Text Line Segmentation in Historical Newspapers

Ladislav Lenc[1,2], Jiří Martínek[1,2], and Pavel Král[1,2]

[1] Dept. of Computer Science & Engineering
Faculty of Applied Sciences
University of West Bohemia
Plzeň, Czech Republic
[2] NTIS - New Technologies for the Information Society
Faculty of Applied Sciences
University of West Bohemia
Plzeň, Czech Republic
{llenc,jimar,pkral}@kiv.zcu.cz

Abstract. This paper deals with page segmentation into individual text lines used as an input of a line-based OCR system. This task is usually solved in one step which directly identifies text lines in whole documents. However, a direct approach may jeopardize the reading order of the lines and thus deteriorate the overall transcription result.
We propose a novel approach which decomposes this problem into two steps: text-block and text-line segmentation. The particular tasks are handled by algorithms based on fully convolutional neural networks.
The proposed method is evaluated on two standard corpora, Europeana and RDCL 2019, and on a novel dataset created from data available in Porta fontium portal. This dataset is freely available for research purposes.

Keywords: Document image segmentation · Layout analysis · Fully Convolutional network · FCN.

1 Introduction

Preservation of historical documents stored in various archives is very important. Many efforts have been invested into digitisation of such archival documents. Nevertheless, the digitisation is just the first step in the process of making the documents accessible and exploitable.

The goal of the subsequent processing is to convert the documents into a text form and allow efficient indexing, searching, or even more sophisticated tasks from the natural language processing (NLP) field, such as classification or summarisation.

The first step of document image processing, which determines the success of all the following tasks is the segmentation. Therefore, the main goal of this paper consists in proposing an efficient and accurate segmentation method that also preserves a reading order.

The final outcome of many page segmentation methods is a segmentation mask for text regions, eventually also for other document elements such as images or tables, and do not consider cropping the regions and further processing of them. We want to go further and create a complete method which converts the input image into an ordered set of text regions and text-line images suitable for processing by an optical character recognition (OCR) system. Our work is dedicated to the processing of archival documents from the Czech-Bavarian border area stored in Porta fontium portal[3]. Namely, we process newspapers from the end of the nineteenth century printed in Fraktur script.

Current OCR algorithms usually rely on neural networks that recognise whole text lines [24,2,23]. We thus aim at segmentation of pages into individual text lines. The task can be carried out directly on whole pages. However, in this case, it is difficult to determine the reading order of the extracted lines which is crucial for further processing of page transcription obtained by an OCR engine. It is obvious especially in the case of complex page layouts with more columns. To be able to find the reading order, it is necessary to determine the page structure and first create an ordered set of text regions which can be further split into text lines. Therefore, the presented approach combines the results of three partial algorithms based on fully convolutional neural networks (FCNs):

1. Text and non-text segmentation on a pixel level to differentiate foreground and background;
2. Separator detection and subsequent cropping of text regions and reading order estimation;
3. Baseline detection within the cropped text regions and extraction of text-line images.

The main strength of the proposed approach is that the particular tasks complement each other and can reduce final error rate of the whole method. One example is the erroneous merging of text regions by the text segmentation method. Such merging can be solved by detecting separators that divide the regions, and thus the final error is reduced. We must also mention that if we apply the baseline detection algorithm directly on the whole page it may lead to merged lines over a separator. It typically occurs in low quality scans with text regions lying very close to each other. In this case, the overall reading order of the page is incorrect. The presented method solves the above mentioned issues which is the main contribution of our work.

The approach is evaluated on a newly created dataset collected from Porta fontium. This dataset is freely available for research purposes[4] and represents another contribution of this work. We further evaluate the system on Europeana and RDCL 2019 corpora.

[3] http://www.portafontium.cz/
[4] http://ocr-corpus.kiv.zcu.cz/

2 Related Work

Most of nowadays methods treat the document segmentation as a pixel labelling problem and use deep learning for this task. An approach aiming at historical documents proposed by Chen et al. [3] is based on super-pixel calculation using simple linear iterative clustering [1]. The patches around super-pixels are classified by a convolutional neural network (CNN). This method outperformed a previously presented method [4] which solves document segmentation by convolutional auto-encoders.

Fully convolutional networks (FCNs) were developed for semantic image segmentation. The concept of FCNs is different from the above mentioned approaches. The training objective is to directly apply several convolutional layers followed by de-convolutions or up-sampling of the input image and obtain the final pixel-level segmentation as a result. A pioneering work was presented by Long et al. [15]. The authors took concepts of classification networks such as VGG net [25] or AlexNet [12] and incorporated it into the fully convolutional networks. The architecture has outperformed state-of-the-art results in the semantic segmentation task.

A well known example of an FCN is U-Net [21]. The method was first applied on bio-medical data segmentation. However, it can be also used for document image segmentation. Another FCN-based model in the biomedical domain was proposed by Novikov et al. [17]. It performs a multi-class segmentation of anatomical organs in chest radiographs (x-rays), namely for lungs, clavicles, and heart.

Sherrah [22] applies an FCN to carry out semantic segmentation in aerial images. He performed a fine-tuning of a pre-trained CNN to make better image features for high-resolution aerial images where it is crucial to find boundaries.

A modification of U-net architecture proposed by Wick and Puppe [26] was successfully applied to document image segmentation. The convolutional layers use kernel size of 5 and padding is used to keep image dimension. The network improved state-of-the-art results on several document segmentation datasets.

U-Net architecture enriched with spatial attention (A) and residual structures (R) was proposed by Grüning et al. in [11]. The network was named ARU-Net and it was designed mainly for the task of baseline detection. However, it is possible to use it for arbitrary pixel-labelling problem when appropriate training data are available. An adaptive version of U-Net was utilized for text-line segmentation in [16]. It is a modified and optimized version of U-Net. Smaller number of convolutional filters are used in this work and up-sampling operation in the decoder part is replaced by 2D de-convolution.

Another segmentation method based on an FCN was presented in [27]. First 5 convolutional layers are taken from the VGG network [25]. It is followed by three additional layers with kernel size 3×3. All de-convolution layers have kernels of size 2×2 and stride 2. The approach was tested on DIVA-HisDB and achieved pixel-level accuracy of 99%. The input images are not used directly. Instead, smaller crops with size 320×320 pixels are utilized.

A complex document segmentation and evaluation method was proposed by Li et al. [14]. The label pyramid network (LPN) utilizes an FCN as a core. The label map pyramid is transformed from region class label-map by distance transformation and multi-level thresholding. One single probability map is obtained by summing up the outputs of the LPN. The authors use intersection over union (IoU) and ZoneMap metric [9] for document region segmentation evaluation.

Zhong et al. [28] propose a large dataset for document layout analysis with several deep learning algorithms for baseline evaluation. The authors compare different pre-trained F-RCNN and M-RCNN models for fine tuning using the data from this dataset with interesting results. For evaluation they present mean average precision (MAP) and IoU of bounding boxes.

3 Document Image Segmentation

The presented method is composed of two main sub-tasks. The first one is to distinguish text from background and other non-text content such as images and illustrations and to extract the text blocks. We rely on FCN networks trained for text / background segmentation.

Text segmentation is complemented by separator segmentation within this sub-task. Generally, we differentiate black and white separators. We denote the black ones as explicit and the white ones as implicit separators. The white separators are actually the gaps between text regions. The goal of the separator segmentation step is to detect black separators within the processed page. Such separators determine the overall page structure and are used to define the reading order. It would be possible to omit the separator segmentation step and define the layout only according to white separators inferred from the text segmentation mask in the same matter as X-Y cuts and similar methods. However, our preliminary experiments have shown that the black separators can help to divide regions where text segmentation mask erroneously merges the regions. This happens mainly in cases where two text columns are very close to each other. We use another FCN for the separator segmentation task.

Combining the separator segmentation and the result of the text segmentation allows us to define the page layout and to divide the page into several text blocks. Additionally, we determine the reading order of the extracted regions which is very important for further processing of the page content.

The second sub-task is segmentation of text lines within the detected text blocks. For this task, we utilize an FCN model trained for baseline detection. The baseline positions together with connected component analysis are then directly used for line images extraction.

The architecture of the presented approach is shown in Figure 1. We describe the above-mentioned tasks in more detail in the followings sections.

3.1 Text-block Segmentation

Segmentation using an FCN can be seen as a mapping of an input image to several maps of probabilities with the same dimensions as the input image. Each

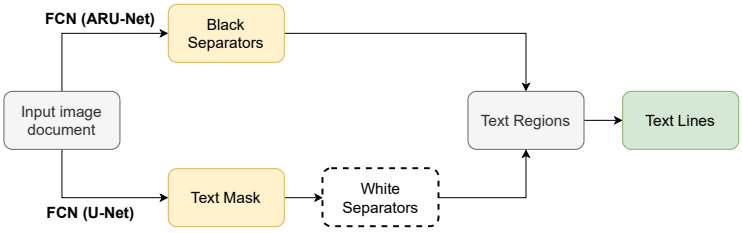

Fig. 1. Architecture of the presented approach

map indicates probabilities of pixel membership in a given class. In our case, only one map indicating probabilities that a pixel belongs to a text region is considered. The probability map contains values between 0 and 1 and we thus need to threshold it to obtain a segmentation mask. Threshold value is set to standard value of 0.5 in all cases.

Ground-truths (GT) for such a network are presented in a form of binary images where the text content is marked by 1 (white) and the background pixels by 0 (black). We do not distinguish between the background and the foreground pixels within a text block area. The ground-truth images are constructed according to the GT stored in the PAGE XML format [18].

The trained network is used to predict the probability map. The thresholded output then forms a segmentation mask. Using a connected components analysis we can separate masks for individual text regions. By multiplying the obtained region mask and the original image we obtain the segmented text region and use it for further processing. In our preliminary experiments, we have compared the performance of two state-of-the-art candidates from the FCN family, namely U-Net [21] and high performance FCN (HP-FCN) [26]. The experiments have shown that the performances of both networks are very similar. Due to the fact, that the HP-FCN has lower number of parameters and is faster, we decided to use this one in the system. It is basically an adaptation of U-Net designed for processing of historical documents. The main difference is that it does not use the skip connections.

3.2 Separator Segmentation

Separators split the page into smaller logical units (columns or smaller sections depending on the level of the separators). In all but a few cases, they can prevent merging of two or more text regions that can occur if we segment the page only according to a mask obtained in the text segmentation step.

A fragment of the visualisation of two considered regions: text regions (blue surrounding polygons) and separator regions (bold black lines) is depicted in Figure 2.

Within this task, we first apply an FCN trained for separator segmentation on the input image. Even though we can use any of the above-mentioned FCNs for the separator segmentation task, our preliminary experiments have shown

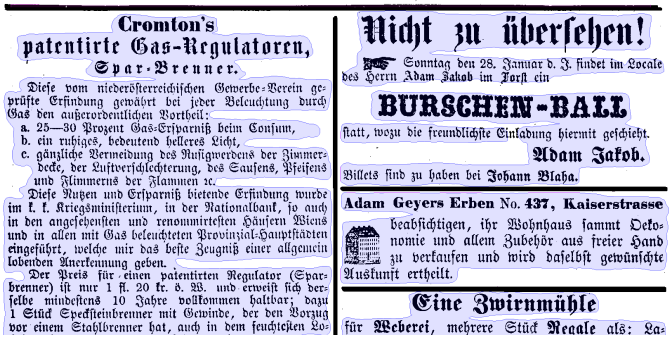

Fig. 2. Examples of detected text regions (blue surrounding polygons) and separator regions (black bold lines) in a document

that the most suitable candidate is ARU-Net [11] which extends U-Net with residual blocks and attention mechanism and is designed for detection of line objects.

The GT images for network training are constructed in the same way as the ones for text segmentation. The separator regions are marked according to the corresponding PAGE XML file. The result of the network prediction is a map that indicates the probability that a given pixel belongs to the separator class.

The second step is the post-processing of the map. We apply morphological opening in order to remove noise. It is followed by closing with a rectangular structural element which should remove small gaps in separator lines.

Then we detect the separator regions by the means of connected component analysis. Some separators may be still torn apart into several pieces. We therefore try to merge the small regions into larger ones using a simple clustering algorithm and get a final set of horizontal and vertical separators. In this task, we solve the explicit (black) separators. The white ones are inferred from the text segmentation map according to gaps between the regions.

The layout of processed pages is considered to be hierarchical. We thus apply a recursive region splitting algorithm which starts with one region containing the whole page. We first search for horizontal separators. If no horizontal ones are detected we search for the vertical ones. The separators are used for dividing the region into several parts and the same algorithm is then applied on each of the resulting parts. If we cannot divide the region further, the algorithm is finished (see Figure 3).

The result is a set of text blocks sorted with respect to assumed reading order.

3.3 Text-line Segmentation

The goal of this task is to extract images of individual text lines. For baseline prediction, we utilize ARU-Net which has proven to be very successful mainly on the task of baseline detection in handwritten text.

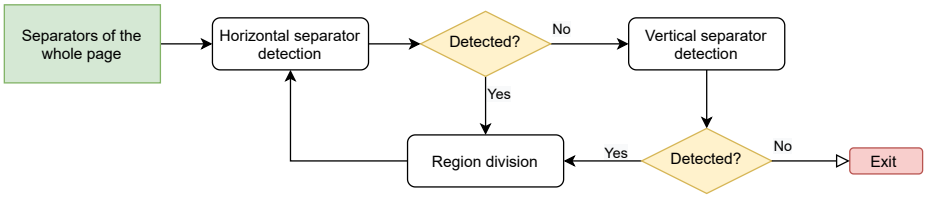

Fig. 3. Flowchart of the recursive region splitting algorithm

Baseline (a line where most characters rest upon and descenders extend below) representation of a text line is very simple and it has a number of advantages. The baseline representation is able to tackle skew and/or orientation as well as multi-column layout. Last but not least, the annotation of baselines is more manageable than annotating surrounding polygons which is another example of a text-line representation.

Because of the lack of annotated data, we utilize the original model which was trained on handwritten documents. Our preliminary experiments have shown that it is sufficient also for printed data.

The trained ARU-Net predicts a baseline map and its output needs to be post-processed in order to get exact baseline coordinates and to be able to extract the text-line image. Using the connected components statistics we derive the x-height (height of small letters without ascenders) and based on it and the baseline position we can extract the line image.

4 Experimental Setup

In this section, we first describe the data collections that were used for our experiments. We also report the evaluation criteria that we utilised for measuring the performance of the individual tasks and for the overall evaluation.

4.1 Corpora

Porta Fontium This dataset was created from document images digitised within the Porta fontium project[5]. We have selected a newspaper called *"Ascher Zeitung"*. This newspaper dates back to the second half of the nineteenth century and it is printed in German with Fraktur font. We collected 25 pages in total. The pages were annotated using Aletheia [6] and the layout is saved in the PAGE XML format. The transcription was utilised using tools presented in [13].

All pages are annotated on the paragraph level by bounding polygons. The test set contains also text lines with corresponding baselines.

Table 1 shows statistical information of this dataset. Regions include the number of text regions and separator regions.

[5] http://www.portafontium.eu/

Table 1. Porta fontium dataset statistical information (the number of documents, regions and text-lines)

Part	Documents	Regions	Text-lines
Train	10	338	1,156
Test	10	292	1,267
Validation	5	110	651

Europeana The goal of the Europeana project [5] is allowing access to digitised European cultural heritage. It contains a huge amount of various historical documents. There are also scanned newspapers among others.

We have selected a number of newspaper pages with similar characteristics as the data we are processing. The resulting set contains 95 pages written in German with highly variable page layouts. The pages used in this paper have corresponding ground-truths stored in the PAGE format. We have split the pages into train, validation and test parts containing 68, 9 and 18 pages respectively.

RDCL 2019 RDCL 2019 was designed for the ICDAR 2019 Competition on recognition of documents with complex layouts. This dataset contains scanned pages from contemporary journals. The character of images differs significantly from the two datasets described above. The images are of high quality and easy to binarize because they contain nearly no noise. We use the example set which contains 15 images with corresponding ground truths for testing.

4.2 Evaluation Criteria

In this section, we summarise the metrics we use to measure the performance and how we evaluated all parts of the approach as well as the overall evaluation.

Text Segmentation We use two pixel-based metrics, namely F1-score [19] and intersection over union (IoU) also known as Jaccard index [20] that are standard metrics in the segmentation task.

Both metrics are computed for each class (foreground and background) separately and the final result is an average value.

PRIMA Evaluator PRIMA Evaluator framework [7] was utilised both for the overall evaluation and for the evaluation of the separator detection algorithm. It offers a nice visualisation of segmentation errors and has many possibilities how to set the configuration (including different weights for different types of error: merge, split, false detection, miss or partial miss) according to the desired scenario.

To measure the performance of the separator detection algorithm we kept the default error weights and we filtered the success rate particularly for separator regions.

The overall evaluation was performed with a standard *General Document Recognition* evaluation profile. The result thus takes into consideration errors in text regions and in separator regions as well. Reading order is also incorporated into the overall score.

Baseline Detection We decided to completely follow the evaluation scheme presented by Grüning et. al in [10]. This scheme was utilized for ICDAR 2017 [8] and ICDAR 2019[6] competition on baseline detection (cBAD). The evaluation is carried out by a standalone JAR[7].

The *R-value* indicates how reliably the text is detected – ignoring layout issues while the *P-value* indicates how reliable the structure of the text lines (layout) is [10].

5 Experiments

The goal of the performed experiments is to evaluate the suitability of the above described networks for the tasks of text segmentation, separator segmentation and, above all, baseline detection. We also evaluate the whole presented method and compare the results with the state of the art.

5.1 Text-block Segmentation

In this section, we first evaluate and compare text segmentation based on the FCN network using three training scenarios:

1. Training from scratch on Europeana (E);
2. Training from scratch on Porta fontium (P);
3. Training from scratch on Europeana and then fine-tuning on Porta fontium (E+P).

We show the impact of three different input sizes on the segmentation results. The aim of this experiment is to find the most suitable configuration of the network. Table 2 shows the results on the validation part of Porta fontium dataset. All results are measured five times and we report the average and standard deviation values. During training we utilise early stopping based on validation loss.

Based on the results, we can conclude that larger dimensions of the input are beneficial for text segmentation. The pre-training on Europeana and fine-tuning on Porta fontium brings only slight improvement. The resolution of images is set to 960×672. We always train the model on Europeana and fine-tune it on the target dataset train part if available. Table 3 shows results of our system obtained on the three utilised datasets.

[6] https://scriptnet.iit.demokritos.gr/competitions/11/
[7] https://github.com/Transkribus/TranskribusBaseLineEvaluationScheme

Table 2. Text segmentation results (best results in bold)

Size	Data	F1	IoU
320x224	E	93.6 ± 0.16	88.1 ± 0.28
320x224	P	95.2 ± 0.08	91.0 ± 0.14
320x224	E+P	95.7 ± 0.05	91.8 ± 0.10
640x448	E	94.7 ± 0.11	90.0 ± 0.19
640x448	P	97.1 ± 0.07	94.4 ± 0.12
640x448	E+P	97.3 ± 0.02	94.8 ± 0.04
960x672	E	95.1 ± 0.08	90.8 ± 0.16
960x672	P	97.4 ± 0.03	95.0 ± 0.05
960x672	E+P	**97.4 ± 0.04**	**95.1 ± 0.08**

Table 3. Text segmentation results obtained with HP-FCN network and image size 960×672

	Europeana	Porta	RDCL 2019
F1	92.6	97.3	89.4
IoU	86.8	94.9	82.3

Next we employ the separator segmentation. We measure the performance of our separator detection algorithm (see Section 3.2). We consider the black (explicit) separators in this task. Finally, we evaluate the whole text-block segmentation algorithm. We use the PRIMA Evaluator tool for separator detection evaluation as well as for the final text-block segmentation evaluation.

For each testing page, we detect the separator regions and compare them with the corresponding ground truth using PRIMA Evaluator. We report the average results over 10 testing pages (see left part of Table 4). We present arithmetic mean (AM) of the separator success rates.

The right part of Table 4 shows the final evaluation of the whole text-block segmentation algorithm (see Section 3). We report the success rates obtained by PRIMA Evaluator with *document structure* evaluation profile as mentioned in Section 4.2.

Table 4. Separator detection results of the proposed approach (left) and the performance of the overall system (right) on three different datasets; average values of arithmetic mean (AM) are used

Database	**Separator detection**	**Overall evaluation**
Porta fontium	96.5	83.0
RDCL 2019	78.9	79.2
Europeana	83.1	82.2

The first line shows the results for the Porta fontium dataset. The separator detection (performed by ARU-Net) has excellent results. The evaluation

on RDCL 2019 brought, as expected, worse general results, including separator detection. This is caused primarily by the large number of different types of regions that our tools did not anticipate. Nevertheless, the general results for the relatively difficult RDCL 2019 datasest, which has not been part of a training, are decent. On Europeana, we have achieved comparable result for the overall evaluation. The results on separator detection are worse in this case.

5.2 Text-line Segmentation

This experiment describes the results of our text-line segmentation (see Section 3.3) being the final output of the proposed method. This experiment is realised only on the newly created Porta fontium dataset using annotation with baselines, because the other two corpora do not have this type of annotation.

We run the baseline detection in two ways. The first one is applying ARU-Net on the whole page. We will denote this approach as *page-level*. The second way is applying ARU-Net on already separated text regions. We will report it as *region-level*. We show these two approaches in order to point out the impact of our two-step algorithm in direct comparison with single-step approach.

We compare the results of our approach with Transkribus which is able to process a whole document and export a PAGE XML file containing baselines (see Table 5).

Table 5. Comparison of our text-line detection algorithm applied on the whole page (*page-level*) and results of the presented system (*region-level*); evaluation metrics are adopted from [10];we report average values for P-value, R-value and F-value

Model	P-value	R-value	F-value
Transkribus (baseline)	0.866	0.951	0.906
page-level approach	0.921	**0.994**	0.957
region-level approach	**0.960**	0.993	**0.976**

The results of *region-level* approach are better than those of *page-level* one. The main reason is that when applying the detection algorithm on regions we can reduce some false detections near region borders and we can also merge incorrectly split baselines which are frequent in ARU-Net predictions. Merging line candidates in the whole page would often lead to baselines crossing region separators.

Table 6 shows the numbers of detected baselines and it confirms better result for the *region-level* approach since the number of detected lines is closer to the ground truth number.

6 Conclusions and Future Work

In this paper, we have presented a novel approach for page segmentation into individual text lines. We have decomposed the problem into two separate steps:

Table 6. Number of detected (hypothesis) lines in Porta fontium

Model	Detected lines
Ground Truth	1,267
Transkribus	1,618
page-level	1,358
region-level	**1,293**

text-block segmentation and baseline detection. The text-block segmentation is solved by fully convolutional networks. The baseline detection is carried out using ARU-net architecture.

For evaluation, we have created a novel dataset from the data available in Porta fontium portal which is freely available for research purposes. We have compared the results of baseline detection algorithm applied directly on the whole page and that of our final system. By this approach, we managed to reduce the number of false positives in the detected baselines. We believe that by decomposing the problem into separate tasks, we can better express the document structure and not defile the reading order. Our two-step approach for baseline detection outperforms the single-step one applied directly on the whole page.

We have compared our text-line detection approach with Transkribus system and we have outperformed it with both page-level and region-level approaches. Moreover, it is evident from the results that the two-step approach performs better than the single-step one. We would like to highlight that the task of text-line segmentation is crucial for the following OCR processing that is usually the main goal of the historical document analysis.

One direction for further work would be to learn both the text segmentation and separator detection in one step. Another possibility is to train ARU-Net also for the task of x-line detection which could improve the x-height estimation process.

Acknowledgements

This work has been partly supported from ERDF "Research and Development of Intelligent Components of Advanced Technologies for the Pilsen Metropolitan Area (InteCom)" (no.: CZ.02.1.01/0.0/0.0/17_048/0007267)

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels. Tech. rep. (2010)
2. Breuel, T.M., Ul-Hasan, A., Azawi, M.I.A.A., Shafait, F.: High-performance ocr for printed english and fraktur using lstm networks. 2013 12th International Conference on Document Analysis and Recognition pp. 683–687 (2013)

3. Chen, K., Seuret, M., Hennebert, J., Ingold, R.: Convolutional neural networks for page segmentation of historical document images. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 965–970. IEEE (2017)
4. Chen, K., Seuret, M., Liwicki, M., Hennebert, J., Ingold, R.: Page segmentation of historical document images with convolutional autoencoders. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 1011–1015. IEEE (2015)
5. Clausner, C., Papadopoulos, C., Pletschacher, S., Antonacopoulos, A.: The enp image and ground truth dataset of historical newspapers. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 931–935. IEEE (2015)
6. Clausner, C., Pletschacher, S., Antonacopoulos, A.: Aletheia-an advanced document layout and text ground-truthing system for production environments. In: 2011 International Conference on Document Analysis and Recognition. pp. 48–52. IEEE (2011)
7. Clausner, C., Pletschacher, S., Antonacopoulos, A.: Scenario driven in-depth performance evaluation of document layout analysis methods. In: 2011 International Conference on Document Analysis and Recognition. pp. 1404–1408. IEEE (2011)
8. Diem, M., Kleber, F., Fiel, S., Grüning, T., Gatos, B.: cbad: Icdar2017 competition on baseline detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 1355–1360. IEEE (2017)
9. Galibert, O., Kahn, J., Oparin, I.: The zonemap metric for page segmentation and area classification in scanned documents. In: 2014 IEEE International Conference on Image Processing (ICIP). pp. 2594–2598. IEEE (2014)
10. Grüning, T., Labahn, R., Diem, M., Kleber, F., Fiel, S.: Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 351–356. IEEE (2018)
11. Grüning, T., Leifert, G., Strauß, T., Labahn, R.: A Two-Stage Method for Text Line Detection in Historical Documents (2019), http://arxiv.org/abs/1802.03345
12. 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)
13. Lenc, L., Martínek, J., Král, P.: Tools for semi-automatic preparation of training data for ocr. In: MacIntyre, J., Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds.) Artificial Intelligence Applications and Innovations. pp. 351–361. Springer International Publishing, Cham (2019)
14. Li, X.H., Yin, F., Xue, T., Liu, L., Ogier, J.M., Liu, C.L.: Instance aware document image segmentation using label pyramid networks and deep watershed transformation. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 514–519. IEEE (2019)
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
16. Mechi, O., Mehri, M., Ingold, R., Amara, N.E.B.: Text line segmentation in historical document images using an adaptive u-net architecture. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 369–374. IEEE (2019)

17. Novikov, A.A., Lenis, D., Major, D., Hladuvka, J., Wimmer, M., Bühler, K.: Fully convolutional architectures for multiclass segmentation in chest radiographs. IEEE transactions on medical imaging **37**(8), 1865–1876 (2018)
18. Pletschacher, S., Antonacopoulos, A.: The page (page analysis and ground-truth elements) format framework. In: 2010 20th International Conference on Pattern Recognition. pp. 257–260. IEEE (2010)
19. Powers, D.: Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. Journal of Machine Learning Technologies **2**(1), 37–63 (2011)
20. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 658–666 (2019)
21. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. Miccai pp. 234–241 (2015)
22. Sherrah, J.: Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv preprint arXiv:1606.02585 (2016)
23. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence **39**(11), 2298–2304 (2016)
24. Simistira, F., Ul-Hassan, A., Papavassiliou, V., Gatos, B., Katsouros, V., Liwicki, M.: Recognition of historical greek polytonic scripts using lstm networks. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. pp. 766–770. IEEE (2015)
25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
26. Wick, C., Puppe, F.: Fully convolutional neural networks for page segmentation of historical document images. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 287–292. IEEE (2018)
27. Xu, Y., He, W., Yin, F., Liu, C.L.: Page segmentation for historical handwritten documents using fully convolutional networks. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 541–546. IEEE (2017)
28. Zhong, X., Tang, J., Yepes, A.J.: Publaynet: largest dataset ever for document layout analysis. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1015–1022. IEEE (2019)