

# Text line extraction using fully convolutional network and energy minimization

Berat Kurar Barakat\*, Ahmad Droby\*, Reem Alaasam\*, Boraq Madi\*,

Irina Rabaev<sup>§</sup> and Jihad El-Sana\*

\* Ben-Gurion University of the Negev

{drobya,berat,rym,borak}@post.bgu.ac.il

<sup>§</sup> Shamoon College of Engineering

irinar@ac.sce.ac.il

**Abstract**—Text lines are important parts of handwritten documents and easier to be analyzed by further applications. Despite recent progress in text line detection, text line extraction from a handwritten document remains an unsolved task. This paper proposes to use a fully convolutional network for text line detection and energy minimization for text line extraction. Detected text lines are represented by blob lines that strike through the text lines. These blob lines assist an energy function for text line extraction. The detection stage can locate arbitrarily oriented text lines. Furthermore, the extraction stage is capable of finding out the pixels of text lines with various heights and interline proximity independent of their orientations. Besides, it can finely split the touching and overlapping text lines without an orientation assumption. We evaluate the proposed method on VML-AHTE, VML-MOC and Diva-HisDB datasets. The first contains overlapping, touching and close text lines with rich diacritics. The second is very challenging by its multiply oriented and skewed text lines. The third exhibits distinct text line heights and touching text lines. The results demonstrate the effectiveness of the method despite various types of challenges, yet using the same parameters in all the experiments.

**Index Terms**—Text line segmentation, text line extraction, text line detection, handwritten document

## I. INTRODUCTION

Segmentation in computer vision is the task of dividing an image into parts that are easier to analyse. Text lines of a handwritten document image are widely used for word segmentation, recognition and spotting, manuscripts alignment and writer recognition. Text lines need to be provided to these applications either by their locations or by complete set of their pixels. The task of identifying the location of each text line is called detection whereas the task of finding out the pixels of each text line is called extraction. Much research in the recent years has focused on text line detection. However, detection defines the text lines loosely by baselines or main body blobs. On the other hand, extraction is a harder task which defines text lines precisely by pixel labels or bounding polygons.

The challenges in text line extraction arise due to variations in text line heights and orientations, presence of overlapping and touching text lines, and diacritical marks within close interline proximity. It has been generally demonstrated that deep learning based methods are effective at detecting text lines with various orientations [1]–[4]. Only few of the recent researches [1], [5] have addressed the problem of extraction

given the detection, yet with the assumption of horizontal text lines.

This paper proposes a text line extraction method (FCN+EM) which uses Fully Convolutional Network (FCN) to detect text lines in the form of blob lines (Fig. 1(b)), followed by an Energy Minimization (EM) function assisted by these blob lines to extract the text lines (Fig. 1(c)). FCN is capable of handling curved and arbitrarily oriented text lines. However, extraction is problematic due to the Sayre's paradox [6] which states that exact boundaries of handwritten text can be defined only after its recognition and handwritten text can be recognized only after extraction of its boundaries. Nevertheless, humans are good at understanding boundaries of text lines written in a language they do not know. Therefore, we consider EM framework to formulate the text line extraction in compliance with the human visual perception, with the aid of the Gestalt proximity principle for grouping [7]. The proposed EM formulation for text line extraction is free of an orientation assumption and can be used with touching and overlapping text lines with disjoint strokes and close interline proximity (Fig. 1(a)).

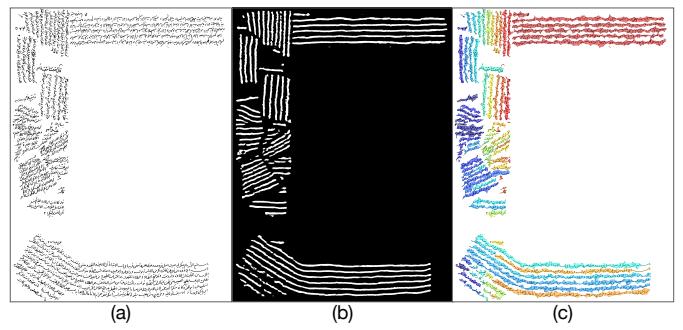


Fig. 1. Given a handwritten document image (a), FCN learns to detect blob lines that strike through text lines (b). EM with the assistance of detected blob lines extracts the pixel labels of text lines which are in turn enclosed by bounding polygons (d).

The proposed extraction method (FCN+EM) is evaluated on Visual Media Lab Arabic Handwritten Text line Extraction (VML-AHTE) dataset, Multiply Oriented and Curved (VML-MOC) dataset, and DIVA Historical Manuscript Database (DIVA-HisDB). VML-AHTE dataset is characterized by

touching and overlapping text lines with close proximity and rich diacritics. VML-MOC dataset contains arbitrarily oriented and curved text lines. DIVA-HisDB dataset exhibits varying text line heights and touching text lines.

The rest of the paper is organized as follows. Related work is discussed in Section II and the datasets are described in Section III. Later, the method is presented in Section IV. The experimental evaluation and the results are then provided in Section V. Finally, Section VI draws conclusions and outlines future work.

## II. RELATED WORK

A text line is a set of image elements, such as pixels or connected components. Text line components in a document image can be represented using basic geometric primitives such as points, lines, polylines, polygons or blobs. Text line representation is given as an input to other document image processing algorithms, and therefore important to be complete and correct.

There are two main approaches to represent text lines: text line detection and text line extraction. Text line detection detects the lines, polylines or blobs that represent the locations of spatially aligned set of text line elements. Detected line or polyline is called a baseline [4], [8] if it joins the lower part of the character main bodies, and a separator path [5], [9] if it follows the space between two consecutive text lines. Detected blobs [10] that cover the character main bodies in a text line are called text line blobs.

Text line extraction determines the constituting pixels or the polygons around the spatially aligned text line elements. Pixel labeling assigns the same label to all the pixels of a text line [1], [9], [11]. Bounding polygon is used to enclose all the elements of a text line together with its neighbourhood background pixels [12], [13]. Most of the extraction methods assume horizontally parallel text lines with constant heights, whereas some methods [14], [15] are more generic. ICDAR 2009 [16] and ICDAR 2013 [17] datasets are commonly used for evaluating text line extraction methods and ICDAR 2017 [18] dataset is used for evaluating text line detection methods. Additional ICDAR 2017 dataset [19] is used for both types of evaluations: detection and extraction. Therefore, we select to use ICDAR 2017 [19] as it provides ground truth for detection and extraction. However, this dataset is not enough representative of all the segmentation problems to evaluate a generic method. Hence, we also evaluated the proposed method on publicly available VML-MOC [14] dataset that contains multiply oriented and curved text lines with heterogeneous heights, and on VML-AHTE dataset that contains crowded diacritics.

## III. DATASETS

We evaluated the proposed method on three publicly available handwritten datasets. We suppose that these datasets demonstrate the generality of our method. As VML-AHTE dataset contains lines with crowded diacritics, VML-MOC

dataset contains multiply oriented and curved lines, and Diva-HisDB dataset contains consecutively touching multiple lines. In this section we present these datasets.

### A. VML-AHTE

VML-AHTE dataset is a collection of 30 binary document images selected from several manuscripts (Fig. 2). It is a newly published dataset and available online for downloading<sup>1</sup>. The dataset is split into 20 train pages and 10 test pages. Its ground truth is provided in three formats: bounding polygons in PAGE xml [20] format, color pixel labels and DIVA pixel labels [19].



Fig. 2. Some samples of challenges in VML-AHTE dataset.

### B. Diva-HisDB

DIVA-HisDB dataset [19] contains 150 pages from 3 medieval manuscripts: CB55, CSG18 and CSG863 (Fig. 3). Each book has 20 train pages and 10 test pages. Among them, CB55 is characterized by a vast number of touching characters. Ground truth is provided in three formats: baselines and bounding polygons in PAGE xml [20] format and DIVA pixel labels [19].

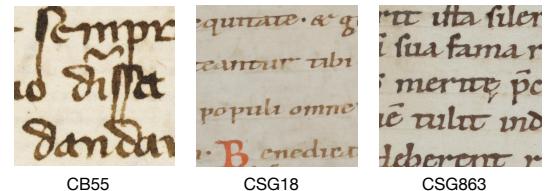


Fig. 3. Diva-HisDB dataset contains 3 manuscripts: CB55, CSG18 and CSG863. Notice the touching characters among multiple consecutive text lines in CB55.

### C. VML-MOC

VML-MOC dataset [14] is a multiply oriented and curved handwritten text lines dataset that is publicly available<sup>2</sup>. These text lines are side notes added by various scholars over the years on the page margins, in arbitrary orientations and curvy forms due to space constraints (Fig. 4). The dataset contains 30 binary document images and divided into 20 train pages and 10 test pages. The ground truth is provided in three formats: bounding polygons in PAGE xml [20] format, color pixel labels and DIVA pixel labels [19].

<sup>1</sup>[https://www.cs.bgu.ac.il/~berat/data/ahte\\_dataset](https://www.cs.bgu.ac.il/~berat/data/ahte_dataset)

<sup>2</sup>[https://www.cs.bgu.ac.il/~berat/data/moc\\_dataset](https://www.cs.bgu.ac.il/~berat/data/moc_dataset)

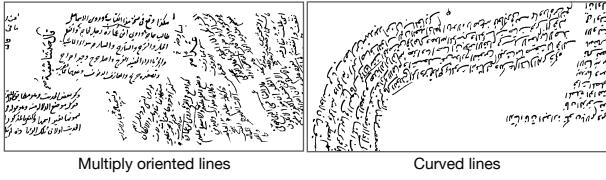


Fig. 4. VML-MOC dataset purely contains binarized side notes with arbitrary orientations and curvy forms.

#### IV. METHOD

We present a method (FCN+EM) for text line detection together with extraction and show its effectiveness on handwritten document images. In the first phase, the method uses an FCN to densely predict the pixels of the blob lines that strike through the text lines (Fig. 1(b)). In the second phase, we use an EM framework to extract the pixel labels of text lines with the assistance of detected blob lines (Fig. 1(c)). In the rest of this section we give a detailed description of FCN, EM and how they are used for text line detection and text line extraction.

##### A. Text line detection using FCN

Fully Convolutional Network (FCN) is an end-to-end semantic segmentation algorithm that extracts the features and learns the classifier function simultaneously. FCN inputs the original images and their pixel level annotations for learning the hypothesis function that can predict whether a pixel belongs to a text line label or not. A crucial decision have to be made about the representation of text line detection. Text line detection labels can be represented as baselines or blob lines.

We use blob line labeling that connects the characters in the same line while disregarding diacritics and touching components among the text lines. Blob line labeling for VML-AHTE and DIVA-HisDB datasets is automatically generated using the skeletons of bounding polygons provided by their ground truth (Fig. 5(d)). Blob line labeling for VML-MOC dataset is manually drawn using a sharp rectangular brush with a diameter of 12 pixels (Fig. 5(b)).

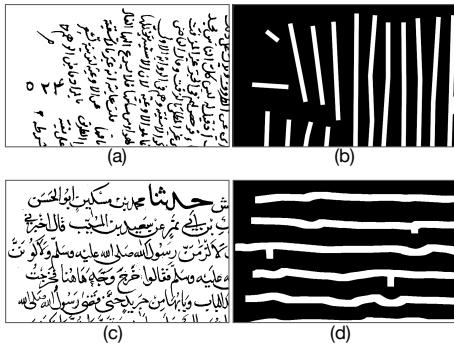


Fig. 5. Sample patches from document images of VML-MOC (a) and VML-AHTE (c). Blob line labeling for VML-AHTE and DIVA-HisDB is generated automatically (d). Blob line labeling for VML-AHTE is manually drawn using a paint brush with a diameter of 12 pixels (b).

**1) FCN architecture:** The FCN architecture (Fig. 6) we used is based on the FCN8 proposed for semantic segmentation [21]. Particularly FCN8 architecture was selected because it has been successful in page layout analysis of handwritten documents [22]. It consists of an encoder and a decoder. The encoder downsamples the input image and the filters can see coarser information with larger receptive field. Consequently the decoder adds final layer of encoder to the lower layers with finer information, then upsamples the combined layer back to the input size. Default input size is  $224 \times 224$ , which does not cover more than 2 to 3 text lines. To include more context, we changed the input size to  $350 \times 350$  pixels. We also changed the number of output channels as 2, which is the number of classes: blob line or not.

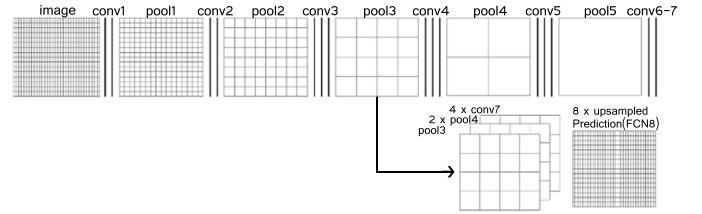


Fig. 6. The FCN architecture used for text line detection. Vertical lines show the convolutional layers. Grids show the relative coarseness of the pooling and prediction layers. FCN8 4 times upsamples the final layer, 2 times upsamples the pool4 layer and combine them with pool3 layer. Finally, upsamples the combination to the input size.

**2) FCN training:** For training, we randomly crop 50,000 patches of size  $350 \times 350$  from inverted binary images of the documents and their corresponding labels from the blob line label images (Fig. 5). We adopted this patch size due to memory limitation. Using full pages for training and prediction is not feasible on non-specialized systems without resizing the pages to a more manageable size. Resizing the pages will result in details loss, which usually reduces the accuracy of segmentation results.

The FCN was trained by a batch size of 12, using Stochastic Gradient Descent (SGD) with momentum equals to 0.9 and learning rate equals to 0.001. VGG was initialized with its publicly available pre-trained weights.

**3) FCN testing:** During the testing, a sliding window of size  $350 \times 350$  was used for prediction, but only the inner window of size  $250 \times 250$  was considered to eliminate the edge effect. Page was padded with black pixels at its right and bottom sides if its size is not an integer multiple of the sliding window size, in addition to padding it at 4 sides for considering only the central part of the sliding window.

##### B. Text line extraction using EM

We adapt the energy minimization (EM) framework [23] that uses graph cuts to approximate the minima of arbitrary functions. These functions can be formulated in terms of image elements such pixels or connected components. In this section we formulate a general function for text line extraction using text line detection. Then, we adapt this general function to be used with connected components for text line extraction.

1) *EM formulation*: Let  $\mathcal{L}$  be the set of binary blob lines, and  $\mathcal{E}$  be the set of elements in the binary document image. Energy minimization finds a labeling  $f$  that assigns each element  $e \in \mathcal{E}$  to a label  $l_e \in \mathcal{L}$ , where energy function  $\mathbf{E}(f)$  has the minimum.

$$\mathbf{E}(f) = \sum_{e \in \mathcal{E}} D(e, l_e) + \sum_{\{e, e'\} \in \mathcal{N}} d(e, e') \cdot \delta(l_e \neq l_{e'}) \quad (1)$$

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

$$d(e, e') = \exp(-\beta \cdot d_e(e, e')) \quad (2)$$

where  $d_e(e, e')$  is the Euclidean distance between the centroids of the elements  $e$  and  $e'$ , and  $\beta$  is defined as

$$\beta = (2 \langle d_e(e, e') \rangle)^{-1} \quad (3)$$

$\langle \cdot \rangle$  denotes expectation over all pairs of neighbouring elements [24] in a document page image.  $\delta(l_e \neq l_{e'})$  is equal to 1 if the condition inside the parentheses holds and 0 otherwise.

2) *EM adaptation to connected components*: The presented method extracts text lines using results of the text line detection procedure by FCN. Extraction level representation labels each pixel of the text lines in a document image. The major difficulty in pixel labeling lies in the computational cost. A typical document image in the experimented datasets includes around 14,000,000 pixels. Due to this reason, we adapt the energy function (Eq. 1) to be used with connected components for extraction of text lines.

Data cost of the adapted function measures how appropriate a label is for the component  $e$ , given the blob lines  $\mathcal{L}$ . Actually, the data cost alone would be equal to clustering the components with their nearest neighbour blob line. However, simply nearest neighbour clustering would be deficient to correctly label the free components that are disconnected from the blob lines (Fig. 7).

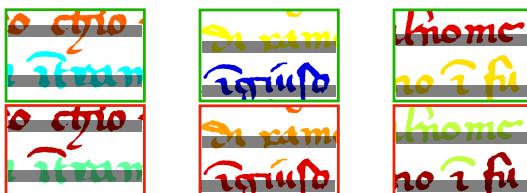


Fig. 7. Segmented samples that show the necessity of smoothness cost for text line extraction. Samples in the first row are true and achieved with smoothness cost. Samples in the second row are false and caused by the lack of a smoothness cost. Notice that smoothness cost pulls the diacritics to the components they belong to, in spite of their proximity to the wrong blob line.

A free component tends to exist closer to the components of a line it belongs to, but can be a nearest neighbour of a blob line that it does not belong to. This is because the proximity grouping strength decays exponentially with Euclidean distance [25]. This phenomenon is formulated using the smoothness cost (Eq. 2). Semantically this means that closer components have a higher probability to have the same label than distant components. Hence, the competition between data cost and smoothness cost dictates free components to be labeled spatially coherent with their neighbouring components.

## V. EXPERIMENTS

We experiment with three datasets that are different in terms of the text line segmentation challenges they contain. VML-AHTE dataset exhibits crowded diacritics and cramped text lines, whereas DIVA-HisDB dataset contains consequently touching text lines. Completely different than them VML-MOC exhibits challenges caused by arbitrarily skewed and curved text lines. The performance is measured using the line segmentation evaluation metrics of ICDAR 2013 [26] and ICDAR 2017 [27].

### 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 4

$$MatchScore(i, j) = \frac{T(G_j \cap R_i)}{T(G_j \cup R_i)} \quad (4)$$

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. 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:

$$DR = \frac{M}{N_1} \quad (5)$$

$$RA = \frac{M}{N_2} \quad (6)$$

$$FM = \frac{2 \times DR \times RA}{DR + RA} \quad (7)$$

### 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:

$$IU = \frac{IP}{UP} \quad (8)$$

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:

$$\text{Pixel } IU = \frac{TP}{TP + FP + FN} \quad (9)$$

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:

$$\text{Line precision} = \frac{\text{line } TP}{\text{line } TP + \text{line } FP} \quad (10)$$

$$\text{Line recall} = \frac{\text{line } TP}{\text{line } TP + \text{line } FN} \quad (11)$$

Accordingly, line IU is:

$$\text{Line } IU = \frac{\text{CL}}{\text{CL} + \text{ML} + \text{EL}} \quad (12)$$

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.

### C. Results on VML-AHTE dataset

Since VML-AHTE and VML-MOC datasets are recently published datasets we run two other supervised methods. First method is a holistic method that can extract text lines in one phase and is based on instance segmentation using MRCNN [28]. Second method is based on running the EM framework using the blob line labels from the ground truth and we refer to it Human+EM. On VML-AHTE dataset, FCN+EM outperforms all the other methods in terms of all the metrics except Line IU. It can successfully split the touching text lines and assign the disjoint strokes to the correct text lines.

TABLE I  
RESULTS ON VML-AHTE DATASET

	Line IU	Pixel IU	DR	RA	FM
FCN+EM	94.52	<b>90.01</b>	<b>95.55</b>	<b>92.8</b>	<b>94.3</b>
MRCNN	93.08	86.97	84.43	58.89	68.77
Human+EM	<b>97.83</b>	89.61	88.14	87.78	87.96

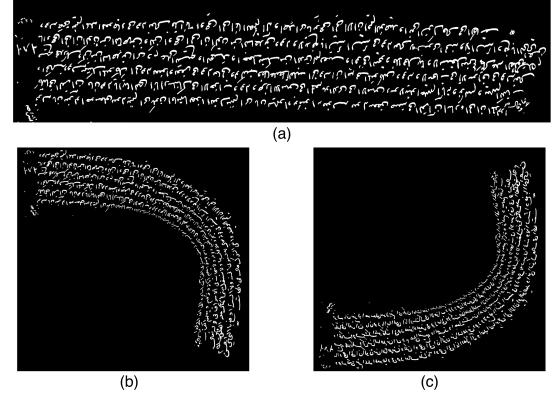


Fig. 8. Example of generated curved lines: (a) shows the original straight lines section, (b) is the result of warping (a) by 90 degrees in the middle to generate curved lines, and (c) is the mirrored image of (b) in the vertical direction.

### D. Results on VML-MOC dataset

The VML-MOC dataset contains both types, straight text lines and curved text lines. Number of straight text lines is substantially greater than the number of curved text lines. This imbalance causes the FCN to overfit on the straight text lines. This in turn leads to fragmented blob lines when predicting over the curved text lines. Therefore, to compensate this imbalance, we generated images containing artificially curved text lines. We selected the document image parts with straight lines and warp these images 90 degrees from their middle. Furthermore, each one of those warped lines was mirrored in the horizontal and vertical directions resulting in curved lines in four directions. Figure 8 illustrates this procedure. The FCN+EM that is trained with augmented curved text lines (FCN+EM+Aug) outperforms the FCN+EM that is trained only with the training set (Table II). But FCN+EM+Aug still underperforms a learning free algorithm [14].

TABLE II  
RESULTS ON VML-MOC DATASET

	Line IU	Pixel IU	DR	RA	FM
FCN+EM	25.04	48.71	26.45	17.73	31.09
FCN+EM+Aug	35.12	60.97	84.43	58.89	68.77
[14]	60.99	80.96	-	-	-
Human+EM	<b>96.62</b>	<b>99.01</b>	<b>90.41</b>	<b>91.74</b>	<b>91.03</b>

### E. Results on DIVA-HisDB dataset

We compare the results with the results of Task-3 from ICDAR 2017 competition on layout analysis for medieval manuscripts [19]. Task-3's scope of interest is only the main text lines but not the interlinear glosses. We removed these glosses prior to all our experiments using the ground truth. It should be noticed that Task-3 participants removed these glosses using their own algorithms.

Table III presents a comparison of our methods with the participants of ICDAR 2017 competition on layout analysis for challenging medieval manuscripts for text line extraction.

The FCN+EM method outperformed all the competition participants in both the Line IU and the Pixel IU measures on the books CB55 and CSG18. It can obtain a perfect Line IU score on the book CSG863 and its Pixel IU is on par with the best performing method in the competition.

TABLE III

COMPARISON WITH THE TASK-3 RESULTS OF THE ICDAR2017 COMPETITION ON LAYOUT ANALYSIS FOR CHALLENGING MEDIEVAL MANUSCRIPTS [19].

	CB55		CSG18		CSG863	
	LIU	PIU	LIU	PIU	LIU	PIU
FCN+EM	<b>100</b>	<b>97.64</b>	<b>97.65</b>	<b>97.79</b>	<b>100</b>	97.18
System-2	84.29	80.23	69.57	75.31	90.64	93.68
System-6	5.67	30.53	39.17	54.52	25.96	46.09
System-8	99.33	93.75	94.90	94.47	96.75	90.81
System-9+4.1	98.04	96.67	96.91	96.93	98.62	<b>97.54</b>

### F. Discussion

An observable pattern in the results is the parallel flow of line IU values and pixel IU values while RA values are fluctuating in comparison to DR values. Clearly such counter-intuitive behaviour of a metric is not preferable in terms of interpretability of the results. On the other hand, ICDAR 2017 evaluator can not handle the cases where a text line consists of multiple polygons. Such case arises from MRCNN results. MRCNN segments a text line instance correctly but represents it as multiple polygons with the same label. Evaluating MRCNN results in their raw form yields to low values unfairly (Table 9). Because ICDAR 2017 evaluator calculates an IU score for each possible pair of ground truth polygons and prediction polygons then selects the pairs with maximum IU score as the matching pairs. Consequently a text line represented by multiple polygons is considered only by its largest polygon.

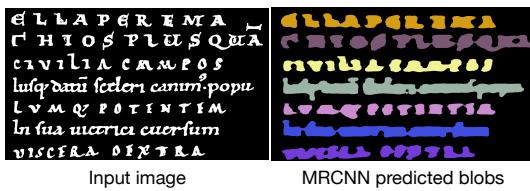


Fig. 9. MRCNN method correctly predicts text line pixels but its results are not fairly evaluated due to disconnected polygons.

## VI. CONCLUSION

This paper presents a supervised text line segmentation method FCN+EM. The FCN detect the blob lines that strike through the text lines and the EM extracts the pixels of text lines with the guidance of the detected blob lines. FCN+EM does not make any assumption about the text line orientation or text line height. The algorithm is very effective in detecting cramped, crowded and touching text lines. It has a superior performance on VML-AHTE and DIVA-HisDB datasets but comparable results on VML-MOC dataset.

## ACKNOWLEDGMENT

The authors would like to thank Gunes Cevik for annotating the ground truth. This work has been partially supported by the Frankel Center for Computer Science.

## REFERENCES

- [1] Q. N. Vo and G. Lee, "Dense prediction for text line segmentation in handwritten document images," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3264–3268.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] V. B. Campos, V. R. Gómez, A. H. T. Rossi, and E. V. Ruiz, "Text line extraction based on distance map features and dynamic programming," in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, 2018, pp. 357–362.
- [6] K. M. Sayre, "Machine recognition of handwritten words: A project report," *Pattern recognition*, vol. 5, no. 3, pp. 213–228, 1973.
- [7] K. Koffka, *Principles of Gestalt psychology*. Routledge, 2013, vol. 44.
- [8] G. Renton, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, "Handwritten text line segmentation using fully convolutional network," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 5. IEEE, 2017, pp. 5–9.
- [9] R. Saabni, A. Asi, and J. El-Sana, "Text line extraction for historical document images," *Pattern Recognition Letters*, vol. 35, pp. 23–33, 2014.
- [10] B. Barakat, A. Drobny, 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.
- [11] 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.
- [12] A. Fischer, M. Baechler, A. Garz, M. Liwicki, and R. Ingold, "A combined system for text line extraction and handwriting recognition in historical documents," in *2014 11th IAPR International Workshop on Document Analysis Systems*. IEEE, 2014, pp. 71–75.
- [13] T. Gruening, 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.
- [14] B. K. Barakat, R. Cohen, J. El-Sana, and I. Rabaev, "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.
- [15] D. Aldavert and M. Rusiñol, "Manuscript text line detection and segmentation using second-order derivatives," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 293–298.
- [16] B. Gatos, N. Stamatopoulos, and G. Louloudis, "ICDAR2009 handwriting segmentation contest," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 1, pp. 25–33, 2011.
- [17] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, and A. Alaei, "ICDAR 2013 handwriting segmentation contest," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1402–1406.
- [18] M. Diem, F. Kleber, S. Fiel, T. Grüning, and B. Gatos, "cbad: ICDAR2017 competition on baseline detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1355–1360.

- [19] 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.
- [20] S. Pletschacher and A. Antonacopoulos, “The page (page analysis and ground-truth elements) format framework,” in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 257–260.
- [21] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.
- [22] B. Barakat and J. El-Sana, “Binarization free layout analysis for arabic historical documents using fully convolutional networks,” in *Arabic Script Analysis and Recognition (ASAR), 2nd International Workshop*. IEEE, 2018, pp. 26–30.
- [23] 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.
- [24] 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.
- [25] M. Kubovy and M. Van Den Berg, “The whole is equal to the sum of its parts: A probabilistic model of grouping by proximity and similarity in regular patterns.” *Psychological review*, vol. 115, no. 1, p. 131, 2008.
- [26] 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.
- [27] 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.
- [28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.