Unsupervised Deep Learning for Handwritten Page Segmentation

Ahmad Droby, Berat Kurar Barakat, Borak Madi, Reem Alaasam and Jihad El-Sana
Ben-Gurion University of the Negev
{drobya,berat,borak,rym}@post.bgu.ac.il
el-sana@cs.bgu.ac.il

***Abstract*—Segmenting handwritten document images into regions with homogeneous patterns is an important pre-processing step for many document images analysis tasks. Hand-labeling data to train a deep learning model for layout analysis requires significant human effort. In this paper, we present an unsupervised deep learning method for page segmentation, which revokes the need for annotated images. A siamese neural network is trained to differentiate between patches using their measurable properties such as number of foreground pixels, and average component height and width. The network is trained that spatially nearby patches are similar. The network's learned features are used for page segmentation, where patches are classified as main and side text based on the extracted features. We tested the method on a dataset of handwritten document images with quite complex layouts. Our experiments show that the proposed unsupervised method is as effective as typical supervised methods.**

***Index Terms*—layout analysis, segmentation, historical, documents, unsupervised, Siamese network, deep-learning, page segmentation, hand-written**

I. Introduction

Manually copying of manuscripts was the ultimate way scholars shared knowledge before the popularisation of the printing press. Notes were frequently added by scholars to the margin of pages, and often contribute valuable information concern the main text and the manuscript as a whole. In addition, the content of a manuscript's marginal notes help historians to analyze the authenticity, temporal, and geographical origin of the manuscript.

The increasing number of available digital scans of historical manuscripts, call for reliable automatic processing systems, which would allow historians and scholars to access and explore this knowledge more efficiently.

Page segmentation is an essential preprocessing step for many document image processing tasks. Due to the irregular structure, varying writing styles, and non-rectangular layout of historical handwritten documents [1], [2], segmenting them into main and side text poses a challenging research problem.

Learning free based page layout analysis methods rely on human crafted features, such as connected component statistics [3], [4], SIFT of points of interests [5], color and texture features [6]–[8], etc. Due to the highly irregular structure and varying text style of historical handwritten documents, those methods do not generalize well. Therefore, researchers have been opting to learn those features instead. Page segmentation methods with a learning component generally outperform traditional learning free based methods. However, those methods require a large amount of manually annotated data for training in order to perform well. Obtaining such data is tedious and time-consuming; and in some cases requires domain experts.

We present an unsupervised deep learning method for page segmentation that utilizes measurable features such as spatial proximity, number of foreground pixels and average character height and width. The method first trains a siamese neural network model, M, then uses M for feature extraction. A siamese network model contains two Convolutional Neural Networks (CNNs) with shared weights. The CNNs work in parallel on two different inputs to extract comparable feature vectors. We train a siamese network to discriminate between patches with statistical differences of connected components; e.g., various number of foreground pixels and different background areas. Typically, in documents with side notes nearby patches belong to the same class (main or side text) with high probability. Based on this basic assumption, the network is trained that two spatially nearby patches are similar. Following training, we use the CNN component of the Siamese network to extract feature vector for every patch in a given page. The extracted feature vectors are then used to segment the page into main and side text regions. Our experimental results show that the accuracy of this method is comparable and in most cases surpasses the accuracy of supervised methods.

The rest of the paper is structured as follows. Section II reviews related work. In Section III we present our method in detail. Experimental results are reported in Section IV. Finally, conclusions are drawn, and future work is discussed in Section V.

II. Related Work

Typically, page segmentation algorithms use features in order to segment pages into regions with homogeneous patterns. Existing page segmentation algorithms can be classified into two categories based on the type of used features.

A. Hand-Crafted Features

Traditional page segmentation algorithms rely upon hard-coded features, specification of documents structure, assumptions and statistical rules. Graz *et al.* [5] presented an approach to analyze the layout of handwritten documents using Scale Invariant Feature Transform (SIFT). The method uses Difference of Gaussian (DOG) to compute interest points, which guide the detection of layout entities. Finally, Support Vector

Machine (SVM) is applied to classify the points into entity classes. Bukhari *et al.* [3] first extracts discriminative and simple features in the level of connected components, such as relative distance, foreground area, orientation, normalized height, and neighborhood information. Then a Multi-Layer Perceptron (MLP) classifies the connected components into side notes and main body texts. Finally, a voting scheme refines final classification results. Asi *et al.* [9] proposed a learning-free approach for page segmentation of Arabic manuscripts. This is a two-step method: coarse segmentation and fine segmentation. Coarse segmentation utilizes Gabor texture filter and fine segmentation optimizes the results using energy minimization. Wong *et al.* [10] use Run Length Smearing Algorithm (RLSA) for page segmentation. RLSA links together the neighboring areas that are black within predefined c pixels. Two distinct bit maps are generated by applying RSLA row-by-row and column-by-column to a document. These maps are combined using 'AND' logical operator to produce segmented regions. These regions are then classified into text and non-text according to several criteria, such as black-white transitions and the total number of black pixels. Akiyama and Hagita [11] divides an input document into smaller regions using basic features such as projection profiles, crossing counts, and circumscribed rectangles. These regions are then classified into headlines, text lines, and graphics regions. Apostolos [12] identifies background space surrounding the page regions and describes them using white tiles, which are horizontal rectangular white spaces. The algorithm can segment and identify regions with severe skew, but it does not classify them. Journet *et al.* [13] extracts texture features and applies a multi-resolution analysis to avoid any assumption about the document's structure. Mehri *et al.* [14] segment a document into homogeneous regions by clustering texture features. Mehri *et al.* [15] compared different approaches such as Gabor filters, auto-correlation function, and Grey Level Co-occurrence Matrix (GLCM). They conclude that for clustering and segmentation of document images, Gabor features are preformed best. Wei *et al.* [6] address segmentation as a pixel-level classification. Each pixel is a vector of its coordinates and its color values. They use SVM, MLP, and GMM to classify these vectors. Similarly, Chen *et al.* [7] formulate layout analysis as a problem of pixel classification, where each pixel could belong to either periphery, background, text, or decoration. This method represents each pixel as a vector of its coordinates, color and texture. In addition, irrelevant features are removed by a feature selection algorithm Chen *et al.* [7] outperforms [6] by including more features such as texture information and applying feature-selection algorithm for better classification result.

B. Learned Features

In the past decade, learning features using CNN has become the dominant approach in the page-layout analysis domain.

Chen *et al.* [16] apply convolutional autoencoders for learning the features from pixels. These features are used to train an SVM for page segmentation. The classifier assigns to each pixel one of four classes: periphery, background, text block, and decoration. Later they applied SVM to classify superpixels instead of pixels [17] to reduce the classification time complexity. In addition, segmentation results are further refined in [18] using Conditional Random Field (CRF) that utilizes local and contextual information. These works [16], [18] consider feature extraction and classifying as two separate steps. On the other hand, [19] introduced an end-to-end method by combining feature learning and classifier training into one step.

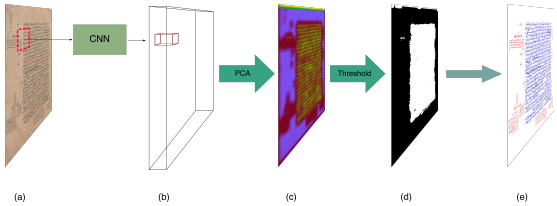

Fig. 1. Method flow: (a) Input page, (b) the resulting feature map by applying the trained CNN on the input page image using a sliding window, (c) a visualization of the first three principal components of the feature map, (d) applying a threshold on the first and second principal components of the feature map to extract the main-text mask, and (e) the final segmentation of the page, where foreground pixels inside the main-text mask are determined to be part of the main-text; otherwise, they are part of the side text

Recently, Kurar *et al.* [20] and Alaasam *et al.* [21] trained Fully Convolutional Network (FCN) and siamese neural network, respectively, to apply page segmentation. Both Kurar *et al.* [20] and Alaasam *et al.* [21] reported their results on the same dataset that we use for evaluation.

III. Method

Our method is composed of two main steps: feature extraction and segmentation. Feature extraction is a crucial step in any layout analysis algorithm. We delegate this step to a CNN trained as a branch of a siamese network, which is then used to extract features from patches in a given page. The siamese network is trained using patches prepared according to multiple strategies. We apply principal component analysis to the feature map and use the first and the second principle components to guide classifying the map into two categories: main text and side notes.

A. Data preparation

Data preparation consists of generating patches of the size 200×200 pixels, cropped randomly from document images and labeling. Every pair of patches are labeled either similar or different based on a set of principles we discuss below. Patch size is estimated as four times the average character height in the input document images. Without loss of generality and by analogy with distance, we label similar pairs of patches by zero and different pairs by one. We use four strategies to generate pairs of image patches with labels. One of them is for similar pairs of patches and the remaining three are for different pairs. The principles used to generate and label the patches are dataset independent and generalize to other datasets with heterogeneous text line-heights.

Next we discuss the four strategies to generate pairs of image patches with their labels.

1) Patches similar by spatial proximity: Patches are labeled by a simple principle, neighbouring patches are similar [22]. Given a document image we randomly sample a first patch, p_1 and an arbitrarily second patch, p_2, from the eight possible neighbouring locations around p_1 (see illustration in Fig. 2). In order to avoid trivial solutions, we perturb the location of p_2 by a quarter of the patch's height. Naturally, some neighbouring patches are not similar (e.g. patches located between main and side text). However, such patches are rare enough relative to similar neighbouring patches to be considered as noise.

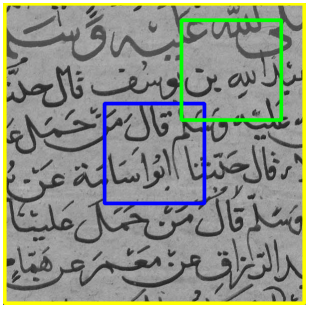

Fig. 2. Neighbouring patches are sampled and their relative locations are randomly perturbed. The first patch (blue) and the second patch (green) within the eight neighbouring area (yellow) of the first patch.

2) Patches different by average component sizes: Given randomly cropped two image patches, let h_i and w_i be the average component height and width of patch i, respectively, where $i \in \{1, 2\}$. Our algorithm iteratively sample, at random, pairs of patches until the similarity score s_1 satisfies the following condition:

$$s_1 = \frac{\min(h_1 \times w_1, h_2 \times w_2)}{\max(h_1 \times w_1, h_2 \times w_2)} < 0.5 \quad (1)$$

In a loosely manner this strategy generates pairs of patches where one from main text area and the other from side text area (Fig. 3). This is based on the assumption that the side text is written in a relatively small and restricted margins on the page, resulting in text with smaller font size. Therefore, the average component's height and width in side text area are relatively less than the average component's height and width in main text area.

3) Patches different by number of foreground pixels: Due to the font size difference between main and side text, the number of foreground pixels in side text area is relatively less than the number of foreground pixels in main text area. This assumption is used in this strategy to differentiate between patches from main text area and patches from side text areas.

Given randomly cropped two image patches, let a_i be the number of foreground pixels in patch i, where $i \in \{1, 2\}$. The algorithm continues selecting two random patches until the similarity score s_2 satisfies the following condition:

$$s_2 = \frac{\min(a_1, a_2)}{\max(a_1, a_2)} < 0.5 \quad (2)$$

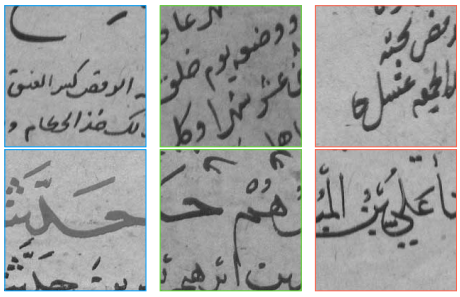

Fig. 3. Every column shows a pair of patches different by average component height and width. Such pairs train the machine to discriminate between the side text (above patches) and the main text areas (below patches).

In a loosely manner this strategy generates pairs of patches where one from main text area and the other from side text area, as illustrated in Fig. 4.

Fig. 4. Every column shows a pair of patches different by the number of foreground pixels. Such pairs train the machine to discriminate among the side text (above patches) and the main text areas (below patches).

4) Patches different by background area: A significant difference between background areas and text areas exists often in document images. This strategy iteratively sample pair of patches at random until one of the patches is from background area and the other from text area, as shown in Figure Fig. 5. We assume a patch belongs to a background area if more than its half belongs to a background area.

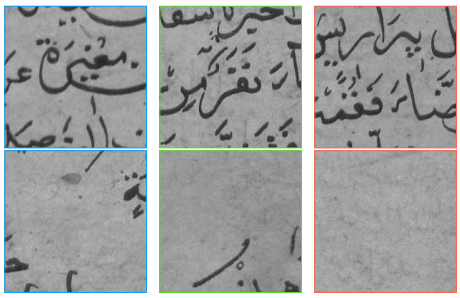

Fig. 5. Every column shows a pair of different patches. In a loosely manner, either of patches in each pair contain background area or foreground area. Such pairs train the machine to discriminate the text areas from the background areas.

B. Siamese network

We train a Siamese network with two identical CNN branches. The input is a pair of image patches of size 200 $\times$

200. The CNN branches extract representations of the input patches. Subsequently, these representations are concatenated and fed into fully connected layers in order to classify whether the two image patches are similar or different (further details are given in Section IV.)

C. Feature extraction

We use one of the branches of the trained siamese network for feature extraction. This branch takes a patch of size 200×200 and applies a number of convolutional layers followed by two fully connected layers and outputs a feature vector of size 512, as shown in Figure Fig. 6.

In the feature extraction step, a sliding window is used to extract a feature vector for each pixel in the input image using the CNN branch of the trained siamese network. As can be seen in Fig. 1, the feature extraction step outputs a feature map of size $w \times h \times 512$, where w and h are the width and height of the input image respectively.

D. Segmentation

The obtained feature map is used to guide the segmentation of the page into main and side text regions. The construction of the feature map aims at representing the two segments differently to simplify the segmentation procedure.

We have investigated applying PCA on the feature map and study (visualize and analyze) the resulting subspace. The first and the second principal components lead to a good indication of main text location, where the values of the first and the second principal components are higher for main text areas than side text areas. Therefore, we thresholded the feature map based on the first and second principal components to segment the main-text; i.e., a pixel p in the image is denoted main-text if the following condition holds:

$$PC_1(p) < T_1 \text{ and } PC_2(p) < T_2$$

where $PC_i(p)$ is the i'th principal component of the feature vector at pixel p and T_1, T_2 are predefined thresholds based on experimental results.

The network learns to extract meaningful information about the patches, such as text orientation, number background and foreground pixels, and connected component statistics. As a result, it extracts similar features from main text patches which are different from those extracted from side text patches, and similar features from side text patches which are different from those extracted from main text patches. We searched for a scheme to reduce the dimensions of the feature space to two while maintaining the distances between the data points. Since PCA does this well, we adopted it for dimension reduction. We have found that the first two components provide good results and the segmentation (int main and side text) is carried out using a simple threshold.

IV. Experiments

In this section we present the dataset we adopted for training and test, discuss training procedure, and analyse the obtained results.

A. Dataset

We have choose to evaluate our approach using the dataset presented by Bukhari *et al.* [3]. The dataset consists of 38 handwritten document images from 7 different historical Arabic Books. It is split as follows: 28 documents for training and the remaining 10 images are used for testing. The main-text and the side-text are labeled in each document in the dataset. To train the Siamese network we use 24 documents from the training set and the remaining 6 documents are used for validation.

B. Training

We built the Siamese network's branches similar to the Alexnet [23] model and through experiments we tune the hyperparameters to fit our problem. The final architecture consists of two CNN branches, each one has five convolutional layers as shown in Fig. 6. Dotted lines indicate identical weights, and the numbers in parentheses represent the number of filters, filter size, and stride. All convolutional and fully connected layers are followed by ReLU activation functions, except fc5, which feeds into a sigmoid binary classifier. The learning rate is 0.00001 and the optimizing algorithm is ADAM.

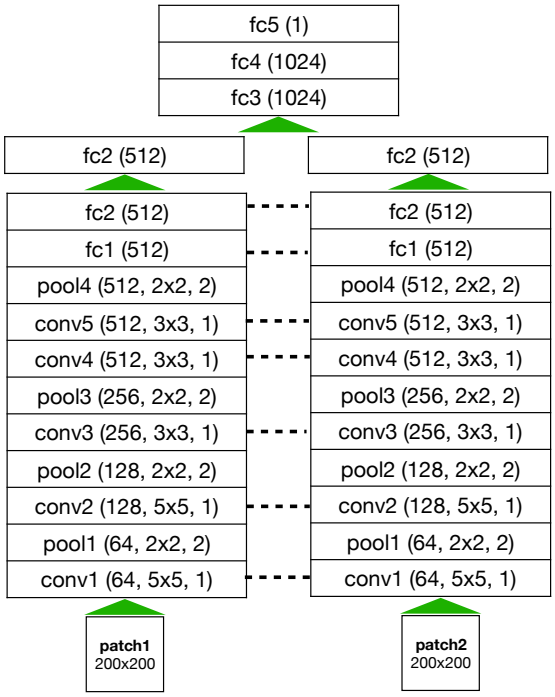

Fig. 6. Siamese architecture for classifying pairs as similar or different. Dotted lines stand for identical weights, conv stands for convolutional layer, fc stands for fully connected layer and pool is a max pooling layer.

We trained this model from scratch using $60,000$ pairs with balanced classes and reached a validation loss value of 0.30 after 11 epochs (Fig. 7). When the training is done, we cut out a branch of the Siamese network to be used for feature extraction.

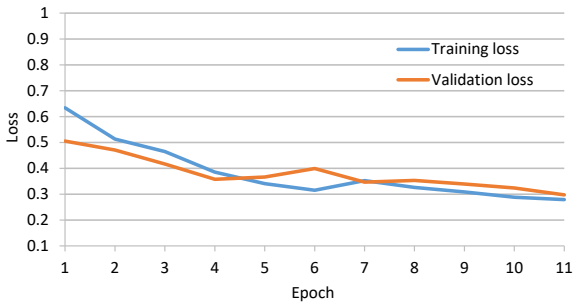

Fig. 7. Loss over the epochs of model training.

C. Results

We applied our method to segment the pages in the test set of the dataset into main and side text regions. The non-binarized images were used in the feature extraction step of the method. Extracting the feature vector for every possible patch in the image is expensive time-wise. Therefore, we used a sliding window with a step size of 50 pixels resulting in a feature map with dimensions smaller than the original image. In order to match the original image dimension, the feature map was resized with bi-linear interpolation.

In Table I we compare the performance of the proposed method using F-measure against the layout analysis methods [3], [20], [21]. Note that those three methods uses labeled data to train a ML model, while the proposed method is trained in an unsupervised manner. Our method outperformed both Bukhari *et al.* [3] and Kurar *et al.* [20] on both the main-text and the side-text. While it outperformed Alaasam *et al.* [21] on the side-text, we obtained slightly lower results on the main-text. However, it worth noting that Alaasam *et al.*preformed post-processing on their results whereas we do not.

Fig. 8 shows an example runs of our method. The second row shows a visualization of the extracted feature map using the Siamese network's CNN. The feature map is visualized by mapping the first three principal components to the RGB channels of the image. The visualization show that the CNN were able to extract the meaningful features regarding the main and the side text.

TABLE I
COMPARISON OF F-MEASURE VALUES. [3] AND [21]'S RESULTS ARE WITH SUPERVISED LEARNING AND POST PROCESSING, [20]'S RESULTS ARE WITH SUPERVISED LEARNING AND WITHOUT POST PROCESSING WHEREAS OUR RESULTS ARE WITH UNSUPERVISED LEARNING AND WITHOUT POST-PROCESSING.

Method	Main text	Side text
Bukhari et al. [3]	95.02	94.68
Kurar et al. [20]	95.00	80.00
Alaasam et al. [21]	**98.59**	96.89
Proposed	98.56	**96.97**

V. CONCLUSION

This paper presents an unsupervised page segmentation method for hand-written document images. We train a Siamese network to discriminate between patches with different writing attributes. In addition, the network is trained that two neighboring patches are similar. Our method uses one of the CNN branches of the trained Siamese network to extract a feature map from hand-written document images. The main-text region is extracted based on the first and second principal components of the feature map, which is then used to segment the image into main and side text. We have shown that the proposed method is on par with the supervised state of the art page layout analysis of historical manuscripts in terms of performance. In future work, we plan to adapt this method for text line segmentation. In addition, we aim to expand on the idea of using established hand-crafted features to train deep learning networks to tackle other document analysis tasks in an unsupervised setting.

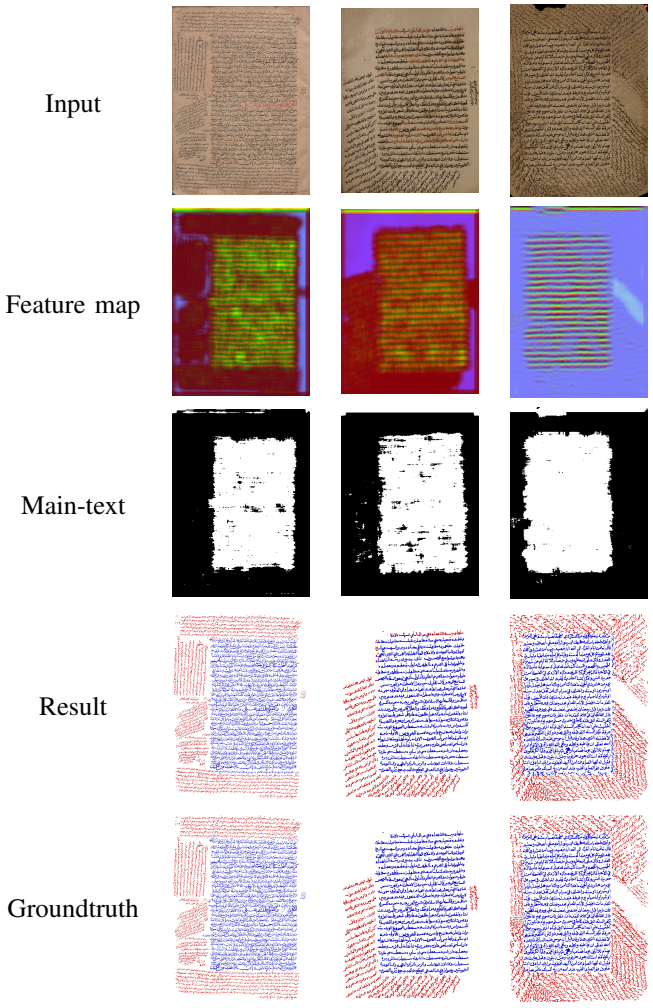

Fig. 8. Example runs from the test set. First row shows the input image from the test set. Second row, shows visualisation of the feature map. Third, shows mask of the main-text extracted from the feature map. Forth, shows the segmentation result of the method. And the last row shows the groundtruth from the dataset.

ACKNOWLEDGMENT

This research was partially supported by The Frankel Center for Computer Science at Ben-Gurion University.

REFERENCES

[1] A. Antonacopoulos and A. C. Downton, "Special issue on the analysis of historical documents," 2007.

[2] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 9, no. 2-4, pp. 123–138, 2007.

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

[4] S. S. Bukhari, M. I. A. Al Azawi, F. Shafait, and T. M. Breuel, "Document image segmentation using discriminative learning over connected components," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pp. 183–190, 2010.

[5] A. Garz, R. Sablatnig, and M. Diem, "Layout analysis for historical manuscripts using sift features," in *2011 International Conference on Document Analysis and Recognition*, pp. 508–512, IEEE, 2011.

[6] H. Wei, M. Baechler, F. Slimane, and R. Ingold, "Evaluation of svm, mlp and gmm classifiers for layout analysis of historical documents," in *2013 12th International Conference on Document Analysis and Recognition*, pp. 1220–1224, IEEE, 2013.

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

[8] H. Wei, K. Chen, R. Ingold, and M. Liwicki, "Hybrid feature selection for historical document layout analysis," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, pp. 87–92, IEEE, 2014.

[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 *2014 14th International Conference on Frontiers in Handwriting Recognition*, pp. 140–145, Sep. 2014.

[10] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM Journal of Research and Development*, vol. 26, pp. 647–656, 1982.

[11] T. Akiyama and N. Hagita, "Automated entry system for printed documents," *Pattern Recogn.*, vol. 23, p. 1141–1154, Oct. 1990.

[12] A. Antonacopoulos, "Page segmentation using the description of the background," *Comput. Vis. Image Underst.*, vol. 70, p. 350–369, June 1998.

[13] N. Journet, J.-Y. Ramel, R. Mullot, and V. Eglin, "Document image characterization using a multiresolution analysis of the texture: Application to old documents," *Int. J. Doc. Anal. Recognit.*, vol. 11, p. 9–18, Sept. 2008.

[14] M. Mehri, P. Héroux, P. Gomez-Krämer, A. Boucher, and R. Mullot, "A pixel labeling approach for historical digitized books," in *2013 12th International Conference on Document Analysis and Recognition*, pp. 817–821, Aug 2013.

[15] 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 2nd International Workshop on Historical Document Imaging and Processing*, HIP '13, (New York, NY, USA), p. 102–109, Association for Computing Machinery, 2013.

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

[17] K. Chen, C. 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 *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pp. 299–304, April 2016.

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

[19] K. Chen and M. Seuret, "Convolutional neural networks for page segmentation of historical document images," *CoRR*, vol. abs/1704.01474, 2017.

[20] B. K. Barakat and J. El-Sana, "Binarization free layout analysis for arabic historical documents using fully convolutional networks," in *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, pp. 151–155, IEEE, 2018.

[21] R. Alaasam, B. Kurar, and J. El-Sana, "Layout analysis on challenging historical arabic manuscripts using siamese network," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 738–742, IEEE, 2019.

[22] D. Danon, H. Averbuch-Elor, O. Fried, and D. Cohen-Or, "Unsupervised natural image patch learning," *Computational Visual Media*, vol. 5, no. 3, pp. 229–237, 2019.

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