Full-Page Text Recognition: Learning Where to Start and When to Stop

Bastien Moysset*,†, Christopher Kermorvant†, Christian Wolf‡§

*À²îA SA, Paris, France
†Teklia SAS, Paris, France
‡Université de Lyon, CNRS, France
§INSA-Lyon, LIRIS, UMR5205, F-69621

Abstract—Text line detection and localization is a crucial step for full page document analysis, but still suffers from heterogeneity of real life documents. In this paper, we present a new approach for full page text recognition. Localization of the text lines is based on regressions with Fully Convolutional Neural Networks and Multidimensional Long Short-Term Memory as contextual layers.

In order to increase the efficiency of this localization method, only the position of the left side of the text lines are predicted. The text recognizer is then in charge of predicting the end of the text to recognize. This method has shown good results for full page text recognition on the highly heterogeneous Maurdor dataset.

I. INTRODUCTION

Most applications in document analysis require text recognition at page level, where only the raw image is available and no preliminary hand-made annotation can be used. Traditionally, this problem has mainly been addressed by separating the process into two distinct steps; namely the text line detection task, which is frequently proceeded by additional paragraph and word detection steps, and the text recognition task. In this work we propose a method, which couples these two steps tighter by unloading some of the burden from the difficult localization step to the recognition task. In particular, the localization step detects the starts of the text lines only. The problem of finding where to stop the recognition is solved by the recognizer itself.

A. Related work

Numerous algorithms have been proposed for the text line localization. Some are used in a bottom-up approach by grouping sub-components like connected components or black pixels into lines. RLSA [31] uses morphological opening on black pixels to merge the components that belong to the same text line. Similarly, Shi et al. [27] resort to horizontal ellipsoidal steerable filters to blur the image and merge the components of the text line. In [30], gradients are accumulated and filtered. Louloudis et al. [14] employ a Hough algorithm on the connected component centers while Ryu et al. [26] cluster parts of the connected components according to heuristic based successive splits and merges.

Other methods follow a top-down approach and split the pages into smaller parts. The XY-cut algorithm [20] looks for vertical and horizontal white spaces to successively split the pages in paragraphs, lines and words. Similarly, projection profile algorithms like Ouwayed et al. [22] are aimed at finding the horizontal whiter parts of a paragraph. This technique is extended to non-horizontal texts by methods like Nicolaou et al. [21] that dynamically finds a path between the text lines or by Tseng et al. [29] that use a Viterbi algorithm to minimize this path.

Techniques like the ones proposed by Mehri et al. [15] or Chen et al. [5] classify pixels into text or non-text but need post-processing techniques to constitute text lines.

These techniques usually work well on the homogeneous datasets they have been tuned for but need heavy engineering to perform well on heterogeneous datasets like the Maurdor dataset [4]. For this reason, Machine learning has proven to be efficient, in particular deep convolutional networks. Early work from Delakis et al. [6] classifies scene text image parts as text and non-text with a Convolutional Neural Network on a sliding window. In [17], paragraph images are split vertically using a recurrent neural network and CTC alignment. More recently, methods inspired from image object detection techniques like MultiBox [7], YOLO [25] or Single-Shot Detector (SSD) [13] have arisen. Moysset et al. [16] proposed a MultiBox based approach for direct text line bounding boxes detection. Similarly, Gupta et al. [10] and Liao et al. [12] use respectively YOLO based and SSD based approach for scene text detection. Moysset et al. [18] also propose the separate detection of bottom-left and top-right corners of line bounding boxes.


Finally, Bluche et al. [2] use a hard attention mechanism to directly perform full page text recognition without prior localization. The iterative algorithm finds the next attention point based on the sequence of seen glimpses modeled through the hidden state of a recurrent network.

B. Method overview

In this work, we address full page text recognition in two steps. First, a neural network detects where to start to recognize a text line, and a second network performs the text recognition and decides when to stop the process. More precisely, the former network detects the left sides of each text lines by predicting the value of the object position coordinates.
as a regression problem. This detection neural network system is detailed in Part II and the left-side strategy is explained in Part III-A. The latter network recognizes the text and predicts the end of the text of the line as described in Part III-B. The experimental setup is described in Part IV and results are shown and analyzed in Part V.

II. OBJECT LOCALIZATION WITH DEEP NETWORKS

A. Network description

In the lines of [7], we employ a neural network as a regressor to predict the positions of objects in images. The network predicts a given number \( N \) of object candidates. Each of these object candidates is indexed by a linear index \( n \) and defined by \( K \) coordinates \( l_n = \{ l_{nk} \}, k = 1 \ldots K \) corresponding to the position of the object bounding box in the document and a confidence score \( c_n \). As the number of objects in an image is variable, at test time, only the objects with a confidence score over a threshold are kept.

In order to cope with the small amount of training data available for document analysis tasks and to detect a large number of objects corresponding to our text lines, we adopted the method described in [16]. We do not use a fully connected layer at the end of the network that has as inputs features conveying information about the whole page image and, as outputs, all the object candidates of the page. Instead, our method is fully convolutional, which allows the network to share parameters over the different regressors. More precisely, we use a \( 1 \times 1 \) convolution to predict the objects locally and, consequently, to highly reduce the number of parameters in the network.

Layers constituted of Two-Dimensional Long-Short-Term-Memory cells (2D-LSTM) [8] are interleaved between the convolutional layers in order to recover the context information lost by the local nature of the detection.

The architecture is similar to the one in [16]. It is described in Table I and illustrated in Figure 1.

B. Training

We used the same training process as the one described in [7]. The cost function is a weighted sum between a confidence cost and the Euclidean distance between the two object positions (predicted and ground-truth):

\[
Cost = \sum_{n=0}^{N} \sum_{m=0}^{M} X_{nm} \left( \alpha ||l_n - t_m||^2 - \log(c_n) \right) - \sum_{n=0}^{N} (1 - \sum_{m=0}^{M} X_{nm}) \log(1 - c_n)
\]  

(1)

Here, the \( N \) object candidates have position coordinates \( l_n \) and confidence \( c_n \) while the \( M \) reference objects have position coordinates \( t_m \). \( \alpha \) is a parameter weighting localisation and confidence costs. As the output of the network (as well as the ground-truth information) is structured, a matching between the two of them is necessary in order to calculate the loss in equation 1. This matching is modelled through the variable \( X = \{ X_{nm} \}, a binary matrix. In particular, X_{nm}=1 if network output n has been matched to ground truth object m in the given image. Equation 1 needs to be minimised under constraints enforcing one-to-one matches, which can be solved efficiently through the Hungarian algorithm [19].

We could not confirm the claims reported in [7] who apply this matching process for object detection in natural images. In particular, no improvement was found when using anchor positions associated to objects which were mined through k-means clustering. On the other hand, we found it useful to employ different weights \( \alpha \) for the two different uses of equation 1. A higher value for \( \alpha \) was used during matching (solving for \( X \)) than for backpropagation (learning of network parameters). This favours the use of all outputs during training — details are given in section V.

III. LOCALIZATION AND RECOGNITION

A. Line detection with left-side triplets

The first step detects the left-side of each lines through the network described in Section II. The model predicts
three position values, i.e. \( K = 3 \): 2 coordinates for the lower left corner plus the text height. Additionally, a prediction confidence score is output.

We also compare this method with two competing strategies:

i) point localization [18], \( K = 2 \), where only the \( x \) and \( y \) coordinates of lower left points are detected,

ii) and full box localization [16], where \( K = 4 \) and the \( x \) and \( y \) coordinates of the bottom-left corners of the text line bounding boxes are predicted with the width and the height of the text lines.

We also found that expanding the text box by a 10 pixel margin improves the full-page text recognition rates.

### B. End-of-line detection integrated with recognition

Detecting only the left side of the text lines and extending it toward the right part of the image as illustrated in Figure 2 b) means that for documents with complex layouts, some text from other text lines can be present in the image to be recognized.

For this reason, a 2D-LSTM based recognizer similar to the one described in [23] is trained with the Connectionist Temporal Classification [9] (CTC) alignment procedure to recognize the text present in, and only in, the designed text line. We found that the results were slightly improved by adding an End-of-line (EOL) label at the end of the text labels. This means that the network will learn, through the CTC training, to align the text labels with the frames corresponding to these image characters, as usual. But it will also learn to predict when the line is over, mark it with the EOL label, and learn not to predict anything else on the right side of the image. The context conveyed by the LSTM recurrent layers is responsible for this learning ability.

Two different recognition networks are trained, respectively for French and English. They are trained to recognize both printed and handwritten simultaneously.

### IV. EXPERIMENTAL SETUP

#### A. Datasets

We evaluate our method on the Maurdor dataset [4], which is composed of 8773 heterogeneous documents in French, English or Arabic, both printed and handwritten (Train: 6592, Validation: 1110, Evaluation: 1071). Because the annotation is given at paragraph level, we used the technique described in [3] to check the quality of line candidates with a constrained text recognition in order to obtain annotation at line level. All these lines are used to train the text recognizers described in section III-B and, on the test set, for the recognition experiments in Table IV.

For training the text line detection systems, only the 5308 pages for which we are confident enough that all the lines are detected are kept (Train: 3995, Validation: 697, Evaluation: 616). This subset is also used for the precision experiments shown in Tables II and III.

Finally, for the end-to-end evaluation results shown in Table V, we kept all the original documents of the test set in only one language, in order to avoid the language identification problem. We obtain 507 pages in French and 265 pages in English.

#### B. Metrics

Three metrics were used for evaluation:

1) **F-Measure** metrics is used in Tables II and III in order to measure precision of the detected objects being in the neighbourhood of reference objects. A detected object \( l \) is considered as correct if it is the closest hypothesis from the reference object \( t \) and if \( ||l - t|| < T \) for all \( k \in [0, K] \) where \( K \) is the number of coordinates per object set to 2 for Table II (points) and to 3 in Table III (triplets) and \( T \) is the size of the acceptance zone given as a proportion of the page width.

2) **Word Error Rate** (WER) metrics is the word level Levenshtein distance [11] between recognized and reference sequences of text.

3) **Bag of Word** (BOW) metrics is given at page level as a F-Measure of words recognized or not in the page. As explained in [24], it is a proper metric to compute recognition rate at page level because it does not need any alignment or ordering of the text lines that can be ambiguous for unconstrained documents.

#### C. Hyper-parameters

We trained with the RmsProp optimizer [28] with an initial learning rate of \( 10^{-3} \) and dropout after each convolutional layer. The \( \alpha \) parameter is set to \( \alpha = 1000 \) for matching (solving for \( X \)) and to \( \alpha = 100 \) for gradient computation.
TABLE II: Comparison of the F-Measure scores for the detection of bottom-left points with respect to the acceptance zone size for networks trained to detect points, triplets or boxes. Acceptance zones given as proportion of page width.

<table>
<thead>
<tr>
<th>Acceptance zone size</th>
<th>0.003</th>
<th>0.01</th>
<th>0.03</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box network (16, K=4)</td>
<td>6.8%</td>
<td>45.0%</td>
<td>82.8%</td>
<td>89.9%</td>
</tr>
<tr>
<td>Point network (18, K=2)</td>
<td>10.7%</td>
<td>57.4%</td>
<td>85.7%</td>
<td>91.7%</td>
</tr>
<tr>
<td>Triplet network (Ours, K=3)</td>
<td>11.2%</td>
<td>58.4%</td>
<td>87.0%</td>
<td>92.6%</td>
</tr>
</tbody>
</table>

TABLE III: Comparison of the F-Measure scores for the detection of left-side triplets with respect to the acceptance zone size for networks trained to detect triplets or boxes. Acceptance zones given as proportion of page width.

<table>
<thead>
<tr>
<th>Acceptance zone size</th>
<th>0.003</th>
<th>0.01</th>
<th>0.03</th>
<th>0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box network (16, K=4)</td>
<td>3.4%</td>
<td>24.6%</td>
<td>71.4%</td>
<td>89.7%</td>
</tr>
<tr>
<td>Triplet network (Ours, K=3)</td>
<td>4.2%</td>
<td>47.2%</td>
<td>84.8%</td>
<td>92.4%</td>
</tr>
</tbody>
</table>

V. EXPERIMENTAL RESULTS

A. Precision of the object localizations

Similarly to what is described in [18], we observed some instability in the position of the predicted objects when trying to detect boxes. Our intuition is that precisely detecting objects which ends outside of the convolutional receptive field of the outputs is difficult.

Characters may have a size of 1 or 2 mm in standard printed pages, corresponding to 0.005 and 0.01 as a proportion of the page width. Interlines may have similar sizes. Therefore, it is important that the position prediction is close enough in order not to harm the text recognition process.

The method described in [18] was dealing with this problem by detecting separately the bottom-left and top-right points and posteriorly pairing them. We observed that the precision was not harmed by the detection of triplets of coordinates (left, top, bottom).

In Table II, we show the F-measure of the detection of left-bottom points for several acceptance zone sizes. The results emphasize that detecting full text boxes reduces precision. Meanwhile, the precision of bottom-left point prediction is equivalent when the network is trained to detect triplets and not points.

Table III shows the same experiment with a 3D acceptance zone defined on the triplet positions, showing the same improved results for the triplet detection for small acceptance zones.

B. Detection of the line end with the text recognizer

In Table IV we compared two text line recognizers trained respectively on the reference text line images and on text line images defined only by the left sides coordinates of the text line and extended toward the right end of the page. These two recognizers are evaluated in both cases with the WER metric.

While the network trained on reference boxes is obviously not working well on extended test images, we see that the network trained on extended lines works on both tasks nearly as well as the network trained on reference boxes. This confirms that we can rely on the text recognizer to ignore the part of the line that does not belong to the text line.

C. Full page text recognition

Finally, we compared our method with baselines and concurrent approaches for full page recognition. The evaluation was carried out using the BOW metric and is shown on Table V. We show that the proposed methods yield good results on both the French and English subsets, consistently outperforming the document analysis baselines based on image processing, the object localisation baseline and the concurrent box detection and paired point detection systems. At decoding time, for an image of size 598x838, the object detection part takes a mean time of 245 ms per page, which is faster than the text recognition part, done in higher resolution, that takes 638 ms per page on average. Performances are CPU only on Intel Xeon E5-2640-v4 with 64 MB of RAM.

Some illustrations of the left-side triplets detection alongside with the final full-page text recognition are given in Figure 3 and emphasize the ability of the system to give good results on various types of documents.

VI. CONCLUSION

We described a full page recognition system for heterogeneous unconstrained documents that is able to detect and recognize text in different languages. The use of a neural network localisation process helps to be robust to the intra-dataset variations. In order to simplify the process and to gain both in precision and in preciseness, we focus on predicting the starting point (left) of the text line bounding boxes and leave the prediction of the end point (right) to a 2D-LSTM based text recognizer. We report excellent results on the Maurdor dataset and show that our method outperform both image-based and concurrent learning-based methods.
Invitation pour mon
la Famille Rochefleur :
Nous avons le plaisir de vous inviter à l’ouverture
exceptionnelle du salon du mariage qui se tiendra à
le verte-end du 6 et 7 Octobre 2007 de 20h à 19h.
Vous aurez ainsi le plaisir d’assister à un
défilé de robes de mariées, toutes plus oubliées les unes,
que les autres ; de rencontrer des animateurs de
sinées, des décorateurs, des joailliers, des traiteurs et :
bien plus encore !
Grâce à cette invitation vous pourrez bénéficier de :
de prénations exceptionnelles et vous pourrez
egalement profiter d’un délicieux buffet, mis à disposi-
tion!
dans le salt bout d’égaler vos papilles!
Une Tombola sera également organisée vous donnant.
ainsi l’opportunité de gagner de nombreux cadeaux.
En plus d’un séjour pour deux personnes, tout frais
passées à l’Ile haunité.
En espérant votre présence au salon, je vous
demanderai de remplir le talon réponse ci-joint dans
les meilleurs délais !
Cordialement,
l’organisatrice de
LEU*****
PREVOST Myriam
*****

• Invitation pour mon
• la Famille Rochefleur :
• Nous avons le plaisir de vous inviter à l’ouverture
• exceptionnelle du salon du mariage qui se tiendra à
• le verte-end du 6 et 7 Octobre 2007 de 20h à 19h.
• Vous aurez ainsi le plaisir d’assister à un
défilé de robes de mariées, toutes plus oubliées les unes,
• que les autres ; de rencontrer des animateurs de
• sinées, des décorateurs, des joailliers, des traiteurs et :
• bien plus encore !
• Grâce à cette invitation vous pourrez bénéficier de :
• de prénations exceptionnelles et vous pourrez
egalement profiter d’un délicieux buffet, mis à disposi-
tion!
• dans le salt bout d’égaler vos papilles!
• Une Tombola sera également organisée vous donnant.
• ainsi l’opportunité de gagner de nombreux cadeaux.
• En plus d’un séjour pour deux personnes, tout frais
• passées à l’Ile haunité.
• En espérant votre présence au salon, je vous
demanderai de remplir le talon réponse ci-joint dans
• les meilleurs délais !
• Cordialement,
• l’organisatrice de
• LEU*****
• PREVOST Myriam
• *****
REFERENCES


