2017 14th IAPR International Conference on Document Analysis and Recognition

Convolutional Neural Networks for Page Segmentation of Historical Document Images

Kai Chen*, Mathias Seuret*, Jean Hennebert*†, and Rolf Ingold*
*DIVA, University of Fribourg, Switzerland, Email: {firstname.lastname}@unifr.ch
†University of Applied Sciences, HES-SO//FR, Fribourg, Switzerland, Email: jean.hennebert@hefr.ch

***Abstract*—This paper presents a page segmentation method for handwritten historical document images based on a Convolutional Neural Network (CNN). We consider page segmentation as a pixel labeling problem, i.e., each pixel is classified as one of the predefined classes. Traditional methods in this area rely on hand-crafted features carefully tuned considering prior knowledge. In contrast, we propose to learn features from raw image pixels using a CNN. While many researchers focus on developing deep CNN architectures to solve different problems, we train a simple CNN with only one convolution layer. We show that the simple architecture achieves competitive results against other deep architectures on different public datasets. Experiments also demonstrate the effectiveness and superiority of the proposed method compared to previous methods.**

***Keywords*-convolutional neural network; page segmentation; layout analysis; historical document images; deep learning;**

I. Introduction

Page segmentation is an important prerequisite step of document image analysis and understanding. The goal is to split a document image into regions of interest. Compared to segmentation of machine printed document images, page segmentation of historical document images is more challenging due to many variations such as layout structure, decoration, writing style, and degradation. Our goal is to develop a generic segmentation method for handwritten historical documents. In this method, we consider the segmentation problem as a pixel-labeling problem, i.e., for a given document image, each pixel is labeled as one of the predefined classes.

Some page segmentation methods have been developed recently. These methods rely on hand-crafted features [1], [2], [3], [4] or prior knowledge [5], [6], [7], or models that combine hand-crafted features with domain knowledge [8], [9]. In contrast, in this paper, our goal is to develop a more general method which automatically learns features from the pixels of document images. Elements such as strokes of words, words in sentences, sentences in paragraphs have a hierarchical structure from low to high levels. As these patterns are repeated in different parts of the documents. Based on these properties, feature learning algorithms can be applied to learn layout information of the document images.

Convolutional Neural Network (CNN) is a feed-forward artificial neural network which shares weights among neurons in the same layer. By enforcing local connectivity pattern between neurons of adjacent layers, CNN can discover spatial correlations at different granularity of local context [10]. With multiple convolutional layers and pooling layers, CNN has achieved many successes in various fields, e.g., handwriting recognition [11], image classification [12], and text recognition in natural images [13].

In [14], the authors show that an autoencoder can be used to learn features automatically on the training images. An autoencoder is a feed forward neural network trained to reconstruct its input. Hidden layers outputs are then used as features to feed an off-the-shelf classifier. In [15], the authors show that by using superpixels as units of labeling, the speed of the method is increased. In [16], a Conditional Random Field (CRF) [17] is applied in order to model the local and contextual information jointly to refine the segmentation results which have been achieved in [15]. Following the same idea of [16], we consider the segmentation problem as an image patch labeling problem. The image patches are generated by using superpixels algorithm. In contrast to [14], [15], [16], in this work, we focus on developing an end-to-end method. We combine feature learning and classifier training into one step. Image patches are used as input to train a CNN for the labeling task. During training, the features used to predict labels of the image patches are learned on the convolution layers of the CNN.

While many researchers focus on developing very deep CNN to solving various problems [12], [18], [19], we train a simple CNN of one convolution layer. Experiments on public historical document image datasets show that despite the simple structure and little tuning of hyperparameters, the proposed method achieves comparable results compared to other CNN architectures.

II. Methodology

In order to create general page segmentation method without using any prior knowledge of the layout structure of the documents, we consider the page segmentation problem as a pixel labeling problem. We propose to use a CNN for the pixel labeling task. The main idea is to learn a set of feature detectors and train a nonlinear classifier on the features extracted by the feature detectors. With the set of feature detectors and the classifier, pixels on the unseen document images can be classified into different classes.

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.161

A. Preprocessing

In order to speed up the pixel labeling process, for a given document image, we first apply a superpixel algorithm. A superpixel is an image patch which contains pixels belong to the same object. Then instead of labeling all the pixels, we only label the center pixel of each superpixel and the remaining pixels in that superpixel are assigned to the same label. The superiority of the superpixel labeling approach over the pixel labeling approach for the page segmentation task has been demonstrated in [15]. The simple linear iterative clustering (SLIC) algorithm [20] is applied as a preprocessing step to generate superpixels for given document images.

B. CNN Architecture

The architecture of our CNN is given in Figure 1. The structure can be summarized as $28 \times 28 \times 1 - 26 \times 26 \times 4 - 100 - M$, where M is the number of classes. The input is a grayscale image patch. The size of the image patch is 28×28 pixels. Our CNN architecture contains only one convolution layer which consists of 4 kernels. The size of each kernel is 3×3 pixels. Unlike other traditional CNN architecture, the pooling layer is not used in our architecture. Then one fully connected layer of 100 neurons follows the convolution layer. The last layer consists of a logistic regression with softmax which outputs an estimation of the probability of each class, such that

$$P(y = i|x, W_1, \cdots, W_M, b_1, \cdots, b_M) = \frac{e^{W_i x + b_i}}{\sum_{j=1}^{M} e^{W_j x + b_j}}, \quad (1)$$

where x is the output of the fully connected layer, W_i and b_i are the weights and biases of the i^{th} neuron in this layer, and M is the number of the classes. The predicted class $\hat{y}$ is the class which has the max probability, such that

$$\hat{y} = \arg\max_i P(y = i|x, W_1, \cdots, W_M, b_1, \cdots, b_M). \quad (2)$$

In the convolution and fully connected layers of the CNN, Rectified Linear Units (ReLUs) [21] are used as neurons. An ReLU is given as: $f(x) = \max(0, x)$, where x is the input of the neuron.

C. Training

To train the CNN, for each superpixel, we generate a patch which is centred on that superpixel. The patch is considered as the input of the network. The size of each patch is 28×28 pixels. The label of each patch is its center pixel's label. The patches of the training images are used to train the network.

In the CNN, the stride length is 1 and the weights are initialized by using Xavier initialization [22]. The cost function is defined as the cross-entropy loss, such that

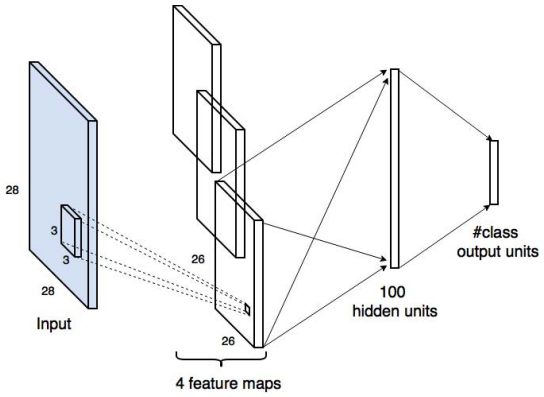

Figure 1: The architecture of the proposed CNN

$$\mathcal{L}(X, Y) = -\frac{1}{n} \sum_{i=1}^{n} (\ln a(x^{(i)}) + (1 - y^{(i)}) \ln(1 - a(x^{(i)}))), \quad (3)$$

where $X = \{x^{(1)}, \cdots, x^{(n)}\}$ is the set of training image patches and $Y = \{y^{(1)}, \cdots, y^{(n)}\}$ is the corresponding set of labels. The number of training image patches is n. For each $x^{(i)}$, $a(x^{(i)})$ is the output of the CNN as defined in Eq. 1. The CNN is trained with Stochastic Gradient Descent with the dropout [23] technique. The goal of dropout is to avoid overfitting by introducing random noise to training samples. Such that during the training, the outputs of the neurons are masked out with the probability of 0.5.

III. Experiment

Experiments are conducted on six public handwritten historical document image datasets.

A. Datasets

The datasets are of very different nature. The *G. Washington* dataset consists of the pages written in English with ink on paper and the images are in gray levels. The other two datasets, i.e., *Parzival* and *St. Gall* datasets consist of images of manuscripts written with ink on parchment and the images are in color. The *Parzival* dataset consits of the pages written by three writers in the 13th century. The *St. Gall* dataset contains the manuscripts from a medieval manuscript written in Latin. The details of the ground truth for both datasets are presented in [24].

Three new datasets with more complex layouts have been recently created [25]. The *CB55* dataset consists of manuscripts from the 14th century which are written in Italian and Latin languages by one writer. The *CSG18* and *CSG863* datasets consist of manuscripts from the 11th century which are written in Latin language. The number of writers of both datasets is not specified. The details of the three datasets are presented in [25].

In the experiments, all images are scaled down with a scaling factor 2^{-3}. Table I gives the details of training, test, and validation sets of the six datasets.

Table I: Details of training, test, and validation sets. TR, TE, and VA denote the training, test, and validation sets respectively.

	image size (pixels)	$\|TR\|$	$\|TE\|$	$\|VA\|$
G. Washington	2200×3400	10	5	4
St. Gall	1664×2496	20	30	10
Parzival	2000×3008	20	13	2
CB55	4872×6496	20	10	10
CSG18	3328×4992	20	10	10
CSG863	3328×4992	20	10	10

B. Metrics

The most used metrics for page segmentation of historical document images are precision, recall, and pixel level accuracy. Besides of these standard metrics, we also adapt the metrics which are well defined and has been widely used for common semantic segmentation and scene parsing evaluations to evaluate different page segmentation methods. These metrics have been proposed in [26]. They are based on pixel accuracy and region intersection over union (IU). Consequently, the metrics used in the experiments are: pixel accuracy, mean pixel accuracy, mean IU, and frequency weighted IU (f.w. IU).

In order to obtained the metrics, we define the variables:

- n_c: the number of classes.
- n_{ij}: the number of pixels of class i predicted to belong to class j. For class i:
  - n_{ii}: the number of correctly classified pixels (true positives).
  - n_{ij}: the number of wrongly classified pixels (false positives).
  - n_{ji}: the number of wrongly not classified pixels (false negatives).
- t_i: the total number of pixels in class i, such that

$$t_i = \sum_j n_{ji}. \tag{4}$$

With the defined variables, we can compute:

- pixel accuracy:

$$acc = \frac{\sum_i n_{ii}}{\sum_i t_i}. \tag{5}$$

- mean accuracy:

$$acc_{mean} = \frac{1}{n_c} \times \sum_i \frac{n_{ii}}{t_i}. \tag{6}$$

- mean IU:

$$iu_{mean} = \frac{1}{n_c} \times \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}. \tag{7}$$

- f.w. IU:

$$iu_{weighted} = \frac{1}{\sum_k t_k} \times \sum_i \frac{t_i \times n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}. \tag{8}$$

C. Evaluation

We compare the proposed method to the previous methods [15], [16]. Similar to the proposed method, superpixels are considered as the basic units of labeling. In [15], the features are learned on randomly selected grayscale image patches with a stacked convolutional autoencoder in an unsupervised manner. Then the features and the labels of the superpixels are used to train a classifier. With the trained classifier, superpixels are classified into different classes. In [16], a Conditional Random Field (CRF) is applied in order to model the local and contextual information jointly for the superpixel labeling task. The trained classifier in [15] is considered as the local classifier in [16]. Then the local classifier is used to train a contextual classifier which takes the output of the local classifier as input and output the scores of given labels. With the local and contextual classifiers, a CRF is trained to label the superpixels of a given image. In the experiments, we use a multilayer perceptron (MLP) as the local classifier in [15], [16] and an MLP as the contextual classifier in [16]. Simple Linear Iterative Clustering algorithm (SLIC) [20] is applied to generate the superpixels. The superiority of SLIC over other superpixel algorithms is demonstrated in [15]. In the experiments, for each image, 3000 superpixels are generated.

Table II reports the pixel accuracy, mean pixel accuracy, mean IU, and f.w. IU of the three methods. It is shown that the proposed CNN outperforms the previous method. Figure 2 gives the segmentation results of the three methods. We can see that visually the CNN achieves more accurate segmentation results compared to other methods.

D. Max Pooling

Pooling is a widely used technology in CNN. Max pooling is the most common type of pooling which is applied in order to reduce spatial size of the representation to reduce the number of parameters of the network. In order to show the impact of max pooling for the segmentation task. We add a max pooling layer after the convolution layer. The pooling size is 2×2 pixels. Table II reports the performance of the CNN with a max pooling layer. We can see that only on the *CB55* dataset, with max pooling the mean pixel accuracy and mean IU are slightly improved. In general, adding a max pooling layer does not improve the performance of the segmentation task. Figure 3 reports the f.w. IU of the CNN with different max pooling sizes. We define the max pooling size as $m \times m$, such that $m = \{2 \times n \mid n \in \mathbb{N}, 0 \leq n \leq 13\}$. We can see that increasing the pooling size decreases the performance. The reason is that for some computer vision problems, e.g., object recognition and text extraction in natural images, the exact location of a feature is less important than its rough location relative to other features. However, for a given document image, to label a pixel in the center of a patch, it is not sufficient to know if there is text somewhere in that patch, the location of the text is

Table II: Performance (in percentage) of superpixel labeling with only local MLP, CRF, and the proposed CNN.

	G. Washington				*Parzival*				*St.Gall*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
Local MLP [15]	87	89	75	83	91	64	58	86	95	89	84	92
CRF [16]	**91**	90	76	85	93	70	63	88	97	88	84	94
CNN	**91**	**91**	**77**	**86**	**94**	**75**	**68**	**89**	**98**	**90**	**87**	**96**
CNN (max pooling)	**91**	90	**77**	**86**	**94**	**75**	**68**	**89**	**98**	**90**	**87**	**96**

	CB55				*CSG18*				*CSG863*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
Local MLP [15]	83	53	42	72	83	49	39	73	84	54	42	74
CRF [16]	84	53	42	75	86	47	37	77	86	51	42	78
CNN	**86**	59	47	**77**	**87**	**53**	41	79	**87**	**58**	**45**	**79**
CNN (max pooling)	**86**	**60**	**48**	**77**	**87**	**53**	**42**	**80**	**87**	57	**45**	**79**

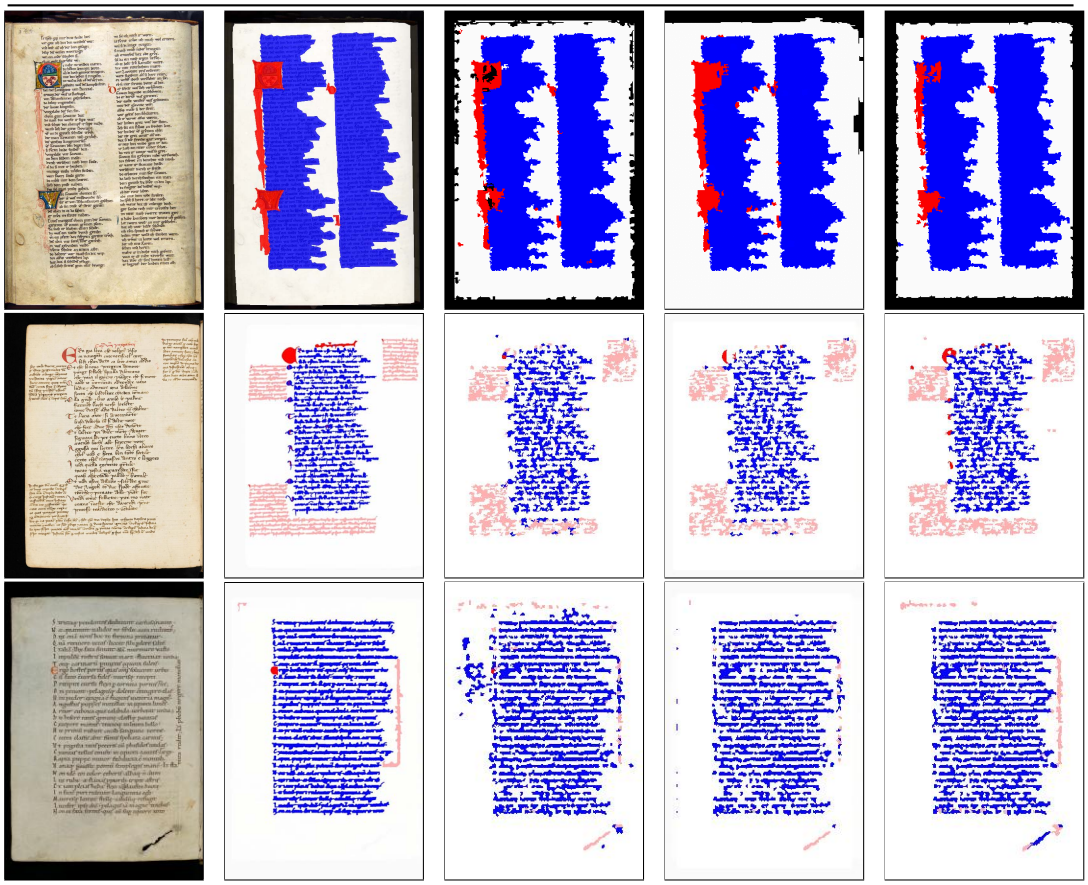

Figure 2: Segmentation results on the *Parzival*, *CB55*, and *CSG863* datasets from top to bottom respectively. The colors: black, white, blue, red, and pink are used to represent: *periphery*, *page*, *text*, *decoration*, and *comment* respectively. The columns from left to right are: input, ground truth, and segmentation results of the local MLP, CRF, and CNN respectively.

needed. Therefore, the exact location of a feature is helpful for the page segmentation task.

E. Number of Kernels

In order to show the impact of the number of kernels of the convolution layer on the segmentation task. We define the number of kernels as K. In the experiments, we set $K \in \{1, 2, 4, 6, 8, 10, 12, 14\}$. Figure 4 reports the f.w. IU of the one convolution layer CNN with different number of kernels. We can see that except on the *CS18* dataset, when $K \geq 4$ the performance is not improved.

F. Number of Layers

In order to show the impact of the number of convolutional layers on the page segmentation task. We incrementally add convolutional layers, such that there is two more kernels on the current layer than the previous layer. Figure 5 reports the f.w. IU of the CNN with different

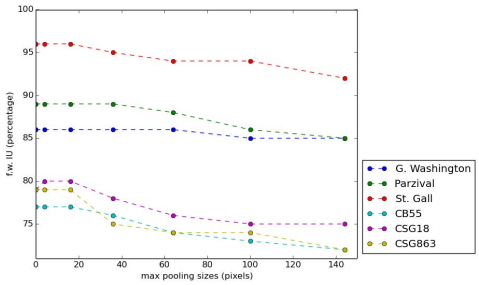

Figure 3: f.w. IU of the CNN on different max pooling sizes.

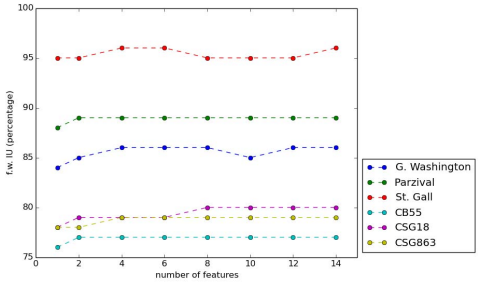

Figure 4: f.w. IU of the CNN on different numbers of kernels.

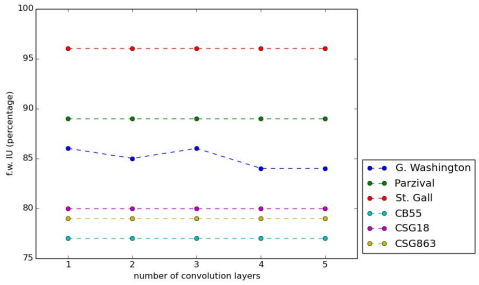

Figure 5: f.w. IU of the CNN on different numbers of conv layers.

number of convolution layers. It is show that the number of layers does not affect the performance of the segmentation task. However, on the *G. Washington* dataset, with more layers, the performance is degraded slightly. The reason is that compared to other datasets, the *G. Washington* dataset has fewer training images. Furthermore, the layouts of the pages in the *G, Washington* dataset are more varied.

G. Number of Training Images

In order to show the performance under different amount of training images. For each dataset, we choose N images in the training set to train the CNN. For each experiment, the number of batches is set to 5000. Figure 6 reports the f.w. IU under different values of N, such that $N \in \{1, 2, 4, 8, 10, 12, 14, 16, 18, 20\}$[1]. We can see that in general, when $N > 2$, the performance is not improved. However, on the *G. Washington* dataset, with more training images, the performance is degraded slightly. The reason is that compared to the other datasets, on the *G. Washington*

[1]In the *G. Washington* dataset, there is 10 training images. Therefore, $N \in \{1, 2, 4, 8, 10\}$.

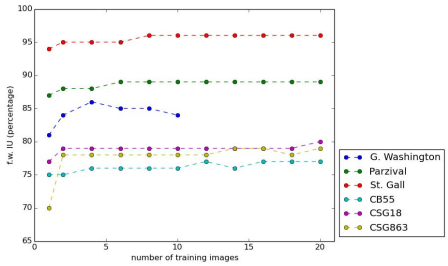

Figure 6: f.w. IU of the CNN on different numbers of training images.

dataset the pages are more varied and the ground truth is less consistent.

H. Run Time

The proposed CNN is implemented with the python library Theano [27]. The experiments are performed on a PC with an Intel Core i7-3770 3.4 GHz processor and 16 GB RAM. On average, for each image, the CNN takes about 1 second processing time. The superpixel labeling method [15] and CRF model [16] take about 2 and 5 seconds respectively.

IV. Conclusion

In this paper, we have proposed a convolutional neural network (CNN) for page segmentation of handwritten historical document images. In contrast to traditional page segmentation methods which rely on off-the-shelf classifiers trained with hand-crafted features, the proposed method learns features directly from image patches. Furthermore, feature learning and classifier training are combined into one step. Experiments on public datasets show the superiority of the proposed method over the previous methods. While many researchers focus on applying very deep CNN architectures for different tasks, we show that with the simple one convolution layer CNN, we have achieved comparable performance compared to other network architectures.

Acknowledgment

This work is supported by the Swiss National Science Foundation project HisDoc 2.0 with the grant number: 205120 150173 and National Natural Science Foundation of China with the grant numbers: 61202257 and 61650110512.

References

[1] C. Grana, D. Borghesani, and R. Cucchiara, "Automatic segmentation of digitalized historical manuscripts," *Multimedia Tools and Applications*, vol. 55, no. 3, pp. 483–506, 2011.

[2] S. S. Bukhari, T. M. Breuel, A. Asi, and J. El-Sana, "Layout analysis for arabic historical document images using machine learning," in *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*. IEEE, 2012, pp. 639–644.

[3] K. Chen, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold, "Robust text line segmentation for historical manuscript images using color and texture," in *2014 22nd International Conference on Pattern Recognition (ICPR)*. IEEE, 2014, pp. 2978–2983.

[4] K. Chen, H. Wei, J. Hennebert, R. Ingold, and M. Liwicki, "Page segmentation for historical handwritten document images using color and texture features," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 488–493.

[5] M. Bulacu, R. van Koert, L. Schomaker, and T. van der Zant, "Layout analysis of handwritten historical documents for searching the archive of the cabinet of the dutch queen," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 1. IEEE, 2007, pp. 357–361.

[6] C. Panichkriangkrai, L. Li, and K. Hachimura, "Character segmentation and retrieval for learning support system of japanese historical books," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, 2013, pp. 118–122.

[7] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of historical handwritten documents into text zones and text lines," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 464–469.

[8] R. Cohen, A. Asi, K. Kedem, J. El-Sana, and I. Dinstein, "Robust text and drawing segmentation algorithm for historical documents," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, 2013, pp. 110–117.

[9] A. Asi, R. Cohen, K. Kedem, J. El-Sana, and I. Dinstein, "A coarse-to-fine approach for layout analysis of ancient manuscripts," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*. IEEE, 2014, pp. 140–145.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3304–3308.

[14] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, "Page segmentation of historical document images with convolutional autoencoders," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*. IEEE, 2015, pp. 1011–1015.

[15] K. Chen, C.-L. Liu, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, "Page segmentation for historical document images based on superpixel classification with unsupervised feature learning," in *Document Analysis System (DAS), 2016 12th IAPR International Workshop on*. IEEE, 2016, pp. 299–304.

[16] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, C.-L. Liu, and R. Ingold, "Page segmentation for historical handwritten document images using conditional random fields," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 90–95.

[17] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the eighteenth international conference on machine learning, ICML*, vol. 1, 2001, pp. 282–289.

[18] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[20] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[22] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[23] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[24] K. Chen, M. Seuret, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold, "Ground truth model, tool, and dataset for layout analysis of historical documents," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2015, pp. 940 204–940 204.

[25] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold, "Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 471–476.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[27] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a cpu and gpu math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.