Deep Learning Based Semantic Page Segmentation of Document Images in Chinese and English

Yajun Zou and Jinwen Ma(✉)

Department of Information and Computational Sciences, School of Mathematical Sciences and LMAM, Peking University, Beijing 100871, China
jwma@math.pku.edu.cn

Abstract. Semantic page segmentation of document images is a basic task for document layout analysis which is key to document reconstruction and digitalization. Previous work usually considers only a few semantic types in a page (e.g., text and non-text) and performs mainly on English document images and it is still challenging to make the finer semantic segmentation on Chinese and English document pages. In this paper, we propose a deep learning based method for semantic page segmentation in Chinese and English documents such that a document page can be decomposed into regions of four semantic types such as text, table, figure and formula. Specifically, a deep semantic segmentation neural network is designed to achieve the pixel-wise segmentation where each pixel of an input document page image is labeled as background or one of the four categories above. Then we can obtain the accurate locations of regions in different types by implementing the Connected Component Analysis algorithm on the prediction mask. Moreover, a Non-Intersecting Region Segmentation Algorithm is further designed to generate a series of regions which do not overlap each other, and thus improve the segmentation results and avoid possible location conflicts in the page reconstruction. For the training of the neural network, we manually annotate a dataset whose documents are from Chinese and English language sources and contain various layouts. The experimental results on our collected dataset demonstrate the superiority of our proposed method over the other existing methods. In addition, we utilize transfer learning on public POD dataset and obtain the promising results in comparison with the state-of-the-art methods.

Keywords: Semantic page segmentation · Document layout analysis · Document reconstruction · Deep learning

1 Introduction

With the rapid development of the Internet and digital equipment, there are a huge number of text documents in electronic version (e.g. camera captured page images) generated every day. So, document reconstruction and digitalization become particularly important. Actually, we need to convert those document pages into editable and searchable forms so that they can be further utilized in information extraction and retrieval.

D.-S. Huang et al. (Eds.): ICIC 2021, LNCS 12836, pp. 484–498, 2021.
https://doi.org/10.1007/978-3-030-84522-3_40

For the reconstruction and digitalization of an input text document image, it is effective to firstly segment the regions in different semantic types and then recognize the contents of the segmented regions by the type-related recognition systems. For example, text can be recognized by the OCR system. As a result, the page is reconstructed by assembling the recognized contents of the regions according to their location. Therefore, page segmentation is a crucial step in the document reconstruction workflow. Generally, page segmentation aims at segmenting a page into a set of homogenous regions which can be categorized into several semantic types, like tables and figures. As well known, there are text documents with various styles and layouts. For instance, a document page can be single-column or multi-column. And documents with different languages may contain different texture features with respect to the text types. In addition, there is a high similarity between different semantic types, e.g. table and figure. The grid chart has the same structure of intersecting horizontal and vertical ruling lines as the table. Moreover, regions of a specific type vary greatly in aspect ratios and scales among them. Therefore, it is rather challenging to make the page segmentation in multi-language document images effective and robust.

Most of the conventional document segmentation methods [1–3] consist of unsupervised segmentation and supervised classification. They usually make an assumption on page layouts and segment a page into a number of regions by certain heuristic rules for multiple cases. Then for region classification, they extract a group of hand-craft features and then employ machine learning algorithms to classify a segmented region into different types. In this way, they have high experience dependency that can't fit in diverse documents. Nevertheless, some of the deep learning based methods [3] adopt an end-to-end trainable convolutional network to automatically extract features for the better robustness. Besides, some of the deep learning based methods [4, 5] formulate this problem as a typical object detection in natural images. These methods take the document image as an input and then output the bounding boxes of objects with corresponding labels. Moreover, there are some methods based on deep semantic segmentation network where each pixel is classified into one semantic type [6–9]. The pixel level understanding is more precise than the bounding box level one. However, they usually consider only a few semantic types. For instance, most of them only distinguish text from non-text in a page or assume that no formula regions exist in a document. This is not sufficient for document reconstruction. Moreover, their experiments are typically performed on English documents.

In this paper, we propose a deep learning based method to achieve better semantic page segmentation. For the goal of document reconstruction, four semantic types are taken into consideration, i.e. text, table, figure, formula. For an input text document image, we firstly use a semantic segmentation neural network to classify each pixel as either background or one of the four categories above. Our network leverages context features and local features of a document image to get more precise segmentation results. Then, by implementing the Connected Component Analysis (CCA) algorithm on the prediction mask, we obtain the accurate locations of regions in different types. We further develop a simple Non-Intersecting Region Segmentation Algorithm (NIRSA) to improve the segmentation result and facilitate the future page regeneration task. Furthermore, to address the issue of lacking annotated training data, we manually annotate a dataset

consisting of Chinese and English documents that contain various styles and layouts. And we perform transfer leaning and domain adaption during our training procedures. Finally, we conduct the experiments on our collected dataset and public POD (Page Object Detection) dataset to demonstrate the effectiveness of our proposed method.

The rest of the paper is organized as follows. We firstly review the related work in Sect. 2. Our proposed method is then presented in Sect. 3. In Sect. 4, we summarize the experiment results and comparisons on a collected dataset and several public datasets. We finally make a brief conclusion in the last section.

2 Related Work

In recent years, there have been many methods for semantic page segmentation in document images. Most of the conventional methods [1–3] have two stages, i.e., unsupervised segmentation and supervised classification. The unsupervised segmentation stage is usually based on bottom-up or top-down structure. The bottom-up structure [1] starts to segment characters or lines and gradually groups them into homogenous regions. While the top-down structure [2] operates directly on the entire document and recursively segments the resulting regions. The greatest shortcoming of these methods is to decide a large amount of parameters by experience, which leads to poor robustness. During the classification procedure, hand-craft features of the segmented regions are firstly extracted and then fed into a classifier to determine the semantic labels.

Nowadays, the CNN based networks [3] are utilized to complete automatic feature extraction with better generalization ability. Currently, some methods formulate page segmentation as a typical object detection problem. They usually focus on a specific type, e.g. table region segmentation. DeepDeSRT [10] model adjusts the convolution kernel of the backbone in Faster R-CNN to detect the table regions. Prasad et al. [11] propose the Cascade Mask Region-based CNN High-Resolution Network that solves both problems of table detection and structure recognition simultaneously.

In addition, both PubLayNet [5] and GOD [4] use Faster R-CNN [12] and Mask R-CNN [13] to detect regions in different types. PubLayNet [5] considers five semantic types that can be applied to most documents. But it's not suitable for some statistical reports because formula type is excluded. GOD [4] only detects regions of three semantic types, leaving the text type out. However, text is the most common semantic type in documents. Cross-domain DOD model [14] is built on top of the Feature Pyramid Networks [15], which mainly addresses the domain shift problem that arises in the absence of labeled data.

There are also some methods based on semantic segmentation models. Yang et al. [9] first introduce semantic segmentation to page segmentation. But an additional tool is adopted to specify the segmentation boundary. Lee et al. [7] propose trainable multiplication layers (TMLs) and incorporate them into U-Net architecture [16] to gain better performance. But they only perform binary segmentation that only pixels in text type are identified. And they only complete pixel-wise segmentation task. DeepLayout [8] doesn't distinguish text type from background. They choose the DeepLab v2 structure [17] to segment these pixels that belong to table, figure and formula types. As a result, text regions can't be segmented during the subsequent post-processing procedure. He

et al. [6] train FCN [18] to segment three types of document elements: text, table, and figure. They use multi-scale training strategy to capture multi-scale information. Also, they add a contour detection branch to improve the results of semantic segmentation. However, they only segment table regions by an additional verification net without results of regions in other semantic types.

3 Methodology

As is shown in Fig. 1, our proposed method mainly consists of two parts. A semantic segmentation network begins to classify each pixel to a certain type. Then a series of regions that do not overlap each other are generated through Connected Component Analysis and Non-Intersecting Region Segmentation Algorithm. We introduce these two main steps.

Fig. 1. Illustration of our method. The results of both pixel-wise and region-wise segmentation are shown in the rightmost image, where each region is represented by a bounding box, corresponding label and confidence score (white: background, red: text, green: table, blue: figure, black: formula) (Color figure online).

3.1 Semantic Segmentation Network

In our framework, a deep semantic segmentation network is firstly utilized to assign a semantic label to each pixel. There are five categories including the background label. As shown by the observations in [6], unlike general semantic segmentation in natural images, a large receptive field is required in the semantic segmentation network for document images to guarantee sufficient context information. For example, the text block in a table can't be recognized as part of the table without a large context. However, there is an inherent conflict between context information and spatial information in the segmentation network. The acquisition of a large context weakens the details for region boundary prediction. So we alleviate this problem by aggregating multi-scale information as in [6]. An image pyramid model based on FCN is adopted in [6], where several images with different scales are all taken as input. Since it's obvious that the image pyramid model is time-consuming, we adopt several improved networks based

on Skip Connection or Atrous Spatial Pyramid Pooling (ASPP) to achieve multi-scale information fusion, e.g. U-Net [16], FPN [15], DeepLab series networks [17, 19, 20].

U-Net [16] is based on a typical encoder-decoder structure. The skip connections between low layers in encoder phase and high layers in decoder phase promote the fusion of low-level and high-level features. The low-level features contain abundant spatial information while the context information is included in the high-level features. In fact, features in different layers can be regarded as the corresponding features at different scales. FPN [15] shares the similar core idea, but the difference is that the prediction layer is added to every feature map during decoder process so as to enhance the supervision information at different scales. Instead of typical convolution, atrous convolution is used to enlarge receptive field and attain spatial information at the same time, which doesn't increase the number of parameters. DeepLab series networks use the Atrous Spatial Pyramid Pooling (ASPP) module to capture multi-scale information by concatenating these feature maps output from atrous convolution layers with different rates. Besides, ASPP module is augmented with image-level features to capture long range information. Moreover, a simple decoder module is included in DeepLab v3+ [20] to get more precise segmentation especially for region boundary.

In our experiments, DeepLab v3+ achieves better performance than other networks (such as U-Net, FPN) when they are trained on our collected dataset. At inference time, for each input document image, a prediction mask with five channels is output. That is, for pixel p_j, a normalized possibility vector $v_j = \left(v_j^0, v_j^1, v_j^2, v_j^3, v_j^4\right)$ is obtained. And its label l_j satisfies $l_j = \underset{k}{\operatorname{argmax}}\left\{v_j^k\right\}$.

3.2 Region Segmentation

CCA. To restore the definite region in different types, we extract the connected components of each category from the prediction mask respectively. Then each connected component with its corresponding label is regarded as a candidate region. And we take the rectangular bounding boxes of connected components to specify the boundary of regions. For the bounding box b_i with label c_i, its confidence score s_i is defined as follows.

$$s_i = \frac{1}{N_i} \sum\nolimits_{p_j \in b_i} v_j^{c_i} \tag{1}$$

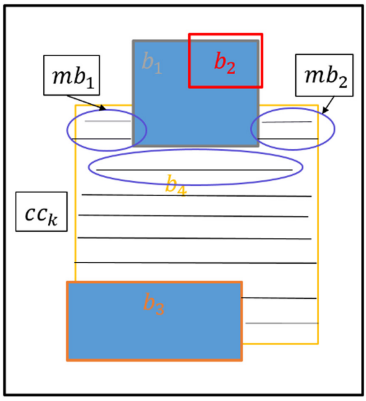
Here, N_i is the number of pixels in b_i.

NIRSA. As is shown in Fig. 3, there are some intersecting bounding boxes after CCA due to the error from semantic segmentation (unclear boundary). For image (a), text and figure regions are confined into one box. For image (c), two text boxes overlap each other and one of them contains incomplete word. Thus, a Non-Intersecting Region Segmentation Algorithm is proposed to obtain more precise page decomposition results. And it can also eliminate the position conflict that may appear in the document reconstruction workflow.

Our proposed algorithm is similar to the non-maximum suppression algorithm (NMS) in object detection task. Firstly, we sort the candidate bounding boxes by their corresponding confidence sore. The pipeline of our proposed algorithm is to generate the bounding box one by one on an empty page. As is shown in Fig. 2, b_1 that has the highest confidence score is first generated on the page. At the same time, we mark all pixels of its corresponding region on the page with a non-empty flag (blue fill). Each pixel of the page is marked with an empty flag at the beginning. Then for the next selected box, we consider three possible cases:

1. If the non-empty pixels' portion of its corresponding region on the page is below a certain percentage, we drop the box directly (b_2)
2. If pixels of its corresponding region on the page are all marked empty flag, we generate the box directly on the page (b_3).
3. Otherwise, we use several small boxes to approximate the empty part of its corresponding region (b_4).

We accomplish the case 3 by exploiting the local information. There are two operations performed: splitting and merging. The foreground area of the empty part is firstly identified by simple threshold method. Next, for each row, we perform horizontal run length smoothing algorithm (RLSA). Thus the empty area is split into a series of connected components (black lines inside b_4) with a height of 1. It should be noticed that for text region, we add an additional vertical RLSA to extract text lines. This prevent character from being destroyed. It's obvious that these connected components are non-intersecting. Then we scan the connected components from top to bottom and left to right and merge them into several boxes. Suppose we currently have two merged boxes mb_1, mb_2, for the connected component cc_k, there is a new box generated because it can't be merged to mb_1 or mb_2. In Fig. 3, we show the effectiveness of our proposed NIRSA.



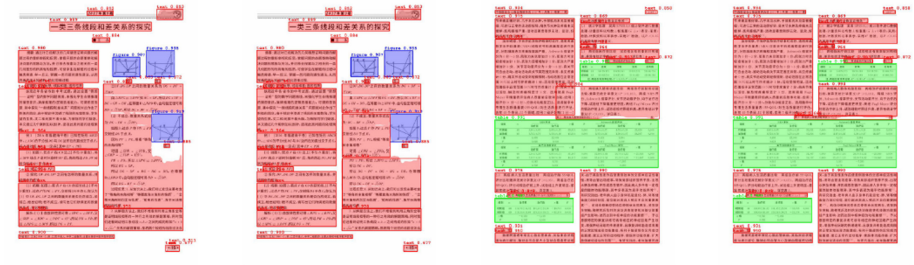
Algorithm 1 Non-Intersecting Region Segmentation Algorithm (NIRSA)

Input:
1. Bounding box set $B = \{b_1, b_2, \cdots, b_n\}$ where $s_1 > s_2 > \cdots > s_n$;
2. Page $P[:]$;
3. Pixel p;
4. Threshold th;

Output:
1. Non-intersecting bounding box set O;

Initialize $P[:] = 0, O = \{\}$;
$\forall p \in b_1$, set $P[p] = 1$;
delete b_1 from set B;
for b_i in B **do**
  if $\forall p \in b_i, P[p] = 0$ **then**
    add b_i into set O;
  else
    $emp = \frac{\text{NUM}(P(p)=0, p \in b_i)}{\text{NUM}(p \in b_i)}$
    if $emp < th$ **then**
      drop b_i;
    else
      split the empty part into multiple connected components $\{cc_k\}$;
      merge the connected components to several boxes $\{mb_j\}$;
      for each mb_j **do**
        add mb_j into O;
        $\forall p \in mb_j$, set $P[p] = 1$;

Fig. 2. The description of Non-Intersecting Region Segmentation Algorithm procedure.



(a) (b) (c) (d)

Fig. 3. Examples of the results of region segmentation before and after NIRSA. (a) and (c) represent the results after CCA. (b) and (d) represent the results after CCA and NIRSA.

4 Experiments

4.1 Datasets

POD [21]. This is a competition dataset consisting of 2,000 English document images, about 800 of which are used for testing and the rest of which are for training. These document images are extracted from scientific papers that have simple white background and vary in layout styles. There are regions with three semantic types: table, figure, formula. And Each region is presented as a rectangular bounding box and a corresponding label. Meanwhile, each subfigure is labelled as a separate region. Each formula line of multi-line formula is labelled in the similar way. In our experiments, we use the POD extend training dataset as in [8], which contains about 10,000 training images and all text line regions are appended in the ground truth.

Collected Dataset. To our knowledge, current dataset contains only document images in English, and most documents are from a certain range of fields thus are weak in diversity. Moreover, the semantic types are not comprehensive. For example, formula type is usually absent. Therefore, we annotate a collected dataset with a scale of 30,000 document images which are from a large search library. And Chinese is the major language of these documents. In addition, these documents are collected from scientific papers, magazines and statistical yearbooks, involving various fields of medical science, literature, education, natural science, etc. Thus the dataset is qualified for diversity in document layouts and contents.

As for the ground truth format, we should mark document images by pixel in our framework. However, it's not cost-effective. So we resort to the region-wise annotation in the same way as POD dataset. Four semantic types are taken into consideration: text, figure, table and formula. Different from the POD dataset, we regard paragraph as a region unit for text type. And the titles are also taken as independent regions. And for formula type, an entire formula region is labeled even for multi-line formula. It is worth noting that there is no overlap between the annotated bounding boxes. We split the intersecting area and label it with several rectangular boxes when encountering the inevitable intersection of rectangular bounding boxes, e.g. when a picture is surrounded by text. In this way, the ground truth mask for pixel-wise segmentation can be acquired,

i.e. the ground truth label of pixels in the bounding boxes are set to be the same as the label of the bounding box. Examples in our collected dataset are shown in Fig. 5.

4.2 Metrics

We use the IOU metric to evaluate the segmentation results that output from semantic segmentation networks. Based on the confusion matrix of pixel classification, the IOU metric computes the intersection over union on each category. And the mean IOU (mIOU) over all classes is also calculated to measure the overall performance.

While for region segmentation task, we refer to the POD competition evaluation tool [21] where F1 score and average precision (AP) metric are both adopted. Specifically, given a IoU threshold α, a segmented region b_i is regarded as a true positive if it satisfies:

$$\mathrm{IoU}(b_i, gt_j) = \frac{b_i \cap gt_j}{b_i \cup gt_j} > \alpha \tag{2}$$

Here, gt_j is the corresponding ground truth region. Then we can compute the precision and recall over regions. F1 score and AP are both comprehensive evaluation metric based on precision and recall. mAP and Avg. F1 calculate the mean AP and F1 over all semantic types, respectively.

In addition, we propose a page-wise evaluation method which is inspired by [22]. In order to facilitate future recognition, the segmented regions must be complete and pure. Therefore, a segmented box b_i is allowable when it contains a complete ground truth region and does not have overlap with the remaining ground truth regions. On the one hand, all b_i should be allowable. On the other hand, all gt_j should be segmented. In this case, the page segmentation is regarded as exactly correct. We count the percentage of the exact segmentation. Besides, similar to [22], we add an additional allowable case (Merging Text) when a segmented text box b_i contains multiple complete text ground truth boxes. That means multiple paragraph can be merged into one bounding box.

4.3 Experimental Results on Our Collected Dataset

We randomly select 1,000 images for testing, and the rest images are for training (about 29,000 images). We take DeepLab v3+ as our semantic segmentation network. During the training procedure, the input image is firstly random rescaled within a scale range from 0.5 to 1.5. Then an image patch with a size of 768×768 is randomly cropped from the rescaled images. The batch size is set to 4. Besides, we take the Stochastic Gradient Descent algorithm as our optimizer. And the initial learning rate is set to 0.001, which is decreased by a factor of 0.5 after 10 epochs. Moreover, we use transfer learning to accelerate the training process and enhance performance, especially in areas where there is a lack of adequate annotated data. So each network is pre-trained on ImageNet dataset and fine-tuned on our collected training set. At the inference time, we pad each input image to confirm that its width and height are both divisible by 32. The padding image is then fed into the trained network and further get the final region-wise segmentation results.

Comparison with other Methods. As we mentioned in Sect. 2, current related state-of-the-art methods [4, 5] are mainly based on Faster R-CNN and Mask R-CNN. As for parameter setting, we follow the values in [4] which achieves the best results in POD up to now. Our code inherits from Detectron2 as with [4, 5]. As with the official POD evaluation tool, the IoU threshold is set to 0.8. The region segmentation results are shown in Table 1. Among them, our results are bolded. "Ours" represents the results after semantic segmentation and CCA while "Ours + NIRSA" represents the results that are performed by CCA and NIRSA. Both of them achieve significant improvement over other methods, especially in text and formula category. "Ours + NIRSA" boosts the average F1 for 4.4% and mAP for 2.4% compared with the Mask R-CNN. This is probably because the aspect ratios of regions in these two categories vary a lot and usually extreme, which increase the difficulty of the object detection task. Among all categories, the overall performance on table category is the best while the segmentation of formula category is the hardest. This may be caused by the class-imbalance problem on the document page. For most pages, text blocks take the main part while formula blocks usually only appear in specific documents. In addition, we also label Chinese formulas which is composed of operators and Chinese characters. This makes it difficult to distinguish text from formula.

Table 1. The results of region segmentation on our collected dataset. The last two rows represent our results.

Methods	F1 score					AP				
	Text	Table	Figure	Formula	Avg.F1	Text	Table	Figure	Formula	mAP
Faster R-CNN	0.760	0.981	0.830	0.575	0.786	0.701	0.982	0.860	0.449	0.748
Mask R-CNN	0.774	0.975	0.828	0.624	0.800	0.718	0.980	0.860	0.531	0.771
Ours	0.837	0.991	0.823	0.692	0.836	0.796	0.996	0.781	0.629	**0.801**
Ours + NIRSA	0.841	0.997	0.825	0.714	**0.844**	0.795	0.996	0.788	0.602	0.795

The page-wise segmentation results are shown in Table 2. When merging multiple paragraphs into one region is allowable, about 658 pages out of 1000 pages are correctly segmented with the "Ours + NIRSA" method. It's meaningful to realize the automatic semantic segmentation of pages. Through analysis, semantic segmentation network has richer detail information than Faster R-CNN and Mask R-CNN. So we can restore more precise location information with our method, as Fig. 4 shows. But Faster R-CNN and Mask R-CNN are better at distinguishing different instances. This is also indicated by the differences between the results with "Merging Text" and without "Merging Text". For text regions, our method is likely to merge two paragraph into one region. When "Merging Text" is allowable, the percentage of exact segmentation increased by 30%.

Table 2. The percentage of exact segmentation (page-wise) on our collected dataset.

Methods	Merging text	
	✓	✗
Faster R-CNN	0.132	0.151
Mask R-CNN	0.154	0.172
Ours	0.340	0.602
Ours + NIRSA	**0.358**	**0.653**

(a) Faster R-CNN (b) Mask R-CNN (c) Ours

Fig. 4. Comparison between different methods. Each row is an example. Our results are showed in the rightmost. First line: incomplete text region. Second line: split table region.

Moreover, our proposed NIRSA also boosts the performance. It can make up for the error of semantic segmentation network and improve the whole page segmentation.

Discussions. We adopt some mainstream semantic segmentation networks and compare their performance on our task. As is shown in Table 3, all of them can produce promising results. Based on the mean IOU metric, DeepLab v3+ achieves the best segmentation results while FPN is closely behind. DeepLab v3+ makes a significant improvement on the formula category in comparison with other architectures. Here, we contain the DeepLab v3 architecture in our contrast experiments, which lacks the decoder structure compared with the DeepLab v3+ architecture. It can be observed that the decoder structure is effective because it can raise the segmentation results especially for formula and text categories. Besides, atrous convolution in DeepLab series can increase the receptive

field and retain the spatial information simultaneously, i.e., it doesn't reduce the size of the output feature map. Thus it leads to a high time consumption. To make a balance between time complexity and accuracy, we set the output stride of DeepLab v3+ to 16. The segmentation of formula categories is harder than other categories as the IOU of formula type is lower by a great margin. In fact, we try to tackle the class-imbalance problem by adopting some effective loss functions, e.g. focal loss. But it doesn't bring a significant improvement. It requires further investigation. Moreover, we explore the impact of IoU threshold on our results of the region segmentation as [4] does. A high IoU threshold requests that a true positive should have high overlap with the ground truth. As Table 4 shows, with the increase of IoU threshold, the performance of our method does not decrease sharply. In fact, our method on the table category is robust in contrast to text and formula categories. Figure 5 demonstrates some visualization results on collected dataset with our method.

Table 3. The IOU of different semantic segmentation architectures on our collected dataset.

Networks	Background	Text	Table	Figure	Formula	mIOU
U-Net	0.936	0.949	0.964	0.874	0.780	0.901
FPN	0.951	0.961	0.983	0.901	0.823	0.924
DeepLab v3	0.939	0.948	0.978	0.899	0.801	0.913
DeepLab v3+	**0.951**	**0.961**	**0.983**	**0.905**	**0.846**	**0.929**

Table 4. The results of region segmentation with different IoU threshold.

IoU	F1 score					AP				
	Text	Table	Figure	Formula	Avg.F1	Text	Table	Figure	Formula	mAP
0.5	0.926	0.997	0.861	0.811	0.899	0.892	0.996	0.839	0.715	0.860
0.6	0.913	0.997	0.847	0.789	0.887	0.876	0.996	0.811	0.692	0.844
0.7	0.894	0.997	0.837	0.760	0.872	0.854	0.996	0.801	0.659	0.827
0.8	0.841	0.997	0.825	0.714	0.844	0.795	0.996	0.788	0.602	0.795

4.4 Experimental Results on Public POD Dataset

We test the performance on POD competition dataset by adopting different training strategies to illustrate the effect of transfer learning. In Table 5, "POD" represents the results that training on POD dataset. "Collected" represents the results that training on our collected dataset. "Collected + POD" represents the results that pre-training on our collected dataset and fine-tuning on POD dataset, which achieves the best performance. The mean IOU has increased by up to 5% than the model without collected dataset

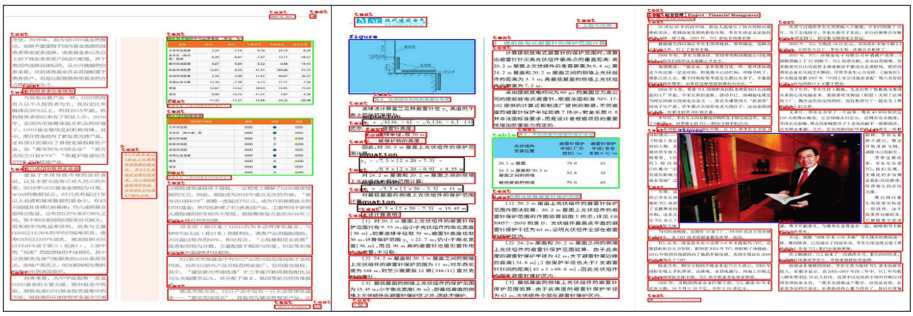


Fig. 5. Three examples of the results of our method on the collected dataset.

pre-trained. This proves that effectiveness of transfer learning. In addition, "Collected" is worse than "POD". Especially for text and formula type, there is a huge decline. This demonstrates the difference between POD data and our collected data. As we introduced above, our collected dataset has a different ground truth format from the POD dataset in these two semantic types. So domain adaption is necessary to enhance the performance.

Furthermore, in Table 6, we show the comparison between our method and other state-of-the-art methods for region segmentation. The first three lines are the results in article papers. The middle four lines are the top results of competition participants. As for formula category, our method achieves the best AP value. However, we have a poor performance at figure category compared to the top results. As Fig. 6 shows, our method tends to split a figure into several figures (b) or merge several subfigures (c). And the segmented regions in table and figure categories by our method usually contain the corresponding caption parts (a). So there is still a certain gap between our result and the top results. However, these methods are all concentrated on the segmentation of table, figure and formula regions. It's not sufficient for some real applications like page construction. The segmentation of text regions is ignored in other methods while our method can additionally achieve 0.931 F1 score and 0.911 AP for text category. Overall,

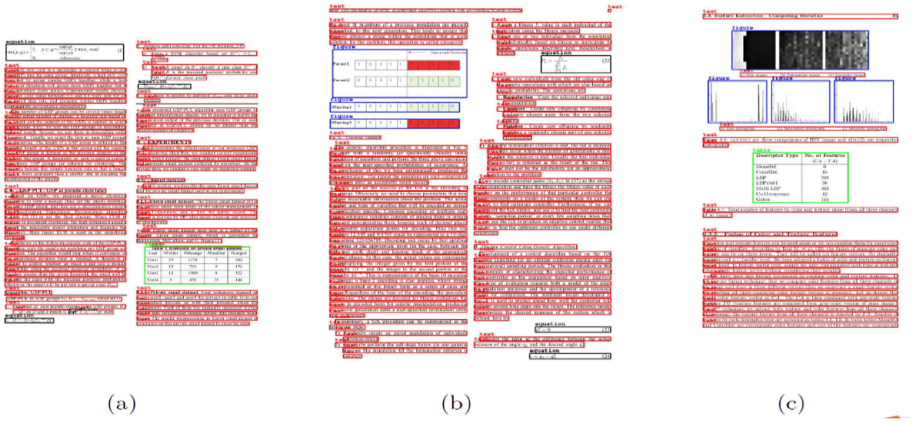


Fig. 6. Failure cases on POD dataset.

our method comes to a good place. And compared with other methods, our method can still get a promising result in multi-language documents images.

Table 5. The pixel segmentation results of different training strategies on POD dataset.

Training	Background	Text	Table	Figure	Formula	mIOU
POD	0.945	0.916	0.899	0.761	0.875	0.879
Collected	0.852	0.687	0.819	0.862	0.642	0.772
Collected + POD	**0.968**	**0.933**	**0.966**	**0.897**	**0.912**	**0.935**

Table 6. Comparison of our proposed method with the state-of-the-art methods. It should be noted that our method can additionally achieve 0.931 F1 score and 0.911 AP for text type.

Methods	F1 score				AP			
	Formula	Table	Figure	Avg.F1	Formula	Table	Figure	mAP
Li [23]	**0.932**	0.959	**0.917**	**0.936**	0.863	0.923	**0.854**	0.880
GOD [4]	0.919	**0.968**	0.912	0.933	0.869	**0.974**	0.818	**0.887**
DeepLayout [8]	0.716	0.911	0.776	0.801	0.506	0.893	0.672	0.690
NLPR-PAL [21]	0.902	0.951	0.898	0.917	0.816	0.911	0.805	0.844
icstpku [21]	0.841	0.763	0.708	0.770	0.815	0.697	0.597	0.703
FastDetector [21]	0.636	0.896	0.616	0.717	0.427	0.884	0.365	0.559
Vislnt [21]	0.241	0.826	0.643	0.570	0.117	0.795	0.565	0.492
Ours	0.923	0.914	0.812	0.883	**0.910**	0.944	0.731	0.862

5 Conclusion

We have established a deep learning based method to achieve better semantic page segmentation in Chinese and English document images. We use the DeepLab v3+ architecture which can capture multi-scale information to get precise pixel-wise classification results. Then we can get a series of candidate regions in different categories by making CCA on the prediction mask. And a Non-Intersecting Region Segmentation Algorithm is developed to solve the problem of intersection between regions, which boosts the performance and facilitates the document reconstruction applications. The promising results are both obtained on our collected dataset and public POD dataset. In the future, we plan to extend our framework to include more categories, like table or figure captions. Besides, we consider a fine-grained annotation format rather than just annotation of rectangular boxes.

Acknowledgment. This work was supported by the Natural Science Foundation of China under the grant 62071171.

References

1. Cesarini, F., Lastri, M., Marinai, S., Soda, G.: Encoding of modified XY trees for document classification. In: Proceedings of 6th International Conference on Document Analysis and Recognition, pp. 1131–1136. IEEE (2001)
2. Chen, K., Yin, F., Liu, C.L.: Hybrid page segmentation with efficient whitespace rectangles extraction and grouping. In: 12th International Conference on Document Analysis and Recognition, pp. 958–962. IEEE (2013)
3. Yi, X., Gao, L., Liao, Y., Zhang, X., Liu, R., Jiang, Z.: CNN based page object detection in document images. In: 14th IAPR International Conference on Document Analysis and Recognition, vol. 1, pp. 230–235. IEEE (2017)
4. Saha, R., Mondal, A., Jawahar, C.V.: Graphical object detection in document images. In: 15th International Conference on Document Analysis and Recognition, pp. 51–58. IEEE (2019)
5. Zhong, X., Tang, J., Yepes, A.J.: PubLayNet: largest dataset ever for document layout analysis. In: 15th International Conference on Document Analysis and Recognition, pp. 1015–1022. IEEE (2019)
6. He, D., Cohen, S., Price, B., Kifer, D., Giles, C.L.: Multi-scale multi-task FCN for semantic page segmentation and table detection. In: 14th IAPR International Conference on Document Analysis and Recognition, vol. 1, pp. 254–261. IEEE (2017)
7. Lee, J., Hayashi, H., Ohyama, W., Uchida, S.: Page segmentation using a convolutional neural network with trainable co-occurrence features. In: 15th International Conference on Document Analysis and Recognition, pp. 1023–1028. IEEE (2019)
8. Li, Y., Zou, Y., Ma, J.: DeepLayout: a semantic segmentation approach to page layout analysis. In: De-Shuang Huang, M., Gromiha, M., Han, K., Hussain, A. (eds.) Intelligent Computing Methodologies, pp. 266–277. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95957-3_30
9. Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Lee Giles, C.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5315–5324. IEEE (2017)
10. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: DeepDeSRT: deep learning for detection and structure recognition of tables in document images. In: 14th IAPR International Conference on Document Analysis and Recognition, vol. 1, pp. 1162–1167. IEEE (2017)
11. Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: CascadeTabNet: an approach for end to end table detection and structure recognition from image-based documents. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 572–573. IEEE (2020)
12. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969. IEEE (2017)
14. Li, K., et al.: Cross-domain document object detection: benchmark suite and method. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12915–12924. IEEE (2020)
15. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125. IEEE (2017)

16. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
17. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. **40**(4), 834–848 (2017)
18. 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. IEEE (2015)
19. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.
20. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 833–851. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_49
21. Gao, L., Yi, X., Jiang, Z., Hao, L., Tang, Z.: ICDAR2017 competition on page object detection. In: 14th IAPR International Conference on Document Analysis and Recognition, vol. 1, pp. 1417–1422. IEEE (2017)
22. Antonacopoulos, A., Bridson, D.: Performance analysis framework for layout analysis methods. In: 9th International Conference on Document Analysis and Recognition, vol. 2, pp. 1258–1262. IEEE (2007)
23. Li, X.H., Yin, F., Liu, C.L.: Page object detection from pdf document images by deep structured prediction and supervised clustering. In: 24th International Conference on Pattern Recognition, pp. 3627–3632. IEEE (2018)