 23 (d), and has a better overall visual quality and OCR transcription rate, although the dSLR algorithm is an order of magnitude faster than the other two algorithms.

5.1.4 Evaluation by Compressed Image File Size

A new measure introduced in this work and recently published [37] is the size of monochromatic image files compressed using the Tag Image File Format Group 4 (TIFF_G4) with Run-length encoding (RLE) [56]. Such a compression scheme is part of the Facsimile (FAX) recommendation and was implemented in most FAX systems at a time when transmitting resources were scarce. The TIFF_G4 file format is possibly the most efficient lossless compression scheme for binary images [24]. One central part of such an algorithm is to apply run-length encoding [140]. Thus, the less salt-and-pepper noise present in the binary image, the longer the sequences of the same color bits, yielding a smaller TIFF_G4 file, which claims for less bandwidth for network transmission and less storage space for archiving. The compression rate is denoted by CR_{G4} and is calculated by:

$$CR_{G4} = 100 \times \frac{S_{G4}}{S_{PNG}}, \quad (5.4)$$

where S_{G4} denotes the size of the compressed TIFF G4 file and S_{PNG} is the size of the Portable Network Graphics (PNG) compressed file with compression level 4. It is important to note that this measure should not be used as an isolated quality measure, it actually provides a secondary fine-grained quality measure and should be used to choose between equally good performance, but that provide smaller files on average.

5.1.5 Quality, Space and Time Evaluation

For each of the six devices studied, the assessment was performed with the strobe flash *on* and *off*, in two different ways:

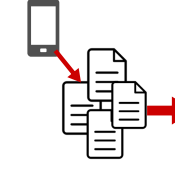
1. **Image-specific best quality-time:** makes use of PL and $[L_{dist}]$ (Tables 18 to 21).
The ranking is performed by first sorting according to the quality measure, and when the quality results are the same, sorted by processing time. This is illustrated in Figure 24.
2. **Best quality-time and compression:** applies the ranking by summation, followed by sorting by processing time, but clustering by device and observing the compression rate for the top-rated algorithms (Table 17).

The ranking summation applied to binarization was first applied in the DIBCO series of binarization competitions [19] and has been used in many subsequent competitions and evaluations [22]. In Figure 25 a visual description of this criterion is presented. First, the algorithms are ranked in the context of each image individually, then the ranking position is summed up across the images, composing the score for each algorithm. The final ranking is determined by sorting the algorithms by the score, and the global mean of all images is presented to provide a quantitative overall ordering.

Sorting directly by the mean of the quality measure gives less precise results, as one here seeks the algorithm that most frequently appears at the top of the ranking, which does not necessarily mean that it is the best quality all the time. In the example of Figure 25, if one sorted by the $[L_{dist}]$ mean alone, the Li-Tam algorithm would be the top ranked algorithm, as for Image 2 its $[L_{dist}]$ is higher than most of the other algorithms, raising its mean value. However, it only appears as the top algorithm for that single image. For most images, Moments is better ranked, indicating that for any given image in such a data set, Moments may provide better results.

The simple mean sorting method is applicable to the first way of assessing the algorithms, as the aggregated images have very similar features (capturing device and print type). As for the second way, the different printing types are aggregated to give an overall result for each device, increasing the variability and making the ranking summation more appropriate.

Figure 24 – Example of ranking by the quality-time criteria



Aggregate images with similar features

Sorting by Quality Mean				Sorting by Quality-Time			
Rank	Algorithm	$[L_{dist}]$	Time	Rank	Algorithm	$[L_{dist}]$	Time
1	jia-shi-R	0.971	22.39	1	ISauvola-B	0.971	0.45
2	ISauvola-B	0.971	0.45	2	jia-shi-R	0.971	22.39
3	Bradley-L	0.970	0.35	3	Bradley-L	0.970	0.35
4	CNW-R	0.970	5.51	4	ISauvola-C	0.970	0.45
5	ISauvola-C	0.970	0.45	5	WAN-B	0.970	1.20
6	WAN-B	0.970	1.20	6	CNW-R	0.970	5.51

Source: The author (2024)

Figure 25 – Example of sorting by the ranking summation criterion

Ranking Summation Sorting

Rank	Image 1		Image 2		Image 3		Overall Best		Mean $[L_{dist}]$
	Algorithm	$[L_{dist}]$	Algorithm	$[L_{dist}]$	Algorithm	$[L_{dist}]$	Algorithm	Score	
1	Moments-R	0.90	Li-Tam-R	0.95	Otsu-R	0.70	Moments-R	1+3+2 = 6	0.753
2	Mean-G	0.80	IsoData-R	0.75	Moments-R	0.68	Li-Tam-R	3+1+4 = 8	0.767
3	Li-Tam-R	0.75	Moments-R	0.68	IsoData-R	0.62	IsoData-R	9	0.657
4	IsoData-R	0.60	Otsu-R	0.62	Li-Tam-R	0.60	Otsu-R	10	0.607
5	Otsu-R	0.50	Mean-G	0.55	Mean-G	0.53	Mean-G	12	0.627

Source: The author (2024)

5.2 Choosing the Best Channel

Following the Direct Binarization approach (Section 4.3), the algorithms were fed with different version of the image, but only the best channel for each image has been considered in the analysis. The PL summation ranking was used as a reference for the choice for each algorithm. In several cases, there was a nearly equal quality result between the *red* or *blue* channels and the *color* image. In some other cases, providing a single channel actually increased the final quality and the channel that more often provided better quality was the *red* channel. Thus, whenever an algorithm yields similar quality results having the full *color* image and one of the channels as input, the *red* channel is chosen as that often means less processing time and space.

Six of the best-ranked algorithms are presented in Table 16 with their respective average *PL* and the score of the ranking summation, stressing that the lower the score, the better the algorithm. The Singh algorithm was one of the few that the *blue* channel offered better

Table 16 – Example of the choice of a channel with some of the best algorithms

Team	Best Channel	Best Channel		Color Image		Luminance	
		Score	Mean <i>PL</i>	Score	Mean <i>PL</i>	Score	Mean <i>PL</i>
michalak21a	Red	632	96.10	817	96.11	727	96.16
YinYang22	Red	649	93.03	825	93.42	687	93.42
Singh	Blue	658	96.14	846	95.42	694	94.98
Wolf	Red	635	94.53	844	93.09	687	95.07
Sauvola	Red	644	93.37	897	90.37	650	93.03

Source: The author (2024)

results. Among the best algorithms, Sauvola was the one with the greatest difference between applying a single channel or the original *color* image.

5.3 Results

For each device model, with the in-built strobe-flash *on* and *off*, the binarization algorithms were evaluated in two contexts: clustering by the specific image characteristics; and aggregating the entire dataset (global evaluation). In all results, the letter after the original algorithm indicates the version of the image used: R – *red*; G – *green*; B – *blue*; L – *luminance*; C – original *color* image. The mean processing time was taken to evaluate the order of magnitude of the time complexity of the algorithms, thus minor time differences are not relevant. The *grayscale* conversion time was not considered here.

Table 17 presents the results for each device using the ranking summation strategy. YinYang22 and Michalak21a are often among the top 5 for any of the tested devices. For Samsung Note 10 +, only HuangUNet showed a significant improvement using a single channel other than red. For Samsung S21 Ultra 5G, ElisaTV presented good results compared to recent efficient ones such as YinYang22. For Motorola G9, Michalak21a would be recommended either with flash *on* or *off*, due to high quality and low processing time. For Samsung A10S, Michalak21a would also be recommended. For Samsung S20, even the most classical algorithm (Otsu) could properly binarize photos taken with flash *on*. It is important to note that Dataset 2 has less complex images than Dataset 1. For iPhone SE 2 and flash *in*, which also used Dataset 2, Otsu again appeared as recommended.

The detailed results for each device are presented in Tables 18 to 21. The quality-time criteria was used (Table 24), as the variation in image characteristics is lower, and therefore

the standard variation is small enough to allow a fair assessment. It is important to note that the standard deviation (SD) of the $[L_{dist}]$ for the Laser and Deskjet dataset was, for all the top 5 and nearly all the other algorithms, approximately 0.04, and for the book dataset it was 0.01, being in some cases close to zero. Only for devices Samsung S21 Ultra 5G and Samsung Note 10+ there was a more significant variation, with the standard deviation varying from 0.1 to 0.3. Those results show that the top 5 algorithms for all test datasets provide excellent binarization results for OCR in general.

The PL standard variation was higher due to a higher variation of the P_{err} measure, which is part of it. For all devices, the SD of the Deskjet and Laser dataset was approximately 4.00, while for the book dataset, it was less than 1 for the devices Motorola G9, Samsung S20, Samsung A10S and between 1 and 3 for devices Samsung Note 10+, iPhone SE 2, Samsung S21 Ultra 5G. The overall quality perceived by visually inspecting the resulting images produced by the top-ranked algorithms is good.

In order to choose the most suitable algorithm for some specific application, the first thing to consider is the intrinsic characteristics of the printing, as different types of ink and printing methods imply entirely different recommendations, as shown in the tables of results. If the document was printed with a deskjet device, it is recommended to check if the strobe flash should be *on* or *off* prior to image acquisition. After that, the binarization algorithm with the best quality-time balance must be applied. If an application has no significant time constraint but the quality is so crucial that even a small amount of lost information is not acceptable, one should choose the top quality-time. However, if the image binarization is part of an embedded application, its processing time is a crucial factor, thus the best quality-time trade-off must be chosen.

Two quality measures were used to support the decision of two types of applications: OCR transcription and printing, archiving, or transmission through computer networks. For the first application (OCR transcription), the $[L_{dist}]$ measure should be used, as it does not take into account visual quality, but only OCR precision, giving algorithms the best chance to provide the best possible transcription. For the second application, visual quality is also important; thus, the measure PL is used, which allows the choice of the best algorithm for OCR transcription and, at the same time, for printing or transmitting.

In general, keeping the strobe flash *on* or *off* does not imply any significant difference in the quality of the best-ranked algorithms, however, in most cases, the set of recommended algorithms varies across the devices. For example, using the Samsung S21 Ultra 5G, the

algorithms recommended for deskjet printed documents are similar if one keeps the flash *on* or *off*, but they are completely different for offset printed books. The same happens for most other devices, using the $[L_{dist}]$ or the PL measure when comparing different setups. This fact highlights the importance of considering as many more algorithms as possible, as in some cases, one algorithm that offers excellent results with one configuration may have totally different results with a different set of capturing conditions, devices, and setup.

In the results table for $[L_{dist}]$ measure, the first red line represents the performance of applying the original color image directly on Google Vision OCR without prior binarization. In most cases, the results are equivalent to the performance of providing a binary image. However, for the Motorola G9 and iPhone SE 2, no OCR output is provided for most captured images. The standard deviation in all cases was nearly zero, which means that there were almost no results for the images. This shows that general-purpose OCR engines can be greatly improved when provided with a clean binary image.

In several cases, the recommended algorithms for OCR ($[L_{dist}]$) match the recommendations using the PL measure with the same input channel or a different one. For example, using Wolf-R to binarize laser documents with flash *off* captured by the Samsung S21 Ultra 5G yields not only excellent OCR results but also good visual quality images. If one checks the example binary image using that algorithm in Figure 26(b), it is possible to see how well this algorithm worked, generating a clear binary image with almost no noise.

It is remarkable how classical global algorithms such as Otsu, dSLR and WAN were quality-time top ranked, but only when using the in-built strobe flash *on*. This happened because the flash was sufficient to diminish the shadows and allow those global algorithms to work well, and highlights that very simple and fast algorithms can still be used for uniform images, even if photographed in different places and by different smartphones.

Figures 26 and 27 present some example images. For each input color image, one of the most recommended algorithms is used, according to the global ranking of Table 17. The cropped portion of the image shows the critical regions where shadows and the flash light reflex can be noticed. For nearly all images, an almost perfect binary image was generated. The laser printing process creates a surface that reflects more light than other types of printing, thus even on the color image, some pixels inside the text stroke are very close to the background ones, making it almost impossible to generate a perfect binary image (Figure 26 (c)). No algorithm tested here did better than that, which highlights a possible problem to be solved by future proposals.

Table 17 – Mobile captured overall results by device sorted according to the ranking summation criterion.

FLASH OFF						FLASH ON				
Rank	Algorithm	Score	PL	CR _{G4}	Time (s)	Algorithm	Score	PL	CR _{G4}	Time (s)
Dataset 1										
Samsung Note 10+										
1	HuangUNet-B	245	96.46	75.22%	58.67	YinYang22-R	261	96.43	79.99%	5.85
2	YinYang22-R	263	96.25	80.25%	6.50	HuangUNet-B	266	96.37	74.79%	58.05
3	Yasin-R	263	96.18	65.60%	1.90	ElisaTV-R	315	95.79	47.36%	8.82
4	iNICK-R	266	96.11	49.26%	3.46	HuangBCD-R	321	96.04	74.88%	249.90
5	Michalak-R	283	96.22	49.17%	0.06	Yasin-R	329	95.65	64.91%	1.76
Samsung S21 Ultra 5G										
1	ElisaTV-R	235	96.30	47.81%	10.38	YinYang22-R	273	91.36	80.20%	5.54
2	YinYang22-R	243	96.13	80.05%	6.36	Michalak21a-R	276	95.98	48.40%	0.04
3	Yasin-R	265	95.95	65.02%	1.78	Singh-B	285	95.45	76.03%	0.34
4	Michalak21a-R	269	91.51	48.02%	0.05	Nick-R	286	95.26	76.07%	0.16
5	Singh-B	289	94.34	75.68%	0.32	ElisaTV-R	310	95.74	48.06%	10.07
Dataset 2										
Motorola G9										
1	Michalak21a-R	218	96.92	47.51%	0.05	Gattal-R	138	97.23	63.09%	53.09
2	ElisaTV-R	230	96.75	45.83%	12.47	Michalak21a-R	150	97.26	47.83%	0.05
3	Michalak-R	230	96.88	47.51%	0.05	YinYang-R	164	97.23	78.48%	1.81
4	YinYang21-R	231	96.83	69.14%	1.71	ElisaTV-R	181	97.18	47.18%	12.21
5	Michalak21c-R	231	96.90	46.71%	1.48	YinYang21-R	214	97.12	69.33%	1.64
Samsung A10S										
1	YinYang22-R	232	97.08	80.84%	4.63	Wolf-R	140	97.24	75.19%	0.16
2	Michalak21a-R	247	97.03	44.06%	0.03	Singh-B	147	97.23	75.19%	0.24
3	Michalak-R	248	97.01	44.13%	0.03	Yasin-R	149	97.26	62.78%	1.30
4	Michalak21c-R	265	96.99	44.07%	0.84	Michalak21a-R	155	97.17	44.03%	0.03
5	YinYang21-R	282	96.85	66.65%	1.08	Nick-R	174	97.21	75.11%	0.11
SamsungS20										
1	Michalak21c-R	199	97.00	47.97%	1.09	Gattal-R	170	97.20	63.78%	52.14
2	Michalak-R	216	96.86	48.16%	0.04	Otsu-R	189	97.11	75.93%	0.02
3	Michalak21a-R	230	96.88	48.13%	0.04	YinYang-R	210	97.08	77.29%	1.42
4	Bradley-R	251	96.82	76.34%	0.29	YinYang22-R	226	97.13	81.39%	5.07
5	YinYang-R	266	96.82	78.03%	1.45	Li-Tam-R	246	97.04	75.89%	0.12
Apple iPhone SE 2										
1	Yasin-R	156	95.44	63.18%	1.59	Otsu-R	192	97.03	75.11%	0.01
2	Sauvola-R	162	96.93	75.49%	0.14	YinYang22-R	211	96.94	81.19%	5.29
3	Singh-B	163	96.94	75.47%	0.23	Yasin-R	229	96.89	62.80%	1.40
4	YinYang22-R	167	96.87	81.32%	5.51	YinYang21-R	235	96.88	67.15%	1.14
5	Nick-R	173	96.90	75.46%	0.14	Gattal-R	235	96.88	62.28%	51.36

Source: The author (2024)

Table 18 – Mobile captured summary of results - PL measure and flash OFF (quality-time criteria).

DESKJET				LASER			BOOK		
Rank	Algorithm	PL	Time (s)	Algorithm	PL	Time (s)	Algorithm	PL	Time (s)
Dataset 1—Flash OFF									
Samsung Note 10+									
1	iNICK-R	96.47	3.48	Sauvola-R	96.59	0.19	Vahid22-C	98.41	29.22
2	Sauvola-R	96.07	0.19	Nick-R	96.58	0.19	HuangUNet-B	98.18	50.22
3	Yasin-R	95.99	1.77	iNICK-R	96.57	3.49	CNW-R	97.97	3.60
4	Nick-R	95.88	0.19	Yasin-R	96.50	1.94	DPLinkNet-C	97.87	9.10
5	Singh-B	95.78	0.40	ElisaTV-R	96.50	11.66	DocDLink-C	97.81	7.01
Samsung S21 Ultra 5G									
1	Sauvola-R	96.59	0.19	Wolf-R	96.75	0.26	Michalak-R	97.78	0.04
2	iNICK-R	95.89	3.43	Nick-R	96.54	0.19	CNW-R	97.75	3.37
3	Wolf-R	95.81	0.25	Singh-B	96.45	0.38	ElisaTV-R	97.65	8.73
4	Singh-B	95.66	0.37	Yasin-R	96.22	1.85	Vahid22-C	97.45	29.14
5	Nick-R	95.62	0.18	iNICK-R	96.14	3.49	Jia-Shi-R	97.44	18.45
Dataset 2 – Flash OFF									
Motorola G9									
1	Nick-R	96.20	0.21	YinYang21-R	96.52	1.67	Michalak21b-R	99.10	3.13
2	iNICK-R	95.63	3.53	YinYang-R	96.51	1.74	Michalak21c-R	99.06	1.48
3	YinYang21-R	95.56	1.73	iNICK-R	96.46	3.50	CNW-R	99.01	3.55
4	Singh-B	95.48	0.51	Nick-R	96.34	0.20	Michalak-R	98.99	0.05
5	Yasin-R	95.44	2.13	Michalak21a-R	96.28	0.05	DPLinkNet-C	98.86	11.86
Samsung A10S									
1	Sauvola-R	96.31	0.12	YinYang22-R	96.70	4.59	ISauvola-R	99.14	0.31
2	Singh-B	96.23	0.26	ElisaTV-R	96.55	7.39	Michalak21c-R	98.97	0.84
3	Nick-R	96.15	0.12	YinYang-R	96.51	1.08	Michalak-R	98.80	0.03
4	Yasin-R	95.90	1.30	Michalak21a-R	96.41	0.03	Vahid22-C	98.80	17.47
5	iNICK-R	95.80	3.27	YinYang21-R	96.36	1.04	WAN-R	98.77	0.78
Samsung S20									
1	Nick-R	96.10	0.15	YinYang-R	96.10	1.41	Michalak21c-R	99.10	1.04
2	Singh-B	95.83	0.34	Michalak21c-R	96.07	1.14	DocUNet-L	99.07	45.50
3	iNICK-R	95.63	3.35	Michalak21a-R	95.98	0.04	Michalak-R	99.06	0.04
4	Yasin-R	95.31	1.63	Bradley-R	95.98	0.31	ISauvola-R	99.05	0.38
5	YinYang-R	95.19	1.37	Michalak-R	95.95	0.04	Bradley-R	99.04	0.28
Apple iPhone SE 2									
1	Yasin-R	95.51	1.67	Yasin-R	96.65	1.60	Singh-B	98.70	0.17
2	Nick-R	95.40	0.14	YinYang22-R	96.52	6.02	YinYang21-R	98.66	1.11
3	Sauvola-R	95.35	0.15	ElisaTV-R	96.50	7.38	Sauvola-R	98.59	0.12
4	YinYang22-R	95.31	5.76	Nick-R	96.37	0.16	Wolf-R	98.53	0.17
5	iNICK-R	95.30	3.31	Sauvola-R	96.28	0.16	Nick-R	98.42	0.12

Source: The author (2024)

Table 19 – Mobile captured summary of results - PL measure and flash ON (quality-time criteria).

DESKJET				LASER			BOOK		
Rank	Algorithm	PL	Time (s)	Algorithm	PL	Time (s)	Algorithm	PL	Time (s)
Dataset 1—Flash ON									
Samsung Note 10+									
1	Sauvola-R	96.25	0.19	YinYang22-R	96.69	6.35	HuangUNet-B	97.62	48.25
2	Yasin-R	96.07	1.98	ElisaTV-R	96.68	11.88	Calvo-Z-R	97.59	1.26
3	Nick-R	96.01	0.19	Yasin-R	96.65	1.82	DocDLink-C	97.29	6.55
4	Singh-B	95.94	0.37	Sauvola-R	96.60	0.20	DocUNet-L	97.27	39.87
5	Yen-CC-C	95.92	0.16	YinYang21-R	96.52	1.55	Vahid22-C	97.24	27.96
Samsung S21 Ultra 5G									
1	Nick-R	96.11	0.18	Singh-B	96.66	0.41	HuangBCD-R	98.12	202.48
2	Singh-B	96.09	0.40	Nick-R	96.58	0.18	WAN-R	97.78	0.87
3	Wolf-R	95.68	0.25	Michalak21a-R	96.02	0.05	HuangUNet-B	97.65	47.00
4	Michalak21a-R	95.27	0.05	Yasin-R	95.97	1.91	CNW-R	97.62	3.35
5	Yasin-R	95.27	1.80	YinYang21-R	95.91	1.55	DocDLink-C	97.48	6.28
Dataset 2—Flash OFF									
Motorola G9									
1	Sauvola-R	96.66	0.22	Nick-R	96.74	0.20	Michalak21a-R	99.29	0.05
2	Nick-R	96.08	0.21	YinYang-R	96.62	1.69	ElisaTV-R	99.28	11.42
3	Singh-B	95.81	0.49	Gattal-R	96.60	53.34	Bradley-R	99.24	0.35
4	Wolf-R	95.57	0.29	Singh-B	96.58	0.45	Michalak21c-R	99.15	1.30
5	YinYang-R	95.56	1.83	YinYang21-R	96.44	1.59	Michalak-R	99.06	0.05
Samsung A10S									
1	Sauvola-R	96.23	0.12	Nick-R	96.40	0.11	Wolf-R	99.46	0.16
2	Yasin-R	95.68	1.25	Yasin-R	96.38	1.27	Michalak21c-R	99.41	0.80
3	ElisaTV-R	95.62	5.95	YinYang-R	96.18	1.05	Michalak21a-R	99.35	0.03
4	Nick-R	95.56	0.12	Wolf-R	96.12	0.16	Singh-B	99.32	0.23
5	Singh-B	95.56	0.25	Singh-B	96.12	0.25	YinYang22-R	99.20	4.47
Samsung S20									
1	Shanbhag-R	96.36	0.13	Sauvola-R	96.67	0.16	Ergina _L -L	99.42	0.56
2	Nick-R	95.77	0.15	Yasin-R	96.66	1.59	Michalak21c-R	99.36	0.95
3	Singh-B	95.57	0.33	Otsu-R	96.57	0.02	Michalak21a-R	99.35	0.04
4	Gattal-R	95.30	52.04	YinYang22-R	96.51	5.27	Bradley-R	99.35	0.26
5	Sauvola-R	95.26	0.16	Gattal-R	96.49	52.64	Ergina _G -L	99.28	0.42
Apple iPhone SE 2									
1	ElisaTV-R	96.11	3.18	Otsu-R	96.57	0.02	YinYang21-R	98.74	1.09
2	Gattal-R	95.93	51.76	Nick-R	96.55	0.15	Ergina _G -L	98.60	0.36
3	Li-Tam-R	95.87	0.12	ElisaTV-R	96.54	4.07	YinYang-R	98.58	1.34
4	Nick-R	95.83	0.15	Singh-B	96.53	0.26	Ergina _L -L	98.56	0.49
5	Singh-B	95.79	0.26	YinYang22-R	96.51	5.51	YinYang22-R	98.56	4.26

Source: The author (2024)

Table 20 – Mobile captured summary of results - L_{dist} measure and flash OFF (quality-time criteria).

DESKJET				LASER			BOOK		
Rank	Algorithm	$[L_{dist}]$	Time (s)	Algorithm	$[L_{dist}]$	Time (s)	Algorithm	$[L_{dist}]$	Time (s)
Dataset 1—Flash OFF									
Samsung Note 10+									
0	Google Vision	0.971	–	Google Vision	0.971	–	Google Vision	0.984	–
1	HuangUNet-B	0.971	64.271	HuangUNet-B	0.971	64.329	iNICK-R	0.990	3.421
2	Michalak-R	0.970	0.051	Michalak-R	0.970	0.051	Vahid22-C	0.990	29.224
3	Nick-R	0.970	0.188	Michalak21a-R	0.970	0.052	Singh-B	0.988	0.255
4	Sauvola-R	0.970	0.194	Nick-R	0.970	0.188	Yasin-R	0.986	1.967
Samsung S21 Ultra 5G									
0	Google Vision	0.971	–	Google Vision	0.971	–	Google Vision	0.982	–
1	Jia-Shi-R	0.971	22.391	Wolf-R	0.971	0.259	Niblack-C	0.988	0.133
2	Wolf-R	0.970	0.254	CNW-R	0.971	3.506	ElisaTV-R	0.986	8.726
3	ISauvola-R	0.970	0.453	Jia-Shi-R	0.971	22.470	Michalak-R	0.985	0.038
4	WAN-R	0.970	1.209	Nick-R	0.970	0.187	Bradley-R	0.984	0.266
Dataset 2—Flash OFF									
Motorola G9									
0	Google Vision	0.000	–	Google Vision	0.000	–	Google Vision	0.001	–
1	Bradley-R	0.968	0.401	iNICK-R	0.970	3.503	WAN-R	0.997	1.226
2	CNW-R	0.968	3.595	ISauvola-R	0.969	0.491	CNW-R	0.997	3.547
3	YinYang22-R	0.968	6.636	YinYang21-R	0.969	1.672	Jia-Shi-R	0.997	23.597
4	Michalak21a-R	0.967	0.055	CNW-R	0.969	3.578	Michalak21a-R	0.996	0.050
Samsung A10S									
0	Google Vision	0.970	–	Google Vision	0.971	–	Google Vision	0.995	–
1	dSLR-R	0.971	0.030	YinYang22-R	0.969	4.588	Michalak21a-R	0.996	0.033
2	WAN-R	0.970	0.795	CNW-R	0.968	3.240	ISauvola-R	0.996	0.308
3	ISauvola-R	0.969	0.294	Vahid22-C	0.968	16.820	WAN-R	0.996	0.776
4	Michalak21c-R	0.969	0.849	Vahid-B	0.968	17.314	Michalak21c-R	0.996	0.838
Samsung S20									
0	Google Vision	0.971	–	Google Vision	0.971	–	Google Vision	0.995	–
1	ISauvola-R	0.970	0.376	Michalak21c-R	0.968	1.141	Nick-R	0.996	0.147
2	YinYang22-R	0.970	5.789	CNW-R	0.968	3.441	WAN-R	0.996	0.973
3	Vahid22-C	0.970	21.839	Vahid22-C	0.968	22.565	DE-GAN-G	0.996	3.334
4	WAN-R	0.969	1.032	Michalak-R	0.967	0.043	CNW-R	0.996	3.410
Apple iPhone SE 2									
0	Google Vision	0.804	–	Google Vision	0.000	–	Google Vision	0.990	–
1	Ergina _G -L	0.972	0.409	Otsu-R	0.971	0.017	WAN-R	0.991	0.798
2	Gattal-R	0.972	50.697	WAN-R	0.971	1.027	CNW-R	0.991	3.416
3	Otsu-R	0.971	0.015	DPLinkNet-C	0.971	9.845	Singh-B	0.990	0.173
4	Li-Tam-R	0.971	0.105	Vahid-B	0.971	22.857	Bradley-R	0.990	0.214

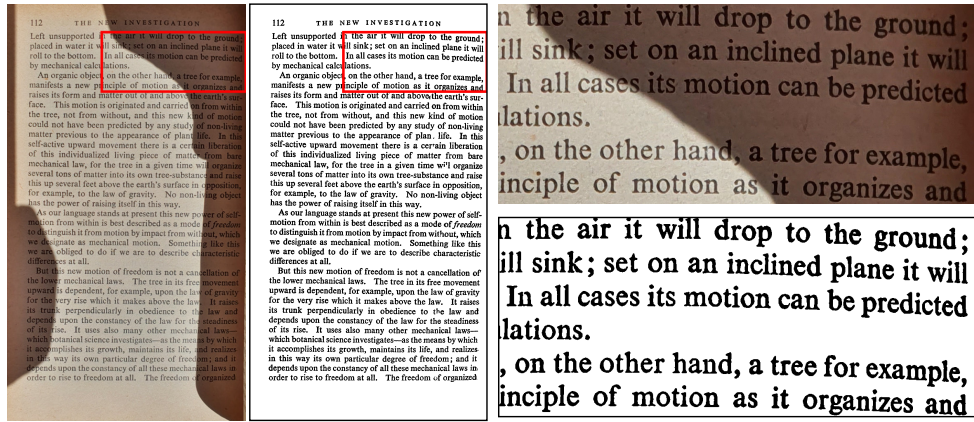
Source: The author (2024)

Table 21 – Mobile captured summary of results - L_{dist} measure and flash ON (quality-time criteria).

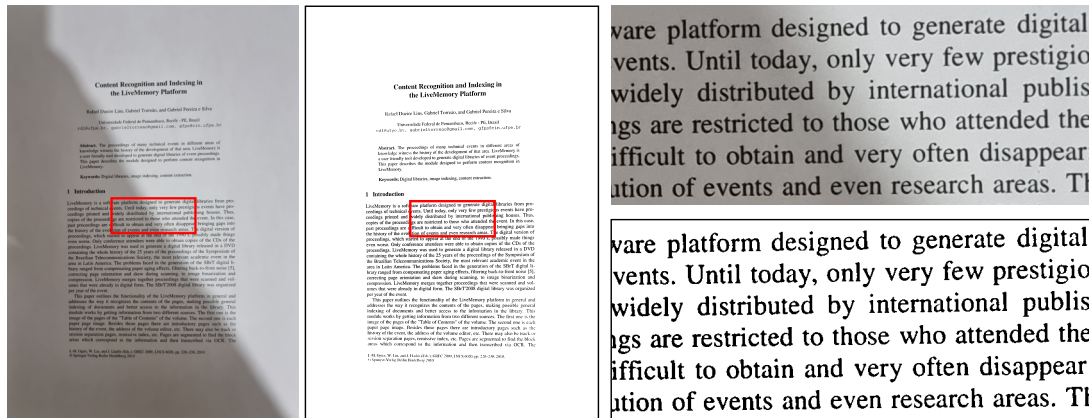
DESKJET				LASER				BOOK			
#	Algorithm	$[L_{dist}]$	Time (s)	Algorithm	$[L_{dist}]$	Time (s)	Algorithm	$[L_{dist}]$	Time (s)		
Dataset 1—Flash ON											
Samsung Note 10+											
0	Google Vision	0.971	–	Google Vision	0.971	–	Google Vision	0.984	–		
1	DocDLink-C	0.971	8.926	Michalak21b-R	0.970	3.230	Nick-R	0.984	0.134		
2	DPLinkNet-C	0.971	12.102	Yasin-R	0.969	1.822	YinYang22-R	0.983	5.227		
3	Jia-Shi-R	0.971	23.264	Vahid-B	0.969	29.386	Calvo-Z-R	0.981	1.256		
4	DilatedUNet-G	0.971	36.097	HuangUNet-B	0.969	65.967	HuangUNet-B	0.981	48.253		
Samsung S21 Ultra 5G											
0	Google Vision	0.971	–	Google Vision	0.971	–	Google Vision	0.983	–		
1	ISauvola-R	0.971	0.434	Vahid-B	0.969	27.036	HuangBCD-R	0.987	202.484		
2	Michalak21a-R	0.970	0.049	Singh-B	0.968	0.414	Michalak21a-R	0.982	0.037		
3	WAN-R	0.970	1.183	Nick-R	0.967	0.181	Singh-B	0.982	0.245		
4	CNW-R	0.970	3.502	Michalak21c-R	0.967	1.318	WAN-R	0.982	0.865		
Dataset 2—Flash ON											
Motorola G9											
0	Google Vision	0.000	–	Google Vision	0.000	–	Google Vision	0.001	–		
1	Michalak21a-R	0.971	0.055	Michalak21a-R	0.970	0.053	Vahid-B	0.997	26.296		
2	Bataineh-R	0.971	0.153	Michalak-R	0.970	0.053	Yen-CC-C	0.996	0.170		
3	Nick-R	0.971	0.209	Bataineh-R	0.970	0.147	Singh-B	0.996	0.360		
4	Sauvola-R	0.971	0.216	ISauvola-R	0.970	0.478	Ergina _G -L	0.996	0.562		
Samsung A10S											
0	Google Vision	0.967	–	Google Vision	0.971	–	Google Vision	0.997	–		
1	ElisaTV-R	0.970	5.952	Michalak21a-R	0.968	0.032	Michalak21a-R	0.998	0.034		
2	HuangBCD-R	0.970	171.542	Michalak-R	0.968	0.032	Nick-R	0.998	0.115		
3	dSLR-R	0.969	0.025	Bradley-R	0.968	0.218	WAN-R	0.998	0.754		
4	Moments-R	0.969	0.026	Singh-B	0.968	0.254	Jia-Shi-R	0.998	15.750		
Samsung S20											
0	Google Vision	0.967	–	Google Vision	0.971	–	Google Vision	0.997	–		
1	Nick-R	0.970	0.154	ISauvola-R	0.970	0.362	Otsu-R	0.997	0.014		
2	ISauvola-R	0.970	0.372	YinYang22-R	0.970	5.271	dSLR-R	0.997	0.098		
3	CNW-R	0.970	3.419	Bataineh-R	0.969	0.111	Li-Tam-R	0.997	0.098		
4	YinYang22-R	0.970	5.221	Jia-Shi-R	0.969	20.096	Wolf-R	0.997	0.186		
Apple iPhone SE 2											
0	Google Vision	0.638	–	Google Vision	0.000	–	Google Vision	0.987	–		
1	WAN-R	0.971	0.992	ISauvola-R	0.969	0.347	YinYang21-R	0.991	1.087		
2	Otsu-R	0.970	0.016	WAN-R	0.969	0.958	Michalak21b-R	0.991	2.254		
3	Michalak-R	0.970	0.041	DE-GAN-G	0.969	3.181	DE-GAN-G	0.991	2.860		
4	Bataineh-R	0.970	0.114	YinYang22-R	0.969	5.508	Vahid22-C	0.991	16.958		

Source: The author (2024)

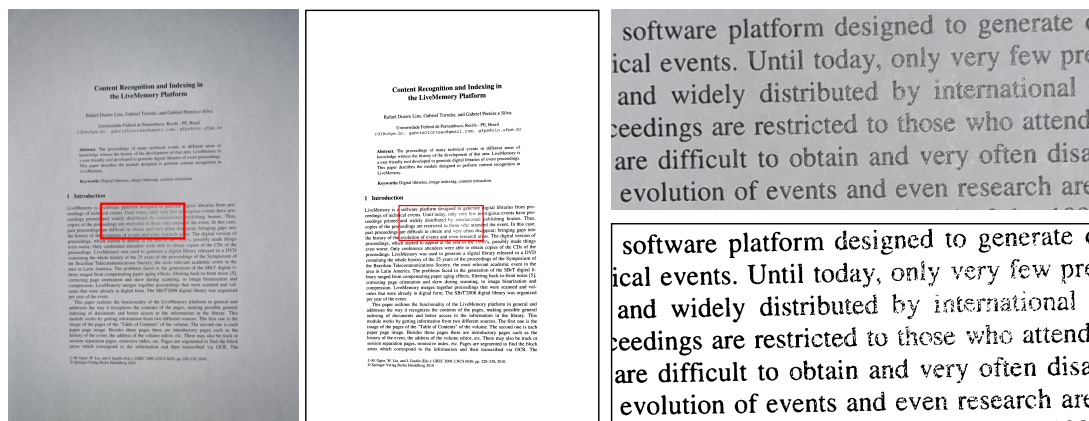
Figure 26 – Dataset 1 example images. **(a)** Samsung Note 10+, book offset page, strong natural light, flash off with strong shadow, binarized by HuangUNet-B; **(b)** Samsung S21, laser printed, artificial light, medium shadow, flash off, binarized by Wolf-R; **(c)** Same as (b), but with flash on and binarized by YinYang22-R.



(a)



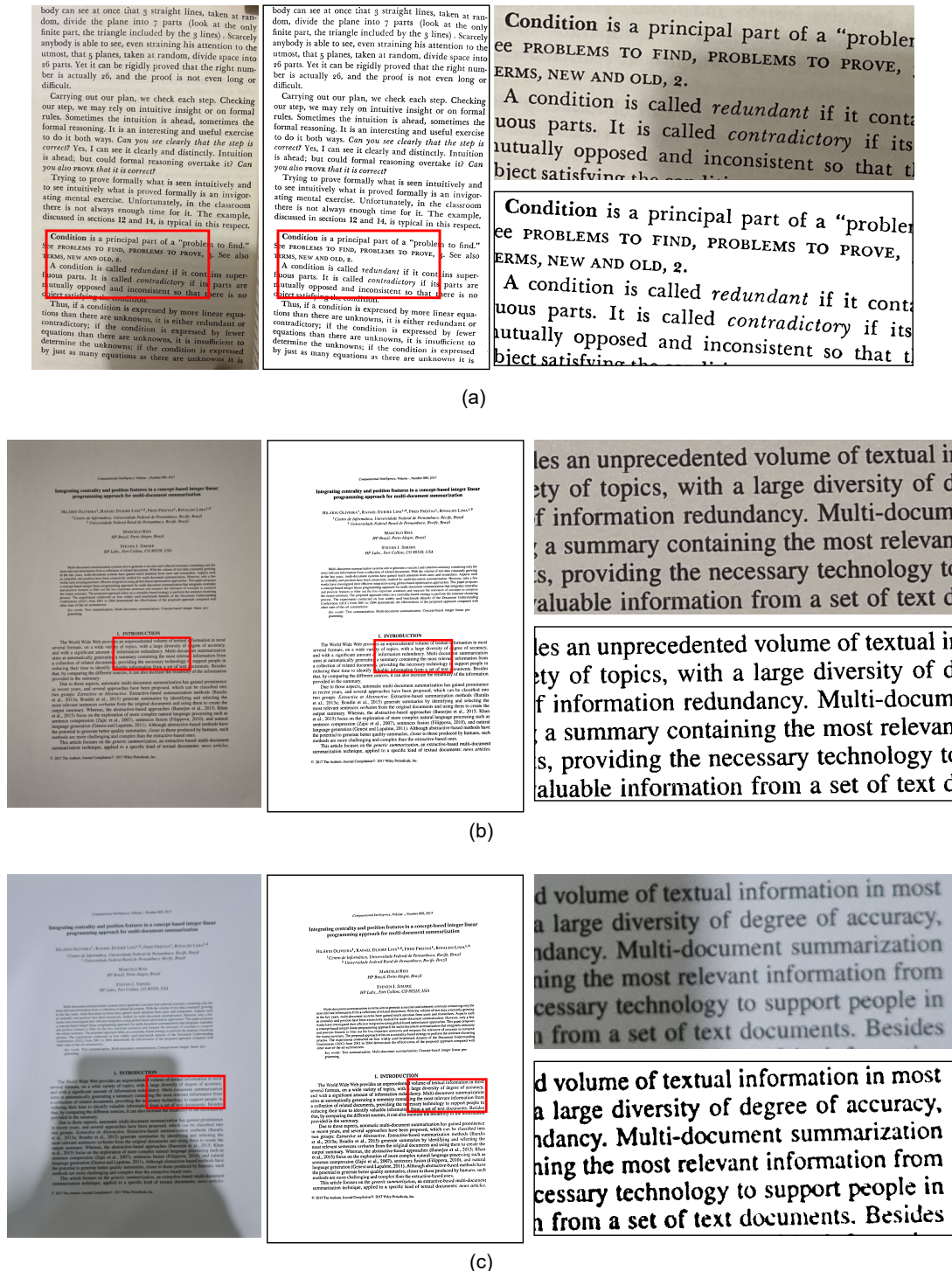
(b)



(c)

Source: The author (2024)

Figure 27 – Dataset 2 example images. **(a)** iPhone SE 2, book offset page, artificial light, flash *off* with medium shadow; **(b)** Samsung S20, deskjet printed, artificial light, medium shadow, flash *off*; **(c)** Same as (b), but with flash *on*, note that on deskjet printed pages no flash reflex interfere on the photo



Source: The author (2024)

5.4 Conclusions

Mobile captured document image still offer many challenges in several document processing applications, with a continuous demand for improving existing methods and developing new ways of analysing existing ones. In this part of the research, 68 binarization algorithms were evaluated in images acquired using six different models of smartphones from three different manufacturers, widely used today. The quality, size of the produced image and processing time of the binarization algorithms are assessed. Given the traditional OCR-based evaluation does not handle noises in non-textual area, a novel quality measure is proposed that combines the Levenshtein distance with the overall visual quality of the binary image. The mean compression rate of the TIFF G4 file with RLE compression was also analyzed and proposed as an addition to the analysis. It provides a quality analysis as the quantity of salt-and-pepper noise in the final image degrades file compression performance, thus it is an extra reference for the overall quality.

The results were presented through two perspectives: a detailed evaluation considering the device, the built-in strobe flash state (*on* or *off*), and the printing technology (deskjet, laser, or offset); a device-based evaluation considering visual quality and compressed binary image file size.

Several conclusions may be drawn from the results presented:

1. Keeping the strobe flash *on* or *off* may not imply in a better quality image, but one needs to make the right choice of the binarization algorithm in order to have the best monochromatic image.
2. The ranking order is nearly completely different through all the different possible setups, thus it reinforces the claim that no binarization algorithm is good for all document images.
3. The quality of the images yielded by the top-rated algorithms with the offset-printed documents (book) dataset is almost perfect if considering the OCR transcription precision.
4. In several cases, such as for iPhone SE 2, some global algorithms had the best performance. They are much faster than the newer algorithms and, in some rare cases, even generate cleaner images (better *PL*).

5. Even when not in the top rank, newer algorithms, such as Michalak or YinYang algorithms and their variants, dominate the results. It is important to stress that they were developed having as target photographed documents, while most of the other algorithms, overall, the global ones, were developed aiming at scanned document images.
6. If compression rate is a priority, YinYang22, with any of the input versions of the image, would be the algorithm that is the most recommended overall, as it offers the best compression rates while maintaining high quality.
7. If processing time is a priority, Michalak21a with the red channel would be the algorithm overall recommended, as it requires a small processing time, comparable to one of the classical algorithms, while providing high-quality binary images.
8. The PL measure provides a better overall quality evaluation of binarization algorithms compared to traditional mobile-captured image assessment measures.
9. Analyzing the TIFF G4 compression rate with RLE has also proved valuable, as, on several occasions, two algorithms provided similar quality results, but one may be two times more efficient in this compression scheme.
10. None of the algorithms tested could perfectly binarize the regions of the laser-printed documents in which the strobe flash (whenever *on*) created a strong noise in the central region of the image, which suggests that such a set-up should be avoided when photographing laser-printed documents.

6 CONCLUSIONS AND FUTURE WORK

In this chapter, we present the global conclusions drawn from the work presented in this thesis, highlighting how the objectives were achieved and the innovative aspects of each contribution. In addition, we discuss potential applications and suggest future research directions.

This thesis has made significant advances in the field of document image binarization by addressing critical questions and introducing novel methodologies that guide the application of binarization techniques in real-world scenarios. Given the vast diversity of binarization methods available, our research provides a crucial framework for selecting the most suitable method under various conditions. The results extend beyond algorithmic improvements, demonstrating significant potential for practical applications in domains such as historical preservation, education, and digital archiving.

In Chapter 2, an updated literature review has been conducted which presents details on the implementation of the most effective algorithms tested in this study. The most remarkable algorithms of the last few decades have been streamlined, so future researchers can easily grasp the evolution of the methods. Note how algorithms proposed as old as 1979 (Otsu), 2002 (Wolf), and 2013 (Howe) are still among the best, even with many recent advances with Deep Learning. On the other hand, even in the era of advanced neural networks, clustering-based (like Gosh), simple image processing (like Michalak, YinYang and HBUT) techniques can still be used to create fast and reliable solutions.

On Chapter 3, the algorithms described and listed on Chapter 2 are extensively tested with historical and modern scanned documents. A discussion of the existing evaluation methodologies is presented, from which we can conclude that some gaps are filled with the results of this thesis. Although this is an important research area, the largest studies so far are either too old (Ismail, in 2018) or do not test the algorithms, but only discuss about the scattered results present on the authors' papers (Tensmeyer, in 2020).

A discussion of the evaluation methodologies is also presented, where Cohen Kappa is introduced as a quality measure for binarization. Even though it is normally used for multiclassifier predictions assessment or medical studies, here it has been shown to be useful given its interpretability (the closer to 1.0 the better) and good correspondence with visual inspection. The classical measures like PSNR or DRD have also been applied for completeness, but in most cases the Kappa has been considered the main measure and a careful visual inspection

confirmed it as being appropriate.

The most important algorithms have been assessed and a set of recommendations have been made depending on the input document characteristics. In some cases, older algorithms will outperform modern ones and that is mainly due to the fact that the first researchers found more general solutions, theory-based, while the modern ones are data-based. Specifically to the assessment conducted here, Sauvola and Wolf are the most outstanding old algorithms with good results, but on most times, newer algorithms outperformed them, even if not trained specifically for the tested dataset.

Given the large variety of results and algorithms, a methodology for automatically choosing the best binarization algorithm for a given document image given its feature of paper texture has been proposed. In Chapter 4, this strategy is described along with an extensive assessment with 63 binarization schemes. The texture features of the paper have been effectively utilized to determine the optimal binarization algorithm utilizing 39 document images from the Nabuco dataset.

In order to produce a more in-depth analysis, the “Direct Binarization” approach has been applied, where each input image is converted into 5 different variations based on the RGB channels. For some specific cases, such as the deep learning-based DocDLink algorithm, providing only the green channel offers a better result, while for YinYang21, providing the luminance version is better than the standard full RGB color image.

Finally, the texture matching approach was applied to this vast space of results using the LOOCV validation approach to verify its applicability. In summary, texture-based binarization consists of finding the most recommended algorithm for a large set of previously binarized images and then, for a given input image, the most similar among them. This gives us a set of recommended binarization schemes for the input image without the need to manually binarize and choose one algorithm among the several options.

In order to choose the most similar image, 12 texture descriptors were applied from two different previous studies. They were chosen based on their applicability to document image representation and time performance. Three distance measures were initially considered (Euclidean, Cityblock and Cosine), however, the first two proved to be useful in this context, as the Cosine often provided inconsistent texture matching. In the end, the combination of using the EFOS features with Euclidean distance was the overall recommendation. If one wants a more precise result in the matching, two other features (FOS and GLCM-range) with Cityblock distance were found to be more appropriate.

The overall results have shown that the paper texture plays a critical role in the effectiveness of the binarization process and that it can be effectively used to make a good choice of binarization algorithm.

Improving the binarization of scanned historical document images is vital in scientific applications, however, the majority of the document images produced every day are modern, mobile-captured images. They are captured in unconstrained conditions, with a wide variety of capturing devices, resolutions, illumination, and perspective. Given the complexity of these types of images, our study has undertaken a comprehensive evaluation of mobile captured documents in order to provide new insights and solutions for this kind of application.

A key point in this context is to provide a consistent and comprehensive evaluation measure, which currently is only based on the efficacy of OCR transcriptions. While useful for situations where only the text extraction is important, it can fall short in scenarios requiring visual quality preservation, such as when preparing images for printing. In Chapter 5 is presented the new evaluation measure (PL), which is proposed in this thesis, that combines OCR accuracy with non-textual noise to determine the quality of the generated image. This measure has been tested with 240 document images, which were binarized with 68 binarization algorithms. In order to provide a diverse test set, six devices were used to capture the documents from three different angles, three illumination conditions, three types of ink, and with the strobe flash on and off.

The PL measure is a combination of the proportion of black pixels present in the image with the normalized Levenshtein distance. The first measure the overall visual quality, as too much noise would either increase or decrease the number of black pixels when compared with the ground-truth image. As shown in the experiments, in many cases it does provide good correspondence with visual inspection (generates a readable binary image); however, sometimes it can be misleading, and thus the Levenshtein distance from the original text (ground truth) to the generated text is added. In order to be able to compare the results with different images, the Levenshtein distance has been normalized with the text length.

Direct binarization has also been applied; however, at this point of the research it has been concluded that choosing a single channel for each algorithm would be enough for a proper evaluation, thus a brief study pointed to the best channel for each algorithm. Whenever there was no difference between the original color image or a single channel, the red channel was chosen.

After an extensive analysis, it has been found that among the 68 tested methods, YinYang22

and Michalak21a were the most successful. They use a combination of traditional image processing techniques and fine-tuned parameters and were idealized to work best with photographed documents. However, for one device (Samsung S21), an older ElisaTV algorithm was even better. The datasets used here are from two different competitions, where Dataset 2 has a more uniform illumination, and thus traditional methods like Otsu were enough to binarize them.

The main contribution to performing the assessment using the new *PL* measure, as detailed exposed in the document, is the ability to generate documents that are not only readable but also good for printing. In several cases, the binarization might add extra noise outside the text region, which does not comprise the reading, but impacts the overall appearance if one wants to print or, even further, increases the file size. The detailed analysis presented here allows future research to start from an advantage point, testing only the best algorithms for each case, or even expanding the image matcher to detect photographed document conditions and choose the best algorithm for it.

This research, which analyzed thousands of results generated by the nearly 70 algorithms, has culminated in five binarization competitions, where the entire process was managed by the author. It was necessary to develop a sophisticated framework to uniformly capture and analyze the performance of each binarization scheme. In addition, new datasets have been developed and published on an IAPR-recognized platform, marking a significant contribution to the research community.

In summary, the key achievements of this research are as follows.

1. **Expansion of the DIB Platform:** Integration of 46 new algorithm implementations, 24 new historical images with manually generated ground truth, 296 new captured images on mobile devices, and results from five binarization competitions.
2. **Introduction of New Evaluation Methodologies:** Contextualized evaluation based on document characteristics such as paper texture, luminosity, back-to-front interference strength and the specific features of documents captured on a mobile device, such as flash condition of the strobe or device type.
3. **RGB Channels Evaluation:** A novel approach to binarization using individual RGB channels, demonstrating that comparable or even superior results can be achieved compared to traditional grayscale images.

4. **Texture-Based Binarization:** A validated texture-based approach for selecting the most appropriate binarization algorithm, with significant implications for automating document processing.
5. **Application of Cohen's Kappa:** Introducing Cohen's Kappa as a robust statistical measure to assess the quality of scanned document binarization.
6. **New Quality Measure (PL):** Development of a comprehensive quality measure for photographed documents, which incorporates both OCR transcription accuracy and visual quality.
7. **Processing Time and Image Compression:** Introduction of processing time and compressed binary image size (CR_{G4}) as key performance indicators, offering a holistic evaluation of binarization algorithms.

6.1 Future Works

Although this thesis has addressed many challenges, several avenues for future research remain open. Beyond paper texture, other features such as stroke width, background contrast, noise type, and additional document characteristics could be incorporated into the image matching tool to improve its accuracy and adaptability across different document types. For instance, there are several historical documents in European libraries with several colored letters which have not been properly studied.

Several libraries made a large part of their historical documents freely available online. A very important work would be to generate new datasets of manually retouched binary images or even the creation of a tool that could assist a human to generate the best binary image based on a combination of several algorithms and an interface to manually choose the best result.

The deep learning algorithms are mostly trained with the default DIBCO library, thus testing the training with different subsets, especially the datasets developed during this thesis, could provide insights into the evolution of these methods and identify opportunities for further refinement. Extending the analysis to include more images from ancient documents in Asia and the Middle East, with their unique noise profiles, could reveal new challenges and opportunities to improve binarization techniques.

Given the large diversity of document features, a promising application would be to split the image into different regions and binarize each region with the best algorithm based on the contrast information for that region.

Most algorithms have parameters that were heuristically or manually set by their authors. One possible expansion of this thesis would be to systematically test several combinations of parameter values together with many algorithms. This could lead to a significant increase in quality even with older algorithms.

Regarding the mobile-captured images, a promising direction for future work involves refining the use of strobe flash in auto mode for smartphone-captured images. By enabling devices to dynamically adjust flash usage based on ambient lighting conditions, the quality and consistency of captured images could be significantly enhanced. Different quality measures could be used, such as counting the number of detected words that are present in the dictionary of the target language. One particularly little studied topic is how to binarize images affected by the Moire effect when taking photos from screens. Finally, increasing the number of devices tested and clustering them based on shared characteristics (such as camera specifications, software versions, or hardware configurations) could lead to more nuanced insights into the performance of binarization algorithms across different platforms.

REFERENCES

- 1 O'GORMAN, L. G. V. K. R. Document image analysis: A primer. *Sadhana*, v. 27, n. February, p. 3–22, 2002. ISSN 0256-2499.
- 2 OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 9, n. 1, p. 62–66, 1979. ISSN 0018-9472.
- 3 SEZGIN, M.; SANKUR, B. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, v. 13, n. 1, p. 146, jan. 2004. ISSN 1017-9909.
- 4 CHAKI, N.; SHAIKH, S. H.; SAEED, K. A Comprehensive Survey on Image Binarization Techniques. *Studies in Computational Intelligence*, v. 560, p. 5–16, 2014. ISSN 1860949X.
- 5 TENSMEYER, C.; MARTINEZ, T. Historical Document Image Binarization: A Review. *SN Computer Science*, Springer Singapore, v. 1, n. 3, p. 1–26, 2020. ISSN 2662-995X.
- 6 GODSE, S. P.; NIMBHORE, S.; SHITOLE, S.; KATKE, D.; KASAR, P. Recovery of badly degraded Document images using Binarization Technique. *International Journal of Scientific and Research Publications*, v. 4, n. 5, p. 433–438, 2014.
- 7 WHITE, J. M.; ROHRER, G. D. Image Thresholding for Optical Character Recognition and Other Applications Requiring Character Image Extraction. *IBM Journal of Research and Development*, v. 27, n. 4, p. 400–411, jul. 1983. ISSN 0018-8646, 0018-8646.
- 8 NIBLACK, W. *An Introduction to Digital Image Processing*. Birkerød, Denmark: Strandberg Publishing Company, 1985.
- 9 BERNSEN, J. Dynamic thresholding of gray-level images. In: *International Conference on Pattern Recognition*. Paris, France: [s.n.], 1986. p. 1251–1255.
- 10 EIKVIL, L.; TAXT, T.; MOEN, K. A fast adaptive method for the binarization of document images. In: *1st International Conference on Document Analysis and Recognition*. [S.l.: s.n.], 1991. v. 1, p. 435–443.
- 11 LINS, R. D. A Taxonomy for Noise in Images of Paper Documents - The Physical Noises. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.]: Springer Berlin Heidelberg, 2009. v. 5627 LNCS, p. 844–854. ISBN 3-642-02610-9.
- 12 SULAIMAN, A.; OMAR, K.; NASRUDIN, M. F. Degraded historical document binarization: A review on issues, challenges, techniques, and future directions. *Journal of Imaging*, v. 5, n. 4, 2019. ISSN 2313433X.
- 13 LINS, R. D.; NETO, M. G.; NETO, L. F.; ROSA, L. G. An environment for processing images of historical documents. *Microprocessing and Microprogramming*, v. 40, n. 10-12, p. 939–942, dez. 1994. ISSN 01656074.
- 14 MELLO, C. A. B.; LINS, R. D. Image segmentation of historical documents. *Visual 2000*, 2000.

- 15 TRIER, O.; JAIN, A. Goal-directed evaluation of binarization methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 17, n. 12, p. 1191–1201, 1995. ISSN 01628828.
- 16 LINS, R. D.; SILVA, J. M. M.; MARTINS, F. M. J. Detailing a quantitative method for assessing algorithms to remove back-to-front interference in documents. *Journal of Universal Computer Science*, v. 14, n. 2, p. 266–283, 2008. ISSN 0958695X.
- 17 STATHIS, P.; KAVALLIERATOU, E.; PAPAMARKOS, N. An Evaluation Technique for Binarization Algorithms. *Journal of Universal Computer Science*, v. 14, n. 18, p. 3011–3030, 2008. ISSN 0958695X.
- 18 NTIROGIANNIS, K.; GATOS, B.; PRATIKAKIS, I. An Objective Evaluation Methodology for Document Image Binarization Techniques. In: *2008 The Eighth IAPR International Workshop on Document Analysis Systems*. [S.l.]: IEEE, 2008. p. 217–224. ISBN 978-0-7695-3337-7.
- 19 GATOS, B.; NTIROGIANNIS, K.; PRATIKAKIS, I. ICDAR 2009 Document Image Binarization Contest (DIBCO 2009). In: *2009 10th International Conference on Document Analysis and Recognition*. [S.l.]: IEEE, 2009. p. 1375–1382. ISBN 978-1-4244-4500-4.
- 20 PRATIKAKIS, I.; ZAGORIS, K.; KARAGIANNIS, X.; TSOCHATZIDIS, L.; MONDAL, T.; WANG, X.; XIONG, W.; LI, M.; WANG, C.; GUAN, L.; XIONG, Z.; LI, M. ICDAR 2019 Competition on Document Image Binarization (DIBCO 2019). In: *2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.: s.n.], 2019.
- 21 PAREDES, R.; KAVALLIERATOU, E.; LINS, R. D. ICFHR 2010 Contest: Quantitative Evaluation of Binarization Algorithms. In: *2010 12th International Conference on Frontiers in Handwriting Recognition*. [S.l.]: IEEE, 2010. p. 733–736. ISBN 978-1-4244-8353-2.
- 22 LINS, R. D.; KAVALLIERATOU, E.; SMITH, E. B.; BERNARDINO, R. B.; JESUS, D. M. de. ICDAR 2019 Time-Quality Binarization Competition. In: *2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. Sydney, Australia: IEEE, 2019. p. 1539–1546. ISBN 978-1-72813-014-9.
- 23 SILVA, G. P.; LINS, R. D. PhotoDoc: A Toolbox for Processing Document Images Acquired Using Portable Digital Cameras. In: *CBDAR 2007*. Curitiba, Brazil: [s.n.], 2007. p. 107–114.
- 24 LINS, R. D.; AVILA, B. T. A New Algorithm for Skew Detection in Images of Documents. *International Conference Image Analysis and Recognition*, Springer, Berlin, Heidelberg, v. 3212, n. 2, p. 234–240, 2004. ISSN 0302-9743.
- 25 DOERMANN, D.; Jian Liang; Huiping Li. Progress in camera-based document image analysis. In: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings*. Edinburgh, UK: IEEE Comput. Soc, 2003. v. 1, p. 606–616. ISBN 978-0-7695-1960-9.
- 26 LINS, R. D.; SILVA, G. E.; Gomes e Silva, A. R. Assessing and Improving the Quality of Document Images Acquired with Portable Digital Cameras. *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2*, p. 569–573, 2007. ISSN 1520-5363.

- 27 LINS, R. D.; BERNARDINO, R. B.; JESUS, D. M. de; OLIVEIRA, J. M. Binarizing Document Images Acquired with Portable Cameras. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.]: IEEE, 2017. p. 45–50. ISBN 978-1-5386-3586-5.
- 28 LINS, R. D.; ALMEIDA, M. M. D.; BERNARDINO, R. B.; JESUS, D.; OLIVEIRA, J. M. Assessing binarization techniques for document images. In: *DocEng 2017 - Proceedings of the 2017 ACM Symposium on Document Engineering*. [S.l.: s.n.], 2017. p. 183–192. ISBN 978-1-4503-4689-4.
- 29 CHATTOPADHYAY, T.; REDDY, V. R.; GARAIN, U. Automatic Selection of Binarization Method for Robust OCR. In: *2013 12th International Conference on Document Analysis and Recognition*. [S.l.]: IEEE, 2013. p. 1170–1174. ISBN 978-0-7695-4999-6.
- 30 MOGHADDAM, R. F.; MOGHADDAM, F. F.; CHERIET, M. Unsupervised Ensemble of Experts (EoE) Framework for Automatic Binarization of Document Images. In: *2013 12th International Conference on Document Analysis and Recognition*. [S.l.]: IEEE, 2013. p. 703–707. ISBN 978-0-7695-4999-6. ISSN 15205363.
- 31 LINS, R. D.; SIMSKE, S. J.; BERNARDINO, R. B. DocEng'2020 Time-Quality Competition on Binarizing Photographed Documents. In: *Proceedings of the ACM Symposium on Document Engineering, DocEng 2020*. New York, NY, USA: ACM, 2020. p. 1–4. ISBN 978-1-4503-8000-3.
- 32 LINS, R. D.; BERNARDINO, R. B.; da Silva Barboza, R.; LINS, Z. D. Direct binarization a quality-and-time efficient binarization strategy. In: *Proceedings of the 21st ACM Symposium on Document Engineering*. New York, NY, USA: ACM, 2021. v. 1, p. 1–4. ISBN 978-1-4503-8596-1.
- 33 LINS, R. D.; BERNARDINO, R. B.; SMITH, E. B.; KAVALLIERATOU, E. ICDAR 2021 Competition on Time-Quality Document Image Binarization. In: *ICDAR 2021 Competition on Time-Quality Document Image Binarization*. [S.l.: s.n.], 2021. p. 708–722.
- 34 LINS, R. D.; SIMSKE, S. J.; BERNARDINO, R. B. Binarisation of photographed documents image quality and processing time assessment. In: *Proceedings of the 21st ACM Symposium on Document Engineering*. New York, NY, USA: ACM, 2021. v. 1, p. 1–6. ISBN 978-1-4503-8596-1.
- 35 LINS, R. D.; BERNARDINO, R.; BARBOZA, R. d. S.; OLIVEIRA, R. C. D. Using Paper Texture for Choosing a Suitable Algorithm for Scanned Document Image Binarization. *Journal of Imaging*, v. 8, n. 10, p. 272, out. 2022. ISSN 2313-433X.
- 36 LINS, R. D.; BERNARDINO, R. B.; BARBOZA, R.; OLIVEIRA, R. The Winner Takes It All: Choosing the “best” Binarization Algorithm for Photographed Documents. In: UCHIDA, S.; BARNEY, E.; EGLIN, V. (Ed.). *Document Analysis Systems*. Cham: Springer International Publishing, 2022. v. 13237, p. 48–64. ISBN 978-3-031-06554-5 978-3-031-06555-2.
- 37 BERNARDINO, R.; LINS, R. D.; BARBOZA, R. d. S. A Quality, Size and Time Assessment of the Binarization of Documents Photographed by Smartphones. *Journal of Imaging*, v. 9, n. 2, p. 41, fev. 2023. ISSN 2313-433X.

- 38 BERNARDINO, R.; LINS, R. D.; BARBOZA, R. Texture-based Document Binarization. In: *Proceedings of the 22st ACM Symposium on Document Engineering*. [S.l.]: ACM, 2024. ISBN 979-8-4007-1169-5/24/08.
- 39 MOGHADDAM, R. F.; MOHAMED, C. AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization. *Pattern Recognition*, Elsevier, v. 45, n. 6, p. 2419–2431, jun. 2012. ISSN 00313203.
- 40 SADDAMI, K.; MUNADI, K.; AWAY, Y.; ARNIA, F. Improvement of binarization performance using local otsu thresholding. *International Journal of Electrical and Computer Engineering*, v. 9, n. 1, p. 264–272, 2019. ISSN 20888708.
- 41 HE, S.; SCHOMAKER, L. DeepOtsu: Document Enhancement and Binarization using Iterative Deep Learning. *Pattern Recognition*, v. 91, p. 379–390, jan. 2019. ISSN 00313203.
- 42 KAPUR, J.; SAHOO, P.; WONG, A. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, v. 29, n. 1, p. 140, jan. 1985. ISSN 0734189X.
- 43 PUN, T. Entropic thresholding, a new approach. *Computer Graphics and Image Processing*, v. 16, n. 3, p. 210–239, 1981. ISSN 0146664X.
- 44 SAXENA, L. P. Niblack's binarization method and its modifications to real-time applications: A review. *Artificial Intelligence Review*, Springer Netherlands, v. 51, n. 4, p. 673–705, 2019. ISSN 0269-2821.
- 45 SAUVOLA, J.; PIETIKÄINEN, M.; PIETIKAINEN, M. Adaptive document image binarization. *Pattern Recognition*, v. 33, n. 2, p. 225–236, 2000. ISSN 00313203.
- 46 WOLF, C.; JOLION, J.-M.; CHASSAING, F. Text localization, enhancement and binarization in multimedia documents. In: *Object Recognition Supported by User Interaction for Service Robots*. [S.l.]: IEEE Comput. Soc, 2003. v. 2, p. 1037–1040. ISBN 0-7695-1695-X.
- 47 KHURSHID, K.; SIDDIQI, I.; FAURE, C.; VINCENT, N. Comparison of Niblack inspired binarization methods for ancient documents. In: *SPIE 7247*. [S.l.: s.n.], 2009. p. 72470U. ISSN 0277786X.
- 48 HOWE, N. R. Document binarization with automatic parameter tuning. *International Journal on Document Analysis and Recognition (IJ DAR)*, v. 16, n. 3, p. 247–258, set. 2013. ISSN 1433-2833.
- 49 WESTPHAL, F.; GRAHN, H.; LAVESSON, N. Efficient document image binarization using heterogeneous computing and parameter tuning. *International Journal on Document Analysis and Recognition*, Springer Berlin Heidelberg, v. 21, n. 1-2, p. 41–58, 2018. ISSN 14332825.
- 50 SADDAMI, K.; MUNADI, K.; MUCHALLIL, S.; ARNIA, F. Improved Thresholding Method for Enhancing Jawi Binarization Performance. In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.]: IEEE, 2017. v. 1, p. 1108–1113. ISBN 978-1-5386-3586-5. ISSN 15205363.
- 51 CHAN, C. Memory-efficient and fast implementation of local adaptive binarization methods. *arXiv preprint arXiv:1905.13038*, maio 2019.

- 52 SU, B.; LU, S.; TAN, C. L. Combination of Document Image Binarization Techniques. In: *2011 International Conference on Document Analysis and Recognition*. [S.l.]: IEEE, 2011. p. 22–26. ISBN 978-1-4577-1350-7.
- 53 ZHOU, L.; ZHANG, C.; WU, M. D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, v. 2018-June, p. 192–196, 2018. ISSN 21607516.
- 54 XIONG, W.; YUE, L.; ZHOU, L.; WEI, L.; LI, M. FD-Net: A Fully Dilated Convolutional Network for Historical Document Image Binarization. In: MA, H.; WANG, L.; ZHANG, C.; WU, F.; TAN, T.; WANG, Y.; LAI, J.; ZHAO, Y. (Ed.). *Pattern Recognition and Computer Vision*. Cham: Springer International Publishing, 2021. v. 13019, p. 518–529. ISBN 978-3-030-88003-3 978-3-030-88004-0.
- 55 MICHALAK, H.; OKARMA, K. Fast binarization of unevenly illuminated document images based on background estimation for optical character recognition purposes. *Journal of Universal Computer Science*, v. 25, n. 6, p. 627–646, 2019. ISSN 09486968.
- 56 LINS, R. D.; BERNARDINO, R. B.; BARBOZA, R. d. S.; SIMSKE, S. J. Binarization of photographed documents image quality, processing time and size assessment. In: *Proceedings of the 22nd ACM Symposium on Document Engineering*. San Jose California: ACM, 2022. p. 1–10. ISBN 978-1-4503-9544-1.
- 57 STATHIS, P.; KAVALLIERATOU, E.; PAPAMARKOS, N. An evaluation survey of binarization algorithms on historical documents. In: *2008 19th International Conference on Pattern Recognition*. [S.l.]: IEEE, 2008. v. 393, p. 1–4. ISBN 978-1-4244-2174-9. ISSN 1051-4651.
- 58 ISMAIL, S. M.; ABDULLAH, S. N. H. S.; FAUZI, F. Statistical binarization techniques for document image analysis. *Journal of Computer Science*, v. 14, n. 1, p. 23–36, jan. 2018. ISSN 15493636.
- 59 LI, D.; WU, Y.; ZHOU, Y. SauvolaNet: Learning Adaptive Sauvola Network for Degraded Document Binarization. In: LLADÓS, J.; LOPRESTI, D.; UCHIDA, S. (Ed.). *Document Analysis and Recognition – ICDAR 2021*. Cham: Springer International Publishing, 2021. v. 12824, p. 538–553. ISBN 978-3-030-86336-4 978-3-030-86337-1.
- 60 SADDAMI, K.; AFRAH, P.; MUTIAWANI, V.; ARNIA, F. A New Adaptive Thresholding Technique for Binarizing Ancient Document. In: *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*. [S.l.]: IEEE, 2018. p. 57–61. ISBN 978-1-5386-9422-0.
- 61 CANNY, J. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, n. 6, p. 679–698, 1986. ISSN 01628828.
- 62 JIA, F.; SHI, C.; HE, K.; WANG, C.; XIAO, B. Degraded document image binarization using structural symmetry of strokes. *Pattern Recognition*, Elsevier Ltd, v. 74, p. 225–240, fev. 2018. ISSN 00313203.
- 63 AKBARI, Y.; BRITTO~JR., A. S.; Al-Maadeed, S.; OLIVEIRA, L. S. Binarization of Degraded Document Images using Convolutional Neural Networks based on predicted

Two-Channel Images. In: *International Conference on Document Analysis and Recognition*. [S.l.: s.n.], 2019.

64 SU, B.; LU, S.; TAN, C. L. Robust Document Image Binarization Technique for Degraded Document Images. *IEEE Transactions on Image Processing*, v. 22, n. 4, p. 1408–1417, abr. 2013. ISSN 1057-7149.

65 SU, B.; LU, S.; TAN, C. L. A learning framework for degraded document image binarization using Markov Random Field. *Proceedings - International Conference on Pattern Recognition*, IEEE, n. Icpr, p. 3200–3203, 2012. ISSN 10514651.

66 PENG, X.; SETLUR, S.; GOVINDARAJU, V.; SITARAM, R. Markov random field based binarization for hand-held devices captured document images. *Proceedings of the Seventh Indian Conference on Computer Vision, Graphics and Image Processing*, n. i, p. 71–76, 2010.

67 LELORE, T.; BOUCHARA, F. Document Image Binarisation Using Markov Field Model. In: *2009 10th International Conference on Document Analysis and Recognition*. [S.l.]: IEEE, 2009. p. 551–555. ISBN 978-1-4244-4500-4. ISSN 15205363.

68 KUK, J. G.; CHO, N. I.; LEE, K. M. MAP-MRF approach for binarization of degraded document image. In: *2008 15th IEEE International Conference on Image Processing*. [S.l.]: IEEE, 2008. p. 2612–2615. ISBN 978-1-4244-1765-0.

69 WOLF, C.; DOERMANN, D. Binarization of low quality text using a Markov random field model. In: *Object Recognition Supported by User Interaction for Service Robots*. [S.l.]: IEEE Comput. Soc, 2002. v. 3, p. 160–163. ISBN 0-7695-1695-X. ISSN 10514651.

70 PRATIKAKIS, I.; ZAGORIS, K.; BARLAS, G.; GATOS, B. ICDAR2017 Competition on Document Image Binarization (DIBCO 2017). In: *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.]: IEEE, 2017. v. 1, p. 1395–1403. ISBN 978-1-5386-3586-5. ISSN 15205363.

71 PRATIKAKIS, I.; ZAGORI, K.; KADDAS, P.; GATOS, B. ICFHR 2018 Competition on Handwritten Document Image Binarization (H-DIBCO 2018). In: *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. [S.l.]: IEEE, 2018. p. 489–493. ISBN 978-1-5386-5875-8.

72 LINS, R. D.; BERNARDINO, R.; JESUS, D. M. A Quality and Time Assessment of Binarization Algorithms. In: *2019 International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.]: IEEE, 2019. p. 1444–1450. ISBN 978-1-72813-014-9.

73 MICHALAK, H.; OKARMA, K. Improvement of Image Binarization Methods Using Image Preprocessing with Local Entropy Filtering for Alphanumeric Character Recognition Purposes. *Entropy*, v. 21, n. 6, p. 562, jun. 2019. ISSN 1099-4300.

74 MICHALAK, H.; OKARMA, K. Adaptive Image Binarization Based on Multi-layered Stack of Regions. In: VENTO, M.; PERCANNELLA, G. (Ed.). *Computer Analysis of Images and Patterns*. Cham: Springer International Publishing, 2019. v. 11679, p. 281–293. ISBN 978-3-030-29890-6 978-3-030-29891-3.

75 CALVO-ZARAGOZA, J.; GALLEGO, A.-J. A selectional auto-encoder approach for document image binarization. *Pattern Recognition*, Elsevier Ltd, v. 86, p. 37–47, fev. 2019. ISSN 00313203.

- 76 VO, Q. N.; KIM, S. H.; YANG, H. J.; LEE, G. Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition*, Elsevier Ltd, v. 74, p. 568–586, 2018. ISSN 00313203.
- 77 BHOWMIK, S.; SARKAR, R.; DAS, B.; DOERMANN, D. GiB: A Game Theory Inspired Binarization Technique for Degraded Document Images. *IEEE Transactions on Image Processing*, v. 28, n. 3, p. 1443–1455, mar. 2019. ISSN 1057-7149.
- 78 AZAD, R.; Asadi-Aghbolaghi, M.; FATHY, M.; ESCALERA, S. Bi-directional ConvLSTM U-net with densely connected convolutions. *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, p. 406–415, 2019.
- 79 KONG, X.; SUN, G.; WU, Q.; LIU, J.; LIN, F. Hybrid pyramid u-net model for brain tumor segmentation. In: *International Conference on Intelligent Information Processing*. [S.l.]: Springer, 2018. p. 346–355.
- 80 BERA, S. K.; GHOSH, S.; BHOWMIK, S.; SARKAR, R.; NASIPURI, M. A non-parametric binarization method based on ensemble of clustering algorithms. *Multimedia Tools and Applications*, v. 80, n. 5, p. 7653–7673, fev. 2021.
- 81 AKBARI, Y.; Al-Maadeed, S.; ADAM, K. Binarization of Degraded Document Images Using Convolutional Neural Networks and Wavelet-Based Multichannel Images. *IEEE Access*, v. 8, p. 153517–153534, 2020. ISSN 2169-3536.
- 82 SOUIBGUI, M. A.; KESSENTINI, Y. DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.
- 83 OLIVEIRA, S. A.; SEGUIN, B.; KAPLAN, F. dhSegment: {A} generic deep-learning approach for document segmentation. *CoRR*, abs/1804.1, 2018.
- 84 DOYLE, W. Operations Useful for Similarity-Invariant Pattern Recognition. *Journal of the ACM*, v. 9, n. 2, p. 259–267, abr. 1962. ISSN 00045411.
- 85 ZACK, G. W.; ROGERS, W. E.; LATT, S. A. Automatic measurement of sister chromatid exchange frequency. *The Journal of Histochemistry and Cytochemistry*, v. 25, n. 7, p. 741–753, 1977. ISSN 0022-1554.
- 86 VELASCO, F. R. *Thresholding Using the Isodata Clustering Algorithm*. University of Maryland, Maryland, 1979. 14 p.
- 87 JOHANNSEN, G.; BILLE, J. A threshold selection method using information measures. In: *Int'l Conf. Pattern Recognition*. Munich, Germany: [s.n.], 1982. p. 140–143.
- 88 TSAI, W.-H. Moment-preserving thresholding: A new approach. *Computer Vision, Graphics, and Image Processing*, v. 29, n. 3, p. 377–393, 1985. ISSN 0734189X.
- 89 KITTLER, J.; ILLINGWORTH, J. Minimum error thresholding. *Pattern Recognition*, v. 19, n. 1, p. 41–47, jan. 1986. ISSN 00313203.
- 90 GLASBEY, C. An Analysis of Histogram-Based Thresholding Algorithms. *Graphical Models and Image Processing*, v. 55, n. 6, p. 532–537, nov. 1993. ISSN 10773169.

- 91 SHANBHAG, A. G. Utilization of Information Measure as a Means of Image Thresholding. *CVGIP: Graphical Models and Image Processing*, v. 56, n. 5, p. 414–419, 1994. ISSN 10499652.
- 92 HUANG, L. K.; WANG, M. J. J. Image thresholding by minimizing the measures of fuzziness. *Pattern Recognition*, v. 28, n. 1, p. 41–51, 1995. ISSN 00313203.
- 93 CHANG, F. J. C. S. Y. J. C.; YEN, J. C.; CHANG, F. J.; CHANG, S. A New Criterion for Automatic Multilevel Thresholding. *IEEE Transactions on Image Processing*, v. 4, n. 3, p. 370–378, 1995. ISSN 19410042.
- 94 SAHOO, P.; WILKINS, C.; YEAGER, J. Threshold selection using Renyi's entropy. *Pattern Recognition*, v. 30, n. 1, p. 71–84, 1997. ISSN 00313203.
- 95 SAUVOLA, J.; SEPPANEN, T.; HAAPAKOSKI, S.; PIETIKAINEN, M. Adaptive document binarization. In: *Proceedings of the Fourth International Conference on Document Analysis and Recognition*. [S.l.]: IEEE Comput. Soc, 1997. v. 1, p. 147–152. ISBN 0-8186-7898-4.
- 96 LI, C.; TAM, P. An iterative algorithm for minimum cross entropy thresholding. *Pattern Recognition Letters*, v. 19, n. 8, p. 771–776, 1998. ISSN 01678655.
- 97 LU, W.; SONGDE, M.; LU, H. An effective entropic thresholding for ultrasonic images. *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, p. 1552–1554, vol. 2, 1998.
- 98 HADJADJ, Z.; MEZIANE, A.; CHERFA, Y.; CHERIET, M.; SETITRA, I. ISauvola: Improved Sauvola's Algorithm for Document Image Binarization. In: CAMPILHO, A.; KAMEL, M. (Ed.). Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. v. 3212, p. 737–745. ISBN 978-3-540-23240-7.
- 99 KAVALLIERATOU, E. A binarization algorithm specialized on document images and photos. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, v. 2005, n. 1, p. 463–467, 2005. ISSN 15205363.
- 100 KAVALLIERATOU, E.; STATHIS, S. Adaptive binarization of historical document images. *Proceedings - International Conference on Pattern Recognition*, v. 3, p. 742–745, 2006. ISSN 10514651.
- 101 PREWITT, J. M. S.; MENDELSON, M. L. The Analysis of Cell Images. *Annals of the New York Academy of Sciences*, v. 128, n. 3, p. 1035–1053, dez. 2006. ISSN 00778923.
- 102 SILVA, J. M. M.; LINS, R. D.; ROCHA, V. C. Binarizing and Filtering Historical Documents with Back-to-Front Interference. In: *Proceedings of the 2006 ACM Symposium on Applied Computing*. Dijon, France: [s.n.], 2006. p. 853–858. ISBN 1-59593-108-2.
- 103 BRADLEY, D.; ROTH, G. Adaptive Thresholding using the Integral Image. *Journal of Graphics Tools*, v. 12, n. 2, p. 13–21, jan. 2007. ISSN 1086-7651.
- 104 SMITH, E. H. B.; LIKFORMAN-SULEM, L.; DARBON, J. Effect of pre-processing on binarization. In: LIKFORMAN-SULEM, L.; AGAM, G. (Ed.). *Document Recognition and Retrieval XVII*. [S.l.: s.n.], 2010. v. 7534, p. 75340H. ISBN 978-0-8194-7927-3. ISSN 0277786X.

- 105 LU, S.; SU, B.; TAN, C. L. Document image binarization using background estimation and stroke edges. *International Journal on Document Analysis and Recognition (IJDAR)*, v. 13, n. 4, p. 303–314, dez. 2010. ISSN 1433-2833.
- 106 BATAINEH, B.; ABDULLAH, S. N. H. S.; OMAR, K. An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. *Pattern Recognition Letters*, Elsevier B.V., v. 32, n. 14, p. 1805–1813, out. 2011. ISSN 01678655.
- 107 SINGH, T. R.; ROY, S.; SINGH, O. I.; SINAM, T.; SINGH, K. M. A New Local Adaptive Thresholding Technique in Binarization. *IJCSI International Journal of Computer Science Issues*, v. 08, n. 6, p. 271–277, dez. 2011. ISSN 1694-0814.
- 108 GATTAL, A.; ABBAS, F.; LAOUAR, M. R. Automatic Parameter Tuning of K-Means Algorithm for Document Binarization. In: *Proceedings of the 7th International Conference on Software Engineering and New Technologies - ICSENT 2018*. New York, New York, USA: ACM Press, 2018. p. 1–4. ISBN 978-1-4503-6101-9.
- 109 MUSTAFA, W. A.; KADER, M. M. M. A. Binarization of Document Image Using Optimum Threshold Modification. *Journal of Physics: Conference Series*, v. 1019, n. 1, p. 012022, jun. 2018. ISSN 1742-6588.
- 110 SADDAMI, K.; MUNADI, K.; AWAY, Y.; ARNIA, F. Effective and fast binarization method for combined degradation on ancient documents. *Heliyon*, 2019. ISSN 24058440.
- 111 XIONG, W.; ZHOU, L.; YUE, L.; LI, L.; WANG, S. An enhanced binarization framework for degraded historical document images. *EURASIP Journal on Image and Video Processing*, v. 2021, n. 1, p. 13, dez. 2021. ISSN 1687-5281.
- 112 LEE, S. U.; CHUNG, S. Y.; PARK, R. H. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing*, v. 52, n. 2, p. 171–190, nov. 1990. ISSN 0734189X.
- 113 LEEDHAM, G.; Chen Yan; TAKRU, K.; Joie Hadi Nata Tan; Li Mian. Comparison of some thresholding algorithms for text/background segmentation in difficult document images. In: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings*. [S.l.]: IEEE Comput. Soc, 2003. v. 1, p. 859–864. ISBN 0-7695-1960-1. ISSN 15205363.
- 114 LU, H.; KOT, A. C.; SHI, Y. Q. Distance-Reciprocal Distortion Measure for Binary Document Images. *IEEE Signal Processing Letters*, v. 11, n. 2, p. 228–231, 2004. ISSN 1070-9908.
- 115 KEFALI, A.; SARI, T.; SELLAMI, M. Evaluation of several binarization techniques for old Arabic documents images. *The First International Symposium on Modeling and Implementing Complex Systems MISC*, n. 1, p. 88–99, 2010.
- 116 NTIROGIANNIS, K.; GATOS, B.; PRATIKAKIS, I. Performance Evaluation Methodology for Historical Document Image Binarization. *IEEE Transactions on Image Processing*, v. 22, n. 2, p. 595–609, fev. 2013. ISSN 1057-7149.

- 117 ŞEKEROĞLU, B.; KHASHMAN, A. Performance Evaluation of Binarization Methods for Document Images. In: *Proceedings of the International Conference on Advances in Image Processing*. New York, NY, USA: ACM, 2017. Part F1312, p. 96–102. ISBN 978-1-4503-5295-6.
- 118 LINS, R. D.; SILVA, G. F. P.; FORMIGA, A. A. HistDoc v. 2.0 Enhancing a Platform to Process Historical Documents. *Historical Document Imaging and Processing*, p. 169–176, 2011.
- 119 LINS, R. D.; TORREÃO, G.; SILVA, G. P. E. Content Recognition and Indexing in the LiveMemory Platform. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 6020 LNCS, p. 220–230, 2010. ISSN 03029743.
- 120 CLAUSNER, C.; PAPADOPOULOS, C.; PLETSCHACHER, S.; ANTONACOPOULOS, A. The ENP image and ground truth dataset of historical newspapers. In: *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. [S.l.]: IEEE, 2015. p. 931–935.
- 121 POWERS, D. M. W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011. ISSN 2229-3981.
- 122 CONGALTON, R. G. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, v. 37, n. 1, p. 35–46, jul. 1991. ISSN 00344257.
- 123 MEHRI, M.; HÉROUX, P.; Gomez-Krämer, P.; MULLOT, R. Texture feature benchmarking and evaluation for historical document image analysis. *International Journal on Document Analysis and Recognition (IJDAR)*, v. 20, n. 1, p. 1–35, mar. 2017. ISSN 1433-2833, 1433-2825.
- 124 BEYERER, J.; LEÓN, F. P.; FRESE, C. Texture Analysis. In: *Machine Vision*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016. p. 649–683. ISBN 978-3-662-47793-9 978-3-662-47794-6.
- 125 BARBOZA, R. d. S.; LINS, R. D.; JESUS, D. M. de. A Color-Based Model to Determine the Age of Documents for Forensic Purposes. In: *2013 12th International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE, 2013. p. 1350–1354. ISBN 978-0-7695-4999-6.
- 126 ALAEI, F.; ALAEI, A.; PAL, U.; BLUMENSTEIN, M. A comparative study of different texture features for document image retrieval. *Expert Systems with Applications*, v. 121, p. 97–114, maio 2019. ISSN 09574174.
- 127 BIANCONI, F.; FERNÁNDEZ, A.; SMERALDI, F.; PASCOLETTI, G. Colour and Texture Descriptors for Visual Recognition: A Historical Overview. *Journal of Imaging*, v. 7, n. 11, p. 245, nov. 2021. ISSN 2313-433X.
- 128 MEHRI, M.; HÉROUX, P.; Gomez-Krämer, P.; MULLOT, R. Texture feature benchmarking and evaluation for historical document image analysis. *International Journal on Document Analysis and Recognition (IJDAR)*, v. 20, n. 1, p. 1–35, mar. 2017. ISSN 1433-2833, 1433-2825.

- 129 FERNÁNDEZ, A.; ÁLVAREZ, M. X.; BIANCONI, F. Texture Description Through Histograms of Equivalent Patterns. *Journal of Mathematical Imaging and Vision*, v. 45, n. 1, p. 76–102, jan. 2013. ISSN 0924-9907, 1573-7683.
- 130 ALAEI, A.; CONTE, D.; BLUMENSTEIN, M.; RAVEAUX, R. Document Image Quality Assessment Based on Texture Similarity Index. In: *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. Santorini, Greece: IEEE, 2016. p. 132–137. ISBN 978-1-5090-1792-8.
- 131 FERNÁNDEZ, A.; ÁLVAREZ, M. X.; BIANCONI, F. Image classification with binary gradient contours. *Optics and Lasers in Engineering*, v. 49, n. 9-10, p. 1177–1184, set. 2011. ISSN 01438166.
- 132 SILVA, A. R. G.; LINS, R. D. Background Removal of Document Images Acquired Using Portable Digital Cameras. In: *Image Analysis and Recognition*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. v. 3656, p. 278–285. ISBN 978-3-540-29069-8 978-3-540-31938-2.
- 133 NUNNAGOPPULA, G.; DEEPAK, K. S.; HARIKRISHNA, G.; RAI, N.; KRISHNA, P. R.; VESDAPUNT, N. Automatic blur detection in mobile captured document images: Towards quality check in mobile based document imaging applications. In: *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*. [S.l.]: IEEE, 2013. p. 299–304. ISBN 978-1-4673-6101-9.
- 134 FAN, J.; LIN, Q.; LIU, J. Mobile document scanning and copying. *Proceedings of the international conference on Multimedia*, v. 9700, p. 1531–1532, 2010.
- 135 MILYAEV, S.; BARINOVA, O.; NOVIKOVA, T.; KOHLI, P.; LEMPITSKY, V. Image Binarization for End-to-End Text Understanding in Natural Images. In: *2013 12th International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE, 2013. p. 128–132. ISBN 978-0-7695-4999-6.
- 136 SIMON, C.; CHOE, J.; YUN, I. D.; PARK, I. K. Correcting Photometric Distortion of Document Images on a Smartphone. *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, n. 3, p. 199–200, 2014. ISSN 21607516.
- 137 SINGH, B. M.; SHARMA, R.; GHOSH, D.; MITTAL, A. Adaptive binarization of severely degraded and non-uniformly illuminated documents. *International Journal on Document Analysis and Recognition*, v. 17, n. 4, 2014. ISSN 14332825.
- 138 LEVENSHTAIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, v. 10, n. 8, p. 707–710, 1966.
- 139 MICHALAK, H.; OKARMA, K. Robust combined binarization method of non-uniformly illuminated document images for alphanumerical character recognition. *Sensors (Switzerland)*, MDPI AG, v. 20, n. 10, maio 2020. ISSN 14248220.
- 140 ROBINSON, A.; CHERRY, C. Results of a prototype television bandwidth compression scheme. *Proceedings of the IEEE*, v. 55, n. 3, p. 356–364, 1967. ISSN 0018-9219.