2019 International Conference on Document Analysis and Recognition (ICDAR)

A YOLO-based Table Detection Method

Yilun Huang*, Qinqin Yan†, Yibo Li*, Yifan Chen†,
Xiong Wang†, Liangcai Gao*, Zhi Tang*
*Institute of Computer Science & Technology, Peking University, Beijing, China
†State Key Laboratory of Digital Publishing Technology, Founder Group Co., LTD, Beijing, China

***Abstract*—Due to various table layouts and styles, table detection is always a difficult task in the field of document analysis. Inspired by the great progress of deep learning based methods on object detection, in this paper, we present a YOLO-based method for this task. Considering the large difference between document objects and natural objects, we introduce some adaptive adjustments to YOLOv3, including an anchor optimization strategy and two post processing methods. For anchor optimization, we use k-means clustering to find anchors which are more suitable for tables rather than natural objects and make it easier for our model to find exact positions of tables. In post-processing process, the extra whitespaces and noisy page objects (e.g. page headers, page footers) are removed from the predicted results, so that our model can get more accurate table margins and higher IoU scores. The proposed method is evaluated on two datasets from ICDAR 2013 Table Competition and ICDAR 2017 Page Object Detection (POD) Competition and achieves state-of-the-art performance.**

***Index Terms*—table detection; document analysis; deep learning; post processing**

I. Introduction

Nowadays, table has been a very important way to organize data and present information clearly and intuitively. At the same time, PDF becomes a popular document format all over the world, due to its stability and invariance on different operating systems. As a matter of course, in the current big data time, there are plenty of tables appearing in PDF files, especially in financial documents and government documents. It brings a large demand for detecting tables from documents automatically and quickly, so lots of researchers make their contributions to this field.

Years ago, methods based on layout analysis and rules were popular. Researchers tried to analyse the layout features of tables in documents. These features were used then to make several rules to cluster elements in documents into tables or divide the whole page into different parts and extract tables from them. However, because of the diversity of table styles and layouts, systems based on these methods were usually huge, bloated and complex. Above all, they lack of enough generalization ability, which means that they cannot work well on different kinds of documents at the same time.

In the last few years, deep learning showed its superiority on most fields of computer science, especially on computer vision and natural language processing. Deep learning based methods are data-driven, and data is everywhere nowadays. Compared with hand-crafted features obtained by conventional methods, features extracted by deep learning based methods are more robust and expressive. As a sub-field of computer vision, document analysis and recognition was influenced a lot by deep learning as well. Many researchers started to apply deep learning based methods to table detection task. In these methods, they usually converted PDF documents into page images, then fed them into deep neural networks to train and detect. Even though page objects like tables and figures in documents are quite different from natural objects such as cars and birds, deep learning models can also work well with little adjustment.

In this paper, we propose a deep learning based method for table detection. YOLOv3 [1] is used as our basic model. In addition, we make some adaptive adjustments and optimizations on original model according to the particular features of tables in documents. Some post-processing methods are also applied to the results outputted by basic model and make it perform a better result. Our method is evaluated on two datasets from ICDAR 2013 Table Competition [2] and ICDAR 2017 Page Object Detection (POD) Competition [3]. Experiment results show that our method achieves state-of-the-art performance and has nice generalization ability.

The main contributions of our work can be summarized as follows:

1) Application of the famous YOLOv3 model on table detection task. We are the first to adapt YOLOv3 to table detection task and achieve a fast and accurate detection model.
2) An anchor optimization method for YOLOv3. Due to the difference between natural objects and tables, we proposed an anchor optimization method to make anchors used in YOLOv3 more suitable for tables.
3) Two post-processing methods applied on outputs of basic model. These methods erase the whitespace margins of predicted table regions to obtain a higher IoU score and filter noisy page objects from positive predictions to improve the precision of our model.

The rest of this paper is organized as follows: in Section II, there is a brief review of related work on table detection. Section III describes the proposed method in detail. Section IV shows information of datasets, evaluation metrics and analyses the results of our experiments. Section V concludes this paper and discusses possible future works.

II. Related Work

Researches on table detection started from many years ago. Methods based on layout analysis and heuristic rules always existed during tens of years.

2379-2140/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDAR.2019.00135

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:24:15 UTC from IEEE Xplore. Restrictions apply.

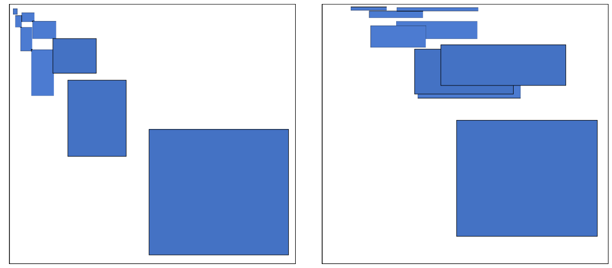
Ha et al. [4] proposed recursive X-Y cut method to analyse logical structure of documents for later page segmentation work. Kieninger et al. [5] proposed T-Recs System, which took document images as input and used a bottom-up analysis approach to output logical text block units. Later, Yildiz et al. [6] presented the pdf2table system. It extracted text lines from PDF documents and merged them into multi-line blocks and then tables. Hassan et al. [7] proposed a more systematic and complex method on table detection. They first divided tables into three different kinds according to the kinds of ruling lines. Then they proposed a principle of rectangle containment. Based on this principle, they found candidate columns and tables using adjacency graphs of text segments. However, due to lacking consideration, these methods can not work well on multi-column pages.

Fang et al. [8] used layout analysis methods proposed by Breuel [9] to analyse whitespace separators and get the number of columns of pages. In addition to this, they also made use of graphic lines obtained from PDF documents together with texts and separators to detect tables. Chen et al. [10] proposed a rectangle mining method to extract semantic relations between table rows and considered global features as well. Koci et al. [11] proposed a graph representation of layout regions of the page and an algorithm called *Remove and Conquer (RAC)*. With this algorithm, they recognized tables from graph representation of layout regions.

In recent years, a lot of deep learning based methods on object detection appeared and brought great performance improvements. There are several famous deep network models: VGG proposed by Simonyan et al. [12], Faster R-CNN proposed by Ren et al. [13], FCN proposed by Long et al. [14], ResNet and Mask R-CNN proposed by He et al. [15], [16] and YOLO series proposed by Redmon et al. [1], [17], [18]. These methods worked well on object detection and semantic segmentation tasks on natural images according to their experiments on multiple datasets.

Some of these deep learning based methods were also modified and used in table detection task. For sementic segmentation methods, Yang et al. [19] proposed a multimodal fully convolutional neural network for page segmentation to detect tables, figures and other page objects. They made use of both visual features from page images and and linguistic features from page texts. He et al. [20] proposed a multi-scale multi-task FCN to detect table regions and contours, and used results of contour detection task to help improve table detection task. Kavasidis et al. [21] presented a saliency-based CNN for table detection. They detected several semantic saliencies and classified them into different page objects including tables. Then they used conditional random field (CRF) to smooth table edges to obtain better performance.

For object detection methods, Hao et al. [22] proposed a table detection method based on CNN. They first selected some table-like regions by some loose rules, then used CNN to determine whether these regions are real tables. Besides, they made use of both visual features of PDF page images and invisible information extracted from corresponding PDF documents such as rendering instructions and so on. Schreiber et al. [23] and Gilani et al. [24] proposed a table detection model based on Faster R-CNN [13] with some processing on input and output data. Li et al. [25] first used layout analysis methods to get several candidate regions, then applied CRF and CNN to classify them into tables, formulas and other categories, which won the first place in ICDAR 2017 POD Competition [3]. Siddiqui et al. [26] proposed a deep deformable CNN model for table detection. It's a combination of deformable CNN with faster R-CNN/FPN [13] and could adapt its receptive field according to the input.



(a) Original anchor sizes (b) Optimized anchor sizes

Fig. 1: Comparison between original and optimized anchor sizes.

In addition, some interesting machine learning based methods are noteworthy. Farrukh et al. [27] proposed a bottom-up reasoning approach, which combines random forest, Markov Logical Network, k-means clustering to find the cell relation hierarchy. Rashid et al. [28] proposed a table recognition method based on auto tunable multilayer perceptron (AutoMLP). They used some artificial features to classify each cell into "belongs to table" or "not belongs to table" and results were enhanced by a contextual post-processing process. Besides, Clinchant et al. [29] presented experiments to compare two machine learning approaches they proposed, which are Conditional Random Field (CRF) and Graph Convolutional Network (GCN) or Edge Convolutional Network (ECN), for table recognition task. The results showed that CRF was slightly better than GCN/ECN, but ECN ran faster than CRF.

III. YOLO-based Table Detection

Our method is based on YOLOv3 [1], which is a deep learning model for detecting objects accurately and quickly. However, tables are quite different from natural objects. So we propose some adaptive adjustments to original YOLOv3, including anchor optimization and post processing.

A. Basic Model

YOLO was first proposed by Redmon et al. [17] in 2016. It's famous for its processing speed in real-time at 45 frames per second. At the same time, it still achieved competitive even state-of-the-art performance on popular datasets.

Different from region proposal methods such as Faster R-CNN [13], YOLO treated object detection as a regression

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:24:15 UTC from IEEE Xplore. Restrictions apply.

problem. That's the main reason for its high speed. In the later versions of YOLO, YOLO9000 [18] made several design decisions to improve the performance and speed, including using anchor boxes to predict bounding boxes of objects. In the latest YOLOv3 [1], Redmon et al. made more improvements including deepening the network structure, which obtained a better performance than previous two versions. That's why we choose YOLOv3 as our basic model.

B. Anchor Optimization

The conception "anchor" was proposed by Ren et al. [13] in the famous Faster R-CNN, where anchor was used in region proposal. Regions are outputted by predicting offsets from anchors instead of coordinates directly, which makes it easier for the network to learn. Redmon et al. added this nice design decision into their YOLO9000 model [18].

Different from the hand-picked anchors in Faster R-CNN, they find good anchors by clustering on ground truth bounding boxes of training set automatically. The distance metric used in clustering is shown as follow:

$$D(box, centroid) = 1 - IoU(box, centroid) \qquad (1)$$

where box denotes the bounding box that needs clustering and $centroid$ denotes the centroid bounding box of a cluster. The reason of using this distance instead of Euclidean distance is that the latter would make large boxes generate more error than smaller ones.

As mentioned above, tables are far different from natural objects so that the original anchor sizes proposed by YOLOv3 might not work in table detection task. So we need this anchor optimization processing. We applied k-means clustering on tables in training set of ICDAR 2017 POD Competition, with distance metric in equation 1 to optimize anchors, and k is 9. Optimized anchor sizes are more suited to tables in documents and help improve performance a lot. Figure 1 shows the comparison between original anchor sizes and optimized anchor sizes. Original anchor sizes are (10, 13), (16, 30), (33, 23), (30, 61), (62, 45), (59, 119), (116, 90), (156, 198), (373, 326), and optimized anchor sizes are (74, 6), (170, 7), (112, 14), (115, 45), (169, 36), (207,94), (214, 98), (262, 85), (295, 243).

As we can see, among original anchors, there are "vertical" and "horizontal" anchors for natural objects and they are nearly as many as each other. After optimization, anchors are more like tables. No matter how small a table is, the width of it would not be very small. The width of small tables is usually larger than the height, which means there are more "horizontal" anchors than "vertical" anchors. Only when tables are big enough, the width and height could be nearly the same or the height could be larger than the width. Optimized anchors present this special feature of tables. That is to say, anchor optimization makes it fitter for table detection.

C. Post Processing

We also propose two simple but effective post-processing methods.

One of them is erasing the whitespace margin from predicted regions. We know that for most PDF documents, the background is always white. Sometimes, predicted regions may be larger than ground truth regions. So that there might be whitespace margins in predicted regions but they should not have belonged to target tables according to ground truth. We check that if there are any whitespace margins in predicted regions in four directions, they are erased. The proposed algorithm is showed in Algorithm 1. However, sometimes this method doesn't work because there are some pages that have a non-white background. This method can improve both precision and recall.

Algorithm 1: Erasing whitespace margin algorithm

Input: P: all pixels of the predicted region.
Output: R: result region after erasing the whitespace margins.

$orientations = \{up, down, left, right\}$;
$R = P$;
for ori ***in*** $orientations$ **do**
  $M = \{$all pixels belong to the margin of R in direction $ori\}$;
  while *all pixels* $\in M$ *are white color* **do**
    $R = R - M$;
    $M = \{$all pixels belong to the margin of R in direction $ori\}$;
  end
end
return R;

The other method is filtering noisy page objects. According to our observation on training set, there are some noisy page objects. For example, there are a few page headers, footers and separator lines that are wrongly labelled as tables. These noisy page objects mislead our model to make the wrong decisions. In order to fix that, we make some rules and use them to filter false positive predictions. If a predicted table violates any of following heuristic filter conditions:

1) The minimum distance between it and the top or bottom of the page is less than $0.05 \times page\_height$.
2) The area of it is smaller than 500 pixels.
3) The ratio of its width over height or height over width is larger than 12.

we think it's not a table and remove it from positive predictions. This method improves precision a lot.

IV. Experiments

A. Datasets

Our method is evaluated on two datasets. One is from ICDAR 2013 Table Competition [2], the other is from ICDAR 2017 Page Object Detection (POD) Competition [3].

In the dataset from ICDAR 2013 Table Competition, there are PDF documents and XML ground truth files. Because our method needs images as input, we convert these PDF

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:24:15 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Experiment results on ICDAR 2017 POD Competition test set during optimizing our method by applying some adaptive adjustments. In this table, 'a' means anchor optimization and 'p' means proposed post-processing methods.

Model	IoU = 0.6			IoU = 0.8		
	Precision	Recall	F1-measure	Precision	Recall	F1-measure
YOLOv3	0.937	0.931	0.934	0.835	0.830	0.832
YOLOv3+p	0.940	0.934	0.937	0.911	0.905	0.908
YOLOv3+a	**0.978**	**0.972**	**0.975**	**0.975**	**0.968**	**0.971**
YOLOv3+a+p	**0.978**	**0.972**	**0.975**	**0.975**	**0.968**	**0.971**

TABLE II: Experiment results on ICDAR 2013 Table Competition dataset during optimizing our method by applying adaptive adjustments. In this table, 'a' means anchor optimization and 'p' means proposed post-processing methods.

Model	IoU = 0.6			IoU = 0.8		
	Precision	Recall	F1-measure	Precision	Recall	F1-measure
YOLOv3+p	0.913	0.872	0.891	0.819	0.782	0.800
YOLOv3+a	**0.986**	**0.936**	**0.961**	0.838	0.795	0.816
YOLOv3+a+p	**0.986**	**0.936**	**0.961**	**0.892**	**0.846**	**0.868**

TABLE III: Experiment results of YOLOv3 with anchor optimization and proposed post-processing methods on ICDAR 2017 POD Competition test set and ICDAR 2013 Table Competition dataset. Results are compared with competition teams and some prior work.

Dataset	IoU Threshold	Model	Precision	Recall	F1-measure
ICDAR 2017 POD Competition, Test set, 817 images	0.6	**Ours (YOLOv3+a+p)**	**0.978**	**0.972**	**0.975**
		Li et al. [25]	0.974	0.962	0.968
		DeCNT (2018) [26]	0.965	0.971	0.968
		NLPR-PAL	0.968	0.953	0.960
		School of Software	0.934	0.940	0.937
		FastDetectors	0.903	0.940	0.921
		VisInt	0.924	0.918	0.921
		maitai-ee	0.842	0.890	0.865
		icstpku	0.857	0.773	0.813
		UITVN	0.670	0.940	0.782
		IU-vision	0.230	0.221	0.225
		HustVision	0.071	0.959	0.132
	0.8	**Ours (YOLOv3+a+p)**	**0.975**	**0.968**	**0.971**
		Li et al. [25]	0.965	0.953	0.959
		DeCNT (2018) [26]	0.967	0.937	0.952
		NLPR-PAL	0.943	0.958	0.951
		FastDetectors	0.879	0.915	0.896
		VisInt	0.829	0.823	0.826
		School of Software	0.793	0.798	0.796
		maitai-ee	0.755	0.798	0.776
		icstpku	0.804	0.726	0.763
		UITVN	0.544	0.763	0.635
		IU-vision	0.118	0.114	0.116
		HustVision	0.062	0.836	0.115
ICDAR 2013 Table Competition, Complete set, 238 images	0.5	**DeCNT (2018) [26]**	0.996	**0.996**	**0.996**
		Kavasidis et al. (2018) [21]	0.975	0.981	0.978
		Ours (YOLOv3+a+p)	**1.000**	0.949	0.973
		Sebestian et al. (2017) [23]	0.974	0.962	0.968
		Tran et al. (2015) [30]	0.952	0.967	0.958
		Hao et al. (2016) [22]	0.972	0.922	0.946

documents into page images. As a result, there are 238 images in this dataset.

In the dataset from ICDAR 2017 POD Competition, there are 2417 images which include 1600 training images and 817 test images, and they form the training set and test set respectively. Although this competition requires to detect 3 kinds of page objects from page images, including table, figure and formula, we only focus on table detection task.

Our method is trained on the training set of ICDAR 2017 POD Competition and evaluated on test set of it and the whole dataset of ICDAR 2013 Table Competition.

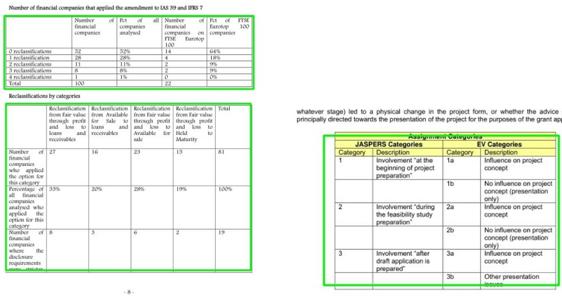
B. Evaluation Metric

In both competition, the classical Precision, Recall and F1-measure metric with specific Intersection over Union (IoU) threshold is used. IoU can be computed as:

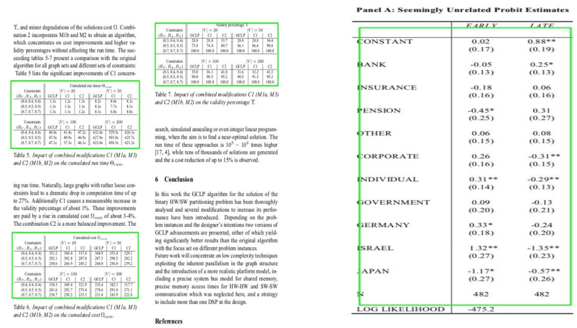
$$IoU_i = \frac{P_i \cap G_i}{P_i \cup G_i} \tag{2}$$

where P_i denotes the region i predicted by our method and G_i denotes the corresponding region in ground truth. If the IoU of a predicted region is larger than specific IoU threshold, it's regarded as a true positive (TP) sample. Otherwise, it's a

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:24:15 UTC from IEEE Xplore. Restrictions apply.



(a) ICDAR 2013 Table Competition good/bad result samples



(b) ICDAR 2017 POD Competition good/bad result samples

Fig. 2: Two sets of table detection result samples on two datasets. In each set, the left image is a sample in which tables are detected correctly. On the contrary, there are some detection results are not good enough or totally wrong in the right image of each set.

false positive (FP) sample. Regions that are in ground truth but not truly detected are false negative (FN) samples. With these three conceptions, we could compute Precision (P), Recall (R) and F1 value as follows:

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP + FN} \tag{4}$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \tag{5}$$

Our method is optimized and evaluated on ICDAR 2017 POD Competition dataset and ICDAR 2013 Table Competition dataset with the IoU threshold of 0.6 and 0.8. Besides, due to most of prior work that is evaluated on ICDAR 2013 Table Competition dataset used 0.5 as IoU threshold, we also provide a evaluation result on this IoU threshold in order to compare the performance of our method with prior work's.

C. Experiments Result and Analysis

Our method is trained on the training set of ICDAR 2017 POD Competition. The size of input image is 416 $\times$ 416, the same as YOLOv3. Throughout training, we use a batch size of 64, a learning rate of 0.001, a momentum of 0.9 and a decay rate of 0.0005. And we train our model for 198200 iterations in total.

Our model is modified and optimized by applying adaptive adjustments on basic model YOLOv3. After every modification and optimization, it's trained and evaluated on the datasets again from the beginning. Table I and table II show the experiment results of our method on two datasets mentioned above. It verifies that our adaptive adjustments, including anchor optimization and post-processing methods, are useful and effective, especially the former.

An interesting point is that although both methods improve the performance of our model respectively, after combining them together, it seems that post-processing process doesn't help at all as table I shows. However, experiments on ICDAR 2013 Table Competition dataset show that in some cases, when IoU threshold is large, post processing does improve the performance. We think that anchor optimization works well on table region detection, but there are still some slight errors in determining table edges. When IoU threshold is small, this problem is tolerable. But when IoU threshold is large enough, predicted tables with inaccurate edges are filtered so that performance declines. Our proposed post-processing methods help to correct table edges and make them more accurate in this situation.

For the reason mentioned above, we choose the YOLOv3 model with anchor optimization and post processing as our final model. Table III shows experiment results on ICDAR 2013 Table Competition dataset and ICDAR 2017 POD Competition test set compared with other methods. The results of competition teams and some prior work are presented in the table as well. As it shows, our method achieves state-of-the-art performance on ICDAR 2017 POD Competition test set on both 0.6 and 0.8 IoU thresholds. It also presents leading and competitive performance on ICDAR 2013 Table Competition dataset. Our method shows 100% table detection precision on this dataset even it didn't involve any samples in this dataset during the training phase. After all, experiment results on both datasets demonstrate that our method has powerful generalization ability.

There are some table detection result samples on two datasets shown in figure 2. In both datasets, predicted bounding boxes are nearly the exact edges of table regions in good result samples. However, there are also some bad result samples. One bad result sample of ICDAR 2013 Table Competition dataset is shown in the right image of subfigure 2a. Although the position of table is accurate, bias exists in predicted width and height. When IoU threshold is large, this prediction would be regarded as a false positive and cause the decline of performance. The other bad result sample is from ICDAR 2017 POD Competition test set, which is shown in the right image of subfigure 2b. We can see that main problem is still wrong width and height prediction. Due to the lack

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:24:15 UTC from IEEE Xplore. Restrictions apply.

of vertical ruling lines on the left and right, predicted edges shrink into tables. It's a problem from unruled and less-ruled tables that we have to face. In a word, we can see that the accuracy of predicting width and height of tables still needs improving, especially for tables with missing ruling lines.

V. Conclusion

This paper proposed a deep learning based method for table detection, which uses the famous YOLOv3 model as the basic model. When transforming from natural images to documents images, some adaptive adjustments are applied to it, including anchor optimization and some special post-processing methods. Experiments on two different datasets from two different ICDAR competition show that this method is effective and robust and has nice generalization ability.

We believe that more adjustments should be applied to deep learning based methods transformed from natural images to documents images due to the difference between them. More efficient post-processing methods are waiting for being mined as well. Besides, how to find better methods to predict width and height of tables more accurately is also a significant issue in the future, and it's worth considering special strategies to deal with the particularity of unruled and less-ruled tables.

Acknowledgment

This work is supported by the projects of National Natural Science Foundation of China (No. 61573028 and 61472014), which is also a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

References

[1] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[2] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "Icdar 2013 table competition," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1449–1453.

[3] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "Icdar2017 competition on page object detection," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 1417–1422.

[4] J. Ha, R. M. Haralick, and I. T. Phillips, "Recursive xy cut using bounding boxes of connected components," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 2. IEEE, 1995, pp. 952–955.

[5] T. G. Kieninger, "Table structure recognition based on robust block segmentation," in *Document Recognition V*, vol. 3305. International Society for Optics and Photonics, 1998, pp. 22–33.

[6] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A method to extract table information from pdf files," in *IICAI*, 2005, pp. 1773–1785.

[7] T. Hassan and R. Baumgartner, "Table recognition and understanding from pdf files," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE, 2007, pp. 1143–1147.

[8] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage pdf documents via visual seperators and tabular structures," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*. IEEE, 2011, pp. 779–783.

[9] T. M. Breuel, "Two geometric algorithms for layout analysis," in *International workshop on document analysis systems*. Springer, 2002, pp. 188–199.

[10] X. Chen, L. Chiticariu, M. Danilevsky, A. Evfimievski, and P. Sen, "A rectangle mining method for understanding the semantics of financial tables," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, 2017, pp. 268–273.

[11] E. Koci, M. Thiele, W. Lehner, and O. Romero, "Table recognition in spreadsheets via a graph representation," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 2018, pp. 139–144.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *International Conference on Neural Information Processing Systems*, 2015, pp. 91–99.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[18] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[19] 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," 2017.

[20] D. He, S. Cohen, B. Price, D. Kifer, and C. L. Giles, "Multi-scale multi-task fcn for semantic page segmentation and table detection," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, 2017, pp. 254–261.

[21] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," *arXiv preprint arXiv:1804.06236*, 2018.

[22] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for pdf documents based on convolutional neural networks," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. IEEE, 2016, pp. 287–292.

[23] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, 2017, pp. 1162–1167.

[24] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, 2017, pp. 771–776.

[25] X.-H. Li, F. Yin, and C.-L. Liu, "Page object detection from pdf document images by deep structured prediction and supervised clustering," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3627–3632.

[26] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, "Decnt: Deep deformable cnn for table detection," *IEEE Access*, vol. 6, pp. 74 151–74 161, 2018.

[27] W. Farrukh, A. Foncubierta-Rodriguez, A. N. Ciubotaru, G. Jaume, C. Bejas, O. Goksel, and M. Gabrani, "Interpreting data from scanned tables," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, 2017, pp. 5–6.

[28] S. F. Rashid, A. Akmal, M. Adnan, A. A. Aslam, and A. Dengel, "Table recognition in heterogeneous documents using machine learning," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, 2017, pp. 777–782.

[29] S. Clinchant, H. Déjean, J.-L. Meunier, E. M. Lang, and F. Kleber, "Comparing machine learning approaches for table recognition in historical register books," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*. IEEE, 2018, pp. 133–138.

[30] D. N. Tran, T. A. Tran, A. Oh, S. H. Kim, and I. S. Na, "Table detection from document image using vertical arrangement of text blocks," *International Journal of Contents*, vol. 11, no. 4, pp. 77–85, 2015.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on May 19,2021 at 10:24:15 UTC from IEEE Xplore. Restrictions apply.