Page Segmentation of Historical Document Images with Convolutional Autoencoders

Kai Chen*, Mathias Seuret*, Marcus Liwicki*†, Jean Hennebert*‡, and Rolf Ingold*,
*DIVA (Document, Image and Voice Analysis) research group
Department of Informatics, University of Fribourg, Switzerland
Email: {firstname.lastname}@unifr.ch
†DFKI - German Research Center for Artificial Intelligence
Email: liwicki@dfki.uni-kl.de
‡University of Applied Sciences, HES-SO//FR, Bd. de Pérolles 80, 1705 Fribourg, Switzerland
Email: jean.hennebert@hefr.ch

***Abstract*—In this paper, we present an unsupervised feature learning method for page segmentation of historical handwritten documents available as color images. We consider page segmentation as a pixel labeling problem, i.e., each pixel is classified as either *periphery*, *background*, *text block*, or *decoration*. Traditional methods in this area rely on carefully hand-crafted features or large amounts of prior knowledge. In contrast, we apply convolutional autoencoders to learn features directly from pixel intensity values. Then, using these features to train an SVM, we achieve high quality segmentation without any assumption of specific topologies and shapes. Experiments on three public datasets demonstrate the effectiveness and superiority of the proposed approach.**

I. Introduction

Nowadays, a large number of historical documents have been digitized and made available to the public. There is an increasing need to develop robust image analysis systems to retrieve textual information from these documents. Page segmentation is a prerequisite step of document image analysis and understanding. It aims at splitting a page image into regions of interest and distinguishing text blocks from other regions. In contrast to printed contemporary documents, page segmentation on historical documents is more difficult, due to many variations, such as layout structure, decoration, writing style, texture, and degradation.

Some page segmentation methods have been developed recently. These systems rely on hand-crafted features [3], [14] or prior knowledge [11], [15], [16], or models that combine hand-crafted features with domain knowledge [2], [8], [10]. Perceiving the problem from another viewpoint, 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, they can be used for feature learning to extract the layout information of the document images.

In this paper, we propose a novel page segmentation approach based on unsupervised feature learning. We consider the segmentation as a pixel classification problem. Each pixel is represented by a feature vector. By training a classifier with the features, we classify each pixel into one of the four classes: *periphery*, *background*, *text block*, and *decoration*. Instead of using carefully hard-coded features to train a classifier, we use an unsupervised feature learning method.

Creating algorithms that can automatically learn useful representations from unlabeled data is important. Hand coding features is cumbersome because the user has to select features a priori good to handle the specificities of a given dataset. Furthermore, labeling data is time consuming. Feature learning algorithms allow us to generate large amounts of features from unlabeled data. They have been applied in many applications, such as image classification [17] and audio recognition [13]. In this work, we propose a convolutional autoencoder (CAE) system which is based on the encoder/decoder architecture [9]. We use a single hidden layer neural network as an autoencoder (AE) to learn features from the input. Concretely, we take the pixel values as input and target of the AE. By having few neurons on the hidden layer and applying the backpropagation, the network has to learn a compressed representation. We first apply an AE on small image patches to learn low-level features. Then we take the learned feature mapping functions and convolve them with larger patches [12]. The outputs of the AE are wired to the inputs of a successive AE to learn higher level features. Finally, we use the learned features to train a support vector machine (SVM) to predict class labels for each pixel. Experiments on three public historical document images datasets [7] show that it is possible to learn reliable features with the proposed approach. Compared to our previous work [6], the proposed system achieves superior performance on the *George Washington* and the *Parzival* datasets and comparable performance on the *Saint Gall* dataset.

In summary, our approach differs from traditional page segmentation methods in two properties. (1) The features are learned directly from the pixels without supervision. (2) Preprocessing (i.e., binarization, connected components extraction) and prior knowledge are not needed. The rest of the paper is organized as follows. Section II gives an overview of some related work in page segmentation for historical document images. Section III describes the proposed segmentation method. Section IV reports on our experimental results and Section V presents our conclusions.

978-1-4799-1805-8/15/$31.00 ©2015 IEEE

II. Related Work

Some methods have been presented in the literature to perform the segmentation task. However, most of these methods rely on image preprocessing such as binarization, connected components (CCs) extraction, off-the-shelf classifiers trained on hand-crafted features and prior knowledge.

In the Historical Document Layout Analysis Competition (ICDAR 2011) [1], four layout analysis methods for printed historical document images were evaluated and compared with a state-of-the-art commercial software. The results indicate that there is a convergence to a certain methodology with some variations in the approach, i.e., CCs aggregation on binary images. However, it is also clear that there is still a considerable need to develop robust methods for layout analysis on historical document images.

Panichkriangkrai et al. [15] proposed text line and character extraction system of Japanese historical woodblock printed books. Text lines were separated by using vertical projection on binarized images. To extract kanji characters, rule-based integration was applied to merge or split the CCs which were extracted by applying an adaptive binarization on the gray scale images. Van Phan et al. [16] use the area Voronoi diagram to represent the neighborhood and boundary of CCs. By applying predefined rules, characters were extracted by grouping adjacent Voronoi regions. Bukhari et al. [3] extracted the features of CCs. The normalized height, foreground area, relative distance, orientation, and neighbourhood information of the CCs were considered as features to train a multilayer perceptron to classify CCs to relevant classes of text. Similarly, Cohen et al. [8] applied Laplacian of Gaussian on multi-scale binarized image to extract CCs which were considered as candidate text lines. Based on prior knowledge, noise CCs were removed. Features such as bounding box size, area, stroke width, estimated text lines distance were used to label each CC into text or non-text by using an energy minimization method. Asi et al. [2] proposed a two-steps segmentation method for arabic historical document images. The main text area was first segmented by using Gabor filters. Then the refine segmentation was formulated as an energy minimization task. Mehri et al. [14] proposed a texture based segmentation method. First they computed textual features, i.e., autocorrelation, Grey Level Co-occurrence Matrix, and Gabor features, on randomly selected foreground pixels. Then by applying the Consensus Clustering method, they estimated the clusters number. Nearest Neighbor Search algorithm with the Mahalanobis distance was used to assign the same label for each similar cluster.

III. System Description

In contrast to the state-of-the-art methods, our objective is to use features which are automatically learned from unlabeled pixels. The architecture of our work is based on [9], [12]. In the feature learning process, we use the unlabeled training set to train a CAE. The process consists of two steps: (1) Learn low-level features by using an AE on a set of image patches randomly selected from the unlabeled training set. (2) Use CAE to learn higher level features.

Given a set of labeled training images and a trained CAE, we perform feature extraction with the following steps: (1) Extract features with the CAE on randomly selected image patches from the training images. (2) Concatenate the activation values of each layer of the CAE as feature vectors to train an SVM.

A. Feature learning

Feature learning is the key component of our system. Our aim is to develop a simple feature learning system based on the encoder and decoder paradigm. We use a single-layer fully-connected neural network as an AE which tries to reconstruct the input data. Features can be discovered in the hidden layer. The number of hidden neurons is smaller than the input size to prevent learning of identity functions. Concretely, the AE learns the weights W_1 and W_2, such that $f(W_2 f(W_1 x)) = \hat{x}$, where x is the input vector, the output $\hat{x}$ is similar to x; f is the activation function, we choose f to be the soft-sign function, such that $f(x) = \frac{x}{1+|x|}$; W_1 and W_2 are the weights on the first and second layer respectively; W_1 is used for encoding and W_2 is used for decoding. After using backpropagation to minimize squared reconstruction error, W_1 is used to compute the learned features, i.e., the mapping from input vector x to feature vector z where $z = f(W_1 x)$. In our system, the input vector x is the concatenation of each pixel's RGB values of a $w \times w$ pixels image patch extracted from a document image.

In order to learn high-dimensional feature representations from unlabeled pixels, our feature learning system contains three levels. We denote $x^{(k)}$ as the input vector and $W^{(k)}$ as the weights of the AE on the k-th level. Our feature learning strategy on each level is described as follows:

First Level. We randomly select 500K 5×5 pixels image patches $P^{(1)}$ from the training set. Therefore, the size of input vector which contains the three RGB values is $x^{(1)} \in \mathbb{R}^{75}$. We set the number of hidden units of the AE to 40. Applying stochastic gradient descent method on the image patches, we estimate the weights $W_1^{(1)}$ and $W_2^{(1)}$. Figure 2a depicts the feature learning process on this level.

Second Level. Document images have the property that one part of an image shares similarities with other part [12]. Thanks to this property, we can use the learned feature mapping function of the previous level and convolve them with larger image patch to learn high-order feature representations. The 15×15 pixels image patch $P^{(2)}$ is composed by 3×3 patches $P^{(1)}$ without overlapping. The input vector of each $P_n^{(1)}$ is denoted by $x_n^{(1)}$, where n is the patch number. The input vector

(a) (b) (c)

Figure 1: The example pages 1a, 1b, and 1c are taken from the George Washington (letterbook 1, page 307), Parzival (Cod. 857, page 144) and Saint Gall (Cod. Sang, 562, page 27) datasets respectively.

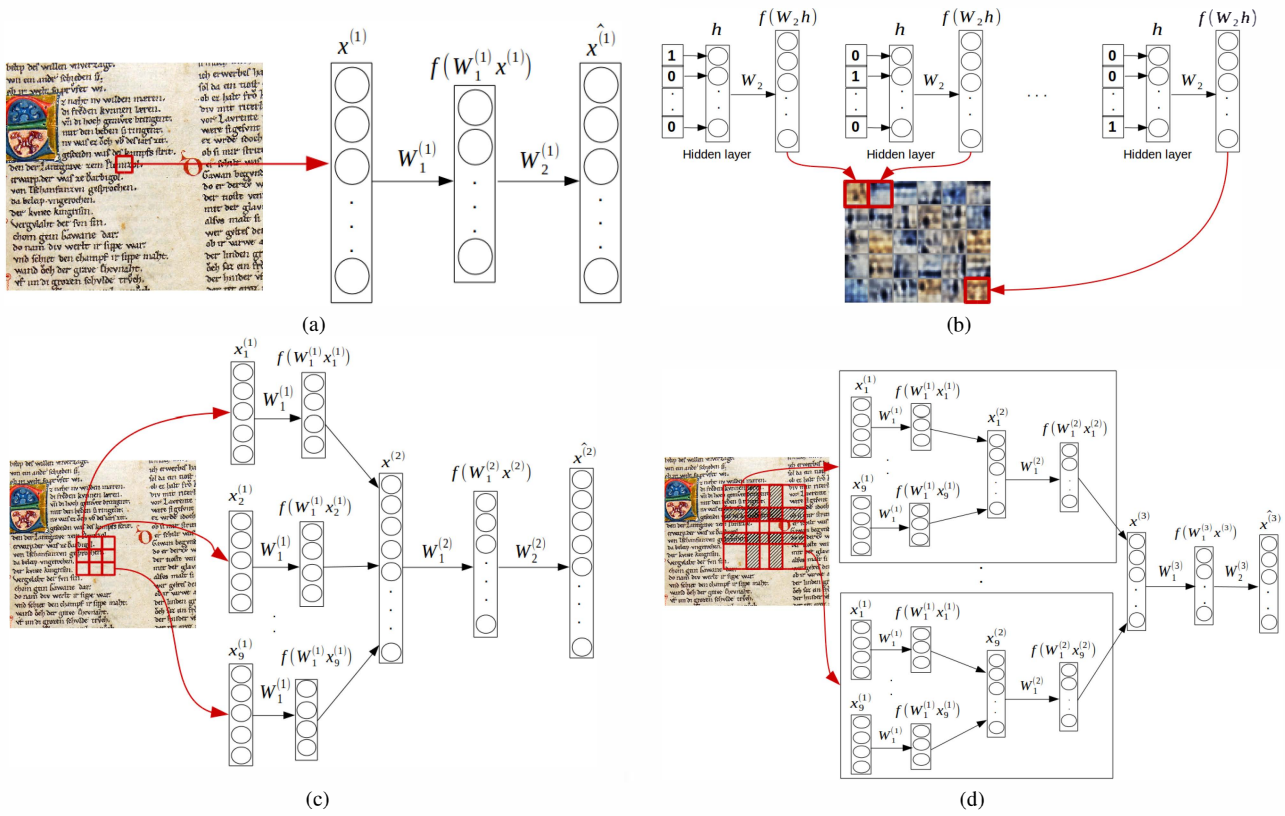


Figure 2: CAE architecture: first level 2a, second level 2c, and third level 2d. Feature visualization 2b. The red areas are for illustration purposes and do not correspond to real-sizes of the windows.

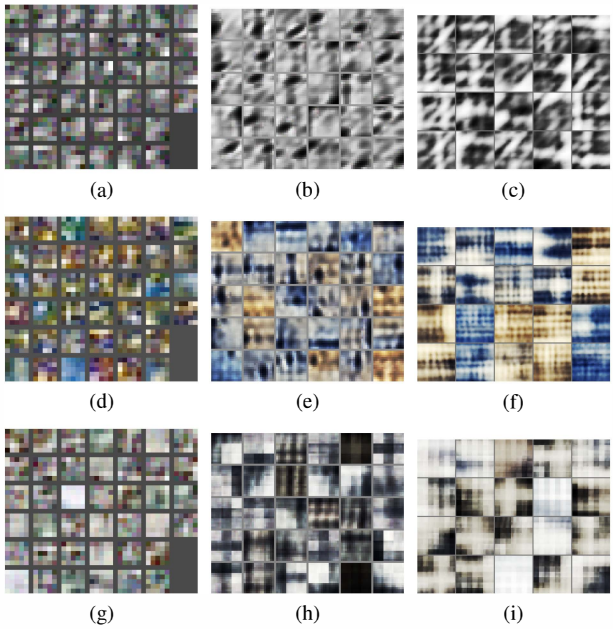


Figure 3: Learned features on the *George Washington* dataset: Level 1 3a, Level 2 3b, and Level 3 3c. Learned features on the *Parzival* dataset: Level 1 3d, Level 2 3e, and Level 3 3f. Learned features on the *St. gall* dataset: Level 1 3g, Level 2 3h, and Level 3 3i.

$x^{(2)}$ is the concatenation of $f(W_1^{(1)}x_1^{(1)}), \cdots, f(W_1^{(1)}x_9^{(1)})$. Figure 2c depicts the feature learning process on this level. In our settings, the number of hidden units of the second-level AE is 30 and 500K random patches $P^{(2)}$ are used for training.

Third Level. We repeat the same procedure as for the second level. However, we add an overlapping of 5 pixels

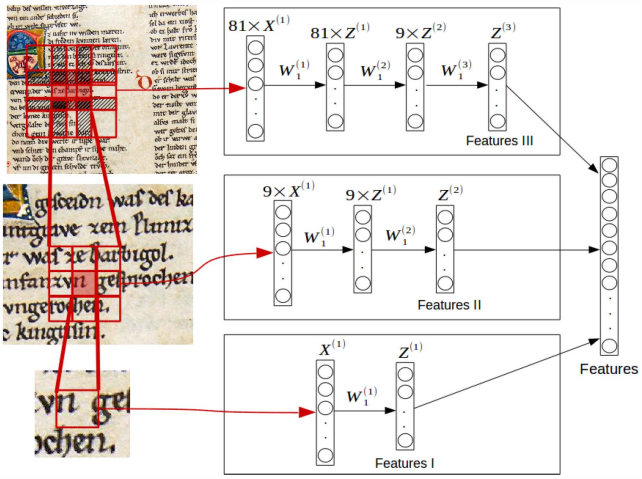


Figure 4: Feature extraction with convolutional autoencoders.

on the patches $P^{(2)}$ in order to get more information on the boundaries, so $P^{(3)}$ covers 35×35 pixels. Figure 2d depicts the feature learning process. The AE has 20 hidden neurons and is trained on 200K random patches

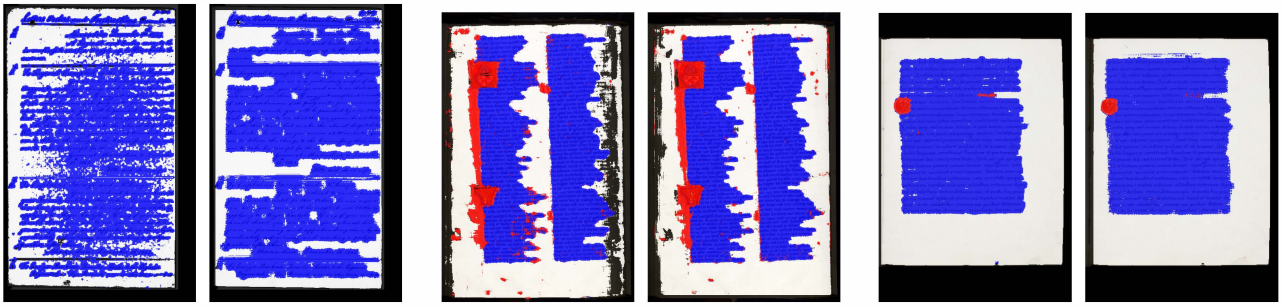
The settings have been tuned in our cross validation procedure to reach a trade off between accuracy and CPU load.

B. *Feature visualization*

We show in Figure 3 the learned features from the different levels on three historical document image datasets [7]. Figure 1 gives some example pages. To visualize the learned feature of a hidden neuron of the first-level AE, we set its output to 1 and the outputs of the other hidden neurons to 0, then decode the

Table I: Classification accuracy on three historical document images datasets.

	George Washington		Parzival		Saint Gall	
	Features size	Accuracy (%)	Features size	Accuracy (%)	Features size	Accuracy (%)
hand-crafted features [6]	124	90	200	92.14	162	**97.73**
Feature learning with CAE						
One level (5×5)	40	87.03	40	92.12	40	97.24
One level (15×15)	40	89.83	40	95.31	40	95.86
One level (35×35)	40	92.32	40	86.58	40	86.61
Two levels	70	89.2	70	96.31	70	96.72
Three levels	90	**92.65**	90	**96.64**	90	97.66



(a) (b) (c) (d) (e) (f)

Figure 5: Segmentation ($\alpha = 2^{-2}, N = 100$k) results of pages: 1a, 1b, and 1c. The segmentation results using the hand-crafted features are: 5a, 5c, and 5e respectively. The segmentation results using the learned features are: 5b, 5d, and 5f respectively. The color: black, white, blue, and red are used to represent: *periphery*, *background*, *text block*, and *decoration*.

corresponding image patch, as shown in Figure 2b. For higher-level features, instead of directly constructing an image with the decoded outputs (which correspond to lower-level features, not pixels), we iteratively decode the outputs with lower-level decoders until we reach a pixel-level representation.

C. Feature extraction and classifier training

The features of a given pixel are the concatenation of the nth-level features $z^{(n)}$ from patches $P^{(n)}$ centred on the pixel where $z^{(n)} = f(W_1^{(n)} x^{(n)})$, $n \in \{1, 2, 3\}$. The details of the construction of $P^{(n)}$ and input vector $x^{(n)}$ are given in Subsection III-A. An SVM is trained on randomly selected pixels features with their label on the training set. Figure 4 depicts the feature extraction strategy.

IV. Experiments

In order to compare the proposed method with our previous work [6], we use the same datasets[1] [7], scaling factor[2] α and follow the same evaluation protocol. Some example pages of the datasets are given in Figure 1. In our earlier work [5], [6], hand-crafted features such as color variance, smoothness, Laplacian, Local Binary Pattern, and Gabor Dominant Orientation Histogram are used for classification. Our objective is to compare the learned features by using the proposed feature learning system with the "selected to be optimal" hand-crafted features determined in [6], therefore we use the optimal configuration for the previous method[3]. The same classifier, i.e., SVM [4] is used for the classification.

[1] http://diuf.unifr.ch/main/hisdoc/divadia

[2] Due to the large size of the images, we scale images to smaller size with a scaling factor $\alpha < 1.0$.

[3] The results of our previous system shown in this paper are not the same as in [6], because in this work we have used the new ground truth. The motivation and details of the new ground truth are explained in [7].

A. Effect of feature learning

We first show the experiments with the same setting as in [6], i.e., $\alpha = 2^{-4}$, all pixels of the training images are used for training. To show how feature learning affects the performance, we use various features. These features are: (1) hand-crafted features [6], (2) features learned with one-level CAE[4] with patch sizes: 5×5, 15×15, 35×35 pixels, (3) features learned with two-levels and three-levels CAE as described in Section III-A.

Table I reports on the performances of the classification with different features on three datasets. We observe in comparison to our previous method [6], that by using the three-levels CAE, we achieved superior performances on the *George Washington* and *Parzival* datasets. The differences of performances on the *Saint Gall* dataset are small, probably due to the fact that this dataset contains less noise and neater layout. Notably, the hand-crafted features sizes are 124, 200, and 162 for the *George Washington*, *Parzival*, and *Saint Gall* datasets respectively. The three-levels learned features size is 90 for the three datasets. We observe that using only one-level AE for feature learning provides already interesting results but is less effective for the classification. Since different datasets have different context information, e.g., the characters sizes, writing style, space between lines and paragraphs, the size of patch affects the performance. Using the convolutional strategy for feature learning is robust in the sense that it avoids us to estimate patch sizes manually for different datasets. As illustrated in Figure 3, the features learned on the first level contain low-level information, while higher level patterns such as strokes and corners are captured in the second- and third-level features.

[4] A one-level CAE is just an AE.

Table II: Classification accuracy (%) on various scaling factor values and number of training samples.

	$\alpha = 2^{-3}$			$\alpha = 2^{-2}$		
	10k	50k	100k	10k	50k	100k
George Washington						
hand-crafted features	79.69	82.67	83.61	63.77	74.15	78.57
learned features	**84.97**	**86.10**	**86.42**	**81.30**	**85**	**85.82**
Parzival						
hand-crafted features	76.26	90.17	92.39	44.39	72.99	79.06
learned features	**86.54**	**95.26**	**96.13**	**58.64**	**87.33**	**90.49**
Saint Gall						
hand-crafted features	**96.67**	**97.46**	**97.66**	**94.87**	**96.89**	**97.22**
learned features	95.52	96.82	97.26	94.33	96.43	96.87

B. Effect of image resolution and number of training samples

In these experiments, we focus on evaluating how image resolution and number of training samples affect the performance for the feature learning approach and the hand-crafting feature approach. When an image has a high resolution, it is time consuming to take all the pixels of the training set to train a classifier. Moreover, many parts of the document images are similar, therefore many pixels are redundant. Instead of using all the labeled pixels of the training set, we randomly select some pixels and use the features of these pixels with their labels to train the classifier. We select equivalent number of pixels N for each class as training samples[5].

In Table II it is shown that with the proposed feature learning approach, we achieve superior performances on the *George Washington* and the *Parzival* datasets. The performances on the *Saint Gall* dataset are comparable. Some results are given in Figure 5. Notably, with only 10K labeled training pixels, the accuracies are improved from 79.69% to 84.97% and from 76.26% to 86.54% for the *George Washington* and *Parzival* datasets using a scaling factor $\alpha = 2^{-3}$. Furthermore, it is shown in Table II that when images have higher resolution, i.e., $\alpha = 2^{-2}$, the improvement of the performance becomes more significant, e.g., 17.53% and 14.25% for the *George Washington* and *Parzival* datasets with 10K training pixels. An interpretation could be since in higher resolution, each pixel contains more information, the learned features are more reliable than the hand-crafted features. Compared to using hand-crafted features, using the learned features, we are able to achieve superior performance by using fewer labeled pixels to train a classifier.

V. Conclusion

In this paper we presented a novel page segmentation method for color historical document images. The method is based on unsupervised feature learning algorithms with single-layer neural networks. We show that with the three-levels convolutional autoencoder architecture and layer-by-layer unsupervised training strategy, reliable feature representations can be learned directly from pixels. With the learned features we achieved superior performance compared to our previous system [6] which is based on the hand-crafted features. While much research on page segmentation focused on developing methods with hand-crafted features and prior knowledge, our results show that it is possible to achieve high performance with automatic feature learning algorithms. With more sophisticated feature learning methods currently developed by machine learning researchers, we believe that the proposed system might achieve better performance compared to other methods that rely on hand-crafted features and domain knowledge. Our future work will focus on discovering how the parameters, such as the number of levels, the number of hidden units, and the sparsity of active hidden units of the convolutional autoencoders affect the performance.

[5]Since there are fewer pixels of *decoration* and *periphery* than others, if the total number of pixels of the class N' is less than N, then we take N' pixels as training samples.

References

[1] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, "Historical Document Layout Analysis Competition." in *International Conf. on Document Analysis and Recognition*, pp. 1516-1520, 2011.

[2] A. Asi, R.Cohen, K. Kedem, J. El-Sana, and I. Dinstein, "A Coarse-to-Fine Approach for Layout Analysis of Ancient Manuscripts." in *ICFHR*, pp. 140-144, 2014.

[3] S. S. Bukhari, T. M. Breuel, A. Asi, and J. El-Sana, "Layout Analysis for Arabic Historical Document Images Using Machine Learning." in *ICFHR*, pp. 639-644, 2012.

[4] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines." *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27, 2011.

[5] K. Chen, H. Wei, M. Liwicki, J. Hennebert, and R. Ingold, "Robust Text Line Segmentation for Historical Manuscript Images using Color and Texture." in *ICPR*, pp. 2978-2983, 2014.

[6] K. Chen, W. Hao, J. Hennebert, R. Ingold, and M. Liwicki, "Page Segmentation for Historical Handwritten Document Images Using Color and Texture Features." in *ICFHR*, pp. 488-493, 2014.

[7] K. Chen, M. Seuret, W. Hao, M. Liwicki, J. Hennebert, and R. Ingold, "Ground Truth Model, Tool, and Dataset for Layout Analysis of Historical Documents." *IS&T/SPIE Electronic Imaging*, 940204-10, 2015.

[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 International Workshop on Historical Document Imaging and Processing*, pp. 110-117, 2013.

[9] G. E. Hinton, R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks." *Science*, 313, no. 5786, pp. 504-507, 2006.

[10] J. Ji, L. Peng, and B. Li, "Graph Model Optimization Based Historical Chinese Character Segmentation Method." in *IAPR International Workshop on Document Analysis Systems*, pp. 282-286, 2014.

[11] S. Khedekar, V. Ramanaprasad, S. Setlur, and V. Govindaraju, "Text-Image Separation in Devanagari Documents." in *International Conf. on Document Analysis and Recognition*, vol. 3, pp. 1265-1269. 2003.

[12] Y. LeCun, B. Léon, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition." in *Proceedings of the IEEE*, 86, no. 11, pp. 2278-2324, 1998.

[13] H. Lee, R. Grosse, R. Ranganath, and A.Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." in *Proceedings of the Annual International Conference on Machine Learning*, pp. 609-616, 2009.

[14] M. Mehri, P. Gomez-Krämer, P. Héroux, A. Boucher, and R. Mullot, "Texture feature evaluation for segmentation of historical document images." in *Proceedings of the International Workshop on Historical Document Imaging and Processing*, pp. 102-109, 2013.

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

[16] T. Van Phan, B. Zhu, and M. Nakagawa, "Development of Nom character segmentation for collecting patterns from historical document pages." in *Proceedings of the Workshop on Historical Document Imaging and Processing*, pp. 133-139, 2011.

[17] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification." in *Conference on Computer Vision and Pattern Recognition*, pp. 1794-1801, 2009.