Unsupervised deep learning for text line segmentation

Berat Kurar Barakat*, Ahmad Droby*, Reem Alaasam*, Boraq Madi*,
Irina Rabaev§, Raed Shammes* and Jihad El-Sana*
* Ben-Gurion University of the Negev
{berat,drobya,rym,borak,rshammes}@post.bgu.ac.il
§ Shamoon College of Engineering
irinar@ac.sce.ac.il

***Abstract*—We present an unsupervised deep learning method for text line segmentation that is inspired by the relative variance between text lines and spaces among text lines. Handwritten text line segmentation is important for the efficiency of further processing. A common method is to train a deep learning network for embedding the document image into an image of blob lines that are tracing the text lines. Previous methods learned such embedding in a supervised manner, requiring the annotation of many document images. This paper presents an unsupervised embedding of document image patches without a need for annotations. The number of foreground pixels over the text lines is relatively different from the number of foreground pixels over the spaces among text lines. Generating similar and different pairs relying on this principle definitely leads to outliers. However, as the results show, the outliers do not harm the convergence and the network learns to discriminate the text lines from the spaces between text lines. Remarkably, with a challenging Arabic handwritten text line segmentation dataset, VML-AHTE, we achieved superior performance over the supervised methods. Additionally, the proposed method was evaluated on the ICDAR 2017 and ICFHR 2010 handwritten text line segmentation datasets.**

I. Introduction

Text line segmentation is a classical document image analysis problem that has impact on the performance of subsequent analysis operations. The objective of text line segmentation is to recognize all the pixels that belong to a text line, as shown in Fig. 1(d). Text line segmentation contains both, text line detection and text line extraction. Text line detection roughly locates text line patterns, whereas text line extraction precisely assigns pixels to the text lines. Detection results can be represented by baselines or blob lines (Fig. 1(c)). Extraction can be represented by pixel labels (Fig. 1(d)) or bounding polygons. The final goal of a text line segmentation procedure is to provide text lines one by one into the next document analysis procedure.

Recently, numerous deep learning based methods have been proposed for text line segmentation of handwritten documents. Learning based methods [1]–[4] can inherently handle the problems arising from complex layout of text lines and heterogeneity of documents. However, they require a vast amount of labeling effort which consumes time not less than carefully designed ad-hoc heuristics [5]–[8]. Intuitively, labeling effort is favorable over designing ad-hoc heuristics because the former can be accomplished by human recognition skills, whereas the latter requires further mathematical skills.

This paper presents a simple but interestingly successful unsupervised convolutional network for text line segmentation. The input for the network is an unlabeled document image, and the output is segmentation of text lines. The main idea can be formulated that the visual discrimination of number of

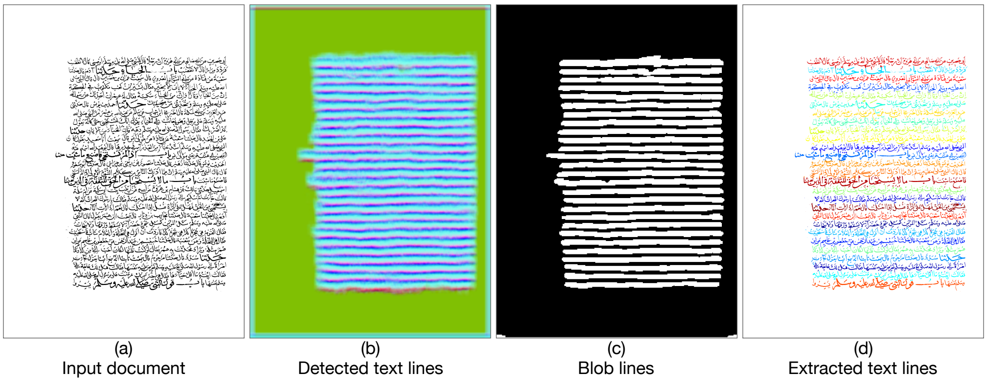

Fig. 1. Given a handwritten document image (a), UTLS learns to extract representation vectors of image patches where the distances between these vectors are proportional to the similarity of patches. Three principal components of patch representation vectors are visualized as a pseudo-RGB image (b). The pseudo-RGB images are thresholded onto blob lines that strike through text lines (c). Energy minimization with the assistance of detected blob lines extracts the pixel labels of text lines (d).

foreground pixels in document image patches requires machine to learn features that represent proximity and similarity of the elements in the document image. According to the Gestalt principle [9], such relevance among the elements of a document image forms the basis of unsupervised segmentation of text lines. In the first phase, we train a siamese network to learn that two document image patches with relatively same/distinct number of foreground pixels are similar/different. Certainly, this measurement assigns many pairs improperly. However, the outliers do not harm the convergence of the machine learning [10]. Next, we extract representation vectors of document image patches using the penultimate layer of a single branch of the siamese network. Then, we reduce dimensions of these vectors into their three principle components, which enables producing pseudo-RGB images where similar pixels in the embedded space correspond to similar colors [10]. The pseudo-RGB images are thresholded into blob lines that hover the text lines. In the last phase, text lines are labeled in pixel level using an energy minimization framework with the assistance of the detected blob lines [11]. Experiments on an Arabic handwritten textline extraction dataset, which possesses challenges by crowded and cramped text lines, show that Unsupervised Text Line Segmentation (UTLS) is more effective than supervised methods. In addition, we achieved comparable results on ICDAR 2017 [12] and ICFHR 2010 [13] handwritten text line segmentation datasets.

II. Related Work

Text line detection and segmentation in historical document images have been widely studied during the last decades, but still remains an open problem for challenging documents.

During the years, numerous methods for text line extraction have been proposed. Among the early approaches are projection profiles based methods, which were first applied to documents with horizontal text lines [14], [15], and subsequently adapted to document with skewed [16], [17] and multi-skewed text lines [18]. Another wide class of methods are grouping or clustering methods that aggregate elements (such as pixels or connected components) in a bottom up strategy [7], [19]–[21]. Smearing based methods [5]–[7], [22]–[24] target to enhance the text line structure. Seam-carving methods build energy map and compute seams that separate text lines (or seams that pierce through text lines) [25]–[28]. Recently, learning-based methods have shown promising results when applied for text line segmentation of handwritten documents. Renton *et al.* [1] employed a variant of Fully Convolutional Network (FCN) with dilated convolutions for text line extraction. The model is trained to output an X-height pixel labeling as text line representation. Oliveira *et al.* [3] presented a CNN-based pixel-wise predictor for addressing multiple tasks simultaneously: page extraction, layout analysis, baseline extraction, and illustration and photograph extraction. Their network is trained to predict the binary mask of polygonal lines that represent baselines. Kurar *et al.* [4] build a FCN to predict text line masks. Their method targeted challenging documents, which contain curved, multi-skewed and multi-directed text lines of different fonts types and sizes. Kiessling *et al.* [29] presented method based on a fully convolutional encoder-decoder network to detect baselines in document images. The baseline definition was modified slightly towards manuscripts written in Arabic scripts. Mechi *et al.* [30] and Neche *et al.* [31] used an U-net and RU-net deep-learning models, which are variants of FCN. The models are trained for X-height based pixel-wise classifications of text lines.

All of the learning based methods reviewed above are supervised methods. We are not aware of any unsupervised deep learning approach for text line segmentation. In this paper we present an unsupervised deep learning method for text line segmentation, and evaluate it on three publicly available datasets.

III. Method

We present a method for unsupervised text line segmentation (UTLS) and show its effectiveness on handwritten document images. The method uses a siamese convolutional network to predict whether two given document image patches are similar or different, driven by the number of foreground pixels in the patches. After the training phase, a single branch of the trained network is used to extract features of document image patches, which are in turn visualized as pseudo-RGB images (Fig. 1(b)) and thresholded into blob lines that strike through text lines (Fig. 1(c)). Finally, we use an energy minimization framework [24] to extract the pixel labels of text lines with the assistance of the detected blob lines (Fig. 1(d)). This section provides the details of data preparation, training, visualization of blob lines and energy minimization procedures.

A. Data preparation

Data preparation consists of generating patches of the size $h_p \times w_p$ pixels, cropped randomly from document images and labeling every pair of patches either similar or different. The patch height h_p is estimated as three times of the average character height in the document images. The patch width w_p is estimated experimentally per dataset. The labeling is done automatically using a similarity score between two patches.

Given randomly cropped two image patches, let a_i be the number of foreground pixels in patch i where $i \in \{1, 2\}$. We define the similarity score s as:

$$s = \frac{\min(a_1, a_2)}{\max(a_1, a_2)} \quad (1)$$

Assume that $a_2 > a_1$ then, a_1 and a_2 are most similar when

$$(a_2 - a_1) \to 0 \text{ and } \frac{a_1}{a_2} \to 1 \text{ and in turn } s \to 1. \quad (2)$$

a_1 and a_2 are most different when

$$(a_2 - a_1) \to \infty \text{ and } \frac{a_1}{a_2} \to 0 \text{ and in turn } s \to 0. \quad (3)$$

1) Patches similar by number of foreground pixels: This strategy continues cropping two random patches until the similarity score s satisfies the following condition:

$$s \geq 0.7 \tag{4}$$

Intuitively this strategy generates pairs where both centralize either a text line part or a part of space between text lines (Fig. 2).

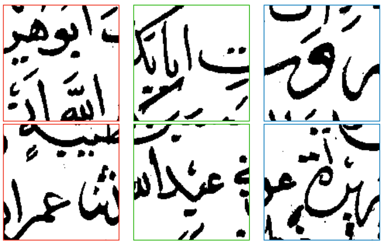

Fig. 2. Every column shows a pair of similar patches. In a loosely manner, both patches in each pair centralize either a text line part or a part of space between text lines.

2) Patches different by number of foreground pixels: This strategy continues cropping two random patches until the similarity score s satisfies the following condition:

$$s \leq 0.4 \tag{5}$$

Intuitively this strategy generates pairs where one centralizes a text line part and the other centralizes a part of space between text lines (Fig. 3).

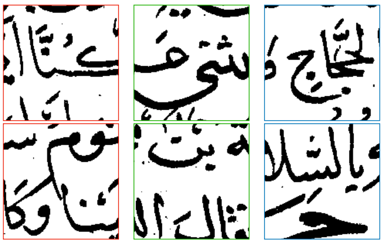

Fig. 3. Every column shows a pair of different patches. In a loosely manner, one of the patches in each pair centralizes a text line part and the other centralizes a part of space between text lines.

3) Patches different by background area: There also exist a significant difference between the background areas and the text areas in the document image. This strategy continues cropping two random patches until one of the patches is from background area and the other is from text area (Fig. 4). We assume a patch is from background area if most of its pixels are background pixels.

B. Training

The common deep learning practice for handwritten text line segmentation is to adapt an embedding from the text lines image into a blob lines image. The classifier is first trained on a labeled set of text lines, and then expected to predict blob lines. Unlike these methods, UTLS does not need labeled data for mapping the text line image into a blob line image. It is simply trained to distinct the text lines from the spaces between text lines.

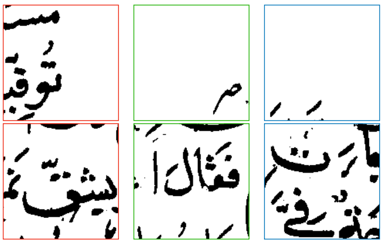

Fig. 4. Every column shows a pair of different patches. In a loosely manner, either of patches in each pair contains either background area or foreground area.

The overall architecture is a siamese network with two identical branches. Each branch inputs an image patch and outputs a feature representation of that image patch. Consequently, these feature representations are concatenated and fed to fully connected layers in order to classify whether the two image patches are similar or different. The branches of siamese network model is based on AlexNet [32] and through experiments we tune the hyperparameters to fit our task. The final architecture contains two branches of CNN, each of the branches has five convolutional layers as presented in Fig. 5. Dotted lines indicate identical weights, and the numbers in parentheses are 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.

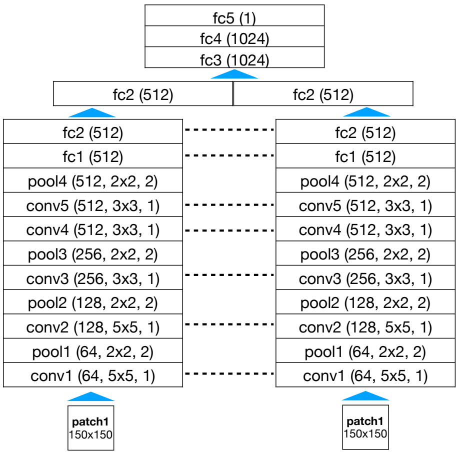

Fig. 5. Siamese architecture for pair similarity. Dotted lines stand for identical weights, conv stands for convolutional layer, fc stands for fully connected layer and pool is a max pooling layer.

We trained this model from scratch using $30,000$ pairs that are generated and labeled according to the strategies described in section III-A, and reached a validation loss value of 0.29 after 11 epochs (Fig. 6).

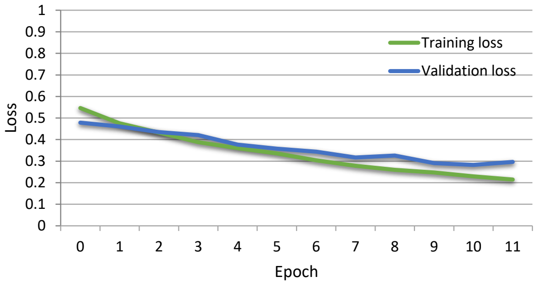

Fig. 6. Loss over the epochs of model training.

C. *Visualization of blob lines for text line detection*

Once the siamese network is trained, we use a single branch to extract the features of patches. This embeds every patch into a feature vector of 512 dimensions. To visualize the features of a complete document image, a sliding window of the size $h_p \times w_p$ is used, but only the inner window of the size $h_i \times w_i$ is considered to eliminate the edge affect. In our experiments we find $(h_i \times w_i) = (10, 10)$ gives the best results on all the datasets. We also pad the document image with white pixels at its right and bottom sides if its size is not an integer multiple of the sliding window size. An additional padding is added at 4 sides of the document image for considering only the central part of the sliding window. As a result, a document image with the size $h_d \times w_d$ is mapped to a representation matrix of the size $h_d \times w_d \times 512$. We project $512D$ vectors into their three principle components and use these components to construct pseudo-RGB image in which similar patches are assigned the similar colors (Fig. 1(b)). Binary blob lines image is an outcome of thresholded pseudo-RGB image (Fig. 1(c)).

D. *Energy minimization for text line extraction*

We adopt the energy minimization framework [33] that uses graph cuts to approximate the minima of an arbitrary function. We adapt the energy function to be used with connected components for extracting the text lines. Minimum of the adapted function correspond to a good extraction which urges to assign components to the label of the closest blob line while straining to assign closer components to the same label.

Let $\mathcal{L}$ be the set of binary blob lines, and $\mathcal{C}$ be the set of components in the binary document image. Energy minimization finds a labeling f that assigns each component $c \in \mathcal{C}$ to a label $l_c \in \mathcal{L}$, where energy function $\mathbf{E}(f)$ has the minimum.

$$\mathbf{E}(f) = \sum_{c \in \mathcal{C}} D(c, \ell_c) + \sum_{\{c,c'\} \in \mathcal{N}} d(c, c') \cdot \delta(\ell_c \neq \ell_{c'}) \quad (6)$$

The term D is the data cost, d is the smoothness cost, and δ is an indicator function. Data cost is the cost of assigning component c to label l_c. $D(c, \ell_c)$ is defined to be the Euclidean distance between the centroid of the component c and the nearest neighbour pixel in blob line l_c for the centroid of the component c. Smoothness cost is the cost of assigning neighbouring elements to different labels. Let $\mathcal{N}$ be the set of nearest component pairs. Then $\forall \{c, c'\} \in \mathcal{N}$

$$d(c, c') = \exp(-\beta \cdot d_c(c, c')) \quad (7)$$

where $d_c(c, c')$ is the Euclidean distance between the centroids of the components c and c', and β is defined as

$$\beta = (2 \langle d_c(c, c') \rangle)^{-1} \quad (8)$$

$\langle \cdot \rangle$ denotes expectation over all pairs of neighbouring components [34] in a document page image. $\delta(\ell_c \neq \ell_{c'})$ is equal to 1 if the condition inside the parentheses holds and 0 otherwise.

IV. Datasets

We evaluated the proposed method on three publicly available handwritten datasets: VML-AHTE, ICDAR 2017 [12] and ICFHR 2010 [13].

A. *VML-AHTE*

Visual Media Lab - Arabic Handwritten Textline Extraction (VML-AHTE) dataset is a collection of 30 pages selected from several manuscripts. It is a newly published dataset and available online for downloading[1]. VML-AHTE dataset is challenging in terms of rich diacritics, and touching and overlapping characters, as shown in Fig.7.

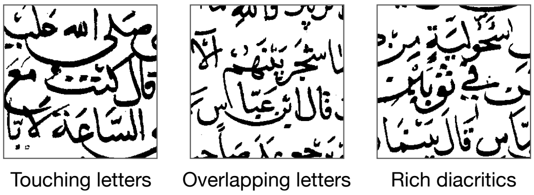

Fig. 7. Some samples of challenges in VML-AHTE dataset.

B. *ICDAR 2017*

ICDAR 2017 dataset [12] contains 150 pages from 3 medieval manuscripts: CB55, CSG18 and CSG863, see Fig. 8 for an example. Among them, CB55 is characterized by a vast number of touching text lines.

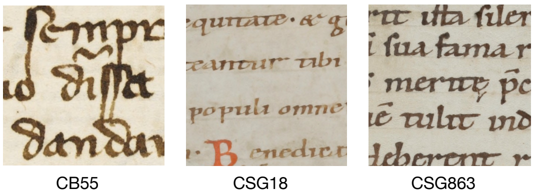

Fig. 8. Diva-HisDB dataset contains 3 manuscripts: CB55, CSG18 and CSG863. Notice the touching characters among multiple consecutive text lines in CB55.

C. *ICFHR 2010*

ICFHR 2010 dataset [13] is particularly challenging as it comprises handwriting from different languages and writers. The text lines are skewed and have varying sizes as well as interline spacing, as shown in the example page in Fig. 9.

[1] https://www.cs.bgu.ac.il/~berat/data/ahte_dataset

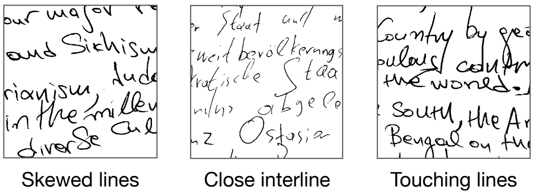

Fig. 9. ICFHR 2010 dataset contains unconstrained handwritten documents.

V. Experiments

Our experimental study covers three datasets that are different in terms of the text line segmentation challenges they contain. On one hand, VML-AHTE dataset exhibits crowded diacritics and cramped text lines, whereas ICDAR 2017 dataset contains consequently touching text lines. On the other hand, ICFHR 2010 dataset is heterogeneous by document resolutions, text line heights and skews. Therefore, the proposed algorithm does not use universal values for all the experimented datasets. In this section we present the effect of patch size and similarity score values on the method's performance. The performance is measured using the line segmentation evaluation metrics of ICDAR 2013 [13] and ICDAR 2017 [35].

A. ICDAR 2013 line segmentation evaluation metrics

ICDAR 2013 metrics calculate recognition accuracy (RA), detection rate (DR) and F-measure (FM) values. Given a set of image points I, let R_i be the set of points inside the i^{th} result region, G_j be the set of points inside the j^{th} ground truth region, and $T(p)$ is a function that counts the points inside the set p, then the $MatchScore(i,j)$ is calculated by Equation 9

$$MatchScore(i,j) = \frac{T(Gj \cap Ri)}{T(Gj \cup Ri)} \tag{9}$$

The evaluator considers a region pair (i,j) as a one-to-one match if the $MatchScore(i,j)$ is equal or above the threshold, which we set to 90 for all evaluations except for ICFHR 2010 dataset to 95 for results to be comparable. Let N_1 and N_2 be the number of ground truth and output elements, respectively, and let M be the number of one-to-one matches. The evaluator calculates the DR, RA and FM as follows:

$$DR = \frac{M}{N_1} \tag{10}$$

$$RA = \frac{M}{N_2} \tag{11}$$

$$FM = \frac{2 \times DR \times RA}{DR + RA} \tag{12}$$

B. ICDAR 2017 line segmentation evaluation metrics

ICDAR 2017 metrics are based on the Intersection over Union (IU). IU scores for each possible pair of Ground Truth (GT) polygons and Prediction (P) polygons are computed as follows:

$$IU = \frac{IP}{UP} \tag{13}$$

IP denotes the number of intersecting foreground pixels among the pair of polygons. UP denotes number of foreground pixels in the union of foreground pixels of the pair of polygons. The pairs with maximum IU score are selected as the matching pairs of GT polygons and P polygons. Then, pixel IU and line IU are calculated among these matching pairs. For each matching pair, line TP, line FP and line FN are given by:

- Line TP is the number of foreground pixels that are correctly predicted in the matching pair.
- Line FP is the number of foreground pixels that are falsely predicted in the matching pair.
- Line FN is the number of false negative foreground pixels in the matching pair.

Accordingly pixel IU is:

$$\text{Pixel } IU = \frac{TP}{TP + FP + FN} \tag{14}$$

where TP is the global sum of line TPs, FP is the global sum of line FPs, and FN is the global sum of line FNs.

Line IU is measured at line level. For each matching pair, line precision and line recall are:

$$\text{Line precision} = \frac{\text{line } TP}{\text{line } TP + \text{line } FP} \tag{15}$$

$$\text{Line recall} = \frac{\text{line } TP}{\text{line } TP + \text{line } FN} \tag{16}$$

Accordingly, line IU is:

$$\text{Line } IU = \frac{\text{CL}}{\text{CL+ML+EL}} \tag{17}$$

where CL is the number of correct lines, ML is the number of missed lines, and EL is the number of extra lines.

For each matching pair:

- A line is correct if both, the line precision and the line recall are above the threshold value.
- A line is missed if the line recall is below the threshold value.
- A line is extra if the line precision is below the threshold value.

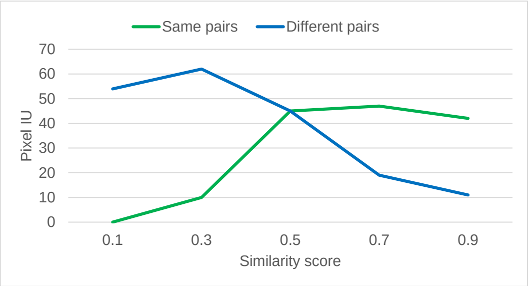

Fig. 10. Effect of different similarity score values for similar pairs and different pairs.

C. *Effect of similarity scores*

As we describe in section III-A two patches are most similar when similarity score $s \to 1$ and most different when $s \to 0$. We study this argument on a single page from VML-AHTE dataset. First, we fix $s \geq 0.5$ for similar pairs and report the effect of t in $s \leq t$ for different pairs. Then, we fix $s \leq 0.5$ for different pairs and report the effect of t in $s \geq t$ for similar pairs. The effect of different similarity score values can be observed in Fig. 10.

D. *Effect of patch size*

We have found that the patch size is a critical value for the effective performance of the algorithm. If the documents in a dataset contain text lines that have severely different heights, then the algorithm does not produce good results, as shown in Fig. 11. This inaccuracy is caused by the inappropriate height estimation, which is taken as three times the average text line height in the documents. On the other hand we observe that a constant patch size can detect text lines with slightly different heights (Fig. 12).

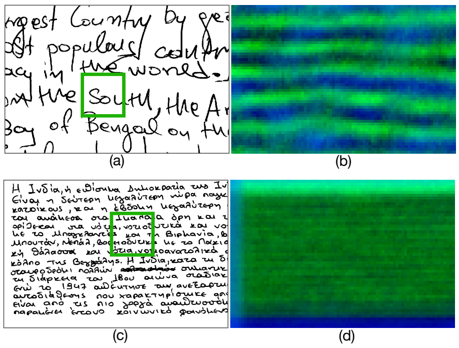

Fig. 11. Two sample document images from ICFHR 2010 dataset with very different text line heights (a) and (c). Same patch size can detect the text lines when its height is approximately 3 times the text line height (b) and can not detect the text lines when it spans several text lines together (d).

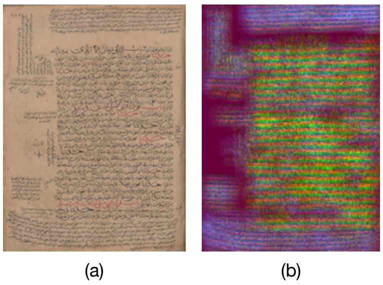

Fig. 12. A sample document image with heterogeneous text line heights (a). The pseudo-RGB output from the proposed method (b).

VI. Results

A. *Results on VML-AHTE dataset*

We compare our results with those of supervised methods, Mask-RCNN and FCN+EM and Human+EM. Mask-RCNN method is fully supervised using the pixel labels of the text lines. The advantage of this method is that it directly outputs pixel labels of text lines and does not need an additional procedure. FCN+EM method is also fully supervised but using blob lines that pass over the text lines. It uses EM framework to extract the pixel labels of text lines. Human+EM method is supervised by blob lines that are drawn by a human and uses EM framework to extract the pixel labels of text lines.

The comparison in terms of ICDAR 2013 metrics are reported in Table I.

TABLE I
DR, RA and FM values on VML-AHTE dataset.

Method	DR	RA	FM
Unsupervised			
UTLS	93.62	**93.95**	93.78
Supervised			
Mask-RCNN	84.43	58.89	68.77
FCN+EM	**95.55**	92.80	**94.30**
Human+EM	95.15	95.15	95.15

The comparison in terms of ICDAR 2017 metrics are reported in Table II.

TABLE II
Line IU and pixel IU values on VML-AHTE dataset.

Method	Line IU	Pixel IU
Unsupervised		
UTLS	**98.55**	88.95
Supervised		
Mask-RCNN	93.08	86.97
FCN+EM	94.52	**90.01**
Human+EM	99.29	91.49

On VML-AHTE dataset, UTLS successfully learns and discriminates between the text lines and the spaces among text lines. Moreover it outperforms all the supervised methods in terms of RA and line IU, and is competitive in terms of the other metrics. The error cases arise from few number of touching blob lines. Such errors can easily be eliminated but this is out of the focus of this paper.

B. *Results on ICDAR 2017 dataset*

The second evaluation is carried out on the Task 3 of ICDAR 2017 Competition on Layout Analysis for Challenging Medieval Manuscripts. Within the Task 3 only the main body lines are in the scope of interest. We run our algorithm on pre-segmented text block areas by the given ground truth. Hence, we can compare our results with unsupervised System 8 and System 9 which are based on layout analysis prior to text line segmentation. The comparison in terms of ICDAR 2017 metrics are reported in Table III.

Main challenge in this dataset for UTLS is the wide spaces between the words in a text line. The wider the space between words the much likely the algorithm detects it as a space instead of a text line, which in turn leads to over segmentation.

TABLE III
RESULTS FOR THE ICDAR 2017 COMPETITION ON LAYOUT ANALYSIS FOR CHALLENGING MEDIEVAL MANUSCRIPTS. LINE IU AND PIXEL IU RESULTS FOR TASK 3

	CB55		CSG18		CSG863	
	LIU	PIU	LIU	PIU	LIU	PIU
Unsupervised						
UTLS	80.35	77.30	94.30	95.50	90.58	89.40
System-8 (CIT-lab)	**99.33**	93.75	94.90	94.47	96.75	90.81
System-9+4.1 (DIVA+MG1)	98.04	**96.67**	**96.91**	**96.93**	**98.62**	**97.54**

LIU denotes Line IU and PIU denotes Pixel IU

C. Results on ICFHR 2010 dataset

This dataset contains very heterogeneous text lines with excessively different heights, interline spaces, and skews. The comparison on ICFHR 2010 dataset using ICDAR 2013 metrics are reported in Table IV.

Main challenge in this dataset for UTLS is the severely different text line heights. The algorithm can not detect the text lines with heights that are very greater than or very less than the patch height (Figure 11).

TABLE IV
DR, RA AND FM VALUES ON ICFHR 2010 DATASET.

Method	DR	RA	FM
Unsupervised			
UTLS	73.22	72.38	72.36
Winner	**97.54**	**97.72**	**97.63**
Supervised			
[36]	97.18	96.94	97.06

VII. CONCLUSION

We have presented an unsupervised text line segmentation method UTLS, trained to discriminate the text lines from the spaces between text lines. UTLS learn feature representations that are comparable or superior to other models trained with full supervision. The method is convenient in terms of average prediction time per page using a single Intel Xeon GPU (Table V).

The algorithm is very effective in detecting cramped and crowded text lines with nearly constant heights, interline spaces and interword spaces. However heterogeneity of aforementioned features decreases the performance of UTLS significantly.

TABLE V
AVERAGE PREDICTION RUN TIMES PER PAGE FOR EACH DATASET IN TERMS OF MINUTES.

	VML-AHTE	ICDAR 2017	ICFHR 2010
Average run time per page	2.62	2.20	1.49

ACKNOWLEDGMENT

The authors would like to thank Gunes Cevik for data preparation. This work was supported by Frankel Center for Computer Science at Ben-Gurion University.

REFERENCES

[1] G. Renton, Y. Soullard, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, "Fully convolutional network with dilated convolutions for handwritten text line segmentation," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 3, pp. 177–186, 2018.

[2] T. Grüning, G. Leifert, T. Strauß, J. Michael, and R. Labahn, "A two-stage method for text line detection in historical documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, no. 3, pp. 285–302, 2019.

[3] S. A. Oliveira, B. Seguin, and F. Kaplan, "dhSegment: A generic deep-learning approach for document segmentation," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 7–12.

[4] B. Kurar Barakat, A. Droby, M. Kassis, and J. El-Sana, "Text line segmentation for challenging handwritten document images using fully convolutional network," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 374–379.

[5] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger, "Script-independent text line segmentation in freestyle handwritten documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1313–1329, 2008.

[6] Z. Shi, S. Setlur, and V. Govindaraju, "A steerable directional local profile technique for extraction of handwritten arabic text lines," in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 176–180.

[7] S. S. Bukhari, F. Shafait, and T. M. Breuel, "Script-independent handwritten textlines segmentation using active contours," in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 446–450.

[8] R. Cohen, A. Asi, K. Kedem, J. El-Sana, and I. Dinstein, "Robust text and drawing segmentation algorithm for historical documents," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*. ACM, 2013, pp. 110–117.

[9] K. Koffka, *Principles of Gestalt psychology*. Routledge, 2013, vol. 44.

[10] 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.

[11] B. Kurar Barakat, A. Droby, B. Madi, R. Alaasam, I. Rabaev, and J. El-Sana, "Text line extraction using text line detection," in *2020 14th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2020, pp. –.

[12] F. Simistira, M. Bouillon, M. Seuret, M. Würsch, M. Alberti, R. Ingold, and M. Liwicki, "ICDAR2017 competition on layout analysis for challenging medieval manuscripts," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1361–1370.

[13] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICFHR 2010 handwriting segmentation contest," in *2010 12th International Conference on Frontiers in Handwriting Recognition*. IEEE, 2010, pp. 737–742.

[14] J. Ha, R. M. Haralick, and I. T. Phillips, "Document page decomposition by the bounding-box project," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 2. IEEE, 1995, pp. 1119–1122.

[15] R. Manmatha and N. Srimal, "Scale space technique for word segmentation in handwritten documents," in *International conference on scale-space theories in computer vision*. Springer, 1999, pp. 22–33.

[16] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A statistical approach to handwritten line segmentation," *Document Recognition and Retrieval XIV, Proceedings of SPIE, San Jose, CA*, pp. 6500T–1, 2007.

[17] I. Bar-Yosef, N. Hagbi, K. Kedem, and I. Dinstein, "Line segmentation for degraded handwritten historical documents," in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 1161–1165.

[18] N. Ouwayed and A. Belaïd, "A general approach for multi-oriented text line extraction of handwritten documents," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 15, no. 4, pp. 297–314, 2012.

[19] I. Rabaev, O. Biller, J. El-Sana, K. Kedem, and I. Dinstein, "Text line detection in corrupted and damaged historical manuscripts," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 812–816.

[20] R. Cohen, I. Dinstein, J. El-Sana, and K. Kedem, "Using scale-space anisotropic smoothing for text line extraction in historical documents," in *International Conference Image Analysis and Recognition*. Springer, 2014, pp. 349–358.

[21] T. Gruuening, G. Leifert, T. Strauss, and R. Labahn, "A robust and binarization-free approach for text line detection in historical documents," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 236–241.

[22] 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.

[23] A. Alaei, U. Pal, and P. Nagabhushan, "A new scheme for unconstrained handwritten text-line segmentation," *Pattern Recognition*, vol. 44, no. 4, pp. 917–928, 2011.

[24] B. K. Barakat, R. Cohen, I. Rabaev, and J. El-Sana, "VML-MOC: Segmenting a multiply oriented and curved handwritten text line dataset," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 6. IEEE, 2019, pp. 13–18.

[25] R. Saabni and J. El-Sana, "Language-independent text lines extraction using seam carving," in *2011 International Conference on Document Analysis and Recognition*. IEEE, 2011, pp. 563–568.

[26] A. Asi, R. Saabni, and J. El-Sana, "Text line segmentation for gray scale historical document images," in *Proceedings of the 2011 workshop on historical document imaging and processing*, 2011, pp. 120–126.

[27] M. Alberti, L. Vögtlin, V. Pondenkandath, M. Seuret, R. Ingold, and M. Liwicki, "Labeling, cutting, grouping: an efficient text line segmentation method for medieval manuscripts," *arXiv preprint arXiv:1906.11894*, 2019.

[28] A. Scius-Bertrand, L. Voegtlin, M. Alberti, A. Fischer, and M. Bui, "Layout analysis and text column segmentation for historical vietnamese steles," in *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, 2019, pp. 84–89.

[29] B. Kiessling, D. S. B. Ezra, and M. T. Miller, "BADAM: A public dataset for baseline detection in Arabic-script manuscripts," in *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, 2019, pp. 13–18.

[30] O. Mechi, M. Mehri, R. Ingold, and N. E. B. Amara, "Text line segmentation in historical document images using an adaptive U-Net architecture," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 369–374.

[31] C. Neche, A. Belaïd, and A. Kacem-Echi, "Arabic handwritten documents segmentation into text-lines and words using deep learning," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 6. IEEE, 2019, pp. 19–24.

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

[33] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[34] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, vol. 1. IEEE, 2001, pp. 105–112.

[35] M. Alberti, M. Bouillon, R. Ingold, and M. Liwicki, "Open evaluation tool for layout analysis of document images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 4. IEEE, 2017, pp. 43–47.

[36] M. Diem, F. Kleber, and R. Sablatnig, "Text line detection for heterogeneous documents," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 743–747.