Multi-task Layout Analysis for Historical Handwritten Documents Using Fully Convolutional Networks

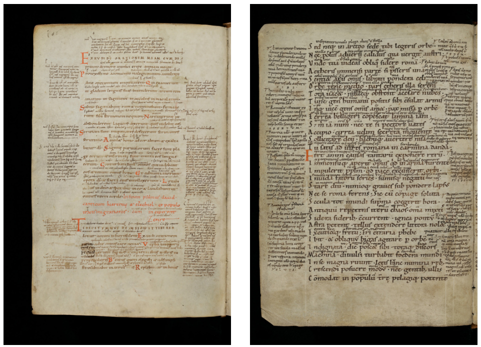
Yue Xu[1,2], **Fei Yin**[1,2], **Zhaoxiang Zhang**[1,2,3], **Cheng-Lin Liu**[1,2,3]
[1] National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences,
95 Zhongguan East Road, Beijing 100190, P.R. China
[2] University of Chinese Academy of Sciences, Beijing, P.R. China
[3] CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing, P.R. China
{yue.xu, fyin, liucl}@nlpr.ia.ac.cn, zhaoxiang.zhang@ia.ac.cn

Abstract

Layout analysis is a fundamental process in document image analysis and understanding. It contains three key sub-processes which are page segmentation, text line segmentation and baseline detection. In this paper, we propose a multi-task layout analysis method that uses a single FCN model to solve the above three problems simultaneously. In our work, a multi-task FCN is trained to segment the document image into different regions (background, main text, comment and decoration), circle the contour of text lines and detect the centerlines of text lines by classifying pixels into different categories. By supervised learning on document images with pixel-wise labeled, the FCN can extract discriminative features and perform pixel-wise classification accurately. Based on the above results, text lines can be segmented and the baseline of each text line can be determined. After that, post-processing steps are taken to reduce noises, correct wrong segmentations and produce the final results. Experimental results on the public dataset DIVA-HisDB [Simistira *et al.*, 2016] containing challenging medieval manuscripts demonstrate the effectiveness and superiority of the proposed method.

1 Introduction

Layout analysis is a fundamental task in document image analysis. It is the basic step of the OCR (optical character recognition) system and influence the performance of subsequent modules. Layout analysis is challenging especially for historical documents (see Fig. 1) due to the complexity and variability of page layout, the mixing of different elements and the degradation of historical documents. In this paper, we focus on three key sub-processes of layout analysis: page segmentation, text line segmentation and baseline detection.



(a) (b)

Figure 1: Document samples from DIVA-HisDB.

Page segmentation is the process that segments the document images into different regions with uniform elements like background, main texts, comments, decorations, etc. Previous methods on page segmentation can be divided into two categories: top-down and bottom-up. Top-down approaches [Nagy *et al.*, 1992],[Uttama *et al.*, 2005],[Ouwayed and Belaïd, 2008] start from the whole page and cut it into small areas. These areas will be split or merged to produce homogeneous regions. Bottom-up approaches [Bukhari *et al.*, 2012], [Mehri *et al.*, 2015] usually take the pixels or connected components as the basic elements. These elements will be merged into larger homogeneous regions by analysing their features. The top-down method is easily applicable but not suitable for complex layout. On the contrary, the bottom-up approach is superior to the documents of irregular layout but need more computational demand.

Text line segmentation is a challenging task for handwritten documents because the text lines tend to be skewed and curved, while the interline spacing is not uniform. Therefore, text line segmentation methods based on projection analysis and Hough transform are not suitable for handwritten documents in many cases. The baseline is the fictitious line which follows and joins the lower part of the text lines. It can be used for both skew correction and character segmentation. For many handwritten text recognition systems, baseline detection is an essential stage and crucial for the character recognition performance. Many algorithms on baseline de-

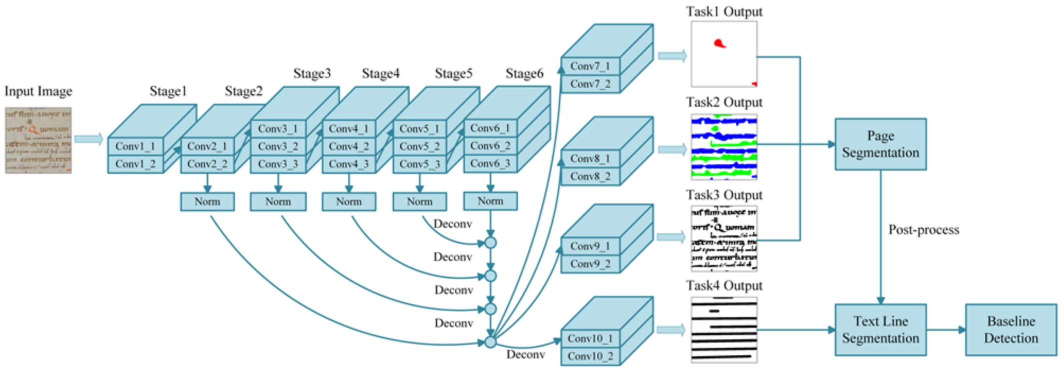


Figure 2: Pipeline of the proposed method.

tection have been proposed, for example, the histogram projection based methods [Al-Badr and Mahmoud, 1995], the Hough transform based methods [Razak *et al.*, 2008], the word skeleton based methods [Pechwitz and Margner, 2002], the word contour based methods [Faisal *et al.*, 2005], and so on.

The classical fully convolutional network (FCN) has made great success in many area [Long *et al.*, 2015], [Xie and Tu, 2015], which gives us a lot inspiration. Most of the previous method only solve one sub-problem in the document layout analysis. In our work, we proposed a multi-task layout analysis framework based on the fully convolutional network (FCN) for historical handwritten documents. This framework uses one FCN model to solve the page segmentation, text line segmentation and baseline detection problem simultaneously. Firstly, the framework trains a multi-task FCN to predict pixel-wise classes. The first two branch predict the region of main text, comment and decoration, as well as the coarse contour outside of the text lines. The third branch is trained to binarize the image. And the fourth branch is used to learn center lines of text lines. Then, heuristic based post-processing is adopted to reduce noise and correct misclassification. Finally, the prediction of the four branches will be combined to produce the result of page segmentation, text line segmentation and baseline detection.

The contributions of our work are mainly in three points: 1) it is the first FCN based framework which can solve three layout analysis problems simultaneously. 2) we provide a complete layout analysis framework including the strategies for large images and post refinement. 3) we achieve the state-of-the-art performance on the public competition dataset [Simistira *et al.*, 2016], indicating the effectiveness of our method.

The remainder is organized as follows. Section II reviews related works on layout analysis for document images. Section III describes the details of the proposed method. Section IV presents our experimental results and analysis. Finally, concluding remarks are given in Section V.

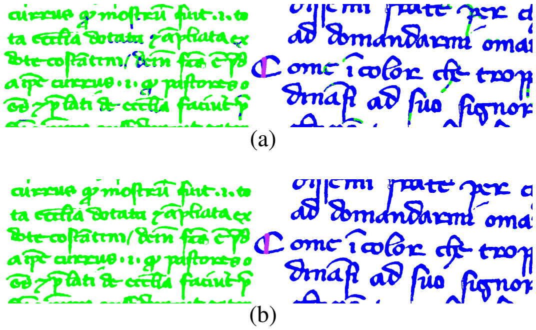
2 Related Work

Page Segmentation Page segmentation methods can be broadly divided into two categories: top-down approach and bottom-up approach. [Chen and Wu, 2009] proposed a top-down method for complex document images. This method cuts the document into blocks which are then multi-thresholded to create several layers. Then the connected components of each layer are identified and grouped across blocks based on a predefined set of features. [Mehri *et al.*, 2015] proposed a bottom-up method based on learning texture features for historical document image enhancement and segmentation. This method used the simple linear iterative clustering (SLIC) super-pixels, Gabor descriptors and support vector machines (SVM) to classify pixels into foreground and background.

Text Line Analysis Many algorithms have been proposed for text line segmentation and baseline detection. [Li *et al.*, 2008] segments text lines based on density estimation. For an input document image, this method estimates a probability map where each element represents the probability of the underlying pixel belonging to a text line. The method is then exploited to determine the boundary of neighboring text lines by evolving an initial estimate. [Yin and Liu, 2009] is a text line segmentation method for unconstrained handwritten documents based on the minimal spanning tree (MST) clustering without artificial parameter. The connected components of document image are grouped into a tree structure. Then, Text lines are extracted by dynamically cutting the edges of the tree. [Chakraborty and Pal, 2016] proposed a baseline detection scheme for handwritten text lines. This method detects the contour points of the text lines. Then, a SVM will be trained by using the orientation invariant features of the contour curves. Finally, the curves will be classified and sorted to get the optimal baselines. [Al-Badr and Mahmoud, 1995] introduced a baseline detection method by analysing the projection of the writing tracing points according to a predefined direction. The baseline detected by this method is coincided with the local maximum of these histograms.

3 Proposed Method

In this section, we firstly introduce the FCN based network for multi-task layout analysis and some crucial modifications from previous works. Secondly, the algorithm for page segmentation and its post-process like small region correction


(a)

(b)

Figure 3: (a) the coarse prediction of page segmentation. (b) the refinement result of page segmentation. The colors: white, blue, green, purple represent background, main texts, comments, overlap regions of main texts and decorations respectively.

are illustrated. Thirdly, the method for text line segmentation will be introduced. Finally, the method for baseline detection will be given.

3.1 Network Structure

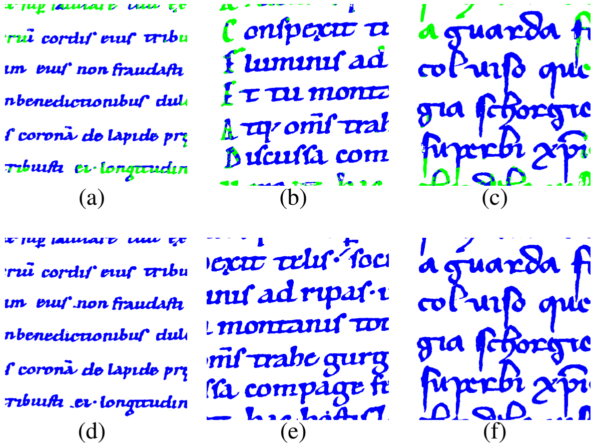
The multi-task layout analysis network is based on the FCN for semantic segmentation. The first 5 convolutional stages follow the design of VGG-16 network [Simonyan and Zisserman, 2015]. However, there are three modifications from previous works.

Firstly, unlike segmentation for generic objects, document layout analysis task like page segmentation requires more accurate partitions on strokes. Consequently, we combine more low-level features (Stage 2, Fig. 2) which could provide more information of details.

Secondly, the maximum receptive field of VGG 16-layer net is around 224, which may not contain enough context if the character and line space is large. To understand the context better, we design a deeper network with larger receptive field by adding three additional 3 × 3 convolutional layers (Stage 6, Fig. 2) to the top of the Stage 5 in VGG 16-layer net.

Thirdly, in order to make full use of annotation information and solve the three problems in one network simultaneously, we use the multi-task architecture with four output branches (see Fig. 2) and it offers superior performance for the layout analysis problem. The first branch is used to learn the red decoration regions while the second branch is used to learn the main text body and comment regions, these two branch learn not only the text category information, but also the coarse contour outside of the text lines. The third branch is trained to binarize the documents and extract the foreground pixels. And the fourth branch is used to learn the center line of each text line.

The ground truth for the first three task is provided by the competition data set directly, and the ground truth for task 4 is obtained by processing the contour information for each text line offered by the competition data set. All the ground truth is in pixel level. By simultaneously training on the four task, the network is able to obtain a finer pixel-wise classification result.


(a) (b) (c)

(d) (e) (f)

Figure 4: (a)(b)(c) The prediction for testing patches. (d)(e)(f) the ground truth for testing patches.

The structure of the multi-task FCN used in our work is shown in Fig. 2. Before feature fusion and deconvolution, we normalize the channel size of each stage to 32 with a 1 × 1 kernel. All the deconvolution layers have a 2 × 2 kernel with the stride 2. We employ the *Softmax Loss* for the pixel-wise optimization.

3.2 Page Segmentation

To solve the page segmentation problem, we combine the prediction of the first three branches. Foreground pixels (predicted by branch 3) that fall in the decoration regions (predicted by branch 1) will be classified as decoration. Similarly, foreground pixels that fall in the main text regions or comment regions (predicted by branch 2) will be classified as main text or comment respectively. It is worth noting that, decorations may overlap the main texts or comments. Pixels in overlap regions belong to two categories. Through this combination, we get a coarse page segmentation result with noises and misclassified components. To refine the prediction, we perform the following three steps.

Firstly, before combining the first three branches, we pay attention to the prediction of branch 2. Since the main texts and comments are interlaced together, they are easily to be misclassified. To correct the wrong regions, we make the following assumptions. First, the category of small isolated regions tends to be the same as that of its surroundings. Second, the length of contacted boundary between main texts and comments is short. Suppose C_a and C_b are the adjacent CCs belonging to different classes (main text or comment), and their categories are predicted to be A and B respectively. L_a is the boundary length of C_a, and L_{ab} is the length of contacted boundary between C_a and C_b. If $L_{ab} \geq L_a/3$, C_a will be considered as an isolate CC surrounded by Class B. Then, pixels in 320×320 window will be counted. N_a is the pixel number of class A and N_b is the pixel number of class B. If $N_b > N_a$, A will be corrected to B. And vice versa. After that, we combine the first three branches. The refinement result is shown in Fig 3.

Secondly, for each foreground connected component (CC),

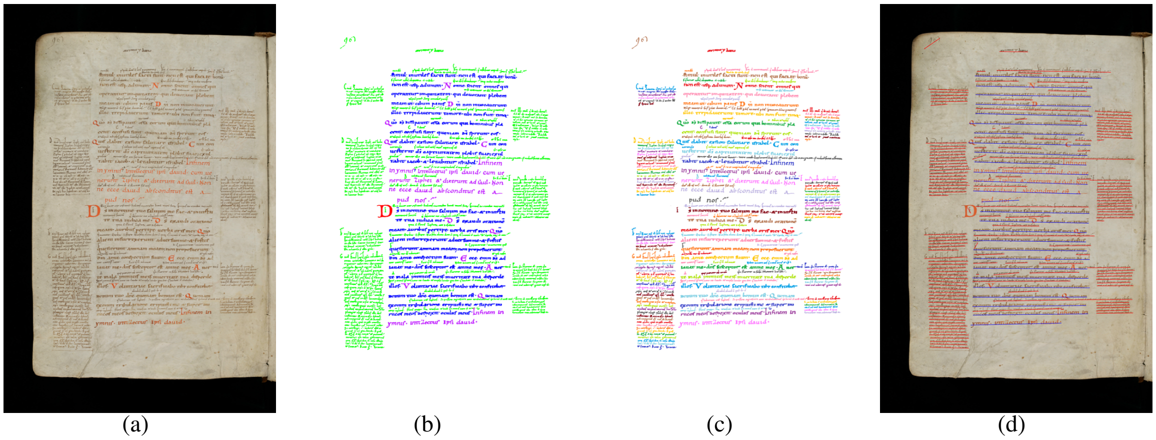


Figure 5: (a) Original document image. (b) Page segmentation result. The colors: blue, green, red and purple represent main texts, comments, decorations, overlap regions of main texts and decorations respectively. (c) Text line segmentation result. (d) Baseline Detection result.

if 80% of it is covered by decoration regions, the whole CC can be classified as decoration. And according to the characteristics of the decorations described by competition data set, a decoration CC will be identified as pure decoration if it is large enough. Here, pure decoration means it is classified as decoration only.

Finally, isolated foreground CCs in small size will be dropped to reduce the noise.

After these processes, we obtain the final page segmentation result (see Fig 5. (b)).

3.3 Text Line Segmentation

To solve the text line segmentation problem, we combine the page segmentation result and the output of branch 4. For each text line, the prediction of branch 4 gives a thick center line which connects all the text line CCs ideally. According to this point, we can solve this problem by analysing the center line.

Firstly, considering that the main texts and comments are interlaced together, we divide text CCs and text center lines into two categories (main text and comment) based on the page segmentation result. Secondly, we count the number of pixels in each center line CCs. center lines with too few pixel number will be regarded as the noise and removed. Finally, for each category, we segment the text lines separately. Text CCs connected by a center line will be merged into the same text line. The Isolated CCs can be assigned to the nearest line according to the distance information. A CC can also be cut off based on the distance information if it is crossed by different lines.

The final result of text line segmentation has been shown in Fig. 5(c). Compared to the traditional algorithm, our method is not affected by the direction of text lines as well as the size and form of characters. In addition, this method can produce accurate text line segmentation results without character over-segmentation for the complex documents with text lines stroke adhesion and interline comments.

3.4 Baseline Detection

We combine the result of text line segmentation and the output of branch 2 to produce the baseline of each text line. The output of branch 2 offers the coarse contour outside of the text lines.

Firstly, we attach the contour to different text lines based on the text line segmentation result. Secondly, lower contour points of each text line is extracted. Thirdly, we use the least square methods to find a baseline through these points. Then, the median distance between the points and the line is computed. Outliers points which are farther than the median distance will be discard. After that, we compute the baseline of the remaining points and obtain the final result (see Fig. 5(d)).

4 Experiments

In this section, firstly, we introduce the dataset and the evaluation metric. Secondly, we list the implementation detail of our experiments. Finally, we give the experiment results.

4.1 Dataset

The proposed method is tested on DIVA-HisDB dataset [Simistira *et al.*, 2016]. DIVA-HisDB is a historical manuscript dataset that consists of three types of medieval manuscripts (CSG0018, CSG0863, CB0055) with complex layout elements, diverse scripts, and challenging degradations (see Fig. 1). There are total 150 annotated pages, including 60 images for training, 30 images for validation, and 60 images for testing. This dataset offers the annotated images for page segmentation, text line segmentation and baseline detection.

For the page segmentation task, pixels are divided into four categories: background, main text body, comments (marginal and interlinear glosses, explanations, corrections) and decorations (characters/signs that exceed the size of a text line and written in red). It is worth noting that a pixel can have more than one label.

For the text line segmentation task, the text line contour surrounding the foreground is given in polygon form. In particular, when evaluate the page segmentation task, background pixels within the special contour can be classified as either foreground or background.

For the baseline detection task, the baseline is given in the form of a straight line. The dataset offers two endpoints of the baseline for each text line.

4.2 Implementation Details

The network is optimized by stochastic gradient descent (SGD) with back-propagation and the maximum iteration is 200,000. The learning rate is fixed to be 0.01 for the first 50,000 iterations and then degraded to 0.001 until the end of training. For the initialization of the network, layers are all initialized by "xavier" [Xavier and Yoshua, 2010]. Weight decay is 4×10^{-4} and momentum is 0.9. The whole experiments are conducted on Caffe [Jia *et al.*, 2014] and run on a workstation with 2.9GHz, 12-core CPU, 256G RAM GTX Titan X and Ubuntu 64-bit OS.

Limited by the memory of GPU, the whole image can not be trained or tested directly. Therefore, we crop images into small patches. During the training stage, the input images are randomly cropped from the origin training images. All the cropped patches follow the same size of 320×320. In this work, we generate about 180,000 training patches.

Discriminative features such as character size, stroke shape and context information are important for prediction. However, during the test stage, characters at the boundary of the patch images will be cut off and vital features would be lost a lot, which brings misclassification (Fig. 4). To solve this problem, we crop the testing images into 640×640 patches with stride of 480 in a sliding window manner. And then we only use the result of the center 480×480 region and abandon the border area whose size is corresponding with the average character size.

4.3 Experiment Result

The performance of our method is evaluated by IU, F1-score, Precision, and Recall, they are defined as Eq. (1). Where TP denotes the True Positives, FP denotes the False Positives and FN denotes the False Negatives.

$$Precision = \frac{TP}{TP + FP} \tag{1a}$$

$$Recall = \frac{TP}{TP + FN} \tag{1b}$$

$$IU = \frac{TP}{TP + FP + FN} \tag{1c}$$

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{1d}$$

Page Segmentation

For each page, the evaluation metric is computed category-wise (background, main text, comment, decoration). For each category i, first, we count IU_i (Intersection over Union), $F1_i$ (F1-score), P_i (Precision) and R_i (Recall) in pixel level within the whole image. Then, the final evaluation based on the mean value and the frequency will be obtained according to Eq. 2 respectively. Where F_i means the pixel frequency of class i.

$$\begin{cases} IU_{mean} &= \Sigma IU_i \\ F1_{mean} &= \Sigma F1_i \\ P_{mean} &= \Sigma P_i \\ R_{mean} &= \Sigma R_i \end{cases} \begin{cases} IU_{freq} &= \Sigma IU_i \times F_i \\ F1_{freq} &= \Sigma F1_i \times F_i \\ P_{freq} &= \Sigma P_i \times F_i \\ R_{freq} &= \Sigma R_i \times F_i \end{cases} \tag{2}$$

Fig. 5(b) shows the final page segmentation results for the testing image. The performance of the proposed method is evaluated on pixel level. Compared with the top two algorithms of the competition [Simistira *et al.*, 2017], as shown in Table. 1, our proposed method achieves the state-of-the-art performance with 95.47% for $meanIU$, which surpass the rank1 method for 0.57%. In order to provide a more exhaustive evaluation of our method, other standard metrics including F1-score, Precision and Recall are given. Table. 2 listed the standard metrics that averaged over classes while Table.3 gives the metrics based on frequency. The performance of the contrast method (Proposed*, without combining the low-level feature) is also listed below.

	CSG0018	CSG0863	CB0055	Total
Proposed	**95.38**	**94.35**	96.69	**95.47**
Proposed*	94.98	91.26	97.76	94.67
Rank1	93.65	92.71	98.35	94.90
Rank2	93.57	89.63	**98.64**	93.95

Table 1: Mean IU result for page segmentation (in %).

	IU_{mean}	$F1_{mean}$	P_{mean}	R_{mean}
Proposed	**95.47**	**97.52**	**99.00**	96.52
Proposed*	94.67	96.55	97.93	96.53
Rank1	94.90	96.81	97.58	**97.20**
Rank2	93.95	96.04	96.55	97.10

Table 2: Category-average metric for page segmentation (in %).

	IU_{freq}	$F1_{freq}$	P_{freq}	R_{freq}
Proposed	98.93	99.47	96.52	99.45

Table 3: Frequency-average metric for page segmentation (in %).

Text Line Segmentation

The evaluation of the line segmentation task is calculated in line level and pixel level. The line-level metric evaluate how many of the lines have been correctly detected. In this case, TP is the number of correctly detected lines, FP is the number of extra lines and FN is the number of missed lines. The pixel-level metric evaluate how well are the line detected. In this case, TP is the number of correctly detected pixels, FP is the number of extra pixels and FN is the number of missed pixels. Two versions of the Pixel-level IU are reported. The first one PIU takes all pixels into account, while the second one $MPIU$ only reports on pixels within the matched lines. The former metric gives an overall evaluation of the method,

	CSG0018			CSG0863			CB0055			Total		
	LIU	PIU	MPIU	LIU	PIU	MPIU	LIU	PIU	MPIU	LIU	PIU	MPIU
Proposed	**99.01**	**98.97**	**99.24**	**99.83**	**98.74**	**98.80**	**99.38**	97.84	97.89	**99.41**	**98.51**	**98.64**
Proposed*	98.32	98.12	98.29	95.30	98.12	98.54	99.28	**98.22**	**98.54**	97.63	97.97	98.17
Rank1	94.90	94.47	96.24	96.75	90.81	92.29	99.33	93.75	94.02	96.99	93.01	94.18
Rank2	69.57	75.31	92.28	90.64	93.68	96.07	84.29	80.23	88.82	81.50	83.07	91.27

Table 4: IU Result for text line segmentation (main text, in %).

	IU			F1			Precision			Recall		
	LIU	PIU	MPIU	LF1	PF1	MPF1	LP	PP	MPP	LR	PR	MPR
main text	99.41	98.51	98.64	99.69	99.25	99.31	99.88	99.34	99.39	99.51	99.15	99.24
main text + comment	97.63	97.97	98.17	98.79	98.97	99.07	99.08	99.24	99.30	98.52	98.71	98.85

Table 5: Detail Result for text line segmentation (in %).

	CSG0018	CSG0863	CB0055	Total
Proposed	**99.48**	**99.89**	**99.36**	**99.57**
Proposed*	98.79	99.51	98.51	98.94
Rank1	98.53	97.16	98.96	98.22
Rank2	98.79	98.30	95.97	97.68

Table 6: F1-score result for base line detection (main text, in %).

	Precision	Recall	F1-score
main text	99.67	99.49	99.57
main text + comment	97.42	97.33	97.36

Table 7: Detail result for base line detection (in %).

while the latter is interesting for assessing the quality of the matched lines.

Fig. 5(c) shows the final text line segmentation result for the sample image. The performance of the proposed method for text line segmentation is shown in Table. 4. Compared with the top two algorithms of the competition, our method makes the best results in all the evaluation metrics. It worth noting that, among 60 public testing images, 30 of them offer the complete text line information for main text and comment, while 30 of them only provide main text information. The public competition only evaluate the text line segmentation and baseline detection result for main text, which may not judge the algorithm performance completely. We test our method for both main text and comment on 30 testing images, and the result is shown in Table. 5 and Table. 7.

Baseline Detection

The evaluation of the baseline extraction is based on a new performance measure [Gruning *et al.*, 2017] that finds the start and end points of the he baseline of each textline. The evaluation threshold (20 pixels difference in y-direction is considered as being correct) is used to measure the precise quality of the detected baseline. Fig. 5(d) shows the baseline detection result for the sample image. The performance of the proposed method is listed in Table. 6, Table. 7. Compared with the top two algorithms, our methods achieves a new state-of-the-art performance and outperformed the rank1 method in all three competition tasks.

Necessity of Low-level Feature

Compared with generic objects segmentation, document layout analysis requires more accurate partitions on strokes, which bring us make some modification from previous works. To receive more information of details, we combine more low-level features (Stage 2, Fig. 2) in our framework. To verify the necessity of this low-level feature in our method, we did a contrast experiment. The contrast experiment delete the connection with Stage. 2. The performance of the contrast method (Proposed*) has been listed in the above tables. It obvious that the performance of the contrast method is lower than our proposed method. The experiment results show that low-level feature is very necessary for document layout analysis.

5 Conclusion

In this paper we present a multi-task layout analysis framework for historical handwritten documents. The whole framework contains two parts: FCN based classification and post-processing. First, we innovatively adopted a multi-task FCN based network to predict the foreground categories, the text contour and the text line center. Then, heuristic based post-processing is used to reduce noise and correct misclassifications. Finally, the previous predictions are combined to produce the final results of page segmentation, text line segmentation and text line detection.

This paper provide a complete layout analysis framework for complex historical handwritten documents, and introduce the effective strategies for large images and post refinement. On the DIVA-HisDB competition dataset, we have achieved a new state-of-the-art performance and outperformed the rank1 method by a large margin, which shows the effectiveness and superiority of the proposed method.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (NSFC) grants 61721004, 61411136002, 61573355, 61773375 and 61761146004.

References

[Al-Badr and Mahmoud, 1995] Badr Al-Badr and Sabri Mahmoud. Mahmoud, s.a.: Survey and bibliography of arabic optical text recognition. In *Proceedings of the Signal Processing*, pages 49–77, 1995.

[Bukhari *et al.*, 2012] Syed Saqib Bukhari, Thomas M. Breuel, Abedelkadir Asi, and Jihad El-Sana. Layout analysis for arabic historical document images using machine learning. In *Proceedings of the 13th International Conference on Frontiers in Handwriting Recognition*, pages 639–644, 2012.

[Chakraborty and Pal, 2016] Dibyayan Chakraborty and Umapada Pal. Baseline detection of multi-lingual unconstrained handwritten text lines. In *Proceedings of the Pattern Recognition Letters*, pages 74–81, 2016.

[Chen and Wu, 2009] Yenlin Chen and Bingfei Wu. A multiplane approach for text segmentation of complex document images. In *Proceedings of the Pattern Recognition*, pages 1419–1444, 2009.

[Faisal *et al.*, 2005] Farooq Faisal, Govindaraju Venu, and Perrone Michael. Pre-processing methods for handwritten arabic documents. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 267–271, 2005.

[Gruning *et al.*, 2017] Tobias Gruning, Roger Labahn, Markus Diem, Florian Kleber, and Stefan Fiel. Read-bad: A new dataset and evaluation scheme for baseline detection in archival documents. In *CoRR*, page abs/1705.03311, 2017.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 675–678, 2014.

[Li *et al.*, 2008] Yi Li, Yefeng Zheng, David Doermann, and Stefan Jaeger. Script-independent text line segmentation in freestyle handwritten documents. In *Proceedings of the Pattern Analysis and Machine Intelligence*, pages 1313–1329, 2008.

[Long *et al.*, 2015] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, page 3431–3440, 2015.

[Mehri *et al.*, 2015] Maroua Mehri, Nibal Nayef, Pierre Héroux, Petra Gomez-Krämer, and Rémy Mullot. Learning texture features for enhancement and segmentation of historical document images. In *Proceedings of the 3rd International Workshop on Historical Document Imaging and Processing*, pages 47–54, 2015.

[Nagy *et al.*, 1992] George Nagy, Mahesh Viswanathan, and Sharad Seth. A prototype document image analysis system for technical journals. In *Proceedings of the IEEE*, pages 10–22, 1992.

[Ouwayed and Belaïd, 2008] Nazih Ouwayed and Abdel Belaïd. Multi-oriented text line extraction from handwritten arabic documents. In *Proceedings of the Eighth Iapr International Workshop on Document Analysis Systems*, pages 339–346, 2008.

[Pechwitz and Margner, 2002] M. Pechwitz and V. Margner. Baseline estimation for arabic handwritten words. In *Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition*, pages 479–484, 2002.

[Razak *et al.*, 2008] Zaidi Razak, Khansa Zulkiflee, Mohd Idris, Emran Mohd Tamil, Mohd Noorzaily Mohamed Noor, Rosli Salleh, Mohd Yaakob, Zulkifli Mohd Yusof, and Mashkuri Yaacob. Off-line handwriting text line segmentation: A review. In *International Journal of Computer Science and Network Security*, pages 12–20, 2008.

[Simistira *et al.*, 2016] Foteini Simistira, Mathias Seuret, Nicole Eichenberger, Angelika Garz, Marcus Liwicki, and Rolf Ingold. Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts. In *Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition*, pages 471–476, 2016.

[Simistira *et al.*, 2017] Fotini Simistira, Manuel Bouillon, and Mathias Seuret. Icdar2017 competition on layout analysis for challenging medieval manuscripts. In *Proceedings of the 14th International Conference on Document Analysis and Recognition*, pages 1361–1370, 2017.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

[Uttama *et al.*, 2005] Surapong Uttama, Jean Marc Ogier, and Pierre Loonis. Top-down segmentation of ancient graphical drop caps: lettrines. In *Proceedings of the International Workshop on Graphics Recognition*, pages 87–96, 2005.

[Xavier and Yoshua, 2010] Glorot Xavier and Bengio Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[Xie and Tu, 2015] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the 15th International Conference on Computer Vision*, pages 1395–1403, 2015.

[Yin and Liu, 2009] Fei Yin and Cheng-Lin Liu. Handwritten chinese text line segmentation by clustering with distance metric learning. In *Pattern Recognition*, pages 3146–3157, 2009.