A Fair Evaluation of Various Deep Learning-Based Document Image Binarization Approaches

Richin Sukesh, Mathias Seuret(✉), Anguelos Nicolaou, Martin Mayr, and Vincent Christlein

Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany
{richin.sukesh,mathias.seuret}@fau.de

Abstract. Binarization of document images is an important preprocessing step in the field of document analysis. Traditional image binarization techniques usually rely on histograms or local statistics to identify a valid threshold to differentiate between different aspects of the image. Deep learning techniques are able to generate binarized versions of the images by learning context-dependent features that are less error-prone to degradation typically occurring in document images. In recent years, many deep learning-based methods have been developed for document binarization. But which one to choose? There have been no studies that compare these methods rigorously. Therefore, this work focuses on the evaluation of different deep learning-based methods under the same evaluation protocol. We evaluate them on different Document Image Binarization Contest (DIBCO) datasets and obtain very heterogeneous results. We show that the DE-GAN model was able to perform better compared to other models when evaluated on the DIBCO2013 dataset while DP-LinkNet performed best on the DIBCO2017 dataset. The 2-StageGAN performed best on the DIBCO2018 dataset while SauvolaNet outperformed the others on the DIBCO2019 challenge. Finally, we make the code, all models and evaluation publicly available (https://github.com/RichSu95/Document_Binarization_Collection) to ensure reproducibility and simplify future binarization evaluations.

Keywords: Binarization · Deep learning · Evaluation

1 Introduction

Image binarization is a process that converts a color or grayscale image into an image whose pixels can have only two different values, usually black and white. In the domain of document image analysis, binarization typically consists in separating the text (foreground) from its support (background), e. g., the paper. While it became less popular for text recognition, it remains an important preprocessing step in many other tasks, such as writer identification [4,5], word

© Springer Nature Switzerland AG 2022
S. Uchida et al. (Eds.): DAS 2022, LNCS 13237, pp. 771–785, 2022.
https://doi.org/10.1007/978-3-031-06555-2_52

spotting or optical character recognition (OCR) [10]. In traditional global binarization, the grayscale intensity frequency histogram of an image is analyzed and an appropriate threshold is set, e. g., Otsu's thresholding [17]. Alternatively, binarization is applied locally using statistics such as mean and standard deviation like the popular Sauvola method [27]. However, these methods have problems with ink bleed-through artifacts and other artifacts such as stains, blurring, faint characters and noise [15]. An error that may be generated through incorrect binarization may propagate forward and lead to performance reduction in subsequent tasks. Document binarization also acts as a means to filter out these undesirable features. A thorough overview of binarization techniques, datasets, and metrics is given in a survey by Tensmeyer and Martinez [31].

In recent years, rather than relying on traditional image binarization techniques, many studies have been conducted that employ deep learning models to binarize document images. The advent of deep learning has brought a multitude of changes to the domain of computer vision and image processing. Convolutional neural networks (CNNs) identify features automatically by learning from training data. The image features are discovered at multiple layers and are learned gradually from lower-levels to higher-levels. This multi-layered architecture performs a series of convolutions on the input image. A training process is implemented to adjust the parameters of the network to achieve the desired output.

In the past decade, there have been immense progress in the field of binarization of contemporary and historical documents using deep learning techniques. Although many approaches using deep learning for document binarization have been put forward, it is difficult to identify which among these models perform best when compared to one another. The root cause of this problem is the fact that most of these models have never been trained and tested on a common dataset using the same evaluation protocol. This paper aims to resolve this disparity by training and testing some well-known binarization models [2,8,10,13,28,29,32] on common datasets from the well-known Document Image Binarization Contests (DIBCO) [6,16,18–25]. While we evaluated the results of the models using four metrics, we omitted investigations on the relationship between result quality and processing time as Lins *et al.*. did [11]. Our evaluations draw a very heterogeneous picture. All four evaluation datasets have a different winner. Overall, DE-GAN ranks best across the four chosen DIBCO test datasets while metric-wise, the 2-Stage GAN outperforms the other models.

The following Sect. 2 of the paper provides a brief overview on the network architectures and methodologies used in the different binarization models that would be compared against one another. Section 3 gives a detailed description on the various datasets, validation metrics and on how all the models were trained. Section 4 shows the results of evaluating all models on the various test datasets and provides a brief discussion on the outcome of the experiments.

2 Overview of Evaluated Binarization Methods

2.1 Document Enhancement Generative Adversarial Network

The work presented by Souibgui *et al.* [28] models the document binarization problem as an image-to-image translation task. The Document Enhancement Generative Adversarial Network (DE-GAN) model basically consists of a generator and a discriminator. The generator follows a U-Net architecture [26] and its objective is to generate a clean image given the original degraded image. The goal of the discriminator is then to determine if the image shown is a fake image generated by the generator or the original binarized ground truth. An adversarial loss function is employed for training the model [28]:

$$\begin{aligned} L_{GAN}(\phi_G, \phi_D) = E_{I^W, I^{GT}} log[D_{\phi D}(I^W, I^{GT})] \\ + E_{I^W, I^{GT}} log[1 - D_{\phi D}(I^W, G_{\phi G}(I^W))], \end{aligned} \tag{1}$$

where $G_{\phi G}$ and $D_{\phi D}$ are the generator and discriminator functions respectively, I^W is the degraded image and I^{GT} is the ground truth. After a few epochs, the network is able to generate images similar to the ground truth. To maintain a good text quality and to improve training speed an additional log loss function is added. The objective is that the text output from the generator is identical to the ground truth text [28]:

$$L_{\log}(\phi G) = E_{I^{GT}, I^W}[-(I^{GT} \log(G_{\phi G}(I^W)) + ((1 - I^{GT}) \log(1 - G_{\phi G}(I^W)))]. \tag{2}$$

The overall loss of the network is denoted as [28]:

$$L_{\text{net}}(\phi G, \phi D) = \min_{\phi G} \max_{\phi D} L_{\text{GAN}}(\phi G, \phi D) + \lambda L_{\log}(\phi G), \tag{3}$$

where L_{GAN} is the adversarial loss function used to train the cGAN and λ is a hyper-parameter that is set to 500 for document binarization. The generator follows an encoder-decoder structure. The encoder performs down-sampling of the given input up to a certain layer and the decoder then up-samples the encoder output. The discriminator used is a simple Fully-Connected Network (FCN) with 6 convolutional layers. To train the DE-GAN model, overlapped patches of size 256×256 pixel are obtained from the degraded images and fed as input to the generator.

2.2 SauvolaNet

Inspired by the traditional Sauvola thresholding algorithm [27], the work by Li *et al.* [10] presents a deep learning approach to learn the Sauvola parameters, called the "SauvolaNet". The network aims to making the model computationally efficient. The model also comprises of an attention mechanism that aims to estimate the required Sauvola window sizes for each pixel location. One main drawback of the traditional Sauvola thresholding approach is that the algorithm achieves its

highest performance only when the right hyperparameters are manually tuned for each input image (window size, estimated level of document degradation and dynamic range of input image intensity). SauvolaNet uses three modules, the Multi-Window Sauvola (MWS), Pixelwise Window Attention (PWA), and Adaptive Sauvola Threshold (AST) to learn an auxiliary threshold estimation function.

The MWS module takes an image as input and uses the Sauvola algorithm to estimate the local thresholds for different window sizes. The PSA module also takes the same image as input to estimate the window sizes for each pixel location. The AST module then predicts the final threshold for each pixel location by fusing the thresholds of different windows from the MWS and weights from the PWA modules. The SauvolaNet function is modelled as [10]:

$$T = g_{SauvolaNet}(D), \tag{4}$$

where, T is the output, $g_{SauvolaNet}$ is the auxiliary threshold estimation function and D is the input image. The PWA uses instance normalization instead of batch normalization in order to avoid overfitting when training with a small dataset. When training the SauvolaNet, the input image D is normalized to values in the range (0,1) and a modified hinge loss was developed [10]:

$$\text{loss}[i,j] = \max(1 - \alpha \cdot (D[i,j] - T[i,j]) \cdot B[i,j], 0), \tag{5}$$

where B is the binarization ground truth with values -1 for foreground and $+1$ for the background. i and j are indices that specify the location of a pixel. α is a parameter to control the margin of the decision boundary and only the pixels close to the decision boundary are used in gradient-backpropagation.

2.3 Two-Stage GAN

The work presented by Suh *et al.* proposes a two-stage color document image binarization deep learning architecture using generative adversarial neural networks (GANs) [29]. The GAN architecture generally consists of two networks, i.e., the generator and the discriminator. For this model, the EfficientNet [30] was used as the generator on account of its efficiency in the domain of image classification. In the case of the discriminator, the discriminator network from the PatchGAN [9] was implemented.

The first part of the network consists of four color independent generators that are trained with the red, green, blue, and gray channels in order to generate an enhanced image by removing background color information. The resulting channel images and corresponding ground truths first concatenated and then fed to the discriminator network. The binarization in the first stage is performed using local predictions in small patches. In order to cater to regions with larger backgrounds, the second stage of the network performs global binarization with the resized original input image and local binarization using the first stage output. Except for the input image channels, the structure for the generators in the second stage is identical to that of the first stage. During training, the images

are divided into patches of 256 × 256 pixels resolution without scaling. When training GANs in general, it is common to observe an instability in loss function convergence [29]. To solve this issue, the Wasserstein GAN with penalty was used which implements the Wasserstein K-distance as the loss function. Further, instead of the typically used L1 loss, pixel-wise binary cross-entropy is defined as the additional loss term for the generator update.

2.4 Robin U-Net Model

The implementation by Mikhail Masyagin [13] presents the Robust Documentation Binarization (ROBIN) tool. ROBIN makes use of a simple U-Net model [26] to perform document binarization. The U-Net model was originally developed for the purpose of semantic segmentation of medical images. The U-Net architecture can be described as an encoder-decoder network. The input image is first fed into the encoder network, where multiple convolution blocks are applied followed by a maxpool downsampling layer. The idea here is to encode the input image into feature representations at multiple levels. The output from the encoder is then sent to the decoder where the activation map undergoes upsampling or deconvolution. Skip connections are also introduced between the encoder and decoder structure such that the deep and shallow features can be combined.

When training the model, the input images are split into patches of 128 × 128 px resolution. The learning rate was set to 0.0001 with the Adam optimizer. The training is trained using the dice coefficient loss and run for 250 epochs with an early stopping criteria.

2.5 DP-LinkNet

The DP-LinkNet is a segmentation model introduced by Xiong *et al.*. It makes use of the D-LinkNet [33] and LinkNet [3] models with a pre-trained encoder as the backbone.

The model consists of: 1) an encoder, 2) a hybrid dilated convolution module, 3) a spatial pyramid pooling (SPP) module, and 4) a decoder [32]. Firstly, the input image is fed to the encoder where the text stroke features are extracted. The series of convolutions and down-sampling occurring at the encoder causes a reduction in the resolution of the obtained feature map. To counter this effect, dilated convolutions are introduced into the model. Dilated convolutions help in exponentially increasing the size of the receptive field without affecting the spatial resolution. An issue that still persists here is the fact that the dilated convolution module may still find it difficult to identify objects of different sizes with a fixed-sized field-of-view. To counter this effect, the spatial pyramid pooling is employed. This helps to present the input feature maps at different scales. Lastly, the decoder performs transposed convolution. Skip connections between the decoder and encoder structure are present to combine the shallow-level and high-level features, helping to compensate any loss encountered by convolution and pooling operations. When training the model, the binary cross entropy and dice coefficient losses are used. The input images were split into patches of size

128×128 px. The adam optimizer was set with an initial learning rate of 2×10^{-4}. The model was trained for 500 epochs with an early stopping criteria to avoid overfitting.

2.6 Selectional Auto-Encoder

The work presented by Calvo-Zaragoza *et al.* [2] uses an auto-encoder network topology to perform an image-to-image processing task. Such a task results in higher computational efficiency since all pixels in the input image are processed at the same time. Generally, an auto-encoder network is trained to learn the identity function. However, in the selectional auto-encoder (SAE), the network is trained to learn a selectional map over a $w \times h$ image, preserving the input shape. The activation of each pixel depends on whether the pixel belongs to the foreground or the background. When training the SAE, the images along with their corresponding ground-truth (binarized image) are fed as input to the network. Auto-encoders are feed-forward networks and generally consist of two sections, i.e., the encoder and decoder. The encoder learns to extract the latent representation given an input image, downsampling the image until an intermediate representation is achieved. The output from the encoder is then upsampled and reconstructed to the original input image dimensions by the hidden layers of the decoder. The last layer consists of a set of neurons and a sigmoid activation layer which then gives an output prediction between the range of 0 and 1.

Since the binarized output image should consist of pixel values being 0 or 1 and not in between, a thresholding process is implemented to decide whether the certain pixel belongs to the background or foreground. The encoder and decoder both consisted of 5 layers each and the sampling operators were fixed at 2×2. Network weights were initialized using Xavier initialization [7]. Optimization is handled with stochastic gradient descent and a mini-batch size of 10. The initial learning rate is set to 0.001 and the network is trained for 200 epochs with an early stopping criteria kept in place.

2.7 DeepOtsu

The work presented by He *et al.* [8] proposes an iterative deep learning approach to obtain binarized images called the DeepOtsu model. However, unlike the aforementioned methods in this section, the deep learning network in this case aims to remove artifacts and generate a non-degraded version of the input image. The degraded input image $\mathbf{x}$ is modeled as:

$$\mathbf{x} = \mathbf{x}_{\mathrm{u}} + \mathbf{e}, \tag{6}$$

where $\mathbf{x}_{\mathrm{u}}$ is the latent uniform image and e is the degradation. The aim of the deep learning network is to ultimately obtain $\mathbf{x}_{\mathrm{u}}$.

The network was trained with images split into patches of size 256×256. The patches are first fed to the CNN model and the obtained output is then compared to the ground truth, which in this case should be representative of the uniform, clean version of the input image. To obtain this ground truth, the degraded input image is compared to the already available binarized images from the dataset. Then, the ground truth image is computed as the average pixel value with the same label within the image patch. Once the non-degraded, uniform version of the input image is obtained, the binarized version of the image can be easily obtained using Otsu thresholding [17]. The basic U-Net model [26] is used for learning the degradation. The down-sampling path of the network consisted of 5 convolutional layers with a 3×3 kernel size, followed by a leaky-ReLU activation [12] and 2×2 max pooling. The batch size was set to 8 and the learning rate set to 10^{-4}.

3 Materials and Methods

3.1 Datasets

All models mentioned in the previous section are trained and tested on document images from the DIBCO dataset. To keep the comparison between the models fair and precise, the training set and validation set remain the same for all models. The training set consists of the DIBCO2009, DIBCO2010, DIBCO2011, DIBCO2012, DIBCO2014, and DIBCO2016 datasets. The models are evaluated on DIBCO2013, DIBCO2017, DIBCO2018, and DIBCO2019 datasets. The four test sets were chosen based on the unique properties present in the three sets. DIBCO2013 consists of both handwritten and printed documents. The images from DIBCO2017 had more textual content in them. The DIBCO2018 dataset consisted of images of textual content present towards the borders or corners of the papers and higher intensity of bleed-through artifacts. The DIBCO2019 dataset had large variations in the types of images. Note that we used only track A since track B, containing text content on papyri, are not present in any training data which lead to rather poor learning-based results. Evaluations based on these four datasets give an idea of how well the models are able to generalize on different types of unseen images. Figure 1 shows some samples of images that belong to the DIBCO datasets used for validating the models.

3.2 Metrics

Our evaluation of the various models is based on the standard evaluation metrics used in the DIBCO challenges: (1) F-measure (FM), (2) pseudo F-measure (pFM), (3) peak signal to noise ratio (PSNR), and (4) distance reciprocal distortion (DRD). The FM and pFM reach their best value at 1 and worst at 0 (Eqs. (7) and (8)). PSNR describes how close the binarized and ground truth images are (Eq. (9)). The higher the PSNR, the better is the binarized result.

(a) DIBCO2013 (b) DIBCO2017

(c) DIBCO2018 (d) DIBCO2019

Fig. 1. Image examples from the different DIBCO datasets used for testing the models.

The DRD is based on the reciprocal of distance, matching well to subjective evaluation by human visual perception (Eq. (10)).

$$\text{FM} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}, \tag{7}$$

where, Recall $= \frac{TP}{TP+FN}$ and Precision $= \frac{TP}{TP+FP}$. TP, FP and FN denote true positive, false positive and false negative values respectively.

$$\text{pFM} = \frac{2 \times \text{pRecall} \times \text{pPrecision}}{\text{pRecall} + \text{pPrecision}}, \tag{8}$$

where, pRecall and pPrecision, respectively the pseudo-recall and the pseudo-precision, are metrics weighted based on the distance to the contours of the foreground in the ground truth. For the pseudo-recall, pixels around strokes have weights starting from 1, and reaching 0 at a distance corresponding to the stroke's width, and pixels inside of the strokes have a weight of 1. For the pseudo-precision, pixels outside strokes but not further than the stroke's thickness have a weight of 1, and inside the stroke their weight increase toward the center, where they reach a value of 2.

$$\text{PSNR} = \log_{10}\left(\frac{C^2}{\text{MSE}}\right), \tag{9}$$

where, $\text{MSE} = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n}(L(x,y)-L'(x,y))^2}{mn}$. The terms m and n denote the dimensions of the image. C denotes the difference present between the text and background.

$$\text{DRD} = \frac{\sum_k \text{DRD}_k}{\text{NUBN}}, \tag{10}$$

where DRD_k is the distortion of the kth flipped pixel and NUBN is the number of non-uniform 8×8 blocks in the ground truth image.

3.3 Training

All models are trained on the DIBCO datasets as mentioned in the previous sections. Based on the configuration of the models, the degraded images along with the accompanying ground truths are first split into patches of size 256×256 pixel or 128×128 pixel resolutions. The patches are further augmented by random horizontal flipping, vertical flipping and rotations. The number of epochs for training each model is set based on the recommendation of the authors for each model, along with an early stopping criteria to monitor any possibility of overfitting the models. If the validation loss of the model does not show significant changes for 15 consecutive epochs, the training would stop and the model would be saved. Certain pre-processing and post-processing operations on the images exclusive to specific models have also been implemented. Such an example is the application of Otsu's thresholding on the output of the DeepOtsu method. The hyper-parameters for the models are optimized using the python library "optuna" [1].

4 Evaluation

The results of testing each model on the different test DIBCO datasets are as shown in the following tables. Table 1a shows the results of validating the models on the DIBCO2013 dataset. The DIBCO2013 dataset contains images that have a good representation of the training data, without any major artifacts or degradation present. All methods display comparable performance with the DE-GAN performing best. For reference, we also show the DIBCO winners of the respective challenge. Note that the participants of 2017 and later potentially used more data for training.

Table 1b shows the results of validating the models on the DIBCO2017 dataset. Here, the performance of the models start to fluctuate more when compared to Table 1a. This might be due to the fact that the DIBCO2017 dataset contains more images that have more densely packed textual content. The DP-LinkNet model outperforms the other models in terms of PSNR, FM and DRD whereas the DE-GAN model has a higher performance in terms of pFM. However, it can be observed that the DRD value for DE-GAN is quite high, indicating

(a) Input image (b) Ground Truth (c) DE-GAN

(d) Robin (U-Net) (e) DeepOtsu (f) 2-Stage GAN

(g) DP-LinkNet (h) SAE (i) SauvolaNet

Fig. 2. Illustration of some results for an image from DIBCO-2017. Pixels in cyan are false positives. The few pixels in orange are false negatives. Pixels in white or black match the ground truth. (Color figure online)

that the resulting binarized images have higher rate of distortions. This may be attributed to the training process of the DE-GAN model, which may have introduced distortions to the generated images. Qualitative results for a randomly chosen sample from DIBCO2017 can be seen in Fig. 2.

The results for the DIBCO2018 dataset is shown in Table 1c. The winner is clearly the 2-Stage GAN approach, outperforming all other methods in each metric. For the pFM and the DRD metrics, the DE-GAN ranks second. Interestingly, the DP-LinkNet struggles with black page borders, see Fig. 3b. While

Table 1. Results of different image binarization methods on the (a) DIBCO2013, (b) DIBCO2017, (c) DIBCO2018, and (d) DIBCO2019 datasets. Note that the winners of the respective DIBCO2017, DIBCO2018 and DIBCO2019 challenge had more data available.

Model	PSNR↑	FM↑	pFM↑	DRD↓
DE-GAN	**24.08**	**97.68**	**98.09**	**1.11**
Robin (U-Net)	22.81	95.07	95.82	1.99
DeepOtsu	21.19	93.46	95.99	2.25
2-Stage GAN	22.60	95.75	96.40	1.46
DP-LinkNet	23.63	96.49	97.24	1.10
SAE	20.88	93.35	94.44	3.17
SauvolaNet	23.41	96.31	97.53	1.28
Winner [21,24]	20.68	92.12	94.19	3.10

(a) DIBCO2013

PSNR↑	FM↑	pFM↑	DRD↓
18.31	96.23	**98.10**	3.22
19.99	92.05	94.06	2.23
18.02	89.01	91.84	3.50
20.89	95.56	96.54	1.33
22.84	**97.92**	97.94	**0.77**
16.73	87.59	90.41	5.60
19.40	93.33	96.26	2.20
18.28	91.04	92.86	3.40

(b) DIBCO2017

Model	PSNR↑	FM↑	pFM↑	DRD↓
DE-GAN	15.98	76.21	83.29	8.01
Robin (U-Net)	15.78	78.80	81.11	12.20
DeepOtsu	12.72	66.60	68.83	42.52
2-Stage GAN	**19.93**	**92.40**	**94.90**	**2.67**
DP-LinkNet	15.73	78.56	80.70	13.72
SAE	14.48	73.45	76.33	15.45
SauvolaNet	16.03	77.94	81.92	10.41
Winner [22,25]	19.11	88.34	90.24	4.92

(c) DIBCO2018

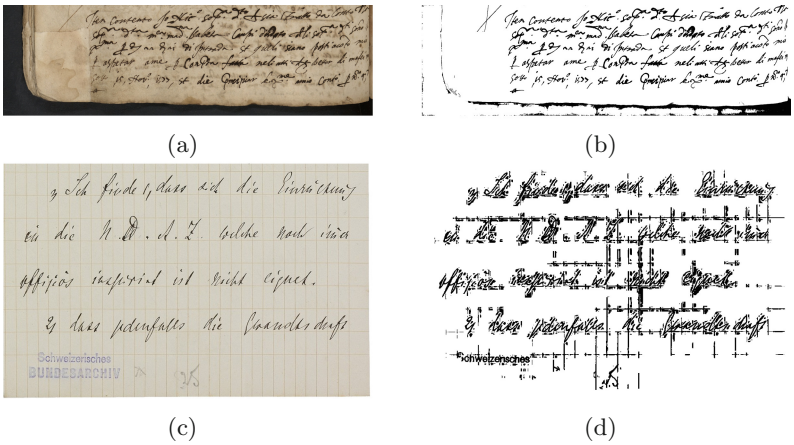
PSNR↑	FM↑	pFM↑	DRD↓
15.12	70.86	70.69	6.23
14.39	65.55	65.34	7.36
14.82	70.81	70.91	7.59
12.87	65.09	65.72	12.71
14.20	61.84	61.55	7.58
12.50	62.17	61.90	13.43
15.83	**72.04**	**71.59**	**5.55**
14.48	72.88	72.15	16.24

(d) DIBCO2019

it wins for the 2017 dataset that does not have borders, it performed poorly on images that have borders that are present in the DIBCO2018 dataset, cf. Fig. 1c.

While SauvolaNet ranks behind these two methods in the DIBCO2018 challenge, it outperforms both methods on the DIBCO2019 dataset, see Table 1d. The 2-Stage GAN, which performs very well for the 2013 to 2018 datasets had some difficulties to deal with the squared paper (check paper, quadrille paper) of the 2019 dataset, which can be observed in Fig. 3d. When we average all metrics for all different evaluated datasets, see Table 2a, the 2-Stage GAN seems to be on average the most suitable binarization method appearing to be consistent in terms of performance. Interestingly, computing the average rank over all metrics, i. e., the average over all 16 ranks for each method, it falls behind DE-GAN and SauvolaNet, cf. Table 2b.

We also evaluated the runtime, reported as throughput, i. e., images per second in the last column of Table 2a. The best throughput has the Robin binarization method. Note, however that we evaluated the methods on a small-sized



(a) (b) (c) (d)

Fig. 3. Qualitative examples of failure modes: (b) shows that DP-LinkNet binarizes the large black borders present in images of DIBCO2018 to white; (d) shows that the 2-Stage GAN struggles with the squared paper given in images of DIBCO2019, and additionally produces halo-artifacts.

Table 2. Average over (a) all metrics and (b) all ranks. Runtimes evaluated using an NVIDIA RTX 2060 GPU (12 GB RAM). Note that DeepOtsu and 2-Stage GAN were limited by the available memory.

Model	PSNR↑	FM↑	pFM↑	DRD↓	img/sec↑
DE-GAN	18.37	85.25	87.54	4.64	0.67
Robin (U-Net)	18.24	82.87	84.08	5.95	1.99
DeepOtsu	16.69	79.97	81.89	13.96	0.01
2-Stage GAN	19.07	**87.20**	**88.39**	**4.54**	0.01
DP-LinkNet	**19.10**	83.70	84.36	5.79	0.49
SAE	16.15	79.14	80.77	9.41	0.68
SauvolaNet	18.67	84.91	86.83	4.86	0.37

(a) Average metrics

Avg. rank↓
2.44
4.19
5.50
3.25
3.38
6.63
2.63

(b) Average ranks

GPU (NVIDIA RTX 2060) with 12 GB GPU-RAM. Unfortunately, this affected the throughput of DeepOtsu and 2-Stage GAN because multiple images of the DIBCO 2013 dataset contain very large images, e. g., image sizes of 4161×1049.

5 Conclusion

In this paper, we thoroughly evaluated seven deep learning-based methods in a fair evaluation where we fixed the data and augmentation used. We evaluated the methods using all ten available DIBCO datasets. Therefore, we used six datasets for training and the remaining four datasets for testing. Our evaluations show

that the results are very diverse on the four different tested datasets and no clear winner could be established. Overall, the DE-GAN approach achieved the best rank averaged over all four different datasets followed by SauvolaNet. When we compare the metrics individually, then the 2-Stage GAN approach performed best followed by the DE-GAN. In the very different DIBCO2019 dataset, however, the SauvolaNet outperformed these methods.

For future work, we would like to evaluate the methods also with a different protocol. In particular, we would like to simulate the DIBCO scenario of each year's challenge to be comparable with the single DIBCO papers, i. e., training with the datasets 2015–2016, then evaluating with 2017, adding 2017 to the training set, re-train and evaluate on 2018, and so on. The use of additional augmentation techniques as well as additional training datasets is also worth investigating and might have huge impact on the overall performance of the binarization methods. Furthermore, pixel-based evaluation is not optimal [31]. While the pFM metric incorporates the distance to the script contour, it might be worth investigating indirect measures, such as OCR/HTR accuracy or purely skeleton-based metrics [14]. From a practical point of view, the inference time is also worth investigating. This has been mainly studied in the competitions on time-quality document image binarization.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, pp. 2623–2631. Association for Computing Machinery, New York, NY, USA (2019)
2. Calvo-Zaragoza, J., Gallego, A.J.: A selectional auto-encoder approach for document image binarization. Pattern Recogn. **86**, 37–47 (2019)
3. Chaurasia, A., Culurciello, E.: LinkNet: exploiting encoder representations for efficient semantic segmentation. In: 2017 IEEE Visual Communications and Image Processing (VCIP), pp. 1–4 (2017)
4. Christlein, V., Bernecker, D., Hönig, F., Maier, A., Angelopoulou, E.: Writer identification using GMM supervectors and exemplar-SVMs. Pattern Recogn. **63**, 258–267 (2017)
5. Christlein, V., Gropp, M., Fiel, S., Maier, A.: Unsupervised feature learning for writer identification and writer retrieval. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 01, pp. 991–997 (2017)
6. Gatos, B., Ntirogiannis, K., Pratikakis, I.: ICDAR 2009 document image binarization contest (DIBCO 2009). In: 2009 10th International Conference on Document Analysis and Recognition, pp. 1375–1382 (2009)
7. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, 13–15 May 2010, vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy (2010)
8. He, S., Schomaker, L.: DeepOtsu: document enhancement and binarization using iterative deep learning. Pattern Recogn. **91**, 379–390 (2019)

9. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5967–5976 (2017)
10. Li, D., Wu, Y., Zhou, Y.: `SauvolaNet`: learning adaptive Sauvola network for degraded document binarization. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12824, pp. 538–553. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86337-1_36
11. Lins, R.D., Bernardino, R.B., Smith, E.B., Kavallieratou, E.: ICDAR 2021 competition on time-quality document image binarization. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12824, pp. 708–722. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86337-1_47
12. Maas, A.L.: Rectifier nonlinearities improve neural network acoustic models (2013)
13. Masyagin, M.: Robust document image binarization. https://github.com/masyagin1998/robin. Accessed 1 Apr 2022
14. Monteiro Silva, A.C., Hirata, N.S.T., Jiang, X.: Skeletal similarity based structural performance evaluation for document binarization. In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 37–42 (2020)
15. Mustafa, W.A., Kader, M.M.M.A.: Binarization of document images: a comprehensive review. J. Phys.: Conf. Ser. **1019**, 012023 (2018)
16. Ntirogiannis, K., Gatos, B., Pratikakis, I.: ICFHR 2014 competition on handwritten document image binarization (H-DIBCO 2014). In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 809–813 (2014)
17. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
18. Pratikakis, I., Gatos, B., Ntirogiannis, K.: H-DIBCO 2010 - handwritten document image binarization competition. In: 2010 12th International Conference on Frontiers in Handwriting Recognition, pp. 727–732 (2010)
19. Pratikakis, I., Gatos, B., Ntirogiannis, K.: ICDAR 2011 document image binarization contest (DIBCO 2011). In: 2011 International Conference on Document Analysis and Recognition, pp. 1506–1510 (2011)
20. Pratikakis, I., Gatos, B., Ntirogiannis, K.: ICFHR 2012 competition on handwritten document image binarization (H-DIBCO 2012). In: 2012 International Conference on Frontiers in Handwriting Recognition, pp. 817–822 (2012)
21. Pratikakis, I., Gatos, B., Ntirogiannis, K.: ICDAR 2013 document image binarization contest (DIBCO 2013). In: 2013 12th International Conference on Document Analysis and Recognition, pp. 1471–1476 (2013)
22. Pratikakis, I., Zagori, K., Kaddas, P., Gatos, B.: ICFHR 2018 competition on handwritten document image binarization (H-DIBCO 2018). In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 489–493 (2018)
23. Pratikakis, I., Zagoris, K., Barlas, G., Gatos, B.: ICFHR 2016 handwritten document image binarization contest (H-DIBCO 2016). In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 619–623 (2016)
24. Pratikakis, I., Zagoris, K., Barlas, G., Gatos, B.: ICDAR 2017 competition on document image binarization (DIBCO 2017). In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 01, pp. 1395–1403 (2017)
25. Pratikakis, I., Zagoris, K., Karagiannis, X., Tsochatzidis, L., Mondal, T., Marthot-Santaniello, I.: ICDAR 2019 competition on document image binarization (DIBCO 2019). In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1547–1556 (2019)

26. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
27. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recogn. **33**(2), 225–236 (2000)
28. Souibgui, M.A., Kessentini, Y.: DE-GAN: a conditional generative adversarial network for document enhancement. IEEE Trans. Pattern Anal. Mach. Intell. **44**(3), 1180–1191 (2022)
29. Suh, S., Kim, J., Lukowicz, P., Lee, Y.O.: Two-stage generative adversarial networks for document image binarization with color noise and background removal. CoRR abs/2010.10103 (2020). https://arxiv.org/abs/2010.10103
30. Tan, M., Le, Q.: EfficientNet: rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research, 09–15 June 2019, vol. 97, pp. 6105–6114. PMLR (2019)
31. Tensmeyer, C., Martinez, T.: Historical document image binarization: a review. SN Comput. Sci. **1**(3), 1–26 (2020). https://doi.org/10.1007/s42979-020-00176-1
32. Xiong, W., Jia, X., Yang, D., Ai, M., et al.: DP-LinkNet: a convolutional network for historical document image binarization. KSII Trans. Internet Inf. Syst. **15**(5), 1778–1797 (2021)
33. Zhou, L., Zhang, C., Wu, M.: D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 192–1924 (2018)