



Article

From Classic to Cutting-Edge: A Near-Perfect Global Thresholding Approach with Machine Learning

Nicolae Tarbă , Costin-Anton Boiangiu  and Mihai-Lucian Voncilă 

Computer Science and Engineering Department, Faculty of Automatic Control and Computers, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania; costin.boiangiu@upb.ro (C.-A.B.); mihai_lucian.voncila@stud.acs.upb.ro (M.-L.V.)

* Correspondence: nicolae.tarba@upb.ro

Abstract

Image binarization is an important process in many computer-vision applications. This transforms the color space of the original image into black and white. Global thresholding is a quick and reliable way to achieve binarization, but it is inherently limited by image noise and uneven lighting. This paper introduces a global thresholding method that uses the results of classical global thresholding algorithms and other global image features to train a regression model via machine learning. We prove through nested cross-validation that the model can predict the best possible global threshold with an average F-measure of 90.86% and a confidence of 0.79%. We apply our approach to a popular computer vision problem, document image binarization, and compare popular metrics with the best possible values achievable through global thresholding and with the values obtained through the algorithms we used to train our model. Our results show a significant improvement over these classical global thresholding algorithms, achieving near-perfect scores on all the computed metrics. We also compared our results with state-of-the-art binarization algorithms and outperformed them on certain datasets. The global threshold obtained through our method closely approximates the ideal global threshold and could be used in a mixed local-global approach for better results.



Academic Editors:
Douglas O'Shaughnessy and
Andrea Prati

Received: 2 June 2025

Revised: 9 July 2025

Accepted: 17 July 2025

Published: 21 July 2025

Citation: Tarbă, N.; Boiangiu, C.-A.; Voncilă, M.-L. From Classic to Cutting-Edge: A Near-Perfect Global Thresholding Approach with Machine Learning. *Appl. Sci.* **2025**, *15*, 8096. <https://doi.org/10.3390/app15148096>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: document image binarization; global thresholding; machine learning; image preprocessing; ideal global threshold; ensemble learning; computer vision; optical character recognition (OCR) pre-processing

1. Introduction

Image binarization is the process of converting an image with varying pixel intensities into one with only two: black and white. The simplest way of doing this is by choosing a threshold intensity and converting every pixel with a lower intensity to a black pixel in the binarized image and every pixel with a higher intensity to a white pixel. This is called global threshold image binarization. Choosing different thresholds for different pixels based on their neighborhoods is called local threshold image binarization.

A popular use case for image binarization is the preprocessing of document scans in preparation for digitization. For this use case, several datasets are available, containing low-quality images of documents and corresponding ground-truth images that have black pixels where there was text in the original image and white pixels everywhere else. The quality of the binarized image can be measured using objective metrics by comparing the binarized image to the ground-truth image.

Global algorithms operate in the image histogram domain, resulting in a complexity based on the number of pixel intensities in the image, whereas local algorithms operate on each pixel for a complexity based on the number of pixels in the image, which is always much larger than the number of pixel intensities. Generally, global algorithms are much faster than local ones but yield lower-quality results on low-quality images owing to uneven lighting, noise, etc. The faster processing time and lower complexity are ideal for real-time or resource-constrained applications, such as embedded systems. For any given image, the maximum possible binarization quality achievable through global thresholding is not necessarily the highest possible binarization quality. Local thresholding algorithms have no such limitations; however, their computational cost is higher, and they only consider features from surrounding pixels, making them prone to specific failures.

Global thresholding algorithms provide varying results based on the type of noise and lighting in an image. Non-uniform lighting can be mitigated by applying global algorithms to image patches, such as the patch-based methods presented by Tahseen et al. [1], resulting in a mixed global–local approach.

We propose a machine learning model whose novelty lies in employing various thresholds produced by classical global thresholding algorithms as input features alongside other global image features to produce results as close as possible to the highest possible quality achievable through global thresholding.

The proposed solution is described in detail in Section 2. Evaluation metrics that allow comparison of the performance of global thresholding on different images are defined, and the methodology for evaluating the tested algorithms and employed datasets is laid out.

Section 3 presents the results of the proposed solution and compares them with the classic global thresholding algorithms used as features and the best possible global threshold. The proposed solution outperformed the other algorithms and achieved results very close to the ideal threshold, with a promising confidence interval.

Section 4 presents a discussion based on the presented results and visual examples, followed by conclusions and possible avenues for future work in Section 5.

In the Following Subsections, Previous Works Are Presented and Divided into Classical and Deep-Learning-Based Approaches.

1.1. Classical Approaches

Classical approaches can be split into two categories, in terms of how they operate on the image histogram space, linear and non-linear solutions. While the linear methods are straightforward and efficient, they require further exploration in our context; thus, we have opted not to implement them yet. They could be added to the features in future work without changing the time complexity of the solution. On the other hand, non-linear approaches, while potentially more powerful in handling complex relationships, introduce significant complexity. Given this trade-off, we chose not to pursue these methods either, prioritizing simplicity and computation time in our current implementation.

Wang and Fan [2] define the optimal threshold as the one that maximizes the objective function of the one-dimensional Tsallis entropy correlation single-threshold method:

$$S_q(t) = \frac{1}{q-1} \left(2 - \sum_{i=0}^t \left(\frac{h(i)}{c(t)} \right)^q - \sum_{i=t+1}^{I_{max}} \left(\frac{h(i)}{1-c(t)} \right)^q \right), \quad (1)$$

where q is a real number whose value can be experimented with.

Elen and Dönmez [3] separate the image histogram into three regions: α between $\mu - \sigma$ and $\mu + \sigma$, β_L between 0 and $\mu - \sigma$, and β_H between $\mu + \sigma$ and I_{max} , where μ is the average pixel value and σ is the standard deviation. Then, they compute $\mu_\alpha = \sum_{x \in \alpha} ip(i)$ and $\mu_\beta = \sum_{x \in \beta} ip(i)$ where $\beta = \beta_L \cup \beta_H$. They define the optimum thresh-

old as $T_{opt} = (\mu_\alpha + \mu_\beta)/2$. They compared their algorithm to several PDE-based ones, such as Guo and He [4], Guo et al. [5], and Mahani et al. [6].

Bogiatzis and Papadopoulos [7] compute two threshold candidates using a fuzzy inclusion indicator based on Zadeh's fuzzy implication and the following fuzzy entropy formula:

$$E_1 = 2 \frac{\sum_{x \in X} \min(m_{A^C}(x), m_A(x))}{\sum_{x \in X} \max(m_{A^C}(x), m_A(x))}, \quad (2)$$

where X is the set of pixels in the image; $m_A(x)$ is the membership grade of x in the fuzzy subset A ; and A^C is the complementary subset of A . Then, they compute a classification function and choose one of the two candidates based on a criterion.

Ashir [8] proposed a multilevel thresholding algorithm that can be extended to implement image binarization. It is based on computing the mean pixel intensity for the entire image μ , generating a gradient image I_g by computing the intensity difference between the original image and μ , and computing the mean gradient pixel μ_g . They choose the following threshold:

$$T = \begin{cases} \mu - \mu_g, & \text{if } \sum_{i=\mu-\mu_g}^{\mu} h(i) < \sum_{i=\mu}^{\mu+\mu_g} h(i) \\ \mu + \mu_g, & \text{otherwise} \end{cases}. \quad (3)$$

Rani et al. [9] proposed a longer algorithm that starts by computing the Laplacian of the image and subtracting the Laplacian from the image. Then, they slide a 5×5 window on the image and compute its average value in each position, resulting in an average image I_{avg} . The threshold T is selected as

$$T = \max(I_{avg}) + \text{median} \left(\min_{\text{on each column}} (I_{avg}) \right). \quad (4)$$

1.2. Deep-Learning Approaches

Deep-learning approaches currently dominate the state-of-the-art of document image binarization. Convolutional neural networks are the most popular implementation, followed by vision transformers, but both produce competitive results.

SAE [10] is an auto-encoder that harnesses a convolutional neural network to encode the original image into a probability map. A global threshold is then applied to the probability map to generate the binarized image. MRAM [11] uses a convolutional random field to generate the binarized image from a probability map produced by a multi-resolution attention model.

DeepOtsu [12] applies Otsu's thresholding algorithm on an enhanced image generated by a convolutional neural network that iteratively removes noise from the original image. 2DMN [13] is a bidimensional morphological network that removes noise from the original image using simple operations such as dilation and erosion.

DD-GAN [14] is a dual discriminator generative adversarial network comprising two networks, one to process the image globally and one locally. A threshold is applied to the generated image to obtain the binarized image. Suh [15] uses six generative adversarial networks: four colour-independent ones that extract information from the input image and two that transform the output images of the first four into the final binarized image.

cGANs [16] use conditional generative adversarial networks to combine multi-scale information. First, text pixels are extracted from the original image at various scales, and then, the multi-scale binarized images are combined to obtain the final binarized image. DE-GAN [17] also uses conditional generative adversarial networks.

D2BFormer [18] uses a vision transformer for semantic segmentation, transforming the original image into a two-channel feature map, one channel for the foreground and the other for the background. They then use the feature map to binarize the original image.

DocDiff [19] consists of a coarse predictor, which recovers primary low-frequency information, and of a high-frequency residual refinement module, which predicts high-frequency information using diffusion models.

DocEntr [20] and DocBinFormer [21] use vision transformers in an encoder-decoder architecture. The encoder transforms pixel patches into latent representations without using any convolutional layers, and the decoder generates the binarized image from the latent representations.

GDB [22] uses gated convolutions that, unlike regular convolutions, precisely extract stroke edges. First, a coarse sub-network generates a precise feature map, and then, a refinement sub-network uses gated convolutions to further refine the feature map. GDBMO is a version of GDB that combines local and global features to further improve the results.

2. Materials and Methods

2.1. Features

The following algorithms were directly used in the proposed solution as features in the model that we trained to predict the optimal global threshold. They operate directly on the image histogram, optimizing various metrics, and can be computed in constant time ($O(1)$) in relation to the image size. Although some of these algorithms are iterative, they converge rapidly.

Otsu [23] proposed a popular global thresholding algorithm. For a given threshold k , the pixels are divided into two classes: the 0 class, composed of pixels with intensities lower than or equal to k , and the 1 class, composed of pixels with intensities greater than k . Between-class variance is defined as the weighted sum of the variances of the two classes. Otsu's threshold is defined as the threshold with the largest between-class variance, defined as follows:

$$\sigma_{01}^2 = W_0(t)W_1(t)(\mu_1(t) - \mu_0(t)), \quad (5)$$

where $W_0(t)$ is the ratio of pixels with intensities lower than or equal to t ; $W_1(t)$ is the ratio of pixels with intensities greater than t ; and μ_0 , μ_1 are the mean values of the 0 class and 1 class pixel intensities, respectively.

Kittler and Illingworth [23] proposed a computationally efficient solution for minimum error thresholding. They assumed that the two pixel classes follow normal distributions, splitting the probability density function $p(i)$ into two components, $p_0(i)$ and $p_1(i)$, normally distributed with means μ_0 and μ_1 , standard deviations σ_0 and σ_1 , and a priori probabilities $P_0(i)$ and $P_1(i)$. They choose a threshold that minimizes the following criterion function:

$$J(t) = 1 + 2(P_0(t)\log \sigma_0(t) + P_1(t)\log \sigma_1(t)) - 2(P_0(t)\log P_0(t) + P_1(t)\log P_1(t)) \quad (6)$$

The function is easy to compute, and finding its minimum is relatively simple, as the function is smooth.

Lloyd [23] proposed an iterative thresholding algorithm that minimizes the mean squared error between the original and the binarized images. In the first iteration, the average intensity is chosen as the threshold, and in further iterations, the threshold is updated via

$$t_{i+1} = \frac{\mu_1(t_i) + \mu_0(t_i)}{2} + \frac{\sigma^2 \log(W_0(t_i)/W_1(t_i))}{\mu_1(t_i) - \mu_0(t_i)}, \quad (7)$$

where σ^2 is the variance of the entire image. The algorithm converges when the final threshold is equal to the previous threshold.

Sung et al. [24] argued that the within-class variance is more useful than the between-class variance as a selection criterion for the optimal threshold. The within-class variance is defined as

$$\sigma_{0+1}^2 = W_0(t)\sigma_0^2(t) + W_1(t)\sigma_1^2(t), \quad (8)$$

where σ_0^2, σ_1^2 are the variances of each class. If there is more than one threshold for which the within-class variance is minimal, then they consider the average of those thresholds to be the optimal one. Their experiments show that their method performs better than Otsu's on their synthetic dataset.

Ridler and Calvard [23] proposed an iterative thresholding algorithm very similar to Lloyd's. In the first iteration, the average intensity is chosen as the threshold, and in further iterations, the threshold is updated via

$$t_{i+1} = \frac{\mu_0(t_i) + \mu_1(t_i)}{2}. \quad (9)$$

The algorithm converges when the final threshold is equal to the previous threshold.

Huang and Wang [23] proposed an algorithm based on computing a measure of fuzziness, such as Shannon's entropy, for each possible threshold and choosing the lowest threshold for which the minimum fuzziness is reached. They use the following formula for fuzziness:

$$E(t) = \sum_{i=1}^{I_{max}} S(\mu_t(i))h(i), \quad (10)$$

where $h(i)$ represents how many times the i th intensity appears in the image,

$$S(\mu_t(i)) = \mu_t(i)\ln \mu_t(i) - (1 - \mu_t(i))\ln(1 - \mu_t(i)) \quad (11)$$

is Shannon's entropy, and the membership function is

$$\mu_t(i) = \begin{cases} (1 + |i - \mu_0|/C)^{-1}, & \text{if } i \leq t \\ (1 + |i - \mu_1|/C)^{-1}, & \text{otherwise} \end{cases} \quad (12)$$

where C is a constant value such that $0.5 \leq \mu_t(i) \leq 1$.

Ramesh et al. [23] proposed an algorithm based on functional approximation of the histogram composed of recursive bilevel approximations that minimize an error function. They compare the results of two error functions:

- The sum of the square errors:

$$E(t) = \sum_{i=1}^t (i - \mu_0)^2 + \sum_{i=t+1}^{I_{max}} (i - \mu_1)^2 \quad (13)$$

- The sum of the variances of each class:

$$E(t) = \sigma_0^2 + \sigma_1^2. \quad (14)$$

Their experimental results show that the latter error function produces better results than the former.

Li and Lee [25] proposed an algorithm based on minimizing the cross-entropy between the original image and the binarized image. The cross-entropy is computed for each possible threshold with the following formula:

$$H(t) = \sum_{i=1}^{t-1} ih_i \ln \frac{i}{\mu_0(t)} + \sum_{i=t}^{I_{max}} ih_i \ln \frac{i}{\mu_1(t)}, \quad (15)$$

where h_i is the number of pixels with intensity i in the image, and I_{max} is the maximum intensity. Li and Tam [26] proposed an iterative variation of this algorithm, boasting improved computational speed at the cost of accuracy. Brink and Pendock [27] proposed a similar algorithm, but they compute the cross-entropy with the following formula:

$$H(t) = \sum_{i=1}^t \mu_0(t) \log \frac{\mu_0(t)}{i} + \sum_{i=t+1}^{I_{max}} \mu_1(t) \log \frac{\mu_1(t)}{i}. \quad (16)$$

Kapur et al. [26] proposed an algorithm based on maximizing an evaluation function:

$$f(t) = \frac{H(t)}{H(I_{max})} \frac{\ln c(t)}{\ln(\max_{i=1}^t p(i))} + \frac{1-H(t)}{H(I_{max})} \frac{\ln(1-c(t))}{\ln(\max_{i=t+1}^{I_{max}} p(i))}, \quad (17)$$

where they use the classic information entropy formula:

$$H(t) = - \sum_{i=1}^t p(i) \ln p(i) \quad (18)$$

and

$$c(t) = \sum_{i=1}^t p(i) \quad (19)$$

is the cumulative probability density function.

Sahoo et al. [26] proposed an algorithm based on maximizing the sum of Renyi's entropies associated with each class:

$$\begin{aligned} H_0^\alpha &= (1 - \alpha)^{-1} \ln \sum_{i=0}^t \left(\frac{p(i)}{W_0} \right)^\alpha \\ H_1^\alpha &= (1 - \alpha)^{-1} \ln \sum_{i=t+1}^{I_{max}} \left(\frac{p(i)}{W_1} \right)^\alpha. \end{aligned} \quad (20)$$

They observed that there are three distinct thresholds t_1, t_2, t_3 that maximize the sum depending on whether α is lower than, equal to, or greater than 1, and use all three values to determine the optimal threshold:

$$t = t_{[1]} \left(p(t_{[1]}) + 0.25\omega\beta_1 \right) + 0.25t_{[2]}\omega\beta_2 + t_{[3]} \left(1 - p(t_{[3]}) + 0.25\omega\beta_3 \right), \quad (21)$$

where $t_{[1]}, t_{[2]}, t_{[3]}$ are the order statistics of the thresholds t_1, t_2, t_3 , respectively;

$$\omega = p(t_{[3]}) - p(t_{[1]}); \quad (22)$$

and $\beta_1, \beta_2, \beta_3$ take different values based on the distances between $t_{[1]}, t_{[2]}, t_{[3]}$.

Shanbhag [26] proposed a modified version of Kapur's algorithm. They propose a different information measure:

$$\text{info}_0(t) = |\text{info}_0(t) - \text{info}_1(t)| \quad (23)$$

and choose the threshold that minimizes this measure. They define

$$\begin{aligned} \text{info}_0(t) &= -\frac{1}{W_0(t)} \sum_{i=1}^t p(i) \log P_{0,t}(i) \\ \text{info}_1(t) &= -\frac{1}{W_1(t)} \sum_{i=t+1}^{I_{\max}} p(i) \log P_{1,t}(i), \end{aligned} \quad (24)$$

where

$$\begin{aligned} P_{0,t}(t-i) &= 0.5 + \frac{1}{2W_0(t)} \sum_{j=0}^i p(t-i+j) \\ P_{1,t}(t+i) &= 0.5 + \frac{1}{2W_1(t)} \sum_{j=0}^i p(t+i-j). \end{aligned} \quad (25)$$

Yen et al. [26] proposed a maximum entropy criterion based on the discrepancy between the binarized image and the original image and the number of bits required to represent the binarized image. They define entropy as

$$E(t) = -\sum_{i=1}^t \frac{p(i) \ln p(i)}{W_0(t)} - \sum_{i=t+1}^{I_{\max}} \frac{p(i) \ln p(i)}{W_1(t)}. \quad (26)$$

They choose the threshold with the maximum entropy.

Tsai [26] proposed an algorithm based on preserving the moments of the original image in the binarized image. They use the first 4 raw moments and obtain the following system:

$$\begin{aligned} W_0(t) + W_1(t) &= m_0 \\ W_0(t)z_0 + W_1(t)z_1 &= m_1 \\ W_0(t)z_0^2 + W_1(t)z_1^2 &= m_2 \\ W_0(t)z_0^3 + W_1(t)z_1^3 &= m_3, \end{aligned} \quad (27)$$

where m_i is the i -th raw moment, and z_0, z_1 are representative gray values for the two classes of pixels. Considering that

$$W_0(t) + W_1(t) = 1, \quad \forall t \in \overline{1, I_{\max}}, \quad (28)$$

solving the system is trivial, and the threshold can be determined from the resulting $W_0(t)$ as the $W_0(t)$ -th percentile.

Sarle's bimodality coefficient (BC) [28] uses skewness and kurtosis to ascertain how separable two populations that form a bimodal distribution are. It can also be generalized (GBC) [29] by using standardized moments of higher orders. For both BC and GBC, values range from 0 to 1, with 1 indicating no overlap between the two populations and lower values indicating more overlap. The formula for the k -th GBC is

$$\text{GBC}_k = \begin{cases} 0, & \text{for distr. concentrated on one point} \\ \frac{\tilde{\mu}_{2k+1}^2 + 1}{\tilde{\mu}_{2k+2} \tilde{\mu}_{2k}}, & \text{otherwise} \end{cases} \quad (29)$$

where $\tilde{\mu}_n$ is the n -th standardized moment. Sarle's bimodality coefficient is GBC_1 .

2.2. Tools

ML.NET [30] is a framework that allows users to train and evaluate a plethora of machine-learning models. Its AutoML component allows users to automatically explore various hyperparameter configurations for said models using one of the several search strategies, such as random search, grid search, and cost frugal tuner (CFT) [31]. There are several model types available in AutoML for regression, but LightGbm [32] always scored

best in our tests. LightGbm is a gradient boosting decision tree that uses gradient-based one-sided sampling to exclude a significant proportion of data instances with small gradients, using only the rest to estimate the information gain, and exclusive feature bundling to bundle mutually exclusive features to reduce the number of features.

ML.NET does not allow for custom metrics. The metrics available for regression are the mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and R-squared (R²), also known as the coefficient of determination [33]. For the first three metrics, values range from 0 to infinity, with 0 indicating a perfect fit. For R² values range from negative infinity to 1, with 1 indicating a perfect fit. Because MSE and RMSE square the errors, they are more sensitive to larger errors than MAE.

Training a model on a dataset might cause overfitting, i.e., maximizing the performance of the model for the training dataset at the cost of lower performance on new datasets. Resampling the dataset can alleviate or even eliminate overfitting. According to Berrar [34], 10-fold cross-validation is one of the most widely used data resampling methods. k-fold cross-validation splits a dataset into k equally sized disjunct partitions and forms k training sets and k test sets. Each test set consists of one of the k partitions, and its corresponding training set consists of the other k-1 partitions. For each training set, the model is fitted on it and evaluated on its corresponding test set, resulting in k performance metrics, one for each test set, which can be averaged to obtain the performance metric of the model.

Bates et al. [35] introduced nested cross-validation and proved that the method is more robust than cross-validation in many examples. The idea behind nested cross-validation is splitting the dataset into train/test sets twice, fitting the model on the inner train sets and evaluating them on the inner test sets, fitting them on the outer train sets and evaluating them on the outer test sets and then using all the obtained metric on all the test sets to compute a robust approximand for the metric consisting of an average value and an MSE.

2.3. Dataset and Metrics

Our dataset is composed of all the images from the following datasets: the DIBCO datasets [36–45], a synthetic dataset from [46], the CMATERdb dataset [47], a subset of 10 pages from the Einsiedlen Stiftsbibliothek [48], a subset of 10 pages from the Salzinnen Antiphonal Manuscript [49], a subset of 15 images from the LiveMemory project [50], the Nabuco dataset [51], the Noisy Office dataset [52], the LRDE-DBD dataset [53], the palm leaf dataset [54], the PHIBD 2012 dataset [55], the Rahul Sharma dataset [56], and a multispectral dataset [57]. To enrich the dataset, 15 variants were generated for each image by adjusting the gamma value with values ranging from 0.5 to 2. For each image, 48 features were computed: the thresholds resulting from the 15 algorithms presented in Section 2.1; the confidence values for the 7 algorithms that provide them, Otsu, Lloyd, Ridler, Huang, Ramesh, Li and Lee, and Li and Tam (which represent the normalized values of their respective optimization functions for their selected thresholds); the normalized between-class variances associated with the computed thresholds; the pixel average, standard deviation, and normalized standardized moments up to the eighth order; Sarle's bimodality coefficient; and the generalized bimodality coefficients up to the third order. Higher-order moments require significantly better precision than that offered by the standard 64-bit double floating-point representation. In order to ensure the required precision, the Boost Multiprecision C++ library [58] was employed using 256 bits for the `cpp_bin_float` back-end.

All features were normalized using min-max normalization, which ensures all values are in the $[0, 1]$ range by subtracting *min* from all values and then dividing them by $max - min$. The min value was considered 0 for all features that could not have negative values. The max value was considered the maximum possible value for all features with

values that, in practice, reach values close to their theoretical upper limit (255 for thresholds, 127.5 for the standard deviation, 127.5^2 for all variances, and 1 for all bimodality coefficients). For the features that do not have a finite or obvious lower/upper limit (both limits for standardized moments of odd orders and upper limits for standardized moments of even orders), quantiles were used instead of min and max values (0.001 quantile for the lower limit, and 0.999 quantile for the upper limit) to clamp outlier values to a more compact interval, ensuring a less sparse distribution of feature values.

A popular metric used to evaluate binarized images is the F-measure (FM), defined as

$$FM = \frac{2TP}{2TP + F}, \quad (30)$$

where TP is the number of pixels correctly classified as pixels containing text, and F is the number of pixels incorrectly classified. FM takes values between 0 and 1, with 1 indicating a perfect classifier. However, because we are using global thresholding, the maximum value of FM can differ from image to image. The maximum possible value of FM for an image is not necessarily provided by a single threshold. If two thresholds T_1 and T_2 provide the maximum possible FM , then all thresholds in the interval $\overline{T_1}, \overline{T_2}$ provide the maximum possible FM . The ideal threshold of an image, in the global thresholding scenario, was defined as the middle of the largest interval of thresholds that provide the maximum possible FM for that image. The ideal threshold was computed for each image in the dataset and used as the column to be predicted by the ML models.

The goal of the experiment is to find a regression model instance that approximates the ideal threshold as closely as possible. Because the maximum possible FM on different image sets is not necessarily the same, we define the relative FM (FM_r) as

$$FM_r = \frac{FM}{FM_{Max}}, \quad (31)$$

where FM is the FM provided by the predicted threshold, and FM_{Max} is the maximum possible FM of the image. With FM_r , the performance on different images can be compared as the values it can take are always between 0 and 1, and the maximum value of 1 can always be reached regardless of the image.

Other popular metrics are peak signal-to-noise ratio (PSNR) and MSE. Both PSNR and FM are better when their values are higher, so we define relative PSNR ($PSNR_r$) similarly to FM_r as

$$PSNR_r = \frac{PSNR}{PSNR_{Max}}, \quad (32)$$

where $PSNR$ is the PSNR provided by the predicted threshold, and $PSNR_{Max}$ is the maximum possible PSNR of the image. MSE is better when its values are lower, so we define relative MSE (MSE_r) as

$$MSE_r = \frac{MSE_{Min} + \epsilon}{MSE + \epsilon}, \quad (33)$$

where MSE is the MSE provided by the predicted threshold; MSE_{Min} is the minimum possible MSE of the image, and we add ϵ to avoid division by 0, while still making sure that MSE_r can take all values in $(0, 1]$.

2.4. Proposed Solution

In order to achieve nested cross-validation, the dataset was split into 11 folds, one fold was set apart as the outer test set, and the remaining 10 folds were set as the outer training set. An ML.NET AutoML regression experiment was set up with cross-validation

on the outer training set, with a time limit of 60 min, using MSE as the optimizing metric, exploring all the model types available for regression, searching for hyperparameters using an implementation of CFT for hierarchical search spaces. Lower time limits were also tested but produced slightly worse results. Higher time limits were not tested due to increased computational cost. Each of the 10 folds in the outer training set formed an inner test set, with the remaining 9 forming an inner training set. The model instances were fitted on the inner train sets and evaluated on the inner test sets, thus producing 10 different values for the evaluation metrics. The model instances were then refitted on the outer training set and evaluated on the outer test set. Each of the 10 folds was then switched with the one in the outer test set and the model instances were refit on the inner training sets and evaluated on the inner test sets, resulting in a total of 110 different values for the evaluation metrics for models trained on 9 folds and evaluated on one fold and a total of 11 different values for the evaluation metrics for models trained on 10 folds and evaluated on one fold.

We denote FM_r^{in} the values for FM_r obtained on images from the inner test sets and FM_r^{out} the values for FM_r obtained on images from the outer test sets. The nested cross-validation estimand of FM_r is denoted as $\hat{FM}_r$ and is the average over all the inner test sets of FM_r^{in} . For each outer test set, two values are computed:

$$MSE_{in-out} = \left(\overline{FM_r^{in}} - \overline{FM_r^{out}} \right)^2$$

$$\sigma_{out}^2 = \text{variance}(FM_r^{out}), \quad (34)$$

where the $\bar{x}$ denotes the average value of the measure x , and FM_r^{in} refers to all the inner test sets corresponding to the outer test set. The nested cross-validation estimand of the mean squared error of $\hat{FM}_r$ is denoted as $MS\hat{E}_{FM}$ and is the difference between $\overline{MSE_{in-out}}$ and $\overline{\sigma_{out}^2}$. Analogous estimands were computed for the PSNR and the MSE of the binarized image but were not used in the ranking of the models.

A score was defined as

$$\hat{FM}_r - 3\sqrt{MS\hat{E}_{FM}}, \quad (35)$$

and assigned to each model instance and used to determine the best one (a higher score is better). The best instance was then refit on the whole dataset, and the average metrics were computed over the entire dataset to compare the model's results with the input algorithms. Algorithm 1 illustrates, using pseudocode, how the best model was selected using NCV.

Algorithm 1. NCV experiment pseudocode.

Input: dataset

Output: ML model

1. partitions = split the dataset into 11 equally sized partitions
 2. for i from 1 to 11
 3. outerFolds[i].testSet = partitions[i]
 4. outerFolds[i].trainSet = all partitions except partitions[i]
 5. innerFolds = split outerFolds [1].trainSet into 10 folds
 6. models = run AutoML 10-fold cross-validation experiment on innerFolds
 7. bestScore = $-\infty$, bestModel = null
 8. for each model in models
 9. esfm = empty array
 10. for oi from 1 to 11
 11. innerFolds = split outerFolds[oi].trainSet into 10 folds
 12. einfm = empty array
-

```

13.   for ii from 1 to 10
14.       trainSet = innerFolds[ii].trainSet
15.       testSet = innerFolds[ii].testSet
16.       innerModel = model.fit(trainSet)
17.       predicted = innerModel.transform(testSet)
18.       for i from 1 to predicted.size
19.           einfm.add(relative FM of predicted[i])
20.   trainSet = outerFolds[oi].trainSet
21.   testSet = outerFolds[oi].testSet
22.   predicted = outerModel.transform(testSet)
23.   eoutfm = empty array
24.   for i from 1 to predicted.size
25.       eoutfm.add(relative FM of predicted[i])
26.   afm[oi] = pow(average(einfm)-average(eoutfm),2)
27.   bfm[oi] = variance(eoutfm)/eoutfm.size
28.   esfm.concatenate(einfm)
29.  $\widehat{MSE}_{FM}$  = average(afm)-average(bfm)
30.  $\widehat{FM}_r$  = average(esfm)
31. score =  $\widehat{FM}_r - 3\sqrt{\widehat{MSE}_{FM}}$ 
32. if score > bestScore
33.     bestScore = score
34.     bestModel = model

```

3. Results

Table 1 shows the resulting best model instances of several runs of the same nested cross-validation (NCV) experiment, as described in Section 2.4, using all features. The procedure produces consistent results, with a LightGbm regression model always taking first place.

Table 1. NCV results with full feature set.

$\widehat{FM}_r(\%)$	$\widehat{MSE}_{FM}(\%)$	$\widehat{PSNR}_r(\%)$	$\widehat{MSE}_{PSNR}(\%)$	$\widehat{MSE}_r(\%)$	$\widehat{MSE}_{MSE}(\%)$
90.86	7.99	81.64	11.08	62.32	22.01
90.46	8.08	81.16	11.72	61.81	23.69
90.56	8.43	81.11	11.18	61.70	23.86
90.80	8.06	81.40	11.60	61.97	23.97

Three feature subsets were found to impact the results the most: 1—the 15 thresholds, 2—the moments (average, standard deviation, standardized moments from the third to the eighth), and 3—the BCs (Sarle’s bimodality coefficient and the next two generalized bimodality coefficients). Table 2 shows how these subsets were combined, and Table 3 compares the NCV results obtained with each subset and with the full set. The results found for the fourth, fifth, and seventh feature subsets are the best out of the seven and in line with the results of the experiments using all features, showing that the discarded features’ contributions are insignificant. Subset 1 seems to be the most relevant, followed closely by subset 2; however, their combinations yield significantly better results.

Table 2. Feature subsets.

Feature Set	Thresholds	Moments	BCs
1	+	–	–
2	–	+	–
3	–	–	+
4	+	+	–
5	+	–	+
6	–	+	+
7	+	+	+

+ indicates the presence of a subset and – indicates the absence of a subset.

Table 3. NCV result comparison.

Feature Set	$\widehat{FM}_r$ (%)	$\widehat{MSE}_{FM}$ (‰)	$\widehat{PSNR}_r$ (%)	$\widehat{MSE}_{PSNR}$ (‰)	$\widehat{MSE}_r$ (%)	$\widehat{MSE}_{MSE}$ (‰)
1	89.59	8.64	82.38	7.23	63.05	15.37
2	84.15	8.00	74.79	12.52	50.50	24.45
3	66.88	56.19	66.81	15.50	42.98	25.73
4	90.31	6.93	81.58	10.35	62.28	21.18
5	90.29	8.92	82.97	7.43	64.18	15.59
6	86.42	8.13	77.71	15.28	54.90	27.43
7	90.39	7.96	81.75	10.15	62.52	19.89
Full	90.86	7.99	81.64	11.08	62.32	22.01

Table 4 compares the $\overline{FM}$ and $\overline{FM}_r$ for each thresholding algorithm and for the ideal threshold, with the results obtained by refitting the models obtained through NCV to the entire dataset. It should be noted that

$$\frac{\overline{FM}}{\overline{FM}_{max}} \neq \overline{FM}_r \quad (36)$$

because even though

$$FM_r = \frac{FM}{FM_{max}} \quad (37)$$

averaging FM_r produces

$$\overline{FM}_r = n^{-1} \sum FM_r = n^{-1} \sum \frac{FM}{FM_{max}} \quad (38)$$

and this is different from the average value of FM

$$\overline{FM} = n^{-1} \sum FM \quad (39)$$

because FM_{max} is not constant across different images. This is why, in Table 4,

$$\overline{FM}_r \neq \frac{\overline{FM}}{\overline{FM}_{IdealT}} \quad (40)$$

for any thresholding method other than the ideal one. The observation holds true for PSNR and MSE as well.

Table 4. Result comparison on the entire dataset.

Proposed Solution with Feature Set	1	2	3	4	5	6	7	Full
$\overline{FM}$ (%)	81.56	81.65	62.30	81.75	81.73	81.73	81.75	81.55
$\overline{FM}_r$ (%)	99.70	99.79	70.58	99.99	99.97	99.93	99.99	99.74
$\overline{PSNR}$	27.01	27.37	18.17	27.50	27.48	27.49	27.50	27.18
$\overline{PSNR}_r$ (%)	92.94	93.96	69.22	93.99	93.95	93.96	93.99	93.19
$\overline{MSE}$	3202	3236	8156	3244	3250	3236	3244	3271
$\overline{MSE}_r$ (%)	87.50	89.68	70.58	90.60	90.49	90.50	90.60	88.40
Algorithm	Ideal	Otsu	Kittler	Lloyd	Sung	Ridler	Huang	Ramesh
$\overline{FM}$ (%)	81.75	67.11	64.57	63.65	61.68	64.88	53.05	53.67
$\overline{FM}_r$ (%)	100	79.24	73.10	70.62	73.40	77.13	64.48	58.90
$\overline{PSNR}$	27.50	15.89	18.45	18.63	15.35	15.12	11.54	14.41
$\overline{PSNR}_r$ (%)	93.99	60.21	65.84	67.67	55.40	57.91	46.76	59.54
$\overline{MSE}$	3244	10,289	9528	7900	11,790	10,819	16,322	12,205
$\overline{MSE}_r$ (%)	90.6	35.86	45.88	41.01	30.67	33.73	24.22	37.68
Algorithm	Li-Lee	Li-Tam	Brink	Kapur	Sahoo	Shanbhag	Yen	Tsai
$\overline{FM}$ (%)	66.51	68.59	62.34	60.13	59.90	50.22	58.47	60.23
$\overline{FM}_r$ (%)	78.86	79.24	73.10	70.62	73.40	77.13	64.48	58.90
$\overline{PSNR}$	16.75	16.94	14.77	13.40	13.20	11.29	12.85	12.04
$\overline{PSNR}_r$ (%)	61.04	62.79	65.66	62.35	61.53	51.99	60.35	52.38
$\overline{MSE}$	10,332	9656	6444	9383	9387	10,972	9983	9709
$\overline{MSE}_r$ (%)	37.70	39.07	41.21	40.45	39.25	27.60	38.25	25.66

Table 5 shows the average processing time for an image with a known histogram expressed in milliseconds. The images have an average resolution of 3.31 megapixels. Image resolution does not affect processing times because the proposed solution operates linearly on the image histogram. It should be noted that all the times displayed in this paper were obtained on a machine with an Intel 14600 K processor running at up to 5.3 GHz. The application was built on Windows 11 targeting .NET 6.0.

Table 5. Average processing time for an image in the dataset.

Feature Set	1	2	3	4	5	6	7	Full
Feature generation runtime (ms)	2	0 *	0 *	2	2	0 *	2	2
ML regression runtime (ms)	230	340	415	225	214	246	251	234
Total image binarization runtime (ms)	232	340	415	227	216	248	253	236

Times marked with 0 * are less than 0.1 ms.

Table 6 compares the results of the proposed method with those of state-of-the-art global thresholding methods (introduced in Section 1.2) on the H-DIBCO 2014 dataset.

Table 7 shows that the proposed method outperforms state-of-the-art neural-network-based methods (introduced in Section 1.2) on the DIBCO 2019 dataset, but it also shows that, when averaging on all DIBCO datasets, the proposed method cannot compete. This shows that global methods are not obsolete and that local methods are not always the better choice, even when time and resources are not taken into consideration.

Table 6. Result comparison on the H-DIBCO 2014 dataset.

Method	$\overline{PSNR}$ (%)	$\overline{FM}$ (%)
Proposed method	19.26	92.8
Elen and Dönmez	19	92.3
Guo and He	18.54	91.2
Guo et al.	19.17	92.3
Mahani et al.	19.16	92.7

Table 7. Result comparison with state-of-the-art methods on all datasets ($\overline{FM}$) versus DIBCO 2019 dataset $\overline{FM}_{2019}$.

Method	$\overline{FM}$ (%)	$\overline{FM}_{2019}$ (%)	$\overline{PSNR}$	$\overline{PSNR}_{2019}$
Proposed method	83.04	77.71	15.86	16.91
DIBCO winners	88.59	72.88	19.16	14.48
Otsu	76.56	63.87	15.30	12.67
Sauvola	77.66	63.82	15.79	12.66
DeepOtsu	87.04	60.75	19.33	14.44
SAE	88.01	58.78	19.54	12.35
DD-GAN	87.70	58.01	19.87	14.43
MRAM	85.67	49.53	19.18	12.95
2DMN	78.61	42.76	15.55	8.25
cGANs	88.64	62.29	19.66	12.80
DE-GAN	88.52	56.03	19.44	8.37
Suh	89.92	70.63	19.87	14.72
GDB	90.89	73.84	20.41	14.80
GDBMO	91.16	73.50	20.50	14.96
D2BFormer	91.73	66.69	20.83	15.05
DocDiff	-	73.38	-	15.14
DocEnTr	-	59.00	-	13.85
DocBinFormer	92.22	60.31	21.30	14.49

4. Discussion

One thing to note is that the proposed solution's global approach offers various benefits in terms of robustness and computational efficiency. By operating in a global context, rather than depending on local relationships, the algorithm is less affected by noise or small-scale variations that could affect the results, making it especially useful for tasks where global patterns matter more than detailed local structures. This also results in simpler computations, meaning faster processing and lower complexity, which is ideal for real-time or resource-constrained applications, like embedded systems. Additionally, since the solution does not depend on spatial relationships between samples, it generalizes well to various signals that require separation in low- and high-value groups, beyond just bi-dimensional images. The efficiency also benefits OCR systems and other machine learning applications, where local approaches can introduce noise by providing reliable preprocessing and robust features that can be computed quickly and efficiently.

However, while this approach of ignoring local relationships enhances efficiency, it can lead to the loss of important local information, which could be crucial for tasks like edge detection, texture analysis, or applications where neighboring samples are strongly correlated. For such scenarios, methods that take local relationships into account might provide more accurate results.

In the following figures, black represents true positives, white represents true negatives, cyan represents false positives, and red represents false negatives. Figure 1 illustrates the importance of local relationships with two images from the DIBCO 2018 dataset for which global binarization algorithms cannot possibly achieve an acceptable result. For images such as these, there is no global threshold with which you can eliminate both false negatives and false positives.

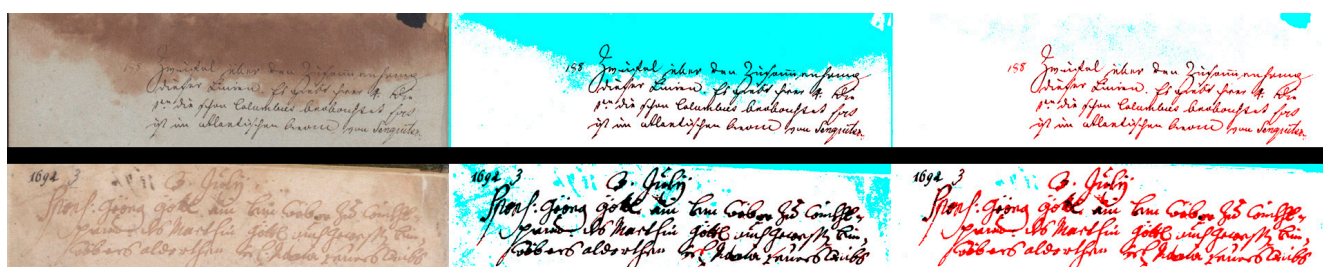


Figure 1. Problematic images for global binarization from DIBCO 2018: 2 (top) and 10 (bottom). From left to right: original image, image binarized with Otsu's method (threshold too high), and image binarized with the proposed method (threshold too low).

Figure 2 shows that on some images, such as image 13 from the challenging DIBCO 2019 dataset, where even state-of-the-art neural-network-based algorithms struggle to produce acceptable results, a global binarization algorithm can produce competitive results.



Figure 2. From top to bottom, left to right: original image, ground truth, Otsu binarization, Sauvola binarization, GDB binarization, proposed solution binarization.

The diverse training dataset produces a model that is more robust in terms of noise, illumination conditions, contrast, etc. Combining multiple datasets carries the risk of hindering model performance for some particular datasets, but the benefit of increasing the robustness outweighs the risk. Cross-validation further reduces this risk, ensuring that the model performs consistently across all folds.

Data augmentation also improves model robustness and performance. The parameters used for all data augmentation transformations were set to produce realistic results that could have been captured from the same documents in different lighting conditions or with different camera settings.

5. Conclusions

The proposed solution produces results very close to the ideal scenario on various datasets and much better than any of the input algorithms, with an average score of around 90.3–90.8% of the maximum possible score and an MSE of around 0.7–0.8% when using nested cross-validation. When refitting the models on the entire dataset, the results improved even further, with the best model reaching an average score of 99.99% of the maximum possible score and an average processing time of just 0.22–0.25 s.

Future improvements could include using FM as the optimization metric, which might boost the average score and MSE across dataset folds, although this was not feasible in the current implementation due to the limitations of the ML.NET framework. The overall accuracy could be enhanced by applying the algorithm locally around each pixel, resulting in a local thresholding algorithm, or by applying the model on image patches, resulting in a mixed global–local thresholding algorithm. Experimenting with different neural network architectures specialized for regression instead of using AutoML might yield better results.

The proposed solution could be used as an estimator in a more complex approach. For example, Yang et al. [22] use Otsu’s method to generate an input noisy mask for a multi-branch coarse sub-network that predicts an output mask, which then feeds into a refinement sub-network to produce the final binarized image. The predicted ideal threshold could also be used as a feature in a local thresholding machine learning model.

Additional algorithms could be implemented into the overall solution to improve performance and adaptability to more complex tasks. For instance, incorporating local or mixed global–local thresholding techniques could improve accuracy in cases where fine-grained details are crucial. By expanding the algorithm to include more advanced techniques, the solution could become even more robust, offering greater flexibility for tasks such as image segmentation, edge detection, or other specialized applications.

Author Contributions: Conceptualization, N.T., C.-A.B. and M.-L.V.; data curation, C.-A.B.; formal analysis, N.T., C.-A.B. and M.-L.V.; investigation, N.T.; methodology, N.T. and M.-L.V.; project administration, C.-A.B.; resources, C.-A.B.; software, N.T.; supervision, C.-A.B.; validation, N.T., C.-A.B., and M.-L.V.; visualization, N.T., C.-A.B. and M.-L.V.; writing—original draft preparation, N.T.; writing—review and editing, N.T., C.-A.B. and M.-L.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BC	Sarle’s Bimodality Coefficient
GBC	Generalized BC
CFT	Cost Frugal Tuner

MSE	Mean Squared Error
RMSE	Root MSE
MAE	Mean Absolute Error
R ²	R-Squared
FM	F-Measure
FM _r	Relative FM
PDE	Partial Differential Equation
PSNR	Peak Signal-to-Noise Ratio
PSNR _r	Relative PSNR
MSE _r	Relative MSE
NCV	Nested Cross-Validation

References

1. Tahseen, A.J.A.; Sotnik, S.; Sinelnikova, T.; Lyashenko, V. Binarization Methods in Multimedia Systems when Recognizing License Plates of Cars. *Int. J. Acad. Eng. Res.* **2023**, *7*, 1–9.
2. Wang, S.; Fan, J. Image Thresholding Method Based on Tsallis Entropy Correlation. *Multimed. Tools Appl.* **2024**, *84*, 9749–9785. [\[CrossRef\]](#)
3. Elen, A.; Dönmez, E. Histogram-Based Global Thresholding Method for Image Binarization. *Optik* **2024**, *306*, 171814. [\[CrossRef\]](#)
4. Guo, J.; He, C. Adaptive Shock-Diffusion Model for Restoration of Degraded Document Images. *Appl. Math. Model.* **2019**, *79*, 555–565. [\[CrossRef\]](#)
5. Guo, J.; He, C.; Zhang, X. Nonlinear Edge-Preserving Diffusion with Adaptive Source for Document Images Binarization. *Appl. Math. Comput.* **2019**, *351*, 8–22. [\[CrossRef\]](#)
6. Mahani, Z.; Zahid, J.; Saoud, S.; Rhabi, M.E.; Hakim, A. Text Enhancement by PDE's Based Methods. In *Lecture Notes in Computer Science*; Springer Nature: Berlin/Heidelberg, Germany, 2012; pp. 65–76.
7. Bogiatzis, A.; Papadopoulos, B. Global Image Thresholding Adaptive Neuro-Fuzzy Inference System Trained with Fuzzy Inclusion and Entropy Measures. *Symmetry* **2019**, *11*, 286. [\[CrossRef\]](#)
8. Ashir, A.M. Multilevel Thresholding for Image Segmentation Using Mean Gradient. *J. Electr. Comput. Eng.* **2022**, *2022*, 1–9. [\[CrossRef\]](#)
9. Rani, U.; Kaur, A.; Josan, G. A New Binarization Method for Degraded Document Images. *Int. J. Inf. Technol.* **2019**, *15*, 1035–1053. [\[CrossRef\]](#)
10. Calvo-Zaragoza, J.; Gallego, A.-J. A Selectional Auto-Encoder Approach for Document Image Binarization. *Pattern Recognit.* **2018**, *86*, 37–47. [\[CrossRef\]](#)
11. Peng, X.; Wang, C.; Cao, H. Document Binarization via Multi-Resolutional Attention Model with DRD Loss. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 45–50.
12. He, S.; Schomaker, L. DeepOtsu: Document Enhancement and Binarization Using Iterative Deep Learning. *Pattern Recognit.* **2019**, *91*, 379–390. [\[CrossRef\]](#)
13. Mondal, R.; Chakraborty, D.; Chanda, B. Learning 2D Morphological Network for Old Document Image Binarization. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 65–70.
14. De, R.; Chakraborty, A.; Sarkar, R. Document Image Binarization Using Dual Discriminator Generative Adversarial Networks. *IEEE Signal Process. Lett.* **2020**, *27*, 1090–1094. [\[CrossRef\]](#)
15. Suh, S.; Kim, J.; Lukowicz, P.; Lee, Y.O. Two-Stage Generative Adversarial Networks for Binarization of Color Document Images. *Pattern Recognit.* **2022**, *130*, 108810. [\[CrossRef\]](#)
16. 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. [\[CrossRef\]](#)
17. Souibgui, M.A.; Kessentini, Y. DE-GAN: A Conditional Generative Adversarial Network for Document Enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 1180–1191. [\[CrossRef\]](#)
18. Yang, M.; Xu, S. A Novel Degraded Document Binarization Model through Vision Transformer Network. *Inf. Fusion* **2022**, *93*, 159–173. [\[CrossRef\]](#)
19. Yang, Z.; Liu, B.; Xiong, Y.; Yi, L.; Wu, G.; Tang, X.; Liu, Z.; Zhou, J.; Zhang, X. DocDiff: Document Enhancement via Residual Diffusion Models. In Proceedings of the 31st ACM International Conference on Multimedia (MM 2023), Ottawa, ON, Canada, 29 October–3 November 2023; pp. 2795–2806.

20. Souibgui, M.A.; Biswas, S.; Jemni, S.K.; Kessentini, Y.; Fornes, A.; Llados, J.; Pal, U. DoCENTR: An End-to-End Document Image Enhancement Transformer. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022. [\[CrossRef\]](#)
21. Biswas, R.; Roy, S.K.; Wang, N.; Pal, U.; Huang, G.B. DocBinFormer: A Two-Level Transformer Network for Effective Document Image Binarization. *arXiv* **2023**, arXiv:2312.03568. [\[CrossRef\]](#)
22. Yang, Z.; Liu, B.; Xiong, Y.; Wu, G. GDB: Gated Convolutions-Based Document Binarization. *Pattern Recognit.* **2024**, *146*, 109989. [\[CrossRef\]](#)
23. Siddiqui, F.U.; Yahya, A. *Clustering Techniques for Image Segmentation*; Springer Nature: Berlin/Heidelberg, Germany, 2021.
24. Sung, J.-M.; Kim, D.-C.; Choi, B.-Y.; Ha, Y.-H. Image Thresholding Using Standard Deviation. *Proc. SPIE Int. Soc. Opt. Eng.* **2014**, *9024*, 90240R. [\[CrossRef\]](#)
25. Li, C.H.; Lee, C.K. Minimum Cross Entropy Thresholding. *Pattern Recognit.* **1993**, *26*, 617–625. [\[CrossRef\]](#)
26. Sekertekin, A. A Survey on Global Thresholding Methods for Mapping Open Water Body Using Sentinel-2 Satellite Imagery and Normalized Difference Water Index. *Arch. Comput. Methods Eng.* **2020**, *28*, 1335–1347. [\[CrossRef\]](#)
27. Brink, A.D.; Pendock, N.E. Minimum Cross-Entropy Threshold Selection. *Pattern Recognit.* **1996**, *29*, 179–188. [\[CrossRef\]](#)
28. Knapp, T.R. Bimodality Revisited. *J. Mod. Appl. Stat. Methods* **2007**, *6*, 8–20. [\[CrossRef\]](#)
29. Tarbă, N.; Vencilă, M.-L.; Boiangiu, C.-A. On Generalizing Sarle’s Bimodality Coefficient as a Path towards a Newly Composite Bimodality Coefficient. *Mathematics* **2022**, *10*, 1042. [\[CrossRef\]](#)
30. Ahmed, Z.; Amizadeh, S.; Bilenko, M.; Carr, R.; Chin, W.-S.; Dekel, Y.; Dupre, X.; Eksarevskiy, V.; Filipi, S.; Finley, T.; et al. Machine Learning at Microsoft with ML.NET. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2448–2458.
31. Wu, Q.; Wang, C.; Huang, S. Frugal Optimization for Cost-Related Hyperparameters. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 10347–10354. [\[CrossRef\]](#)
32. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
33. Hughes, A.J.; Grawoig, D.E. *Statistics, a Foundation for Analysis*; Addison-Wesley Educational Publishers Inc.: Boston, MA, USA, 1971.
34. Berrar, D. Cross-Validation. In *Elsevier eBooks*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 542–545.
35. Bates, S.; Hastie, T.; Tibshirani, R. Cross-Validation: What Does It Estimate and How Well Does It Do It? *J. Am. Stat. Assoc.* **2023**, *119*, 1434–1445. [\[CrossRef\]](#)
36. Gatos, B.; Ntirogiannis, K.; Pratikakis, I. ICDAR 2009 Document Image Binarization Contest (DIBCO 2009). In Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009; pp. 1375–1382.
37. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. H-DIBCO 2010—Handwritten Document Image Binarization Competition. In Proceedings of the 2010 12th International Conference on Frontiers in Handwriting Recognition, Kolkata, India, 16–18 November 2010; pp. 727–732.
38. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. ICDAR 2011 Document Image Binarization Contest (DIBCO 2011). In Proceedings of the International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; pp. 1506–1510. [\[CrossRef\]](#)
39. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012). In Proceedings of the International Conference on Frontiers in Handwriting Recognition, Bari, Italy, 18–20 September 2012; pp. 817–822. [\[CrossRef\]](#)
40. Pratikakis, I.; Gatos, B.; Ntirogiannis, K. ICDAR 2013 Document Image Binarization Contest (DIBCO 2013). In Proceedings of the 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 1471–1476.
41. Ntirogiannis, K.; Gatos, B.; Pratikakis, I. ICFHR2014 Competition on Handwritten Document Image Binarization (H-DIBCO 2014). In Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition, Crete, Greece, 1–4 September 2014; pp. 809–813.
42. Pratikakis, I.; Zagoris, K.; Barlas, G.; Gatos, B. ICFHR2016 Handwritten Document Image Binarization Contest (H-DIBCO 2016). In Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 619–623.
43. Pratikakis, I.; Zagoris, K.; Barlas, G.; Gatos, B. ICDAR2017 Competition on Document Image Binarization (DIBCO 2017). In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–12 November 2017; pp. 1395–1403.
44. Pratikakis, I.; Zagori, K.; Kaddas, P.; Gatos, B. ICFHR 2018 Competition on Handwritten Document Image Binarization (H-DIBCO 2018). In Proceedings of the 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 5–8 August 2018; pp. 489–493.

45. Pratikakis, I.; Zagoris, K.; Karagiannis, X.; Tsochatzidis, L.; Mondal, T.; Marthot-Santaniello, I. ICDAR 2019 Competition on Document Image Binarization (DIBCO 2019). In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1547–1556.
46. Stathis, P.; Kavallieratou, E.; Papamarkos, N. An Evaluation Technique for Binarization Algorithms. *JUCS—J. Univers. Comput. Sci.* **2008**, *14*, 3011–3030.
47. CMATERdb. The Pattern Recognition Database Repository. Available online: <https://code.google.com/archive/p/cmaterdb/> (accessed on 22 May 2025).
48. Einsiedeln, Stiftsbibliothek, Codex 611(89), from 1314. Available online: <https://www.e-codices.unifr.ch/en/sbe/0611/> (accessed on 22 May 2025).
49. Salzinnes Antiphonal Manuscript (CDM-Hsmu M2149.14). Available online: <https://cantus.simssa.ca/manuscript/133/> (accessed on 22 May 2025).
50. Lins, R.D.; Torreão, G.; Silva, G.P.E. Content Recognition and Indexing in the LiveMemory Platform. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 220–230.
51. Lins, R.D. Two Decades of Document Processing in Latin America. *J. Univers. Comput. Sci.* **2011**, *17*, 151–161.
52. Castro-Bleda, M.J.; España-Boquera, S.; Pastor-Pellicer, J.; Zamora-Martínez, F. The NoisyOffice Database: A Corpus to Train Supervised Machine Learning Filters for Image Processing. *Comput. J.* **2019**, *63*, 1658–1667. [[CrossRef](#)]
53. Lazzara, G.; Géraud, T. Efficient Multiscale Sauvola’s Binarization. *Int. J. Doc. Anal. Recognit. (IJ DAR)* **2013**, *17*, 105–123. [[CrossRef](#)]
54. Burie, J.-C.; Coustaty, M.; Hadi, S.; Kesiman, M.W.A.; Ogier, J.-M.; Paulus, E.; Sok, K.; Sunarya, I.M.G.; Valy, D. ICFHR2016 Competition on the Analysis of Handwritten Text in Images of Balinese Palm Leaf Manuscripts. In Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 596–601.
55. Nafchi, H.Z.; Ayatollahi, S.M.; Moghaddam, R.F.; Cheriet, M. Persian Heritage Image Binarization Dataset (PHIBD 2012). Available online: http://tc11.cvc.uab.es/datasets/PHIBD2012_1 (accessed on 22 May 2025).
56. Singh, B.M.; Sharma, R.; Ghosh, D.; Mittal, A. Adaptive Binarization of Severely Degraded and Non-Uniformly Illuminated Documents. *Int. J. Doc. Anal. Recognit. (IJ DAR)* **2014**, *17*, 393–412. [[CrossRef](#)]
57. Hedjam, R.; Cheriet, M. Ground-Truth Estimation in Multispectral Representation Space: Application to Degraded Document Image Binarization. In Proceedings of the 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 190–194.
58. Maddock, J.; Kormanyos, C. Boost Multiprecision. Available online: https://www.boost.org/doc/libs/1_85_0/libs/multiprecision/doc/html/boost_multiprecision/intro.html (accessed on 22 May 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.