Arabic Handwritten Documents Segmentation into Text-lines and Words using Deep Learning

Chemseddine Neche, Abdel Belaïd, Afef Kacem-Echi

Arabic Handwritten Documents Segmentation into Text-lines and Words using Deep Learning

Chemseddine Neche, Abdel Belaïd
Université de Lorraine - LORIA
54500 Vandoeuvre-Lès-Nancy, France
neche.chemseddine@gmail.com, abdel.belaid@loria.fr

Afef Kacem-Echi
University of Tunis, ENSIT-LaTICE, 5 Avenue Taha Hussein
Tunis 1008, Tunisia
afef.kacem@ensit.rnu.tn

***Abstract*—One of the most important steps in a handwriting recognition system is text-line and word segmentation. But, this step is made difficult by the differences in handwriting styles, problems of skewness, overlapping and touching of text and the fluctuations of text-lines. It is even more difficult for ancient and calligraphic writings, as in Arabic manuscripts, due to the cursive connection in Arabic text, the erroneous position of diacritic marks, the presence of ascending and descending letters, etc. In this work, we propose an effective segmentation of Arabic handwritten text into text-lines and words, using deep learning. For text-line segmentation, we used an RU-net which allows a pixel-wise classification to separate text-lines pixels from the background ones. For word segmentation, we resorted to the text-line transcription, as we have not got a ground truth at word level. A BLSTM-CTC (Bidirectional Long Short Term Memory followed by a Connectionist Temporal Classification) is then used to perform the mapping between the transcription and text-line image, avoiding the need of the input segmentation. A CNN (Convolutional Neural Network) precedes the BLST-CTC to extract the features and to feed the BLSTM with the essential of the text-line image. Tested on the standard KHATT Arabic database, the experimental results confirm a segmentation success rate of no less than 96.7% for text-lines and 80.1% for words.**

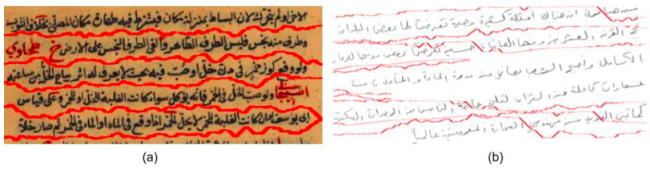
I. Introduction

Text-line and word segmentation is the process by which the basic entities in a text document image are localized and extracted. It is an important step for off-line handwritten text recognition because inaccurate segmentation will cause errors in the recognition step. However, locating text-lines and words in Arabic manuscripts remains a challenge. In fact, the Arabic manuscripts are considered to be more complex than other manuscripts written in other languages. This complexity comes first from handwritten text characteristics which can vary in writing style, size, orientation, alignment and where adjacent text-lines can be touching or overlapped, and second from the Arabic writing nature: cursiveness of the text, character overlapping, diacritic, variety of calligraphic, words are often divided into letters and sub-words and the spaces between them are variable.

The existing handwritten text-line segmentation methods can be categorized as top-down, bottom-up methods or machine learning methods. Top-down methods process the whole image and recursively subdivide it into smaller blocks to isolate the desired part [26]. They resort to projection profile, Hough transform, Gaussian filters and generally assume that gap between adjacent text-lines is significant and the text-lines are reasonably straight, which may not be faithful in handwritten texts. They are also known to be sensitive to the topological changes in handwritten documents. Bottom-up methods use simple rules, analyzing the geometric relationships between neighboring blocks such as the distance or the overlap. They have the merit to deal with noise problems and writing variation [7]. The common bottom-up method is based on the connected components [8], [28]. But, as most of top-down methods, bottom-up methods require prior knowledge about the documents, such as text-line inter-spaces, its orientation, etc. to guide the segmentation. Systems must therefore combine different image processing techniques to consider all possible features. Conversely, machine learning methods, which are free-segmentation, treat the image as a whole without any prior knowledge. During the last years the segmentation free methods, based on machine learning, has been used in different domains and achieved promising results. Some of the existing systems are end-to-end systems, with no further post or pre-processing [29], [9], while others use deep learning like one step among other processing [15], [27].

As text-line and word segmentation tasks are often separated because of the differences in text-line and word features, the global processing for both tasks can be tough. Notice also that deep learning is much less used in word segmentation than in text-line segmentation. The commonly used methods for word segmentation are rather bottom-up methods, based on connected components analysis [24], structural feature extraction [4] or even both of them [13]. They show interesting results on documents written in Latin or Germanic languages. On the other hand, deep learning based semantic segmentation methods have been providing state-of-the-art performance in the last few years. More specifically, these techniques have been successfully applied to image classification, segmentation, and detection tasks. One deep learning technique, U-Net, has become one of the most popular for these applications.

In this work, we propose a text-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 a CTC function (Connectionist Temporal Classification) [14] which is used to automatically learn the alignment between text-line images and the words in the transcription. Note that



(a) (b)

Fig. 1. Results of [6]: a) on their dataset(Al-Majid), b) on our dataset.

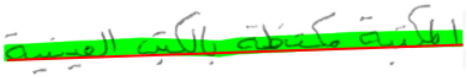


Fig. 2. X-height (green) and baseline (red) of a text-line.

the proposed system can overcomes the problem of overlaps between adjacent text-lines, words or sub-words.

The paper is organized as follows: Section II gives an overview of some related works. Section III presents the proposed system for the text-line and word segmentation. Section IV reports the experimental results and section V concludes the work.

II. Related Works

A wide variety of text-line and word segmentation methods have been proposed in the literature. For text-line segmentation, Cohen & al. [11] used a bottom-up method. Their system starts by applying a multi-scale anisotropic second derivative of Gaussian filter bank to enhance text regions. It then applies a linear approximation to merge connected components of the same text-line using the K-Nearest Neighbors 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.

In [6], the authors used rather a top-down method, called seam carving method. They first compute medial seams on the text-lines, using a projection profile matching approach. Then, they compute separating seam, using a modified version of seam carving procedure [5]. Their method showed good results on the Arabic dataset they used (99.9%), but coarse results on our dataset (see Figure 1).

Renton & al. [29] used a modified version of the FCN (Fully-connected Convolutional neural Network) [23], as deep learning based method for text-line segmentation,. The FCN, used for semantic segmentation and "skip" steps, between first and last layers, were introduced to avoid the coarse results due to many pooling layers. But, the authors in [29] thought that this is not enough for the text-line segmentation task which should be more accurate. So, they proposed a new architecture where the layer convolution and 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 [12].

Grüning & al. [15] proposed a two steps method to segment text-lines. The first step is an ARU-net which is a U-net extended with an attention model (A) and a residual structure (R). Note that the U-net is a variant of FCN [30] which introduces shortcuts between layers of the same spatial dimension to have features from higher level and to reduce the vanishing gradient problem. The authors added residual blocks [17] 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 another to the beginning and end of the text-lines. In the second step, a set of super pixels was calculated from the baseline map, then the state of these super pixels was estimated 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, the super pixels clusters were found which represent the text-lines. Their system reaches 95% of good detection of text-lines on the whole cBad dataset, but the many processing and computations used, especially on the second stage, makes it a heavy procedure.

For word segmentation, Papavassiliou & al. [26] proposed an SVM-based gap metric for adjacent connected components at text-line level. Then, they used a threshold to classify gaps as "within" or "between" words. They tested their method on the datasets from the ICDAR 2007 Handwriting Segmentation Contest [20] and reached an F-measure of 93%.

Al Khateeb & al. [2] applied a component-based method to segment words in Arabic handwritten texts. They analyzed the connected components, using the baselines and reached 85% of correct segmentation rate.

In [1], Al-Dmour & al. proposed a method based on two spatial measures: connected component length and gaps between them. Lengths are clustered to separate between the groups of letters, sub-words and words. Gaps are clustered to figure out whether the gap occurs "between words" or "within a word". These measures are clustered using a SOM (Self-Organizing Map) algorithm [18]. The method has been tested on the AHDB dataset [3] and achieved 86.3% of correct segmentation rate.

III. Proposed System

In segmentation methods, text-lines are generally represented by their 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 performance evaluation. An example of x-height and baseline labeling is provided in the Figure 2.

Inspired by the research of [15], we investigated the use of an RU-net for text-line segmentation, based on an x-height labeling, followed by a simple post-processing which allows baselines extraction. Note that U-Net is considered one of the standard CNN architectures for image classification tasks, when we need not only to define the whole image by its class but also to segment areas of an image by class, that is to produce a mask that will separate an image into several classes.

For word segmentation, we used a CNN followed by a BLSTM (forward and backward) and a CTC decoder at the end. The following subsections provide the architectural details of both text-line and word segmentation.

A. Text-line Segmentation

For text-line segmentation, we extracted the x-heights of text-lines, by the use of a ground truth that separates the input images into three classes: 1) background, 2) paragraphs and 3) text-lines x-heights in each paragraph, as shown in Figure 8(b). It is about a semantic segmentation problem where the objective is to achieve fine-grained inference by making dense predictions inferring labels for every pixel, so that each pixel is labeled with the class of its enclosing region. With the popularity of deep learning in recent years, this has been done using deep architectures, most often Convolutional Neural Networks (CNNs), which surpass other approaches, in Machine Learning, by a large margin in terms of accuracy and efficiency.

As Fully Convolutional Networks (FCNs) [23] can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation, we proposed a FCN for the specific task of text-line segmentation. Note that FCN is an extension of the classical CNN. The main idea is to make the classical CNN take as input arbitrary-sized images. The restriction of CNNs to accept and produce labels only for specific sized inputs comes from the fully-connected layers which are fixed. Contrary to them, FCNs only have convolutional and pooling layers which give them the ability to make predictions on arbitrary-sized inputs. Thus, FCN structure can be described as a symmetric encoder/decoder where the encoder convolves and down samples the input via a number of convolution blocks and pooling. Then, the decoder re-samples the result by the same sub-sampling factor. Thus, the output will be a map with the same size as that of the input and where each pixel is assigned a probability of belonging to a class.

In this work, we used an RU-net as FCN model, extended with residual connections. Note that RU-net is a Recurrent Convolutional Neural Network model based on U-Net which [30] allows an easier combination of low level features and high level ones by introducing shortcuts between the same level blocks. The residual connections greatly reduce the vanishing gradient problem [17]. The proposed RU-Net architecture is illustrated in Figure 3 where the 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. 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 to extract the baselines from the x-heights.

As it can be seen, we resized the input images to 720 pixels for the largest side and kept the same ratio between height and width for all the images to reduce the memory footprint. The encoder part of the proposed network is composed of three convolution and max-pooling blocks and a 4^{th} convolution layer, whereas the decoder part includes three convolution and

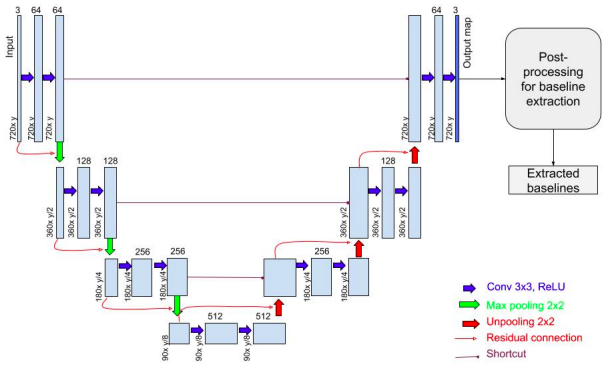


Fig. 3. RU-Net architecture consisted with convolutional encoding and decoding units that take image as input and produce the segmentation feature maps with respective pixel classes

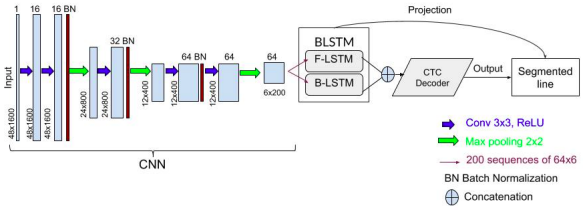


Fig. 4. Proposed word segmentation network.

un-pooling blocks. The network output consists of a map of three classes which will be dilated then eroded to smooth the border of x-heights and merge the over-segmented text-lines. Afterwards, polynomial regression is applied on the connected components, which are the x-heights here, to extract baselines.

1) Word Segmentation: For word segmentation, we firstly used a CNN to extract the most important features from the text-line images, extracted by the previous network. All images have a normalized size of 48×1600. Every convolutional block is followed by a batch normalization [21] that greatly reduces the vanishing gradient problem and makes the use of dropout unnecessary [22]. The output of the CNN is then sequentially passed to a BLSTM (Bidirectional Long Short Term Memory) having 100 neurons for each LSTM and followed by a CTC function [14], as shown in Figure 4 and explained later.

Let us consider the two classes: *word* (1) and *space* (2). The provided ground truth is the text-lines Unicode transcription where each word is labeled 1 and each space 0, as displayed in Figure 5. The CTC decoder output is the found sequence word-space. After the training step, the projection of the probabilities of class *space* (output of the BLSTM) is made on the image.

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

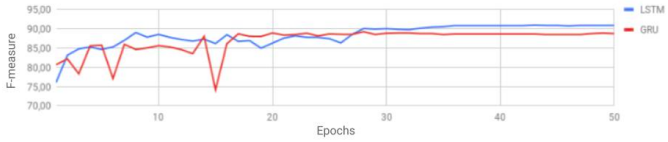


Fig. 6. Obtained results using LSTM and GRU.

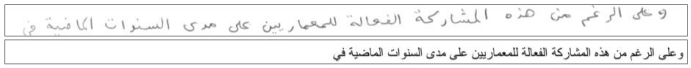


Fig. 7. Example of a text-line and its transcription in the KHATT database.

a) BLSTM: LSTMs and GRUs (Gated Recurrent Units) are the best known and efficient RNNs (Recurrent Neural Networks). While LSTM has three gates (input, forget, output) and a memory cell, GRU has only two gates (reset and update) and no memory cell, thus less parameters. No study has proven which one is better [10], and it all depends on one's case, so the two 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 the bidirectional LSTM (BLSTM), instead of unidirectional one, show very good results as they can understand context better.

b) CTC function:: As there is no obvious alignment between the network's output and the ground truth's input, the CTC function is used to do that automatically. Graves & al. [14] 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, that are the paths with the greatest probability values.

IV. Experiments & Results

To evaluate the text-line and word segmentation performance, we carried experiments on KHATT [25] database which is composed of handwritten Arabic images, written by 1000 persons from different age, gender and nationality. This database provides two 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 two paragraphs. The database also provides text-line segmentation and a Unicode transcription for each text-line (see Figure 7).

From the KHATT database, we labeled 325 paragraph images following the pattern shown in Figure 8(b): 175 for the training set and 150 for the testing set. For all our experiments, a learning rate of 10^{-4} and the Adam Optimizer [19] are used. The network has been trained for 100 epochs. Table I summarizes the parameter settings.

Image pre-processing :	325 paragraph images: 175 for training and 150 for testing. Images are normalized to a size of 720 px 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 I
Parameter settings of the text-line segmentation architecture. The 4 nearest graph modeling further connected these prices to form a structure.

Architecture	Dataset	F-measure
[29]	KHATT	81%
	cBad	75%
[15]	cBad	95%
RU-Net(ours)	KHATT	96.71%

TABLE II
Results from text-line segmentation.

To evaluate the baseline extraction, we used the metric in [16] and computed the F-measure. Table II displays results from some machine learning methods, used for text-line extraction and tested on KHATT dataset. The system, in [15], has not been tested, because the code of the second stage is not provided. The results in [29] are evaluated on the x-heights, while those in [15] and RU-net are evaluated on the baselines. Obtained results are illustrated in Figure 8.

The proposed RU-net is shown to improve the text-line segmentation rate relative to methods mentioned above. Moreover, it is simple, requires less parameters and processing steps and trained on just a small set. A larger training set should lead to better results, but that needs a manual labeling of the images which could be the brake of this system.

For word segmentation, we used 5000 text-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 III summarizes the parameter settings.

Figure 9 gives the curve of loss and Figure 10 gives the

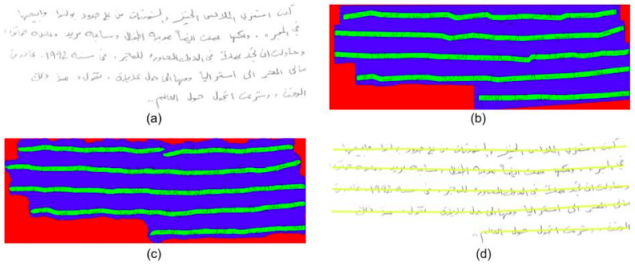


Fig. 8. Results of text-line segmentation: a) the original image, b) the ground truth composed of three classes (background: red; paragraph: blue; x-height: green), c) the output of the RU-net, d) the final result after post-processing.

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 III
PARAMETER SETTINGS OF THE WORD SEGMENTATION ARCHITECTURE.

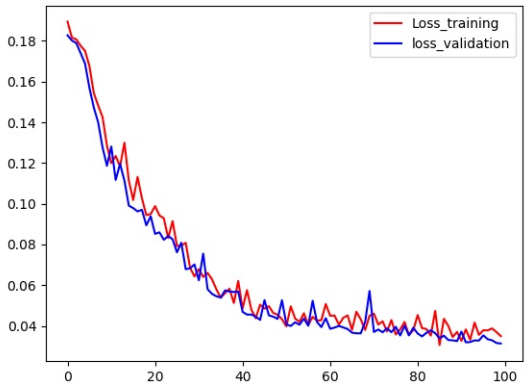


Fig. 9. Loss function.

curves of accuracy.

Figure 11 shows an example of the obtained results and Table IV displays the F-measure values of our system and other works. Since the prediction is done on a down-scaled image, the output needs to be up-scaled to fit the original image. The F-measure was calculated manually by comparing

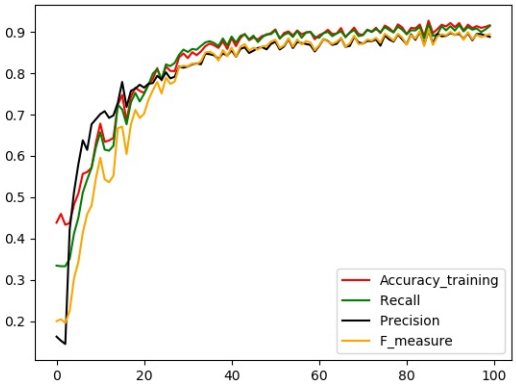


Fig. 10. Accuracy measurements.

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

TABLE IV
RESULTS FROM WORD SEGMENTATION.

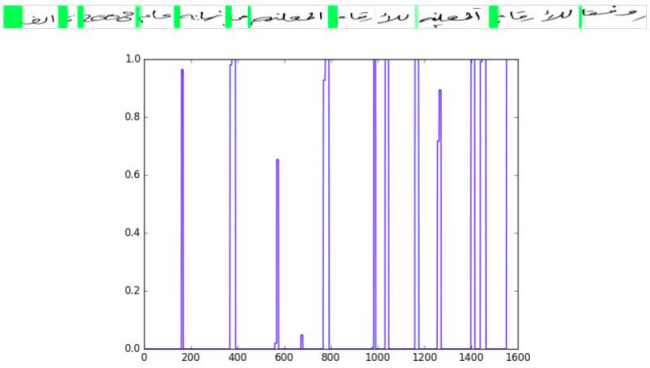


Fig. 11. Results of the proposed system. The lower image is the output of the BLSTM for the class *space (0)*, the 200 probabilities has been up-scaled $\times 8$ to recover the original shape. The upper image is the segmented line after the projection of the probabilities.

the words on the transcription and the resulting segmentation. The most common segmentation errors are misplaced segmentations, like shown in Figure 12 where the segmentation is between sub-words.

Compared to some works on Arabic word segmentation, the proposed system achieves less good results. But, note that our system is a data-driven language independent word segmentation system. That is, no language dependent features are applied for tuning or improving the accuracy. In addition, it has has been trained and tested on KHATT database which involves more complex writings than IFN/ENIT and AHDB databases.

V. CONCLUSION

In this work, we proposed a novel approach to carry out the segmentation of Arabic manuscripts into text-lines and words, using deep learning. The proposed text-line segmentation system uses an RU-net to extract x-heights from text images, then a post-processing step extracts baselines. The word segmentation system uses a CNN with a BLSTM, then a CTC to find the alignment between the text-line transcription and the text-line image. The results show that text-line and word segmentation problems can be solved with no lexicons or language-dependent resources. The obtained results are promising, but still need to be improved, especially for

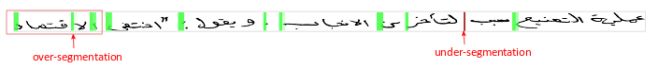


Fig. 12. Example of wrong segmentations. The red box denotes one word and an over-segmentation is made on the sub-words.

word segmentation. As future work, we plan to improve this approach by the use of some post-processing to correct the wrong segmentation.

REFERENCES

[1] A. Al-Dmour and R. Abu Zitar, "Word Extraction from Arabic Handwritten Documents Based on Statistical Measures", International Review On Computer and Software, (11):5, 2016.

[2] J. H AlKhateeb, J. Jiang, J. Ren and Stan S Ipson, "Component-based Segmentation of Words from Handwritten Arabic Text", International Journal of Computer and Information Engineering, (2):5, pp. 1428-1432, 2008.

[3] S. Al-Ma'adeed and D. Elliman and C. A. Higgins, "A data base for Arabic handwritten text recognition research", Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, pages 485-489, 2002.

[4] N. Aouadi and A. Kacem-Echi, "Word Extraction and Recognition in Arabic Handwritten Text", International Journal of Computing & Information Sciences, (12):1, pp. 17-23, 2016.

[5] S. Avidan and A. Shamir, "Seam Carving for Content-aware Image Resizing", journal ACM Trans. Graph., (26)3, July 2007.

[6] N. Arvanitopoulos and S. Süsstrunk, "Seam Carving for Text Line Extraction on Color and Grayscale Historical Manuscripts", 14th International Conference on Frontiers in Handwriting Recognition, pp. 726-731, 2014.

[7] A. Belaïd and N. Ouwayed, "Segmentation of Ancient Arabic Documents", "Guide to OCR for Arabic Scripts", Märgner, Volker and El Abed, Haikal editor, Springer London publisher, pages 103-122, 2012.

[8] Y. Boulid, A. Souhar and M. Y. Elkettani, "Arabic handwritten text line extraction using connected component analysis from a multi agent perspective", 15th International Conference on Intelligent Systems Design and Applications, pp. 80-87, 2015.

[9] K. Chen, M. Seuret, J. Hennebert and R. Ingold, "Convolutional Neural Networks for Page Segmentation of Historical Document Images", 14th IAPR International Conference on Document Analysis and Recognition, pp. 965-970, 2017.

[10] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", arXiv:1412.3555v1 [cs.NE], pp. 1-9, 2014.

[11] R. Cohen, I. Dinstein, J. El-Sana and K. Kedem, "Using Scale-Space Anisotropic Smoothing for Text Line Extraction in Historical Documents", Image Analysis and Recognition, Campilho, Aurélio and Kamel, Mohamed editor, Springer International Publishing, pp. 349-358, 2014.

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

[13] M. Elzobi, A. Al-Hamadi and Z. Al Aghbari, "Off-line Handwritten Arabic Words Segmentation Based on Structural Features and Connected Components Analysis", The 19th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 135-142, 2011.

[14] A. Graves, S. Fernandez, F. Gomez and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks", International Conference on Machine learning, pages 369-376, 2006.

[15] T. Grüning, G. Leifert, T. Strauss and R. Labahn, "A Two-Stage Method for Text Line Detection in Historical Documents", arXiv:1802.03345v1 [cs.CV], pp. 1-38, 2018.

[16] 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", 13th IAPR International Workshop on Document Analysis Systems, pages 351-356, 2018.

[17] K. He and X. Zhang and S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition, pages 770-778, 2016.

[18] T. Kohonen, "The self-organizing map", Proceedings of the IEEE, (78):9, pages 1464-1480, 1990.

[19] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, ICLR 2015, pp. 1-15.

[20] www.iit.demokritos.gr/ bgat/HandSegmCont, 2007.

[21] S. Ioffe and Ch. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Proceedings of the 32nd International Conference on Machine Learning, 2015.

[22] X. Li, S. Chen, X. Hu and J. Yang,"Understanding the Disharmony between Dropout and Batch Normalization by Variance Shift", arXiv preprint arXiv:1801.05134, 2018.

[23] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation", pp. 3431-3440, 2016.

[24] G. Louloudis and B. Gatos and I. Pratikakis and C. Halatsis, "Text Line and Word Segmentation of Handwritten Documents", journal Pattern Recognition,(42):12, pages 3169-3183, December, 2009.

[25] S. A. Mahmoud, I. Ahmad and M. Alshayeb, W. G. Al-Khatib, M. T. Parvez, G. A. Fink, V. Margner, and H. EL Abed, "KHATT: Arabic Offline Handwritten Text Database", Proceedings of the 13^th International Conference on Frontiers in Handwriting Recognition, Bari-Italy, pages 447-452, 2012.

[26] V. Papavassiliou, Th. Stafylakis, V. Katsouros and G. Carayannis, "Handwritten document image segmentation into text lines and words", Journal Pattern Recognition, 43(1), pages 369 - 377, 2010.

[27] J. Pastor-Pellicer, M. Z. Afzal, M. Liwicki and M. J. Castro-Bleda, "Complete System for Text Line Extraction Using Convolutional Neural Networks and Watershed Transform", 12th IAPR Workshop on Document Analysis Systems, pp. 30-35, 2016.

[28] G. G. Rajput and Suryakant B. Ummapure and Preethi N. Patil, "Text-Line Extraction from Handwritten Document images using Histogram and Connected Component Analysis", International Journal of Computer Applications, National conference on Digital Image and Signal Processing, pages 11-17, 2015.

[29] G. Renton and C. Chatelain and S. Adam and C. Kermorvant and T. Paquet, "Handwritten Text Line Segmentation Using Fully Convolutional Network", 14th IAPR International Conference on Document Analysis and Recognition, pp. 5-9, 2017.

[30] O. Ronneberger, Ph. Fischer and Th. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", MICCAI 2015: Medical Image Computing and Computer-Assisted, pp. 234-241, 2015.