A selectional auto-encoder approach for document image binarization

Article *in* Pattern Recognition · June 2017

DOI: 10.1016/j.patcog.2018.08.011

2 authors:

Jorge Calvo-Zaragoza
University of Alicante

Antonio-Javier Gallego
University of Alicante

A selectional auto-encoder approach for document image binarization

Jorge Calvo-Zaragoza[a,*], Antonio-Javier Gallego[a]

[a] *Department of Software and Computing Systems, University of Alicante, Carretera San Vicente del Raspeig s/n, 03690 Alicante, Spain*

Abstract

Binarization plays a key role in the automatic information retrieval from document images. This process is usually performed in the first stages of documents analysis systems, and serves as a basis for subsequent steps. Hence it has to be robust in order to allow the full analysis workflow to be successful. Several methods for document image binarization have been proposed so far, most of which are based on hand-crafted image processing strategies. Recently, Convolutional Neural Networks have shown an amazing performance in many disparate duties related to computer vision. In this paper we discuss the use of a convolutional auto-encoder devoted to learning an end-to-end map from an input image of a fixed size to its selectional output, in which activations indicate whether the pixel must be classified as foreground or background. Once trained, documents can therefore be binarized by parsing them through the model and applying a threshold. Our approach has proven to outperform state-of-the-art techniques in the well-known DIBCO dataset (edition 2016).

Keywords: Binarization, Document Analysis, Auto-encoders, Convolutional Neural Networks

*Corresponding author: Tel.: +346-66-837771

Email addresses: `jcalvo@dlsi.ua.es` (Jorge Calvo-Zaragoza), `ajgallego@dlsi.ua.es` (Antonio-Javier Gallego)

1. Introduction

Image binarization consists in assigning a binary value to every single pixel of the image. Within the context of document analysis systems, the main objective is to distinguish the *foreground* (meaningful information) from the *background*.

Binarization plays a key role in the workflow of most document analysis and recognition systems. It not only helps to reduce the complexity of the task but it is also advisable for procedures involving morphological operations, detection of connected components, or histogram analysis, among others. Many methods have been proposed to accomplish this task. However, it is often complex to attain good results because documents may contain several difficulties—such as irregular leveling, blots, bleed-through, and so on—that may cause the process to fail.

In addition to all these disadvantages, it is convenient to emphasize that it is very difficult for the same method to work successfully in a number of document styles, since the set of potential domains is very heterogeneous. In order to deal with this situation, we propose a framework with which to binarize image documents based on machine learning. That is, a ground-truth of examples is used to train a model to perform the binarization task. This allows using the same approach in a wide range of documents, as long as training data is available.

Specifically, we make use of Convolutional Neural Networks (CNNs). These networks involve multi-layer architectures that perform a series of transformations to the input signal. The parameters of these transformations are adjusted through a training process. CNNs have drastically improved the state of the art in image, video, speech and audio recognition tasks [1]. Thus, its use in a task such as binarization of documents is promising. In this case, our proposal is to consider an image-to-image convolutional architecture, which is trained to convert an input image into its binarized version. Our experiments using the well-known benchmark Document Image Binarization Contest (DIBCO) show that this approach outperforms those of the state of the art, with a competitive computational cost.

The remainder of the paper is structured as follows: Section 2 presents related works to image document binarization; Section 3 presents our binarization framework based on convolutional models; Section 4 contains the experimentation performed and the results obtained; and finally, the current work concludes in Section 5.

2. Background

The most straightforward procedure for image binarization is to resort to simple thresholding, in which all pixels under a certain value are set to 0, and those above to 1. The threshold, however, does not need to be fixed by hand, but algorithms such as Otsu's [2] can compute a different value according to the input image.

However, as the complexity of the image to process increases, this simple procedure usually leads to poor or irregular binarizations, and so it is preferable to resort to other kind of approaches. Instead of computing a single global threshold, different threshold values might be obtained for every pixel by taking into account the features of its local neighborhood (defined by a window centered around the pixel). Niblack's algorithm [3] was one of the first binarization processes following this approach. Let m and s be the mean and standard deviation, respectively, of the considered neighboring region, it computes the threshold as $T = m + k * s$, where k is a parameter to tune. Extensions to this approach are Sauvola's [4], Wolf & Jolion [5] or Feng's [6] methods, which try to boost the accuracy of the binarization by considering more complex equations for the adaptive threshold.

There are also other kind of procedures for document image binarization. Gatos method [7] is an adaptive procedure that follows several steps, namely a low-pass Wiener filter, estimation of foreground and background regions, and a thresholding. It ultimately applies a post-processing step to improve the quality of foreground regions and preserve stroke connectivity. The method proposed by Su et al. [8] considers an adaptive image contrast as a combination of the local image contrast and the local image gradient, which makes it more robust against document degradations. The contrast map is then combined with an edge detector to identify the boundaries of foreground elements. Then, the document is finally binarized by using a local threshold based on the values of those boundaries.

Obviously, aforementioned methods have been successful in their specific contexts. However, it is difficult for these strategies to generalize adequately to any type of document. Therefore, it is interesting to resort to binarization procedures that can be learned, in order to apply such strategies on the largest possible set of domains as long as training data is available. That is, our idea is to develop algorithms that are able to binarize documents provided they have been trained for it.

In this regard, supervised learning techniques have been also considered for binarization tasks. The classification-based approach typically consists in querying every single pixel of the image, performing a feature extraction out of it and using a supervised learning algorithm to output a hypothesis about the two possible categories. The most common choice is to consider Multi-Layer Perceptron [9, 10, 11]. Within this paradigm, it is obvious the consideration of deep neural networks—especially convolutional models—given their striking impact on the machine learning and computer vision community. Some insights on the use of CNN for document image binarization can be found in the work of Pastor-Pellicer et al. [12]. The use of CNNs has also proven to be robust when dealing with document images that do not necessarily depict text information [13].

Although these works have shown acceptable performance, this pixel-wise approach has two important drawbacks. The first is the high computational cost that requires labeling each pixel of the image, as it involves making as many predictions as pixels of the image. This is an important issue since binarization is usually a pre-processing step of a larger workflow, so that complexity represents a bottleneck for the whole process. Secondly, each pixel is classified independently, without taking into account information about the labels assigned to their neighbors. That is, contextual information is somehow wasted.

The pixel-wise approach, however, serves as an inspiration to go on developing learning-driven binarization methods. In this work we present a convolutional approach that tries to alleviate the aforementioned drawbacks. We propose the use of fully-convolutional models that are trained to learn a patch-wise mapping of the image to its corresponding binarization. We do not propose to group pixels to which assign the same label, but to perform a fine-grain mapping in which each pixel of the output map gets a different activation value depending on whether the pixel must be labeled as a background or as a foreground. This approach is thoroughly described in the next section.

3. Selectional auto-encoder for document image binarization

From a machine learning point of view, image binarization can be formulated as a two-class classification task at pixel level. Our strategy follows this idea and, therefore, basically consists in learning which label must be given to every single pixel of the image. Since we are dealing with images of documents, we define the set of labels as *foreground* and *background*. As mentioned above, the straightforward way to implement this approach is to decide which of these labels must be assigned to a pixel one by one. Several works have studied this possibility, even taking into account the recent advances in deep learning.

In this paper we go a step further, proposing an approach that uses a network topology that works in an end-to-end fashion. That is, for each input image, the model is devoted to binarizing it in just one step. This has a number of advantages such that, in this way, the classification of each pixel of the image is not produced independently, but also takes into account the label to be assigned to its neighbors. In addition, several pixels can be processed at the same time, thereby leading to higher efficiency.

Our approach is inspired by classical adaptive algorithms but inversely. In those traditional algorithms, the pixel value is fixed (its value in grayscale), whereas a different threshold is calculated depending on a local context. What we do is to compute a different value for each pixel (neural activation) and to fix a global threshold. Obviously, this activation is not obtained following a hand-crafted equation, which may serve well for a specific domain, but is learned through convolutional auto-encoders.

Auto-encoders consist of feed-forward neural networks for which the input and output shape is exactly the same [14]. The network typically consists of two stages that learn the functions f and g, which are called encoder and decoder functions, respectively, that receive an input vector x. In their traditional formulation, the network must minimize a divergence $L(x, g(f(x)))$. The hidden layers of the encoder perform a mapping of the input —usually decreasing its dimension— until an intermediate representation is attained. The same input is then subsequently recovered by means of the hidden layers of the decoder function.

In their original formulation, auto-encoders are trained to learn the identity function, which might be useful as regards feature learning or dimensionality reduction because the encoder function provides a meaningful, compact representation of the input [15]. In our work, however, we use an auto-encoder topology for a different purpose.

Here the model is trained to perform a function such that $b : \mathbb{R}^{(w \times h)} \rightarrow [0, 1]^{(w \times h)}$. In other words, it learns a selectional map over a $w \times h$ image that preserves the input shape. The selectional value (activation) of each pixel depends on whether the pixel belongs to the foreground or to the background. Given the nature of the output, we call this model *Selectional Auto-Encoder* (SAE).

The hierarchy of layers of our SAE consists of a series of convolutional plus pooling layers, until reaching an intermediate layer in which compact representations of the input are attained (function f). As the pooling layers are applied, filters are able to relate parts of the image that were initially far apart. It then follows a series of convolutional plus upsampling layers that reconstruct the image up to its initial size (function g). The last layer consists of a set of neurons that predict a value in the range of $[0, 1]$, depending on the selectional level of the corresponding input pixel. Note that the approach is completely convolutional in the sense that the prediction is done through successive convolutions and pooling / upsampling operations, without any fully-connected layer. In other words, the approach just applies successive transformations to the input image. These transformations are learned through a training process, rather than being pre-established, with the aim at binarizing the document image. A graphical illustration of this configuration is shown in Fig. 1

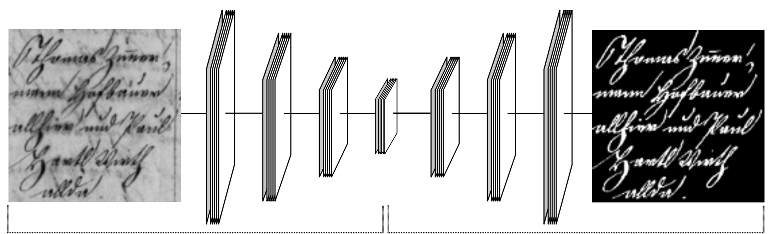

Figure 1: General overview of an SAE used for document image binarization (image taken from [16]). The output layer consists of the activation level assigned to each input feature (white denotes maximum activation)

Thus, once the SAE has been properly trained an image can be parsed through the network, after which a selection level is assigned to each input pixel. In practice, the network hardly outputs either 0 or 1 but an intermediate value. Therefore, a thresholding process is still necessary to convert the obtained scores into actual binary values. Those pixels whose selection value exceeds a certain threshold are considered to belong to the foreground of the document image, whereas the others are labeled as background.

3.1. Implementation details

There are a number of parameters to establish an SAE, such as the depth of the encoding and decoding functions or the size of the input and output layers. In addition, it is also necessary to establish the number and size of the convolutional filters, the size of the pooling and up-sampling layers, and the activation functions. Preliminary experiments determined a suitable topology to be considered in this work, which is described in Table 1. It consists of 3 encoding layers of 3×3 convolutions with 120 filters and Rectifier Linear Unit activations, followed by 2×2 max-pooling operations. The decoding function replicates the encoding stage but replacing pooling by up-sampling. Batch normalization and dropout units of 20 % are included in each convolutional block.

Input	Encoding	Decoding	Output
	Conv(120,3,3)	Conv(120,3,3)	
	BNorm()	BNorm()	
	Actv(ReLU)	Actv(ReLU)	
	MaxPool(2,2)	UpSamp(2,2)	
	Dropout(0.2)	Dropout(0.2)	
	Conv(120,3,3)	Conv(120,3,3)	
	BNorm()	BNorm()	
$[0, 255]^{256\times256}$	Actv(ReLU)	Actv(ReLU)	$[0, 1]^{256\times256}$
	MaxPool(2,2)	UpSamp(2,2)	
	Dropout(0.2)	Dropout(0.2)	
	Conv(120,3,3)	Conv(120,3,3)	
	BNorm()	BNorm()	
	Actv(ReLU)	Actv(ReLU)	
	MaxPool(2,2)	UpSamp(2,2)	
	Dropout(0.2)	Dropout(0.2)	
		Conv(1,3,3)	
		Actv(Sigmoid)	

Table 1: Detailed description of the selected SAE architecture. Conv(f,h,w) stands for a convolution operator of f filters, with $h \times w$ pixel kernels; MaxPool(h,w) stands for the max-pooling operator with a $w \times h$ kernel; UpSamp(h,w) denotes an up-sampling operator of h rows and w columns; BNorm() refers to batch normalization; Dropout(r) denotes a dropout operation that considers a ratio of r neurons at each time; $ReLU$ and *Sigmoid* denote Rectifier Linear Unit and Sigmoid activations, respectively.

It should be emphasized that we are not claiming that this SAE topology is optimal for document image binarization. It is just a model that allows us to show that our approach represents an elegant, effective, and efficient strategy for the problem at issue. Note that the model accepts fixed-size images of 256×256 but document images can be larger, and also variable in size. These cases can be easily processed by dividing the input images into equal pieces of those dimensions, and then combining the independent outputs provided by the

SAE.

The training stage consists of providing the SAE with examples of patches and their corresponding ground-truth. Since an SAE is a type of feed-forward network, the optimization is carried out by means of stochastic gradient descent [17], with a mini-batch size of 10, and considering the adaptive learning rate proposed by Zeiler [18]. Since document binarization typically represents an imbalanced problem (more background than foreground), we make use of the loss function proposed by Pastor-Pellicer et al. [19], which focuses on minimizing the F-measure. In order to increase the number of training samples, the patches of the training documents are extracted with a stride of 128 (*ie.*, 50 % of overlapping). In addition, a slight data augmentation procedure is considered so that new synthetic documents are generated by doing random scaling and flips over the original ones.

To boost the performance we also consider a pre-processing of the input images as follows. The image is inverted so that higher greyscale values indicate a higher *a priori* probability of being classified as foreground. After this, the mean value of the training images is computed and subtracted to each pixel, keeping 0 when negative. Finally, a min-max filter is applied so as to make the pixel values be closer to either the maximum or the minimum.

Consequently, document binarization is performed by applying this pre-processing, parsing the input image through the SAE, and applying the thresholding. A graphical example is illustrated in Fig. 2.

(a) Input image (b) Pre-processing

(c) Activation (d) Thresholding

Figure 2: Example of binarization task (from [16]) using an SAE. After pre-processing the input image, an activation is obtained for each pixel of the input image (illustrated here with a heat map). Finally, a thresholding is applied in order to binarize the image.

4. Experiments

This section details the experimentation carried out to evaluate our proposal. The performance of a binarization algorithm can be evaluated in several ways. For instance, if the algorithm is part of a workflow to perform a particular task, an interesting way to measure the performance is in relation to the full performance. However, this implies that the evaluation of the algorithm may not be totally fair, since it is strongly related to the performance of the rest of the stages of that workflow.

In our case, to perform an objective comparison with other state-of-the-art algorithms, we follow the guidelines of the Document Image Binarization

Contest (DIBCO) competition. This competition has been held several ways from 2009 [20] to 2016 [16]. In each edition, this contest provides a series of images that must be binarized, and the algorithms are evaluated based on a ground truth created in a supervised way.

In this work, the evaluation is carried out following the most recent DIBCO competition (2016)[1]. Therefore, the set of images of such edition will be our test set. However, given that our approach needs training data, we use all the images from previous editions (2009-15) as training set. Therefore, the complete corpora comprises 76 images for training, from which a total of 24 881 patches are extracted considering data augmentation, and 10 images for testing.

As regards performance evaluation we consider the same metrics of the selected edition of DIBCO, which consists of:

F-measure (FM) is a common performance metric for two-class classification problems. It is computed as

$$\mathrm{FM} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}},$$

where TP, FP and FN stand for true positives (foreground pixels classified as foreground), false positives (background pixels classified as foreground), and false negatives (background pixels classified as foreground), respectively.

pseudo-FMeasure (F_{ps}) is computed similarly to the FM but weighing the relevance of each pixel depending whether it is close to stroke boundaries.

Peak signal-to-noise ratio (PSNR) is a pixel-wise term for measuring the similarity between the ground-truth image Y and the predicted by the binarization algorithm H. It is computed as

$$\mathrm{PSNR} = 10 \log \left(\frac{w\,h\,P^2}{\sum_{x=1}^{w} \sum_{y=1}^{h} (Y(x,y) - H(x,y))^2} \right),$$

[1] Up to the date this document was written.

where w and h denote the width and the height, respectively, of the images, and P denotes the maximum possibledifference between two compared pixels.

Distance-reciprocal Distortion (DRD) was proposed by Lu et al. [21] to measure the difference between binary images, trying to approximate human visual perception.

4.1. Comparative assessment

In this section we compare the results obtained using our approach against the participants of the contest. The nature of the methods involved in the competition is quite diverse. Most of them are based on published approaches with specific modifications for participation. Therefore, reader may check the report of the contest itself [16] to know the detailed operation of each of them. Otsu's and Sauvola's methods are also included as baseline.

Table 2 presents the figures obtained as regards the aforementioned metrics of interest. The threshold needed for the SAE has been established to 0.1. In most cases, our method achieves a significant improvement in all metrics of interest. Comparing with the best result of the contest in each of them we can see, however, that the actual improvement is in the FM, staying very close in the rest of metrics. Moreover, the goodness of our work is that we are proposing an efficient document image binarization method based on machine learning that is competitive with the state of the art.

5. Conclusions

In this paper a new approach for document image binarization has been presented, which is based on machine learning and convolutional neural networks. Our strategy is to learn a model that allows an end-to-end transformation. Given a piece of image of a fixed size, it outputs an activation for each pixel depending on the confidence whether the pixel belongs to the foreground of

Method	FM (↑)	F_{ps} (↑)	PSNR (↑)	DRD (↓)
Otsu	86.81	88.67	17.80	5.56
Sauvola	82.52	86.85	16.42	7.49
1	85.57	91.05	17.50	5.00
2	87.61	91.28	18.11	5.21
3-1	88.22	91.42	18.22	4.01
3-2	88.47	91.71	18.29	3.93
3-3	<u>88.72</u>	<u>91.84</u>	<u>18.45</u>	<u>3.86</u>
4	76.28	77.99	14.21	15.14
5	86.24	90.84	17.52	5.25
6	87.97	91.57	18.00	4.49
7-1	87.60	90.87	17.86	4.51
7-2	88.11	91.17	18.00	4.38
8	84.32	85.64	16.59	6.94
9	76.10	79.60	15.35	9.16
SAE	**89.99**	**91.90**	**18.52**	**3.78**

Table 2: Performance comparison against the participants in the *DIBCO 2016* image document binarization contest. Symbols ↑ and ↓ indicate whether the performance metric is better to be high or low, respectively. Values in bold type highlight the best result, on average, in each considered metric. Underlined values indicate the best result, on average, in each metric considering only the participants of the contest.

the document. These activations are eventually thresholded to yield a discrete binary value.

The approach has been evaluated following the guidelines of the DIBCO competition, which can be considered a benchmark of this task. In particular, we have used the documents of the edition of 2016 as test set, while the rest of the documents of previous edition have been used as training set. The obtained results report that the SAE outperforms the rest of binarization algorithms in all the considered metrics. As an example, our approach manages to obtain an FM of 89.99, when the best contest participant achieved an 88.72.

As prospects for future work, the intention is twofold. On the one hand, we are interested in learning the task with a limited number of labeled samples. The idea is, therefore, to follow semi-supervised approaches in which the task can be learned in an unsupervised manner with the addition of a fine-tuning process with a small labeled corpus. On the other hand, it would be of great interest to carry out studies by increasing the variability of the input images. Documents can be highly heterogeneous, and we thus wish to obtain a model that can adapt to any type of them.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[2] N. Otsu, A threshold selection method from gray-level histograms, Automatica 11 (285-296) (1975) 23–27.

[3] W. Niblack, An introduction to digital image processing, Strandberg Publishing Company, 1985.

[4] J. Sauvola, M. Pietikäinen, Adaptive document image binarization, Pattern Recognition 33 (2) (2000) 225–236.

[5] C. Wolf, J. M. Jolion, F. Chassaing, Text localization, enhancement and

binarization in multimedia documents, in: Proceedings of the International Conference on Pattern Recognition, Vol. 2, 2002, pp. 1037–1040.

[6] M.-L. Feng, Y.-P. Tan, Contrast adaptive binarization of low quality document images, IEICE Electronics Express 1 (16) (2004) 501–506.

[7] B. Gatos, I. Pratikakis, S. J. Perantonis, Adaptive degraded document image binarization, Pattern Recognition 39 (3) (2006) 317–327.

[8] B. Su, S. Lu, C. L. Tan, Robust document image binarization technique for degraded document images, IEEE transactions on image processing 22 (4) (2013) 1408–1417.

[9] Z. Chi, K. W. Wong, A two-stage binarization approach for document images, in: Proceedings of the International Symposium on Intelligent Multimedia, Video and Speech Processing, IEEE, 2001, pp. 275–278.

[10] J. L. Hidalgo, S. España, M. J. Castro, J. A. Pérez, Enhancement and cleaning of handwritten data by using neural networks, in: Proceedings of the 2nd Iberian Conference on Pattern Recognition and Image Analysis, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 376–383.

[11] A. Kefali, T. Sari, H. Bahi, Foreground-background separation by feed forward neural networks in old manuscripts, Informatica 38 (4).

[12] J. Pastor-Pellicer, S. E. Boquera, F. Zamora-Martínez, M. Z. Afzal, M. J. C. Bleda, Insights on the use of convolutional neural networks for document image binarization, in: Advances in Computational Intelligence - 13th International Work-Conference on Artificial Neural Networks, IWANN 2015, Palma de Mallorca, Spain, June 10-12, 2015. Proceedings, Part II, 2015, pp. 115–126.

[13] J. Calvo-Zaragoza, G. Vigliensoni, I. Fujinaga, Pixel-wise binarization of musical documents with convolutional neural networks, in: Fifteenth IAPR International Conference on Machine Vision Applications, 2017.

[14] G. E. Hinton, R. S. Zemel, Autoencoders, minimum description length and helmholtz free energy, in: Advances in Neural Information Processing Systems, 1994, pp. 3–10.

[15] W. Wang, Y. Huang, Y. Wang, L. Wang, Generalized autoencoder: A neural network framework for dimensionality reduction, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2014, pp. 490–497.

[16] I. Pratikakis, K. Zagoris, G. Barlas, B. Gatos, ICFHR2016 handwritten document image binarization contest (H-DIBCO 2016), in: 15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016, 2016, pp. 619–623.

[17] L. Bottou, Large-scale machine learning with stochastic gradient descent, in: Proceedings of COMPSTAT'2010, Springer, 2010, pp. 177–186.

[18] M. D. Zeiler, ADADELTA: an adaptive learning rate method, CoRR abs/1212.5701.

[19] J. Pastor-Pellicer, F. Zamora-Martínez, S. España-Boquera, M. J. Castro-Bleda, F-measure as the error function to train neural networks, in: International Work-Conference on Artificial Neural Networks, Springer, 2013, pp. 376–384.

[20] B. Gatos, K. Ntirogiannis, I. Pratikakis, ICDAR 2009 document image binarization contest (DIBCO 2009), in: 10th International Conference on Document Analysis and Recognition, ICDAR 2009, Barcelona, Spain, 26-29 July 2009, 2009, pp. 1375–1382.

[21] H. Lu, A. C. Kot, Y. Q. Shi, Distance-reciprocal distortion measure for binary document images, IEEE Signal Processing Letters 11 (2) (2004) 228–231.