Binarization Free Layout Analysis for Arabic Historical Documents Using Fully Convolutional Networks

Berat Kurar Barakat
Ben-Gurion University of the Negev
Beer-Sheva, Israel
Email: berat@post.bgu.ac.il

Jihad El-Sana
Ben-Gurion University of the Negev
Beer-Sheva, Israel
Email: el-sana@cs.bgu.ac.il

***Abstract*—We present a Fully Convolutional Network based method for layout analysis of non-binarized historical Arabic manuscripts. The document image is segmented into main text and side text regions by dense pixel prediction. Convolutional part of the network can learn useful features from the non-binarized document images and is robust to degradation and uncontrained layouts. We have evaluated the proposed method on a private dataset containing challenging historical Arabic manuscripts to demonstrate it effectiveness.**

I. Introduction

Libraries all around the world provide access to digital copies of historical documents in order to preserve their physical copies from deterioration. Digital documents are not easily explorable in their raw form but need to be transcribed further into machine readable text. Certainly manual transcription of large number of documents is not feasible in a reasonable time. Hence there is a significant need for reliable historical document image processing algorithms.

Page layout analysis is an important pre-processing step for the other document image processing algorithms. The analysis process consists of page segmentation and region classification. Page segmentation segments a document image into homogeneous regions. Region classification classifies the regions into predefined classes such as text, graphic and picture.

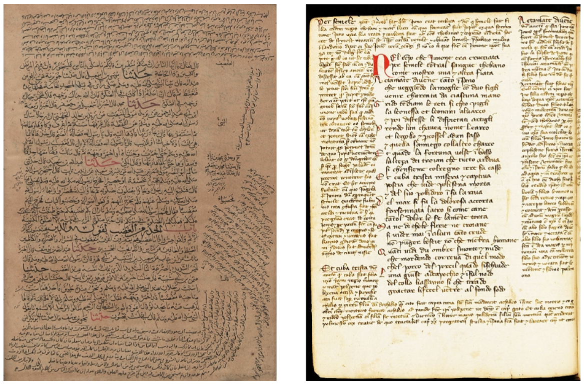


Fig. 1. Sample pages from Arabic dataset(left) and DIVA-HisDB(right).

In this paper we present a Fully Convolutional Network (FCN) based approach that segments side texts and main texts from non-binarized historical manuscripts with complex layout. It trains a FCN to predict the class of each pixel. FCN has been already used for page layout analysis by [1] on DIVA-HisDB dataset [2] and achieved state of the art results. DIVA-HisDB dataset does not cover the full range of difficulties present in the historical documents [3] in comparison to our dataset [4] with variety of degradation such as skewed and curved lines, bleed-through and noise (Figure 1). However, we used non-binarized document images in contrast to [4]. Binarization algorithms tend to introduce artifacts for historical documents. Therefore, we present an approach that is not dependent on foreground and background pixels. This approach achieves comparable results with [4]'s work although input documents are non-binarized.

In the following, Section II reviews the related work on page layout analysis, Section III describes the method, Section IV presents the dataset and experimental results and finally concluding remarks are given in Section V.

II. Related Work

Page segmentation algorithms work in either top-down or bottom-up manner. Top-down algorithms segment a whole page into regions. Bottom-up algorithms aggregate elements into regions. Elements can be pixels, connected components or patches. Patches are presegmented parts of document image according to the algorithm specific definition.

Primitive page layout analysis algorithms are based on a document structure assumption and applied to modern binary documents in a top-down manner. They are applicable to documents with Manhattan layout where the regions are rectangles with horizontal and vertical lines.

Wong et al. [5] use Run Length Smearing Algorithm (RLSA) [6] for page segmentation. RLSA links together the neighbouring black areas that are separated less than c pixels. RLSA is applied row by row as well as column by column to a document yielding two different bit maps. The two bit maps are then combined by a logical AND operation to produce the final segmented regions. The regions are then classified into text and non-text regions according to the measurements in a region such as total number of black pixels, horizontal black-white transitions, height of region and etc. This algorithm has relatively high computational cost of pixel-wise operations with the further potential problem of having to choose an appropriate *c*. Akiyama and Hagita [7] use projection profiles for page segmentation. A projection profile is obtained by counting the number of black pixels along a given direction. It is applied recursively to divide a document into smaller regions based on valleys in vertical and horizontal profiles which correspond to separators between rectangular regions. The regions are then classified into headline, text line and graphics regions according to a set of rules such as headline characters are larger than the text line characters, text line blocks are separated by blanks wider than the line spacing and etc. This algorithm can be slow as the separators have to be identified at each recursive iteration. Antonacopoulos and Apostolos [8] employ white tiles for page segmentation. A white tile is a rectangle that covers the longest possible white space in the horizontal direction. First the white space between text lines of same paragraph and inside the characters are joined by vertical smearing. Then white tiles are constructed sequentially from top to bottom. At last a region contour is represented by a cyclic list of white tile edges that border this region. This algorithm can identify regions even in the presence of severe skew since they will still be surrounded with space but does not classify the regions.

Segmentation based on clustering or classification is the most popular type of algorithms [9]. Below we include the ones for historical documents. They are applied to gray level or color documents in a bottom-up manner. They are applicable to historical documents with any layout, text orientation and text alignment.

Garz et al. [10] first compute interest points by means of Difference of Gaussian (DoG). DoG represents discriminative character parts such as junctions, arcs or endings. Then a Scale Invariant Feature Transform (SIFT) descriptor is calculated for every interest point. It is invariant to scale and rotation which allows the changing script size and orientation in manuscripts. It is robust to illumination changes which allows the variations in the background intensity. Finally the descriptors are classified by a Support Vector Machine (SVM) into initials, headings and text areas. Journet et al. [11], Mehri et al. [12] and Mehri et al. [13] use texture clustering for page segmentation. Texture is a low level feature in the image used to describe coarseness and regularity. Texture features can be extracted using autocorrelation function, Grey Level Co-occurrence Matrix (GLCM), Gabor filters and etc. In these works, a sliding window extracts texture attributes for each pixel. The pixels corresponding to homogeneous regions are then clustered together to form the segmented regions.

Bukhari et al. [4] consider the normalized height, foreground area, relative distance, orientation, and neighborhood information of the connected components as features. Then Multilayer Perceptron (MLP) is used to classify connected components into main body and side note texts. The labels of connected components are updated using the average of main body and side notes component probabilities within a selected region. This coarse to fine approach is outperformed by Asi et al. [14] who proposed a learning free approach to detect main text area. They first segment the main text area by using Gabor texture filter. Then refine the segmentation by minimizing an energy function that assigns higher probability to have same labels to closer pairs of components. Wei et al. [15] consider the segmentation problem as a pixel classification problem where each pixel is represented as a vector containing features based on colors of the image. SVM, Multi-Layer Perceptrons (MLP) and Gaussian Mixture Models (GMM) are used to classify the pixels into periphery, background, text and decoration pixels. SVM and MLP generally outperformed GMM. Chen et al. [16] outperform this work by representing each pixel with more color and texture features such as color variance, smoothness, Laplacian, Local Binary Patterns, and Gabor Dominant Orientation Histogram and then removing irrelevant features by a feature selection algorithm. Wei et al. [17] perform a similar experiment with an improved feature selection algorithm which is combination of the greedy forward selection and the genetic selection. They showed that feature selection reduces the feature vector size and improves the performance with significant features.

Problem of extracting significant features is further studied by Chen et al. [18]. Instead of hand crafted features used in the above algorithms they use convolutional autoencoder. Convolutional autoencoder is an unsupervised learning method and learns feature extractor on a set of image patches randomly selected from the unlabeled training set. Learned feature extractor is then used to train a SVM which can classify the pixels into periphery, background, text block, and decoration pixels. They achieved superior performance compared to their previous method [16]. Wei et al. [19] reduce the feature dimension by a feature selection algorithm based on [17] and increased the classification accuracy of this method.

III. Method

Conventional methods for rectangular layout analysis are not proper for segmenting manuscripts with complex layout. Historical Arabic manuscripts usually contain skewed and curved side notes with non-uniform patterns. We used FCN for segmenting side text and main text in Arabic documents with complex layout. FCN has made great improvements in object segmentation field [20]. It is an end to end segmentation framework that extracts the features and learns the classifier function simultaneously.

A. FCN architecture

The FCN architecture (Figure 2) we used is based on the FCN proposed for object segmentation [20]. First five blocks follow the design of VGG 16-layer network [21] except the discarded final layer. This is a conventional Convolutional Neural Network (CNN) and is called the encoder part of the FCN. Through the encoder, input image is downsampled

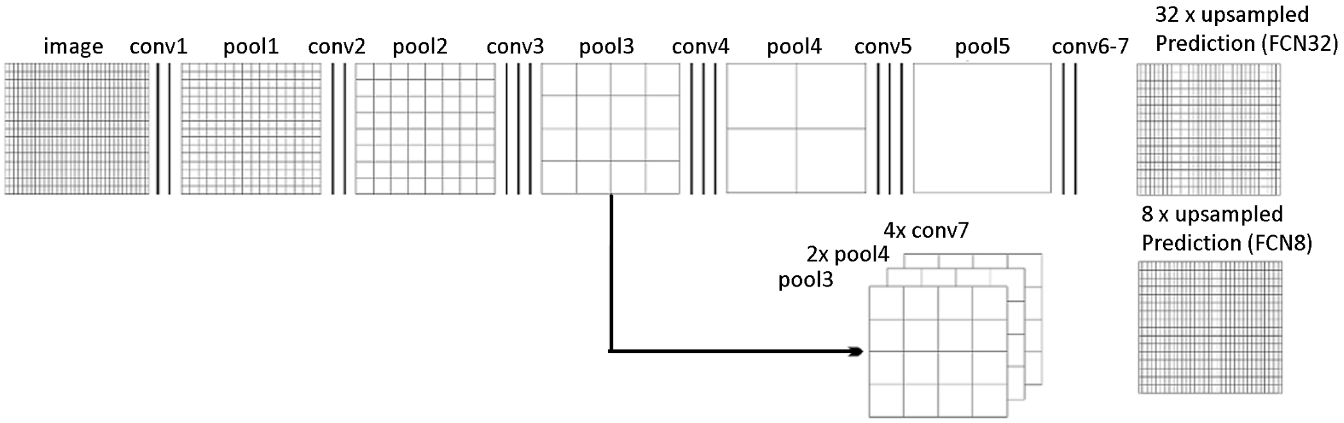


Fig. 2. The FCN architecture. Pooling and prediction layers are shown as grids that show relative coarseness. Convolutional layers are shown as vertical lines. FCN32 upsamples the final layer back to input size in a single step. FCN8 4 times upsamples the final layer, 2 times upsamples the pool4 layer and combine them with pool3 layer to upsample to input size.

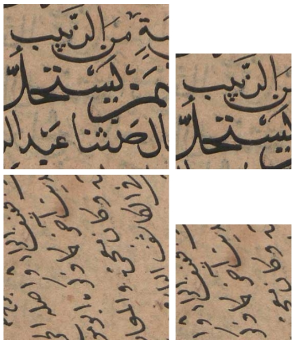


Fig. 3. 224×224 patches and 320×320 patches. 224×224 patches does not cover more than 2 main text lines and more than 3 side text lines in average.

and filters can see coarser information with larger receptive field. Then decoder part of FCN upsamples coarse outputs to dense pixels. Upsampling with a factor f is applying a convolution filter with a stride equal to $\frac{1}{f}$ and is called transpose convolution. Upsampling filters are also learned during the training.

We experimented with two kinds of FCN. FCN32 upsamples the final layer of encoder back to input size in a single step, which limits the scale of detail in the predictions. Therefore we used FCN8 which combines final layer of encoder with lower layers with finer information (Figure 2). Default input size of VGG is 224×224, which does not cover more than 2 main text lines and 3 side text lines (Figure 3). To include more context we changed the input size to 320×320. We also changed the output channel to 3 which is the number of classes, main text, side text and background.

B. Pre-processing

We randomly generate 100.000 and 20.000 patches of 320×320 size for training and validation sets respectively. Random patches potentially increases the variance that can accelerate convergence [20]. Since most of the labels are background, we tried to handle unbalanced labels by creating another dataset in the same way but eliminating the patches with density of background labels greater than 0.83. During the prediction phase the network manages to overcome the edge effect. It rarely fails to classify pixels near the edges. Therefore we didn't employ any post-processing. It is worth to note that during prediction of test patches, marginal regions that are less than patch size were ignored.

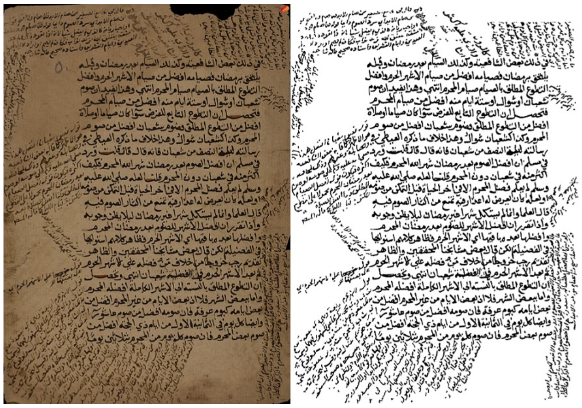


Fig. 4. Input to the proposed method is non-binarized images whereas [4] used binarized images.

IV. Experiments

A. Dataset

We use a dataset from the work of Bukhari et al [4]. It consists of 38 document images from 7 different historical Arabic books. 28 samples are for training and the remaining 10 samples are for testing. Main text and side text are labeled in pixel level. Although the problem is to segment side text from main text, our method also segments the background pixels since we used non-binarized images (Figure 4). In our experiments we used a train set of 24 samples and test set of 8 samples because we could not gather the non-binarized versions of 6 samples.

B. Metrics

We evaluate the segmentation accuracy by F-measure metric to compare our results with Bukhari et al's [4]. F- measure combines precision and recall values into a single scalar. A perfect precision score of 1.0 means that no pixel was falsely predicted but says nothing whether all relevant pixels were predicted. A perfect recall score of 1.0 means that all relevant pixels were predicted but says nothing whether how many false predictions were done. F-measure guarantees that both values are high.

Let n_{ij} be the number of pixels of class i predicted as class j and let True Positive (TP), False Positive (FP) and False Negative (FN) is defined as following:

$$\mathbf{TP} = \sum_i n_{ii}$$

$$\mathbf{FP} = \sum_i n_{ij}$$

$$\mathbf{FP} = \sum_i n_{ji}$$

Then precision and recall is calculated according to the following equations:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Using the precision and the recall scores F-measure is calculated with the following equation:

$$Fmeasure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

C. Training

We train by Stochastic Gradient Descent (SGD) with momentum equals to 0.9 and learning rate equals to 0.001. We initialize VGG with its publicly available pre-trained weights. We first trained on the randomly sampled dataset until overfitting. Then using the output weights we further trained on the dataset of patches with lower density of background labels. We continue training on train set until best F1 measure on validation set. All the experiments are conducted on Keras [22] and run on a single Nvidia 1080GTX.

D. Results

We first made an experiment with FCN32. The qualitative results were very coarse as expected but convergence was faster. Then we made an experiment with FCN8. The qualitative results were with very fine details (Figure 5). Table I shows the F-measure of each test sample. Proposed method achieved outperforming results on the first 6 test samples and poor results in the 7th and the 8th test samples, specially in the side text class. We argue that this was due to less number of training samples from the same book of these test samples. As shown in Table I training set contains 3 pages from the book of 7th test sample and 1 page from the book of 8th test sample. Furthermore related training samples has smaller ratio of side text to main text (RSM) then the RSM of test samples (Table II). Figure 6 shows qualitatively that model was trained on a small amount of side text in relative to the amount of side text in the test sample.

TABLE I
F-MEASURES ON EACH TEST SAMPLE AND NUMBER OF TRAIN SAMPLES FROM THE CORRESPONDING BOOK OF THE TEST SAMPLE.

Sample	Main Text	Side Text	#Train samples
1	0.99	0.98	20
2	0.99	0.98	20
3	1.00	1.00	20
4	1.00	1.00	20
6	0.99	0.95	20
7	0.85	0.10	3
8	0.83	0.51	1

TABLE II
RSM OF THE 6TH AND 7TH TEST SAMPLES, IN TRAIN SET AND TEST SET. PROPOSED METHOD DID NOT PERFORM WELL ON THE 6TH AND 7TH TEST SAMPLES DUE TO SMALL RSM IN TRAIN SET.

Sample	RSM in train set	RSM in test set
7	0.59	0.90
8	0.13	0.37

TABLE III
COMPARISON WITH F-MEASURES

	Main Text	Side Text
Bukhari et al [4]	**0.95**	**0.95**
Proposed method	**0.95**	0.80

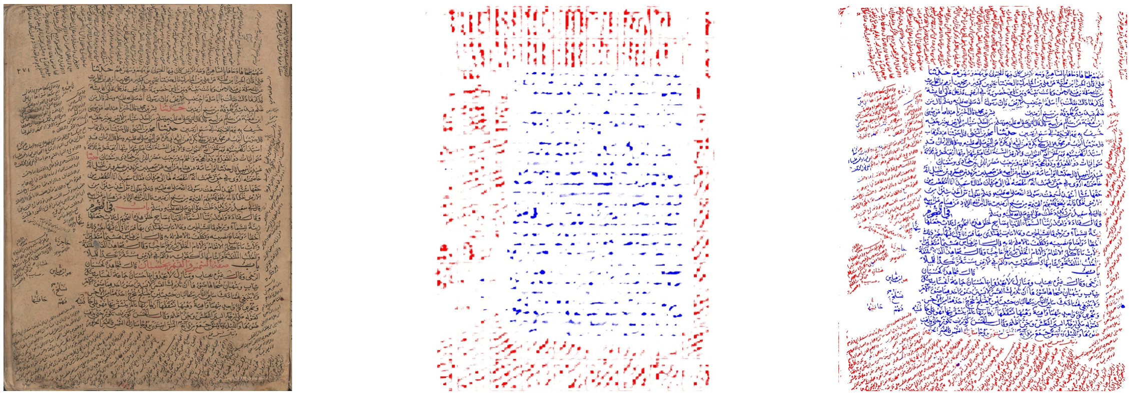
Table III shows the performance of our method compared with Bukhari et al [4]. We have to take into account that their method was tested on 10 samples whereas our one was tested on 8 of them. Our test set was completely blind because our stopping criteria was based on validation set whereas they did not use a validation set and stop training by observing performance on the test set. They use a post processing method called relaxation labeling which also yields overfitted results to the test set.

V. CONCLUSION

This paper presents a method for page layout analysis of historical Arabic manuscripts. It segments main text and side text regions using FCN. Our method is aimed for historical document images since convolutional part of FCN is robust to degradation and complex layout. FCN8 with skip features from early layers of the network yields to finer segmentation. Number of training samples is an important factor in the prediction performance. Prediction performance on a test sample is related to the number of train samples from the same book of the test sample. Competitive performance on the private Arabic manuscript datasetis achieved which validates the proposed method.

ACKNOWLEDGMENT

Authors would like to thank Abed Asi for his helps in preparing the dataset. This research was supported in part by



(a) Input

(b) FCN32 output

(c) FCN8 output

Fig. 5. Input image and its predictions with FCN32 and FCN8 networks. FCN32 output is very coarse whereas FCN8 output is finer.

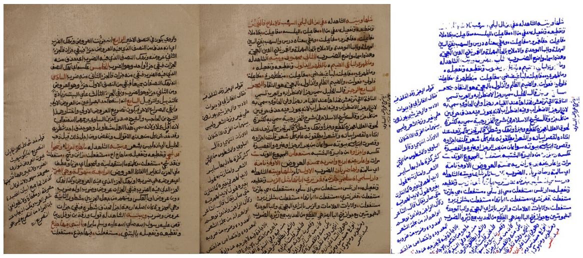


Fig. 6. Model was trained on one training sample (left) with small amount of side text in relative to the amount of side text in test sample (middle). Resultantly prediction performance in the side text class was poor (right).

Frankel Center for Computer Science at Ben-Gurion University of the Negev.

References

[1] Y. Xu, W. He, F. Yin, and C.-L. Liu, "Page segmentation for historical handwritten documents using fully convolutional networks," in *Document Analysis and Recognition (ICDAR), 2017 15th International Conference*. IEEE, 2017.

[2] 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.

[3] T. Grüning, R. Labahn, M. Diem, F. Kleber, and S. Fiel, "Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents," *arXiv preprint arXiv:1705.03311*, 2017.

[4] 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.

[5] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM journal of research and development*, vol. 26, no. 6, pp. 647–656, 1982.

[6] L. Abele, F. Wahl, and W. Scheri, "Procedures for an automatic segmentation of text graphic and halftone regions in document," in *Proc. 2nd Scandinavian Conf. on Image Analysis*, 1981, pp. 177–182.

[7] T. Akiyama and N. Hagita, "Automated entry system for printed documents," *Pattern recognition*, vol. 23, no. 11, pp. 1141–1154, 1990.

[8] A. Antonacopoulos, "Page segmentation using the description of the background," *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 350–369, 1998.

[9] S. Eskenazi, P. Gomez-Krämer, and J.-M. Ogier, "A comprehensive survey of mostly textual document segmentation algorithms since 2008," *Pattern Recognition*, vol. 64, pp. 1–14, 2017.

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

[11] N. Journet, J.-Y. Ramel, R. Mullot, and V. Eglin, "Document image characterization using a multiresolution analysis of the texture: application to old documents," *International Journal on Document Analysis and Recognition*, vol. 11, no. 1, pp. 9–18, 2008.

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

[13] 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*. ACM, 2013, pp. 102–109.

[14] 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.

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

[16] 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.

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

[18] 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.

[19] H. Wei, M. Seuret, K. Chen, A. Fischer, M. Liwicki, and R. Ingold, "Selecting autoencoder features for layout analysis of historical documents," in *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*. ACM, 2015, pp. 55–62.

[20] 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.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[22] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.