SN Computer Science (2020) 1:173
https://doi.org/10.1007/s42979-020-00176-1

REVIEW ARTICLE

Historical Document Image Binarization: A Review

Chris Tensmeyer[1] · **Tony Martinez**[2]

Received: 24 October 2019 / Accepted: 20 April 2020 / Published online: 16 May 2020

Abstract
This review provides a comprehensive view of the field of historical document image binarization with a focus on the contributions made in the last decade. After the introduction of a standard benchmark dataset with the 2009 Document Image Binarization Contest, research in the field accelerated. Besides the standard methods for image thresholding, preprocessing, and post-processing, we review the literature on methods such as statistical models, pixel classification with learning algorithms, and parameter tuning. In addition to reviewing binarization algorithms, we discuss the available public datasets and evaluation metrics, including those that require pixel-level ground truth and those that do not. We conclude with recommendations for future work.

Keywords Binarization · Document analysis · Image processing · Historical documents

Introduction

Binarization is the process converting a multi-tone image into a bi-tonal image. In the case of document images, it is typical to map foreground text pixels to black and the rest of the image (background) to white. In many applications, binarization is a critical preprocessing step and helps facilitate other document processing tasks such as layout analysis and character recognition. In such pipelines, the quality of the binarization can greatly affect system performance, as errors made in the binarization step can propagate to downstream tasks. As a standalone application, binarization can serve as a noise removal process to increase document readability. The file size of binary images is often orders of magnitudes smaller than the original gray or color images, which makes them cheaper to store on disk. Additionally, with the rise of digital archives, file size can become a concern as large numbers of images are viewed over the Internet. If a person can still recognize the text in the binary images, then this compression can be obtained with virtually no loss in semantic image content.

In the last decade, a tremendous amount of progress has been made in the field of historical document binarization. In 2009, the first Document Image Binarization Contest (DIBCO) introduced the first dataset of real degraded images that have ground truth annotations at the pixel level [40]. This enabled a standardized evaluation procedure that allowed for direct comparison between algorithms. This spurred research in the field and the creation of more datasets in new application domains.

The purpose of this review is to highlight the recent advances of the last decade in all facets of historical document binarization. We discuss not only binarization methods, but also topics such as preprocessing, post-processing, algorithm efficiency, datasets, evaluation, and definitions of ground truth.

Challenges

Historical document images are, in general, more difficult to binarize than modern scanned documents. This is in part due to their degraded state and partly because many historical documents are digitized with cameras, which do not have controlled illumination conditions like scanners. Some camera produced images have uneven illumination due to bad lighting or because the page is not flat (e.g., curved edges near book bindings).

This article is part of the topical collection "Document Analysis and Recognition" guest edited by Michael Blumenstein, Seiichi Uchida and Cheng-Lin Liu.

✉ Chris Tensmeyer
tensmeye@adobe.com

[1] Adobe Research, San Jose, USA

[2] Computer Science Department, Brigham Young University, Provo, USA

(a) Large Stain (b) Creases (c) Complex Background

(d) Border Noise (e) Varying Font Size (f) Faint Text

(g) Multiple Text Colors (h) Stamps (i) Compression Artifacts

(j) Thin Strokes (k) Bleed Through Ink (l) Smeared Ink

Fig. 1 Challenging aspects of historical document binarization illustrated with DIBCO images

Figure 1 illustrates several degradation types that commonly occur in historical documents. Stains serve to decrease the contrast between foreground and background inside the stained region, and strong edges around stains can be difficult to distinguish from real foreground/background boundaries. Creases cause similar difficulties as stains. Complex backgrounds arising from paper textures or uneven illumination cause difficulties for statistical features because the statistics vary from one image region to another. Border noise was a major challenge in HDIBCO 2018 [130] because dark surroundings have similar intensity as foreground ink and book edges provide high frequency edges. Many local thresholding approaches use a fixed window size which is not optimal for handling various font sizes and stroke widths. Faint text is challenging due to the low contrast, and multiple text colors can lead to low intensity for some colors when converting to grayscale images. Stamps are similar to stains, but can be darker. Document images are high resolution, and image compression (e.g., JPEG)

is often used to meet limited storage requirements, but compression artifacts introduce high frequency content into background regions. Thin strokes (e.g., single pixel wide) can be discarded during image smoothing or noise removal processes. Bleed-through noise is perhaps the most challenging degradation because the noise is shaped like text, is superimposed over the foreground text, and may co-occur with faint non-bleed-through text. Finally, smeared ink can make it difficult to localize foreground/background boundaries.

Due to the large variety of historical documents, it is commonly accepted that no single algorithm can perform best on every image [76]. For example, in [59] it was noted that the state-of-the-art methods for paper documents do not produce good results for Palm Leaf Manuscripts (PLMs). Thus, many binarization methods are designed to address some particular degradation type or domain.

Previous Reviews

Previous reviews on historical document binarization have taken a much narrower scope than the current article. There are some short review articles that briefly describe about a dozen methods [12, 99, 100]. Other articles do not present themselves as surveys, but do describe and then empirically evaluate a large number of existing algorithms [57, 148, 158].

A seminal review by Sezgin and Sankur [150] was published in 2004 which described 40 algorithms (primarily global thresholding methods) and provided detailed performance analysis on synthetic data. However, the binarization field has progressed much in the last 15 years and a new review is needed to discuss recent techniques not based on thresholding. Similarly, [77] recently performed a time quality assessment of 30 pre-2010 algorithms and only considered algorithms that were fast (e.g., the popular Howe [50] method was excluded for speed). While recent algorithms, e.g., deep learning, may be slower, they also tend to be more accurate [129].

Ismail et al. [51] recently have reviewed 29 thresholding algorithms and organized them into a taxonomy based on what features are used to determine thresholds. While this review provides insight into thresholding methods, it does not explicitly address the many other types of methods and facets of the binarization field. In this review, we give a comprehensive view of the field including non-thresholding methods such as conditional random fields (CRF) and deep learning models. Additionally, we cover facets besides binarization methods such as algorithm efficiency, parameter tuning, datasets, evaluation, construction of binarization Ground Truth (GT), and alternatives to GT evaluation.

Paper Organization

In contrast to prior reviews that provide individual paper summaries, this paper is organized by topic. This choice is motivated by the modular nature of many binarization pipelines that combine many operations to achieve top performance (e.g., [111]). An individual summarization of each method would be insufficient to disentangle which specific parameters and operations (e.g., local threshold computation, noise removal, edge detection) are performance crucial and which may be interchanged without degrading binarization quality. Therefore, within each topic, we compare and contrast analogous operations and their relative merits. Fortunately, the community has recognized the value of comparing individual components and we draw on the analysis in the literature. We hope to highlight the contributions of individual operations that in isolation do not yield state-of-the-art performance, but may do so when incorporated into a full pipeline.

The remainder of the paper is organized in the following sections: "Preprocessing" section covers common preprocessing techniques for noise removal, background estimation, and grayscale conversion. State-of-the-art binarization methods are discussed in "Binarization Methods" section. Post-processing techniques, including morphology and how to combine multiple binary outputs, are found in "Post-processing" section. Efforts for efficient binarization methods are discussed in "Binarization Efficiency" section, and datasets with ground truth (GT) are presented in "Datasets" section. In "Evaluation with Pixel Ground Truth" section, we review standard pixel GT-based evaluation, including GT construction and common metrics. Concerns about pixel GT and alternative evaluation methods are discussed in "Evaluation Without Pixel Ground Truth" section. We offer concluding remarks in "Discussion and Conclusion" section.

Preprocessing

The primary reason binarization is difficult is the presence of noise, degradations, and low contrast between foreground and background. Preprocessing techniques do not directly produce a binary image, but are designed to deal with these challenges in order to make subsequent binarization easier. Though there are many preprocessing techniques, they can be roughly categorized into noise removal, background estimation, and color to grayscale conversion.

Noise Removal

The goal of noise removal is to smooth background and foreground textures, while preserving the contrast between the two regions. This makes subsequent binarization easier as local image patches are more homogeneous.

Wiener filtering [172] is among the most common noise removal techniques used in binarization [39, 62, 87, 89, 147]. For known additive Gaussian noise, it produces an optimally filtered image according to the mean squared error (MSE) criteria. Gatos et al. [39] implemented Wiener filtering in the spatial domain as

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

where $\nu^2(i,j)$ is a local estimate of the variance of additive Gaussian noise. In [39], $\nu^2(i,j) = \sum_{i'} \sum_{j'} \sigma(i',j')$, though as noted in [156], fixing ν^2 as the median $\sigma^2(i,j)$ computed over the whole image leads to better binarization performance.

Simple alternatives to Wiener filtering include low-pass Gaussian filtering [29, 115] and bilateral filtering [4, 108]. Roe and Mello [140] proposed an interesting variation on a bilateral filter, where each pixel is set to the median of its neighbors that have sufficient color similarity. Other low-pass filtering is done by truncating high frequency information through image transforms such as the contourlet transform [177] and a Gabor wavelet decomposition [102].

Total variation (TV) regularization [75, 156] finds a smooth image that resembles the original image. It is formulated as

$$I_f = \underset{I'}{\operatorname{argmin}} \sum_i \sum_j (I'(i,j) - I(i,j))^2 + \beta \sum_i \sum_j \left(|I'(i,j) - I'(i-1,j)| + |I'(i,j) - I'(i,j-1)| \right) \quad (2)$$

where β controls the strength of the edge L_1 magnitude penalty. Setting β too high, however, will cause undesirable smoothing of the edges that separate foreground and background. Other regularization terms, such as gradient L_2 and L_0 [52], can be used, but can be more difficult to optimize.

Non-local means [26, 75, 156] smooths an image by performing a weighted average over neighborhoods, where the averaging weights are determined by patch similarity.

$$I_f(i,j) = \sum_{i'j' \in N(i,j)} w(i,j,i',j') I(i',j') \quad (3)$$

where $N(i, j)$ returns the (perhaps global) neighborhood of (i, j) and $w(i,j,i'j')$ is a weight based on the similarity between the image patches centered at (i,j) and (i',j'). The weights for any given pixel must sum to 1, i.e., $\forall i,j$ $\sum_{i'j'} w(i,j,i',j') = 1$. It is known that averaging many noisy corruptions of an image (e.g., consecutive video frames) with zero-centered i.d.d. noise converge to the uncorrupted image. Non-local means attempts to average out the noise over similar local patches rather than over several images.

Kligler et al. [65] proposed a unique noise removal technique where the intensity image is converted to a point cloud $\{(x, y, I(x, y)\}$ that is embedded on a sphere. The transformed image values are based on the visibility of each point on the sphere from the center of the sphere. This preprocessing combined with Howe's method [50] achieved state-of-the-art performance on DIBCO data.

Background Estimation

Binarization of images with non-uniform background is often addressed by first estimating a grayscale background image and then using that estimate to preprocess the original image. One common way to compensate for the background is simple subtraction [24, 39, 70, 85, 108, 156, 168].

$$I'(i,j) = I(i,j) - B(i,j) \quad (4)$$

where B is the estimated background image. While subtraction is effective, it tends to leave low contrast in darker regions of the background due to small differences between foreground and background. This motivates dividing by the background image [53, 63, 69, 82, 111].

$$I'(i,j) = \frac{C}{B(i,j)} I(i,j) \quad (5)$$

where C is a constant used to scale the contrast. When $B(i, j)$ is dark (low intensity), $\frac{C}{B(i,j)}$ will be large and will serve to emphasize the foreground text in this region. In some cases, histogram stretching is performed on the normalized image to increase contrast [85, 140].

There are a variety of ways to estimate background images. A simple method is to use a median filter with a sufficiently large window size. Nina et al. [108] use a fixed 21×21 window size, while Mitianoudis and Papamarkos [92] progressively grow the window size at each pixel until the local standard deviation stabilizes. Morphological grayscale closing [24, 70, 85, 86] and heuristic estimation [156] have also been used.

Another popular technique is to perform a high-recall initial binarization and then inpaint the foreground regions based on neighboring background intensities [39, 62, 111, 115]. Using simple binarization techniques (e.g., Niblack [106], Sauvola [146]) tuned for high recall is common choice for rough foreground estimation.

The winner of DIBCO 2009 proposed iterative polynomial smoothing [82], but this produces a highly smoothed background estimate. Perret et al. [123] used hypercomponent trees for background estimation, which combined with simple global thresholding achieved similar performance

as [82]. Vo et al. [169] proposed robust regression for background estimation with some success.

Grayscale Conversion

The majority of binarization algorithms operate over grayscale images, so RGB images are generally converted to grayscale before binarization. While most works employ simple conversion methods, such as the luminosity formula $g = 0.21r + 0.72g + 0.07b$, this can lead to decreased contrast for colored foreground ink. This problem is compounded when there are multiple foreground ink colors. For example, Hedjam et al. [48] report improvements of about 5–15% across 11 binarization methods when using their proposed grayscale conversion method on images with multiple and/or non-standard foreground ink colors.

One choice for conversion is to perform principle component analysis (PCA) over the set of RGB pixels and project each pixel onto the first principle component [92, 93, 117]. Because the majority of pixels in any document image are background, the first principle component is close to the background color. Then, after projection, background pixels will have high intensity (lighter) and foreground pixels will have low intensity (darker).

Hedjam et al. [48] proposed learning a fixed linear transform from a dataset of color images that have pixel GT labeling for text/background. The pixel features are its values in the RGB, YIQ, YDbDr, YCbCr color spaces. Then, an optimization procedure learns the optimal linear mapping to grayscale such that average text and background intensities are separated. While their results are impressive, once learned, the mapping is fixed and only reflects the text and background colors in the training dataset.

Bouillon et al. [17] proposed a gratification method that is image adaptive. First, pixel values are clustered in the YPQ color space. Then, the intensity of each pixel is taken as the Mahalanobis distance between the pixel and the center of the largest (presumably background) cluster. This approach is similar to the PCA approach as the Mahalanobis distance depends on the principle components, but this method uses all of the principle components instead of just the first one.

We note that there are a number of (non-trivial) grayscale conversion methods proposed in the literature for photographic conversion that have been relatively unexplored in the context of document image binarization.

Binarization Methods

This section surveys many of the binarization methods proposed in the last decade and constitutes the bulk of this review. While most previous reviews proceed by summarizing methods in isolation, we have attempted to organize topically according to common themes. Table 1 provides an overview of these techniques.

Classic Thresholding Algorithms

There are a number of classical thresholding algorithms for binarization that are simple to implement and have become standard tools that many later algorithms use as subroutines.

Otsu

The global thresholding algorithm proposed in by Otsu [114] remains popular, likely because it effectively handles images with uniform background and has no parameters to tune. The Otsu threshold, T_{Otsu}, is derived from the histogram of grayscale image intensity values, h, which typically has $L = 256$ bins for 8-bit images. Any chosen threshold $0 \leq T \leq L$ partitions the histogram into two clusters. The number of pixels, mean intensity, and variance of both clusters are, respectively, given by

$$w_0(T) = \sum_{i=0}^{T-1} h(i) \qquad w_1(T) = \sum_{i=T}^{L-1} h(i) \tag{6}$$

$$\mu_0(T) = \frac{1}{w_0} \sum_{i=0}^{T-1} ih(i) \qquad \mu_1(T) = \frac{1}{w_1} \sum_{i=T}^{L-1} ih(i) \tag{7}$$

$$\begin{aligned} \sigma_0^2(T) &= \frac{1}{w_0} \sum_{i=0}^{T-1} h(i)(i - \mu_0(T))^2 \\ \sigma_1^2(T) &= \frac{1}{w_1} \sum_{i=T}^{L-1} h(i)(i - \mu_1(T))^2 \end{aligned} \tag{8}$$

T_{Otsu} is then defined as the threshold that minimizes within-cluster variance:

$$T_{\text{Otsu}} = \underset{T}{\operatorname{argmin}}\, w_0(T)\sigma_0^2(T) + w_1(T)\sigma_1^2(T) \tag{9}$$

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

$$T_{\text{Otsu}} = \underset{T}{\operatorname{argmax}}\, w_0(T)w_1(T)(\mu_1(T) - \mu_0(T))^2 \tag{10}$$

Finding T_{Otsu} is done by trying all values of T and seeing which one minimizes Eq. 9 or maximizes Eq. 10. Afterward, binarization is performed globally:

$$B(i,j) = \left\{ \begin{array}{ll} 0 & I(i,j) < T_{\text{Otsu}} \\ 255 & I(i,j) \geq T_{\text{Otsu}} \end{array} \right\} \tag{11}$$

One disadvantage of Otsu or any other global thresholding method is that the background may be non-uniform, leading

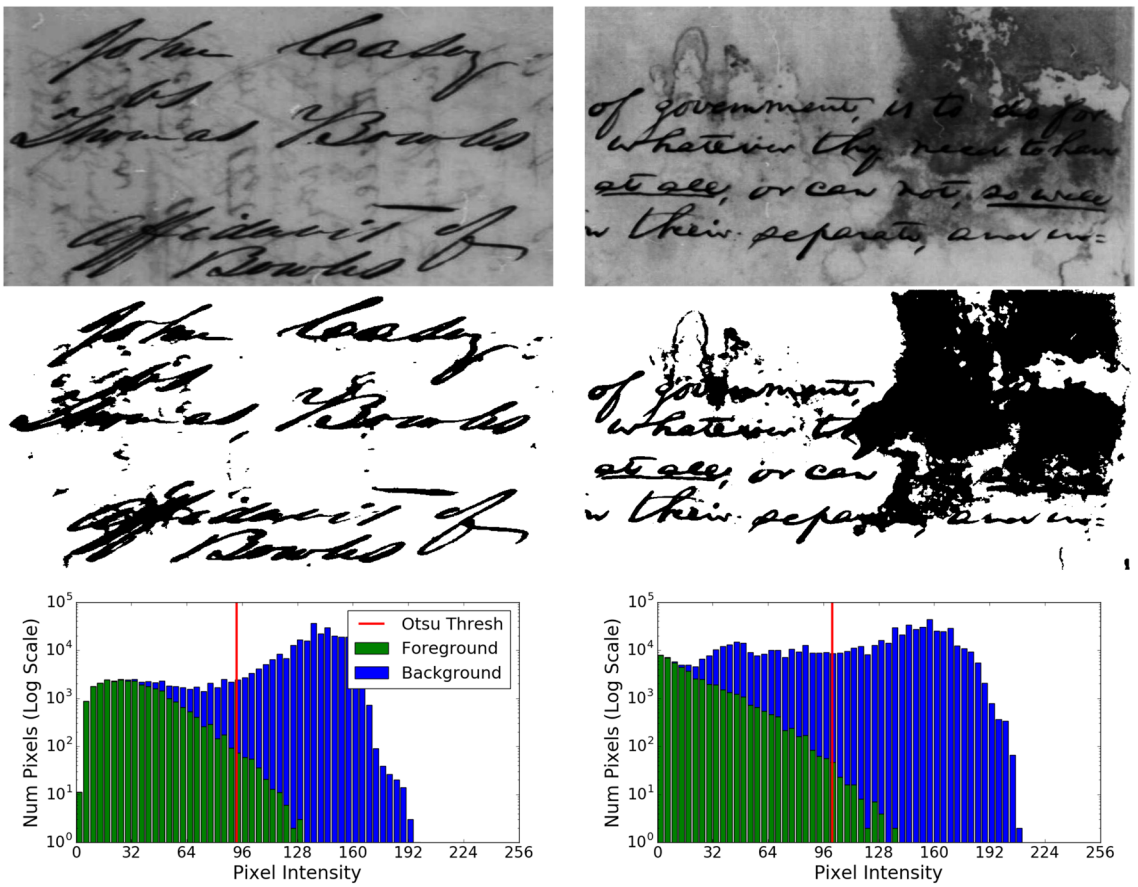
Table 1 Summary of binarization techniques

Method	Category	Description
Otsu [114]	Global threshold	Maximize between-cluster variance of pixel intensity
Niblack [106]	Local threshold	Local threshold based on window mean and std. dev.
Sauvola [146]	Local threshold	Improvement on Niblack
Wolf [173]	Local threshold	Improvement on Sauvola with global normalization
Gatos [38]	Local threshold	Threshold after background removal
NICK [64]	Local threshold	Adapts Niblack based on global mean
Bataineh [10]	Local threshold	Threshold based on local and global statistics
Saddami [144]	Local threshold	Adaptively sets k in NICK based on global std. dev.
AdOtsu [95]	Local threshold	Local version of Otsu
Lu [82]	Edge based	Local thresholding near edges after background removal
Su [162]	Edge based	Filters canny edges using local contrast
Jia [53]	Edge based	Detecting symmetry of stroke edges
Valizadeh [166]	Edge based	Adaptive water flow model
Hadjadj [43]	Edge based	Active contours initialized using contrast edges [162]
Rivest [139]	Edge based	Level-set method
Nafchi [104]	Image transform	Threshold filter response in frequency domain
Sehad [147]	Image transform	Performs background removal using Fourier Transform
Zemouri [177]	Image transform	Uses Contourlet Transform to smooth image
FAIR [73]	Mixture model	Ensemble of MoG with post-filtering
Hedjam [47]	Mixture model	MoG with spatially varying Σ_k
Mishra [91]	Mixture model	10-component MoG for foreground color variation
Mitainoudis [93]	Mixture model	MoG over pairs of intensity co-occurrences
Ramirez [138]	Mixture model	Mixture of log-normal distributions outperforms MoG
Howe [50]	CRF	Laplacian unary term and pairwise Canny-based term
Ayyalasomayajula [7]	CRF	Pairwise terms based on an initial binarization
Peng [120]	CRF	Pairwise terms based on an initial foreground skeleton
Su [161]	CRF	Uses CRF to classify uncertain pixels
Ahmadi [2]	CRF	Learns linear combination of feature functions
GiB [13]	Game theory	Extracts features for clustering using game theory
Hamza [44]	Shallow ML	Self-organizing map to cluster pixels
Rabelo [132]	Shallow ML	MLP to classify pixels using local mean
Kefali [58]	Shallow ML	MLP using local intensities and global statistic features
Pastor [118]	Shallow ML	MLP with F-measure loss function
Kasmin [56]	Shallow ML	Ensemble of 8 SVMs
Wu [174]	Shallow ML	Random forest trained on a rich feature set
Pastor [119]	Deep learning	First CNN for binarization
Peng [121]	Deep learning	Encoder–decoder FCN trained on synthetic data
Calvo-Zaragoza [21]	Deep learning	Encoder–decoder FCN with residual blocks
Tensmeyer [164]	Deep learning	Multi-scale FCN trained with pseudo-FM loss
Vo [169]	Deep learning	Combine multi-scale outputs of FCN
Peng [122]	Deep learning	Multi-scale FCN trained with DRD loss and ConvCRF
PDNet [8]	Deep learning	FCN integrated with Total Variation smoothing
Mondal [98]	Deep learning	2D morphological networks
Kang [55]	Deep learning	Pretrain U-Net networks with image operations
Tensmeyer [165]	Deep learning	Use CycleGAN to generate realistic synthetic data
Zhao [178]	Deep learning	GAN discriminator judges prediction-image agreement
Bhunia [14]	Deep learning	Train model using unpaired training data
Krantz [68]	Deep learning	Find smaller, diverse training set by clustering images
Afzal [1]	Deep learning	Uses 2D BLSTMs to model arbitrary context
Westphal [171]	Deep learning	Grid LSTMs
Cheriet [27]	Supervised tuning	Predicts parameters for Sauvola [146] and Lu [82]

Table 1 (continued)

Method	Category	Description
Chattopadhyay [23]	Supervised tuning	SVM classifier chooses best binarization algorithm
Xiong [176]	Supervised tuning	SVM chooses global threshold for image blocks
Westphal [170]	Supervised tuning	Predicts parameters for Howe [50]
Messaoud [87]	Supervised tuning	Detects 4 types of image noise to choose parameters
Ntirogiannis [111]	Unsuper. tuning	Estimates local window sizes based on stroke width
Boiangiu [15]	Unsuper. tuning	Chooses window size for Niblack based on std. dev.
Liang [74]	Unsuper. tuning	Combines local/global thresholds with a cost function
Ramirez [135]	Unsuper. tuning	Selects binarization algorithm using statistical functions

MoG mixture of Gaussians, *CRF* conditional random field, *ML* machine learning, *SVM* support vector machine, *FCN* fully convolutional network, *LSTM* long short-term memory

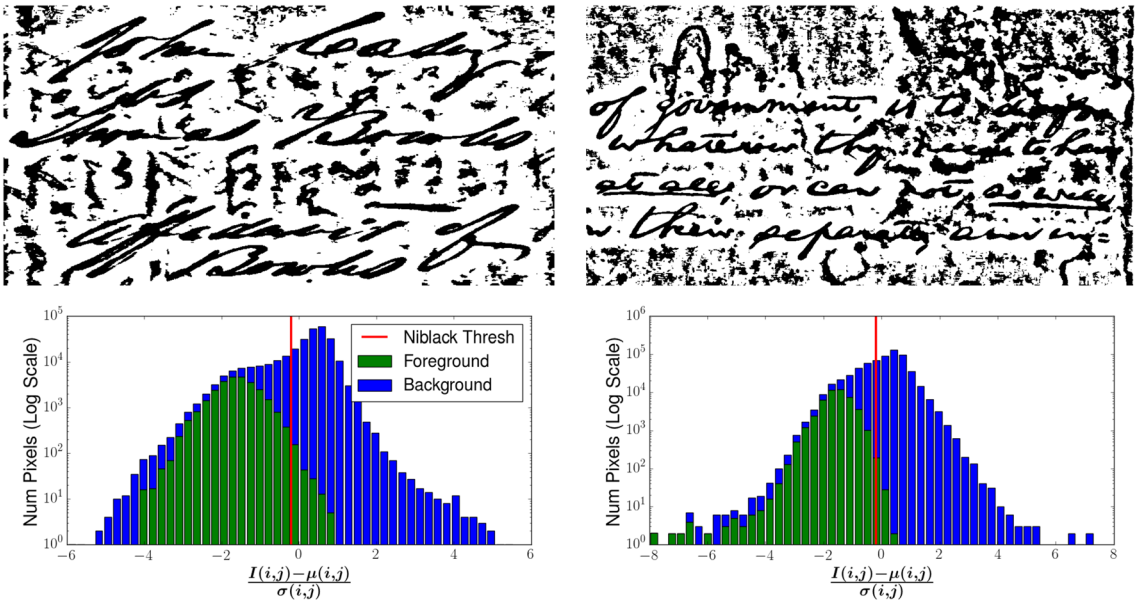


(a) DIBCO09 handwritten image 3.

(b) DIBCO09 handwritten image 4.

Fig. 2 Input image (top), Otsu binarization (middle), and comparative histogram (log-scale) of gray values for foreground and background pixels (bottom). The overlap of foreground and background intensities indicates that no global thresholding algorithm can produce a perfect binarization. The vertical red line shows the threshold computed by Otsu's method

to some background pixels being darker than some foreground pixels (Fig. 2). One consequence of this is that no global thresholding method can perfectly binarize such an image.



(a) DIBCO09 handwritten image 3.

(b) DIBCO09 handwritten image 4.

Fig. 3 Niblack binarization (top) and comparative histogram (log-scale) of normalized gray values for foreground and background pixels (bottom). Input images are the same as shown in Fig. 2. After normalizing the image by $I'(i,j) = \frac{I(i,j)-\mu(i,j)}{\sigma(i,j)}$, Niblack thresholding reduces to a global thresholding by k (red line); here it is set to -0.2. A 51×51 window was used to compute $\mu(i,j)$ and $\sigma(i,j)$

Niblack

Niblack proposed a simple local adaptive threshold, where a threshold is determined for each pixel based on statistics computed from a local window centered on the pixel of interest [106]. Because the threshold is adaptive, it can potentially handle cases of foreground and background intensity distribution overlap (e.g., compare Fig. 2 to Fig. 3). Specifically, Niblack thresholding uses the local mean and local standard deviation:

$$\mu(i,j) = \frac{1}{w^2} \sum_{i'=i-w}^{i+w} \sum_{j'=j-w}^{j+w} I(i',j') \tag{12}$$

$$\sigma(i,j) = \sqrt{\frac{\sum_{i'=i-w}^{i+w} \sum_{j'=j-w}^{j+w} (I(i',j') - \mu(i,j))^2}{w^2}} \tag{13}$$

where w is called the window size and controls how much context is used to compute these statistics. The per-pixel Niblack threshold is then

$$T_N(i,j) = \mu(i,j) + k\sigma(i,j) \tag{14}$$

where k is a user-set parameter that controls the trade-off between foreground detection precision and recall. The recommended parameter setting is $k = -0.2$, though the optimal k depends on the image and chosen window size.

Binarization is then accomplished with

$$B(i,j) = \left\{ \begin{array}{ll} 0 & I(i,j) < T_N(i,j) \\ 255 & I(i,j) \geq T_N(i,j) \end{array} \right\} \tag{15}$$

One issue with Niblack is when the window covers only background pixels, it causes the darkest background pixels to be set to foreground (Fig. 3). While this noise is often large, the background immediately around the text is correctly identified, which makes Niblack thresholding useful in combination with other binarization techniques.

Sauvola

Sauvola and Pietikäinen [146] proposed a variant of Niblack to solve the problem with background-only windows.

$$T_S(i,j) = \mu(i,j)\left[1 + k\left(\frac{\sigma(i,j)}{R} - 1\right)\right] \tag{16}$$

where $\mu(i,j)$ and $\sigma(i,j)$ are computed as in Niblack (Eqs. 12, 13), $k = 0.5$ is the recommended value for the user-set parameter, and R is a constant set to the maximum possible standard deviation, i.e., $R = 128$ for 256 gray levels.

While Niblack takes $\mu(i,j)$ and adjusts downward based only on the $\sigma(i,j)$, Sauvola adjusts downward based on $\mu(i,j)\sigma(i,j)$. In windows of only background, $\mu(i,j)$ is

relatively large, so $T_S < T_N$, which means fewer of these background pixels are set to foreground.

Wolf

Wolf binarization [173] is an extension of Sauvola, where the local statistics are normalized based on global statistics:

$$T_W(i,j) = \mu(i,j) - k\left(1 - \frac{\sigma(i,j)}{S}\right)(\mu(i,j) - M) \tag{17}$$

where $S = \max_{ij} \sigma(i,j)$, $M = \min_{ij} \mu(i,j)$. This allows for better handling of images with limited contrast and limited range of grayscale intensity.

Image Processing

The majority of binarization techniques employ one or more image processing techniques. Here, we have roughly categorized these techniques into thresholding, edge detection, image transforms, and region-based methods, but we recognize that much of the preprocessing ("Preprocessing" section) and post-processing ("Post-processing" section) techniques also involve image processing.

Thresholding

Inspired by classic algorithms, many approaches are based on local thresholding, sometimes combined with preprocessing. One competitive example of this is by Gatos et al. [39], where after background subtraction ("Background Estimation" section), the local threshold is based on an estimate of the average distance between foreground and background pixels. Then, to compensate for low contrast regions, the threshold is adapted using a logistic sigmoid function to produce lower thresholds when the estimated background region is dark.

NICK thresholding [64] (named for the authors initials) perturbs T_{Niblack} based on the global image mean, μ, to lower the number of false positive foreground components. From Eq. 14, we have $T_{\text{Niblack}}(i,j) = \mu(i,j) - k\sqrt{B}$, where $B = \sigma^2(i,j)$. The NICK threshold has a similar form with $T_{NICK} = \mu(i,j) - k\sqrt{A}$, where $A \approx B + \mu^2$ for large window sizes. Bataineh et al. [10] proposed a local threshold computed from both the local and global means and standard deviations that was shown to outperform NICK on DIBCO 2009, though parameters were not necessarily tuned for NICK. Adaptively setting the k value in NICK thresholding based on the global standard deviation was shown to improve performance over a fixed k [144].

While many methods use all pixels in the local window, [53] as well as the winners of DIBCO 2009 [82] and 2013 [162] determines local thresholds based on the intensity of detected edge pixels. Additionally, in [82], foreground pixels must be near $N_{\min}$ number of edge pixels and have a lower intensity than the average nearby edge pixels. Estimating the stroke width (see "Parameter Tuning" section) is often used to determine the window size for edge-based local thresholding.

AdOtsu [95] is a local version of Otsu ("Otsu" section) where the threshold is computed from a local window rather than the global histogram. In addition, there is a switching behavior to use an alternative constant threshold when regions are similar to the estimated background image.

Edge Detection

Several approaches perform edge detection to find the boundary between foreground and background. A popular method is Canny edge detection [22], which finds single pixel wide edges by performing non-maximal suppression and hysteresis thresholding over the gradient magnitude image computed from the input image after Gaussian smoothing. The hysteresis thresholding process first finds strong edges using a high user-set threshold, τ_1, and then expands each strong edge to include edge pixels with gradient magnitude above τ_2, where $\tau_1 > \tau_2$. The winner of DIBCO 2013 [162] filtered edges detected by Canny using a novel measure of local contrast:

$$C(i,j) = \alpha\frac{M(i,j) - m(i,j)}{M(i,j) + m(i,j) + \epsilon} + (1-\alpha)(M(i,j) - m(i,j)) \tag{18}$$

where $M(i, j)$ is the image local maximum, $m(i, j)$ is the image local minimum, α increases with the global image standard deviation, and ϵ is a small constant to prevent division by 0 [162]. Without manually tuning parameters, it is difficult to eliminate false edges with Canny, and while $C(i, j)$ does a good job of eliminating false positives, it only roughly localizes the edges, even if a small window size is used to compute M and m. After this process, spurious horizontal edges may still remain in Palm Leaf Manuscripts (PLMs) due to the background texture of palm leaves, but many of these edges can be removed by examining the aspect ratio of connected edge components [121].

Binarizing based on pairs of structural symmetrical pixels (SSPs) leverages the idea that edge pixels on opposite sides of the stroke should have large gradient magnitudes and opposite orientations [53]. While spurious edges may be detected due to background variations, the noise edges are less likely to be symmetrical and thus are not SSPs. An adaptive water flow model is proposed in [166], where water is poured on detected edges and allowed to flood low intensity basins at a rate and amount controlled by edge strength and stroke edge. After filtering spurious regions, all pixels covered by water are designated to be foreground.

One issue with using edge detection for binarization is that the detected edges do not completely surround the foreground regions. In contrast, active contour methods constrain the detected edges to be closed contours that directly imply a segmentation. Hadjadj et al. [43] use Eq. 18 (with $\alpha = 1$) to initialize closed contours and then optimize those contours according to forces that attempt to align the contours to maximize local intensity differences and minimize the contour length and area.

Image Transforms

Some methods have experimented with extracting local features based on image transforms such as Gabor filter banks, Fourier Transform, and the Contourlet Transform.

In [104], thresholding the response of log-Gabor filters is used to identify the pixels that correspond to maximal local phase components in the Fourier frequency domain. Sehad et al. [147] find the dominate document slant angle using the Fourier Transform and then construct a weighted Gabor filter bank based on this angle to smooth the image. Then, for local thresholding, they compute the local standard deviation from the Gabor-filtered image and compute the local mean from the input image.

The Contourlet transform is composed of a Laplacian pyramid decomposition to obtain multi-scale analysis, and a directional filter bank is applied at each scale [177]. The resulting contourlet coefficients give a more concentrated representation of the image edges, which are thresholded using Niblack. Then, the Inverse Contourlet Transform is applied to produce a smoothed image that is fed to a local thresholding method.

Statistical Models

While the image processing methods presented in "Image Processing" section can perform well, they aren't grounded in a formal framework. In this section, we review statistical methods based on Mixture of Gaussians (MoG) and Conditional Random Fields (CRF), in addition to other methods based in probability theory.

Mixture of Gaussians

A MoG model is a parametric probability distribution whose probability distribution function (pdf) is a weighted sum of Gaussian distributions,

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma_k^2) \tag{19}$$

where $\mathcal{N}$ is the pdf of a Gaussian distribution, K is the number of mixture components, and $\sum_{k=1}^{K} \pi_k = 1$. The MoG can be fit to a set of unlabeled data points (e.g., pixel features) using expectation maximization (EM). Afterward, each data point can be assigned to its most probable cluster, and the clusters can be assigned to either the foreground or background class to induce a binarization over the entire image.

How to apply the MoG in binarization varies among the published literature. For example, foreground and background have been modeled each with a single component using intensity or color as the pixel features [47, 73]. Both these methods extend the basic Gaussian mixture component to have a spatially varying μ_k, and [47] uses a spatially varying Σ_k. In this manner, a single component can model spatially varying changes in the statistics of foreground or background pixels. The FAIR algorithm [73] combines this MoG approach with some preprocessing and post-processing to obtain state-of-the-art results.

In contrast, Mishra et al. [91] use 5 components for foreground and background and use both stroke and color features. Though [91] primarily targeted scene text segmentation, where foreground text may be multiple colors, it performs competitively with the state of the art on DIBCO datasets. Mitianoudis and Papamarkos [93] perform MoG clustering over pairs of intensity values that co-occur in local windows combined with a local contrast feature.

Though a MoG can converge to any arbitrary distribution (given enough components), an empirical study over handwritten images found that foreground and background patches far from character boundaries have intensities that are approximately Gaussian distributed [138]. However, including the (transition) pixels near boundaries creates skew in the foreground and background distributions. Due to this asymmetry, a mixture of log-normal distributions provides a better fit and increases binarization performance compared to a MoG [138].

Conditional Random Fields

CRFs provide an excellent framework for including spatial dependencies among pixels into the binarization process. A CRF is governed by an energy (cost) function that scores how well a given binarization agrees with the input image. For tractability, this energy function is decomposed into local energy functions over sets of pixels. This decomposition could be over sets of any size, though it is typically defined over single pixels and pairs of neighboring pixels:

$$E(B, I) = \sum_{i=1}^{N} E_i(B_i, I) + \sum_{i=1}^{N} \sum_{j=1}^{i} E_{ij}(B_i, B_j, I) \tag{20}$$

where I is the input image with N pixels, B is a discrete segmentation of I, and E_i, E_{ij} are, respectively, the decomposed unary and pairwise energy functions, which optionally can vary based on i or i, j. Though general CRFs can have any number of segmentation classes, for binarization it is typical

to have 2 classes. Typically, E_1 biases individual pixels to be either foreground or background based on pixel intensity, and E_2 yields low values when adjacent, similar pixels are assigned to the same class.

Given that Eq. 20 scores the goodness of a given binarization, we formulate the binarization inference task as energy minimization:

$$\hat{B} = \underset{B}{\operatorname{argmin}} E(B, I) \tag{21}$$

An exact solution to Eq. 21 can be found for binary segmentation tasks if E_{ij} satisfies the following sub-modularity criteria [66]

$$\forall i, j \; E_{ij}(0,0,I) + E_{ij}(1,1,I) \leq E_{ij}(0,1,I) + E_{ij}(1,0,I) \tag{22}$$

If the above is true, then E is can be transformed to a graph with nonnegative edge weights, and a minimum cut over this graph corresponds to $\hat{B}$ [66]. The minimum graph cut problem can be reformulated as the dual max-flow problem, for which there are many algorithms and off-the-shelf solvers [18].

The popular Howe method [50] uses a CRF, where E_i is set to be the image Laplacian. The pairwise energy, E_{ij}, is 0 in uniform regions and discontinuities occur at edges. Specifically, E_{ij} gives a uniform penalty when $B_i \neq B_j$ unless pixels i and j are on opposite sides of detected Canny edges [22]. The edges used to define E_{ij} are improved in [7] using an initial binarization obtained by clustering pixels. Howe's method (and variants to introduce pre- and post-processing) placed first in DIBCO competitions for 2012, 2014, 2016, and 2018 [112, 126, 128, 130].

In [120], a CRF is used to improve an initial binarization, where E_{ij} is determined by the distances between foreground pixels, background pixels, and the foreground skeleton of the initial binarization. Similarly, [161] performs an initial ternary classification into foreground, background, and uncertain before using a CRF to reclassify the uncertain pixels based on detected edges and similarity to nearby foreground and background pixels.

The previously described CRFs use handcrafted energy functions and do not learn parameters. In contrast, Ahmadi et al. [2] define E_i and E_{ij} as a linear combination of feature functions based on window statistics, histogram of oriented gradients (HOG) [30], local binary patterns (LBP) [113], and Canny edges [22]. The linear combination weights (feature importance) are then learned from training data and fixed for novel test images.

Other Statistical Approaches

The idea of a transition pixel was introduced in [133] and later expanded upon [134, 137]. These works define a statistical criteria for modeling pixels near foreground–background transitions that is used for image thresholding. Bhowmik et al. [13] first extract pixel features from a preprocessed image using ideas from game theory before clustering with K-means. This technique ranked second overall in the DIBCO19 competition and achieved state-of-the-art results on previous DIBCO data.

A level-set method for binarization is presented in [139], where the contours separating foreground and background are evolved according to forces that direct the contours to be smooth, have minimum area, and adhere to strong edges. One of the proposed forces is based on modeling foreground and background intensities using locally linear models, and the contours are adjusted to increase the likelihood of the segmentation under this model.

Pixel Classification

In the last few years, machine learning methods, particularly deep learning models, have become the state of the art in historical document binarization. In 2017, the top 6 submissions to DIBCO all used a deep model [129].

The majority of learning methods adopt a framework of directly classifying pixels, while others use machine learning to predict other quantities such as optimal parameter settings or methods. The latter approach is covered in "Parameter Tuning" section, and the pixel classification approaches are covered here. First, we review shallow models and then deep models.

Shallow Models

Hamza et al. [44] used a self-organizing map (SOM) to cluster pixels with similar intensity and a multi-layer perceptron (MLP) to classify them as foreground or background based on intensity. An MLP was also used in [132] to classify individual pixels in Brazilian bank checks that have complex background due to check security features and watermarks. This MLP took as input the pixel intensity as well as the average intensity of 4 3×3 windows around the pixel. In contrast, the MLP in [58] used the raw intensities in a 3×3 window and the global mean and standard deviation to classify the central pixel. However, such small context windows restrict the model to learn very simple features that do not generalize across a wide variety of documents. Pastor-Pellicer et al. [118] used 90 inputs composed of the 81 raw intensity values of a 9×9 window plus 9 intensity values of the estimated background for a 3×3 window. In addition, the authors proposed a new loss function which is a continuous adaptation of the F-measure evaluation metric and showed this resulted in improved MLP training.

Ensembles of shallow models have also proved effective in classifying pixels. In [56], an ensemble of 8 support

vector machines (SVMs) are trained to classify pixels in different positions in a 3×3 window, leading to an error reduction of 25% compared to a single model. Wu et al. [174] used a variant on a random forest ensemble combined with a large, rich feature set to achieve competitive performance. Their features include thresholds determined by classic algorithms, statistics over multiple scales, and histograms for several (non-centered) regions. Another key element was constructing a training set of difficult pixels based on the disagreement regions of other algorithms.

Deep Neural Networks

While early uses of deep Convolutional Neural Networks (CNN) involved small images for character recognition [71, 153], they became popular for large image classification after a CNN was used to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [143]. In [80], CNNs were transformed to Fully Convolutional Networks (FCNs) for semantic segmentation, which is defined as a task where each pixel is classified as one of K classes. Binarization is a semantic segmentation problem with $K = 2$, and methods based on FCNs have proved very effective on this task. However, the basic architecture of [80] only obtains 68% F-measure on DIBCO 13 and 14 [169] and has been extensively modified by the following approaches.

Pastor-Pellicer et al. [119] first explored a CNN for binarization, finding that a CNN with an appropriate architecture outperforms a non-convolutional deep MLP even if the MLP is augmented with additional histogram and median filter features.

Both Peng et al. [121] and Calvo-Zaragoza and Gallego [21] adopted an encoder-decoder architecture, where the encoder progressively downsamples the input image to pool information over increasingly larger input regions. The decoder then performs upsampling to produce a binarization at the original input resolution. In [121], the encoder-decoder is trained on synthetic data in steps, where additional layers at a smaller resolution are added during training. Then, each scale makes a prediction, which is combined using a CRF. Calvo-Zaragoza and Gallego [21] explored 3 basic network blocks and found that residual blocks perform the best. They also find a large gap in performance between models trained on specific document domains (e.g., DIBCO, PLM, PHIBD) and a generalist model.

Tensmeyer and Martinez [164], Vo et al. [169], and Peng et al. [122] proposed FCNs that learn features at multiple scales. While the single model proposed in [164] fuses features learned in parallel at multiple scales, in [169] there are 3 separate networks trained independently at different input image scales. The predictions of the 3 models are combined by taking the union of the individually predicted foreground pixels, which allows for both fine and coarse predictions. Peng et al. [122] used 3 U-net architectures with attention layers that take different sized inputs and are trained with a novel distance reciprocal distance (DRD) loss to achieve better perceptual quality.

The previously discussed FCNs do nothing to enforce spatial smoothness on the output predictions, which has led to some post-processing steps. For example, the model of [164] was submitted to DIBCO 2017 [129] both with and without fully connected CRF post-processing [67]. In contrast, PDNet [8] integrates individual pixel probability prediction with a smoothness term based on total variation. This improves performance by jointly learning (end to end) the pixel probabilities and output smoothness, rather than smoothing outputs in a post hoc fashion. 2D morphological networks [98] use grayscale dilation and erosion operations in the network blocks and achieve competition performance on DIBCO17 and HDIBCO18.

One of the limitations of deep learning models is the availability of labeled data for training. In [55], this is addressed by pretraining models to perform classical image operations including erosion, dilation, histogram equalization, and Canny edge detection. The pretraining greatly improves model performance to achieve state-of-the-art results on DIBCO17. Creating realistic synthetic data with pixel GT using a CycleGAN model for pretraining increased FM by 1.4% [165]. Similarly, a GAN discriminator is used to improve a prediction model in [178]. Bhunia et al. [14] proposed a unique approach to train a binarization network using *unpaired* training data (i.e., the grayscale and binary images do not correspond) and achieved an impressive 97.8% FM on DIBCO13. To reduce the amount of needed training data, Krantz and Westphal [68] proposed a clustering method to ensure only diverse data are labeled. This reduces dataset size by 50% at a modest loss in accuracy.

Recurrent neural networks (RNNs) for binarization have also been proposed, though they are not as effective as FCN-based approaches. In [1], 4 small 2D BLSTMs (bidirectional long short-term memory) are used to model an arbitrarily large context for pixel classification. The 4 models process the image in different directions (top–down, left–right; top–down, right–left; down–top, left–right; down–top, right–left) and fuse their features before making a prediction. Westphal et al. [171] instead use grid LSTMs and report performance near that of [169]. In grid LSTMs, each pixel has a set of memory cells for both vertical and horizontal directions.

Class Imbalance

One classic problem for machine learning algorithms is when the training and/or test data are heavy skewed to one class (e.g., background pixels). This is addressed in [118] by training to optimize the F-measure, which causes the

model to balance the false positive and false negative errors it makes. In [164], the pseudo-F-measure (see "7.2.2" pseudo-F-measure section) loss is used for training, which heavily penalizes errors made near the foreground text. Similarly, [171] uses the per-pixel weights from pseudo-F-measure to weight the per-pixel contribution for a cross-entropy loss.

Another method to address class imbalance is to subsample the background class. This is done in [174] by sampling a training set from disagreement regions of other algorithms. In [36], training patches are sampled according to the standard deviation of intensity, which oversamples foreground regions.

Parameter Tuning

Many binarization methods have tunable parameters such as window sizes or constants used in computing thresholds. The output of these algorithms is generally sensitive to these parameters. For example, the reported F-measure for Sauvola on DIBCO 2009 ranges from 69.54 [37] to 87.26 [41] in the literature, presumably due to different parameter settings used by authors. Even the choice of which binarization algorithm to use for a particular image can be considered a meta-parameter, as it has been repeatedly shown that no single algorithm performs best for all images [76]. Parameter tuning approaches come in 3 flavors: supervised, unsupervised, and interactive.

Supervised

In supervised parameter tuning, there are two primary approaches taken. In the first, a set of training images is used to determine a static set of parameters, which are then applied to each new image without modification. The second approach is more adaptive in that the training set is used to learn a model that predicts the optimal setting of parameters conditioned on the image. These approaches are similar to pixel classification ("Pixel Classification" section), but indirectly infer pixel classes by instead predicting parameters.

The static scenario can be posed as

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} M(f(I_i;\theta), B_i) \tag{23}$$

where θ is the set of parameters being optimized, f is a particular binarization algorithm, $\{(I_i, B_i)\}$ is a training set (size N) of input images I_i with corresponding GT B_i, and M is a metric to be maximized such as F-measure. While a brute force search is viable for a limited number of parameters (as it only needs to be done once), more efficient search strategies have been employed in the literature. These include I/F race [86], Bayesian optimization [167], and swarm optimization techniques [33, 45].

A general formulation of the conditional setting is

$$\hat{\psi} = \underset{\psi}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^{N} M(f(I_i;g(I_i;\psi)), B_i) \tag{24}$$

where g is a learning model with parameters ψ that predicts $\theta_i = g(I_i;\psi)$. For a test image, I, the prediction is made by $f(I, g(I;\hat{\psi}))$. Cheriet et al. [27] predicted parameters for Sauvola and Lu [82] binarization methods, attaining results very close to those obtained using optimal parameter settings. Chattopadhyay et al. [23] segment an image into blocks and then use a SVM classifier (trained to optimize OCR performance) to choose the best binarization algorithm to apply to each region based on region statistics. A similar approach was used in [176], but instead a global threshold is predicted for each block. Westphal et al. [170] replace the unsupervised parameter tuning of Howe [50] with a supervised approach and report improved performance.

An alternate formulation where g is selected based on a training and validation set is presented in [88], and [87] uses a handcrafted g to detect 4 types of image noise and provides appropriate parameters for the detected noise type.

Unsupervised

In unsupervised parameter tuning, parameters are tuned using only the information contained in the test image. A common parameter that is estimated in this way is the average stroke width [16, 79, 82, 91, 94, 166]. One way to estimate the stroke width is to perform an initial binarization (e.g., Sauvola) or edge detection with Canny. Then, create a histogram for the distances between each pair of successive edge pixels on all horizontal scan lines. Then, the stroke width can be estimated as the most frequent distance between the edge pixels [16, 94]. In [53], this is done in an iterative fashion as the binarization result is based on the stroke width and the stroke width is estimated from the binarization.

The window size is often based on the stroke width [111], but other ways are used to estimate it based on the idea that the window should be large enough to cover both foreground and background pixels. In [10], pixels are initially sorted into foreground, background, and low confidence. The window size for thresholding low-confidence pixels is chosen to ensure that a sufficient number of foreground pixels are included in each window. The methods of [15, 29, 92] gradually increase the window size until reaching a stopping criteria based on window standard deviation.

Howe [50] tunes two critical parameters to minimize a criteria that measures binarization output instability. In [74], the weighted average of a globally and a locally computed

threshold is determined by minimizing a cost function. Several statistical cost functions were evaluated in [135] and found to correlate well with OCR performance.

Interactive

Interactive parameter tuning requires the input of a user to improve binarization performance for a particular image. In [83, 84], the user is able to provide feedback to the algorithm by indicating approximate regions that were poorly binarized. This feedback can take the form of scribbles around the region [84], which after segmentation is binarized again with an adjusted threshold. The user feedback in [83] is class based, where the user indicates small regions that are foreground, bleed-through, and background. This small feedback is propagated to other regions, and a new binarization result is produced that can be further refined by the user.

Sokratis and Kavallieratou [157] proposed an interface that allows users to manually set parameters and provide global feedback for the resulting binarization. Based on the feedback, the interface suggests a modification to the parameters. The suggestions, however, are based on predetermined rules, which limits its applicability in new algorithms.

Post-processing

Once a binarization mask has been produced, it can often be improved by post-processing. This is mostly a heuristic process, involving morphological operations based on apriori knowledge of how binarization masks of text should look. Other techniques are based on comparing the resulting binary image to the input grayscale image or based on combining multiple binarized images.

Morphological Approaches

The main purpose of morphological post-processing is to smooth the output binary image to remove noise. Generally, this involves removing small foreground components and smoothing the edges of other foreground components. To obtain the components, connected component analysis is performed using 4-connected or 8-connected neighborhoods.

Removing small foreground components is typically done by counting the number of pixels in the component and comparing it to a threshold. This threshold in the literature ranges from conservative 1–4 pixels [24, 117, 133, 162] to as large as 60 pixels [147]. Other methods set this threshold adaptively, such as 10% of the average foreground component size [90]. While using a larger threshold is likely to remove more spurious components, there are legitimate small foreground components such as the dot of an *i* or diacritic marks that may be removed by this process.

Smoothing the edges of foreground components can be done in several different ways. One simple way is to perform a morphological closing operation followed by a morphological opening operation [123, 147]. Lu et al. [82] explicitly remove single pixel concavities and convexities. Shrink and swell filters have also been used, where pixels are flipped if the number of opposite class pixels in a local window is above some threshold. In [39], the thresholds and window sizes were determined based on the average character height. To recover faint strokes, [26] constructs a bidirectional graph over extracted strokes and then applies a swell filter to try to connect strokes that are detected as broken.

Some methods that explicitly detect edge pixels ensure that pixels on opposite sides of the detected edge are different classes [24, 162].

Approaches Based on Grayscale Image

Besides morphology, other post-processing approaches have been proposed. For example, local thresholding based on both the binary and gray images has been used to fill in the interior of characters with thick strokes [11] and to connect broken strokes [9]. In [117], these regions are called *white islands* and are identified by performing a two-sample z-test of the mean gray value of the white island and the mean gray value of the surrounding pixels. Other approaches for fixing misclassified background pixels involve comparing the intensity of each background pixel to the intensities in the surrounding window [62, 101].

Combining Multiple Binarizations

Combining the output of 2 or more binarization methods is a good way to improve overall binarization quality. One strategy is to compute two binary images where one has high precision of foreground components even if it does not capture the full extent of each component [5, 11, 42, 111, 117, 136]. The other image has high recall, but could have many false positive foreground components (e.g., the output of Niblack). These images can be combined by retaining only the foreground components of the high-recall image that have sufficient overlap with one of the components in the high-precision image. This serves to filter the false positives to greatly improve the precision of the high-recall image.

An unsupervised Ensemble-of-Experts method was proposed in [96], which analyzes the outputs from a large number of binarization algorithms to form a weighted consensus graph. The idea is to find the correct cluster of experts that agree on a given test image and produce a final binarization using a majority vote of the selected experts. Though an ensemble of all submissions to HDIBCO 2012 was unable

to outperform the best system in terms of F-measure, an ensemble of Howe's method [50] using different parameter settings was able to outperform the parameters chosen by the automatic tuning [50]. Similarly, a supervised ensemble of experts combined with a trained CRF was shown to outperform the best of the ensemble [46]. Akbari et al. [3] train an FCN where one of the input channels is the output of SSP binarization [53]. This combination outperforms both FCN and SSP in isolation by 0.5–2.0% F-measure on DIBCO data.

Su et al. [160] proposed a way to combine the output of two binarization methods using features extracted from the input gray image to classify the pixels that the two binary outputs do not agree on. To incorporate more than 2 methods, the combined output of 2 methods can be compared to a third output and so on. On DIBCO 2009, they were able to combine two methods to achieve an average F-measure of 93.18, where the individual binarization methods had average F-measures of 91.06 and 91.24. An extension of this method replaces the handcrafted classification rules for disagreement pixels with sparse reconstruction classification [163]. For each uncertain pixel, a small set of similar patches centered on pixels that the two methods agreed on are used to determine the class of the uncertain pixel.

Binarization Efficiency

Some binarization algorithms are too slow to process very large collections (e.g., 50 million [171]) or to be used interactively [31, 84]. For example, the winning algorithm of the 2014 HDIBCO took 17 seconds to process an average sized image [112]. Such considerations have pushed some researchers to search for ways to speed up binarization algorithms on both CPU and GPU.

CPU

The primary ways that CPU-based algorithms have been made more efficient is through grid-based and integral image techniques.

Grid-Based Methods

In grid-based methods, certain quantities (e.g., local thresholds) are computed for only a subset of pixels on a regular grid (i.e., all pixels whose coordinates are divisible by $N \in \mathbb{Z}^+$). Under the assumption that these quantities smoothly vary across the image, the value for non-grid pixels can be computed through interpolation, which is typically much faster than computing the exact quantity. In [94], grid-based Sauvola is proposed by computing the local threshold for only the grid pixels and inferring the rest of the local thresholds through interpolation. The grid approach was shown to be 640x faster and yielded improved binarization quality compared to basic Sauvola due to the imposed smoothness of the local threshold. While this was shown for Sauvola, this technique can be applied to other local thresholding methods and other quantities (e.g. [47]).

Integral Images

The integral image, L, of image I contains cumulative sums of I at each pixel:

$$L(i,j) = \sum_{i'=1}^{i} \sum_{j'=1}^{j} I(i',j') \tag{25}$$

Once L has been computed (in $O(\mathrm{HW})$ time), the sum of any rectangular region defined by (y_1, x_1, y_2, x_2) of I can be computed in constant time.

$$\sum_{i'=y_1}^{y_2} \sum_{j'=x_1}^{x_2} I(i',j') = L(y_2,x_2) - L(y_1,x_2) - L(y_2,x_1) + L(y_1,x_1) \tag{26}$$

For local thresholding methods that make heavy use of computing statistics over local windows, this can provide significant speedup (5–20×), reducing computational complexity from $O(\mathrm{HW}K^2)$ to $O(\mathrm{HW})$, where K is the chosen window size [151].

Quantities other than sums can be computed by first transforming I before computing L. For example, the integral image of $I^2(i,j)$ can be used to compute the local variance [133, 151]. Integral images can also be used for vector valued transformations of I such as grayscale histograms [107].

GPU

Graphical processing units (GPUs) are designed for efficient single instruction multiple data (SIMD) parallel computing. The most common use of GPUs in binarization is for deep learning models ("Deep Neural Networks" section), which have a high computational burden (e.g., 125 GFLOPs [80]). The CuDNN library [28] provides efficient GPU routines for both training and inference, and it has been integrated into a variety of learning frameworks. However, the recently proposed deep models have not given much consideration to speed, so while they may win contests [129], their use in workflows that require high throughput or low latency may be limited. Techniques to make networks smaller and faster without losing accuracy have been studied in the deep learning literature (e.g. [141]).

Hybrid approaches that utilize both CPU and GPU have been investigated. In [25], Sauvola is applied to the image

Fig. 4 Example Palm Leaf Manuscript. The hole is the center is used for binding several leaves together

several times on the GPU with multiple sets of parameters. Then, SVM-based voting and reconstruction is performed on the CPU.

Westphal et al. [170] performed an extensive study of a CPU/GPU implementation of Howe binarization [50], finding that the fastest configuration performs all steps on the GPU, achieving a 3.5× speedup compared to the reference implementation. They implemented a parallel algorithm to solve the graph-cut energy minimization step on the GPU because the original solver used in [50] is a serial algorithm.

Datasets

A number of datasets for binarization of degraded documents with GT pixel labeling have been proposed and used for evaluation in the literature.

The main dataset used in the literature comes from the DIBCO series of competitions [40, 112, 124–131]. Each year since 2009 (except 2015), a new dataset of 10–20 handwritten/machine-printed images has been released, totaling 136 images in 2019. The images are chosen to have "representative degradations" (e.g., see Fig. 1) and come from a variety of collections and libraries [125]. All images contain text from a Latin-based language. In evaluation, some authors report results on each dataset, separated by year as to be comparable to the competition entrants, while others report averages across several years. These data[1] and evaluation software[2] can be downloaded from the most recent competition website.

The Persian Heritage Image Binarization Dataset (PHIBD)[3] contains 15 handwritten images of Persian writing with various degradations [6, 103] and has attracted less attention than the DIBCO data. The CMATERdb 6 dataset[4] contains 5 color images in various languages, colors, font size, and document type [97]. The Bickley Diary dataset [162] consists of 7 annotated page images of a degraded 1920s diary and has been used in a handful of later works [63, 104, 169], though it appears to no longer be publicly downloadable.

The Irish Script on Screen (ISOS) bleed-through dataset[5] contains 25 registered verso/recto image pairs that are annotated at the pixel level with 3 classes: foreground, background, and bleed-through [142]. Though intended for the related problem of bleed-through removal, such a dataset could be used to evaluate the performance of binarization algorithms on this specific degradation.

In the 2019 Time-Quality Binarization Competition [78], 3 new datasets (Nabuco, DIB, and Camera) were used for testing (104 images total), which were available for download,[6] but currently do not seem to be available.

Palm Leaf

Palm Leaf Manuscripts (PLMs) are characteristically different from the previous collections of paper documents (Fig. 4. Typical degradations for PLMs include faded ink, low contrast, stains, darkened background near edges, bleed-through, and mid-leaf binding. State-of-the-art algorithms for DIBCO datasets do not perform well on PLMs. For example, [59] reports that Howe's method achieves 42.47% F-measure, while the classic Niblack achieves 60.48% on a dataset of PLMs.

The AMADI_LontarSet[7] contains 100 pages of Balinese PLMs, split into 50 images for training and 50 images for testing [61]. Each image was annotated by two different annotators to produce 2 versions of GT. An additional 46 Khmer and 61 Sudanese PLMs were released as part of the 2018 ICFHR competition [20].[8] These images appear distinct from the Balinese PLMs.

Evaluation with Pixel Ground Truth

The primary way that binarization algorithms are currently evaluated is by comparison to reference ground truth (GT) binarizations. In "GT Construction" section, we explain

[1] https://vc.ee.duth.gr/dibco2019/.

[2] https://vc.ee.duth.gr/dibco2019/benchmark.

[3] http://www.iapr-tc11.org/mediawiki/index.php/Persian_Heritage_Image_Binarization_Dataset_(PHIBD_2012).

[4] https://code.google.com/archive/p/cmaterdb/downloads.

[5] https://www.isos.dias.ie/libraries/Sigmedia/english/index.html.

[6] https://dib.cin.ufpe.br/.

[7] http://amadi.univ-lr.fr/ICFHR2016_Contest/index.php.

[8] http://amadi.univ-lr.fr/ICFHR2018_Contest/index.php.

common ways to construct reference GT images. In "Metrics" section, we give common evaluation metrics. Criticism of this method of evaluation and alternatives are presented in "Evaluation Without Pixel Ground Truth" section.

GT Construction

The semi-automated method used to annotate the DIBCO competition images was presented in [109, 110] and is summarized here. First an initial binarization is produced using an automated method (e.g. [39]). This initial binarization is skeletonized so that components are a single pixel thick. The user then removes false alarm components and draws skeletons for missed components. Then, Canny edge detection [22] is applied to the input image, and the user corrects the edge map. Finally, the skeletonized image is iteratively dilated, constrained by the edge image, using a 3×3 cross-type structuring element. Each component is iteratively dilated until 50% of the enclosing edge pixels are covered.

To annotate the AMADI_LONTAR dataset [59, 61] of Palm Leaf Manuscripts (PLMs) for binarization, the DIBCO framework [109] was followed. Because existing binarization algorithms performed very poorly on PLMs, a new semi-local binarization scheme was proposed.

To create the Persian Heritage Image binarization Dataset (PHIBD), a simpler semi-automated method was used [103]. First, the user selects the type of image (handwriting, machine print, combined) and the types of degradations present (e.g., bleed-through). Then, a phase-based binarization method is applied with parameters chosen according to the user input. Then, the user uses the PixLabler [145] tool to manually correct the automated binarization at the pixel level.

Synthetic Data

Stathis et al. [158] and Lins et al. [76] have proposed using synthetic data, created by compositing text and background, to evaluate binarization algorithms. The advantage of using synthetic data is that a large quantity of images can be used, and the foreground annotations are more objective than human-annotated real data. However, the disadvantage is that the data do not always resemble real-world data.

In [158], foreground is extracted from PDFs and is composited on authentic noisy background by taking the element-wise maximum or by average blending. The latter technique leads to a more natural looking result, but results in a lighter background due to the PDF background being white. This method was also used to construct the test set for the ICFHR 2010 Quantitative Evaluation of Binarization Algorithms [116].

Lins et al. [76] create synthetic backgrounds by first clustering background patches from 3351 documents to find 100 distinct background textures. Then, novel backgrounds can be created through random sampling from a particular texture or through image quilting techniques for texture generation. Then foreground text and artificial bleed-through are composited on the generated background through traditional image compositing. They use a parameter, α, to control the strength of the bleed-through and evaluate algorithms at different levels of bleed-through degradation.

Though not specific to synthetic data for binarization, the DocCreator library [54] contains degradation models for introducing ink fading, adaptive blur, bleed-through, and uneven illumination. Seuret et al. [149] proposed inserting stains and spots using gradient domain image editing.

Metrics

Many metrics have been used to evaluate binarizations using pixel GT.

Standard Binarization Metrics

F-measure (FM) is the harmonic mean of precision (P) and recall (R), which in turn are defined by the number of true positives (TPs), false positives (FPs), and false negatives (FNs).

$$\text{FM} = \frac{2PR}{P+R} \qquad P = \frac{\text{TP}}{\text{TP}+\text{FP}} \qquad R = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{27}$$

$$\begin{aligned} \text{TP} &= \sum_{ij} B(i,j) \wedge G(i,j) \quad \text{FP} = \sum_{ij} B(i,j) \wedge \neg G(i,j) \\ \text{FN} &= \sum_{ij} \neg B(i,j) \wedge G(i,j) \end{aligned} \tag{28}$$

where B is the predicted binarization, G is the GT, foreground values are 1 (logically true), background values are 0 (logically false), and the above boolean operations evaluate to 1 for true and 0 for false. FM has been used as part of all DIBCO competitions [40, 112, 124–130], as well as the PLM competition [19]. FM is used because there are often far more background pixels than foreground pixels, and it penalizes methods that make a disproportionate number of FP or FN errors.

Another metric used in all DIBCO competitions is the peak signal-to-noise ratio (PSNR), which is computed as

$$\begin{aligned} \text{PSNR} &= 10 \log \left(\frac{1}{\text{MSE}} \right) \\ \text{MSE} &= \frac{1}{\text{HW}} \sum_{i=1}^{H} \sum_{j=1}^{W} (B(i,j) - G(i,j))^2 \end{aligned} \tag{29}$$

Note that MSE $= \frac{\text{FP+FN}}{\text{HW}}$ is simply the percentage of errors made, which makes the PSNR metric monotonically increasing w.r.t. pixel-wise accuracy. Therefore, PSNR and accuracy always produce identical algorithm rankings.

Distance reciprocal distortion (DRD) has been used in DIBCO 2011–2018 and has been shown to correlate well with human perception of how bad binarization errors are [81]. It is computed by

$$\text{DRD} = \frac{1}{\text{NUBN}(G)} \sum_{ij} \text{DRD}_{ij} |B(i,j) - G(i,j)| \tag{30}$$

$$\text{DRD}_{ij} = \sum_{x=-2}^{2} \sum_{y=-2}^{2} W_{xy} |B(i+x, j+y) - G(i+x, j+y)| \tag{31}$$

where NUBN(G) is the number of non-uniform 8×8 binary patches of G, and W is defined by $W_{00} = 0$, $\forall x \neq 0,\ y \neq 0$, $W_{xy} \propto \frac{1}{\sqrt{(x^2+y^2)}}$, and $\sum_{xy} W_{xy} = 1$. Lower DRD scores are better. From Eq. 30, we see that all wrongly predicted pixels contribute an error penalty according to Eq. 31. Eq. 31 examines the 5×5 window around the wrong pixel and assigns a penalty for every other wrong pixel in this window (but 0 penalty for itself). Thus, if we have two predicted binarizations with an equal number of errors, DRD will assign the higher (worse) score to the prediction that has its wrong predictions close together. Single isolated errors are not as noticeable to the human eye as concentrated groups of wrong pixels [81].

Pseudo-F-Measure

Ntirogiannis et al. [110] proposed a variant of FM, called pseudo-F-Measure (pFM), which is the harmonic mean of pseudo-precision P_{ps} and pseudo-recall R_{ps}. While Precision and Recall ignore the spatial locations of errors, P_{ps} and R_{ps} assign weights to each pixel based on the GT. These weights can be interpreted as how bad an error is at that location.

$$\begin{aligned} \text{pFM} &= \frac{2P_{\text{ps}}R_{ps}}{P_{\text{ps}} + R_{ps}} \quad P_{\text{ps}} = \frac{\sum_{ij} B(i,j)G(i,j)W_P(i,j)}{B(i,j)W_P(i,j)} \\ R_{\text{ps}} &= \frac{\sum_{ij} B(i,j)G(i,j)W_R(i,j)}{G(i,j)W_R(i,j)} \end{aligned} \tag{32}$$

where setting W_P and W_R to be uniform and nonzero reduces to the traditional FM as in Eq. 27. W_P assigns a greater penalty to FPs located near foreground regions. The rationale is that FP errors far from the foreground do not impede the readability of the text. FP pixels very close to the foreground–background boundary have lower weight due to the inherit ambiguity of localizing this boundary. Regions between foreground components also have high weight in order to penalize the joining of adjacent characters. W_R assigns the highest per-pixel error for FNs located on the skeleton of the foreground and on thinner strokes. Thus, binarizations that fragment characters yield a higher error than binarizations that predict thinner, but unfragmented strokes. Foreground pixels on the border are assigned a recall weight of 0, in order to account for single-pixel localization ambiguity of the boundary in the GT.

Ntirogiannis et al. [110] further compute a weighted measure of FN errors into the categories fully missed text, broken text, and partially missed text. Earlier similar definitions (uniform weights) were given in [109]. Fully missed text concerns the foreground components that are completely missed. Broken text errors occur when FN errors break a GT foreground component into 2 or more components. Partially missed text incorporates the rest of the errors, where a component is partially detected and not fragmented.

Likewise, the FP errors are categorized into false alarms, background noise, character enlargement, and character merging. False alarms are predicted components that do not overlap with any GT foreground component. Character enlargement concerns FPs immediately around components that do not cause merging, while character merging errors are those pixels that cause merging of foreground components. Background noise errors are similar to false alarms, but are those pixels that are (1) far away from GT components and are (2) part of a predicted component that overlaps a GT component.

While such a detailed breakdown of errors is certainly useful for comparative analysis of algorithms, these or similar numerical measures are rarely reported in the literature. This may be because there is no public code available for evaluating this error breakdown like there is for pFM, FM, PSNR, and DRD.

In an unfortunate naming collision, a different metric also called pseudo-F-measure [109] was previously used in the DIBCO 2010 and 2012. The newer pseudo-F-measure [110] is now preferred and has been used in DIBCO 2013–2018. The old pseudo-F-measure [109] adapts the traditional FM by computing recall with only the GT skeleton. In contrast, the new pseudo-F-measure [110] evaluates recall with all non-border foreground pixels, increasing the penalty for errors based on distance to the GT skeleton.

Evaluation Without Pixel Ground Truth

Though most recently proposed methods perform evaluation using GT ("Evaluation with Pixel Ground Truth" section), there has been some discussion on whether this is a useful form of evaluation. Given the problems with defining objective GT at the pixel level ("Problems with Pixel GT"

section), some alternative evaluation methods have been proposed that use blind metrics ("Blind Metrics" section), including consensus-based metrics ("Consensus Methods" section) and end-to-end systems ("End-to-End Metrics" section).

Problems with Pixel GT

While the DIBCO framework for ground truth creation allows for efficient user interaction, the heavy use of automation can introduce bias toward the algorithms used for automation. For example, Howe [49] noted that the disproportionate number of errors observed on the north–west side of CCs could be an artifact of the GT construction process. Though [103] uses less automation, there still exists the possibility of bias toward the algorithm used in the initial binarization.

For the ICHFR 2016 PLM competition [19] over the AMADI_LONTAR dataset [61], two sets of binarization GT were released, created by different annotators. The agreement F-measure between the two annotators is 62.40%. In [60], an analysis of annotator variability is done under the scenario where the annotators manually draw all character skeletons, i.e., no initial automatic binarization is performed. They report an average agreement of 59.56% FM, which is lower than the top two binarization methods in the competition [19] (68% and 63% FM).

Smith [154] conducted a study where 6 students were asked to annotate a single DIBCO image using the PixLabler [145] tool. The average pairwise FM for these 6 *ground truths* was 84.9%. Similarly, a single student produced annotations for all 14 DIBCO 2009 images [40] and achieved an average FM score of 89.3% w.r.t the published GT. This is lower than the 91.24% FM achieved by the competition winner [40] and lower than the 94.68% reported by Howe [50].

In a separate study, Smith and An [155] experimented with bias versions of GT (eroded, original, dilated) for training and evaluating a learning-based binarization method. Unsurprisingly, the highest evaluation scores were achieved when the training GT bias matched the evaluation GT bias. However, this highlights the problem of ranking algorithms based on a single inherently biased GT.

Blind Metrics

Blind metrics evaluate performance using no external GT information at test time. Though any such metric can be turned into a binarization algorithm by performing optimization with the blind metric as the objective function [135], they are still useful measures, particularly for parameter tuning.

In [88], GT is constructed automatically and used to choose binarization methods and tune parameters. Stommel and Frieder [159] trained an SVM classifier to predict the global legibility of binary images using a labeled training set of binary images. Afterward, they pick a global threshold that optimizes the predicted legibility of a test image. Ramírez-Ortegón et al. [135] proposed a statistical measure, based on the assumption that foreground and background regions are log-normally distributed, that was shown to correlate well with OCR accuracy.

Shaus et al. [152] proposed a number of blind metrics, many of which are adapted from older global thresholding techniques (e.g., Otsu [114]), based on the separation of foreground and background clusters of pixels. However, turning a binarization algorithm into a metric leads to that binarization algorithm producing the optimal result for that metric, regardless of how well it actually correlated with human perception of binarization performance. For example, one proposed metric is the PSNR of the binary image w.r.t. the input grayscale image. While intuitively lighter pixels should be assigned to background and vice versa, this metric is trivially optimized by choosing a global threshold of $T = 128$ for 8-bit grayscale images.

Consensus Methods

Fedorchuk and Lamiroy [35] proposed a method for evaluating binarizations based on how the predicted binarization agrees with the output of other algorithms in the absence of ground truth. Essentially, the binary GT image is replaced with a probabilistic image, where the probabilities are the percentage of algorithms that consider that pixel as foreground. They found that the probabilistic versions of FM and PSNR had an average correlation coefficient of 85% with the traditional non-probabilistic FM and PSNR measures on the DIBCO datasets. However, the top performing algorithms on the DIBCO data achieve an FM greater than 90%, so these proposed consensus measures could not ascertain if a new algorithm performs better than the current state of the art.

In a similar vein, paired ranking based on agreement with a third, better than random, model has been proposed [34]. Such a scheme works if the third model makes unbiased, random errors, but this assumption would never be satisfied in practice. Different choices of a third model would yield different biases and hence different rankings.

Overall, consensus-based evaluation has similar disadvantages as other blind evaluations, specifically bias introduced by the choice of models to compare with. The majority vote of the algorithms used for consensus trivially optimizes the proposed consensus metrics.

End-to-End Metrics

Most often binarization algorithms are intended to be used as a preprocessing component in a larger system that also

performs other tasks, such as layout analysis or text recognition. Thus, it is logical to compare binarization algorithms based on how they influence the performance of the whole system. Before annotated images were introduced in 2009 by the DIBCO competition, this was the primary way to numerically compare binarization algorithms using real data.

As such, many works also evaluate the quality of their binarization method by measuring the character and word error rates of a third-party OCR engine (e.g., ABBYY[9]) run over their binary images [23, 39, 64, 72, 94, 105, 132, 166]. While this introduces the bias of the OCR engine (it may be more forgiving to some types of errors), one may argue that, in the context of real-world applications, end-to-end performance is more important than intermediate binarization quality.

However, the comparative ranking of algorithms depends on the choice of end-to-end system. To reduce system bias, perhaps evaluation should be done with multiple state-of-the-art end-to-end systems, where binarization method ranking is based on the maximum performance achieved with any systems. Though such evaluation would be taxing on academics that have little resources to implement full systems, recent work in building DIA systems as web services [175] may facilitate such an approach.

We also note that there are no standard datasets designed for evaluating binarization algorithms using OCR performance. The DIBCO data are not annotated with transcriptions and is multi-lingual. All papers surveyed in this review that used OCR to evaluate binarization quality used a distinct set of images (some synthetic), which hinders comparative evaluation. If this method of evaluation is to be used, there needs to be a dataset designed for this purpose.

Discussion and Conclusion

In this work, we have reviewed the recent advances in the field of historical document binarization. Based on our analysis, we provide some discussion on the broad trends, the current technical challenges, and directions for future work.

Trends

For many years, methods based on image processing (e.g. [162]) and CRFs (e.g. [50]) produced state-of-the-art results. Recently, pixel classification approaches using deep learning models have come to outperform other algorithms. However, it should be noted that although deep models were the top performers at DIBCO17 [129], the top 2 submissions to both HDIBCO18 [130] and DIBCO19 [131] did not use deep models.

Another powerful trend is to perform automatic parameter tuning on previously successful algorithms. Many algorithms have a sufficiently rich parameter space that with properly set parameters (global or local) can produce a good binarization for nearly all images. Previous algorithms are also successfully combined with more powerful pre- and post-processing steps. For example, the winner of HDIBCO18 [130] uses Howe binarization [50] with morphological preprocessing and connectivity-preserving post-processing.

There is also a small trend to use less common types of historical documents including Palm Leaf Manuscripts (PLM) [19, 20] and papyrus fragments [131]. There is also emerging work in the domain of multi-spectral document binarization [32].

Technical Challenges

Though deep models are accurate, they are slow and require GPU hardware to obtain reasonable speed [77]. This makes many applications of binarization difficult such as binarization on mobile devices or binarization of large archives. There is also limited labeled training data for deep models. Though some initial explorations around this issue have been made [14, 55, 68, 165], there is still a large gap to address.

There continues to be debate on the proper way to evaluate binarization algorithms, given that humans do not always agree at the pixel level. However, the majority of published research continues to evaluate at the pixel level using the DIBCO provided annotations and metrics. For the first time, processing speed has been considered as a formal metric for algorithm comparison [77, 78].

The general lack of algorithm reproducibility slows down the field as individual researchers reinvent the wheel. It is not always clear which parts (pre-/post-processing, parameter tuning, local threshold selection, etc.) of previous algorithms were successful and should be re-used and which are sub-optimal. Sometimes, algorithm descriptions do not contain implementation-level detail, and few authors have made their code publicly available.

Generalizing algorithms across domains of documents also remains challenging. For example, the top method of DIBCO19 [131] produced very few false positives due to background artifacts (other than bleed-through) on the paper documents, but produced very noisy binarizations for the papyrus documents. A similar thing happened when taking an algorithm developed for paper documents and applying it to PLMs [19].

While binarization has come to be reliable for images with little noise, it is still challenging to find an algorithm that works well across all types of noise. Thresholding

[9] https://www.abbyy.com/en-us/finereader/.

methods tend to break when there are large stains, and few methods are regularly robust to bleed-through text, which remains one of the most challenging types of noise. HDIBCO18 [130] introduced border noise (e.g., book edges), which many submitted methods were not prepared to handle.

Directions for Future Work

Future work in the area might explore how to integrate some of the best practices from image processing techniques with deep learning models or vice versa, for example incorporating preprocessing and thresholding into the learning model or using deep models to replace many of the parameters or fixed constants in traditional methods. A large dataset suitable for data-hungry deep models with high-quality, low-bias pixel annotations also remains elusive, though GAN-based methods seem to help alleviate this data need [14].

While evaluation using the pseudo-F-measure metric [110] partly addresses some concerns with using pixel GT, less biased evaluation measures are needed. One possible avenue forward is to create multiple reproducible end-to-end document processing pipelines (binarization, layout analysis, text recognition, retrieval, writer ID, etc.) for different tasks and domains. To prevent bias toward particular downstream implementations, multiple implementations could be tried and performance recorded for the best combination of downstream implementations. Then, binarization algorithms could be comparatively evaluated in multiple real contexts and domains and give a more comprehensive view of algorithm performance. Such a platform could also contain reference implementations of common components, e.g., edge thresholding [82] and background removal [39].

While earlier work tried to specifically address different types of noise [87], more recent algorithms have made no such distinction. Learning-based models typically classify pixels into either foreground or background, but there could be improvement if they also learned to explicitly identify different types of noise (e.g., bleed-through with the ISOS dataset [142]).

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Afzal MZ, Pastor-Pellicer J, Shafait F, Breuel TM, Dengel A, Liwicki M. Document image binarization using LSTM: a sequence learning approach. In: Workshop on historical document imaging and processing (HIP). ACM; 2015. p. 79–84.
2. Ahmadi E, Azimifar Z, Shams M, Famouri M, Shafiee MJ. Document image binarization using a discriminative structural classifier. Pattern Recognit Lett. 2015;63:36–42.
3. Akbari Y, Britto AS, Al-maadeed S, Oliveira LS. Binarization of degraded document images using convolutional neural networks based on predicted two-channel images. In: International conference on document analysis and recognition (ICDAR). IEEE; 2019. p. 973–8
4. Almeida M, Lins RD, Bernardino R, Jesus D, Lima B. A new binarization algorithm for historical documents. J Imaging. 2018;4(2):27.
5. Arruda A, Mello CA. Binarization of degraded document images based on combination of contrast images. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2014. p. 615–20.
6. Ayatollahi SM, Nafchi HZ. Persian heritage image binarization competition (PHIBC 2012). In: Iranian conference on pattern recognition and image analysis (PRIA). IEEE; 2013. p. 1–4.
7. Ayyalasomayajula KR, Brun A. Document binarization using topological clustering guided Laplacian energy segmentation. In: International conference on frontiers in handwriting recognition (ICFHR); 2014. p. 523–8.
8. Ayyalasomayajula KR, Malmberg F, Brun A. PDNET: semantic segmentation integrated with a primal-dual network for document binarization. Pattern Recognit Lett. 2018;121:52–60.
9. Bag S, Bhowmick P. Adaptive-interpolative binarization with stroke preservation for restoration of faint characters in degraded documents. J Vis Commun Image Represent. 2015;31:266–81.
10. Bataineh B, Abdullah SNHS, Omar K. An adaptive local binarization method for document images based on a novel thresholding method and dynamic windows. Pattern Recognit Lett. 2011;32(14):1805–13.
11. Bataineh B, Abdullah SNHS, Omar K. Adaptive binarization method for degraded document images based on surface contrast variation. Pattern Anal Appl. 2017;20(3):639–52.
12. Bawa RK, Sethi GK. A review on binarization algorithms for camera based natural scene images. In: International conference on advances in computing, communications and informatics. ACM; 2012. p. 873–78.
13. Bhowmik S, Sarkar R, Das B, Doermann D. Gib: a game theory inspired binarization technique for degraded document images. IEEE Trans Image Process. 2018;28(3):1443–55.
14. Bhunia AK, Bhunia AK, Sain A, Roy PP. Improving document binarization via adversarial noise-texture augmentation. In: IEEE international conference on image processing (ICIP). IEEE; 2019. p. 2721–25.
15. Boiangiu CA, Olteanu A, Stefanescu A, Rosner D, Tapus N, Andreica MI. Local thresholding algorithm based on variable window size statistics. In: International conference on control systems and computer science (CSCS); 2011, vol 2. p. 647–52.
16. Bolan S, Shijian L, Tan CL. A self-training learning document binarization framework. In: International conference on pattern recognition (ICPR). IEEE; 2010. p. 3187–90.
17. Bouillon M, Ingold R, Liwicki M. Grayification: a meaningful grayscale conversion to improve handwritten historical documents analysis. Pattern Recognit Lett. 2018;121:4–651.
18. Boykov Y, Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. Trans Pattern Anal Mach Intell. 2004;26(9):1124–37.
19. Burie JC, Coustaty M, Hadi S, Kesiman MWA, Ogier JM, Paulus E, Sok K, Sunarya IMG, Valy D. ICFHR2016 Competition on the analysis of handwritten text in images of Balinese palm leaf manuscripts. In: International conference on front handwrit recognit (ICFHR) ;2016. p. 596–601.

20. Burie JC, Kesiman MWA, Valy D, Paulus E, Suryani M, Hadi S, Verleysen M, Chhun S, Ogier JM. ICFHR2018 competition on document image analysis tasks for southeast Asian palm leaf manuscripts. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2018.
21. Calvo-Zaragoza J, Gallego AJ. A selectional auto-encoder approach for document image binarization. Pattern Recognit. 2019;86:37–47.
22. Canny J. A computational approach to edge detection. Trans Pattern Anal Mach Intell. 1986;8:679–98.
23. Chattopadhyay T, Reddy VR, Garain U. Automatic selection of binarization method for robust OCR. In: International conference on document analysis and recognition (ICDAR). IEEE; 2013. p. 1170–4.
24. Chaudhary P, Saini B. An effective and robust technique for the binarization of degraded document images. Int J Res Eng Technol (IJRET). 2014;3(06):140.
25. Chen X, Lin L, Gao Y. Parallel nonparametric binarization for degraded document images. Neurocomputing. 2016;189:43–52.
26. Chen Y, Wang L. Broken and degraded document images binarization. Neurocomputing. 2017;237:272–80.
27. Cheriet M, Moghaddam RF, Hedjam R. A learning framework for the optimization and automation of document binarization methods. Comput Vis Image Understand. 2013;117(3):269–80.
28. Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B, Shelhamer E. CUDNN: Efficient primitives for deep learning. 2014 arXiv preprint arXiv:14100759.
29. Chiu YH, Chung KL, Yang WN, Huang YH, Liao CH. Parameter-free based two-stage method for binarizing degraded document images. Pattern Recognit. 2012;45(12):4250–62.
30. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Computer vision and pattern recognition (CVPR). IEEE; 2005, vol 1. p. 886–93.
31. Deng F, Wu Z, Lu Z, Brown MS. Binarizationshop: a user-assisted software suite for converting old documents to black-and-white. In: Joint conference on digital libraries. ACM; 2010. p. 255–8.
32. Diem M, Hollaus F, Sablatnig R, Msio: Multispectral document image binarization. In, 12th IAPR workshop on document analysis systems (DAS). IEEE; 2016. p. 84–9.
33. Elfattah MA, Hassanien AE, Abuelenin S, Bhattacharyya S. Multi-verse optimization clustering algorithm for binarization of handwritten documents. In: Recent trends in signal and image processing. Springer; 2019. p. 165–75.
34. Fedorchuk M, Lamiroy B. Binary classifier evaluation without ground truth. In: International conference on advances in pattern recognition (ICAPR) 2017.
35. Fedorchuk M, Lamiroy B. Statistic metrics for evaluation of binary classifiers without ground-truth. In: Ukraine conference on electrical and computer engineering (UKRCON). IEEE; 2017. p. 1066–71.
36. Felix R, da Silva LA, de Castro LN. Thresholding the courtesy amount of brazilian bank checks using a local methodology. In: International conference on practical applications of agents and multi-agent systems. Springer; 2015. p. 213–21.
37. Fernando B, Karaoglu S, Trémeau A. Extreme value theory based text binarization in documents and natural scenes. In: International conference on machine vision (ICMV); 2010. p. 144–51.
38. Gatos B, Pratikakis I, Perantonis SJ. An adaptive binarization technique for low quality historical documents. In: International workshop on document analysis systems (DAS). Springer; 2004. p. 102–13.
39. Gatos B, Pratikakis I, Perantonis SJ. Adaptive degraded document image binarization. Pattern Recognit. 2006;39(3):317–27.
40. Gatos B, Ntirogiannis K, Pratikakis I. Icdar 2009 document image binarization contest (dibco 2009). In: International conference on document analysis and recognition (ICDAR). IEEE; 2009. p. 1375–82.
41. Gatos B, Ntirogiannis K, Pratikakis I. Dibco 2009: Document image binarization contest. Int J Doc Anal Recognit (IJDAR). 2011;14(1):35–44.
42. Hadjadj Z, Meziane A, Cherfa Y, Cheriet M, Setitra I. Isauvola: Improved sauvola's algorithm for document image binarization. In: Image analysis and recognition. Springer; 2016. p. 737–45.
43. Hadjadj Z, Cheriet M, Meziane A, Cherfa Y. A new efficient binarization method: Application to degraded historical document images. Signal Image Video Process. 2017;11(6):1155–62.
44. Hamza H, Smigiel E, Belaid E. Neural based binarization techniques. In: International conference on document analysis and recognition (ICDAR). IEEE; 2005. p. 317–21.
45. Hassanien AE, Elfattah MA, Aboulenin S, Schaefer G, Zhu SY, Korovin I. Historic handwritten manuscript binarisation using whale optimisation. In: International conference on systems, man, and cybernetics (SMC). IEEE; 2016. p. 3842–46.
46. Hebert D, Nicolas S, Paquet T. Discrete crf based combination framework for document image binarization. In: International conference on document analysis and recognition (ICDAR). IEEE; 2013. p. 1165–69.
47. Hedjam R, Moghaddam RF, Cheriet M. A spatially adaptive statistical method for the binarization of historical manuscripts and degraded document images. Pattern Recognit. 2011;44(9):2184–96.
48. Hedjam R, Nafchi HZ, Kalacska M, Cheriet M. Influence of color-to-gray conversion on the performance of document image binarization: toward a novel optimization problem. Trans Image Process. 2015;24(11):3637–51.
49. Howe NR. A Laplacian energy for document binarization. In: International conference on document analysis and recognition (ICDAR). IEEE; 2011. p. 6–10.
50. Howe NR. Document binarization with automatic parameter tuning. Int J Doc Anal Recognit (IJDAR). 2013;16(3):247–58.
51. Ismail SM, Abdullah SNHS, Fauzi F. Statistical binarization techniques for document image analysis. J Comput Sci. 2018;14(1):23–36.
52. Jennifer Ranjani J. Bi-level thresholding for binarisation of handwritten and printed documents. IET Comput Vis. 2015;10.
53. Jia F, Shi C, He K, Wang C, Xiao B. Degraded document image binarization using structural symmetry of strokes. Pattern Recognit. 2018;74:225–40.
54. Journet N, Visani M, Mansencal B, Van-Cuong K, Billy A. Doccreator: a new software for creating synthetic ground-truthed document images. J Imaging. 2017;3(4):62.
55. Kang S, Iwana BK, Uchida S. Cascading modular u-nets for document image binarization. In: International conference on document analysis and recognition (ICDAR). IEEE; 2019. p. 675–80.
56. Kasmin F, Abdullah A, Prabuwono AS. Ensemble of steerable local neighbourhood grey-level information for binarization. Pattern Recognit Lett. 2017;98:8–15.
57. Kefali A, Sari T, Sellami M. Evaluation of several binarization techniques for old Arabic documents images. International symposium on modeling and implementing complex systems; 2010 vol 1. p. 88–99.
58. Kefali A, Sari T, Bahi H. Foreground-background separation by feed-forward neural networks in old manuscripts. Informatica 2014;38(4).
59. Kesiman MWA, Prum S, Burie JC, Ogier JM. An initial study on the construction of ground truth binarized images of ancient palm leaf manuscripts. In: International conference on document analysis and recognition (ICDAR). IEEE; 2015. p. 656–60.

60. Kesiman MWA, Prum S, Sunarya IMG, Burie JC, Ogier JM. An analysis of ground truth binarized image variability of palm leaf manuscripts. In: International conference on image processing, theory, tools and applications (IPTA) 2015.
61. Kesiman MWA, Burie JC, Wibawantara GNMA, Sunarya IMG, Ogier JM. Amadi_lontarset: the first handwritten Balinese palm leaf manuscripts dataset. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2016. p. 168–73.
62. Khankasikam K. Restoration of degraded historical document image: An adaptive multilayer-information binarization technique. J Inf Sci Eng. 2014;30(5):1321–38.
63. Khitas M, Ziet L, Bouguezel S. Improved degraded document image binarization using median filter for background estimation. Elektronika ir Elektrotechnika. 2018;24(3):82–7.
64. Khurshid K, Siddiqi I, Faure C, Vincent N. Comparison of niblack inspired binarization methods for ancient documents. In: Document recognition and retrieval XVI. International Society for Optics and Photonics; 2009. p. 7247:72470U.
65. Kligler N, Katz S, Tal A. Document enhancement using visibility detection. In: Computer vision and pattern recognition (CVPR), 2018;2374–2382.
66. Kolmogorov V, Zabih R. What energy functions can be minimized via graph cuts? Trans Pattern Anal Mach Intell. 2004;26:147–59.
67. Krähenbühl P, Koltun V. Efficient inference in fully connected crfs with Gaussian edge potentials. In: Advances in neural information processing systems (NIPS), 2011;109–117.
68. Krantz A, Westphal F. Cluster-based sample selection for document image binarization. In: 2019 International conference on document analysis and recognition workshops (ICDARW). IEEE; 2019. p. 5:47–52.
69. Kuk JG, Cho NI. Feature based binarization of document images degraded by uneven light condition. In: International conference on document analysis and recognition (ICDAR). IEEE; 2009. p. 748–52.
70. Le THN, Bui TD, Suen CY. Ternary entropy-based binarization of degraded document images using morphological operators. In: International conference on document analysis and recognition (ICDAR). IEEE; 2011. p. 114–8.
71. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE. 1998;86(11):2278–324.
72. Lelore T, Bouchara F. Document image binarisation using Markov field model. In: International conference on document analysis and recognition (ICDAR). IEEE; 2009. p. 551–5.
73. Lelore T, Bouchara F. Fair: a fast algorithm for document image restoration. Trans Pattern Anal Mach Intell. 2013;35(8):2039–48.
74. Liang Y, Lin Z, Sun L, Cao J. Document image binarization via optimized hybrid thresholding. In: International symposium on circuits and systems (ISCAS). IEEE; 2017. p. 1–4.
75. Likforman-Sulem L, Darbon J, Smith EHB. Enhancement of historical printed document images by combining total variation regularization and non-local means filtering. Image Vision Comput. 2011;29(5):351–63.
76. Lins RD, de Almeida MM, Bernardino RB, Jesus D, Oliveira JM. Assessing binarization techniques for document images. In: Symposium on document engineering, ACM, 2017;183–192.
77. Lins RD, Bernardino R, Jesus DM. A quality and time assessment of binarization algorithms. In: International conference on document analysis and recognition (ICDAR). IEEE. 2019. p. 1444–50.
78. Lins RD, Kavallieratou E, Smith EB, Bernardino RB, de Jesus DM. Icdar 2019 time-quality binarization competition. In: 2019 International conference on document analysis and recognition (ICDAR). IEEE; 2019. p. 1539–46.
79. Liu N, Zhang D, Xu X, Liu W, Ke D, Guo L, Shi S, Liu H, Chen L. An iterative refinement framework for image document binarization with Bhattacharyya similarity measure. In: International conference on document analysis and recognition (ICDAR). IEEE; 2017. p. 1:93–98.
80. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Computer vision and pattern recognition (CVPR), 2015;3431–40.
81. Lu H, Kot AC, Shi YQ. Distance-reciprocal distortion measure for binary document images. Signal Process Lett. 2004;11(2):228–31.
82. Lu S, Su B, Tan CL. Document image binarization using background estimation and stroke edges. Int J Doc Anal Recognit (IJDAR). 2010;13(4):303–14.
83. Lu Z, Wu Z, Brown MS. Directed assistance for ink-bleed reduction in old documents. In: Computer vision and pattern recognition (CVPR). IEEE; 2009. p. 88–95.
84. Lu Z, Wu Z, Brown MS. Interactive degraded document binarization: An example (and case) for interactive computer vision. In: Workshop on applications of computer vision (WACV). IEEE; 2009. p. 1–8.
85. Mesquita RG, Mello CA, Almeida L. A new thresholding algorithm for document images based on the perception of objects by distance. Integr Comput Aided Eng. 2014;21(2):133–46.
86. Mesquita RG, Silva RM, Mello CA, Miranda PB. Parameter tuning for document image binarization using a racing algorithm. Expert Syst Appl. 2015;42(5):2593–603.
87. Messaoud IB, El Abed H, Amiri H, Märgner V. New method for the selection of binarization parameters based on noise features of historical documents. In: Joint workshop on multilingual OCR and analytics for noisy unstructured text data. ACM; 2011. p. 1.
88. Messaoud IB, El Abed H, Märgner V, Amiri H. A design of a preprocessing framework for large database of historical documents. In: Workshop on historical document imaging and processing (HIP). ACM; 2011. p. 177–83.
89. Messaoud IB, Amiri H, El Abed H, Margner V. Document preprocessing system-automatic selection of binarization. In: International workshop on document analysis systems (DAS). IEEE; 2012. p. 85–9.
90. Messaoud IB, Amiri H, El Abed H, Märgner V. Region based local binarization approach for handwritten ancient documents. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2012. p. 633–8.
91. Mishra A, Alahari K, Jawahar C. Unsupervised refinement of color and stroke features for text binarization. Int J Doc Anal Recognit (IJDAR). 2017;20(2):105–21.
92. Mitianoudis N, Papamarkos N. Local co-occurrence and contrast mapping for document image binarization. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2014. p. 609–14.
93. Mitianoudis N, Papamarkos N. Document image binarization using local features and gaussian mixture modeling. Image Vis Comput. 2015;38:33–51.
94. Moghaddam RF, Cheriet M. A multi-scale framework for adaptive binarization of degraded document images. Pattern Recognit. 2010;43(6):2186–98.
95. Moghaddam RF, Cheriet M. Adotsu: an adaptive and parameterless generalization of otsu's method for document image binarization. Pattern Recognit. 2012;45(6):2419–31.
96. Moghaddam RF, Moghaddam FF, Cheriet M. Unsupervised ensemble of experts (eoe) framework for automatic binarization of document images. In: International conference on document analysis and recognition (ICDAR). IEEE; 2013. p. 703–7.

97. Mollah AF, Basu S, Nasipuri M. Computationally efficient implementation of convolution-based locally adaptive binarization techniques. In: Wireless networks and computational intelligence. Springer; 2012, p. 159–168.
98. Mondal R, Chakraborty D, Chanda B. Learning 2D morphological network for old document image binarization. In: International conference on document analysis and recognition (ICDAR). IEEE; 2019. p. 65–70.
99. More PK, Dighe D. A review on document image binarization technique for degraded document images. Int Res J Eng Technol. 2016;3:1132–8.
100. Mustafa WA, Aziz H, Khairunizam W, Ibrahim Z, Shahriman A, Razlan ZM. Review of different binarization approaches on degraded document images. In: International conference on computational approach in smart systems design and applications (ICASSDA). IEEE; 2018. p. 1–8.
101. Nafchi HZ, Kanan HR. A phase congruency based document binarization. In: International conference on image and signal processing. Springer; 2012. p. 113–21.
102. Nafchi HZ, Moghaddam RF, Cheriet M. Historical document binarization based on phase information of images. In: Asian conference on computer vision (ACCV). Springer; 2012. p. 1–12.
103. Nafchi HZ, Ayatollahi SM, Moghaddam RF, Cheriet M. An efficient ground truthing tool for binarization of historical manuscripts. In: International conference on document analysis and recognition (ICDAR). IEEE; 2013. p. 807–11.
104. Nafchi HZ, Moghaddam RF, Cheriet M. Phase-based binarization of ancient document images: Model and applications. Trans Image Process. 2014;23(7):2916–30.
105. Natarajan J, Sreedevi I. Enhancement of ancient manuscript images by log based binarization technique. AEU Int J Electron Commun. 2017;75:15–22.
106. Niblack W. An introduction to digital image processing. Birkerod: Strandberg Publishing Company; 1985.
107. Nicolaou A, Liwicki M, Ingolf R. Investigating the power of integral histograms for document images, a binarization case study. 2013. http://nicolaou.homouniversalis.org/assets/lof/nicolaou2013binarization_lowres.pdf. Accessed 15 Jan 2018.
108. Nina O, Morse B, Barrett W. A recursive otsu thresholding method for scanned document binarization. In: Workshop on applications of computer vision (WACV). IEEE; 2011. p. 307–14.
109. Ntirogiannis K, Gatos B, Pratikakis I. An objective evaluation methodology for document image binarization techniques. In: International workshop on document analysis systems (DAS). IEEE; 2008. p. 217–24.
110. Ntirogiannis K, Gatos B, Pratikakis I. Performance evaluation methodology for historical document image binarization. Trans Image Process. 2013;22(2):595–609.
111. Ntirogiannis K, Gatos B, Pratikakis I. A combined approach for the binarization of handwritten document images. Pattern Recognit Lett. 2014;35:3–15.
112. Ntirogiannis K, Gatos B, Pratikakis I. ICFHR2014 competition on handwritten document image binarization (H-DIBCO 2014). In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2014. p. 809–13.
113. Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Trans Pattern Anal Mach Intell. 2002;24(7):971–87.
114. Otsu N. A threshold selection method from gray-level histograms. Trans Syst Man Cybern. 1979;9(1):62–6.
115. Ouafek N, Kholladi MK. A binarization method for degraded document image using artificial neural network and interpolation inpainting. In: International conference on optimization and applications (ICOA). IEEE; 2018. p. 1–5.
116. Paredes R, Kavallieratou E, Lins RD. ICFHR 2010 contest: quantitative evaluation of binarization algorithms. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2010. p. 733–6.
117. Parker J, Frieder O, Frieder G. Robust binarization of degraded document images using heuristics. In: Document recognition and retrieval XXI. International Society for Optics and Photonics; 2014. p. 9021:90210U.
118. Pastor-Pellicer J, Zamora-Martínez F, Espa na-Boquera S, Castro-Bleda MJ. F-measure as the error function to train neural networks. In: International work-conference on artificial neural networks. Springer; 2013. p. 376–84.
119. Pastor-Pellicer J, Espa na-Boquera S, Zamora-Martínez F, Afzal MZ, Castro-Bleda MJ. Insights on the use of convolutional neural networks for document image binarization. In: International work-conference on artificial neural networks. Springer; 2015. p. 115–26.
120. Peng X, Setlur S, Govindaraju V, Sitaram R. Markov random field based binarization for hand-held devices captured document images. In: Indian conference on computer vision, graphics and image processing. ACM; 2010. p. 71–6.
121. Peng X, Cao H, Natarajan P. Using convolutional encoder-decoder for document image binarization. In: International conference on document analysis and recognition (ICDAR). IEEE; 2017. p. 1:708–13.
122. Peng X, Wang C, Cao H. Document binarization via multi-resolutional attention model with DRD loss. In: International conference on document analysis and recognition (ICDAR). IEEE; 2019. p. 45–50.
123. Perret B, Lefèvre S, Collet C, Slezak E. From hyperconnections to hypercomponent tree: application to document image binarization. In: WADGMM 2010; 2010. p. 62.
124. Pratikakis I, Gatos B, Ntirogiannis K. H-DIBCO 2010-handwritten document image binarization competition. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2010. p. 727–32.
125. Pratikakis I, Gatos B, Ntirogiannis K. ICDAR 2011 document image binarization contest (DIBCO 2011). In: International conference on document analysis and recognition (ICDAR). IEEE; 2011. p. 1506–10.
126. Pratikakis I, Gatos B, Ntirogiannis K. ICFHR 2012 competition on handwritten document image binarization (H-DIBCO 2012). In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2012. p. 817–22.
127. Pratikakis I, Gatos B, Ntirogiannis K. ICDAR 2013 document image binarization contest (DIBCO 2013). In: International conference on document analysis and recognition (ICDAR). IEEE; 2013. p. 1471–76.
128. Pratikakis I, Zagoris K, Barlas G, Gatos B. ICFHR2016 handwritten document image binarization contest (H-DIBCO 2016). In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2016. p. 619–23.
129. Pratikakis I, Zagoris K, Barlas G, Gatos B. ICDAR2017 competition on document image binarization (DIBCO 2017). In: International conference on document analysis and recognition (ICDAR). IEEE; 2017. p. 1395–403.
130. Pratikakis I, Zagoris K, Kaddas P, Gatos B. ICFHR2018 competition on handwritten document image binarization contest (H-DIBCO 2018). In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2018. p. 1–1.
131. Pratikakis I, Zagoris K, Karagiannis X, Tsochatzidis L, Mondal T, ICDAR Marthot-Santaniello I. Competition on document image binarization (DIBCO 2019). In: International conference on document analysis and recognition (ICDAR); 2019, p. 1547–56.

132. Rabelo JC, Zanchettin C, Mello CA, Bezerra BL. A multi-layer perceptron approach to threshold documents with complex background. In: International conference on systems, man, and cybernetics (SMC). IEEE; 2011. p. 2523–30.
133. Ramírez-Ortegón MA, Tapia E, Ramírez-Ramírez LL, Rojas R, Cuevas E. Transition pixel: a concept for binarization based on edge detection and gray-intensity histograms. Pattern Recognit. 2010;43(4):1233–43.
134. Ramírez-Ortegón MA, Tapia E, Rojas R, Cuevas E. Transition thresholds and transition operators for binarization and edge detection. Pattern Recognit. 2010;43(10):3243–54.
135. Ramírez-Ortegón MA, Duénez-Guzmán EA, Rojas R, Cuevas E. Unsupervised measures for parameter selection of binarization algorithms. Pattern Recognit. 2011;44(3):491–502.
136. Ramírez-Ortegón MA, Märgner V, Cuevas E, Rojas R. An optimization for binarization methods by removing binary artifacts. Pattern Recognit Lett. 2013;34(11):1299–306.
137. Ramírez-Ortegón MA, Ramírez-Ramírez LL, Märgner V, Messaoud IB, Cuevas E, Rojas R. An analysis of the transition proportion for binarization in handwritten historical documents. Pattern Recognit. 2014;47(8):2635–51.
138. Ramírez-Ortegón MA, Ramírez-Ramírez LL, Messaoud IB, Märgner V, Cuevas E, Rojas R. A model for the gray-intensity distribution of historical handwritten documents and its application for binarization. Int J Doc Anal Recognit (IJDAR). 2014;17(2):139–60.
139. Rivest-Hénault D, Moghaddam RF, Cheriet M. A local linear level set method for the binarization of degraded historical document images. Int J Doc Anal Recognit (IJDAR). 2012;15(2):101–24.
140. Roe E, Mello CA. Thresholding color images of historical documents with preservation of the visual quality of graphical elements. Integr Comput Aided Eng. 2018;25(3):261–72.
141. Romero A, Ballas N, Kahou SE, Chassang A, Gatta C, Bengio Y. Fitnets: hints for thin deep nets. In: International conference on learning representations (ICLR); 2015.
142. Rowley-Brooke R, Pitié F, Kokaram A. A ground truth bleed-through document image database. In: International conference on theory and practice of digital libraries. Springer; 2012. p. 185–96.
143. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, et al. Imagenet large scale visual recognition challenge. Int J Comput Vis (IJCV). 2015;115(3):211–52.
144. Saddami K, Munadi K, Muchallil S, Arnia F. Improved thresholding method for enhancing jawi binarization performance. In: International conference on document analysis and recognition (ICDAR). IEEE; 2017. p. 1:1108–13.
145. Saund E, Lin J, Sarkar P. Pixlabeler: User interface for pixel-level labeling of elements in document images. In: International conference on document analysis and recognition (ICDAR). IEEE; 2009. p. 646–50.
146. Sauvola J, Pietikäinen M. Adaptive document image binarization. Pattern Recognit. 2000;33(2):225–36.
147. Sehad A, Chibani Y, Hedjam R, Cheriet M. Gabor filter-based texture for ancient degraded document image binarization. Pattern Anal Appl. 2018;1–22.
148. Şekeroğlu B, Khashman A. Performance evaluation of binarization methods for document images. In: International conference on advances in image processing. ACM; 2017, p. 96–102.
149. Seuret M, Chen K, Eichenbergery N, Liwicki M, Ingold R. Gradient-domain degradations for improving historical documents images layout analysis. In: International conference on document analysis and recognition (ICDAR). IEEE; 2015. p. 1006–10.
150. Sezgin M, Sankur B. Survey over image thresholding techniques and quantitative performance evaluation. J Electronic Imaging. 2004;13(1):146–66.
151. Shafait F, Keysers D, Breuel TM. Efficient implementation of local adaptive thresholding techniques using integral images. In: Document recognition and retrieval XV. International Society for Optics and Photonics; 2008. p. 6815:681510.
152. Shaus A, Sober B, Turkel E, Piasetzky E. Beyond the ground truth: Alternative quality measures of document binarizations. In: International conference on frontiers in handwriting recognition (ICFHR). IEEE; 2016. p. 495–500.
153. Simard PY, Steinkraus D, Platt JC. Best practices for convolutional neural networks applied to visual document analysis. In: International conference on document analysis and recognition (ICDAR). IEEE; 2003. p. 958.
154. Smith EHB. An analysis of binarization ground truthing. In: International workshop on document analysis systems (DAS). ACM, 2010. p. 27–34.
155. Smith EHB, An C. Effect of "ground truth" on image binarization. In: International workshop on document analysis systems (DAS). IEEE; 2012. p. 250–4.
156. Smith EHB, Likforman-Sulem L, Darbon J. Effect of pre-processing on binarization. In: Document recognition and retrieval XVII. International Society for Optics and Photonics; 2010. p. 7534:75340H.
157. Sokratis V, Kavallieratou E. A tool for tuning binarization techniques. In: International conference on document analysis and recognition (ICDAR). IEEE; 2011. p. 1–5.
158. Stathis P, Kavallieratou E, Papamarkos N. An evaluation technique for binarization algorithms. J Univ Comput Sci. 2008;14(18):3011–30.
159. Stommel M, Frieder G. Automatic estimation of the legibility of binarised historic documents for unsupervised parameter tuning. In: International conference on document analysis and recognition (ICDAR). IEEE; 2011. p. 104–8
160. Su B, Lu S, Tan CL. Combination of document image binarization techniques. In: International conference on document analysis and recognition (ICDAR). IEEE; 2011. p. 22–6.
161. Su B, Lu S, Tan CL. A learning framework for degraded document image binarization using markov random field. In: International conference on pattern recognition (ICPR). IEEE; 2012. p. 3200–3.
162. Su B, Lu S, Tan CL. Robust document image binarization technique for degraded document images. Trans Image Process. 2013;22(4):1408–17.
163. Su B, Tian S, Lu S, Dinh TA, Tan CL. Self learning classification for degraded document images by sparse representation. In: International conference on document analysis and recognition (ICDAR). IEEE; 2013. p. 155–9.
164. Tensmeyer C, Martinez T. Document image binarization with fully convolutional neural networks. In: International conference on document analysis and recognition (ICDAR). IEEE; 2017. p. 1:99–104.
165. Tensmeyer C, Brodie M, Saunders D, Martinez T. Generating realistic binarization data with generative adversarial networks. In: International conference on document analysis and recognition (ICDAR). IEEE; 2019. p. 172–7.
166. Valizadeh M, Kabir E. An adaptive water flow model for binarization of degraded document images. Int J Doc Anal Recognit (IJDAR). 2013;16(2):165–76.
167. Vats E, Hast A, Singh P. Automatic document image binarization using Bayesian optimization. In: Workshop on historical document imaging and processing (HIP). ACM; 2017. p. 89–94.

168. Vo GD, Park C. Robust regression for image binarization under heavy noise and nonuniform background. Pattern Recognit. 2018;81:224–39.
169. Vo QN, Kim SH, Yang HJ, Lee G. Binarization of degraded document images based on hierarchical deep supervised network. Pattern Recognit. 2018;74:568–86.
170. Westphal F, Grahn H, Lavesson N. Efficient document image binarization using heterogeneous computing and parameter tuning. Int J Doc Anal Recognit (IJDAR). 2018;21(1–2):41–58.
171. Westphal F, Lavesson N, Grahn H. Document image binarization using recurrent neural networks. In: International workshop on document analysis systems (DAS). IEEE; 2018. p. 263–8.
172. Wiener N. Extrapolation, interpolation, and smoothing of stationary time series. Cambridge: The MIT Press; 1964.
173. Wolf C, Jolion JM, Chassaing F. Text localization, enhancement and binarization in multimedia documents. In: International conference on pattern recognition (ICPR). IEEE; 2002, vol 2. p. 1037–40.
174. Wu Y, Natarajan P, Rawls S, AbdAlmageed W. Learning document image binarization from data. In: International conference on image processing (ICIP). IEEE; 2016. p. 3763–67.
175. Würsch M, Liwicki M, Ingold R. Web services in document image analysis-recent developments on DIVAservices and the importance of building an ecosystem. In: International workshop on document analysis systems (DAS). IEEE; 2018. p. 334–9.
176. Xiong W, Xu J, Xiong Z, Wang J, Liu M. Degraded historical document image binarization using local features and support vector machine (SVM). Optik. 2018;164:218–23.
177. Zemouri ET, Chibani Y, Brik Y. Restoration based contourlet transform for historical document image binarization. In: International conference on multimedia computing and systems (ICMCS). IEEE; 2014. p. 309–13.
178. Zhao J, Shi C, Jia F, Wang Y, Xiao B. Document image binarization with cascaded generators of conditional generative adversarial networks. Pattern Recognit. 2019;96:106968.