Learning text-line localization with shared and local regression neural networks

Bastien Moysset*‡, Jérôme Louradour*, Christopher Kermorvant§ and Christian Wolf†‡
*A2iA SAS,Paris, France
†Université de Lyon, CNRS, France
‡INSA-Lyon, LIRIS, UMR5205,F-69621
§Teklia SAS, Paris, France
Email : bm@a2ia.com

***Abstract*—Text line detection and localisation is a crucial step for full page document analysis, but still suffers from heterogeneity of real life documents. In this paper, we present a novel approach for text line localisation based on Convolutional Neural Networks and Multidimensional Long Short-Term Memory cells as a regressor in order to predict the coordinates of the text line bounding boxes directly from the pixel values. Targeting typically large images in document image analysis, we propose a new model using weight sharing over local blocks. We compare two strategies: directly predicting the four coordinates or predicting lower-left and upper-right points separately followed by matching. We evaluate our work on the highly unconstrained Maurdor dataset and show that our method outperforms both other machine learning and image processing methods.**

***Keywords*-text-line segmentation; neural network; deep learning; LSTM; regression**

I. Introduction

Locating text lines is an important step in document analysis, which often precedes recognition. It is far from trivial in complicated settings, in particular in the case of handwritten documents and highly unstructured documents. Most of the existing work addresses this problem with low-level image processing techniques based on hand-crafted features. These methods are highly dependent on a large amount of parameters whose manual tuning can be very time-consuming. Moreover, they are also known to suffer from a lack of robustness limiting their use to a corpus with small variations.

In this work, we propose a new method for text-line location entirely based on machine learning. A deep neural network directly performs regression of line coordinates from high-resolution document images taken as raw input. Localization with neural networks is currently widely studied in the context of natural images with networks being trained on the large ImageNet and COCO datasets (see section II). We will show that existing methods are badly suited for applications in document image analysis due to two particularities in this domain: (i) document images are of very high resolution, which translates into networks with an enormous amount of parameters if classical architectures are chosen; (ii) documents contain a large amount of small objects, whereas natural images mostly contain a small amount of larger objects.

We address these challenges by introducing a method which directly regresses text-line bounding boxes through a neural network consisting of multiple local prediction models with shared parameters. This reduces the total amount of parameters, making learning from small datasets of large images possible. The bounding box detection problem is decoupled into two different sub-problems, namely the detection of lower-left and upper-right corner points, followed by matching of point pairs which is solved by minimizing a global energy function. We show that our strategy of local processing and matching, which allows to use networks with small spatial support and few parameters, is far superior to the solution of global regression using a full global network.

The paper is organized as follows: section II reviews existing literature on text-line localization and on object detection using machine learning. Section III explains the proposed method as well as its training and inference. Section IV describes the experimental setup and section V study the results. Finally, section VI concludes.

II. Related work

A. Line segmentation

The current techniques for offline text recognition require the text line positions as input [11], motivating work in detection and segmentation of text lines for end-to-end document processing systems. Many methods have been proposed to tackle this task [8] and evaluated through competitions [25] [15].

Most of the current state of the art methods are based on image processing techniques. Among the most competitive ones, Brodic et al. [2] use waterflow to find the interline spaces. Nicolaou et al. [16] find a path between text lines by following the high color levels of a blurred image. Similarly, Saabni et al. [22] use seam carving to find the low color levels and follow the text lines with different constraints.

Shi et al. [24] use a smearing related techniques where the image is convolved with ellipsoid filters to blur the image. The obtained image is then binarised and the connected components of the results are associated to a text line.

Other algorithms start from the connected components of a binarised document image. Ryu et al. [21] use under segmentation of connected components and group them by minimizing an energy function where the errors are the distances between the center of the elements and low order polynomials associated to lines. Precedently, Louloudis et al. [9] used the Hough transform to find alignments between the centers of the connected components.

These techniques rely on a clear difference between dark text and white background, or vice-versa, which is highlighted by the fact that most of them proceed by binarisation. Other elements of the documents like logos, tables, backgrounds, pictures but also noises and scratches can disturb the process. Text in inverse video is also to be mentioned. Heavy pre-processing and engineering is needed to get rid of the problems that occur on a given dataset and it is hard to perform well on highly heterogeneous databases like the Maurdor database [3].

Machine Learning techniques are a promising option to learn to generalize well to various kinds of documents but, if they have been extensively used for the text recognition step, few have used it for text segmentation.

Moysset et al. [12] use an LSTM recurrent neural network with the CTC framework to segment paragraphs, but the segmentation is only mono dimensional and therefore cannot be used on full pages. In the scene text community, Delakis et al. [4] extended by more recent works [27] have used convolutional neural networks to classify each position of an image as text or non-text with sliding windows. These techniques encounter problems with overlapping objects and handwritten text lines often overlap.

B. Object detection

A large amount of recent work has been performed on the detection of objects with Machine Learning and especially with deep learning [20]. Sermanet et al. [23] use sliding windows of different scales and a CNN classifier to predict the presence of an object. Girshick et al. [6] also use a CNN classifier but on object candidates obtained with selective search. These techniques highly depend on the quality of the object candidates and suffer from the fact that a part of a text line can look like a full text line.

Sun et al. [26] use multi-scale fully-convolutional networks to obtain a feature map for all classes. This technique will not work for text detection due to the overlapping between our lines which are objects of the same class.

For similar reasons, Redmon et al. [19], which predict box candidates using regression and a feature map of objects present at a given place, admit that their technique "struggles with small objects that appear in groups".

Pinheiro et al. [17] use a two-branch CNN to predict both a mask of the object in the center of the image and a confidence score.

Erhan et al. [5] use a CNN as a regressor to directly predict a given number of box coordinates and confidence that the box exists. It enables to detect a varying number of overlapping objects of the same class, the size of the objects being unconstrained. But, when it is needed to detect a large number of objects, the number of predictions have to be increased and, when dealing with large images, the last feature map is larger. The number of parameters is high and a huge amount of data is needed for the training.

III. Proposed method

A. Principle

The objective is to learn a parametrized non-linear mapping f from input image I to a set $\mathcal{B}$ of bounding box locations:

$$\mathcal{B} = f(I, \theta) \tag{1}$$

where θ are parameters and $\mathcal{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_N]$ is a set of detected bounding boxes, and each bounding box $\mathbf{b}_n$ is composed of 2 coordinates x_n and y_n, the width w_n and the height h_n of the box and a detection confidence ρ_n: $\mathbf{b}_n$=$\{x_n, y_n, w_n, h_n, \rho_n\}$, $n = 1 \ldots N$. The confidence value is predicted alongside the position of each object and is aimed to enable the system to detect a varying number of objects. It means that we will just have to ensure that the number N of network outputs is higher or equal to the maximum number of objects (lines) that can be encountered in an image.

The mapping is implemented as a convolutional deep neural network with weights θ. In the targeted scenario in document image analysis, the size of the input image I is very large whereas the sizes of the bounding boxes are typically small, and their number is high. This motivates the introduction of a local model with shared parameters. In the lines of [10], we therefore included a *Space Displacement Localization* (SDL) layer at the end of our network, where bounding boxes are predicted by layers with limited spatial support. The aim of this layer is to share the parameters between several locations in the image and, consequently, to ease the training of the system. To include context into this local network, while at the same time keeping the number of parameters low, we added layers of two dimensional LSTM neurons. The exact architecture will be given in the following sub-sections.

Whereas the local network proposed in [10] predicted origin points of text lines, our objective is to detect full text lines, which are bounding boxes. This task is harder for a localized model, as part of the bounding box may lay outside of its limited spatial support. The LSTM context layer decreases the problem, but cannot fully solve it. We therefore propose to separate the prediction of each bounding box into two different mappings $f^{\ell\ell}$ and f^{ur} as follows:

$$\begin{aligned} \mathcal{B}^{\ell\ell} &= f^{\ell\ell}(I, \theta^{\ell\ell}) \\ \mathcal{B}^{ur} &= f^{ur}(I, \theta^{ur}) \end{aligned} \tag{2}$$

Table I
NETWORK ARCHITECTURE, LAYERS IN ITALIC ARE OPTIONAL.

	Layer	Filter size	Stride	Size of the feature maps	Number of parameters
	Input image			1×(598×838)	
1.	Convolution	(4×4)	(3×3)	12×(199×279)	204
	MD-LSTM			——— " ———	*8880*
2.	Convolution	(4×3)	(3×2)	16×(66×139)	2320
	MD-LSTM			——— " ———	*15680*
3.	Convolution	(6×3)	(4×2)	24×(16×69)	6936
	MD-LSTM			——— " ———	*35040*
4.	Convolution	(4×3)	(3×2)	30×(5×34)	8670
	MD-LSTM			——— " ———	*54600*
5.	Convolution	(3×2)	(2×1)	36×(2×33)	6516
	MD-LSTM			——— " ———	*78480*
6.	SDL points			3×20×(2×33)	2160
	/ boxes			/ 5×20×(2×33)	/ 3600

Here, $\mathcal{B}^{\ell\ell} = \{\mathbf{b}_1^{\ell\ell}, \mathbf{b}_2^{\ell\ell}, \dots \mathbf{b}_N^{\ell\ell}\}$ is the set of lower left points $\mathbf{b}_n^{\ell\ell} = [x_n^{\ell\ell}, y_n^{\ell\ell}, \rho_n^{\ell\ell}]$ and the set of upper right points $\mathcal{B}^{ur}$ is defined accordingly.

Next sub-section III-B describes the network architectures. Subsection III-C describes the follow up step which matches left and right points into bounding boxes.

B. The localization model

The networks $f^{\ell\ell}(\cdot, \theta^{\ell\ell})$ and $f^{ur}(\cdot, \theta^{ur})$ share the same architecture, but not parameters. They are directly fed with the input image gray-level pixels after normalisation and color conversion. Images are previously rescaled to the 70 dpi resolution or to a maximum size of 598x838. Dealing directly with the color pixels has been tried but did not improve the results. As described in table I, the network is made of five convolutional layers with hyperbolic tangent as non linearity. The receptive fields corresponding to each of these layers are illustrated in figure 1. Dropout is used on all these convolutional layers in order to reduce overfitting. LSTM recurrent layers may be added after each convolutional layer (several configurations have been tested, see Section V-A) The last layer is our SDL layer that predicts the coordinates of the text lines.

The output of the network is a fixed number N (in table I, N=20) of k-uplets (k being equal to 3 for points and to 5 for boxes). Each of these k-uplet corresponds to an object.

Unlike what is done in Erhan et al. [5], our system predicts objects at local level. The SDL layer share parameters on a map of feature with width W and height H (W=2 and H=33 in table I). The outputs of our network are the k-uplets computed by the SDL layer as follows:

$$\rho_{i,j,n} = \sigma\left(\mathbf{r}_n{}^{\intercal}\mathbf{h}_{i,j}(I) + b_n\right) \quad (3)$$

$$x_{i,j,n} = \sigma\left(\mathbf{s}_n{}^{\intercal}\mathbf{h}_{i,j}(I) + c_n\right) \times \Delta_{\mathrm{x}} + (i-1) \times \delta_{\mathrm{x}} \quad (4)$$

$$y_{i,j,n} = \sigma\left(\mathbf{t}_n{}^{\intercal}\mathbf{h}_{i,j}(I) + d_n\right) \times \Delta_{\mathrm{y}} + (j-1) \times \delta_{\mathrm{y}} \quad (5)$$

where $i = 1, \dots, W$ and j=$1, \dots, H$ are 2D indices of input features of the SDL layer. The $\mathbf{h}_{i,j}(I)$ values correspond to the outputs of the last layer of the neural network before the SDL layer. $\mathbf{r}_n$, $\mathbf{s}_n$ and $\mathbf{t}_n$ (resp. b_n, c_n and d_n) are vectors of weights (resp. values of bias) of our SDL layer. Δ_{x} and Δ_{y} are the size of the receptive fields corresponding to the $\mathbf{h}_{i,j}(I)$ in the input image. δ_{x} and δ_{y} correspond to the stride between consecutive receptive fields. Finally, σ is the sigmoid function.

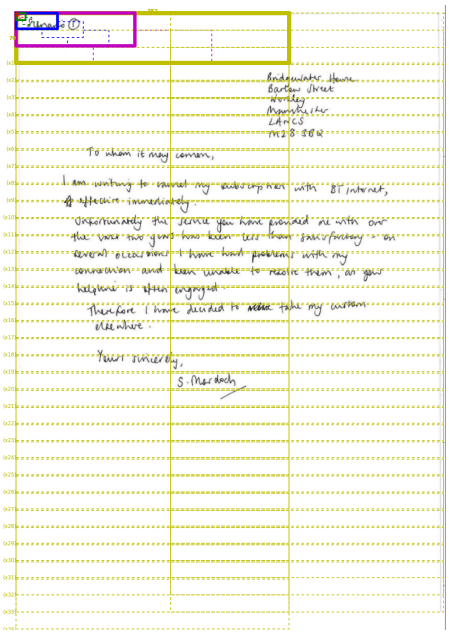

Figure 1. Illustration of the size of the receptive fields and strides for each layer of the network. The correspondence with the layers of table I are 1:red, 2:green, 3:blue, 4:magenta, 5:yellow (best viewed in color).

For straightforward bounding box detection, the predictions of the widths and heights of the boxes are added. We want to be able to predict boxes larger than a receptive field. For this reason, the values are normalised with the page size and not with the size of the local receptive field:

$$w_{i,j,k} = \sigma\left(\mathbf{u}_n{}^{\intercal}\mathbf{h}_{i,j}(I) + e_n\right) \quad (6)$$

$$h_{i,j,k} = \sigma\left(\mathbf{v}_n{}^{\intercal}\mathbf{h}_{i,j}(I) + f_n\right) \quad (7)$$

where $\mathbf{u}_n$ and $\mathbf{v}_n$ (resp. e_n and f_n) are vectors of weights (resp. values of bias).

Note that to be able to predict the same number of objects than in Erhan et al., our system needs a number N of k-tuplets as output of our last neural network layer, which is $H \times W$ smaller. This change allows several outputs to share weights and reduces the number of parameters in the last layer. It reduces overfitting, ease training and give better results for smaller datasets. The drawback is that the prediction of the position of an object has to be done without information from the whole image, without the context. Adding recurrences, namely Multi Dimensional LSTM layers enables to counterbalance this and get back some context knowledge.

C. Pairing points

The sets of predicted points $\mathcal{B}^{\ell\ell}$ and $\mathcal{B}^{ur}$ need to be matched into a set of bounding boxes. We tackle this by solving the following global energy function over variables $z=\{z_{ij}\}$, where $z_{ij} \in \{0,1\}$ and $z_{ij}=1$ is interpreted as point $\mathcal{B}_i^{\ell\ell}$ being paired with point $\mathcal{B}_j^{ur}$:

$$\begin{aligned} \hat{z} = \arg\min_z \sum_i \sum_j z_{ij} D(\mathcal{B}_i^{\ell\ell}, \mathcal{B}_j^{ur}) \\ \text{s.t.} \quad \forall j \textstyle\sum_i z_{ij} \in \{0,1\}, \quad \forall i \textstyle\sum_j z_{ij} \in \{0,1\} \end{aligned} \tag{8}$$

The conditions ensure that a left point is not paired with several right points and vice versa. The distance function $D(.,.)$ indicates whether a given left point is compatible with a given right point, i.e. whether it is probable that this pairing corresponds to a text line. In practice, we learn this distance function from training data using a convolutional neural network. The minimization in equation (8) is carried out efficiently using the Hungarian algorithm [14].

D. Training

Training consists in tuning the parameters of the network that outputs the corners of the bounding boxes ($f^{\ell\ell}(\cdot,\theta^{\ell\ell})$ and $f^{ur}(\cdot,\theta^{ur})$), as well as estimating the distance function D used to match points (sectiion refeq:pairing). The former requires matching hypothesis objects (network outputs) and reference objects from the ground truth (not to be confused with the pairing described in section III-C). Similar to [5], this is done by minimizing a cost composed of a localization and a confidence term as shown in equation (9). The α parameter balances the two components.

$$E_{ij}(X,\theta) \quad = \quad \alpha||o_i(\theta) - g_j||^2 + \log\left(\tfrac{c_i(\theta)}{1-c_i(\theta)}\right) \tag{9}$$

The Hungarian algorithm [14] is used to minimize the global cost while being sure that one and only one hypothesis object is matched to each reference object as shown in equation 10.

$$\begin{aligned} E(X,\theta) = \alpha \sum_{ij} X_{ij}||o_i(\theta) - g_j||^p + \sum_i X_{ij} \log\left(\frac{c_i(\theta)}{1-c_i(\theta)}\right) \\ \text{s.t. } \forall j \textstyle\sum_i X_{ij} \in \{0,1\} \quad \wedge \quad \forall i \textstyle\sum_j X_{ij} \in \{0,1\} \end{aligned} \tag{10}$$

The derivation of this global cost with respect to the network output gives us the gradient that we will back-propagate in our network for a standard SGD.

The pairing function $D(.,.)$ is a Convolutional Neural Network classifier trained on truth line positions.

IV. Experimental setup

A. Datasets

The Maurdor database [3] is made of unconstrained document images handwritten and/or printed in three languages (French, English and Arabic). It is constituted of 6592 training pages, 1110 validation pages and 1071 test pages.

For the training and the validation of our systems, the bounding boxes of the text lines have been obtained in a semi-automatic way by using as information the content of the text in the pages. Several line segmentation algorithms are run on the paragraph images. Then a constrained recognition [1] is performed. We kept only the pages in which all the lines from all the paragraphs were retrieved in order to avoid false negatives in the training samples. 3995 pages (out of 6592) have been kept for the train set, and 697 (out of 1110) for the validation set.

B. Metrics

Two metrics have been used in this paper. The first one is for the point detection task and is used on the validation set to tune the hyperparameters of the point detection network. We consider that a truth point is well detected if there is an hypothesis point in its neighbourhood, defined in equation (11). In particular, a hypothesis point (x_h, y_h) is accepted if it is in a square of size L centred on a reference point (x_r, y_r).

$$|x_r - x_h| < L/2 \text{ and } |y_r - y_h| < L/2 \tag{11}$$

The L parameter is mentioned as a percentage of the page width. The higher it is, the more we will want to focus on the ability of the system to detect all the lines. On the contrary, taking a low value of L will enable to know the preciseness of the position prediction. F-Measure is then computed.

The second metric, is used to evaluate the quality of the box predictions on the test set. It is the Bag Of Words metric (BOW). As stated in [18], it enables to check that the words are readable and, therefore, to go closer from a real end-to-end task but it also avoids relying on a reading order between the detected lines. The drawback is that it does not penalize merges of lines from different columns. The BOW error metric s the sum of the insertion and deletion rates.

For evaluation with BOW, we used a MDLSTM-based text recognizer [13] trained on both printed and handwritten text lines. We used only the documents fully in English or fully in French to avoid language selection biases.

V. Results

A. System design

All the hyper-parameters have been optimized on the validation set and for each network/strategy separately. For the proposed method, this led to the values in table I. For Erhan et al., the original architecture from Krizhevski et al. [7] has also been tried.

As shown in table II, adding context layers in the form of Multi-Dimensional LSTMs after the convolutional layers improves performance. It seems that the LSTMs are more important on the first layers of the system where there are longer dependencies. However, the drawback of these layers is noticeably slower convergence during training.

Table II
COMPARISON OF RESULTS ON POINT DETECTION WITH RESPECT TO THE NUMBER AND THE PLACE OF LSTM LAYERS

Position of the LSTMs	F-Measure $L = 0.01$	F-Measure $L = 0.03$	F-Measure $L = 0.05$
1	0.369	0.749	0.797
1-2	0.483	0.818	0.851
1-2-3	0.486	0.827	0.856
1-2-3-4	**0.612**	**0.882**	**0.908**
1-2-3-4-5	0.569	0.867	0.902
2-3-4-5	0.567	0.860	0.897
3-4-5	0.478	0.820	0.857
4-5	0.410	0.783	0.833
5	0.335	0.684	0.733

Table III
COMPARISON WITH ERHAN ET AL. ON THE POINT DETECTION TASK ON MAURDOR VALIDATION SET

Method	F-Measure $L = 0.01$	F-Measure $L = 0.03$	F-Measure $L = 0.05$
SDL Point	0.612	0.882	0.908
Erhan et al. Point [5]	0.048	0.224	0.367
Erhan Point without K-Means	0.063	0.316	0.529

Table IV
COMPARISON OF BOW RESULTS ON THE FULL ENGLISH OR FRENCH MAURDOR TEST SET FOR THE BOX DETECTION TASK.

Method	French (507 pages)	English (265 pages)
Shi et al. [24]	80.29%	89.09%
Nicolaou et al. [16]	70.93%	82.8%
Erhan et al. Box [5]	111.59%	95.63%
Erhan et al. Box (tuned)	108.96%	94.22%
SDL Box	84.83%	77.22%
SDL Point + Matching	**57.73%**	**61.29%**

Dropout is added after all the convolutional layers and increases the performance by preventing co-adaptation between the neurons. This is especially required due to the low number of training samples we have.

The α parameter which balances the confidence and the position scores has also been tuned. We empirically found that a higher value of α taken during the matching step than during the gradients propagation improves the results. This may be because it helps all the outputs to be used.

B. Comparison of regression strategies

The bottom lines of table IV compare the different strategies in the detection of bounding boxes through regression: direct regression of bounding boxes and prediction of left and right points followed by pairing. It can be seen that the pairing strategies outperforms direct regression convincingly (∼+27 points on French pages). This is also illustrated in figures 2 and 3, respectively. Direct box regression fails at some large text lines, which is most probably due to the fact that the local support of a given block does not see the full text lines. If the structure of these large lines is regular, the context layer seems to help. In irregularly placed lines, however, context is not helpful enough.

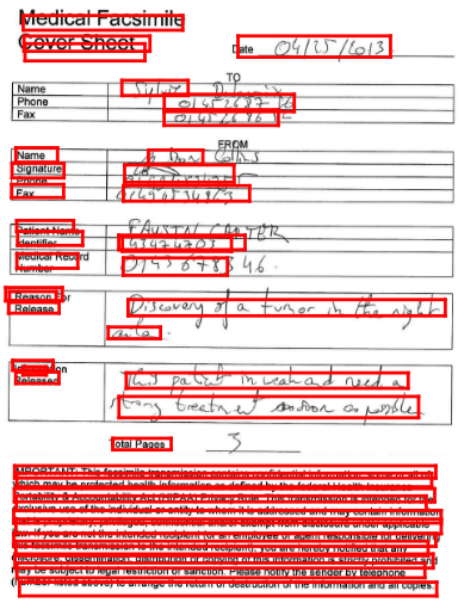

Figure 2. Detection results produced by an SDL network directly predicting boxes.

Since these issues are due to the local nature of the algorithm, we also compare the proposed method with global regression of bounding boxes (Erhan et al. [5]). Global regression does not perform well in this context, which can be explained by the very large images in document analysis resulting in a high number of parameters in the last layer of the network and by the fact that the Maurdor dataset is far smaller than the typical Imagenet for which these methods have been designed [20].

C. Comparison to the state-of-the-art

Table IV also gives a comparison with two other methods at the state of the art [24][16], which are based on traditional image processing. We can see that machine learning again convincingly outperforms handcrafted image processing techniques.

VI. CONCLUSIONS

In this paper, we present a novel technique for text line detection in highly unconstrained documents, based on the direct prediction of line coordinates with a deep recurrent neural network. We introduce a model sharing parameters between local blocks. We also propose the creation of boxes from these coordinate predictions and introduce a neural network classifier to pair left and right coordinate points. We showed that the proposed method significantly outperforms baselines and competition on the Maurdor dataset.

Future work will be to study the interaction between the hyper-parameters of our neural network, to improve the pairing of right and left points and to check to what extent

Figure 3. Detection results produced by an SDL network predicting left and right points (respectively shown in green and red) followed by pairing.

the system is able to generalize to other datasets, with or without adaptation.

References

[1] Bluche, T., Moysset, B., Kermorvant, C.: Automatic line segmentation and ground-truth alignment of handwritten documents. In: Int. Conf. on Frontiers in Handwriting Recognition (2014)

[2] Brodić, D., Milivojević, Z.N.: Text line segmentation with the parametric water flow algorithm. Information Technology And Control 45(1), 52–61 (2016)

[3] Brunessaux, S., Giroux, P., Grilheres, B., Manta, M., Bodin, M., Choukri, K., Galibert, O., Kahn, J.: The maurdor project - improving automatic processing of digital documents (2014)

[4] Delakis, M., Garcia, C.: Text detection with convolutional neural networks. In: Int. Conf. on Computer Vision Theory and Applications. pp. 290–294 (2008)

[5] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: Int. Conf. on Computer Vision and Pattern Recognition (2014)

[6] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conf. on computer vision and pattern recognition (2014)

[7] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[8] Likforman-Sulem, L., Zahour, A., Taconet, B.: Text line segmentation of historical documents: a survey. Int. Journal of Document Analysis and Recognition 9(2-4), 123–138 (2007)

[9] Louloudis, G., Gatos, B., Pratikakis, I., Halatsis, C.: Text line and word segmentation of handwritten documents. Pattern Recognition 42(12), 3169–3183 (2009)

[10] Moysset, B., Adam, P., Wolf, C., Louradour, J.: Space displacement localization neural networks to locate origin points of handwritten text lines in historical documents. In: Workshop on Historical Document Imaging and Processing (2015)

[11] Moysset, B., Bluche, T., Knibbe, M., Benzeghiba, M.F., Messina, R., Louradour, J., Kermorvant, C.: The A2iA Multilingual Text Recognition System at the Maurdor Evaluation. In: Int. Conf. on Frontiers in Handwriting Recognition (2014)

[12] Moysset, B., Kermorvant, C., Wolf, C., Louradour, J.: Paragraph text segmentation into lines with recurrent neural networks. In: Int. Conf. on Document Analysis and Recognition (2015)

[13] Moysset, B., Messina, R., Kermorvant, C.: A comparison of recognition strategies for printed/handwritten composite documents. In: Int. Conf. on Frontiers in Handwriting Recognition. pp. 158–163 (2014)

[14] Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)

[15] Murdock, M., Reid, S., Hamilton, B., Reese, J.: Icdar 2015 competition on text line detection in historical documents. In: Int. Conf. on Document Analysis and Recognition (2015)

[16] Nicolaou, A., Gatos, B.: Handwritten Text Line Segmentation by Shredding Text into its Lines. In: Int. Conf. on Document Analysis and Recognition (2009)

[17] Pinheiro, P.O., Collobert, R., Dollar, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems. pp. 1981–1989 (2015)

[18] Pletschacher, S., Clausner, C., Antonacopoulos, A.: Europeana newspapers ocr workflow evaluation. In: Workshop on Historical Document Imaging and Processing (2015)

[19] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640 (2015)

[20] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. Int. Journal of Computer Vision (2015)

[21] Ryu, J., Koo, H.I., Cho, N.I.: Language-independent text-line extraction algorithm for handwritten documents. Signal Processing Letters 21(9), 1115–1119 (2014)

[22] Saabni, R., Jihad, E.S.: Language-Independent Text Lines Extraction Using Seam Carving. In: Int. Conf. on Document Analysis and Recognition (2011)

[23] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013)

[24] Shi, Z., Setlur, S., Govindaraju, V.: A Steerable Directional Local Profile Technique for Extraction of Handwritten Arabic Text Lines. In: Int. Conf. on Document Analysis and Recognition (2009)

[25] Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: Icdar 2013 handwriting segmentation contest. In: Int. Conf. on Document Analysis and Recognition (2013)

[26] Sun, C., Paluri, M., Collobert, R., Nevatia, R., Bourdev, L.: Pronet: Learning to propose object-specific boxes for cascaded neural networks. arXiv preprint arXiv:1511.03776 (2015)

[27] Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: Int. Conf on Pattern Recognition (2012)