READ

RECOGNITION & ENRICHMENT OF ARCHIVAL DOCUMENTS

D6.5
Basic Layout Analysis P2

Markus Diem, Stefan Fiel, Florian Kleber
CVL

Distribution: http://read.transkribus.eu/

READ
H2020 Project 674943

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943

Project ref no.	H2020 674943
Project acronym	READ
Project full title	Recognition and Enrichment of Archival Documents
Instrument	H2020-EINFRA-2015-1
Thematic priority	EINFRA-9-2015 - e-Infrastructures for virtual research environments (VRE)
Start date/duration	01 January 2016 / 42 Months

Distribution	Public
Contract. date of delivery	31.12.2017
Actual date of delivery	22.11.2017
Date of last update	14.12.2017
Deliverable number	D6.5
Deliverable title	Basic Layout Analysis P2
Type	report
Status & version	in progress
Contributing WP(s)	WP5
Responsible beneficiary	NCSR
Other contributors	CVL,NCSR
Internal reviewers	URO,NCSR
Author(s)	Markus Diem, Stefan Fiel, Florian Kleber
EC project officer	Martin MAJEK
Keywords	Layout Analysis, Baseline Detection

Contents

1 Executive Summary

The basic layout analysis module extracts visual features from document images. These features include page segmentation (paragraphs), text-line segmentation, and the recognition of supplemental elements (e.g. images, ...). In D6.5 we present the further development of the basic layout analysis module from D6.4. We have implemented a new super pixel extraction and a new bottom-up clustering for text-line segmentation in the context of handwritten documents. The component labeling is trained and evaluated on the newly inroduced cBAD dataset [3]. An improved text block detection developed by NCSR is also presented and indirectly evaluated using text lines extraction performance. Furthermore, minor improvements such as the detection of graphical lines for text-line separation were implemented.

The module is part of the CVL READ framework. It is open source under LGPLv3 and available at github[1]. In addition to the command line testing routines, a plugin[2] for nomacs[3] is provided which allows for training and testing on either single images or a batch of images.

2 Super-Pixel

Super Pixels that are based on an improved MSER implementation were introduced in D6.4. Despite their ability to detect text independent to its color, it turned out that MSER regions are sparse. This is a drawback when dealing with cursive handwriting since statistics that are derived from super pixels are prune to local distortions. That is why we propose a new super pixel extraction in D6.5 based on grid cells. Therefore, the image is divided into multiple grids with changing cell sizes. A cell is then considered as activated if its accumulated gradient magnitude is larger than a threshold $t = 0.17$. Neighboring grid cells are merged if their gradient orientation is similar. For memory efficiency, each grid super pixel is approximated by an ellipse which is estimated by means of a PCA. Figure 1 shows MSER regions presented in D6.4 and the newly introduced grid super pixel (middle). The histogram in the top left corner indicates the distribution of edge strengths across all cells. A scale-space adoption of grid super pixels is shown in the right image.

3 Text-Line Segmentation

We presented a simple text line clustering in D6.4 that was based on Delaunay distance thresholds. This text-line segmentation is only applicable for printed text because it relies on equally distributed characters. In order to improve text-line segmentation, a bottom-up clustering approach is implemented that is similar to SLAC [1]. This approach first picks Delaunay edges that are considered good (i.e. minimal and aligned to the local orientation). Then, longer edges are added gradually. Before two text-lines

[1] `https://github.com/TUWien/ReadFramework`
[2] `https://github.com/TUWien/ReadModules`
[3] `https://nomacs.org`

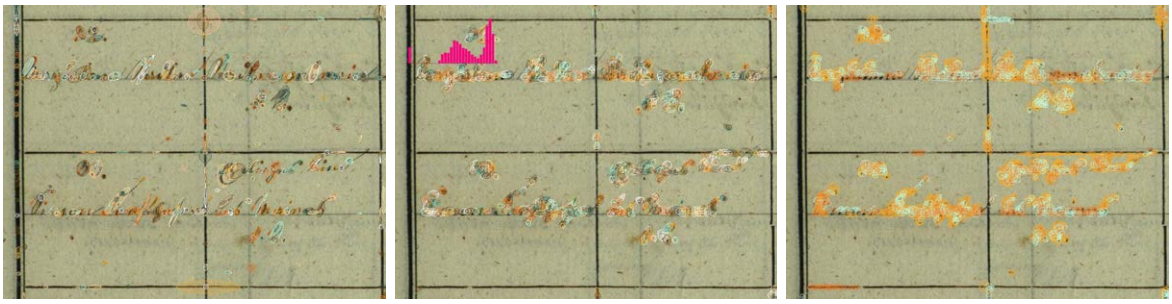

Figure 1: Detail of *0056_S_Alzgern_011-01_0056* with improved MSER output (left, see D6.4), newly introduced grid super pixels (middle) and scale space grid super pixels (right).

are merged, the linearity is checked of the resulting merged text-line. If a merge results in a non-linear text-line, it is rejected. Figure 2 shows merged edges after 900 iterations (left) and the final clustering result (right). Red edges indicate potential merges that were rejected. This text-line clustering is flexible with respect to the material presented and can deal with multiple text line orientations (i.e. the vertical text line in Figure 2). However, it tends to oversegment pages if noisy text is present.

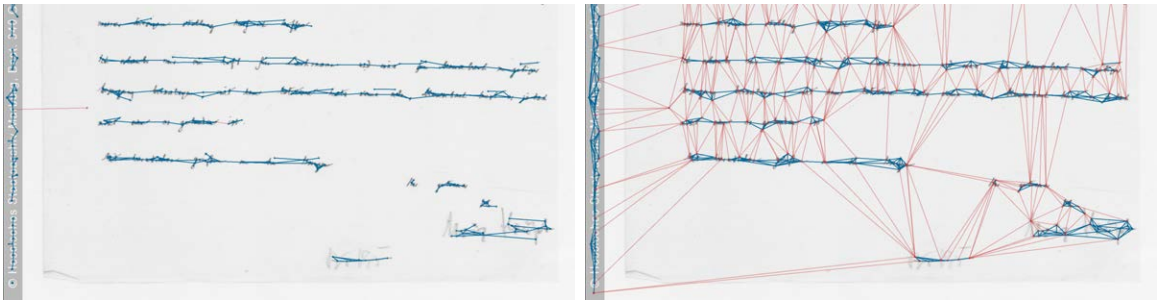

Figure 2: Raw baselines that are generated after clustering (left). Baselines filtered with respect to local orientation and text density (right).

Figure 3 (left) shows the resulting baselines that were generated with the previously discussed clustering approach. It can be seen that false baselines are detected because of background noise. We implement two approaches to tackle this issue. First, super pixels can be classified with respect to handwriting and noise. Rejecting *noise* super pixels would fix most issues shown here. An unsupervised approach that addresses this problem is to filter sparse and wrongly oriented text-lines. In order to find sparse text-lines, the super pixel density is computed and compared to other text-lines in the same image. If a text-lines density can be considered as statistical outlier, it is removed. In addition, text lines whose baseline orientation are significantly different to the local orientation are removed. Figure 3 (middle) shows the resulting text-lines after filtering according to these rules.

Ruling lines that graphically indicate table separators are often found in parish records. We detect these separators by the recently introduced Line Segment Detector (LSD) [2]. Since the LSD also detects strokes, we filter the lines with respect to length and ori-

entation. Hence, graphical separators are assumed to be approximately horizontal or vertical. While clustering, edges are rejected that cross graphical separators. By these means, we can correctly split baselines from neighboring columns that are horizontally aligned. Figure 3 (right) shows an example record where the last line is correctly split because of the separator.

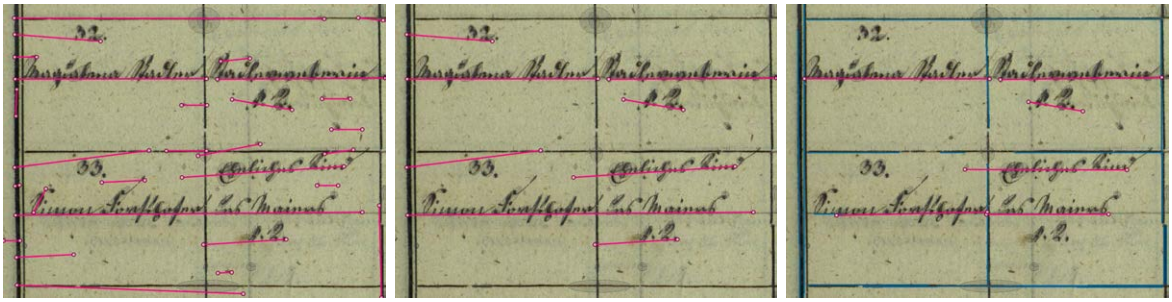

Figure 3: Raw baselines that are generated after clustering (left). Baselines filtered with respect to local orientation and text density (middle). Detected separators (blue) with text-lines split accordingly (right).

The evaluation of this text line segmentation is presented in D6.11. There, it can be seen that this method does not achieve state-of-the-art results in terms of precision and recall. However, its unsupervised nature and high recall make it suitable to different document domains such as cadasters (see Figure 4).

Figure 4: Text detection in a cadaster image of Venice using the proposed method.

4 Component Labeling

The component labeling which was presented in D6.4 can be trained for classifying any (visually distinct) class. We use Oriented FAST and Rotated Brief (ORB) features which

have similar properties as SIFT with a more compact representation (32 byte vs $128 \cdot 4$). The component labeling can be used to remove super pixels that are not located on text areas prior to clustering them. This improves the method's segmentation capabilities for it only has to deal with clean data. Figure 5 is a visualization of the *text non-text* training. We use the cBAD [3] training set to train a Random Forest classifier. Super pixels that are located on text-lines (blue) are labeled as *text* during training, those located outside text-lines (gray) are trained as *non-text*. The labels are sampled on pixel basis. If a super pixel's has an ambiguous local statistic (i.e. the label changes within the interquartile distance), we do not use it for training. Ambigous super pixels are labeled red in Figure 5.

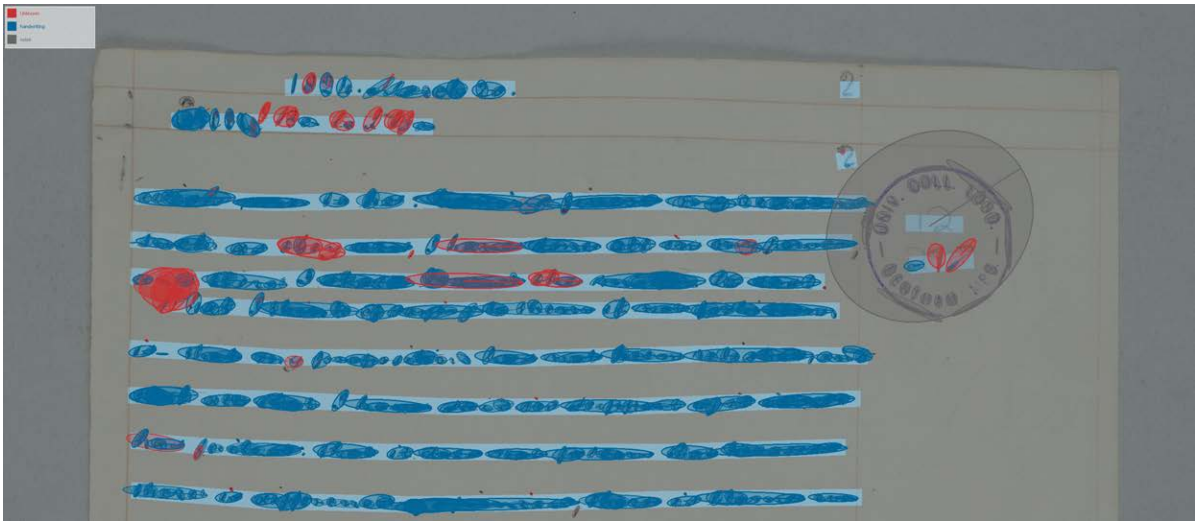

Figure 5: Super Pixel Training.

The classification performance is evaluated on the cBAD [3] test sets (Simple and Complex). For evaluation, we again use the manually annotated text-lines as groundtruth. The harmonic mean of precision (P), recall (R), and F-score (F) are used to determine the quality of component labeling. We evaluated two different training methodologies on the cBAD Simple dataset. First, we randomly choose a subset of 100k feature vectors of each class for training (200k Training). In the second experiment all feature vectors extracted from the cBAD trainingset are chosen. Figure 6 shows the resulting performance with respect to P, R, F. The F-score already indicates, that using all samples (even if they are unbalanced), improves classification. It can also be seen, that especially precision is improved (0.80 vs. 0.91). Hence, more training samples result in less false positives which indicates, that the machine can better tell the difference between bleed-through text and normal text. The only drawback of using more samples is that training time increases from 2 minutes to 30 minutes. Table 1 shows the classification performance for both cBAD test sets. The F-score does not drop significantly from cBAD Simple to cBAD Complex (0.80 vs. 0.77) despite the fact, that cBAD Complex is more challenging (see [3]).

Figure 7 shows a sample image from the cBAD complex dataset. The left image shows the classification results with blue being machine printed or handwritten text and gray

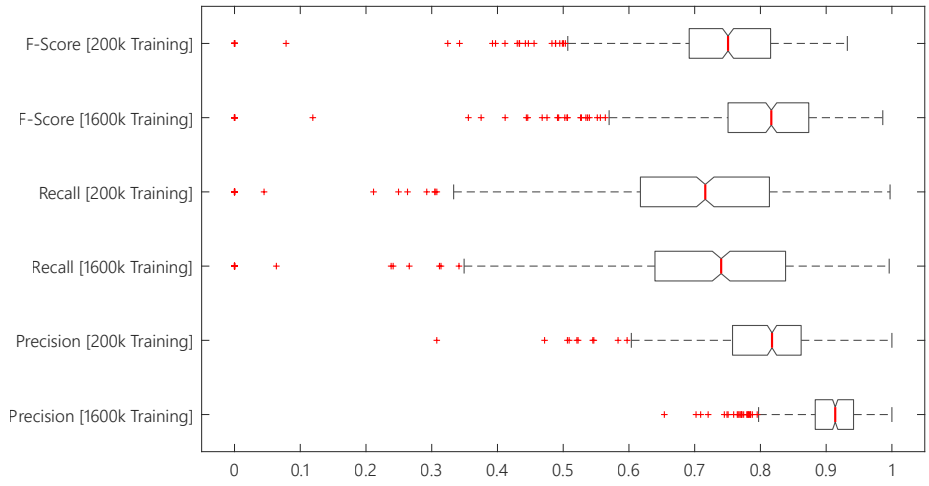

Figure 6: Training the Random Forest with different training set sizes.

	# Images	# Super Pixels	P	R	F
cBAD - Simple	539	2 458 844	0.91	0.71	0.80
cBAD - Simple (median)	539	2 458 844	0.91	0.74	0.82
cBAD - Complex	1010	5 021 494	0.89	0.67	0.77
cBAD - Complex (median)	1010	5 021 494	0.90	0.75	0.81

Table 1: Results of the *text, non-text* classification on the cBAD test sets.

being background clutter. The groundtruth is superimposed in the right image. Here, red indicates falsely classified super pixels while green shows correctly classified ones.

As mentioned in D6.4, we have also tested if an MRF can improve the labeling. The Random Forest's class probabilites are used here to construct the MRF costs. The graph is created using a Delaunay triangulation with Euclidean edge weights. Table 2 shows the evaluation result on a reduced cBAD testset with and without MRF voting. It can be seen, that the results are not significantly improved when applying an MRF.

	# Images	# Super Pixels	P	R	F
Default	33	143 212	0.922	0.741	0.822
With MRF	33	143 212	0.928	0.741	0.824

Table 2: Comparison with and without MRF voting.

5 NCSR Layout Analysis Method

During year 2 of the project, the NCSR group improved the existing layout analysis method that was developed during the tranScriptorium project [4]. This method is based on vertical line as well as vertical white runs detection, it uses polygons in order to represent the segmentation result and also includes classification of text regions to basic categories (main text – marginalia – header) (see Figure 8).

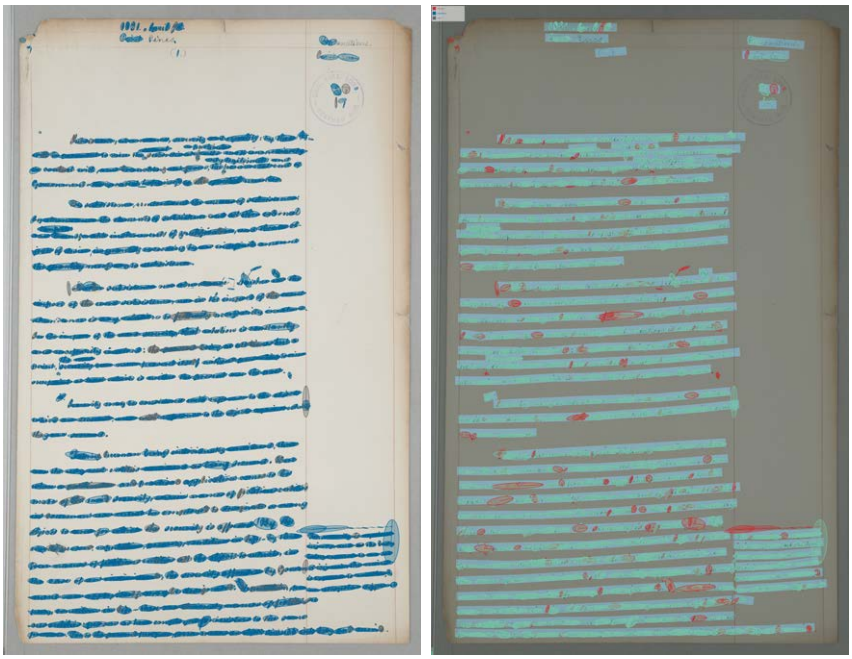

Figure 7: A sample image (*037_019_001*) with the classified label output (left) and the annotated labels (right).

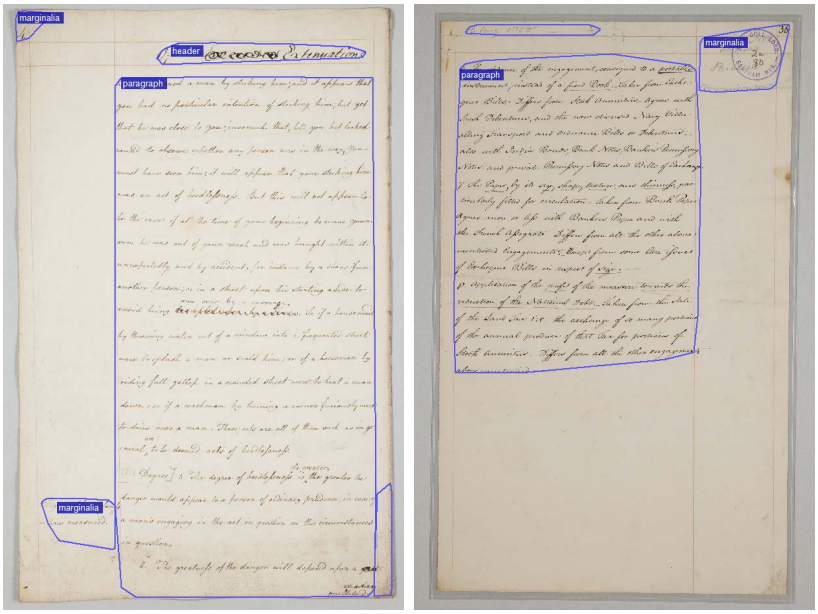

Figure 8: Representative results of the layout analysis method [4].

- Better formation of the rules involved in order to avoid cases of erroneously detecting text areas near the image border (see the erroneously detected text block at the bottom right part of the left image of Figure 8). - Add an image border removal starting step in order (a) to avoid confusing noise with the text areas and (b) to better calculate the average character height of the image (important parameter that is used in several

subsequent steps). In Fig.2, an example of the result with and without applying the image border removal step is presented. We indirectly evaluated the performance of the new layout analysis tool using the cBAD test dataset (Simple Scenario) by measuring the text line detection performance starting from several layout analysis results. As text line segmentation method, we used the new "NCSR 2nd year" method. The evaluation is done by calculating recall (R), precision (P) and F-value (F) which is the harmonic mean of (P) and (R). For more details for the dataset, the text line segmentation method and the evaluation protocol see also Deliverable D6.11 "Line and Word Segmentation Tools P2". Concerning the layout analysis method, we used three scenarios: a) start from the correct text block regions (GT), b) apply layout analysis method [1] and (c) apply the improved READ method. As it can be observed from Table 3, the text line detection performance is significantly improved when using the READ Y2 layout analysis method since the F-value has increased from 73.58% to 86.86%. This is mainly because the number of false alarms (noise that was considered as text) has been diminished. This is reflected at the total number of the detected text lines that was decreased from 46613 to 15257. Finally, it is noticeable that the performance achieved using the new layout analysis method (86.86%) is not far from the one achieved when starting from a 100% correct text block detection result (90.54%).

Layout Analysis	# GT Lines	# Lines Detected	P (%)	R (%)	F (%)
GT	14735	14496	89.54	91.57	90.54
Method [4]	14735	46613	62.32	89.83	73.59
READ Y2 method	14735	15257	85.05	88.74	86.86

Table 3: Indirect evaluation of the layout analysis methods.

References

[1] A. Delaye and K. Lee, "A flexible framework for online document segmentation by pairwise stroke distance learning," *Pattern Recognition*, vol. 48, no. 4, pp. 1197 – 1210, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320314004397

[2] R. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, apr 2010.

[3] M. Diem, F. Kleber, S. Fiel, T. Grüning, and B. Gatos, "cbad: Icdar2017 competition on baseline detection," in *In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*, 2017.

[4] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of historical handwritten documents into text zones and text lines," in *2014 14th International Conference on Frontiers in Handwriting Recognition*. IEEE, sep 2014.

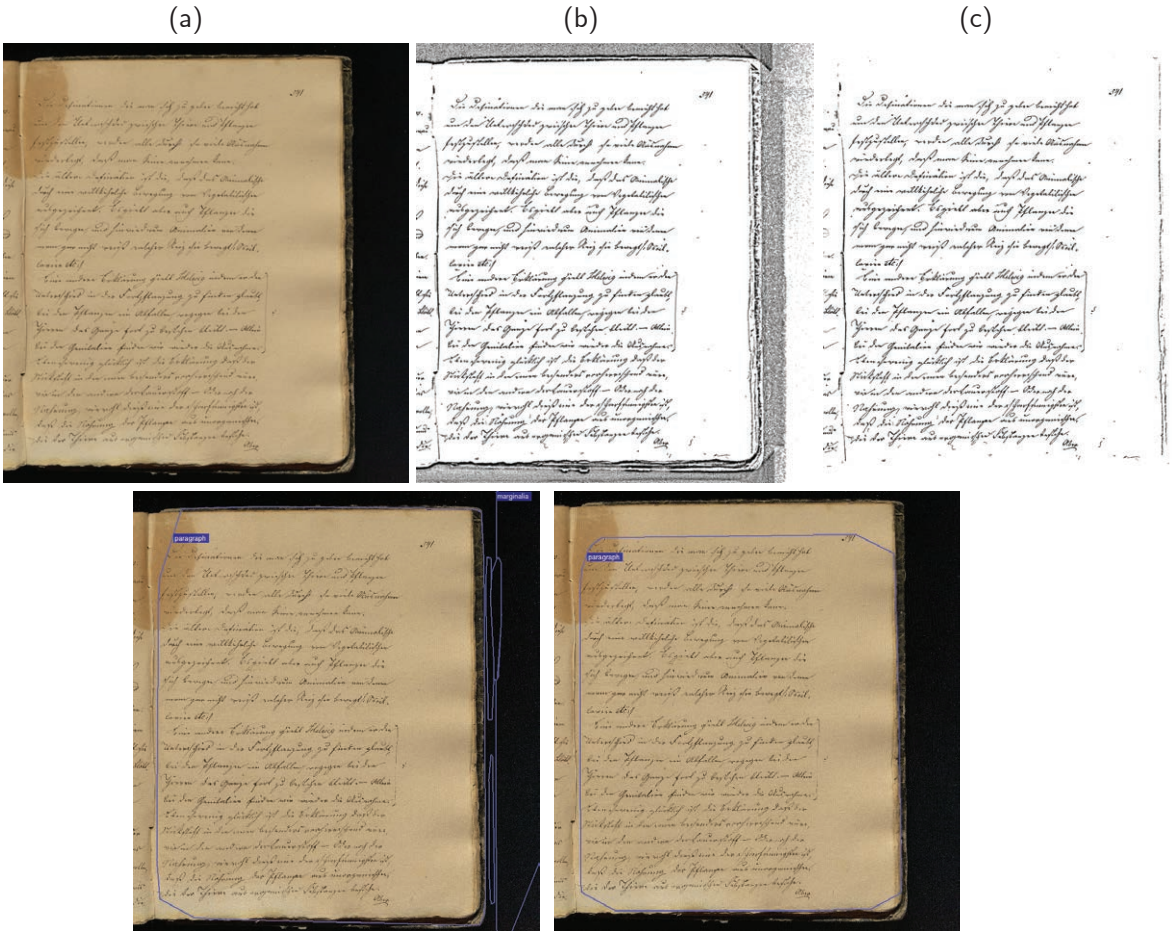

Figure 9: Representative results of the layout analysis method [4].