2019 International Conference on Document Analysis and Recognition (ICDAR)

Faster R-CNN Based Table Detection Combining Corner Locating

Ningning Sun, Yuanping Zhu and Xiaoming Hu
School of Computer and Information Engineering
Tianjin Normal University
Tianjin, China
Email: zhuyuanping@tjnu.edu.cn

***Abstract*—Table detection in document images has achieved remarkable improvement. However, there is still a problem of inaccurate table boundary locating. This paper proposes Faster R-CNN based table detection combining corner locating method. Firstly, coarse table detection and corner locating are implemented through Faster R-CNN network. Secondly, those corners belonging to the same tables are grouped by coordinate matching. At the same time, unreliable corners are filtered. Finally, table boundaries are adjusted and refined by corresponding corner group. The proposed method improves the precision of table boundary locating at pixel-level. Experiment results show that our method effectively improves the precision of table detection. It achieves an F-measure of 94.9% on ICDAR2017 POD dataset. Compared with traditional Faster R-CNN method, our method increases by 2.8% in F-measure and 2.1% at pixel-level localization.**

***Keywords*-table detection; corner locating; Faster R-CNN;**

I. Introduction

Table detection has always been an important area of our research, which promotes the development of layout analysis. Due to external and internal factors, table detection has always faced many problems. External factors mainly include tilt, blur, noise, illumination and occlusion et al, which are inevitable. Therefore, we should take some measures to weaken the influence of external factors. The measures include tilt refining [1], deblurring [2], denoising, illumination preprocessing [3] et al. However, these technologies are still immature and need to be continuously optimized. Internal factors mainly include: (1) complex layouts, such as single-column page, double-column pages and three-column pages; (2) The diversity of tables, whether contain horizontal or vertical lines, font style, font size and content format; (3) The interference of table location, including side-by-side tables, list tables and nested tables. The complication of document images leads to table detection difficulty.

In the past few years, people have implemented table detection based on traditional methods. Traditional methods are divided into two categories, shape-based methods and texture-based methods. Shape-based methods include top-down method [4] and bottom-up method [5]. For example, RLSA (Run-Length Smoothing Algorithm) [6] is a typical shape-based method commonly used in table detection. Texture-based methods include model-based method, feature-based method [7] and multiscale-based method [8]. Traditional methods are limited to many factors, which promotes the rapid development of deep learning based methods. Therefore, deep learning based methods are the current mainstream methods in table detection. A lot of experiments have proved that deep learning shows better in learning features. Deep learning based methods are divided into two categories, bounding-box based method and semantic segmentation method. Bounding-box based method can predict object proposal. However, bounding-box method is limit to the variable length bounding-box list. Hence, this method leads to inaccurate table locating. Semantic segmentation method classifies each pixel in the image, which typically uses a per-pixel softmax, like Mask R-CNN method [9]. Semantic segmentation method is commonly applied to natural scene images.

Different from the previous methods, we propose a new method that combines corner locating method. Corners mentioned in this paper refer to table corners, as shown in Fig. 1. Corners mainly have the following characteristics: (1) The region with a certain range centered on the vertices of table is called table corner. (2) Corners belonging to the same table are called a corner group. (3) Corners have the same size and shape except of position. They locates in top-left, top-right, bottom-left and bottom-right of the table. For convenience, they are named top-left corner, top-right corner, bottom-left

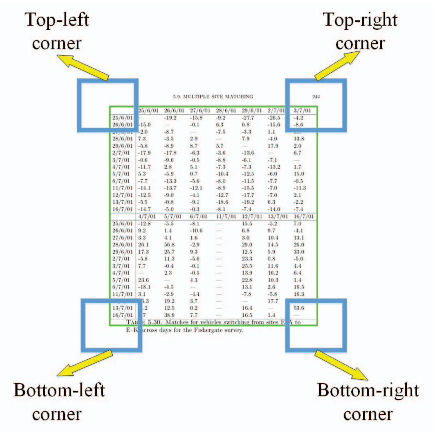


Fig. 1: The green bounding-box is a table and the blue bounding-boxes are corners.

2379-2140/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDAR.2019.00212

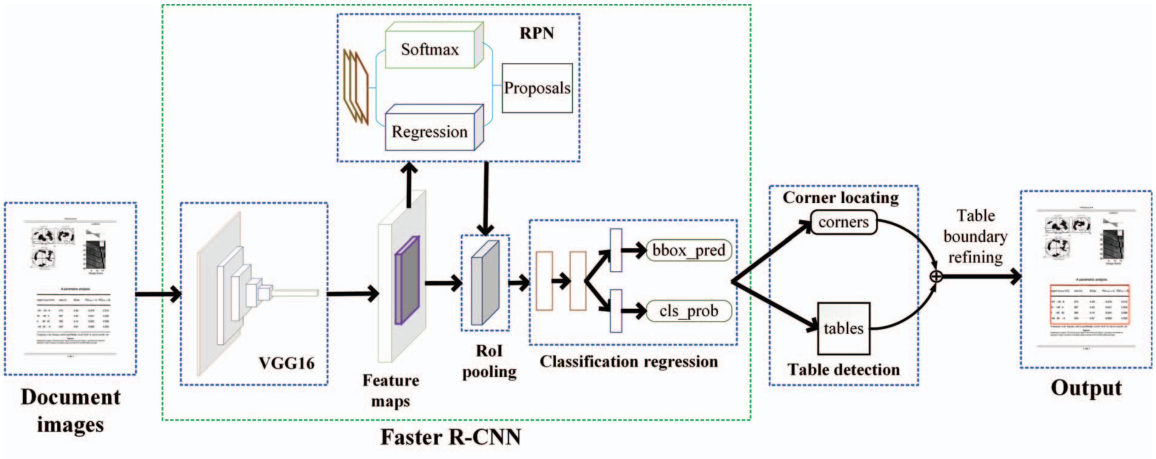


Fig. 2: The framework of Faster R-CNN based table detection combining corner locating.

corner and bottom-right corner, respectively.

Our method can be implemented in two steps. Firstly, table detection is implemented through Faster R-CNN [10]. Secondly, table boundaries are refined by reliable corners. Compared with the previous bounding-box based method, our method has two advantages: (1) For most tables, there exist corner groups for refining table boundary. (2) By refining table boundaries, the precision of table detection has significantly improved. Details will be introduced in Section 3. We have verified our method on a standard layout analysis dataset, ICDAR2017 Competition of Page Object Detection dataset [11]. Results show that the proposed method is superior to the widely used Faster R-CNN and SSD [12] method.

II. Related work

Table detection has always been the important research. Therefore, people have done a plenty of work in it. T. Kieninger et al. [13] [14] have completed one of the groundbreaking work on table detection. However, the method they proposed could only be processed for specific documents. F. Cesarini et al. [15] have proposed a top-down table detection method . The region surrounded by the lines is identified as a table candidate area by detecting horizontal and vertical lines of the document images. Basilios et al. [16] further studies on the basis of Cesari by detecting horizontal lines, vertical lines and intersect points. However, their methods were poorly versatile and not suitable for tables without horizontal and vertical lines. Inspired by [15] [16], we think of refining table boundaries in table detection. Traditional methods have made great progress in table detection, which promotes the development of table detection. At the same time, it is accompanied with many problems, such as partial detection, over-segmented tables, under-segmented tables, and missed tables.

At present, the mainstream table detection methods are deep learning based methods. Deep learning has become a part of leading systems in many fields, especially in the field of layout analysis. R-CNN (region CNN) [17] is a pioneering work using deep learning for object detection. R-CNN is accompanied with the emergence of double counting, which increases the amount of calculation and reduces the performance of the network. Therefore, R. Girshick refers to SPP-net (Spatial Pyramid Pooling) [18] and proposes Fast R-CNN [19] to improve the speed. On this basis, Faster R-CNN method is proposed. Faster R-CNN exponentially reduces computation complexity. Given the fact that Faster R-CNN is superior in both speed and accuracy, we choose Faster R-CNN for table detection.

The method combining corners is a novel method in object detection. A corner detection layer and an analytical sampling layer are used to replace RPN network in DeNet [20]. X. Wang have presented PLN (Point Linking Network) [21], which uses a full convolution network to regress the bounding-box and the corresponding corner/center point. P. Lyu et al [22] have developed a multi-oriented scene text detection model, which can detect multi-oriented scene text in natural scene via corner localization. CornerNet [23] model is proposed for natural scene. Compared with all existing one-step detectors, Cornernet achieves the best result in MS COCO. Inspired by the above methods, we propose a method via corner locating to refine table boundary.

III. Methodology

A. Architecture

Based on Faster R-CNN method, we combine corner locating method in table detection. The proposed method implements table detection and improves the precision. The structure of the network is shown in Fig. 2. Firstly, the backbone network VGG16 extract feature maps from document images. Secondly, RPN and Fast R-CNN implement table detection and corner locating. Thirdly, corners are grouped by coordinate matching. Unreliable corners are filtered through grouping and several prior rules. Finally, tables are adjusted and refined via the corresponding reliable corners. Our method improves the

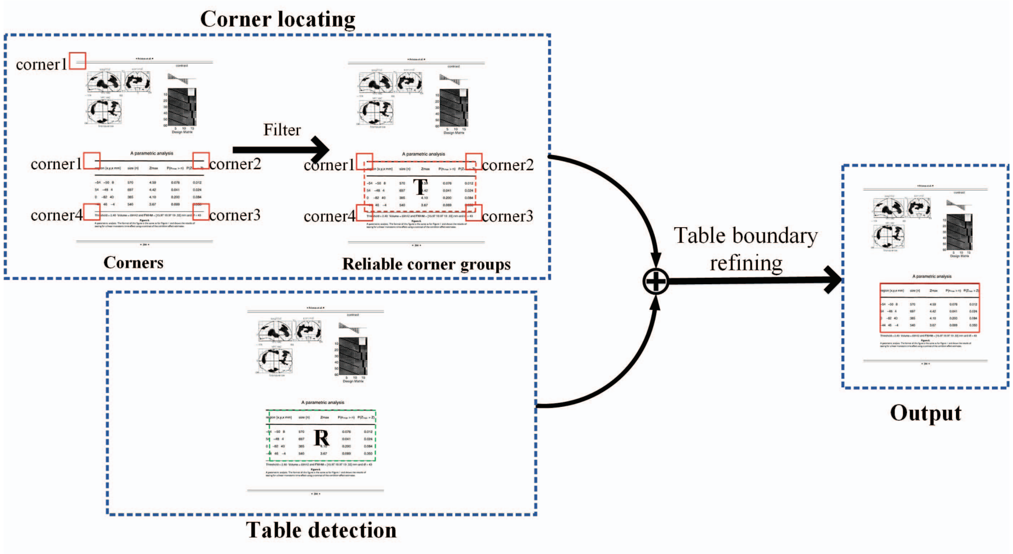


Fig. 3: Table boundary refining.

precision of table detection, especially at pixel-level localization.

B. Table Detection

Table detection is implemented by two steps. Firstly, the backbone network VGG16 extracts feature maps from document images. Secondly, feature maps are sent into the RPN to implement coarse detection. Through coarse detection, object foreground and background that meet the requirements are obtained. Thirdly, feature maps and ROI regions generated by RPN are fed together to the ROI pooling layer. ROI pooling layer generates a fixed-size feature map. Finally, table detection is implemented through classification and regression. For convenience, the table detected by Faster R-CNN is called T.

C. Corner Locating and Filtering

For convenience, we use the same model to implement table detection and corner detection. The process of corner detection is the same as table detection. In this work, we set corner size to 80 in default. We name the corresponding corners $corner1$, $corner2$, $corner3$ and $corner4$, respectively (in a clockwise direction). At the same time, they can also be defined as $C1(x_1, y_1)$, $C2(x_2, y_2)$, $C3(x_3, y_3)$ and $C4(x_4, y_4)$.

Inspired by CornerNet [23], corners belonging to the same table are grouped by coordinate matching. For convenience, we call the rectangular region determined by corner group as:

$$R = \{C_i'(x_i', y_i')|i\epsilon\{1, 2\}\} \tag{1}$$

$$\begin{cases} x_1' = (x_1 + x_4)/2 \\ y_1' = (y_1 + y_2)/2 \\ x_2' = (x_2 + x_3)/2 \\ y_2' = (y_3 + y_4)/2 \end{cases} \tag{2}$$

Not all corners are reliable, unreliable corners can be filtered by corner grouping method. Corners satisfied the following location constraint relationships would be retained:

- $C1$ and $C2$ are on the same horizontal line approximately.
- $C3$ and $C4$ are on the same horizontal line approximately.
- $C1$ and $C4$ are on the same vertical line approximately.
- $C2$ and $C3$ are on the same vertical line approximately.

The deviation of adjacent corners greater than z are rejected (z is suitable threshold), defined as:

$$|(x_i - x_j)| \leq z \qquad \{i = 1, j = 4\}or\{i = 2, j = 3\} \tag{3}$$

and

$$|(y_i - y_j)| \leq z \qquad \{i = 1, j = 2\}or\{i = 3, j = 4\} \tag{4}$$

in which, z is related to image size. $z = \sqrt{\theta wh}$ ($0.0002 \leq \theta \leq 0.0005$), image is of size $w \times h$. In this paper, the default value of θ is 0.0003.

D. Table Boundary Refining

Observing table documents, we noticed that many tables contain horizontal lines but without vertical lines. Observing table detection results implemented by Faster R-CNN, We found that inaccurate T were mainly caused by inaccurate table detection of left and right boundaries, while upper and lower boundaries had little effect on table detection. Therefore, only left and right boundaries of T are refined through reliable corner groups. The steps are shown Fig. 3. The meanings of the two items are explained as follows:

- Search corresponding R for T. The rule we used is IOU (Intersection-over-Union). If $IOU(R, T)$ is greater than the threshold we set (the default is 0.5), R is retained.
- Refine left and right boundaries of T by merging the corresponding R and T.

Suppose (x''_1, y''_1), (x''_2, y''_2) are coordinates on the top-left and bottom-right vertex of the T, respectively. Therefore, T is defined as $(x''_1, y''_1, x''_2, y''_2)$. Same as before, R is defined as (x'_1, y'_1, x'_2, y'_2) and refined table is defined as $(x^*_1, y^*_1, x^*_2, y^*_2)$. The relationship between them described as:

$$\begin{cases} x^*_1 = \frac{x''_1 + x'_1}{2} \\ y^*_1 = y''_1 \\ x^*_2 = \frac{x''_2 + x'_2}{2} \\ y^*_2 = y''_2 \end{cases} \tag{5}$$

After the above steps, table boundaries are refined by reliable corner groups. The proposed method is also applicable to other datasets.

E. Loss Function

The loss function consists of two parts. (1)The classification layer determines the proposal is foreground or background. (2) The regression layer is used to predict the central anchor point of the proposal, the coordinates corresponding to x, y, w and h. Faster R-CNN follows the definition of multi-task loss. The loss function consists of classification loss and regression loss, defined as:

$$\begin{aligned} L(\{p_i\}, \{t_i\}) = & \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p^*_i) \\ & + \lambda \frac{1}{N_{reg}} \sum_i p^*_i L_{reg}(t_i, t^*_i) \end{aligned} \tag{6}$$

in which, $L_{cls}(p_i, p^*_i)$ is the classification loss, expressed as a logarithmic function of two categories (foreground and background), calculated as: $L_{cls}(p_i, p^*_i) = -\log[p_i p^*_i + (1 - p^*_i)(1 - t^*_i)]$. $L_{reg}(t_i, t^*_i)$ is the regression loss, the calculation formula is: $L_{reg}(t_i, t^*_i) = R(t_i - t^*_i)$. R is a smooth L1 function, $p^*_i L_{reg}$ means that the regression loss is only activated for positive anchors ($p^*_i = 1$) and is disabled for otherwise($p^*_i = 0$).

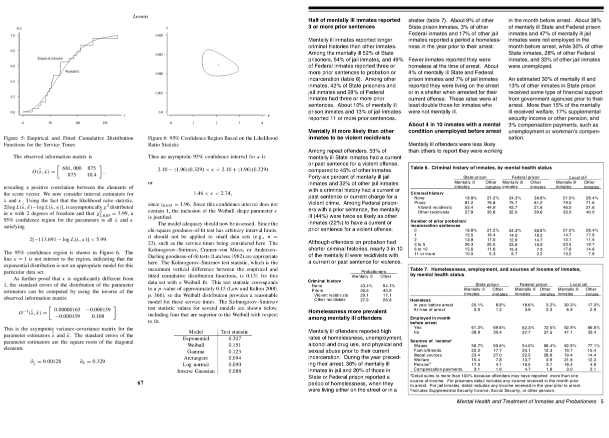
IV. Experiment

In order to verify the effectiveness of our method, we comprehensively evaluate our method on ICDAR2017 POD dataset [11].

A. Experimental Details

The experiment is trained in tensorflow 1.3.0 environment, with one GTX1080Ti, 11G of GPU memory is sufficient. The network is randomly initialized under the default setting of tensorflow with no pretraining on any external dataset. The network is fine-tuned with an initial learning rate of 0.0001 and slowly reduce, minibatch from 2 to 10. Model is continued to train for 50k iterations.

The input resolution is set to 600 × h when the network is trained (h ≤1000). The output resolution ie set to 1200 × 1200. Corner size is set to a fixed value (related to image size). To reduce overfitting, we use random horizontally-flipped images for data augmentation. Not only original images but also horizontally-flipped images are used in the testing phase. Output contains five classes together through Faster R-CNN, which are tables and four corners. Table boundary is refined through all reliable corners. Because the initial generated anchor is restricted by the radio and size, so small objects (for example: corners) below the minimum anchor size are frequently missed in the latter process.



Leemis

Figure 5: Empirical and Fitted Cumulative Distribution Functions for the Service Times

The observed information matrix is

$$O(\hat{\lambda}, \hat{\kappa}) = \begin{bmatrix} 681,000 & 875 \\ 875 & 10.4 \end{bmatrix},$$

revealing a positive correlation between the elements of the score vector. We now consider interval estimators for λ and κ. Using the fact that the likelihood ratio statistic, $2[\log L(\hat{\lambda}, \hat{\kappa}) - \log L(\lambda, \kappa)]$, is asymptotically χ^2 distributed in n with 2 degrees of freedom and that $\chi^2_{2,0.05} = 5.99$, a 95% confidence region for the parameters is all λ and κ satisfying

$$2[-113.691 - \log L(\lambda, \kappa)] < 5.99.$$

The 95% confidence region is shown in Figure 6. The line $\kappa = 1$ is not interior to the region, indicating that the exponential distribution is not an appropriate model for this particular data set.

As further proof that κ is significantly different from 1, the standard errors of the distribution of the parameter estimators can be computed by using the inverse of the observed information matrix

$$O^{-1}(\hat{\lambda}, \hat{\kappa}) = \begin{bmatrix} 0.00000165 & -0.000139 \\ -0.000139 & 0.108 \end{bmatrix}.$$

This is the asymptotic variance-covariance matrix for the parameter estimators $\hat{\lambda}$ and $\hat{\kappa}$. The standard errors of the parameter estimators are the square roots of the diagonal elements

$$\hat{\sigma}_{\hat{\lambda}} = 0.00128 \qquad \hat{\sigma}_{\hat{\kappa}} = 0.329.$$

Figure 6: 95% Confidence Region Based on the Likelihood Ratio Statistic

Thus an asymptotic 95% confidence interval for κ is

$$2.10 - (1.96)(0.329) < \kappa < 2.10 + (1.96)(0.329)$$

or

$$1.46 < \kappa < 2.74,$$

since $z_{0.025} = 1.96$. Since this confidence interval does not contain 1, the inclusion of the Weibull shape parameter κ is justified.

The model adequacy should now be assessed. Since the chi-square goodness-of-fit test has arbitrary interval limits, it should not be applied to small data sets (e.g., n = 23), such as the service times being considered here. The Kolmogorov–Smirnov, Cramer–von Mises, or Anderson–Darling goodness-of-fit tests (Lawless 1982) are appropriate here. The Kolmogorov–Smirnov test statistic, which is the maximum vertical difference between the empirical and fitted cumulative distribution functions, is 0.151 for this data set with a Weibull fit. This test statistic corresponds to a p-value of approximately 0.15 (Law and Kelton 2000, p. 366), so the Weibull distribution provides a reasonable model for these service times. The Kolmogorov–Smirnov test statistic values for several models are shown below, including four that are superior to the Weibull with respect to fit.

Model	Test statistic
Exponential	0.307
Weibull	0.151
Gamma	0.123
Arctangent	0.094
Log normal	0.090
Inverse Gaussian	0.088

67

Half of mentally ill inmates reported 3 or more prior sentences

Mentally ill inmates reported longer criminal histories than other inmates. Among the mentally ill 52% of State prisoners, 54% of jail inmates, and 49% of Federal inmates reported three or more prior sentences to probation or incarceration (table 6). Among other inmates, 42% of State prisoners and jail inmates and 28% of Federal inmates had three or more prior sentences. About 10% of mentally ill prison inmates and 13% of jail inmates reported 11 or more prior sentences.

Mentally ill more likely than other inmates to be violent recidivists

Among repeat offenders, 53% of mentally ill State inmates had a current or past sentence for a violent offense, compared to 45% of other inmates. Forty-six percent of mentally ill jail inmates and 32% of other jail inmates with a criminal history had a current or past sentence or current charge for a violent crime. Among Federal prisoners with a prior sentence, the mentally ill (44%) were twice as likely as other inmates (22%) to have a current or prior sentence for a violent offense.

Although offenders on probation had shorter criminal histories, nearly 3 in 10 of the mentally ill were recidivists with a current or past sentence for violence.

	Probationers	
	Mentally ill	Other
Criminal history		
None	43.4%	54.1%
Priors	56.6	45.9
Violent recidivists	29.1	17.1
Other recidivists	27.6	28.8

Homelessness more prevalent among mentally ill offenders

Mentally ill offenders reported high rates of homelessness, unemployment, alcohol and drug use, and physical and sexual abuse prior to their current incarceration. During the year preceding their arrest, 30% of mentally ill inmates in jail and 20% of those in State or Federal prison reported a period of homelessness, when they were living either on the street or in a shelter (table 7). About 9% of other State prison inmates, 3% of other Federal inmates and 17% of other jail inmates reported a period a homelessness in the year prior to their arrest.

Fewer inmates reported they were homeless at the time of arrest. About 4% of mentally ill State and Federal prison inmates and 7% of jail inmates reported they were living on the street or in a shelter when arrested for their current offense. These rates were at least double those for inmates who were not mentally ill.

About 4 in 10 inmates with a mental condition unemployed before arrest

Mentally ill offenders were less likely than others to report they were working in the month before arrest. About 38% of mentally ill State and Federal prison inmates and 47% of mentally ill jail inmates were not employed in the month before arrest, while 30% of other State inmates, 28% of other Federal inmates, and 33% of other jail inmates were unemployed.

An estimated 30% of mentally ill and 13% of other inmates in State prison received some type of financial support from government agencies prior to their arrest. More than 15% of the mentally ill received welfare, 17% supplemental security income or other pension, and 3% compensation payments, such as unemployment or workman's compensation.

Table 6. Criminal history of inmates, by mental health status

	State prison		Federal prison		Local jail	
	Mentally ill inmates	Other inmates	Mentally ill inmates	Other inmates	Mentally ill inmates	Other inmates
Criminal history						
None	18.8%	21.2%	24.3%	38.8%	21.0%	28.4%
Priors	81.2	78.8	75.7	61.2	79.0	71.6
Violent recidivists	53.4	44.9	43.7	21.6	46.0	31.6
Other recidivists	27.8	33.8	32.0	39.6	33.0	40.0
Number of prior probation/ incarceration sentences						
0	18.8%	21.2%	24.3%	38.8%	21.0%	28.4%
1	15.5	19.4	14.0	18.2	14.7	17.9
2	13.8	17.0	12.9	14.7	10.1	11.5
3 to 5	26.3	25.5	23.6	18.9	23.5	19.7
6 to 10	15.6	11.6	15.4	7.3	17.6	14.6
11 or more	10.0	5.3	9.7	2.2	13.2	7.8

Table 7. Homelessness, employment, and sources of income of inmates, by mental health status

	State prison		Federal prison		Local jail	
	Mentally ill inmates	Other inmates	Mentally ill inmates	Other inmates	Mentally ill inmates	Other inmate
Homeless						
In year before arrest	20.1%	8.8%	18.6%	3.2%	30.3%	17.3%
At time of arrest	3.9	1.2	3.9	0.3	6.9	2.9
Employed in month before arrest						
Yes	61.2%	69.6%	62.3%	72.5%	52.9%	66.6%
No	38.8	30.4	37.7	27.5	47.1	33.4
Sources of income[a]						
Wages	56.7%	65.6%	54.0%	66.4%	62.9%	77.1%
Family/friends	22.0	17.7	20.1	12.3	19.7	15.4
Illegal sources	23.4	27.0	22.5	28.8	19.4	14.4
Welfare	15.4	7.8	13.7	3.9	21.9	12.3
Pension[b]	17.3	4.1	16.5	3.7	18.4	4.9
Compensation payments	3.1	1.9	4.7	1.8	3.0	2.1

[a] Detail sums to more than 100% because offenders may have reported more than one source of income. For prisoners detail includes any income received in the month prior to arrest. For jail inmates, detail includes any income received in the year prior to arrest.
[b] Includes Supplemental Security Income, Social Security, or other pension.

Mental Health and Treatment of Inmates and Probationers 5

Fig. 4: ICDAR2017 POD dataset.

B. Dataset

ICDAR2017 POD dataset is mainly for studying layout analysis. Peking University rigorously labels such data. The official dataset includes 792 document images (contain tables), of which contains 549 images for training and 243 images for testing. The dataset mainly consists of one column and two columns. The types are divided into four categories: text, table, figure and formula. The format of the dataset we used is shown in Fig. 4.

The dataset has complex page layout, more than partial pixel loss. The situation deeply increases the difficulty of table detection. According to our observation, there are three reasons affecting table detection. (1) The interference of background, other three categories (text, figure, formula) should be treated as background. (2) The diversity of tables, tables with lines and tables without lines. (3) The complexity of the page layout, a single column, two columns, list tables and side-by-side tables.

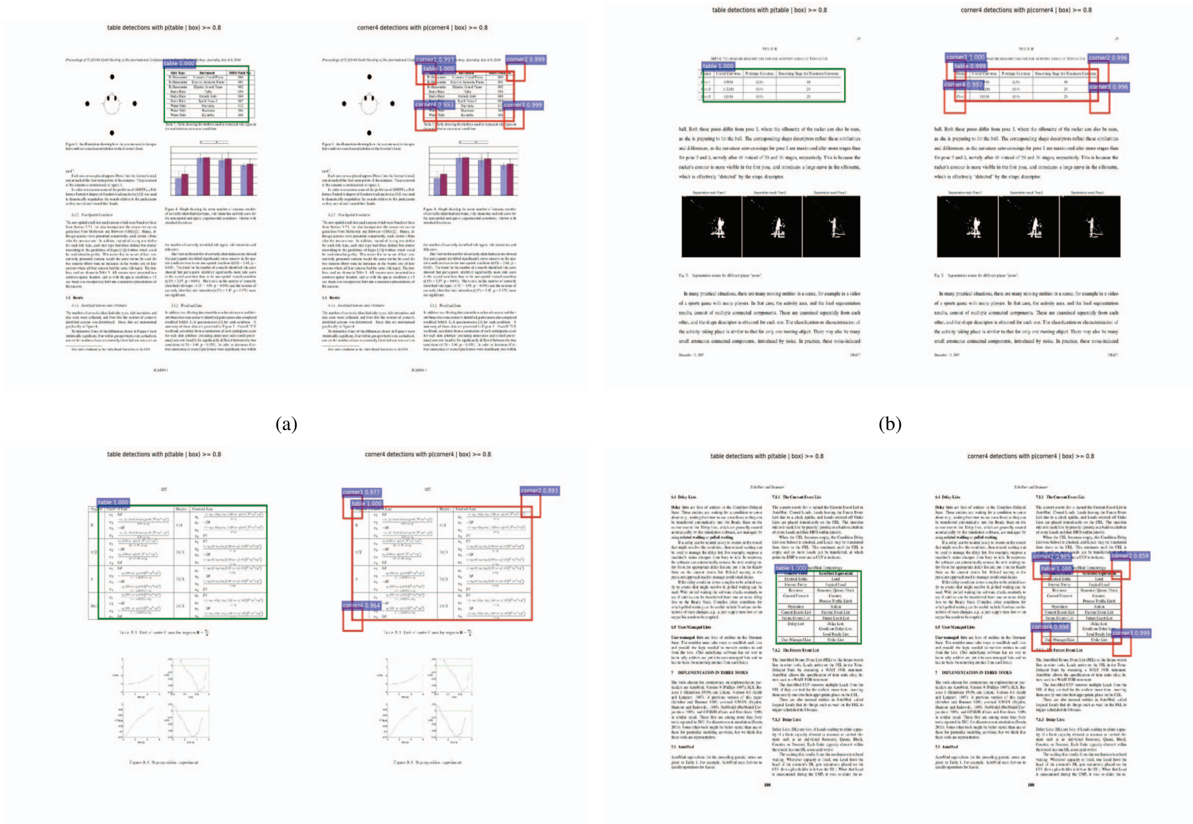
C. Experimental Results

In this section, we carry out ablation studies. Refer to the evaluation metrics of the ICDAR2017 POD competition, IOU is set to 0.6 and the evaluation metrics including precision, recall and F-measure.

Comparison of all results in ICDAR2017 POD dataset. Our method is implemented for comparative experiment. All results are reported in Table 1. Compared with ICDAR2017 POD competition results, it can be seen that our method ranks the second in table detection with F-measure of 94.9%. Faster R-CNN+CCs+CRFs method is complicated, which adds a series of pre-processing steps to improve the results. Such as Connected Component analysis (CCs), Support Vector

TABLE I Comparison of table detection results between the proposed method and ICDAR2017 POD Competition methods.

Method Head	Precision	Recall	F-measure
SSD [12]	0.071	0.959	0.132
CNN [17]	0.230	0.221	0.225
Faster R-CNN(VGG_CNN_M_1024 [11]	0.670	0.940	0.782
Faster R-CNN+edge_based information [11]	0.842	0.890	0.865
Faster R-CNN(VGG16)	0.924	0.918	0.921
Our method	0.943	0.956	0.949
Faster R-CNN+CCs+CRFs [11]	0.968	0.953	0.960



(a)

(b)

(c)

(d)

Fig. 5: Some table detection results on ICDAR2017 POD dataset. (a), (b), (c) and (d) show the comparison of Faster R-CNN method and our method. Color legend: Faster R-CNN(VGG16)method - green; Our method - red.

Machine (SVM) and Conditional Random Fields (CRF). Our method is simple that only combines corner locating to optimize table boundary localization. Our method would achieves better results if it combines with those methods.

Compared with Faster R-CNN method. As a further comparison, Faster R-CNN (VGG16) method and our method were trained for comparative experiment. The evaluation metrics precision, recall and F-measure are the same as Table 1. IOU indicates comparison at pixel-level localization. Comparative results are reported in Table 2. Compared with Faster R-CNN (VGG16) method, the proposed method significantly increased by 2.8% in F-measure and 2.1% at pixel-level localization.

TABLE II Improvement at pixel-level Localization.

Method Head	Precision	Recall	F-measure	IOU
Faster R-CNN(VGG16)	0.924	0.918	0.921	0.811
Our method	0.943	0.956	0.949	0.832

To find out what differences of table detection between Faster R-CNN (VGG16) method and our method, we give some examples in Fig. 5. Obviously, (a) and (d) implies that that our method refines right boundary of the inaccurate tables detected by Faster R-CNN method, (b) and (c) implies that our method refines left boundary of the inaccurate tables detected by Faster R-CNN method.

V. Conclusion

In this paper, We propose a novel table detection method. Our method combines Faster R-CNN and corner locating method. Through table boundary refining via corner locating, the precision of table detection is significantly increasing. Compared with traditional Faster R-CNN method, our method increased by 2.8% in F-measure and 2.1% at pixel-level localization on the ICDAR2017 POD dataset.

The proposed method is also applicable to other datasets. Although the proposed method is quite practical, it is still accompanied with some problems. Firstly, anchor is restricted by the radio and size, which leads to small object proposals (for example: corners) higher miss detection. Secondly, corner size adopts a fixed value, which is also disadvantageous in table boundary refining. We will continue to improve our method in the follow-up work and set adaptive corner size.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 61602345, 61703306) and the Natural Science Foundation of Tianjin(Grant No. 18JCYBJC85000, 16JCQNJC00600). We are grateful for their support.

References

[1] M. Rahman, S. Gustafson, P. Irani, and S. Subramanian, "Tilt techniques: investigating the dexterity of wrist-based input," Proc. SIGCHI Conference on Human Factors in Computing Systems, ACM Press, Apr. 2009, pp. 1943-1952.

[2] P. D. Sankhe, M. Patil, and M. Margaret, "Deblurring of grayscale images using inverse and Wiener filter," Proc. International Conference & Workshop on Emerging Trends in Technology, ACM Press, 2011, pp. 145-148.

[3] G. Antal, R. Martinez, F. Csonka, M. Sbert, and L. Szirmay-Kalos, "Combining global and local global-illumination algorithms," Proc. 19th Spring Conference on Computer Graphics, ACM Press, Apr. 2003, pp. 185-192.

[4] F. Chang, S. Y. Chu, and C. Y. Chen, "Chinese document layout analysis using an adaptive regrouping strategy," Pattern Recognition, 2005, 38(2): 261-271.

[5] J. Xi, J. Hu, and L. Wu, "Page segmentation of Chinese newspapers," Pattern Recognition, 2002, 35(12): 2695-2704.

[6] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," Computer Graphics and Image Processing, 1982, 20(4):375-390.

[7] J. L. Chenxe, "A simplified approach to the HMM based texture analysis and its application to document segmentation," Pattern Recognition Letters, 1997, 18(10): 993-1007.

[8] H. Choi, and R. G. Baraniuk, "Multiscale document segmentation using wavelet-domain hidden Markov models," Proc. SPIE Document Recognition and Retrieval VII, Dec. 1999, vol. 3967, pp. 234-248.

[9] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask R-CNN," Proc. IEEE Symp. International Conference on Computer Vision(ICCV), 2017, pp. 2961-2969.

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

[11] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "ICDAR2017 Competition on Page Object Detection," Proc. 14th International Conference on Document Analysis and Recognition (ICDAR), IEEE Press, Vol. 1, Nov. 2017, pp. 1417-1422.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, and C. Y. Fu, et al. "SSD: Single Shot MultiBox Detector," Proc. European Conference on Computer Vision(ECCV), Springer Press, Cham, Oct. 2016, pp. 21-37.

[13] T. Kieninger, and A. Dengel, "Table recognition and labeling using intrinsic layout features," Proc. International Conference on Advances in Pattern Recognition, London, Springer Press, 1999, pp. 307-316.

[14] T. Kieninger, and A. Dengel, "Applying the T-RECS table recognition system to the business letter domain," Proc. Sixth International Conference on Document Analysis and Recognition, IEEE Press, 2001, pp. 518-522.

[15] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," Proc. Object recognition supported by user interaction for service robots, IEEE Press, 2002, Vol. 3, pp. 236-240.

[16] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis, "Automatic table detection in document images," Proc. International Conference on Pattern Recognition and Image Analysis. Springer Press, Heidelberg, Berlin, Aug. 2005, pp. 609-618.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proc. IEEE Symp. Conference on Computer Vision and Pattern Recognition(CVPR), 2014, pp. 580-587.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," Proc. IEEE Symp. Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916.

[19] R. Girshick, "Fast R-CNN," Proc. IEEE Symp. International Conference on Computer Vision(ICCV), 2015, pp. 1440-1448.

[20] L. Tychsen-Smith, and L. Petersson, "Denet: Scalable real-time object detection with directed sparse sampling," Proc. IEEE Symp. International Conference on Computer Vision(ICCV), 2017, pp. 428-436.

[21] X. Wang, K. Chen, Z. Huang, C. Yao, and W. Liu, "Point linking network for object detection," 2017, arXiv preprint arXiv:1706.03646.

[22] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "multi-oriented scene text detection via corner localization and region segmentation," Proc. IEEE Symp. Conference on Computer Vision and Pattern Recognition(CVPR), 2018, pp. 7553-7563.

[23] H. Law, and J. Deng, "Cornernet: Detecting objects as paired keypoints," Proc. European Conference on Computer Vision(ECCV), 2018, pp. 734-750.