signals

MDPI

Article

Text Line Extraction in Historical Documents Using Mask R-CNN

Ahmad Droby [1,*], Berat Kurar Barakat [1], Reem Alaasam [1], Boraq Madi [1], Irina Rabaev [2] and Jihad El-Sana [1]

[1] Department of Computer Science, Ben-Gurion University of the Negev, Be'er Sheva 8410501, Israel
[2] Department of Software Engineering, Shamoon College of Engineering, Be'er Sheva 8410802, Israel
* Correspondence: drobya@post.bgu.ac.il

Abstract: Text line extraction is an essential preprocessing step in many handwritten document image analysis tasks. It includes detecting text lines in a document image and segmenting the regions of each detected line. Deep learning-based methods are frequently used for text line detection. However, only a limited number of methods tackle the problems of detection and segmentation together. This paper proposes a holistic method that applies Mask R-CNN for text line extraction. A Mask R-CNN model is trained to extract text lines fractions from document patches, which are further merged to form the text lines of an entire page. The presented method was evaluated on the two well-known datasets of historical documents, DIVA-HisDB and ICDAR 2015-HTR, and achieved state-of-the-art results. In addition, we introduce a new challenging dataset of Arabic historical manuscripts, VML-AHTE, where numerous diacritics are present. We show that the presented Mask R-CNN-based method can successfully segment text lines, even in such a challenging scenario.

Keywords: text line extraction; deep learning; handwritten documents image analysis; historical documents; Mask R-CNN

check for updates

Citation: Droby, A.; Kurar Barakat, B.; Alaasam, R.; Madi, B.; Rabaev, I.; El-Sana, J. Text Line Extraction in Historical Documents Using Mask R-CNN. *Signals* **2022**, *3*, 535–549. https://doi.org/10.3390/signals3030032

Academic Editors: Francisco Martínez González and Mohammed K. A. Kaabar

Received: 18 May 2022
Accepted: 1 August 2022
Published: 4 August 2022

1. Introduction

Text line extraction is an early step for higher level document image analysis tasks, such as word segmentation [1,2] and word recognition [3,4]. The performance of these tasks depends on the quality of the text line extraction. Although segmentation of almost horizontal handwritten text lines have been extensively studied, it still presents significant challenges and regarded as an open problem in the domain of historical documents.

With the advances of hardware, deep learning approaches have achieved state-of-the-art performance in many computer vision applications. Recently, the document image analysis community has explored the use of deep learning for handwritten text line detection and segmentation. Early work used convolutional LSTM for text line detection by predicting the bounding boxes of text lines [5,6]. Recent deep learning-based methods are based on Fully Convolutional Networks (FCN) and show promising capabilities for detecting handwritten text lines [7–11]. However, these methods are designed to detect the baselines, which are lines that pass through the bottom part of the main body of the characters that belong to a text line. Thus, those methods do not address the problem of text line segmentation.

Recently, Mask R-CNN has become start-of-the-art model for instance segmentation of objects in images, and also has been applied for text segmentation in natural scene images [12–16]. In this paper, we apply Mask R-CNN [17] for text line segmentation in historical document images. Historical document images are very different from natural scene images. First, historical documents frequently exhibit degradation, e.g., bleed-through, faded ink, and presence of stains and holes. Second, handwritten text lines usually have inconsistent heights, varying interline spaces, and touching and overlapping characters. Therefore, we need to adapt Mask R-CNN for text line segmentation in historical documents.

This paper has two main contributions.

1. We present a text line extraction method that uses Mask R-CNN for text line extraction in historical documents. The Mask R-CNN model is trained to extract text lines from patches, which are further merged to form text lines of an entire page (see Figure 1);
2. We introduce a new challenging dataset of historical documents in Arabic, VML-AHTE (Visual Media Lab—Arabic Handwritten Text Line Extraction). The documents in VML-AHTE exhibit a challenging scenario, where multiple diacritics are present.

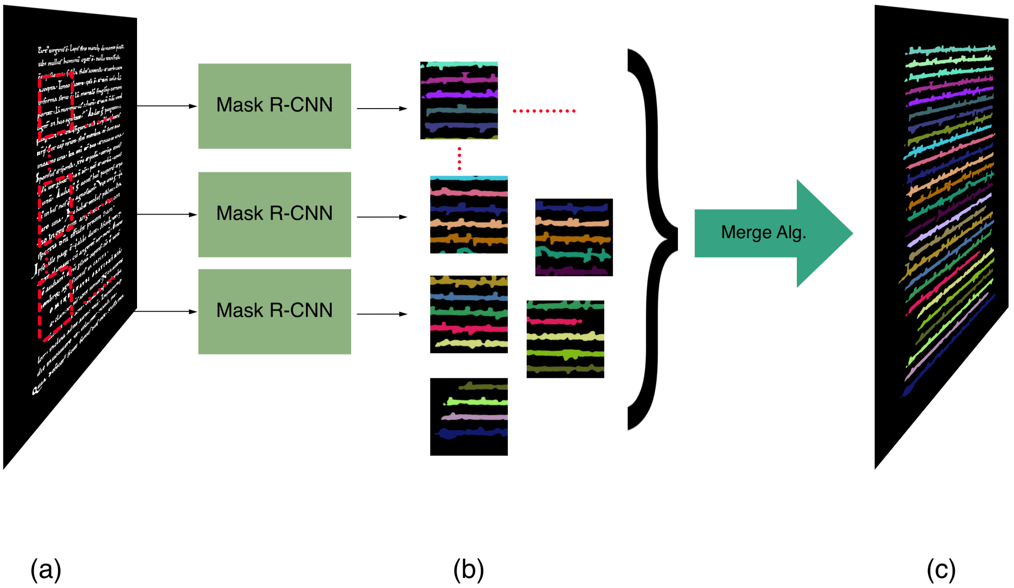


Figure 1. An overview of the proposed method. An overlapping sliding window traverses the input image (**a**) and segments the patches (windows) using the trained Mask R-CNN model. An example of the segmented patches are shown in (**b**). The segmented patches are merged together producing the segmented text lines in a page (**c**).

The proposed Mask R-CNN method was evaluated on three different datasets of historical document, Diva-HisDB [18], ICDAR2015-HTR [19], and the newly introduced VML-AHTE dataset. To measure the performances, we adopted two different methods: the pixel level metrics from ICDAR 2013 handwriting segmentation contest [20], and the polygonal level metrics from Diva Line Segmentation Evaluator [21]. We show that the Mask R-CNN text line extraction obtains state-of-the-art results on Diva-HisDB, and can successfully extract text lines on the ICDAR2015-HTR, and VML-AHTE datasets—a pair of challenging scenarios in which documents include overlapping and crossed out lines and multiple diacritics, respectively.

The remainder of this paper is organised as follows. In Section 2 related works for text line detection and text line extraction are reviewed. Section 3 provide background information about Mask R-CNN. The text line extraction method are described in Section 4, and the used datasets and evaluations are presented in Section 5 and Section 6, respectively. Section 7 present experimental results and discussion. Finally, we conclude and discuss future work in Section 8.

2. Related Work

With the advances of hardware, especially with the incorporated use of GPUs, deep neural networks (DNNs) have achieved new standards in many research frontiers. Early deep learning methods by [5,6] combine Long Short-Term Memory (LSTM) networks with convolutional layers to predict bounding boxes of text lines. Following the sequential LSTM model, dense prediction by FCN has become very popular. Refs. [22,23] estimate the x-height of text lines using FCN and further apply Line Adjacency Graphs to post-process FCN output to split touching lines. Refs. [7,8] use FCN to predict x-height of text lines. Ref. [10] uses FCN to simultaneously perform layout analysis and baseline extraction. Ref. [11] applied FCN for challenging manuscript images with multi-skewed, multi-directed and curved handwritten text lines.

Mask R-CNN has become start-of-the-art model for instance segmentation of objects in natural images, and also has been applied for text segmentation in natural scenes [12–15].

Learning-based methods detect baselines or x-heights of text lines, and only few of them proceed to perform text line extraction. In the case of handwritten document, where text lines have inconsistent heights, interline spaces, and touching and overlapping ascenders and descenders, it is vital to perform a precise text line extraction, which includes clustering disconnected elements along text lines and disconnecting touching elements among adjacent text lines.

3. Background

In this section, we overview Mask R-CNN, which is utilized in our approach without any modification on its architecture, but was retrained using our dataset.

Mask R-CNN

Mask R-CNN [17] is a well known instance segmentation network, where a label is assigned to each pixel of the image. The model first runs the input image through a CNN backbone to extract its feature map, which is then passed to a Region Proposal Network (RPN). From the image's feature map, the RPN generate multiple Regions Of Interest (RoI) in the image. The feature map of each generated RoI is then passed to a fully connected network to predict its semantic class and bounding box. The feature map of RoI is also fed into a CNN to generate a corresponding binary mask (see Figure 2).

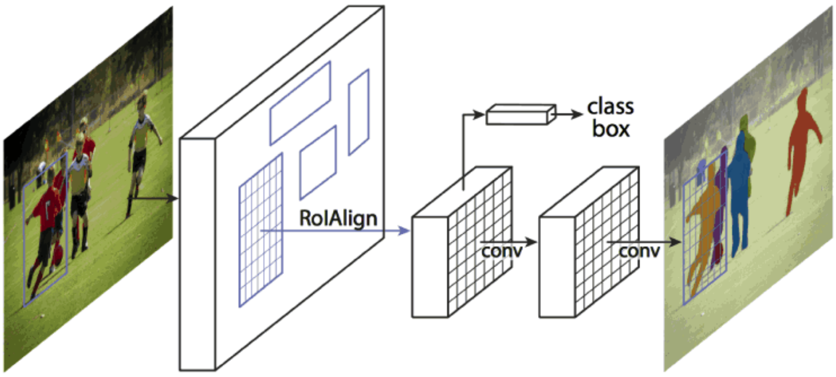


Figure 2. Mask R-CNN framework. (Figure 2 was taken from the Mask R-CNN paper [17]).

4. Method

We define a text line fragment or simply a line fragment as a portion of a text line that may include several characters or fraction of words or characters, at pixel level. Figure 3 shows several line fragment examples.

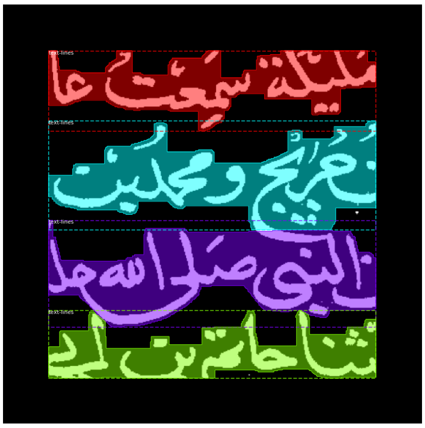


Figure 3. Example of text line fragments in a patch.

We take a divide and concur approach to segment the text line in a page. Initially, we train a Mask R-CNN to segment text line fragment in a fixed size patches.

To segment text lines in a page, P, we perform:

- The page P is sub-divided into overlapping fixed size patches;
- The patches are passed to the Mask R-CNN to segment the text line fragments present in each patch;
- The line fragments extracted from the patches are merged to produce the complete segmentation of the text lines within the whole page.

The Figure 1 illustrates the flow of the proposed method. Next, we describe each of these steps.

4.1. Training Mask R-CNN

The weights of the Mask R-CNN were initialized using the weights from a ResNet [24] network trained on COCO object segmentation dataset [25]. To train Mask R-CNN, we extract 50 K patches of size 350 × 350 from the datasets. Each extracted patch, p, is labeled by a set of masks corresponding to line fragments exits in p. Each mask corresponds to one line fragment. The masks are stacked along the third axis, as illustrated in Figure 4. A mask of a line fragment is a binary image of the same dimension as the patch, with ones in the area of the text line and zeros everywhere else.

We adopted patch size of 350 × 350 due to memory limitation, since 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 often results in details loss, which, in turn, reduces the segmentation accuracy.

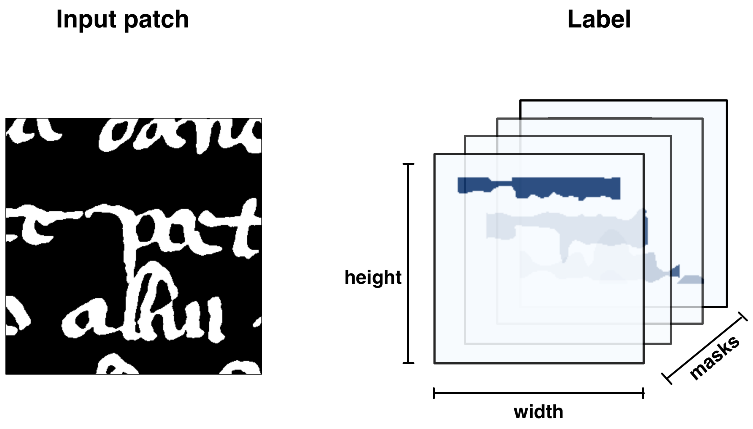


Figure 4. Illustration of patch labeling used to train the Mask R-CNN. Patches are extracted from page images and labeled by stacking the text line mask along the channel axis.

4.2. Dividing a Page into Patches

To divide a page, P, into an overlapping patches we apply a sliding window that traverses the entire page left-to-right and top-to-bottom in an ordered manner, as illustrated in Figure 1. Based on the experimental results, the stride in the vertical and horizontal directions of the sliding window is set to a percentage of the patch size; during our experiments we set the stride to $\lfloor \frac{Patch\_size}{7} \rfloor$, which was determined experimentally.

The trained Mask R-CNN is applied on each window and segments the patch into multiple text line fragments (masks) which are then merged together to form the segmentation of the text lines within the whole page.

4.3. Merging Patches

To obtain text line segmentation, we need to merge the obtained line fragments to form complete text lines. In order to merge fragments of the text lines predicted by the Mask R-CNN model, we employ the following merging algorithm:

We initialize the set of the text lines mask, L, as an empty set. Then, we iterate over the patches generated by the sliding window in order (left to right and top down). Let $m_1, m_2, \ldots, m_n$ be the masks predicted for a patch p. For each mask m_i, we check if it intersects a line(s) and the intersection area is above a predetermined threshold T, we merge m_i with the line l_j which has the largest intersection with m_i. Otherwise, m_i defined a new line, l_k, which is added set L. When all the patches generated by the sliding window are processed, the set L is returned (See Algorithm 1). The value of the threshold T depends on the amount of noise present in the dataset. Since the datasets we used during our experiments are relatively clean, we found that setting $T = 50$ produces good results.

Algorithm 1: Page segmentation method using Mask R-CNN

```
Result: $L$, the set of the lines masks
$L \leftarrow \{\}$;
while has next window do
    $p \leftarrow$ next sliding window;
    $m_1, m_2, \ldots, m_n \leftarrow MRCNN(p)$ ;
    for $m_i \in \{m_1, m_2, \ldots, m_n\}$ do
        if $m_i$ intersects a line in $L$ then
            $l \leftarrow argmax_{l_j \in L}\{|l_j \cap m_i|\}$;
            if $|l \cap m_i| > T$ then
                $L \leftarrow (L \setminus \{l\}) \cup \{(l \cup m_i)\}$
            else
                $L \leftarrow L \cup \{m_i\}$
            end
        else
            $L \leftarrow L \cup \{m_i\}$
        end
    end
end
```

4.4. Text Line Extraction Refinement

In rare cases, the results from the previous step may split an individual text line into two lines or combine two text lines into one. Following the merging step, we identify and fix most of these flaws in a refinement stage. Splitting merged lines and combining sub-lines are handled separately, as discussed next.

To combine masks, we iterate over the detected masks $m_1, m_2, \ldots, m_n$. Mask m_i is combined with mask m_j if σ% of the area of mask m_j is inside the page's horizontal section (dashed red rectangle in Figure 5) containing the mask m_i, as depicted in Figure 5. To make sure that the two masks are inside the same horizontal section, we set $\sigma = 90$.

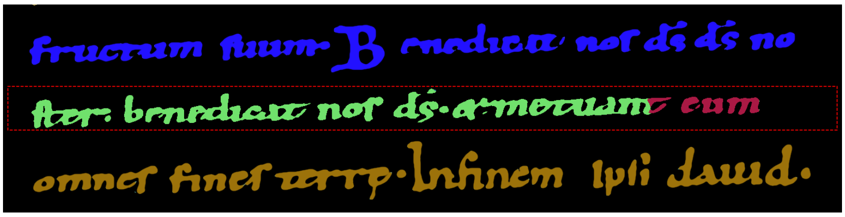
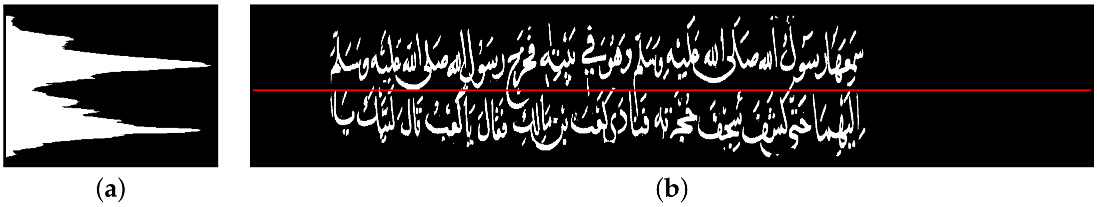


Figure 5. The red and green masks will be combined, because the red mask is completely contained within the horizontal section of the green mask (the dashed red rectangle).

To split merged masks, we calculate the average height, h, of the detected masks. A mask m_i is split if $h_i > \alpha h$, where h_i is the height of the mask m_i, and α is a predefined parameter. In our experimental study, we found that $\alpha = 1.3$ produces the best results. We calculate the horizontal profile and split the mask horizontally at the valley (minimum point) which is closest to the middle of the mask, as shown in Figure 6. Since not all the masks uphold the criteria $h_i > \alpha h$, only masks with height significantly above average are split. In addition, as can be seen in Figure 6, it is assumed that the text lines are horizontal, which allows us to determine the ideal split point by considering the text lines' horizontal profile. Although splitting merged masked this was separate the main body of the text lines, it is not prefect, as some ascenders and descenders will most likely be cut and segmented incorrectly.



(**a**) (**b**)

Figure 6. Splitting combined lines: (**a**) the horizontal profile of an image patch, and (**b**) the mask is split at the minima closest to the center of the mask. The split line is shown in red.

5. Datasets

We evaluate the performance of the method on three different datasets: the newly introduced dataset VML-AHTE, and the well-known historical documents datasets, Diva-HisDB and ICDAR2015-HTR.

1. The VML-AHTE (Visual Media Lab—Arabic Historical Text line Extraction) dataset consists of 30 pages from several historical manuscripts collected from the Islamic manuscripts digitization project, the Leipzig University Library (https://www.islamic-manuscripts.net). This project (accessed on 1 April 2019) provides digital access to a group of about 55 Arabic, Persian, and Turkish manuscripts acquired by the Leipzig University Library. The documents exhibit rich diacritics, and touching and overlapping characters. The dataset was split into training (20 pages) and test (10 pages) sets. For every page image we built a corresponding ground truth in four formats: PAGE xml format [26], pixels labels, Diva pixel labels [27], and bounding polygons, as illustrated in Figure 7.
   The PAGE xml files were created manually by native Arabic speakers using Aletheia tool [28], and the pixel labels and bounding polygons were generated automatically from the PAGE xml files. The dataset is available for downloading (https://www.cs.bgu.ac.il/~berat/) (accessed on 1 Augest 2022), together with the official train and test sets.
2. Diva-HisDB dataset [18] is a collection of three medieval manuscripts $GB55$, $CSG18$ and $CSG863$, with complex layout structure, as shown in Figure 8. The dataset includes 150 pages (50 in each manuscript) with annotation for main text, comments

and decorations. Each manuscript is split into training, validation, public, and private test sets. The train set contains 20 pages and each one of the validation and the test sets contain 10 pages. In this paper, we train on the training set and test on the private test set to compare with the results of ICDAR2017 competition [27].

3. ICDAR2015-HTR dataset [19] was introduced and used in ICDAR 2015 Competition on Handwritten Text Recognition. It includes 796 pages from the Bentham collection, generated by the Scriptorium project. The dataset is split into training and test sets containing 746 and 50 pages, respectively. The dataset poses many difficulties for text line detection and extraction, such as varying quality of images, writing styles, and crossed out text (see Figure 9). Although this dataset is used for text recognition and its ground-truth quality and consistency is not ideal (we will discuss this in Section 7.2.2), it was chosen to demonstrate that the proposed method can deal with the significant variabilities and difficulties present in this dataset.

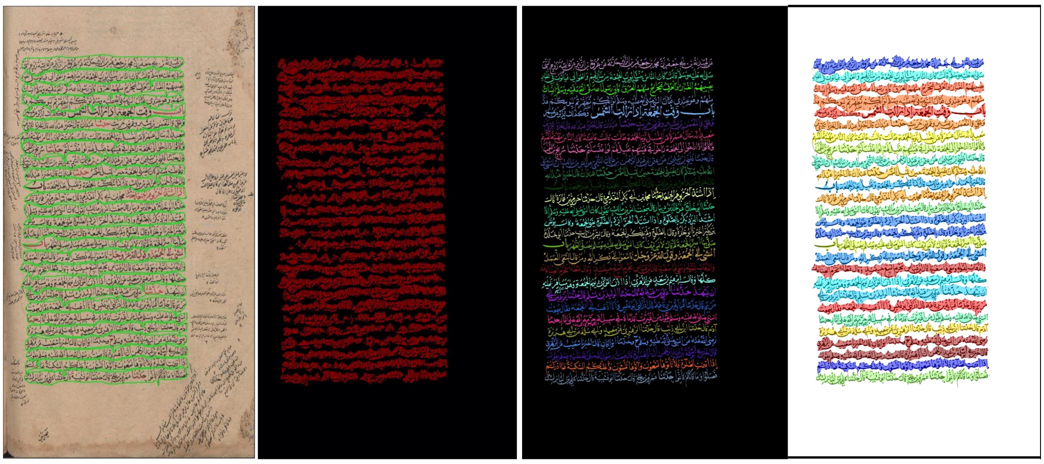


Figure 7. VML-AHTE dataset includes four ground truth formats: PAGE xml, pixel labels, DIVA pixel labels, and bounding polygons.

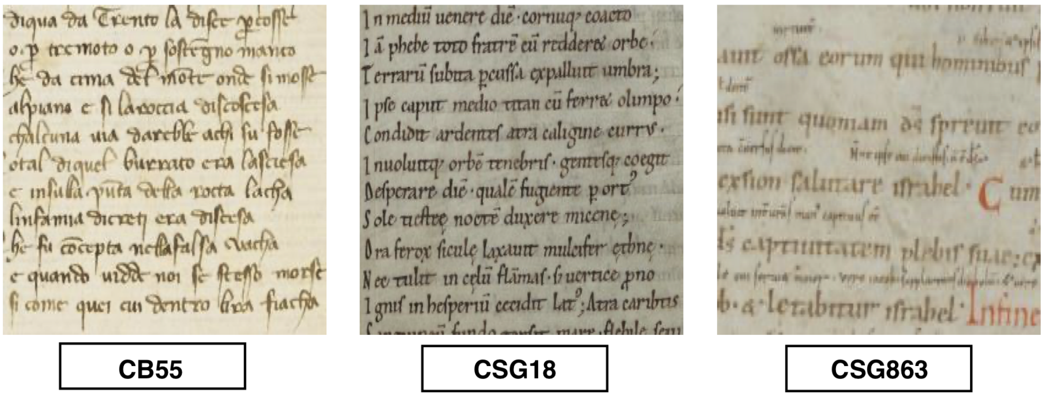


Figure 8. An example of patch from each collection in the Diva-HisDb dataset.

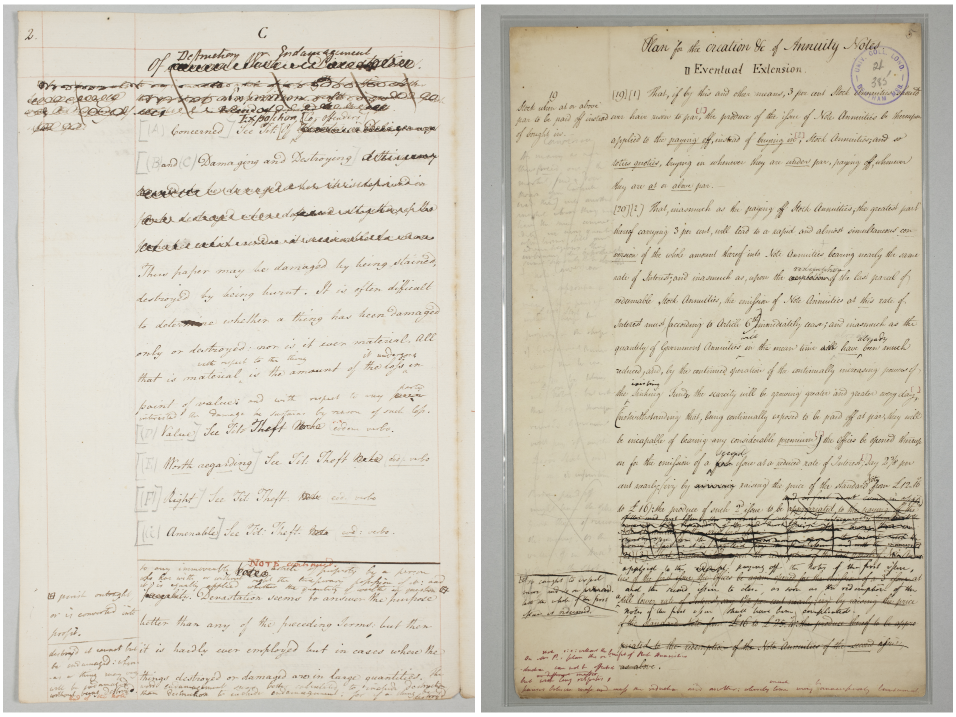


Figure 9. Example pages from the ICDAR2015-HTR dataset.

6. Evaluation

For evaluation, we used two different tools: ICDAR2013 line segmentation evaluator [29] to evaluate the pixel label results, and DIVA Line Segmentation Evaluator tool [21] to evaluate the bounding polygons results. Below, we detail the metrics used by each of the two evaluator tools.

6.1. Diva Evaluation Metrics

The DIVA Line Segmentation Evaluator (https://github.com/DIVA-DIA/DIVA_Line_Segmentation_Evaluator) (accessed on 1 April 2020) is based on the Intersection over Union (IU). We compute the IU score for each possible pair of Ground Truth (GT) and Prediction (P) polygons according to the Equation (1), where IP denotes the number of intersecting foreground pixels among the pair of polygons and 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 among the ground truth (GT) and predicted (P) polygons. Then, pixel IU and text line IU are calculated among these matching pairs.

$$\mathrm{IU} = \frac{\mathrm{IP}}{\mathrm{UP}} \tag{1}$$

Pixel IU is measured at pixel level. For each matching pair, line TP, line FP, and line FN are computed. Line TP is the number of correctly classified text line pixels, line FP is the number of pixels falsely classified as text line pixels, and line FN is the number of pixels falsely classified as background. Then, pixel IU is calculated according to Equation (2), where Σ_{TP} is the global sum of line TPs, Σ_{FP} is the global sum of line FPs, and Σ_{FN} is the global sum of line FNs.

$$\text{Pixel IU} = \frac{\Sigma_{\mathrm{TP}}}{\Sigma_{\mathrm{TP}} + \Sigma_{\mathrm{FP}} + \Sigma_{\mathrm{FN}}} \tag{2}$$

Line IU is measured at line level. For each matching pair, line precision and line recall are computed according to Equations (3) and (4). Line IU is calculated according to

Equation (5), where CL is the number of correctly predicted lines, ML is the number of missed lines, and EL is the number of falsely predicted lines.

$$\text{Line precision} = \frac{\text{line TP}}{\text{line TP} + \text{line FP}} \tag{3}$$

$$\text{Line recall} = \frac{\text{line TP}}{\text{line TP} + \text{line FN}} \tag{4}$$

$$\text{Line IU} = \frac{\text{CL}}{\text{CL+ML+EL}} \tag{5}$$

For each matching pair:

- A line is correctly predicted (CL) if both, the line precision and the line recall, are above a threshold value T;
- A line is missed (ML) if the line recall is below the threshold value T;
- A line is falsely predicted (EL) if the line precision is below the threshold value T.

In order to be able to compare our results to the ICDAR2017 competition's results [27], we used the same threshold they did, $T = 75\%$.

6.2. ICDAR 2013 Evaluation Metrics

The input to the ICDAR2013 evaluation tool is a document image, its ground truth, and the difference between the input and the ground truth. The output is three metrics: the detection rate (DR), the recognition accuracy (RA), and the performance metric (FM).

For a given image, let R_i be the set of points inside the i_{th} detected line segment, G_j be the set of points inside the j_{th} line segment in the ground truth, and $T(p)$ is a function that counts the points in a given set p, then $MatchScore(i, j)$ is calculated as follows:

$$MatchScore(i, j) = \frac{T(G_j \cap R_i)}{T(G_j \cup R_i)} \tag{6}$$

ICDAR2013 evaluation tool considers a region pair (i, j) as a one-to-one match if the $MatchScore(i, j)$ is equal or above the threshold T_a. In our experiment, we set the threshold to be $T_a = 90\%$. Let N_1 and N_2 be the number of ground-truth and the detected lines, respectively, and let M be the number of one-to-one matches. The tool calculates the DR, RA, and FM as follows:

$$DR = \frac{M}{N_1}, \quad RA = \frac{M}{N_2}, \quad FM = \frac{2 \times DR \times RA}{DR + RA}$$

7. Results and Discussion

To evaluate our approach, we conducted an intensive experimental study using the newly introduced dataset VML-AHTE, the DIVA, and ICDAR2015-HTR datasets. The results was evaluated using ICDAR2013's and Diva's evaluation tools.

7.1. Training

The model was trained using a computer equipped with GeForce GTX 1080 GPU for 100 epochs. The training process took between 2.5 and 3.5 h.

7.2. Experimental Results

In this section, we report and analyse the experimental results. In Section 7.2.1 we report the results on VML-AHTE and DIVA datasets, and in Section 7.2.2 we present the results on ICDAR2015-HTR dataset.

7.2.1. VML-AHTE and DIVA Datasets

For DIVA and the VML-AHTE datasets, the model was trained and tested using binary images. Since the refinement stage assumes straight horizontal lines, weakening the generality of the method, we evaluate the method twice, with and without the refinement stage (denoted as MRCNN+ and MRCNN, respectively). In order to evaluate the results using the ICDAR2013 metrics described in Section 6, we have extracted pixel level segmentation by labeling each foreground pixel (text pixels) in the binarized image with its corresponding label in the bounding polygon prediction generated by the proposed method (see Figure 10).

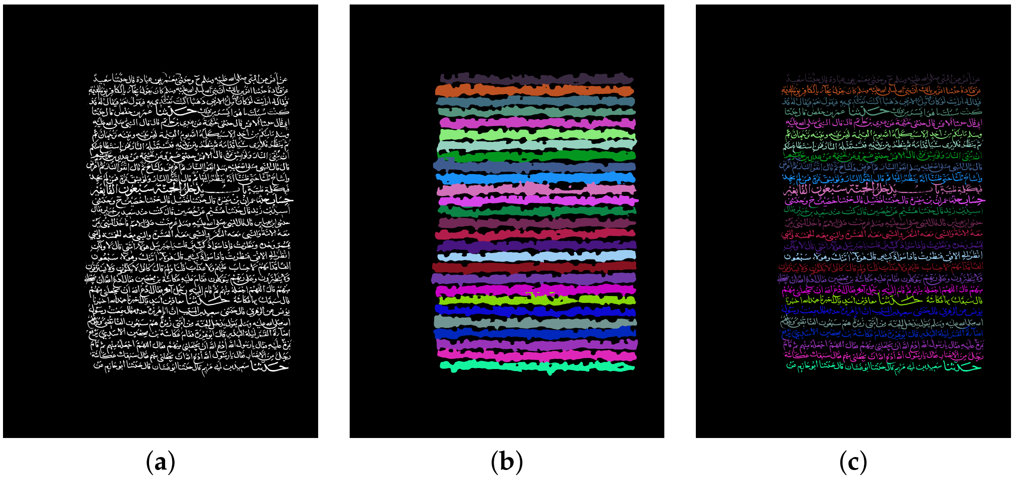


Figure 10. Mask R-CNN segmentation results: (**a**) an input page, (**b**) blobs prediction, and (**c**) pixel level prediction.

We compared the proposed method with the methods in task-3 of ICDAR 2017 competition on layout analysis for medieval manuscripts [27]. The scope of task-3 is limited to the main text lines and 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 also removed these glosses using their own algorithms.

Table 1 presents the results of our method against those from ICDAR 2017 competition on layout analysis for challenging medieval manuscripts for text line extraction (task-3). We can see from the Table 1, the refinement stage almost always improves the accuracy of the MRCNN method. With refinement, the Mask R-CNN method (MRCNN+) outperformed all of the competition participants in Pixel IU. MRCNN without refinement performs similarly to the four best preforming methods in the competition. Furthermore, Table 2 shows the ICDAR 2013 evaluation metrics results of the method on the Diva dataset.

Table 1. The results of the ICDAR2017 competition on layout analysis for challenging medieval manuscripts ([27]). Line IU and Pixel IU results of Task 3 of all competition participants and the proposed method. LIU denotes Line IU, and PIU denotes Pixel IU.

	CB55		**CSG18**		**CSG863**	
	LIU	**PIU**	**LIU**	**PIU**	**LIU**	**PIU**
System-2 (BYU)	84.29	80.23	69.57	75.31	90.64	93.68
System-6 (IAIS)	5.67	30.53	39.17	54.52	25.96	46.09
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
MRCNN	94.13	95.17	96.97	96.90	97.55	97.24
MRCNN+	95.42	**98.71**	**98.39**	**98.65**	97.24	**97.55**

Table 2. Results of the proposed method on the Diva dataset.

	CB55			CSG18			CSG863		
	DR	**RA**	**FM**	**DR**	**RA**	**FM**	**DR**	**RA**	**FM**
MRCNN	96.4	57.8	72.2	90.76	64.53	75.26	98.03	67.58	79.77
MRCNN+	97.70	98	97.85	95.11	97	96	98.63	98.26	98.44

Table 3 shows the proposed method performance on the VML-AHTE dataset. It lists the evaluation measures for the polygons and pixel level results, calculated using DIVA and ICDAR2013 line segmentation evaluator tools, respectively. In the above experiment, DR values are higher than RA values. Since the number of ground truth elements is constant, high DR values mean a high number of one to one matches. Consequently, low RA values in the presence of high number of one-to-one matches are caused by the high number of false predicted lines. This observation directly leads to the question why the Line IU values does not reflect the low RA values. The answer follows from the fact that ICDAR2013 matching score is 90% while the DIVA evaluator matching score is 75%, meaning that the evaluators use different threshold values.

Table 3. Results of the proposed method on the VML-AHTE dataset. DR denotes detection rate, RA denotes recognition accuracy, and FM denotes F-measure.

	Line IU	**Pixel IU**	**DR**	**RA**	**FM**
MRCNN	93.08	86.97	84.43	58.89	68.77
MRCNN+	97.83	89.61	88.14	87.78	87.96

Another observation is the fluctuating pattern of the RA values compared to the RA values, while the flow of Line IU and Pixel IU values is parallel. Such counter-intuitive behaviour of a metric is not preferable as it reduces the interpretability of the experimental results. Such behaviour is a direct result of the fact that the DIVA evaluator cannot handle text lines with multiple disconnected polygons. The results of the MRCNN does contain such cases, where the MRCNN polygons of the text line lead into multiple disconnected polygons with the same label. Therefore, evaluating the MRCNN results in their raw form yields low results unfairly. This fact can be seen in the results shown in Table 4, since DIVA evaluator calculates an IU score for every possible pair of ground truth and prediction polygons and considers the pairs with the maximum IU score as matching pairs. Therefore, text lines represented by multiple polygons are considered only by the largest polygon.

Table 4. Diva evaluator results of MRCNN extraction with disconnected bounding polygons on DIVA and VML-AHTE datasets.

Metric	**CB55**	**CSG18**	**CSG863**	**AHTE**
Line IU	67.98	95.99	63.24	90.45
Pixel IU	38.26	92.80	31.85	86.25

To solve this problem we combined the blobs of a single text line in the MRCNN results (see Figure 11). Two end points from two different blobs are combined if the distance between them is less than the average stroke width of blobs in that text line. Combination is performed by dilating the skeleton of two blobs. After combining the blobs, DIVA evaluator can measure the performance reasonably. Note that Table 1 shows the evaluation results of the MRCNN without combining the blobs.

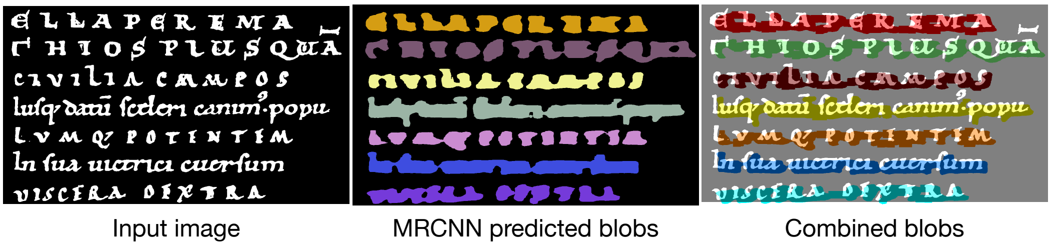


Figure 11. Disconnected MRCNN blobs are connected to enable applying Diva evaluator. As can be seen in the middle image, the Mask MRCNN segmented each line correctly, however, each line contain several disconnected segmented. On the right is the resulting polygon from combining the MRCNN blob prediction.

Figure 12 shows example prediction from both AHTE and DIVA datasets. We can see that in both cases the main body of the text lines are segmented correctly. However, in the AHTE dataset, the Mask R-CNN misses few punctuations and in some cases, part of the descender characters. The difference can be attributed to the more complex layout of the Arabic writing, compared to Latin writing in the DIVA dataset.

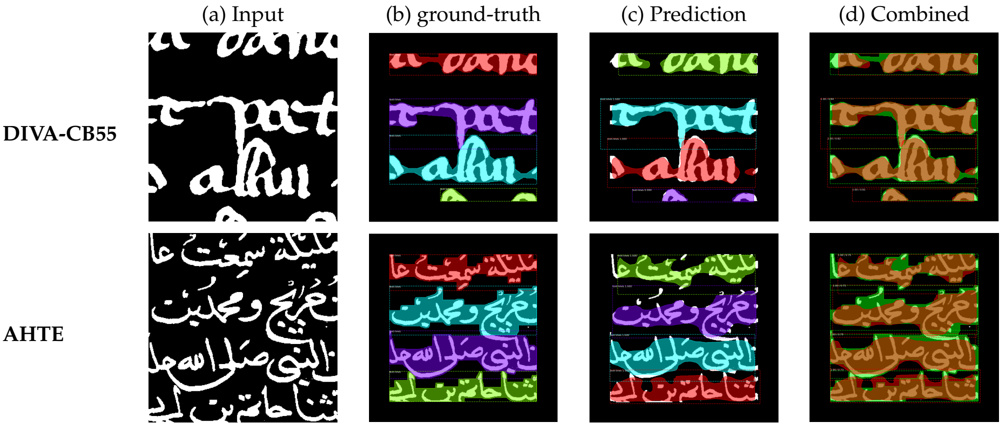


Figure 12. An example of patch prediction using Mask R-CNN: (**a**) an example patches from DIVA-CB55 and VML-AHTE datasets, (**b**) the ground-truth annotation of the patches, (**c**) the prediction results of the Mask R-CNN, and (**d**) the difference between the prediction results (red) and the ground-truth (green).

7.2.2. ICDAR2015-HTR

Similar to the other two datasets, we evaluated the method on the test set of ICDAR2015-HTR dataset and the results are presented in Table 5. As can be noted, the results of the method are comparatively low. By examining the results more closely, the low results can be attributed in part to the inconstancy of the ground-truth. We performed an error analysis and found that in some cases the annotation polygons of the ground-truth are wrapped tightly around the text lines, while in other cases they include comparatively large margins around the text (see Figure 13), which lead to discrepancies in Pixel IU. Furthermore, while the marginal text lines are not included as a part of the dataset annotation, they were segmented by our method, causing additional decrease in evaluation score. Figure 13 illustrates visual results of a number of sample images from the test set.

Table 5. Results of the proposed method on ICDAR2015-HTR dataset.

	DR	RA	FM
MRCNN	58.11	68.64	62.68
MRCNN+	57.21	72.74	63.81

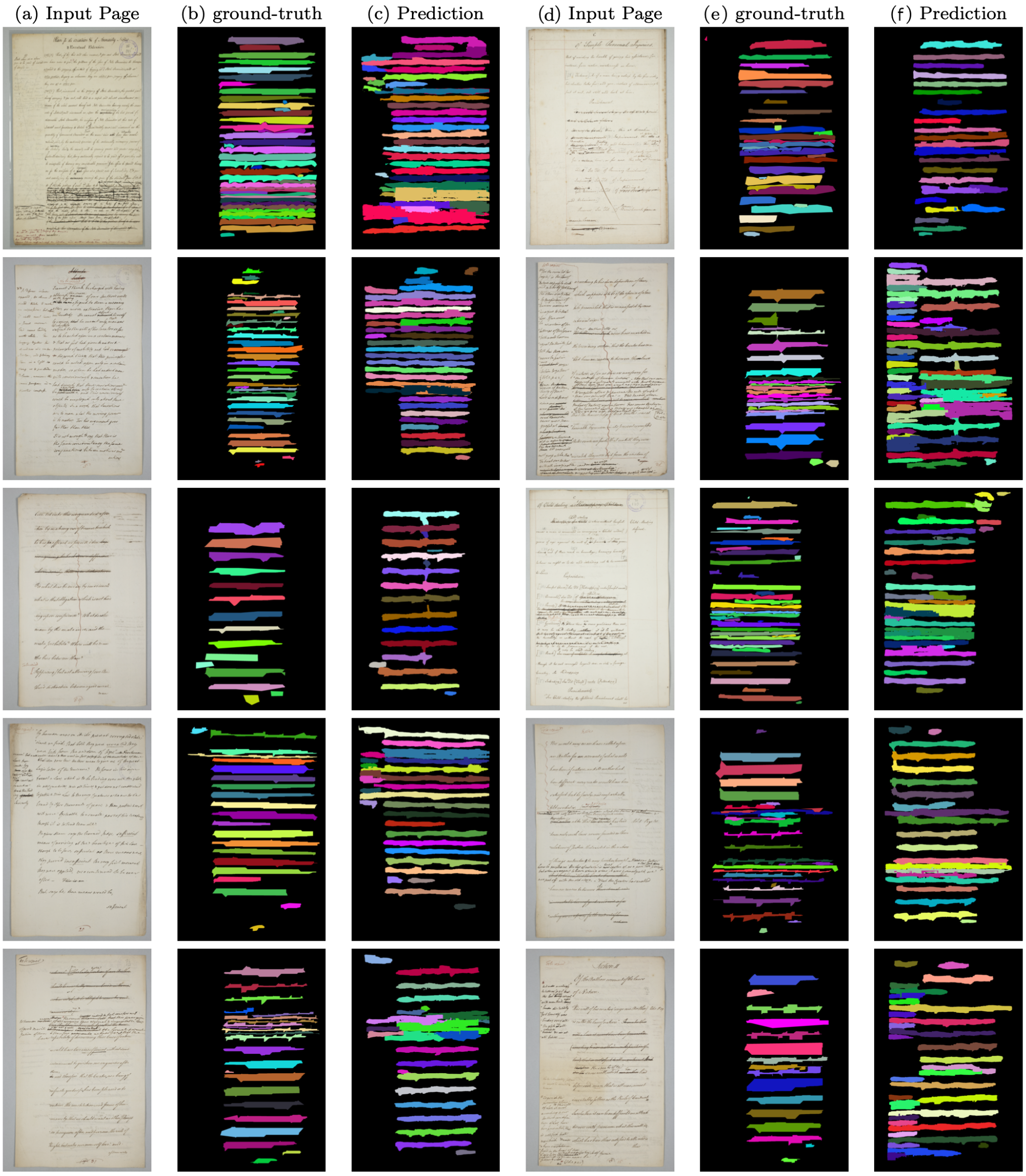


Figure 13. A sample of the segmentation results from ICDAR2015-HTR dataset. Columns (**a**,**d**) show the input page of the method, columns (**b**,**e**) show the ground-truth, and cloumns (**c**,**f**) are the prediction results. Note that the method successfully segments marginal notes which are not present in the ground truth.

8. Conclusions

We have presented a holistic method for text line segmentation in historical documents. A Mask R-CNN model is trained to segment text lines within small patches, which guide the text line segmentation within a whole page. In addition, we introduced a challenging dataset of Arabic manuscripts, named VML-AHTE, to test and evaluate the presented method in a challenging scenario, where numerous diacritics are present. The dataset is freely available for the downloading, together with its training and test sets split. We also set baselines for the text line extraction on the VML-AHTE dataset.

The presented method was tested on the well-known ICDAR2015-HTR and DIVA-HisDB datasets. We show that Mask R-CNN obtains results comparable with the best results of the ICDAR2017 competition on layout analysis for challenging medieval manuscripts. The comparison shows that the method consistently performs on par with the best-performing methods. Moreover, when we add the refinement step, the results surpass the best-performing methods.

It was shown that the proposed method segments lines with disconnected segments into multiple polygons with the same label, however, the Line and Pixel IU metrics are defined for text line segmentation with one polygon. To evaluate the method, we combined the polygons predicted by the MRCNN into one. In addition, we have showed that the evaluation results of presented method are not as good on the ICDAR2015-HTR dataset. However, by considering the visual results, we can see that the ICDAR2015-HTR ground-truth is inconsistent, as part of the annotation polygons are wrapped tightly around the text, while other annotation polygons are includes a large margin around the text.

In future research, we aim to adapt the method for segmenting text lines with different orientations and curvatures.

Author Contributions: Conceptualization, all authors; methodology, A.D.; software, A.D.; validation, A.D., B.K.B., R.A. and B.M.; formal analysis, A.D.; investigation, A.D. and B.K.B.; data curation, all authors; writing—original draft preparation, A.D., B.K.B., B.M., R.A. and I.R.; writing—review and editing, I.R. and J.E.-S.; visualization, A.D. and B.K.B.; supervision, I.R. and J.E.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was partially supported by The Frankel Center for Computer Science at Ben-Gurion University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Manmatha, R.; Srimal, N. Scale space technique for word segmentation in handwritten documents. In Proceedings of the International Conference on Scale-Space Theories in Computer Vision, Corfu, Greece, 26–27 September 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 22–33.
2. Varga, T.; Bunke, H. Tree structure for word extraction from handwritten text lines. In Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea, 31 August 1–September 2005; pp. 352–356.
3. Graves, A.; Liwicki, M.; Fernández, S.; Bertolami, R.; Bunke, H.; Schmidhuber, J. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *31*, 855–868. [CrossRef] [PubMed]
4. Liwicki, M.; Graves, A.; Bunke, H. Neural networks for handwriting recognition. In *Computational Intelligence Paradigms in Advanced Pattern Classification*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 5–24.
5. Moysset, B.; Kermorvant, C.; Wolf, C.; Louradour, J. Paragraph text segmentation into lines with recurrent neural networks. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 456–460.
6. Moysset, B.; Louradour, J.; Kermorvant, C.; Wolf, C. Learning text-line localization with shared and local regression neural networks. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 1–6.
7. Renton, G.; Chatelain, C.; Adam, S.; Kermorvant, C.; Paquet, T. Handwritten text line segmentation using fully convolutional network. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 5, pp. 5–9.
8. Renton, G.; Soullard, Y.; Chatelain, C.; Adam, S.; Kermorvant, C.; Paquet, T. Fully convolutional network with dilated convolutions for handwritten text line segmentation. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2018**, *21*, 177–186. [CrossRef]

9. Gruuening, T.; Leifert, G.; Strauss, T.; Labahn, R. A robust and binarization-free approach for text line detection in historical documents. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 236–241.
10. Oliveira, S.A.; Seguin, B.; Kaplan, F. dhSegment: A generic deep-learning approach for document segmentation. In Proceedings of the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 5–8 August 2018; pp. 7–12.
11. Kurar Barakat, B.; Droby, A.; Kassis, M.; El-Sana, J. Text Line Segmentation for Challenging Handwritten Document Images using Fully Convolutional Network. In Proceedings of the 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), Niagara Falls, NY, USA, 5–8 August 2018; pp. 374–379.
12. Huang, Z.; Zhong, Z.; Sun, L.; Huo, Q. Mask R-CNN With Pyramid Attention Network for Scene Text Detection. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 764–772. [CrossRef]
13. Xie, E.; Zang, Y.; Shao, S.; Yu, G.; Yao, C.; Li, G. Scene Text Detection with Supervised Pyramid Context Network. *CoRR* **2018**. Available online: http://xxx.lanl.gov/abs/1811.08605 (accessed on 1 April 2020).
14. Shivajirao, S.; Hantach, R.; Abbes, S.B.; Calvez, P. Mask R-CNN End-to-End Text Detection and Recognition. In Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), Boca Raton, FL, USA, 16–19 December 2019; pp. 1787–1793.
15. Duan, P.; Pan, J.; Rao, W. MaskS R-CNN Text Detector. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS), Dalian, China, 20–22 March 2020; pp. 5–8. [CrossRef]
16. Zhu, Y.; Zhang, H. Curved Scene Text Detection Based on Mask R-CNN. In *Proceedings of the Image and Graphics*; Zhao, Y., Barnes, N., Chen, B., Westermann, R., Kong, X., Lin, C., Eds.; Springer International Publishing: Cham, Swizterland, 2019; pp. 505–517.
17. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
18. Simistira, F.; Seuret, M.; Eichenberger, N.; Garz, A.; Liwicki, M.; Ingold, R. Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 471–476.
19. Sánchez, J.A.; Toselli, A.H.; Romero, V.; Vidal, E. ICDAR 2015 competition HTRtS: Handwritten Text Recognition on the tranScriptorium dataset. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1166–1170.
20. Stamatopoulos, N.; Gatos, B.; Louloudis, G.; Pal, U.; Alaei, A. ICDAR 2013 handwriting segmentation contest. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 1402–1406.
21. Alberti, M.; Bouillon, M.; Ingold, R.; Liwicki, M. Open Evaluation Tool for Layout Analysis of Document Images. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 4, pp. 43–47.
22. Vo, Q.N.; Lee, G. Dense prediction for text line segmentation in handwritten document images. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3264–3268.
23. Vo, Q.N.; Kim, S.H.; Yang, H.J.; Lee, G.S. Text line segmentation using a fully convolutional network in handwritten document images. *IET Image Process.* **2017**, *12*, 438–446. [CrossRef]
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
25. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, pp. 740–755.
26. Pletschacher, S.; Antonacopoulos, A. The PAGE (page analysis and ground-truth elements) format framework. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Los Alamitos, CA, USA, 23–26 August 2010; pp. 257–260.
27. Simistira, F.; Bouillon, M.; Seuret, M.; Würsch, M.; Alberti, M.; Ingold, R.; Liwicki, M. Icdar2017 competition on layout analysis for challenging medieval manuscripts. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 1361–1370.
28. Clausner, C.; Pletschacher, S.; Antonacopoulos, A. Aletheia—An Advanced Document Layout and Text Ground-Truthing System for Production Environments. In Proceedings of the 2011 International Conference on Document Analysis and Recognition, Beijing, China, 18–21 September 2011; pp. 48–52.
29. Gatos, B.; Stamatopoulos, N.; Louloudis, G. Icfhr 2010 handwriting segmentation contest. In Proceedings of the 2010 12th International Conference on Frontiers in Handwriting Recognition, Kolkata, India, 16–18 November 2010; pp. 737–742.