ZigZag: A Robust Adaptive Approach to Non-Uniformly Illuminated Document Image Binarization

Jean-Luc Bloechle
CoPeLab Group, Faculty of Science and Medicine, University of Fribourg
Fribourg, Switzerland
jean-luc.bloechle@unifr.ch

Jean Hennebert
iCoSys Institute, University of Applied Science Western Switzerland
Fribourg, Switzerland
jean.hennebert@hefr.ch

Christophe Gisler
iCoSys Institute, University of Applied Science Western Switzerland
Fribourg, Switzerland
christophe.gisler@hefr.ch

Abstract

In the era of mobile imaging, the quality of document photos captured by smartphones often suffers due to adverse lighting conditions. Traditional document analysis and optical character recognition systems encounter difficulties with images that have not been effectively binarized, particularly under challenging lighting scenarios. This paper introduces a novel adaptive binarization algorithm optimized for such difficult lighting environments. Unlike many existing methods that rely on complex machine learning models, our approach is streamlined and machine-learning free, designed around integral images to significantly reduce computational and coding complexities. This approach enhances processing speed and improves accuracy without the need for computationally expensive training procedures. Comprehensive testing across various datasets, from smartphone-captured documents to historical manuscripts, validates its effectiveness. Moreover, the introduction of versatile output modes, including color foreground extraction, substantially enhances document quality and readability by effectively eliminating unwanted background artifacts. These enhancements are valuable in mobile document image processing across industries that prioritize efficient and accurate document management, spanning sectors such as banking, insurance, education, and archival management.

CCS Concepts

• **Computing methodologies** → **Image processing**.

Keywords

binarization, image thresholding, image processing, OCR

ACM Reference Format:
Jean-Luc Bloechle, Jean Hennebert, and Christophe Gisler. 2024. ZigZag: A Robust Adaptive Approach to Non-Uniformly Illuminated Document Image Binarization. In *ACM Symposium on Document Engineering 2024 (DocEng '24), August 20–23, 2024, San Jose, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3685650.3685661

DocEng '24, August 20–23, 2024, San Jose, CA, USA

ACM ISBN 979-8-4007-1169-5/24/08
https://doi.org/10.1145/3685650.3685661

1 Introduction

The increasing reliance on smartphones for document capture and management brings challenges due to variable lighting conditions, which often degrade image quality and hinder optical character recognition (OCR) systems. Unlike cloud-based OCR technologies that process images in grayscale, embedded OCR systems in mobile devices require high-contrast black-and-white inputs. This underscores the importance of effective image binarization to enhance readability and usability, particularly in suboptimal conditions like challenging lighting environments.

This paper introduces ZigZag, a binarization algorithm that builds upon our previous work with the YinYang method. Tailored for challenging lighting conditions, ZigZag significantly simplifies the algorithmic framework for improved efficiency and comprehension. By leveraging integral images, ZigZag reduces computational requirements, speeds up processing, and enhances binarization accuracy compared to its predecessor.

ZigZag distinguishes itself as a machine-learning-free solution, offering numerous advantages over machine-learning-based approaches. Its simplicity facilitates easier implementation and comprehension, simplifying debugging, maintenance, and adoption by users. Furthermore, its independence from complex machine learning models reduces computational overhead, making it more lightweight and efficient for resource-constrained environments like mobile devices.

Evaluation of ZigZag in various contexts demonstrates its capability to consistently produce high-quality outputs, enhancing the performance of OCR systems and facilitating document analysis.

This paper is organized as follows: Section 2 provides a comprehensive review of existing binarization algorithms, setting the stage for a comparative analysis of ZigZag's advancements. Section 3 details the methodology and technical implementation of the ZigZag algorithm. Section 4 presents the results of experimental validations, where ZigZag is benchmarked against other non-machine-learning-based adaptive binarization techniques. Additionally, the discussion within Section 4 explores the limitations of the current approach and suggests potential directions for future research. Finally, Section 5 concludes the paper by discussing the broader implications of our findings.

2 Overview of Image Binarization Algorithms

Many algorithms have been proposed for document image binarization. In situations where lighting conditions are controlled, such as with 2D flatbed scanners, applying a simple global thresholding method has proven to be effective. The Otsu [26] method, for

example, is particularly well-suited for providing a single, precise intensity threshold by identifying the value that minimizes intra-class variance, thereby separating the pixels into two distinct classes.

When dealing with document images captured under uncontrolled lighting conditions, more sophisticated methods become necessary to account for local variations in brightness. Bernsen [5] introduced a straightforward adaptive binarization technique based on averaging the minimum and maximum values within a window surrounding the analyzed pixel. Another adaptive approach, proposed by Niblack [16], determines the threshold value based on the mean and standard deviation computed within a local window surrounding the pixel of interest.

Sauvola's algorithm [29] builds upon Niblack's method by incorporating the dynamic range of the image's gray value standard deviation to compute the threshold. However, Sauvola's results can degrade significantly when applied to images where the foreground and background gray values are close. To address this limitation, Wolf et al. [36] proposed normalizing Sauvola's algorithm using the overall image contrast and the mean gray value.

Similarly, Feng's [11] algorithm normalizes Sauvola's method by incorporating image contrast and mean gray value, using a second, larger window that encompasses the first. Before applying Sauvola's algorithm, Gatos et al. [12] suggest preprocessing the image with a low-pass Wiener filter. This step is useful for removing noisy areas, smoothing the background texture, and improving the image contrast.

Several adaptive thresholding techniques based on local Otsu thresholds have also been proposed. AdOtsu [23], for example, combines background estimation with an adaptive Otsu grid. An original modification suggested by Chou et al. [9] combines Otsu's local thresholding with the additional application of support vector machines for background regions.

However, adaptive binarization methods are prone to computational complexity issues due to the need to analyze each pixel's neighborhood to compute local thresholds. To overcome this problem, Bradley [7] introduced a real-time adaptive thresholding technique using the concept of integral images [10]. Bradley's method guarantees a constant number of operations per local window, meaning that the computational cost does not increase with the size of the window, with only a linear amount of preprocessing required.

Shafait et al. [30] combined the statistical constraints of Sauvola's method with integral images. While Bardozzo et al. [3] designed a modified version of image integral calculation for fuzzy integrals that can be used as a grayscale processing tool in real-time and deep learning applications. Michalak et al. [21] introduced another real-time method based on background image subtraction. They used bilinear downsampling and upsampling to retain only low-frequency image data, representing the overall brightness distribution, to estimate the background.

Some binarization algorithms focus on exploiting color information in the image. Tsai and Lee [34], for example, introduced a method suitable for binarizing color images, taking advantage of luminance and saturation characteristics. Tabbone and Wendling [32] adapted an iterative possibilistic c-means algorithm by incorporating a fuzzy entropy criterion to split the membership function into two clusters: background and object. Mysore et al. [25] applied a segmentation technique based on a mean shift algorithm at different image scales, as well as a contrast-enhanced variant of Niblack's thresholding method, specifically designed to binarize deteriorated color document images.

Convolutional Neural Networks (CNNs) have achieved remarkable success in image classification, providing significant improvements over conventional techniques in a wide range of applications. CNNs have also been applied in the field of image binarization. For example, DeepOtsu [14], introduced the concept of training an iterative deep learning neural network to improve input images by removing noise and rectifying various forms of degradation. Subsequently, the cleaned image undergoes binarization using Otsu's global thresholding method. The winner of the DIBCO'17 competition [28] uses the U-Net convolutional network architecture for accurate pixel classification. Tensmeyer et al. [33] applied a fully convolutional neural network to multiple image scales, whereas Peng et al. [27] and Calvo-Zaragoza et al. [8] adopted a deep encoder-decoder architecture to address the challenges of image binarization. A hierarchical deep supervised network was proposed by Vo et al. [35] for the binarization of degraded document images, which achieved state-of-the-art performance on various benchmark datasets. DP-LinkNet [37], a convolutional network for historical document images, builds on the deep learning LinkNet architecture to handle complex scenarios effectively. Similarly, Mondal et al. [24], in their work on Deep Semantic Binarization, also employed the LinkNet architecture and demonstrated superior performance on various datasets, including mobile-captured whiteboard and glass board images. Koloda and Wang [17] introduced a context-aware binarization algorithm that leverages multi-scale Sauvola thresholds with a featurewise attention module.

Combined binarization methods have also been proposed. Su et al. [31] introduced a classification system to combine various thresholding methods, thus improving the overall performance of document image binarization. Their system classifies document image pixels into three sets: foreground pixels, background pixels, and uncertain pixels. It then uses a final classifier to iteratively sort uncertain pixels into either foreground or background. Hebert et al. [15] developed a system based on Conditional Random Field (CRF), combining the strengths of six distinct algorithms. Badekas et al. [2] proposed a system that generates a binarized image by aggregating the results of seven different binarization algorithms, using a Kohonen Self Organizing Map neural network.

Despite advancements, current image binarization methods still face significant challenges when dealing with document images captured under uncontrolled real-world lighting conditions. The variability among document types underscores that no single solution is universally applicable; different document classes require specialized binarization approaches. Tackling uncontrolled lighting conditions demands different strategies than addressing historical images with complex backgrounds, such as textured, structured, or damaged surfaces. This complexity highlights the need for developing new methods that can adapt more effectively to diverse environments and enhance overall binarization performance. Additionally, it's worth noting that while machine learning methods dominate the field today, there remains room for improvement and innovation in non-machine learning approaches.

3 Proposed Method

This paper introduces a document image binarization method specifically designed for images captured under suboptimal lighting conditions. It is based on two fundamental assumptions: first, that the foreground, predominantly text, is typically darker than the background; and second, that background pixels locally outnumber foreground pixels. Emphasizing locality is crucial, as real-world images often show significant brightness fluctuations due to inconsistent lighting conditions or shadows.

To address scenarios where the text appears lighter than its surroundings, such as white text on a dark background, a preprocessing step could be integrated. This step would detect such cases and, if necessary, invert the colors to better suit the binarization process.

Building upon our prior research with the YinYang algorithm [6], which was recognized for its effectiveness in OCR preprocessing during the DocEng'22 [20] and DocEng'23 binarization competitions, the ZigZag algorithm simplifies the design of its predecessor while retaining its successful strategies. ZigZag begins by accurately estimating the background, then isolates the foreground through background subtraction, normalizes the foreground, and applies a threshold to generate a clear binary image.

3.1 About Background Estimation

Background estimation methods vary significantly, each with its own advantages and challenges, as demonstrated in Fig. 2. For instance, the method used by Michalak & Okarma [21] involves downscaling and upscaling the image (e.g., $\times\frac{1}{32}$ and $\times 32$), which provides a quick but coarse estimation of the background, as depicted in Fig. 2b.

Our previous algorithm, YinYang, sought a more precise background estimation by analyzing the most frequent color within localized windows and speeding up this process through the use of subsampling grids and interpolation. This approach, while more accurate, is not perfect since subsampling is a compromise that trades quality for efficiency. Consequently, it still faces challenges under conditions of strong uneven lighting, leading to potential inaccuracies as shown in Fig. 2c.

In contrast, ZigZag is designed to optimize efficiency while achieving the most accurate background estimation possible. Unlike many adaptive algorithms that directly use local statistical measures, such as mean or standard deviation, to compute a local threshold, ZigZag first estimates the background to ensure thorough foreground detection. This dedicated focus on accurate background estimation is crucial for handling uneven lighting conditions, significantly enhancing the algorithm's ability to recognize text effectively in real-world settings.

3.2 ZigZag Algorithm

ZigZag leverages integral images, also known as summed area tables, a cornerstone in computer vision and image processing. Integral images streamline the calculation of pixel value sums within any rectangular area, enabling rapid and efficient computations crucial for real-time image processing.

An integral image, denoted by $ii(x, y)$, accumulates the pixel values of the original image $i(x, y)$. Each pixel in the integral image represents the cumulative sum of all pixel values to its left and above, including itself. This accumulation is mathematically represented by Equation 1. The calculation leverages a recursive approach as shown in Equation 2, allowing the entire image to be processed in linear time. This formulation ensures that even large images can be handled swiftly.

With the integral image computed, extracting the sum of pixel values within any specified rectangular region becomes a matter of simple arithmetic operations. By subtracting the values at the appropriate corners of the rectangle, the sum of the interior pixels is quickly obtained, as demonstrated by Equation 3. This ability to perform rapid calculations regardless of the rectangle's size is what makes integral images particularly valuable for applications that require frequent and dynamic area-based operations.

$$ii(x, y) = \sum_{x'=0}^{x' \le x} \sum_{y'=0}^{y' \le y} i(x', y') \tag{1}$$

$$\begin{aligned} ii(x, y) = i(x, y) + ii(x - 1, y) \\ + ii(x, y - 1) - ii(x - 1, y - 1) \end{aligned} \tag{2}$$

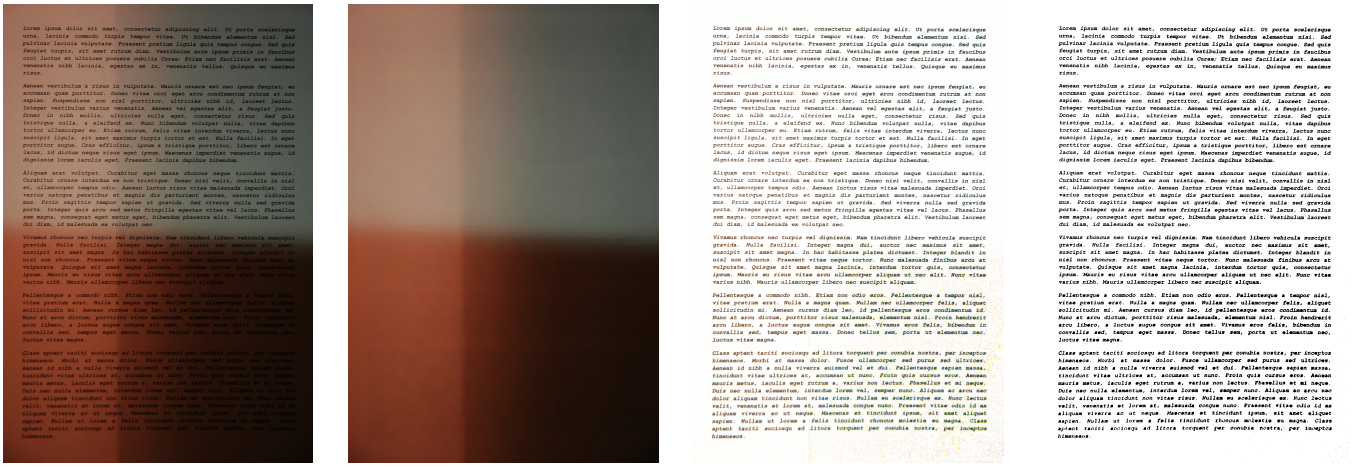
$$\begin{aligned} \sum_{x'=x_1}^{x_2} \sum_{y'=y_1}^{y_2} i(x', y') = ii(x_2, y_2) - ii(x_2, y_1 - 1) \\ - ii(x_1 - 1, y_2) + ii(x_1 - 1, y_1 - 1) \end{aligned} \tag{3}$$

Building on the use of integral images, the ZigZag algorithm is implemented through three main steps, as illustrated in Fig. 1: background estimation (Fig. 1b), foreground extraction (Fig. 1c), and thresholding (Fig. 1d). For a more in-depth understanding, detailed pseudo-code for the entire algorithm is provided in Algorithm 1.

The background estimation in ZigZag proceeds in two passes, utilizing mean filtering. In the first pass, the algorithm calculates local mean intensities for each pixel. Pixels that are darker than their respective local mean are likely associated with text and are thus masked out. This initial masking step reduces the influence of text pixels on the background estimation, allowing the algorithm to focus on the brighter, presumably background, areas. In the second pass, the algorithm recalculates the local mean intensities, this time excluding the previously masked text pixels. This two-pass mean filtering approach results in a cleaner and more precise background estimation compared to single-pass methods, effectively minimizing text interference and improving the overall quality of binarization.

A parameter, $weight$, can be adjusted to lower the local mean intensity values during the first pass, making the process more suitable for complex or irregular backgrounds. This weighted mean, referred to as $mean_w$ in the pseudo-code (line 19), defaults to 1.0 (100%). The $weight$ parameter can be set within a practical range of 0.5 to 1.0, for instance, to enhance background estimation in images with textured surfaces or with moderate back-to-front interference, such as those found in historical documents. The impact of varying $weight$ on the background estimation is visually demonstrated in Figs. 2d through 2h.

In practice, the algorithm computes three integral images as outlined in lines 2 to 27 of the pseudo-code: ii for overall pixel intensities, ii_{bg} for background pixel intensities, and $area_{bg}$ for



(a) Original image (b) Background (c) Foreground (d) Binarization

Figure 1: The three steps of the ZigZag image binarization process – (a) original image, (b) estimated background, (c) extracted foreground, and (d) black and white binarized image.

counting background pixels. Lines 2 through 8 detail the computation of ii, while lines 9 to 27 explain how this data is used to generate ii_{bg}, which filters out foreground pixels using the weighted mean. Concurrently, $area_{bg}$ tracks the count of non-masked pixels, i.e., those likely representing the background.

Lines 28 to 38 use $area_{bg}$ and ii_{bg} to calculate the local mean background intensity, $mean_{bg}$, for each pixel, relying exclusively on non-masked pixels. The local mean intensity calculated for each pixel using $area_{bg}$ and ii_{bg} represents our estimated background for those pixels. Pixels exceeding the estimated background threshold, $mean_{bg}$, are classified as background and set to white, while the remaining pixels use $mean_{bg}$ to normalize the foreground, as described in line 36 and explained hereafter.

Foreground normalization dynamically adjusts each pixel value based on its neighborhood, i.e., the current local window. This step extends the intensity spectrum from the current background estimation pixel value up to white (255), interpolating values from $[0, mean_{bg}]$ to $[0, 255]$. This process boosts local contrast and normalizes pixel intensities, sharpening text and allowing a single threshold to be efficiently applied across the image.

Note that while the algorithm calculates each estimated background pixel, it does not retain a full image of it in memory. Instead, these calculated values are used directly for efficient foreground extraction. Consequently, the images shown in Figs. 1b, and 2d through 2h are generated solely for illustrative purposes.

3.3 ZigZag Output Modes

ZigZag provides three versatile output modes to accommodate diverse image processing requirements, enhancing its functionality across applications from digital archiving to OCR and computer vision.

Grayscale Mode: Grayscale mode extracts the foreground in grayscale, preserving variations in intensity that are essential for modern OCR and computer vision applications. Grayscale images are preferred in deep learning contexts because they efficiently convey texture and contrast information. Well-known systems such as Google Vision OCR [13] and ABBYY Finereader [1] utilize these grayscale inputs to optimize text and object recognition accuracy.

Color Mode: Color mode preserves full foreground color information, which is beneficial for applications like digital archiving and advanced image segmentation. Unlike grayscale mode, it processes the red, green, and blue channels independently, enhancing image fidelity while maintaining maximum information for subsequent processing. This flexibility significantly enhances the algorithm's adaptability across various imaging challenges.

Binarization Mode: To preserve details and contours in the binarized image, ZigZag first applies a two-fold upsampling to the extracted grayscale foreground, ensuring that finer details are retained in the output image due to the higher resolution. Subsequently, Otsu thresholding [26] is applied to convert the image into a high-contrast black and white format. This mode is essential for legacy OCR systems that rely on clear and detailed binary images for accurate text recognition.

Each mode enables ZigZag to meet distinct archival and processing needs, ensuring optimal data preservation, improved image quality, and reduced file size. This adaptability makes ZigZag a valuable tool for image processing professionals and researchers, offering versatile functionalities suited to various technological applications.

4 Experimental Validation

In this section, we compare ZigZag with established adaptive binarization methods, focusing specifically on non-machine learning approaches. Our evaluation includes multiple datasets and quality metrics. The algorithms assessed alongside ZigZag include Bernsen [5], Bradley [7], Michalak [21], Nick [16], Niblack [16], Sauvola [29], and YinYang [6], with Otsu's method [26] serving as our baseline for comparison.

(a) Original image (b) Michalak BGE (c) YinYang BGE (d) ZigZag BGE (weight=100%)

(e) ZigZag BGE, weight=75% (f) ZigZag BGE, weight=50% (g) ZigZag BGE, weight=25% (h) ZigZag BGE, weight=0%

Figure 2: Comparison of background estimation (BGE) algorithms: (a) original image, (b) Michalak, (c) YinYang, and (d) ZigZag, from least to most accurate. Images (e), (f), (g), and (h) show the effect of the mean weight parameter on ZigZag's BGE accuracy.

Each algorithm was executed with its default settings, as specified in the original research papers. All implementations were conducted in Java, and experiments were performed on a system equipped with an Intel(R) Core(TM) i7-10875H CPU at 2.30 GHz.

4.1 Evaluation on Photographed Documents

We conducted evaluations using two distinct datasets to assess the performance of the proposed binarization algorithms. The first dataset, known as the WEZUT OCR Dataset [22], consists of 176 non-uniformly illuminated document images, each featuring the commonly used placeholder text "Lorem ipsum…", captured with a Nikon N70 Digital Single Lens Reflex (DSLR) camera. The images represent the photos of the documents printed using 5 different popular font shapes (Arial, Times New Roman, Calibri, Courier and Verdana) with some typical modifications of attributes (normal, bold and italic versions of all fonts as well as bold italics). This dataset is intended mainly for the evaluation of image binarization algorithms, developed for the pre-processing of non-uniformly illuminated document images subjected to further text recognition using various OCR engines.

The second dataset, termed "mobile-dataset-4", was introduced at the 2022 ACM Symposium on Document Engineering as part of the "Quality, Space, and Time Competition on Binarizing Photographed Documents" [20]. It comprises 48 images of printed scientific articles captured under uneven lighting conditions with various popular smartphone models, utilizing both on and off camera flash settings. We excluded an additional 32 images of hardcover book pages from the dataset due to their curved nature, which made it impractical to establish accurate ground truth because of non-linear geometric transformations.

For a comprehensive and thorough evaluation, we created an OCR ground-truth by mapping text characters in the photographed

Algorithm 1 ZigZag foreground extraction algorithm. The conditional operator $(a\ ?\ b\ :\ c)$ is used to represent inline-if statements, where b is returned if a is true, and c otherwise. Boundary checks are omitted for clarity. In the comments, bg pixels stands for background pixels.

procedure ExtractForeground($input, output, width, height, size, weight$)
  for $x \leftarrow 1$ to $width$ **do**
    $sum \leftarrow 0$ ▷ Initialize sum for integral image
    for $y \leftarrow 1$ to $height$ **do**
      $sum \leftarrow sum + input[x, y]$ ▷ Cumulative sum
      $ii[x, y] \leftarrow (x == 1\ ?\ sum\ :\ ii[x-1, y] + sum)$ ▷ Integral image
    end for
  end for
  $r \leftarrow \lfloor size/2 \rfloor$ ▷ Local window half-size
  $np \leftarrow (2 \times r + 1)^2$ ▷ Number of pixels in the local window
  for $x \leftarrow 1$ to $width$ **do**
    $count \leftarrow 0$ ▷ Initialize count of bg pixels
    $sum \leftarrow 0$ ▷ Initialize sum of intensities for bg pixels
    for $y \leftarrow 1$ to $height$ **do**
      $x_1 \leftarrow x - r - 1$ ▷ Local window boundaries
      $x_2 \leftarrow x + r$
      $y_1 \leftarrow y - r - 1$
      $y_2 \leftarrow y + r$
      $mean_w \leftarrow weight \times (ii[x_2, y_2] - ii[x_2, y_1] - ii[x_1, y_2] + ii[x_1, y_1])/np$ ▷ Weighted mean intensity
      if $(input[x, y] \geq mean_w)$ **then** ▷ Check if pixel is likely background
        $count_{bg} \leftarrow count_{bg} + 1$ ▷ Increment count of bg pixels
        $sum_{bg} \leftarrow sum_{bg} + input[x, y]$ ▷ Accumulate intensity of bg pixels
      end if
      $area_{bg}[x, y] \leftarrow (x = 1\ ?\ count_{bg}\ :\ area_{bg}[x-1, y] + count_{bg})$ ▷ Cumulative count of bg pixels
      $ii_{bg}[x, y] \leftarrow (x = 1\ ?\ sum_{bg}\ :\ ii_{bg}[x-1, y] + sum_{bg})$ ▷ Cumulative sum of intensities for bg pixels
    end for
  end for
  for $x \leftarrow 1$ to $width$ **do**
    for $y \leftarrow 1$ to $height$ **do**
      $x_1 \leftarrow x - r - 1$ ▷ Local window boundaries
      $x_2 \leftarrow x + r$
      $y_1 \leftarrow y - r - 1$
      $y_2 \leftarrow y + r$
      $np_{bg} \leftarrow area_{bg}[x_2, y_2] - area_{bg}[x_2, y_1] - area_{bg}[x_1, y_2] + area_{bg}[x_1, y_1]$ ▷ Number of bg pixels in local window
      $mean_{bg} \leftarrow (ii_{bg}[x_2, y_2] - ii_{bg}[x_2, y_1] - ii_{bg}[x_1, y_2] + ii_{bg}[x_1 - 1, y_1])/np$ ▷ Mean intensity of bg pixels (i.e., BGE)
      $output[x, y] \leftarrow (input[x, y] \geq mean_{bg}\ ?\ 255\ :\ input[x, y] \times 256/mean_{bg})$ ▷ Background removal
    end for
  end for
end procedure

datasets to their respective bounding boxes using Google Vision OCR, followed by painstaking manual corrections to ensure accuracy and reliability.

Performance was measured using six metrics: accuracy, precision, recall, F-score, normalized Levenshtein distance [18], and mean processing time. The Levenshtein distance was normalized by the total number of characters:

$$Levenshtein' = \frac{nbChars - Levenshtein}{nbChars}$$

The F-score, which is the harmonic mean of precision and recall and is calculated as follows, serves as the primary measure of quality:

$$F\text{-}score = 2 \times \frac{precision \times recall}{precision + recall}$$

Precision is the ratio of true positives to the sum of all positives, and recall is the ratio of true positives to the sum of true positives and false negatives. Originally used to assess binarization quality by comparing pixel-level accuracy on binary images, the F-score also serves as an effective general classification metric, particularly valuable for character-level text recognition evaluations.

Table 1 displays the results obtained from the WEZUT OCR and DocEng'22 Smartphone datasets, showcasing a range of adaptive non-machine learning algorithms ranked by their F-score (computed at the character level). Fig. 4 illustrates the impact of window size on F-score performance using the WEZUT OCR dataset. Typically, optimal window sizes range between 20 and 40 pixels for all adaptive algorithms. Remarkably, ZigZag demonstrates

Table 1: Google Vision OCR evaluation on WEZUT OCR and DocEng'22 Smartphone datasets using various adaptive binarization algorithms. Performance is ranked by F-scores, with ties broken by processing time. Quality values are percentages with standard deviations in parentheses, and mean processing times are in milliseconds. Additional rows show the impact of mean weight on F-score. The grayscale mode (GL) is not included in the ranking.

WEZUT OCR Dataset (176 images)

#	Algorithm	Acc.	Prec.	Recall	F-score	Levenshtein'	Time
1	**ZigZag**	99.85	99.94	99.91	99.93 (0.12)	99.74 (1.20)	194
2	YinYang	99.76	99.90	99.85	99.88 (0.22)	99.73 (0.73)	1044
3	Sauvola	99.71	99.86	99.85	99.86 (0.24)	99.62 (1.65)	3473
4	Nick	99.71	99.87	99.84	99.85 (0.25)	99.60 (1.38)	3012
5	Bradley	99.65	99.81	99.84	99.83 (0.31)	99.55 (2.23)	112
6	Michalak	99.43	99.78	99.64	99.71 (0.79)	99.47 (1.50)	73
7	Bernsen	95.56	95.90	99.63	97.67 (2.63)	94.63 (6.88)	1278
8	Niblack	87.70	88.27	99.31	92.91 (7.82)	82.57 (20.55)	3472
9	Otsu	60.59	99.01	60.84	73.48 (16.10)	59.74 (20.38)	24
	ZigZag GL	99.96	99.98	99.98	99.98 (0.06)		
	ZigZag 100%	99.86	99.94	99.91	99.93 (0.12)		
	ZigZag 90%	99.85	99.94	99.91	99.93 (0.12)		
	ZigZag 80%	99.82	99.93	99.89	99.91 (0.17)		
	ZigZag 70%	99.77	99.91	99.87	99.89 (0.24)		
	ZigZag 60%	99.74	99.89	99.84	99.87 (0.28)		
	ZigZag 50%	99.73	99.89	99.83	99.86 (0.28)		

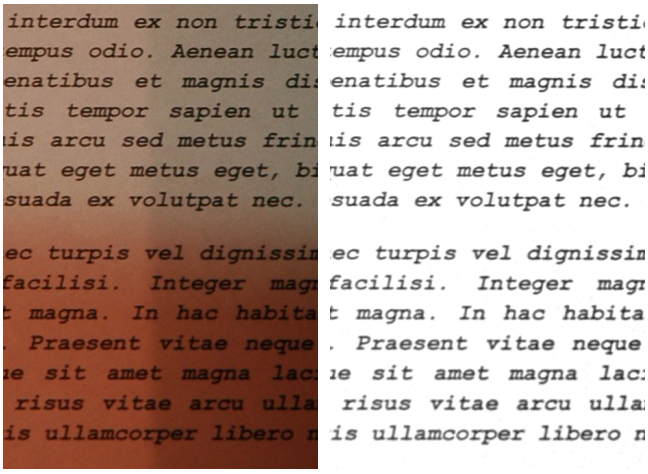
DocEng'22 Smartphone Dataset (48 images)

#	Algorithm	Acc.	Prec.	Recall	F-score	Levenshtein'	Time
1	**ZigZag**	99.69	99.85	99.85	99.85 (0.06)	99.82 (0.17)	347
2	YinYang	99.70	99.85	99.85	99.85 (0.05)	99.82 (0.18)	1477
3	Nick	99.62	99.78	99.84	99.81 (0.20)	99.74 (0.54)	4564
4	Sauvola	99.61	99.77	99.84	99.80 (0.21)	99.74 (0.49)	5729
5	Bradley	99.59	99.74	99.85	99.79 (0.16)	99.57 (1.91)	198
6	Michalak	99.44	99.61	99.83	99.72 (0.15)	99.57 (0.33)	123
7	Niblack	95.92	96.15	99.75	97.86 (2.56)	94.54 (7.31)	5883
8	Bernsen	95.89	96.11	99.76	97.81 (3.15)	94.14 (8.70)	2192
9	Otsu	71.68	98.05	72.29	81.88 (13.93)	71.71 (20.32)	40
	ZigZag GL	99.81	99.90	99.91	99.90 (0.07)		
	ZigZag 100%	99.70	99.85	99.85	99.85 (0.06)		
	ZigZag 90%	99.70	99.85	99.85	99.85 (0.06)		
	ZigZag 80%	99.68	99.84	99.84	99.84 (0.07)		
	ZigZag 70%	99.67	99.84	99.83	99.83 (0.10)		
	ZigZag 60%	99.63	99.82	99.81	99.81 (0.12)		
	ZigZag 50%	99.63	99.82	99.81	99.81 (0.09)		

less sensitivity to window size variations compared to other algorithms, allowing for favorable results even with suboptimal window sizes. The evaluation results in Table 1 show the average outcomes across three window sizes: 20, 30, and 40 pixels. Exceptions include YinYang, which uses a default window size of 64 pixels for optimization, Michalak with a default kernel size of 32, and Otsu, a non-adaptive algorithm unaffected by window size altogether.

ZigZag is expected to perform optimally with a mean weight close to 100% for OCR preprocessing of photographs captured under uneven lighting conditions. This is based on the assumption that despite significant luminosity variations in the background, these images typically lack texture, resulting in a close alignment of the local mean with the local background. Consequently, the estimated background should closely approximate the ideal state, requiring no weighting adjustments. This hypothesis was confirmed by computing the F-score on the WEZUT OCR and DocEng'22 Smartphone datasets using five incremental mean weights ranging from 50% to 100%. For the sake of computation time, additional ZigZag F-scores were calculated using a representative window size of 30 pixels.

In our evaluations, ZigZag consistently outperformed all other adaptive algorithms, showcasing remarkable performance even compared to its predecessor. YinYang itself had previously been recognized in the DocEng'2022 and DocEng'2023 binarization competitions, competing against a broad range of classical and modern



(a) Original image (b) Grayscale mode

Figure 3: ZigZag output mode showing (a) the original image and (b) the grayscale foreground extraction.

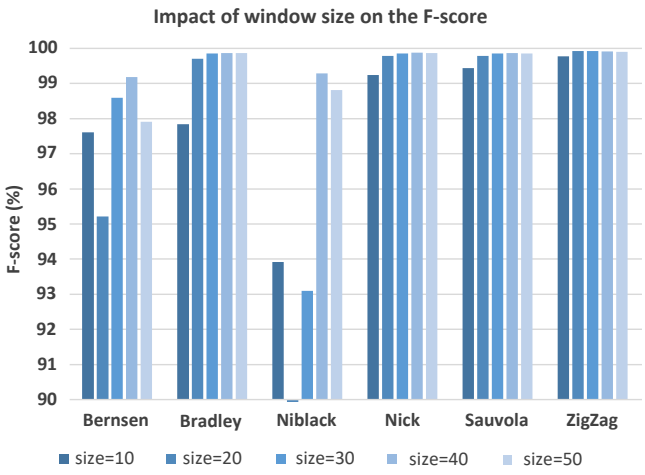


Figure 4: Impact of window size on the F-score performance for optical character recognition using the WEZUT Dataset.

algorithms. A detailed evaluation and comparison of YinYang is available in "A Quality, Size, and Time Assessment of the Binarization of Documents Photographed by Smartphones" by Bernardino et al. (2023) [4].

Similar to its predecessor, ZigZag employs two-fold upsampling for image binarization in OCR applications to minimize information loss when converting grayscale images to black and white. This approach is especially beneficial for images with low resolution or small font sizes. Table 1 also includes additional rows for ZigZag GL, representing the grayscale extracted image foreground just before the Otsu thresholding process (see Fig. 3b). Notably, ZigZag GL achieves the highest F-score results, consistent with Google Vision OCR's preference for grayscale inputs.

Google Vision OCR demonstrated impressive performance, even with barely discernible characters. It operates in three stages: analyzing the image layout to locate text, performing text recognition, and correcting errors during post-processing with a language model. This robust post-processing significantly boosts its results, which is particularly noteworthy given the use of Lorem Ipsum as dummy text in the WEZUT dataset. Without this post-processing stage, ZigZag would likely showcase a more pronounced performance advantage over its competitors.

4.2 Evaluation on Historical Documents

ZigZag is primarily designed for processing non-uniformly illuminated document images, such as those captured with mobile devices. However, the algorithm also effectively removes backgrounds from historical documents, which are often characterized by textured surfaces, signs of degradation, and back-to-front interference.

Historical document images typically originate from controlled environments without the complications of non-uniform lighting but present their own unique challenges, such as aged and textured paper. ZigZag includes an adjustable weight parameter specifically designed for these historical documents. By fine-tuning this parameter, ZigZag can adapt more efficiently to background irregularities, which is essential for accurately extracting the foreground, thereby maintaining the integrity and readability of historical texts.

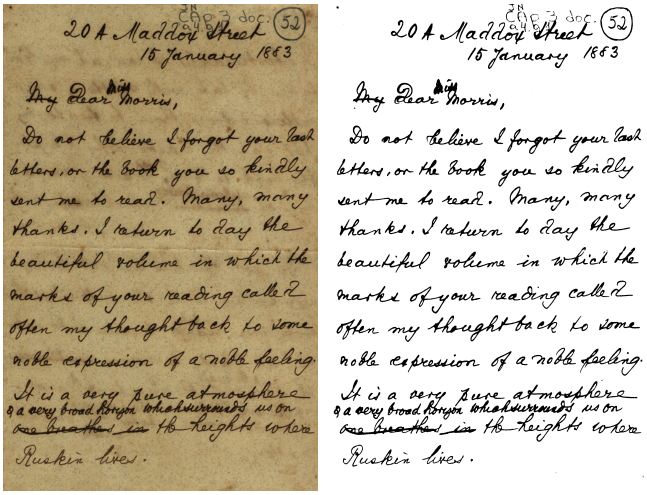
We evaluated ZigZag's performance on the Nabuco dataset [19] using the 35 images that had ground-truth annotations out of a total of 1067 images. The results shown in Table 2 demonstrate ZigZag's effectiveness in processing historical documents, underscoring its adaptability to various document types. The algorithm achieved a Mean Squared Error (MSE) of 1.29 and a Peak Signal-to-Noise Ratio (PSNR) of 20.34, reflecting its accuracy in background estimation and its ability to minimize noise. These metrics confirm that ZigZag effectively reduces artifacts during binarization, preserving critical document details, which is crucial for the legibility and archival quality of historical texts.

We determined the optimal mean weight for the Nabuco dataset to be 60% after evaluating values from 0% to 100% in 10% increments. Even with suboptimal weight settings (as shown in the additional rows of Table 2), ZigZag consistently delivered robust performance, demonstrated by its low MSE and high PSNR values. This evaluation underscores ZigZag's capability to effectively tackle the challenges posed by historical document imagery, ensuring that the binarized output retains both clarity and fidelity to the original content.

Fig. 5 shows the ZigZag algorithm processing a historical document image from the Nabuco dataset, producing a reliable binarized output. These results are notable not only for their quality but also for their potential to lay the groundwork for machine learning approaches. By generating a large volume of good-quality initial binarized samples, ZigZag could significantly accelerate machine learning workflows. Machine learning models trained on these samples can then be further refined using smaller, highly precise ground-truth datasets, which, while offering superior accuracy, are much more time-consuming to produce. This approach could improve overall recognition accuracy and enhance model generalization.

Table 2: Image binarization results on the Nabuco historical dataset. Quality values are presented as percentages, except MSE and PSNR. Additional rows show the impact of mean weight on F-score and other metrics.

Nabuco Dataset (35 images)							
#	**Algorithm**	**Acc.**	**Prec.**	**Recall**	**F-score**	**MSE**	**PSNR**
1	**ZigZag 60%**	98.71	89.63	94.03	91.01	1.29	20.34
2	Nick	98.30	86.23	93.92	88.86	1.70	19.24
3	YinYang	97.99	80.98	97.13	86.96	2.01	18.78
4	Sauvola	97.69	79.18	97.75	86.34	2.31	18.34
5	Otsu	97.64	79.17	97.76	85.99	2.36	18.70
6	Bradley	97.60	78.44	97.48	85.80	2.40	18.05
7	Michalak	91.77	46.26	99.78	61.85	8.23	11.40
8	Bernsen	84.70	32.15	91.88	45.41	15.30	8.52
9	Niblack	76.05	22.75	98.75	35.72	23.95	6.33
	ZigZag 50%	98.73	90.52	93.02	90.98	1.27	20.33
	ZigZag 70%	98.66	88.58	94.87	90.79	1.34	20.28
	ZigZag 80%	98.58	87.40	95.51	90.36	1.42	20.13
	ZigZag 90%	98.48	86.27	96.01	89.85	1.52	19.94
	ZigZag 100%	98.40	85.50	96.29	89.48	1.60	19.82



(a) Original image **(b) Binarized image**

Figure 5: ZigZag binarization with a 30-pixel window size and 60% mean weight: (a) original, and (b) binarized images.

4.3 Limitations and Future Work

While the evaluations presented demonstrate the effectiveness of ZigZag across various datasets, they primarily rely on OCR-based metrics derived from Google Vision OCR. Given the advanced nature of this OCR system, subtle differences between binarization algorithms may be obscured, as evidenced by the consistently high F-scores (i.e., in the 99.x% range) achieved by the leading algorithms. This suggests that the forgiving nature of advanced OCR engines might not fully capture the nuanced performance variations among different binarization methods.

To address this, future studies could benefit from using less sophisticated OCR systems, which may reveal more pronounced differences between algorithms, resulting in lower and more representative F-scores. Additionally, the creation of a dataset with pixel-level ground-truth annotations for smartphone-captured document images would allow for a more direct and robust evaluation of binarization performance, offering deeper insights into the strengths and weaknesses of each method.

One notable limitation of ZigZag is its handling of very large text elements. When processing such large, thick text, the use of a local window for background estimation can sometimes lead to inaccuracies. Specifically, the algorithm may only extract the contours of the text as the foreground (black), while misclassifying the interior as the background (white), effectively erasing crucial parts of the text. This issue arises when the local window fits entirely within a text area, causing the local mean intensity to align with the text intensity, which the algorithm then mistakenly classifies as the background, leading to inaccurate binarization.

To overcome this limitation, future research could explore adaptive window sizing strategies or alternative approaches that more effectively handle large text elements. These improvements would enhance ZigZag's effectiveness across a broader spectrum of document types. Coupling these refinements with a more comprehensive evaluation of binarization performance could further establish ZigZag as a reliable and versatile tool in document image processing.

5 Conclusion

ZigZag represents a significant advancement in the field of adaptive, non-machine learning image binarization, particularly for handling non-uniformly illuminated document images. Our empirical evaluations on the WEZUT OCR and DocEng'22 Smartphone datasets, comprising photographed documents under challenging lighting conditions, demonstrate ZigZag's superior performance.

Furthermore, its effectiveness in processing historical documents, as evidenced by the Nabuco dataset, underscores its versatility in managing aged and textured backgrounds.

The success of ZigZag is primarily due to its two-pass background estimation process, which enhances the precision of foreground extraction. By initially masking potential text pixels, the algorithm achieves a more accurate background estimation, leading to cleaner and more reliable outputs. The versatility of ZigZag is further demonstrated through its multiple output modes, including color and grayscale extraction, making it adaptable to diverse user needs.

In addition to its technical strengths, ZigZag offers notable advantages over machine-learning-based methods, such as ease of implementation, reduced computational complexity, lower power consumption, and consistent, predictable outcomes. These benefits make ZigZag a reliable choice for deployment in various real-world applications, especially in resource-constrained environments where efficiency and reliability are paramount. Its efficient processing and robustness make it particularly well-suited for mobile document imaging, significantly enhancing real-time document management systems.

In conclusion, ZigZag makes a valuable contribution to adaptive image binarization, providing a practical and effective solution to the challenges posed by uneven lighting in document images. Its combination of simplicity, efficiency, and versatility makes it a reliable tool for both researchers and industry professionals. An open-source implementation is available on GitHub at https://github.com/Bloechle/ZigZag to support ongoing studies and real-world applications.

References

[1] ABBYY. 2023. *ABBYY FineReader 2023.* https://www.abbyy.com/en-us/finereader/ Accessed on September 8, 2023.

[2] Euthimios Badekas and Nikos Papamarkos. 2007. Optimal combination of document binarization techniques using a self-organizing map neural network. *Engineering Applications of Artificial Intelligence* 20, 1 (2007), 11–24.

[3] Francesco Bardozzo, Borja De La Osa, Lubomira Horanska, Javier Fumanal-Idocin, Luigi Troiano, Roberto Tagliaferri, Javier Fernandez, Humberto Bustince, et al. 2020. Adaptive binarization based on fuzzy integrals. *arXiv preprint arXiv:2003.08755* (2020).

[4] Rodrigo Bernardino, Rafael Dueire Lins, and Ricardo da Silva Barboza. 2023. A Quality, Size and Time Assessment of the Binarization of Documents Photographed by Smartphones. *Journal of Imaging* 9, 2 (2023), 41.

[5] John Bernsen. 1986. Dynamic thresholding of gray-level images. In *Proc. Eighth Int'l conf. Pattern Recognition, Paris, 1986.*

[6] Jean-Luc Bloechle, Jean Hennebert, and Christophe Gisler. 2023. YinYang, a Fast and Robust Adaptive Document Image Binarization for Optical Character Recognition. In *Proceedings of the ACM Symposium on Document Engineering 2023* (Limerick, Ireland) *(DocEng '23)*. Association for Computing Machinery, New York, NY, USA, Article 19, 4 pages. https://doi.org/10.1145/3573128.3609354

[7] Derek Bradley and Gerhard Roth. 2007. Adaptive thresholding using the integral image. *Journal of graphics tools* 12, 2 (2007), 13–21.

[8] Jorge Calvo-Zaragoza and Antonio-Javier Gallego. 2019. A selectional auto-encoder approach for document image binarization. *Pattern Recognition* 86 (2019), 37–47.

[9] Chien-Hsing Chou, Wen-Hsiung Lin, and Fu Chang. 2010. A binarization method with learning-built rules for document images produced by cameras. *Pattern Recognition* 43, 4 (2010), 1518–1530.

[10] Franklin C. Crow. 1984. Summed-Area Tables for Texture Mapping. *SIGGRAPH Comput. Graph.* 18, 3 (jan 1984), 207–212. https://doi.org/10.1145/964965.808600

[11] Meng-Ling Feng and Yap-Peng Tan. 2004. Contrast adaptive binarization of low quality document images. *IEICE Electronics Express* 1, 16 (2004), 501–506.

[12] Basilios Gatos, Ioannis Pratikakis, and Stavros J Perantonis. 2006. Adaptive degraded document image binarization. *Pattern recognition* 39, 3 (2006), 317–327.

[13] Google. 2023. *Google Cloud Vision OCR.* https://cloud.google.com/vision Accessed on September 8, 2023.

[14] Sheng He and Lambert Schomaker. 2019. DeepOtsu: Document enhancement and binarization using iterative deep learning. *Pattern recognition* 91 (2019), 379–390.

[15] David Hebert, Stephane Nicolas, and Thierry Paquet. 2013. Discrete CRF based combination framework for document image binarization. In *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 1165–1169.

[16] Khurram Khurshid, Imran Siddiqi, Claudie Faure, and Nicole Vincent. 2009. Comparison of Niblack inspired binarization methods for ancient documents. In *Document Recognition and Retrieval XVI*, Vol. 7247. SPIE, 267–275.

[17] Ján Koloda and Jue Wang. 2023. Context Aware Document Binarization and Its Application to Information Extraction from Structured Documents. In *International Conference on Document Analysis and Recognition*. Springer, 63–78.

[18] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.

[19] Rafael Dueire Lins. 2011. Two Decades of Document Processing in Latin America. *J. Univers. Comput. Sci* 17, 1 (2011), 151–161.

[20] Rafael Dueire Lins, Rodrigo Barros Bernardino, Ricardo Barboza, and Raimundo Oliveira. 2022. The Winner Takes It All: Choosing the "best" Binarization Algorithm for Photographed Documents. In *International Workshop on Document Analysis Systems*. Springer, 48–64.

[21] Hubert Michalak and Krzysztof Okarma. 2019. Fast Binarization of Unevenly Illuminated Document Images Based on Background Estimation for Optical Character Recognition Purposes. *J. Univers. Comput. Sci.* 25, 6 (2019), 627–646.

[22] Hubert Michalak and Krzysztof Okarma. 2020. Robust combined binarization method of non-uniformly illuminated document images for alphanumerical character recognition. *Sensors* 20, 10 (2020), 2914.

[23] Reza Farrahi Moghaddam and Mohamed Cheriet. 2012. AdOtsu: An adaptive and parameterless generalization of Otsu's method for document image binarization. *Pattern Recognition* 45, 6 (2012), 2419–2431.

[24] Ajoy Mondal, Chetan Reddy, and CV Jawahar. 2023. Deep semantic binarization for document images. *Multimedia Tools and Applications* 82, 5 (2023), 6531–6555.

[25] Sheshera Mysore, Manish Kumar Gupta, and Swapnil Belhe. 2016. Complex and degraded color document image binarization. In *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 157–162.

[26] Nobuyuki Otsu. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9, 1 (1979), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

[27] Xujun Peng, Huaigu Cao, and Prem Natarajan. 2017. Using convolutional encoder-decoder for document image binarization. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, Vol. 1. IEEE, 708–713.

[28] Ioannis Pratikakis, Konstantinos Zagoris, George Barlas, and Basilis Gatos. 2017. ICDAR2017 competition on document image binarization (DIBCO 2017). In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1. IEEE, 1395–1403.

[29] Jaakko Sauvola and Matti Pietikäinen. 2000. Adaptive document image binarization. *Pattern recognition* 33, 2 (2000), 225–236.

[30] Faisal Shafait, Daniel Keysers, and Thomas M Breuel. 2008. Efficient implementation of local adaptive thresholding techniques using integral images. In *Document recognition and retrieval XV*, Vol. 6815. SPIE, 317–322.

[31] Bolan Su, Shijian Lu, and Chew Lim Tan. 2011. Combination of document image binarization techniques. In *2011 International Conference on Document Analysis and Recognition*. IEEE, 22–26.

[32] Salvatore Tabbone and Laurent Wendling. 2004. Binarization of color images from an adaptation of possibilistic c-means algorithm. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 1. IEEE, 704–707.

[33] Chris Tensmeyer and Tony Martinez. 2017. Document image binarization with fully convolutional neural networks. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, Vol. 1. IEEE, 99–104.

[34] Chun-Ming Tsai and Hsi-Jian Lee. 2002. Binarization of color document images via luminance and saturation color features. *IEEE Transactions on Image Processing* 11, 4 (2002), 434–451.

[35] Quang Nhat Vo, Soo Hyung Kim, Hyung Jeong Yang, and Gueesang Lee. 2018. Binarization of degraded document images based on hierarchical deep supervised network. *Pattern Recognition* 74 (2018), 568–586.

[36] Christian Wolf and J-M Jolion. 2004. Extraction and recognition of artificial text in multimedia documents. *Formal Pattern Analysis & Applications* 6, 4 (2004), 309–326.

[37] Wei Xiong, Xiuhong Jia, Dichun Yang, Meihui Ai, Lirong Li, and Song Wang. 2021. DP-LinkNet: A convolutional network for historical document image binarization. *KSII Transactions on Internet and Information Systems (TIIS)* 15, 5 (2021), 1778–1797.