Line and Word Segmentation of Arabic handwritten documents using Neural Networks

Ahlem Belabiod, Abdel Belaïd

Line and Word Segmentation of Arabic Handwritten Documents using Neural Networks

Ahlem Belabiod and Abdel Belaïd
Universitée de Lorraine - LORIA

November 5, 2018

Abstract

Segmenting documents into lines and words is a very critical step before the recognition task. It is even more difficult for ancient and calligraphic writings, as is often the case in Arabic manuscript documents. In this work, we propose a new attempt to segment documents into lines and words, using deep learning. For line segmentation, we use an RU-net which allows a pixel-wise classification, thus separating pixels of lines from the background pixels. For segmenting lines into words, not having a ground truth for the word segmentation (at the image level), we use the line transcription to find the words. A BLSTM-CTC is used to achieve this mapping directly between the transcription and line image, without any segmentation. A CNN precedes this sequence to extract the features and feeds the BLSTM with the essential of the line image. Tested on KHATT Arabic database, the system achieves good performance that is of the order of 96.7% correct lines and 80.1% correct words.

0.1 Introduction

The processing of handwritten documents for their recognition remains a big challenge. The cursive and ever-changing nature of handwritten lines makes segmentation into lines and then into words usually necessary steps. Among the difficulties encountered in this type of document, we find the irregular inclination of the lines, the occurrence of ascenders and descenders connecting the lines, and the change of style of writing according to the writer. The writing of words in several subwords introduces confusion in the separation of words between them, and finally the presence of diacritics adds an additional difficulty.

The works related to line segmentation can be divided in three categories: 1) top-down methods, 2) bottom-up methods and 3) machine learning methods. The two first methods generally require prior knowledge on the documents, such as line inter-spaces, document orientation, etc. Therefore, systems must combine many kinds of image processing methods to take into consideration all possible features.

In a particular way, top-down methods process the whole image and iteratively subdivide it into smaller blocks to isolate the desired part [1]. These methods include image processing techniques like projection profile, Hough transform or Gaussian filters.

Bottom-up methods, meanwhile, deal with noise problems and writing variation [2]. The connected components (CCs) based methods are the most popular methods used in this approach [3, 4]. These methods generally use simple rules, analyzing the geometric relationships between neighboring blocks such as the distance or the overlap.

Conversely, machine learning methods treats the image as a whole without any prior knowledge. Some of the existing systems are end-to-end systems, with no further post or pre-processing [5, 6], while others use deep learning like one step among other processing [7, 8].

Often, line and word segmentation tasks are separated because of the difference between line and word features, so global processing for both tasks can be tough. Also, we noticed that deep learning is much less used in word segmentation than on line segmentation. The methods used in word segmentation are rather bottom-up methods, based on CC analysis [9, 10], structural feature extraction [11] or even both [12]. These approaches show interesting results on documents written in Latin or Germanic languages.

We propose a line segmentation system using an RU-net, and an end-to-end system for word segmentation, using a CNN (Convolutional Neural Network) followed by a BLSTM (bidirectional Long Short Term Memory) and finally a CTC function (Connectionist Temporal Classification) [13] that will automatically learn the alignment between text line images and the words in the transcription.

The paper is organized as follows: Section 0.1 gives an overview of some related works. Section 0.3 presents the proposed approaches for line and word segmentation. Section 0.4 reports the experimental results and section 0.5 concludes the work.

(a) (b)

Figure 1: Results of [15]: a) on their dataset(Al-Majid), b) on our dataset.

0.2 Related Work

In this section, line segmentation and word segmentation previous works will be presented.

0.2.1 Line Segmentation

As an example of bottom-up methods, Cohen & al. [14] starts their system by applying a multi-scale multi-scale anisotropic second derivative of Gaussian filter bank to enhance text regions, then apply a linear approximation to merge connected components from the same line using a K_{NN} function. The authors proposed a generic system that may be used for any language, and obtained interesting results on Hebrew with 98% for text region detection.

As an example of top-down methods, the authors in [15] use a seam carving method. They first compute medial seams on the lines, using a projection profile matching approach, then compute separating seam using a modified version of seam carving procedure [16]. Their method showed good results on the Arabic dataset they used (99.9%), but coarse results on our dataset (see Figure 1).

Among the several works using deep learning for line segmentation, Renton & al. [5] use a modified version of the FCN (Fully Convolutional Network) [17]. In [17], the authors use FCN for semantic segmentation and introduce "skip" steps between first and last layers to avoid the coarse results due to many pooling layers, but the authors in [5] argue that this is not sufficient for the task of line segmentation which needs to be more precise. Thus, Renton & al. [5] propose a new architecture where the layer convolution + max pooling is replaced by dilated convolutions. Their network has been trained on x-height labeling, reaching up to 75% of F-measure on the cBad dataset [18].

Grüning & al. [7] use a two stages method to extract lines from document images. The first stage is an ARU-net which is a U-net extended with an attention model (A) and a residual structure (R). The U-net is a variant of FCN [19] that introduces shortcuts between layers of the same spatial dimension, so we can have features from higher level and reduce the vanishing gradient problem. The authors add residual blocks [20] to the U-net and an attention model that works simultaneously with the RU-net at multi-scale level. The output of the network is two maps, one related to the detected baselines and

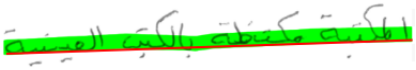

Figure 2: X-height (green) and baseline (red) of a text line.

another to the beginning and end of the lines. In the second stage, they calculate a set of superpixels from the baseline map, then they estimate the state of these superpixels by computing their orientation and interline distance. Finally, using a Delaunay neighborhood system and calculating some functions like the projection profile, the data energy and the data cost, they find superpixels clusters that represent the lines. Their system reaches 95% of good detection of lines on the whole cBad dataset, but the many processing and computations used, especially on the second stage, makes it a heavy procedure.

0.2.2 Word Segmentation

Papavassiliou & al. [1] calculate an SVM-based gap metric for adjacent CCs within each text line, then apply a threshold to classify gaps as "within" and "between" words. They tested their approach on the datasets from the ICDAR07 Handwriting Segmentation Contest [21] and reached an F-measure of 93%. Al Khateeb & al. [9] looked at the Arabic word segmentation problem with a component-based method where they analyze CCs with the help of baselines, and reach 85% of correct rate segmentation. In [22], Al-Dmour & al. propose a method based on two spatial measures: CC length and gaps between them. Lengths are clustered to separate between the groups of letters, subwords and words. Gaps are clustered to figure out whether the gap occurs "between-words" or "within-a word". These measures are clustered using a Self-Organizing Map (SOM) algorithm [23]. The approach has been tested on the AHDB dataset [24] and achieved 86.3% of correct extraction rate.

0.3 Proposed Approach

0.3.1 Global Presentation

Inspired by [7], we choose to use an RU-net for our line segmentation, on an x-height labeling, followed by a simple post-processing that extracts baselines.

There are many representations in the literature for what is a text line in the line segmentation, like baselines or bounding boxes. The x-height represents the area corresponding to the core of the text without ascenders and descenders and seems to be a suitable text line representation for text recognition, while the baseline represents the main orientation of a text line and is mainly used for the evaluation. An example of x-height and baseline labeling is provided in the Figure 2.

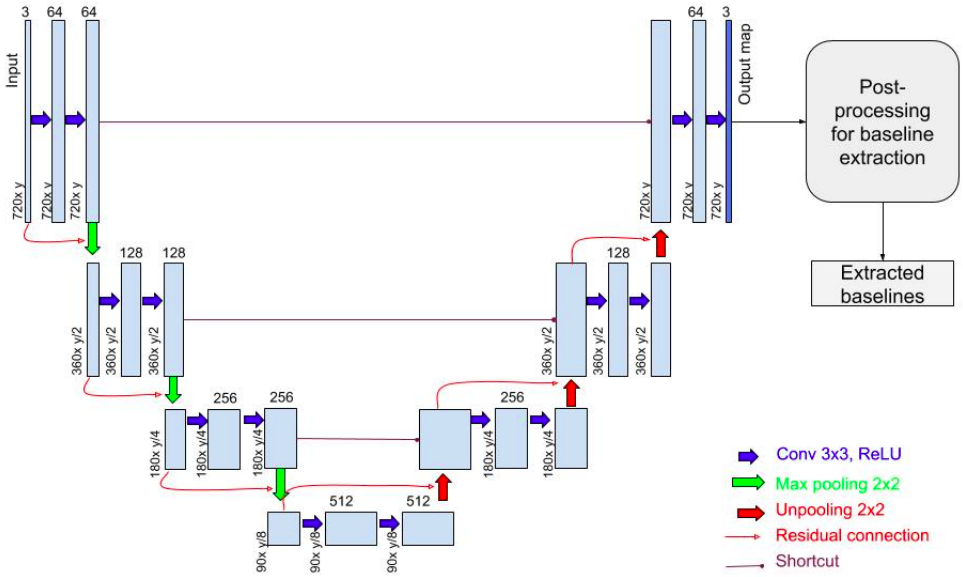

Figure 3: Structure of the RU-net. Blue boxes denote multi-channel feature maps. The number of channels is provided on the top of each box while sizes are provided on the bottom-left of the boxes.

Figure 3 shows the structure of our system. The output of the RU-net is a prediction map with the same size of the input. The network is then followed by a post-processing step that extracts the baselines from the x-heights.

For word segmentation task, we chose a CNN followed by a BLSTM (forward and backward) and finally a CTC decoder. Figure 4 depicts the structure of the network.

0.3.2 Detailed Description of the Approach

Line Segmentation:

Our goal here is first to extract the x-heights of text lines, by providing a ground truth that separates 3 classes (background, paragraphs and lines' x-heights in each paragraph) as shown in Figure 8.b. From then on, our problem becomes similar to a semantic segmentation, which can be solved by an FCN [17].

The FCN structure can be described as a symmetric encoder/decoder network where the encoder part convolves and downsamples the input with a number of convolution+pooling blocks, then the decoder part upsamples the result by the same factor of downsampling. So the output will be a map with the same size of the input where each pixel has been assigned to a probability of belonging to a class.

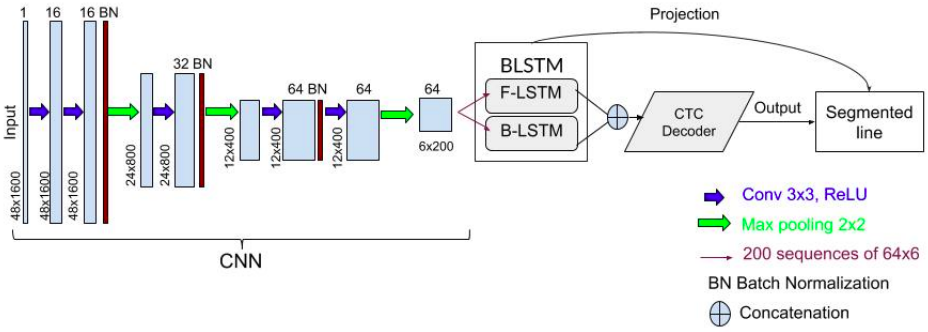

Figure 4: Proposed word segmentation network.

We chose an U-net as FCN model, extended with residual connections. The U-net [19] allows an easier combination of low level features and high level ones by introducing shortcuts between the same level blocks and the residual connections greatly reduce the vanishing gradient problem [20].

We provide, as an input, paragraph images resized to 720 pixels for the largest side and keep the same ratio between height and width. The images are resized so we can feed the network smaller images, thus reduce the memory footprint. The encoder part of our network is composed of 3 convolution + max-pooling blocks and a 4^{th} convolution layer, and the decoder part of 3 convolution + unpooling blocks.

This network is followed by a simple post-processing step on the output of the network (a map of 3 classes): A dilation is performed on the x-heights, so the over-segmented lines can be merged, then an erosion, to recover the original shape of x-heights. After that, polynomial regression is applied on the connected components (x-heights here), so the baselines can be computed. Although this post processing is well adapted to our case, it would not be for lines written on several columns, for example.

Word Segmentation:

The network is fed by line images extracted from the previous network. All images have a normalized size of 48×1600. To deal with the images, we chose to use a CNN, that will extract the most important features of the input. Every convolutional block is followed by a batch normalization [25] that greatly reduces the vanishing gradient problem and makes the use of dropout unnecessary [26]. The output of the CNN is then passed sequentially to a bidirectional LSTM (100 neurons for each LSTM) which is followed by a CTC function [13]. Let's consider that we have two classes (word: 1, space: 0). The ground truth provided is then the Unicode transcription of the text lines where each word is labeled by 1 and each space by 0, as shown in Figure 5, and the CTC decoder's output will be the sequence word-space found.

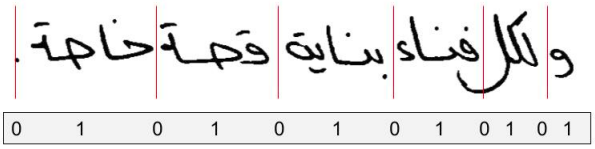

Figure 5: Ground truth provided to the CTC (1: word; 0: space).

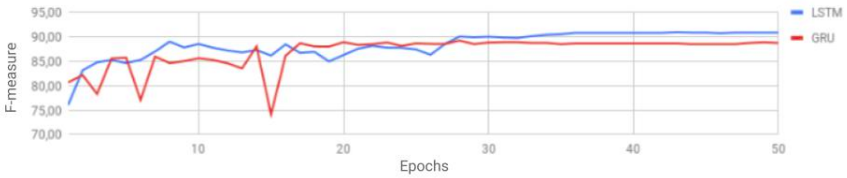

Figure 6: Obtained results using LSTM and GRU.

After the training, a projection of the probabilities of class *space* (output of the BLSTM) is made on the image.

Bidirectional LSTM: LSTMs and GRUs are the best known and efficient RNNs. While LSTM has 3 gates (input, forget, output) and a memory cell, GRU has only 2 gates (reset and update) and no memory cell, thus less parameters. No study has proven which one is better [27], and it all depends on one's case, so the 2 networks have been tested for comparison. Figure 6 shows the difference between LSTM's and GRU's accuracy through epochs, and LSTMs showed the better results.

On the other hand, the use of a BLSTM instead of a simple one may provide additional context for the network, so the learning will be faster and better.

CTC function: As there is no obvious alignment between the network's output and the ground truth, the CTC function is used to perform it automatically. Graves & al. [13] initially introduced the CTC to address the problem of alignment in speech recognition. Given the output of an RNN (a sequence of probabilities), the CTC loss function computes the probability of an alignment per each time-step using a dynamic programming algorithm. A beam search decoder is then used to extract the top paths, i.e. the paths with the greatest probability values.

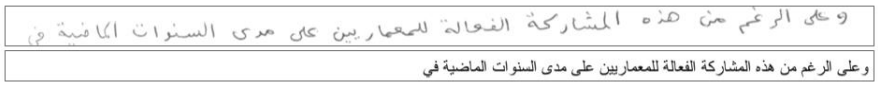

Figure 7: Example of a text line and it's transcription in the KHATT database.

0.4 Experiments & Results

0.4.1 KHATT Database

KHATT [28] is a database of handwritten Arabic images, written by 1000 persons from different age, gender and nationality. It provides 2 sets of 2000 short handwritten paragraphs. The first set groups the images of the same paragraph, written twice by each person. The second set groups images of free paragraphs, each person having also written 2 paragraphs. The database also provides line segmentation and a Unicode transcription for each line (see Figure 7).

0.4.2 Results & Discussion

Line Segmentation:

From the KHATT database, we labeled 50 paragraph images following the pattern shown in Figure 8.b, 35 for the training set and 15 for the testing set. For all our experiments, a learning rate of 10^{-4} and the Adam Optimizer [29] are used. The network has been trained for 100 epochs. Table 1 summarizes the parameter settings of the line segmentation architecture.

Image pre-processings:	50 paragraph images: 35 for training and 15 for testing. Images are normalized to a size of 720px for the largest side. No further processing.
RU-net:	See Figure 3 for the detailed number of filters, kernels size and pool size. Strides : 2. Dropout : 0.5 (after every convolutional layer).
Training settings:	Initial weights: 1.0 for the 3 classes. Initial learning rate: 10^{-4}. Optimizer : Adam. Initial number of epochs: 100.

Table 1: Parameter settings of the line segmentation architecture. The 4 nearest graph modeling further connected these prices to form a structure.

The metric used for the evaluation of the baseline detection is presented in [30] and computes the F-measure on the extracted baselines. Table 2 below gives the results of the machine learning based methods for line extraction,

Architecture	Dataset	F-measure on line extraction
[5]	KHATT	81%
	cBad	75%
[7]	cBad	95%
RU-Net(ours)	KHATT	96.71%

Table 2: Results on line extraction.

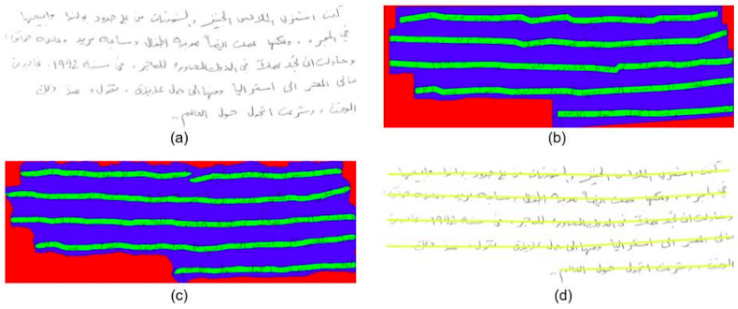

Figure 8: Results obtained for the line segmentation: a) the original image, b) the ground truth composed of three classes (background: red; paragraph: blue; x-height: green), c) the output of the RU-net, d) the final result after post-processing.

tested on KHATT dataset. The system proposed in [7] has not been tested, since the code for the second stage is not provided. The results in [5] are evaluated on the x-heights, while those in [7] and RU-net are evaluated on the baselines. Figure 8 shows the results obtained by the used system.

The results obtained by the used system are encouraging, especially since it is quite simple and requires less parameters and processing steps. Furthermore, we trained it on a small set, a bigger training set may provide better results, but having to label the images by hand could be the brake of this system.

Word Segmentation:

For our experiments, we used 5000 line images, 4800 for the training set and 200 for the testing set. A learning rate of 10^{-4} with an Adam optimizer are also used. Table 3 summarizes the parameter settings of the word segmentation architecture.

Figure 9 shows an example of the results obtained by our network and Table 4 summarizes the F-measure values of our approach and other works. Since the prediction is done on a downscaled image, the output needs to be upscaled to fit the original image. The F-measure was calculated manually by comparing

Image pre-processing:	5000 line images: 4800 for training and 200 for testing. Images are normalized to a size of 48×1600. No further processing.
CNN+BLSTM+CTC	See Figure 4 for detailed description of the CNN used, strides: 1, no dropout. The BLSTM has 100 neurons for each direction. See Figure 5 for an example of the ground truth provided to the CTC function. Beam size: 100.
Training settings:	Weights initializer: Xavier. Initial learning rate: 10^{-4}. Optimizer: Adam. Number of epochs: 100.

Table 3: Parameter settings of the word segmentation architecture.

Architecture	**Dataset**	**F-measure**
[9]	IFN/ENIT	85%
[22]	AHDB	86.3%
CNN+BLSTM+CTC	KHATT	80.1%

Table 4: Results of our approach.

the words on the transcription and the resulting segmentation. The most common segmentation errors are misplaced segmentations, like shown in Figure 10 where the segmentation is between sub-words.

Compared to other works on the Arabic word segmentation, ours gives less good measure but doesn't use any shape specification of the language, which is probably the reason of these results. Furthermore, the databases are not the same and KHATT may provide more difficult writings.

0.5 Conclusion

In this paper, line and word segmentation approaches based on deep learning models has been proposed. The line segmentation system uses an RU-net to extract x-heights from text images, then a post-processing step extracts baselines. The word segmentation approach uses a CNN with a BLSTM, then a CTC to find the alignment between the line transcription and the line image. The results of the two approaches are promising, but still need to be improved, especially the word segmentation approach. A future work would be to add some post-processing on the connected components for example, to correct the wrong segmentation.

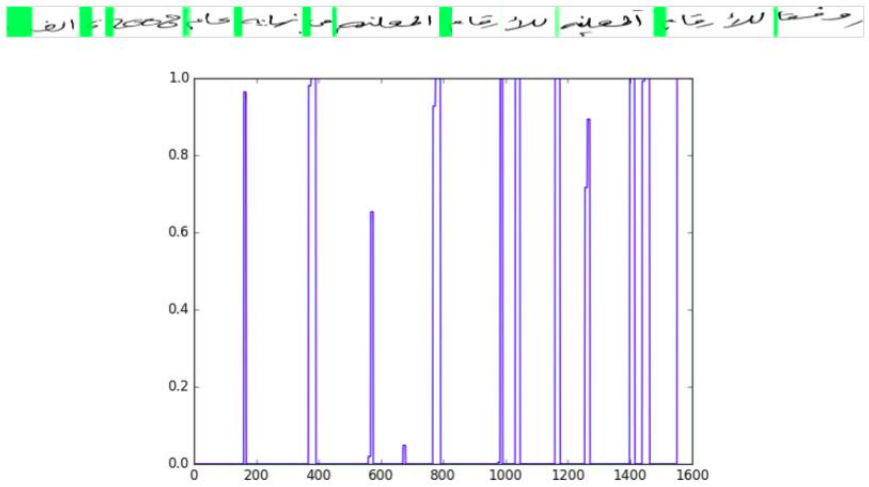

Figure 9: Results of the proposed approach. The lower image is the output of the BLSTM for the class *space (0)*, the 200 probabilities has been upscaled $\times 8$ to recover the original shape. The upper image is the segmented line after the projection of the probabilities.

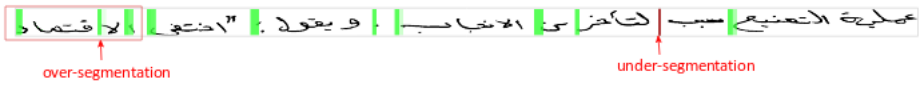

Figure 10: Example of wrong segmentations. The red box denotes one word and an over-segmentation is made on the sub-words.

Bibliography

[1] Papavassiliou, V., Stafylakis, T., Katsouros, V., Carayannis, G.: Handwritten document image segmentation into text lines and words. Pattern Recognition **43** (2010) 369 – 377

[2] Belaïd, A., Ouwayed, N. In: Segmentation of Ancient Arabic Documents. Springer London, London (2012) 103–122

[3] Boulid, Y., Souhar, A., Elkettani, M.Y.: Arabic handwritten text line extraction using connected component analysis from a multi agent perspective. In: 2015 15th International Conference on Intelligent Systems Design and Applications (ISDA). (2015) 80–87

[4] Rajput, G.G., Ummapure, S.B., Patil, P.N.: Text-line extraction from handwritten document images using histogram and connected component analysis. In: International Journal of Computer Applications (09758887). National conference on Digital Image and Signal Processing, DISP 2015. (2015) 11–17

[5] Renton, G., Chatelain, C., Adam, S., Kermorvant, C., Paquet, T.: Handwritten text line segmentation using fully convolutional network. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Volume 05. (2017) 5–9

[6] Chen, K., Seuret, M.: Convolutional neural networks for page segmentation of historical document images (2017) arXiv:1704.01474v2.

[7] Grüning, T., Leifert, G., Strauss, T., Labahn, R.: A two-stage method for text line detection in historical documents (2018) arXiv:1802.03345v1.

[8] Pastor-Pellicer, J., Afzal, M.Z., Liwicki, M., Castro-Bleda, M.J.: Complete system for text line extraction using convolutional neural networks and watershed transform. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS). (2016) 30–35

[9] AlKhateeb, J.H., Jiang, J., Ren, J., Ipson, S.S.: Component-based segmentation of words from handwritten arabic text. International Journal of Computer and Information Engineering **2** (2008)

[10] Louloudis, G., Gatos, B., Pratikakis, I., Halatsis, C.: Text line and word segmentation of handwritten documents. Pattern Recognition **42** (2009) 3169–3183

[11] Aouadi, N., Kacem-Echi, A.: Word extraction and recognition in arabic handwritten text. International Journal of Computing & Information Sciences **12** (2016) 17–23

[12] Elzobi, M., Al-Hamadi, A., Aghbari, Z.A.: Off-line handwritten arabic words segmentation based on structural features and connected components analysis. In: WSCG '2011: Communication Papers Proceedings: The 19th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision. (2011) 135–142

[13] A Graves, S.F., Gomez, F., , Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: International Conference on Machine learning. (2006) 369–376

[14] Cohen, R., Dinstein, I., El-Sana, J., Kedem, K.: Using scale-space anisotropic smoothing for text line extraction in historical documents. In Campilho, A., Kamel, M., eds.: Image Analysis and Recognition, Cham, Springer International Publishing (2014) 349–358

[15] Arvanitopoulos, N., Ssstrunk, S.: Seam carving for text line extraction on color and grayscale historical manuscripts. In: 2014 14th International Conference on Frontiers in Handwriting Recognition. (2014) 726–731

[16] Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. ACM Trans. Graph. **26** (2007)

[17] Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation (2016) arXiv:1605.06211.

[18] Diem, M., Kleber, F., Fiel, S., Grüning, T., Gatos, B.: ScriptNet: ICDAR 2017 Competition on Baseline Detection in Archival Documents (cBAD) (2017)

[19] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015) arXiv:1505.04597.

[20] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 770–778

[21] www.iit.demokritos.gr/bgat/HandSegmCont2007.

[22] Al-Dmour, A., Zitar, R.A.: Word extraction from arabic handwritten documents based on statistical measures. International Review On Computer and Software (IRECOS) **11** (2016)

[23] Kohonen, T.: The self-organizing map. Proceedings of the IEEE **78** (1990) 1464–1480

[24] Al-Ma'adeed, S., Elliman, D., Higgins, C.A.: A data base for arabic handwritten text recognition research. In: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition. (2002) 485–489

[25] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015) arXiv:1502.03167.

[26] Li, X., Chen, S., Hu, X., Yang, J.: Understanding the disharmony between dropout and batch normalization by variance shift (2018) arXiv:1801.05134v1.

[27] Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling (2014) arXiv:1412.3555v1.

[28] Mahmoud, S.A., Ahmad, I., Alshayeb, M., Al-Khatib, W.G., Parvez, M.T., Fink, G.A., Margner, V., Abed, H.E.: Khatt: Arabic offline handwritten text database. In: Proceedings of the 13^th International Conference on Frontiers in Handwriting Recognition (ICFHR 2012), Bari, Italy, IEEE Computer Society (2012) 447–452

[29] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014) arXiv:1412.6980.

[30] Grüning, T., Labahn, R., Diem, M., Kleber, F., Fiel, S.: Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). (2018) 351–356