IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 21, 2021, accepted October 10, 2021, date of publication October 21, 2021, date of current version October 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3121152

A Page Object Detection Method Based on Mask R-CNN

CANHUI XU[1,2], CAO SHI[1], HENGYUE BI[1], CHUANQI LIU[1], YONGFENG YUAN[3], HAOYAN GUO[3], AND YINONG CHEN[2]

[1]School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China
[2]School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287, USA
[3]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China

Corresponding author: Cao Shi (caoshi@yeah.net)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806107 and Grant 61702135, in part by the Shandong Key Laboratory of Wisdom Mine Information Technology, and in part by the Opening Project of State Key Laboratory of Digital Publishing Technology.

ABSTRACT Page object detection is crucial for document understanding. Different granularities for objects can result in different performances. In this study, block level region object detection is considered among the inherent hierarchical structure for document images. Inspired by Mask R-CNN (Region-based Convolutional Neural Networks) method, an end to end network is proposed to perform object classification, bounding box identification, and page object mask generation at the same time. Latex based synthetic document generation is designed for enlarging the training data. A large number of synthetic page images are generated for training to alleviate the insufficient dataset problem. Compared with existing page object competition methods, the proposed method achieves better results, with mAP of 0.917 on page objects such as table, figure and maths detection.

INDEX TERMS Page object detection, document images, deep learning, convolutional neural networks.

I. INTRODUCTION

Document image processing technology has become an important technology for machine understanding and artificial intelligence (AI) tasks. The sustainable development of document image processing technology helps AI algorithms and robots to obtain relevant image information, which contains human intellectual labor (documents). Generally, the pipeline of document image processing consists of three steps: pre-processing (binarization, noise/blur removal, rectification, etc.), page layout analysis (detection, identifying Regions of Interest, RoI) and logic understanding (gaining application-specific information from each RoI). Therefore, document image processing technology play a vital role in machine understanding and AI tasks.

In machine understanding, there are various application scenarios like information retrieval and mobile reading, which are based on page object extraction from document images. Traditionally, there are two major successive parts, layout analysis and logical understanding, taking part in the process. Layout analysis aims to detect and segment document page geometrically into regions, and subsequently logical understanding is to classify the segmented regions semantically into like tables, figures, formulae, text, and other page parts. However, recognition performance highly depends on the front-end layout segmentation results. A possible error in first segmenting stage tends to accumulate the misclassification in second recognition stage.

Recently, deep learning has become the most popular solution to object detection as well as semantic segmentation in natural scene images. It is made possible to detect, segment and classify objects in an end-to-end manner for image processing. Two kinds of state-of-the-art methods are known as one-stage detector and two-stage detector. One-stage detector treats detection as a regression task, such as SSD (Singe Shot MultiBox Detector) [1], YOLOv4 (You Only Look Once v4) [2], and YOLACT++ [3], etc. And two-stage methods using two steps: region proposal and classification/regression, like R-CNN (Region-based Convolutional Neural Networks) [4], Fast R-CNN [5], Faster R-CNN [6], etc. There are also attempts in applying deep learning based methods to document images. Some are designed for end to end pixel level analysis, while others aim to detect and classify regions with bounding boxes. Take FCN

The associate editor coordinating the review of this manuscript and approving it for publication was Yongqiang Cheng.

(Fully Connected Networks) as an example, it is able to simultaneously segment and classify document images at pixel level [7], [8]. It first extracts feature maps from convolutional neural network, and then deconvolution is performed to obtain full resolution semantic segmentation results. Another category is based on region proposal scheme, where the results are presented in regional bounding boxes. R-CNN [4] is popular among the region proposal methods. Various Faster R-CNN based methods [5] have been competed in page object recognition competition to detect the object bounding boxes [9]. In Faster R-CNN [6], the shared features are extracted from backbone network. The RPN (Region Proposed Network) is designed for producing candidate regions of interests, which made the inference faster than general R-CNN. Faster R-CNN can output both class labels and a bounding box offset for each candidate object.

With further advancements, Mask R-CNN [10] adds mask branch output on previous Faster R-CNN basis. Features are first extracted by backbone network, and the proposals are predicted and further refined to regress the bounding boxes for object detection and produce segmentation masks. The mask provides pixel-level semantic segmentation for each candidate object. Thus, Mask R-CNN achieves instance segmentation which involves both object detection and semantic segmentation. It integrates the improvements on both Faster R-CNN and FCN (Fully Connected Network). It applies RoI Align so as to preserve spatial orientation of features without losing the information when downsampling. As a two-stage object recognition method, it has become increasingly popular for various applications.

Inspired by previous works, in this paper, we utilized Mask R-CNN architecture on document image page object detection. A ResNet101 backbone with Feature Pyramid Network was trained for document images. Mask R-CNN performs FCN only for region of interest predicted, from which segmentation masks are produced. Both bounding box level recognition and pixel level classification are available. Insufficient data for training is one of the problems for page object recognition. In this paper, we generate large number of synthetic data for this usage. On six different datasets, including POD2017 (Page Object Detection) dataset, various experiments are designed to evaluate our method. Compared with previous page object detection methods, the proposed method achieves better AP results on page objects like table, figure and maths detection.

Our method extends the general framework Mask R-CNN to document image processing. It can achieve simultaneously page layout analysis and logical understanding. The output of the method comprises three parts: bounding box, mask and classification, as shown in Fig. 1, which represent RoI as layout analysis output (bounding box, mask) and logical understanding (classification), respectively. Aspect ratio of page objects is analyzed for RPN, and synthetic page images are generated for training. And experiments show that our method designed for document image leads to better performance than network designed for image in natural scene in page object detection tasks.

The rest of the paper is organized as follows. Related work on image document analysis is introduced in Section 2. The adapted network architecture is proposed in Section 3. Experimental results are presented at Section 4. The conclusions are given in Section 5.

II. RELATED WORK

Document image understanding is an important application of artificial intelligence and machine understanding. However, before realizing machine understanding, different types of research work are involved, including: binarization [11], rectification [12], optical character recognition (OCR) [13], [14], page layout analysis which includes page segmentation [15], text Line segmentation [16], and character segmentation [17], etc. And then, logic understanding is implemented, such as document classification [18], font recognition [19] and graphics recognition [20], etc.

Page layout analysis aims at detecting and segmenting text from non-text, followed by layout logic understanding so as to accomplish the task of recognizing logical classes like paragraph, figure, and table, etc. Traditionally, layout analysis and logical understanding are two major separated successive parts. For page segmentation, bottom-up and top-bottom methods are major approaches [21]–[24]. As for logic understanding, various classifier were utilized for classification such as Support Vector Machine [25], [26], and CRF [27]–[29].

To detect, segment and classify objects in an end-to-end manner, deep learning has been used as a basic method for object detection and semantic segmentation. Convolutional Neural Network (CNN) is powerful in representing hierarchal features. Deep networks are able to naturally integrate low/mid/high-level features and classifiers. A large number of methods have been proposed. CNN based networks have been applied on document classification and recognition, such as MobileNetV2 [30], dilated convolutional network [7]. Improvement for document image detection, segmentation and classification has been made with CNN [31]. It was claimed that various CNN architectures, including VGGNet, ResNet, GoogLeNet, DeconvNet, etc., were the most frequently used in image document processing [32].

Recently, ICDAR (International Conference on Document Analysis and Recognition) held Page Object Detection (POD) competition, focusing on detecting tables, mathematical equations, and figures. In POD competition of 2017 ICDAR conference, almost all the participated teams used deep learning for object detection, including popular SSD (Single Shot MultiBox Detector), Faster R-CNN based models [9]. It was also stated that there was possible improvement upon detection precision besides using Faster R-CNN. Multi-models also contributed to better performance of deep networks, with the help of extra information from OCRs, or CRF unary and binary features.

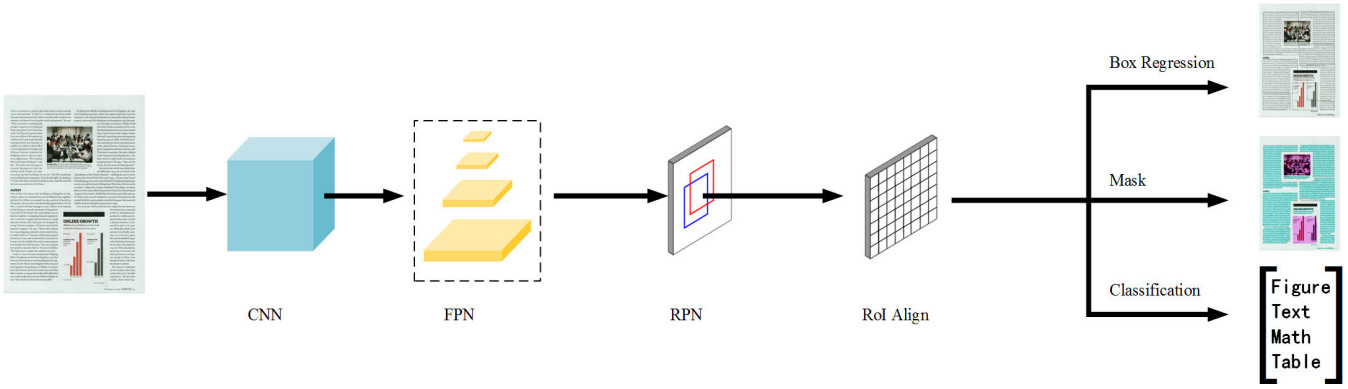


FIGURE 1. The framework of page object detection network based on Mask R-CNN.

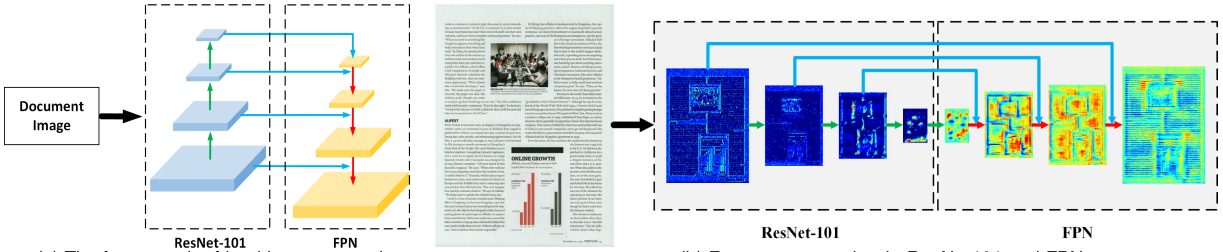


FIGURE 2. The detail of backbone network. ResNet-101 is utilized for CNN in Fig. 1.

Despite the Faster R-CNN based method to detect the outside bounding boxes for document page object, there is another way to predict class labels in pixel-wise level. Fully convolutional network (FCN) was utilized for semantic segmentation [33]. Based on coarse feature map extracted from CNN, deconvolution was performed to obtain full resolution segmentation mask. Dilated Residual Network (DRN) [34] replaced subsampling layers by adding dilation, which could be applied for page document semantic segmentation.

By adding FCN on proposed region candidates, Mask R-CNN added mask branch output on Faster R-CNN. The mask provided pixel-level semantic segmentation for each candidate object [10]. In this paper, our model utilizes Mask R-CNN as basic network architecture on document image page object detection and recognition. A ResNet101 backbone with Feature Pyramid Network is trained for document images. Mask R-CNN performs FCN only for region of interest predicted, from which segmentation masks are produced. Both bounding box level recognition and pixel level classification are produced. Our mask can detect and segment the region as well as semantically label the region.

In the existing studies on page object detection and recognition, high level representation of document remains an open challenging problem. The granularity is crucial for performance. In this paper, the block level is of our consideration. Four page object classes include text, figure, table, and maths. Small objects like maths are referred as isolated formulae. The embedded maths among textlines is considered as text block region in ground truth. Tables have various types, among which some have three lines, and others have non lines. For tables and figures, captions are regarded as text block as well for better performance. To alleviate the problem of limited document ground truth data for deep network training, we generate synthetic document images to enlarge the dataset. Experiments on six datasets are implemented to evaluate the Mask R-CNN based page object detection method.

III. PROPOSED METHOD

A. NETWROK ARCHITECTURE

Inspired by a general Mask R-CNN for object detection and segmentation, the object shape masks are better contours for object than bounding boxes, while semantic segmentation is better for depicting region of interest. In this paper, Mask R-CNN is adapted for page object recognition for document images. As is shown in Fig. 1, the overall framework consists of several parts: a Convolutional Neural Network (CNN) backbone with Feature Pyramid Network (FPN) [35], a Region Proposal Network (RPN) [6], RoI (Region of Interest) features extraction using RoI align, bounding box regression, label classification and mask prediction. Fig. 2 illustrates ResNet-101 [36] is utilized for the CNN in Fig. 1. As seen in Fig. 2(a), there is a bottom-up path in ResNet-101, along which resolution of feature image is reduced. In contrast to ResNet-101, FPN is a top-down process, in which resolution of feature image increases.

Lateral connections between ResNet-101 and FPN combines features with the same resolution from ResNet-101 and FPN respectively, to generate new features in FPN [10]. Fig. 2(b) shows features extraction process, in which resolution reduction pathway in ResNet101 and four features are demonstrated. Whereas, FPN has a resolution increase pathway. Two features with the same resolution from ResNet101 and FPN respectively are combined to generate a new feature along the pathway in FPN.

A ResNet101 backbone with FPN was utilized to train the model for document images. FPN is a top-down feature pyramid architecture for detecting objects at multi-scale level. Four last residual blocks $\{C_2, C_3, C_4, C_5\}$ are used as feature outputs. The lateral connections have enhanced semantically feature maps. FPN outputs final set of feature maps $(P_2, P_3, P_4, P_5, P_6)$.

Based on the feature maps extracted by the backbone network, appropriate proposals for page objects need to be generated. Region Proposal Network (RPN) is adapted to document page objects. In this work, proposals for page objects including figure, table, maths, and text are in block level instead of fragment level. Fig. 3 shows that the aspect ratio distribution vary according to different page objects, including text, table, figure, and maths regions in this scenario. Aspect ratio is measured by the proportion of width and height. The ratio of tables and figures shape varies mostly within 10. While text and maths region have larger values, even reaching to 70. The multiple scales require multiple feature representations and multiple scale region proposal candidates. FPN outputs five stages $(P_2, P_3, P_4, P_5, P_6)$, and anchors are set to (0.5, 1, 2, 3) for each stage. For example, a 1024×1024 document image is fed into the FPN network. From stage P2 to stage P6, there are 256×256, 128×128, 64×64, 32×32, 16×16 five types of feature maps are generated. Document images, being different from natural scene images, have their own data portraits. Page objects vary in multiple scales. Object aspect ratio distribution shows that

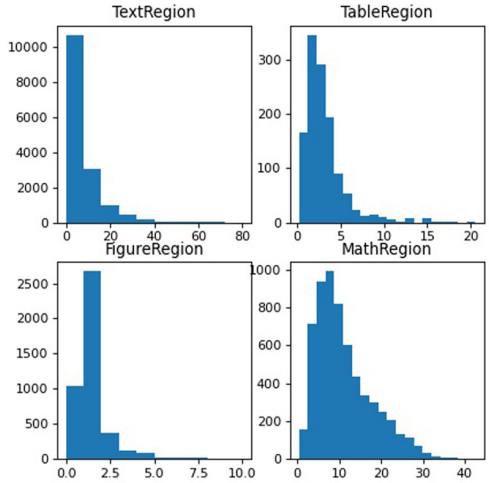


FIGURE 3. The aspect ratio distribution for document page objects.

figures and tables belong to large objects, while compared with small maths objects. Text blocks occupied majority of the distribution. FPN is suitable to extract features for page objects with various scales. A small object appears only in a small area in final feature maps.

After RPN, RoI Align [10] is used to extract accurate features in proposed method. RoI is positive when it has IoU (Intersection over Union) with ground-truth box of at least 0.5. Otherwise, it is considered a negative RoI. All the anchor boxes over the image can be classified as positive or negative according to the object score. Positive, negative, and neutral ratio is 1:1:1. Sampled RoIs is 2000 for FPN backbone.

For bounding boxes regression and classification, Faster R-CNN extracts RoI features from each level of FPN feature maps, and it provides candidate boxes. Positive anchors do not necessarily cover the whole object. RPN regresses a refinement to the anchors in order to correct the object boundaries. For a given positive proposal, to obtain the best matched horizontal rectangle, the matched boxes are shifted and resized to align with the proposal and target map.

For Mask branch, a fully convolutional network is used to produce the region segmentation maps and to make predictions. Mask target is the intersection between a RoI and its associated ground-truth mask. In our scenario, a common four-page object mask maps with size 32×128 and a background map can be predicted.

B. EVALUATION

To accomplish the goals of object detection, object recognition, semantic segmentation, and the multi-task loss include classification loss $Loss_{cls}$, bounding-box loss $Loss_{bbox}$, and mask loss $Loss_{mask}$. The mask branch produces $3 \times m \times m$ dimensional output for each RoI, after sigmoid function, and the loss function of mask branch applied average binary cross-entropy. It is claimed that the binary cross-entropy is better than multinomial cross-entropy loss [10]:

$$Loss_{mask} = -\sum_{i=1}^{n} \hat{y}_i \log \hat{y}_i + (1 - \hat{y}_i) \log(1 - \hat{y}_i) \quad (1)$$

The loss function of classification branch applied to cross-entropy:

$$Loss_{cls} = -\sum_{i=1}^{n} \hat{y}_i \log \hat{y}_i \quad (2)$$

The loss function of bounding-box applied Smooth L_1 Loss:

$$Loss_{bbox} = \begin{cases} 0.5x^2 & if \ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \quad (3)$$

C. SYNTHETIC DATA GENERATION

Insufficient training data may cause the network to overfit the data. To alleviate the problem of limited document ground truth data for deep network training, we selected various datasets used in existing studies and enriched our dataset with synthetic document. There also existed several attempts in

generating synthetic documents. Yang *et al.* produced document images by scrapping data from internet and applying Latex [7]. Yi *et al.* used semi-automatic method to label ground truth [31]. In this paper, a large number of synthetic data is generated to enrich the training data.

The predefined parameters for layout generation include font size, font color, page size, line space, margin space, figure size, page number, and total region number, etc. Given possible page objects set $DO_i, i = 1, 2, 3, \ldots, N$, the corresponding class labels are defined as $y_i, i = 1, 2, 3, \ldots, N$, where $y_i \in \{\text{Type}_j | \text{j} = 1, 2, 3, \ldots, \text{M}\}$. Here, Type_j represents: text, figure, maths, table objects in this dataset. For each Type_j, the data source is denoted as $Set_j, j = 1, 2, 3, \ldots, \text{M}$, which can be crawled from internet or parsed from PDF pages. The generation applied top down method, from overall layout to page objects. The generation process is summarized as follows:

- Generate header;
- Set single column, double columns or multiple columns;
- Starting from first column, according to y_i, generate random page objects DO_i and record the spatial coordinates $DO_i - Coors$ and its specific contents $DO_i - Content$;
- Generate page objects randomly till there is no space left in last column;
- Generate foot and page number. It is unnecessary to have all the page objects appearing in one page. y_i will decide whether there is no header, foot, page number, or other page objects. It is also allowed to configure certain page object included in the page;
- Use TeX mark language to generate the code for target PDF document page, which can be exported as a document image at the same time.

The data source Set_j for each Type_j can either comes from the internet crawling data or the block data exported from eligible PDF parser.

IV. EXPERIMENTAL RESULTS

A. DATASETS FOR TRAINING

To explore the performance of our network architecture on document page object detection, we investigated six datasets for training: DSSE-200 [7], POD2017 [9], RDCL2019 [37], Marmot [28], Doc2020 and SynDoc. The following list elaborated these datasets.

- DSSE-200 provides 200 labeled document images, which were used in Yang's work [7]. This dataset originally has 6 classes, within which text, section, caption and list are aggregated into text blocks in our work. Hence, there are 3 classes including text, figure and table blocks involved in training and testing.
- POD2017 dataset has total 2417 document images selected from CiteSeer scientific papers, including 3 manually labeled classes: table, figure, and maths. This dataset has a variety of page layouts, including single-column, double-column and multi-column scientific papers. This dataset is from page object detection (POD) competition in 2017 ICDAR conference. There are 9422 objects in total, with around 58% formulas, 31% figures and 11% tables.
- RDCL2019 has total 478 images from scanned magazines and technical articles. It was provided for recognition of documents with complex layouts. In this paper, 3 classes are aggregated, including Text, Table, and Figure.
- Marmot selected from 35 English and Chinese books has 244 image pages which were also used in our previous work 28]. It can be accessed through http://www.icst.pku.edu.cn/cpdp/sjzy. Our ground-truthing tool based on wxpython was able to label the document images at a given granularity. In this paper, we mark the document pages at block-level. A set of 3 classes includes text, figure, table and maths.
- Doc2020 has 195 document images manually labeled from scientific paper. As previous datasets, 4 classes include text, figure, table and maths.
- SynDoc document images are generated automatically by applying Latex. 3 classes including text, figure, and table are used for training and testing.

These date set is summarized in Table 1.

TABLE 1. Six datasets for training.

Dataset	Text	Table	Figure	Maths	Blocks	Pages
DSSE	2182	79	285	0	2546	200
POD2017	0	1020	2955	5447	9422	2417
RDCL2019	8271	48	639	34	8992	478
Marmot	1753	38	239	166	2196	244
Doc2020	3358	46	307	423	4134	195
SynDoc	13050	3712	1849	0	18611	1803
Total	28614	4943	6274	6070	45901	5337

In total, there are 5337 document images with 45901 blocks. As is expected in most documents, text block class dominates with 62% blocks. And there are around 14% figures, 13% maths, and 11% tables. The data distribution among these 4 classes is not exactly balanced. All the document images for training and testing are divided approximately with a ratio of 4:1.

In most cases, our labeling tool uses an open source tool called VIA (VGG Image Annotator) [38]. Rectangular boxes and semantic classes are marked. The ground truth is stored in json format. Our self-developed labeling tool is called Marmot [39], which is able to label the classes hierarchically. The ideal solution should be hierarchical, which includes block level, fragment level, and the relationship between different granularity regions. The hierarchical bounding boxes can be stored in a tree structure. Our own data are produced with assistance of a self-developed ground-truthing tool called CLAW. Ideally, the layout for pages should be a hierarchical structure, built upon different level of granularity. In this work, block level is our consideration.

TABLE 2. Comparison of our method with the state-of-the-art on ICDAR2017 POD dataset [9].

Method	AP (IoU = 0.6)				AP (IoU = 0.8)			
	Maths	Table	Figure	mAP	Maths	Table	Figure	mAP
NLPR-PAL	0.839	0.933	0.849	0.874	0.816	0.911	0.805	0.844
icstpku	0.849	0.753	0.679	0.760	0.815	0.697	0.597	0.703
FastDetectors	0.474	0.925	0.392	0.597	0.427	0.884	0.365	0.559
Vislnt	0.524	0.914	0.781	0.740	0.117	0.795	0.565	0.492
SOS	0.537	0.931	0.785	0.751	0.109	0.737	0.518	0.455
UITVN	0.193	0.924	0.786	0.634	0.061	0.695	0.554	0.437
Matiai-ee	0.116	0.781	0.325	0.407	0.005	0.626	0.134	0.255
HustVision	0.854	0.938	0.853	0.882	0.293	0.796	0.656	0.582
Li et al. [40]	0.878	0.946	**0.896**	0.907	0.863	**0.923**	**0.854**	**0.880**
YOLACT++ [3]	0.567	0.940	0.832	0.780	0.185	0.913	0.793	0.630
ours	**0.947**	**0.959**	0.845	**0.917**	**0.901**	0.917	0.813	0.877

The previous PRImA dataset used PAGE format for 2019 ICDAR competition on recognition of documents with complex layout [37]. Within their dataset, non-rectangular shaped regions are annotated. For mask ground truth, RDCL2019 dataset utilized non-rectangular shapes. For other datasets, rectangular bounding boxes mark the ground truth. Generally, regions are defined as rectangular areas. The ground truth is stored in XML format. An XML based ground-truth data format is designed, which can be transformed into COCO format for unified interface.

B. IMPLEMENTATION DETAILS

Our network architecture is implemented in pytorch. All the input images are scaled into 800 pixels on short edge. Popular pretrained models on natural images are not suitable for document images. Therefore, the convolutional network is trained from scratch with random initialization.

We trained on 8 GPUs for 36000 iterations for 108 epochs, with 2 images per GPU. The learning rate is 0.005, with weight decay 0.0001, momentum 0.9. Adam optimization is used. RPN has 5 scales and 4 aspect ratios, where the 5 scales are (8, 16, 32, 64, 128) and the 4 aspect ratios are (0.5, 1, 2, 3). RoI threshold considered positive is 0.5. The ratio of positive to negative RoI is 0.25. Each image has 2000 sampled RoIs for training. The candidate boxes are predicted by non-maximum suppression. Among the highest scoring 100 boxes, mask branch is applied to predict K masks per RoI. K is the predicted class. At test time, the proposal number is 300 for C4 backbone and 1000 for FPN.

IoU (Intersection over Union) threshold is set to 0.6 and 0.8, as is the same in paper [9]. Given precision P, recall R, AP (Average Precision) metric is applied to evaluate the performance. AP is the mean P of 101 points, defined as

$$AP = \frac{1}{101} \sum_{R\in(0.00,0.01,\ldots,0.50,\ldots,0.99,1.00)} \max_{\tilde{R}:\tilde{R}>R} P(\tilde{R}).$$

We train the model with 8 GPUs (TITAN XP 12G). Training takes about 5 hours for 36000 iterations. If only 1 GPU is used, training also takes 5 hours for 36000 iterations, but batch size is 2 (2 images per GPU), whereas, the batch size is 16 corresponding to 8 GPUs. That is to say, with 8 GPUs, the model uses 8-fold data than 1 GPU. In other words, the model is trained faster using 8 GPUs than 1 GPU. For demonstration, Fig. 4 illustrates the loss curves using 1 GPU. In this figure the losses are normalized. "loss_mask", "loss_classifier", and "loss_box_reg" are defined as (1) ∼ (3) respectively, and "total_loss" is the sum of three. "total_loss" and "loss_mask" have dramatic decline by the 5000 iterations. Although "loss_box_reg" and "loss_classifier" don't show the same falling gradient, they decrease steadily. And all loss curves see a steady decrease in the training process. Overall, our method converges very quickly before 5000 iterations. As for inference, the model runs at 0.089s per image. Actually, if the implement is optimized, the better speed would be got.

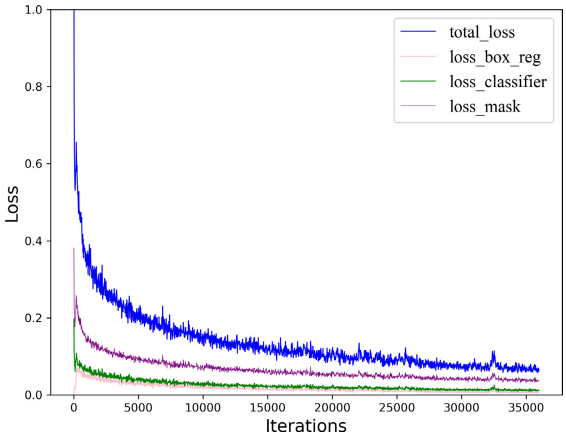


FIGURE 4. The loss curves using 1 GPU.

C. EVALUATIONS USING POD2017

To evaluate our method, we compare our method with 8 methods in the ICDAR2017 POD competition (POD2017) [9], and two recent methods are compared additionally: Li *et al.* [40] (in 1998) and YOLACT++ [3] (in 2020). As shown in Table 2, the first 8 rows of the table show performances of methods in POD2017 competition. Methods Li *et al.* [40]

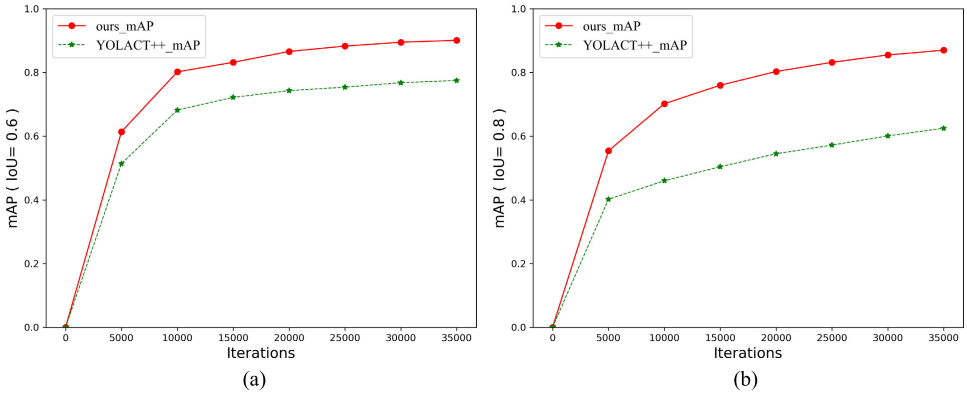


FIGURE 5. The mAP curves of our method and YOLACT++ which is not designed for document image with IoU of 0.6 (a) and IoU 0.8 (b) in training.

and YOLACT++ [3] are trained with the same configuration of ours.

All 11 methods are evaluated with mean of AP (mAP) and AP for different objects. Given IoU threshold of 0.6, our method gains best mAP 0.917, which is slightly higher than Li *et al.* (0.907). Although YOLACT++ is more recent method (in 2020) than Li *et al.* (in 2018), it gets lower mAP 0.780. YOLACT++ was more likely designed for natural image, so it is reasonable to get lower mAP than Li et al, which was designed for document image. NLPR-PAL and HustVision acquire good mAPs: 0.874 and 0.882. The former makes an integration of various strategies, including Support Vector Machine (SVM), Conditional Random Fields (CRF), Faster-RCNN, etc., to deal with different page objects. The latter employs five models based on CNN for different sizes of images. By fully considering page object size distribution (Section III Part A), our method achieves the best mAP. Especially for maths, our method significantly overcomes others with mAP 0.947, and Li *et al.* has the second top 0.878. When it comes to Table class, the top two mAPs are 0.959 (our method) and 0.946 (Li et al). However, evaluation on Figure shows the two highest mAPs are 0.896 (Li et al) and 0.853 (HustVision).

Overall, mAPs of our method are better than others in Table 2 on IoU of 0.6 except that on Figure. NLPR-PAL, icstpku, Li *et al.* and HustVision all consider inherent characteristics of page objects in document image, therefore they perform well. YOLACT++ was not designed for document image. Vislnt, SOS, UITVN, Matiai-ee are Faster R-CNN based methods or its variations, which output bounding boxes for page objects. By contrast, our method based on Mask R-CNN considers not only bounding box based detection but also semantic detection to output pixel-wise detection (mask) for objects, so as to performs better on small objects, such as Maths.

For IoU threshold of 0.8, on Maths, our method still outperforms other methods with mAP 0.901 and Li *et al.* gets second place (mAP 0.863). Considering average mAP with IoU threshold of 0.6 and 0.8, our method gets 0.897 which is slightly better than average mAP 0.8935 of Li *et al.* So it is safe to conclude that our method gets rank one according to evaluation in Table 2.

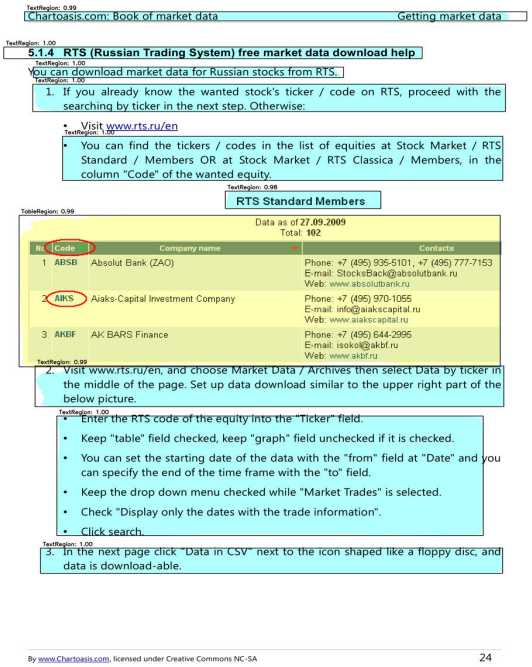
To compare network designed for document image with network designed for image in natural scene, Fig. 5 shows mAP curves of our method and YOLACT++ with IoU of 0.6 and 0.8 in training. In Fig. 5(a), the mAP of our method gets a better start exceeding the mAP of YOLACT++ at the 5000 iterations. After 10000 iterations, two curves remain parallel. And Fig. 5(b) shows similar trend. A network designed for document images can represent the inherent the characteristics of dataset, Hence, it can result in better performance in Table 2.

D. PIXEL-WISE DETECTION BETTER THAN REGION DETECTION

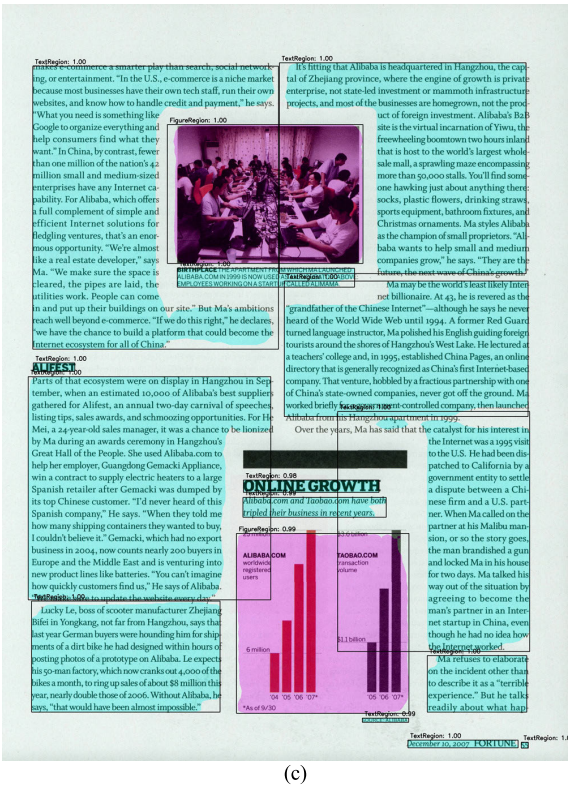
To further investigate the detection accuracy of our method, datasets with more complex layout structures are used to visualize detection results. As is can be seen in Fig. 6, page detection results from dataset RDCL2019 and Marmot are quite promising. Aggregated page objects, including text block, figure, table, and maths are shown in different color. The color palette is specified as [0,255,255] ‘‘aqua’’ for text blocks, [255,0,0] ‘‘red’’ for maths, [255,0,255] ‘‘fuchsia’’ for figures, [255,255,0] ‘‘yellow’’ for tables. For documents with complex layout, such as RDCL2019 dataset, there exist overlapping region between different page objects. Both Fig. 6 (a) and (c) have overlapped area between text and figure blocks when using bounding box recognition results. The pixel-wise mask prediction results marked with coloring pixels are generally better with the use of FCN within each proposed candidate region. Within the tables or figures, text line might appear to be table cells or illustrative texts. These text blocks did not show the misclassification in Fig. 6 (b).

The high confident text proposals with tables or figures are not misclassified. Although small regions are still challenging, the equation number following maths can be missed for detection in Fig. 6 (d). This method is capable of handling

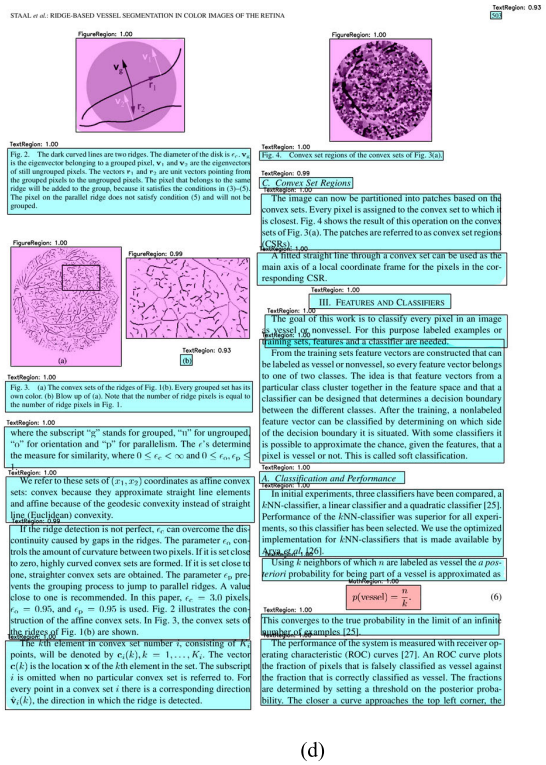
(a)



(b)



(c)



(d)

FIGURE 6. Example documents and their page object detection results. (a) and (c) are from RDCL2019 dataset. (b) and (d) are from Marmot dataset.

various shapes of page objects in complex layout documents. Unlike full page FCN method, its FCN is carried out only within each region candidate instead of the whole page, since pixel level segmentation has more expensive computational cost. And it is unnecessary to take extra post processing to clean the segmented masks. The bounding boxes plus

mask prediction for page object blocks are produced by two branches at the same time.

V. CONCLUSION

In this study, to detect hierarchical page objects for document images, a Mask R-CNN based network was proposed to output end to end results, including object classification, bounding box identification and page object mask generation. Block level region object recognition was of our consideration among various granularities. Latex based synthetic generation was designed to enlarge the training dataset. Compared with previous ICDAR page object detection competition methods, the proposed method achieved promising results with mAP 0.917 on dataset POD2017, which was better than the existing page object competition methods.

REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, 2016, pp. 21–37. [Online]. Available: http://www.eccv2016.org/

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*. [Online]. Available: http://arxiv.org/abs/2004.10934

[3] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT++: Better real-time instance segmentation," 2019, *arXiv:1912.06218*. [Online]. Available: http://arxiv.org/abs/1912.06218

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[7] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. L. Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5315–5324.

[8] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, May 2015, pp. 1520–1528.

[9] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "ICDAR2017 competition on page object detection," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1417–1422.

[10] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. ICCV*, Oct. 2017, pp. 2961–2969.

[11] C. Tensmeyer and T. Martinez, "Historical document image binarization: A review," *Social Netw. Comput. Sci.*, vol. 1, no. 3, pp. 1–26, May 2020.

[12] L. Jagannathan and C. V. Jawahar, "Perspective correction methods for camera based document analysis," in *Proc. Int. Workshop Camera Based Document Anal. Recognit. (CBDAR)*, 2005, pp. 148–154.

[13] A. Singh, G. Pang, M. Toh, J. Huang, W. Galuba, and T. Hassner, "TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8802–8812.

[14] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 161–184, 2020.

[15] C. Tensmeyer, B. Davis, C. Wigington, I. Lee, and B. Barrett, "PageNet: Page boundary extraction in historical handwritten documents," in *Proc. 4th Int. Workshop Historical Document Imag. Process.*, Nov. 2017, pp. 59–64.

[16] A. Mishra, K. Alahari, and C. V. Jawahar, "Unsupervised refinement of color and stroke features for text binarization," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 20, no. 2, pp. 105–121, Jun. 2017.

[17] H. F. Schantz, *The History of OCR, Optical Character Recognition. VT: Recognition Technologies Users Association Manchester*. Manchester, VT, USA: Recognition Technologies Users Association, 1982.

[18] N. Chen and D. Blostein, "A survey of document image classification: Problem statement, classifier architecture and performance evaluation," *Int. J. Document Anal. Recognit.*, vol. 10, no. 1, pp. 1–16, Jun. 2007.

[19] A. Zramdini and R. Ingold, "Optical font recognition using typographical features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 877–882, Aug. 1998.

[20] C. Wu and G. Agam, "Document image de-warping for text/graphics recognition," in *Structural, Syntactic, and Statistical Pattern Recognition. SSPR/SPR* (Lecture Notes in Computer Science), vol. 2396. Berlin, Germany: Springer, 2002, pp. 348–357.

[21] A. Amin and R. Shiu, "Page segmentation and classification utilizing bottom-up approach," *Int. J. Image Graph.*, vol. 1, no. 2, pp. 345–361, Apr. 2001.

[22] A. Simon, J. C. Pret, and A. P. Johnson, "A fast algorithm for bottom-up document layout analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 273–277, Mar. 1997.

[23] J. Ha, R. M. Haralick, and I. T. Phillips, "Recursive X-Y cut using bounding boxes of connected components," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, Aug. 1995, pp. 952–955.

[24] K. Kise, A. Sato, and M. Iwata, "Segmentation of page images using the area Voronoi diagram," *Comput. Vis. Image Understand.*, vol. 70, no. 3, pp. 370–382, Jun. 1998.

[25] X. Lin, L. Gao, Z. Tang, J. Baker, and V. Sorge, "Mathematical formula identification and performance evaluation in PDF documents," *Int. J. Document Anal. Recognit.*, vol. 17, no. 3, pp. 239–255, Sep. 2014.

[26] C. Xu, Z. Tang, X. Tao, Y. Li, and C. Shi, "Graph-based layout analysis for PDF documents," *Proc. SPIE*, vol. 8664, Mar. 2013, Art. no. 866407.

[27] M.-T. Luong, T. D. Nguyen, and M.-Y. Kan, "Logical structure recovery in scholarly articles with rich document features," in *Multimedia Storage and Retrieval Innovations for Digital Library Systems*. Hershey, PA, USA: IGI Global, 2012, pp. 270–292.

[28] X. Tao, Z. Tang, and C. Xu, "Contextual modeling for logical labeling of PDF documents," *Comput. Electr. Eng.*, vol. 40, no. 4, pp. 1363–1375, May 2014.

[29] A. Delaye and C.-L. Liu, "Contextual text/non-text stroke classification in online handwritten notes with conditional random fields," *Pattern Recognit.*, vol. 47, no. 3, pp. 959–968, Mar. 2014.

[30] N. Audebert, C. Herold, K. Slimani, and C. Vidal, "Multimodal deep networks for text and image-based document classification," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases* Cham, Switzerland: Springer, 2020, pp. 427–443.

[31] X. Yi, L. Gao, Y. Liao, X. Zhang, R. Liu, and Z. Jiang, "CNN based page object detection in document images," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 230–235.

[32] N. Vincent and J.-M. Ogier, "Shall deep learning be the mandatory future of document analysis problems?" *Pattern Recognit.*, vol. 86, pp. 281–289, Feb. 2019.

[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[34] F. Yu, V. Koltun, and T. Funkhouser, "Dilated residual networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 472–480.

[35] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

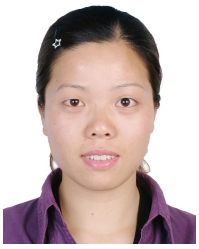
[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[37] C. Clausner, A. Antonacopoulos, and S. Pletschacher, "ICDAR2019 competition on recognition of documents with complex layouts–RDCL2019," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1521–1526.

[38] A. Dutta and A. Zisserman, "The VIA annotation software for images, audio and video," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 2276–2279.

[39] X. Tao, Z. Tang, C. Xu, and Y. Wang, "Logical labeling of fixed layout PDF documents using multiple contexts," in *Proc. 11th IAPR Int. Workshop Document Anal. Syst.*, Apr. 2014, pp. 360–364.

[40] X.-H. Li, F. Yin, and C.-L. Liu, "Page object detection from PDF document images by deep structured prediction and supervised clustering," in *Proc. 24th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2018, pp. 3627–3632.



CANHUI XU received the Ph.D. degree from Central South University, in 2011. She has been a Visiting Scholar with Arizona State University, USA, from 2019 to 2020. She is currently working with the Qingdao University of Science and Technology. Her research interests include document image processing and deep learning.

CAO SHI received the Ph.D. degree from Central South University, in 2011. He is currently working with the Qingdao University of Science and Technology. His research interests include image and video processing and artificial intelligence.

HENGYUE BI is currently pursuing the master's degree majoring in computer science and technology with the Qingdao University of Science and Technology. His research interests include image processing and deep learning.

CHUANQI LIU received the bachelor's degree from the Qingdao University of Science and Technology, in 2021.

YONGFENG YUAN received the bachelor's, master's, and Ph.D. degrees from the Harbin Institute of Technology, from 1998 to 2010. He is currently working with the Harbin Institute of Technology, as an Associate Professor. His research interests include concentrated on image processing, computer vision, and computational biology.

HAOYAN GUO received the Ph.D. degree from the Harbin Institute of Technology, in 2016. She is currently working with the Harbin Institute of Technology. Her research interests focus on data processing, including data analysis, multidimensional data processing, and software development.

YINONG CHEN received the Ph.D. degree from the Karlsruhe Institute of Technology (KIT), University of Karlsruhe, Germany, in 1993. He is currently working with Arizona State University. His research interests include service-oriented computing, visual programming, robotics, and artificial intelligence.