Proceedings of ICSP2020

A Deep Semantic Segmentation Model for Image-based Table Structure Recognition

Yajun Zou, Jinwen Ma
Department of Information and Computational Sciences, School of Mathematical Sciences and LMAM
Peking University, Beijing, China
Email: zouyj@pku.edu.cn, jwma@math.pku.edu.cn

2020 15th IEEE International Conference on Signal Processing (ICSP) | 978-1-7281-4480-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICSP48669.2020.9321003

***Abstract*—Table structure recognition is a crucial step for automatic table information extraction. It is conventional to utilize the features such as ruling lines or words for parsing the rows, columns and cells in a table. However, these conventional methods are ineffective for image-based tables when ruling lines are not visible or the words cannot be recognized through the OCR system. In order to overcome these problems, we propose a deep semantic segmentation model for image-based table structure recognition. Specifically, it is an end-to-end semantic segmentation neural network to determine a pixel-wise prediction map for an input table image where the labels are row separator, column separator, cell content and background. Moreover, by making the connected componnet analysis on the prediction map, we can obtain the bounding boxes of row separators, column separators and cell contents, more accurately. Then we number row/column separators in order by coordinate sorting. Thus, we can make full use of relative positions between row/column separators and cell contents, and further assign the row/column number to each cell. Due to the lack of training data, a large amount of synthetic data are automatically generated in our experiments. It is demonstrated by the experimental results that our proposed model is suitable for various table types, which can achieve 0.9769 and 0.9343 average F1 scores on a generative dataset when the IoU threshold is set to 0.6 and 0.8, respectively.**

***Keywords*—table structure recognition, table information extraction, semantic segmentation and deep learning**

I. Introduction

In our daily working environment, a huge number of documents are generated, and automatic information extraction from these documents is necessary for many tasks of artificial intelligence. As an essential structural unit of the documents, table is widely used in scientific papers, statistical reports and business orders. It provides an effective way to organize and display the data and other information. Therefore, automatic table recognition [1]–[3] has become a popular topic in the fields of document understanding and natural language processing.

In general, table recognition consists of three tasks: table detection, table structure recognition and table content recognition. Table detection starts to parse or locate the tables from a document. Table structure recognition further parses the rows, columns and cells in a recognized table. Table content recognition finally reconizes each cell content which may be text, formula or figure. It should be noted that these three tasks may be defined in different ways. In this paper, we focus on image-based table structure recognition. In fact, while the tables in PDF format can provide words information and the tables in HTML format can parse row information according to their tag sequence, as shown in Fig. 1, our task considers only one table image as the input and determines the structure information of all the cells including their bounding boxes, row numbers and column numbers. As there may be spanning cells in a table, we take two attributes, start-row and end-row, to denote the row number. The column number is denoted in the similar way. In actual documents, tables differ widely from each other. As a table is composed of relevant data, the complexity of the table structure is up to the complexity of the data relation. For instance, some complicated tables may have multi-level headers [4]. Moreover, depending on the table maker, the appearance of a table can be varied in different ways, such as font, color and line thickness. The diversity of table layout makes it difficult to build a general model for table structure recognition.

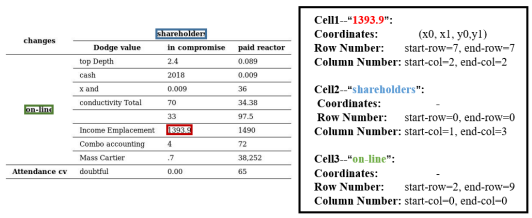


Fig. 1. Description of table structure recognition task.

Most conventional methods [5]–[8] are built on the semantic features such as ruling lines or words, followed by a sequence of predefined heuristic rules. Some of them employ image processing techniques to extract the ruling lines, either visible or invisible. Some of them obtain the location information of words as bounding boxes with the aid of an OCR system. However, on one hand, these methods rely heavily on the extracted features. For instance, the error caused by the OCR system can be accumulated during the subsequent processing. On the other hand, we have to determine many parameters for the heuristic rules by experience while those parameters can hardly fit in the different table styles. Nowadays, with the rapid development of deep learning, DeepDeSRT [9] adopts two deep semantic segmentation networks to parse rows and columns, respectively. But it cannot be applied to the complicated tables where spanning cells occupy at least two

rows or columns. Some recent approaches [10], [11] utilize Graph Convolutional Network to predict the relationship between word (or cell). Nevertheless, word bounding boxes are required in advance. Some other approaches [12], [13] propose image-to-text models to generate HTML tag sequence that represents the arrangement of rows and columns. However, those methods are lack of information of cell coordinates and usually constrained by the length of tag sequence. Moreover, the performances of deep learning based models are generally restricted with the existence of a large scale of annotated data.

In this paper, in order to overcome these problems, we propose a deep semantic segmentation model for image-based table structure recognition. Specifically, we adopt an end-to-end semantic segmentation neural network to determine a pixel-wise prediction map for an input table image in which each pixel is assigned to a concrete semantic meaning like row separator, column separator or cell content. Not like those rule-based methods, we take full advantage of the strong self-learning power of the deep neural network. At the same time, we adopt a post-processing procedure to infer the bounidng boxes of cell contents directly from the prediction map so that we can get rid of the limitations of the OCR system. After the bounding boxes of row separators, column separators and cell contents are determined, we can get the arrangements of rows and columns. We then make full use of relative positions between row/column separators and cell contents to assign row/column number to each cell, whether spanning cells exist or not. To eliminate the negative influence of less annotated training data, we automatically generate enough synthetic data in the experiments, based on which our proposed model can achieve 0.9769 and 0.9343 average F1 scores on a generative dataset when IoU threshold is set to 0.6 and 0.8, respectively.

The rest of the paper is organized as follows. We first review the related work in section II. We then give the detailed description of our proposed model. In section IV, we present the experimental results and comparisons on a generative dataset. We finally make a brief conclusion in section V.

II. Related Work

Table structure recognition has been studied for decades. According to the input file formats, a great number of specific algorithms have been developed. For image-based table structure recognition, early algorithms are mostly rule-based. Some of them [5], [7] use visible lines or continuous white space to find demarcation information. These methods utlize a lot of images processing techniques. Other algorithms [6], [8] often obtain the region of text blocks and design some heuristic rules to merge text blocks into cells, rows and columns. For instance, T-rec system [8] uses a bottom-up approach to form columns from word blocks and further divides columns into cells, with a series of added post-processing steps. In a word, these early algorithms may be valid for specific types of tables, but they have poor generalization ability to fit in diverse table layouts.

Recently, with deep learning has made great breakthrough in computer vision tasks, a series of deep learning based models for table structure recognition have been put forward. Since deep learning based models are data-driven, a large scale of training data is required. However, there are only a few public datasets available, such as UNLV dataset [14], ICDAR 2013 table competition datasets [15], and ICDAR 2019 table competition datasets [16]. There are differences between these datasets. For example, tables in UNLV and ICDAR 2019 datasets are image files while tables in ICDAR 2013 datasets are PDF files. If we use ICDAR 2013 datasets for training, we should first extract image files from original PDF files. What's more, UNLV only identifies rows and columns without definite coordinates for cell content. ICDAR 2019 datasets consist of modern dataset and historical dataset. For the modern dataset, the convex hull of the content describes a cell region whose four attributes indicate its starting row/column and ending row/column. It should be noted that these datasets consist of only a few hundred tables that can hardly meet the requirements of deep learning training. In addition to the above datasets, there are particular annotated datasets which are designated for corresponding task definition, like TableBank dataset [12] and Marmot dataset for table data extraction [17].

To our knowledge, DeepDeSRT [9] model is the first deep learning based attempt to extract table structure. The authors first apply object detection techniques to detect rows and columns in a table. But these techiniques fail to detect rows well because rows are numerous objects in a very confined space as well as the extreme aspect ratio. Then they turn to the fine-grained semantic segmentation techniques and respectively design two networks based on FCN [18] for row segmentation and column segmentation. Though this model has achieved promising results, it ignores the spanning cells which occupy at least two rows or columns. Therefore, the row and column information of spanning cell is not clear. Supposing no spanning cells exist, Shoaib et al. [19] adopt a unified semantic segmentation network for the detection of both rows and columns simultaneously. Unlike DeepDeSRT, their detection of rows and columns shares the same initial feature maps but has two different prediction branches. SPLERGE [20] model is free from spanning cells. The authors use a pair of deep learning model: the split model and the merge model. The split model predicts the basic grid of table, regardless of the spanning cells, while the merge model is responsible for predicting which grid elements should be merged to recover spanning cells. They achieve state-of-the-art performance both in their private dataset and ICDAR 2013 dataset. Besides these CNN based models above, Khan et al. [21] exploit RNN based sequence model to capture the repetitive row/column structures.

There are some algorithms built on the prerequisite that word location is known. Qasim et al. [10] divide the relationships between words into three types: belonging to the same cell/row/column. They take each word as a vertex and construct three graph networks to predict these three relations. And the visual features inside a word bounding box extracted from CNN are fed into the graph network. Similarly, Zhang et al. [11] use graph network to predict adjacent relation between

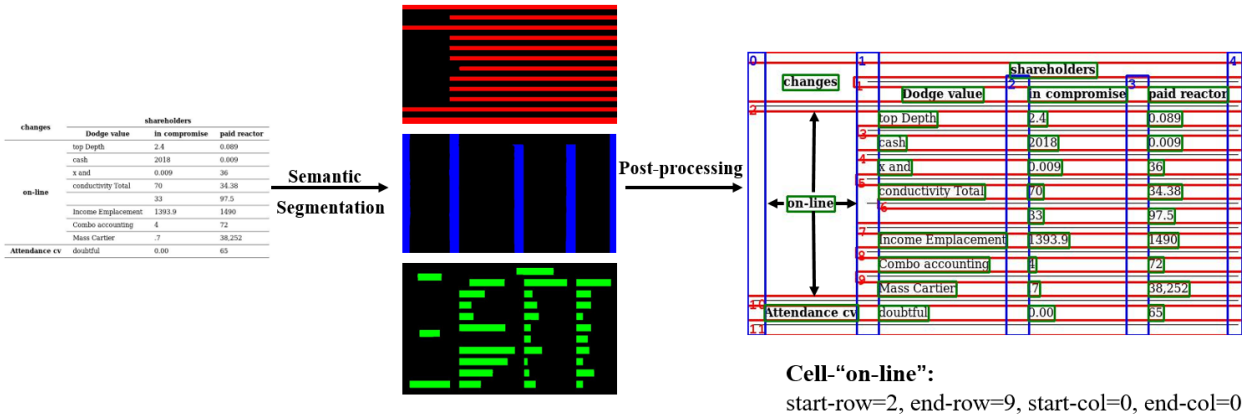


Fig. 2. The pipeline of the proposed deep semantic segmentation model. Red is for row separator. Blue is for column separator. Green is for cell content.

cells. Their models are operated on PDF files so that cell contents and their corresponding bounding box can be obtained in preprocessing steps. Besides, some researchers leverage image-to-text model to generate a sequence of HTML tags for an image-based table. TableBank [12] builds a vocabulary to describe elements like row, cell and header. However, the tag sequence is insufficient to recover the corresponding table structure. It can't get the bounding box of the cell content and corresponding row-span and col-span attributes. Zhong et al. [13] construct a larger vocabulary to identify spanning cells and add a cell decoder to directly recognize cell content. In this way, the bounding box of the cell content is not necessary.

III. Methodology

A. Overview

We begin to introduce our deep semantic segmentation model for image-based table structure recognition, which can be roughly divided into two parts: semantic segmentation part and post-processing part. And the pipeline of our model is demonstrated in Fig. 2. The semantic segmentation part is responsible for identifying the position of row separators, column separators and cell content simultaneously. And a prediction map with three channels is output. During the post-processing procedure, we get the bounding boxes of row separators, column separators and cell contents (marked with red, blue and green rectangular boxes in Fig. 2) by connected component analysis. Then we make full use of relative position between row/column separators and cell contents. Specifically, we number rows and columns (number in red and blue color) according to the coordinates of row separators and column separators. As the arrows shown in Fig. 2, we expand each cell in four directions to fulfil row and column number assignment.

Unlike other methods, our model takes only one table image as input, with no extra meta-data provided like tables in PDF file. In addition, we get rid of the limitations of other commercial systems like OCR because we can infer the position of cell content directly from the prediction map output from the semantic segmentation part. Besides, our post-processing steps are very simple without complex heuristic rules.

B. Semantic Segmentation Network

Being different from other computer vision tasks like image classification and object detection, semantic segmentation is a dense prediction task where every pixel should be labeled. For our specific task, we assign each pixel to three types of semantic meaning: row separators, column separators and cell content. As is shown in Fig. 3, we maximize the size of the separator regions without intersecting any cell contents. And the ground truth region of a row/column separator is rectangular. Meanwhile, pixels inside the bounding box of the cell content are fully regarded as objects. Also, it's worth noting that in a table image, these pixels, which are located in the intersection of ruling lines, belong to row separators as well as column separators. That is, our problem can be formulated as multi-label classification. Specifically, each output channel is designed to distinguish separators or cell contents from the background. At the training time, we choose BCE Loss function to propagate gradient backwards, which is defined as follow:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log x_i + (1 - y_i)\log(1 - x_i) \quad (1)$$

Here, x_i denotes the network output after sigmoid function activation layer. y_i denotes the corresponding label.

As far as we know, FCN [18] is the first introduced end-to-end semantic segmentation network. And deconvolutional layers are proposed to upsample feature maps extracted from a

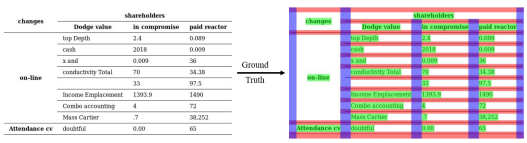


Fig. 3. An example of the ground truth for training the segmentation network. **Red:** row separator. **Blue:** column separator. **Green:** cell content.

series of convolutional and pooling layers. In this way, we can get the same size heat map as the input image. However, there is an inherent conflict between global information and spatial information. Though pooling operators ensure a large receptive field to extract global information, the weakening of spatial information makes it difficult to produce fine segmentation results in the upsampling phase. In our paper, we choose two mainstream architectures: U-Net [22] and DeepLab v2 [23]. As U-Net is based on a typical encoder-decoder structure, the architecture consists of a contracting path and a symmetric expanding path. Concretely, low-layer features from the contracting path are combined with the upsampled output that enables enough spatial information for precise segmentation. While DeepLab v2 uses dilated convolution to enlarge receptive field and retain spatial information at the same time. Besides, it doesn't increase the number of parameters. More importantly, Atrous Spatial Pyramid Pooling (ASPP) that assembles multiple dilated convolutions with different rates are proposed to capture multi-scale information.

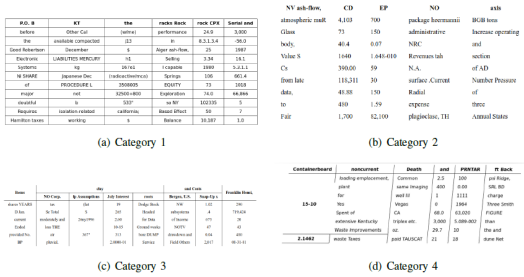
C. Post-Processing Procedure

To obtain the definite regions of row separators, column separators and cell contents, we firstly apply connected component analysis on the prediction map. Then we assign numbers to row separators and column separators according to their horizontal or vertical coordinates. Finally, based on the relative position between separators and cell contents, we determine row and column number for each cell, whether it's a spanning cell or not.

Connected Component Analysis (CCA). To obtain the definite regions of separators and cell contents, we separately extract connected components on the three channels of the prediction map output from our segmentation network. And we take the bounding rectangles to specify the boundary of separators and cell contents. This is shown by the colored rectangles in Fig. 2.

Number row/column separators. Our segmentation network may produce some false positives and false negatives. Thus we firstly drop some bounding boxes of row and column separators according to some heuristic rules. For example, a predicted row/column separator will be filtered out if its length is shorter than a predefined threshold. In addition, some bounding boxes are merged because they are supposed to belong to the same separator. Take column separators as an example, two bounding boxes should be merged if they are close enough vertically and have great overlap horizontally. After these preprocessing steps, we sort row and column separators by their horizontal or vertical coordinates respectively and then assign numbers to row separators and column separators in order. As in Fig. 2, we attach a number to each separator.

Cell expansion. Since the bounding boxes of cell contents have been defined by CCA, we consider making use of relative positions between separators and cell contents to determine row and column number for each cell. As is shown in Fig. 2, each cell is expanded to the nearest parallel separator in four directions. For example, if we expand a cell upwards, when the cell content is almost covered by the nearest row separator vertically, we regard the separator's number as the starting row number. Otherwise, we continue expanding cells upwards until the next nearest row separator is found. Similarly, we can find the corresponding ending row, starting column and ending column for each cell. Finally we make sure that there is no intersecting cell in our table structure. Any two cells that span the same row or column should be merged.



(a) Category 1 (b) Category 2

(c) Category 3 (d) Category 4

Fig. 4. Table images in different categories [10].

IV. Experiments

A. Dataset

Since deep semantic segmentation model is data-driven, a large amount of annotated data is essential for training. However, as we mentioned in Section II, real-world data on a large scale is not publicly available. Therefore, we firstly generate a synthetic dataset with the aid of the open code [10]. The open code is used to generate tables as real as possible. All of the content are extracted from UNLV dataset. Besides, different table border types are taken into consideration. And cells can be randomly selected as blank cells or spanning cells. As is shown in Fig. 4, the generated tables can be divided into four categories. Specifically, tables in Category 1 are full-lined without any spanning cells while tables with the occasional absence of ruling lines are in Category 2. Besides, spanning cells are introduced to tables in Category 3. Finally, linear perspective transform is applied to table images in Category 4 to model camera captured images.

To generate the ground truth for our segmentation network, the bounding box of cell content, the corresponding starting row/column number and the ending row/column number are output after we make slight modification on the open code. It's should be noticed that we restrict the images in Category 4 to ensure that vertical spacing always exists between columns. Finally, we build a training dataset of 100,000 tables, a validation dataset of 2,000 tables and a test dataset of 4,000 tables. The number of each category is equivalent either for training or for testing.

B. Metric

Since our segmentation network outputs a dense prediction map, pixel-wise IoU as a standard metric for segmentation task is used to evaluate our segmentation results of each class. Also, we calculate the mean intersection over union on all classes called Mean IoU. As for structure recognition task,

TABLE I
THE SEGMENTATION RESULTS OF OUR PROPOSED METHODS.

Network	Category	Row Separator	Column Separator	Cell Content	**Mean IoU**
U-Net	Category 1	0.9992	0.9379	0.9807	0.9726
	Category 2	0.9936	0.8635	0.9796	0.9456
	Category 3	0.9987	0.8883	0.9812	0.9561
	Category 4	0.8330	0.8057	0.9483	0.8623
	Average	0.9561	0.8739	0.9725	-
DeepLab v2	Category 1	0.9967	0.9870	0.9614	0.9817
	Category 2	0.9960	0.9872	0.9596	0.9809
	Category 3	0.9969	0.9873	0.9654	0.9832
	Category 4	0.8983	0.9516	0.9163	0.9221
	Average	0.9720	0.9783	0.9507	-

TABLE II
THE STRUCTURE RECOGNITION RESULTS OF OUR PROPOSED METHODS WITH IOU THRESHOLD OF 0.6.

Network / Category	U-Net			DeepLab v2		
	Precision	Recall	F1	Precision	Recall	F1
Category 1	0.9894	0.9756	0.9824	0.9689	0.9567	0.9628
Category 2	0.9853	0.9689	0.9770	0.9679	0.9546	0.9612
Category 3	0.9845	0.9689	0.9766	0.9737	0.9589	0.9662
Category 4	0.9815	0.9619	0.9716	0.9210	0.8815	0.9008
Average	-	-	**0.9769**	-	-	0.9478

TABLE III
THE STRUCTURE RECOGNITION RESULTS OF OUR PROPOSED METHODS WITH IOU THRESHOLD OF 0.8.

Network / Category	U-Net			DeepLab v2		
	Precision	Recall	F1	Precision	Recall	F1
Category 1	0.9501	0.9369	0.9435	0.8262	0.8159	0.8210
Category 2	0.9456	0.9298	0.9376	0.8366	0.8252	0.8308
Category 3	0.9444	0.9294	0.9368	0.8369	0.8242	0.8305
Category 4	0.9284	0.9098	0.9191	0.7809	0.7474	0.7638
Average	-	-	**0.9343**	-	-	0.8115

we use the metric based on adjacency relations between cells [24]. Because our task is almost the same as TRACK B.2 of ICDAR 2019 Table Competition [16], we directly adopt their official evaluation tool to evaluate our recognition results. They firstly identify valid cells by cell mapping. A predicted cell is regarded as a valid cell if the highest IoU between it and the ground truth cells is greater than a threshold. Then a 1-D list of adjacency relations between valid cells and their nearest neighbors in horizontal and vertical directions is generated, where neighbors can be defined by the row and column information. And it should be noticed that blank cells are not taken into consideration. For a predicted relation and a ground truth relation, the predicted relation is marked as a true positive if their corresponding cells are identical and directions are matching. Then we can calculate the precision, recall and F1 measure over relations.

C. Experimental Results and Comparisons

In our experiments, we choose U-Net and DeepLab v2 as our segmentation networks respectively. Because the size of the table images in the dataset varies, bilinear interpolation technique is applied to resize images to the size of 512*512, which is the input size of both networks. At the training time, pthe batch size is set to 4. And the initial learning rate is set to 0.001, which is decreased by a factor of 10 after 10 epochs. Besides, we take the Stochastic Gradient Descent algorithm as our optimizer. The maximum ste of iteration is 20 epochs. We finally choose the model that has the highest Mean IoU on validation dataset for inference.

The overall segmentation results on our test set are demonstrated in TABLE I. To some degree, our segmentation networks, both U-Net and DeepLab architectures can identify the pixels that belong to different classes in a table image. In comparison with DeepLab, U-Net has a better performance on cell content class. The opposite conclusion can be drawn on the row/column separator class. Especially for column separator class, DeepLab improves the IoU from 0.8739 to 0.9783. Also, DeepLab achieves better mean IoU than U-Net for all table types. Moreover, the segmentation performance is relevant

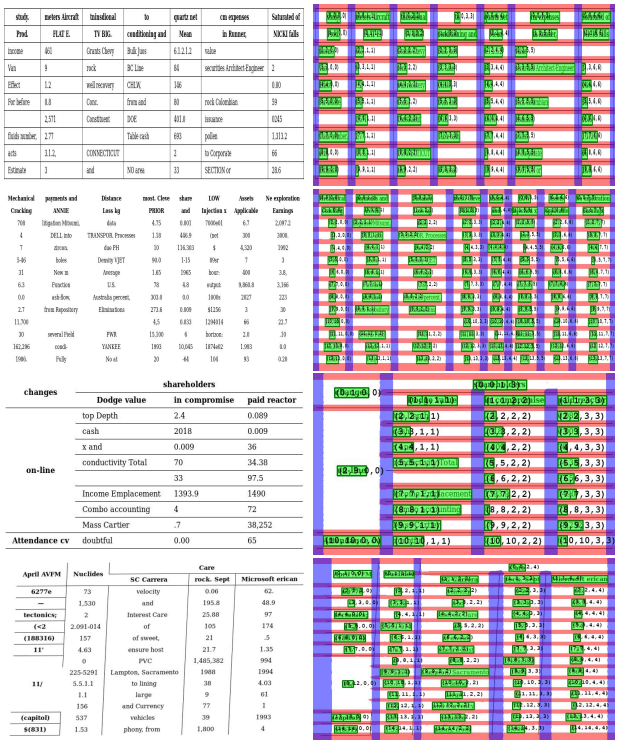
to the table type. As we can see, the segmentation results of tables in Category 1 achieve the best mean IoU in both networks. This is easy to explain because the ruling lines of tables in Category 1 are all visible, which is a distinct feature for identifying row/column separators. Besides, the absence of spanning cells makes the segmentation easier. It's harder to segment tables with spanning cells or occasional absence of ruling lines, which explains why the mean IoU of both Category 2 and Category 3 are lower than that of Category 1. Meanwhile, tables in Category 4 are the most difficult to perform segmentation restricted by the linear perspective transform applied on them. The area of vertical spacing between columns is so small that the segmentation of column separators is under performance. The lowest IoU for column separators in both networks can account for this.

We apply our post-processing steps to the segmentation results to get the final structure recognition results. The final evaluation results are shown in TABLE II and TABLE III. Specifically, the IoU threshold is set to 0.6 in TABLE II and 0.8 in TABLE III. As we mentioned above, the IoU threshold limits the overlap between the bounding boxes of a predicted cell and corresponding ground truth cell. As we can see, the F1 measure is higher in TABLE II than that of TABLE III for any table type and network architecture. In each table, U-Net achieves higher F1 measure than DeepLab over any table category. It can be concluded from that U-Net has better performance than DeepLab on the segmentation of cell content class, as shown in TABLE I. It is the key step to expand the bounding box of the cell content in our post-processing procedure. As is shown in TABLE III, in both networks, there is an apparent decline on the F1 measure in Category 4. And this is consistent with the segmentation results. However, not like the segmentation results, there is no obvious difference between Category 1, 2 and 3 for the recognition results, which proves our post-processing algorithm is robust. When compared with the method in [10], our proposed method can still get considerable results on tables in Category 3 and Category 4. Here, we compute the average F1 measure over four categories. When IoU threshold is set to 0.6 and 0.8, our model can achieve 0.9769 and 0.9343 average F1 score respectively.

Moreover, we demonstrate some visualization results in Fig. 5. Images from top to bottom come from four categories respectively. The segmentation and recognition results are shown simultaneously. Then we use different colors to represent the results of pixel segmentation. The recognition results are represented by a green box and four numbers in brackets. For each cell, a green box denotes the bounding box of the cell content. The four numbers in brackets denote the attributes of start-row, end-row, start-col and end-col respectively. As we can see, our proposed model can achieve great results for a variety of table types.

D. Limitation

Furthermore, we directly take the model trained on our synthetic dataset to evaluate the performance on the modern



(a) Inputs (b) Results

Fig. 5. The visualization results of our proposed methods. **Left**: Inputs. **Right**: Results. The results of pixel segmentation are represented by different clolors. And for each cell, a green box denotes the bounding box of the cell content. The four numbers in brackets denote the attributes of start-row, end-row, start-col and end-col respectively.

dataset of ICDAR 2019 Table competition. That is, with no extra training data provided to finetune our model, we compare our results with other submitted results in TABLE IV. And it should be noticed that the submitted results need to locate table regions from documents at first. There are only two participants and the top result can only achieve F1 measure of 0.3650 when IoU threshold is set to 0.6 [25]. Though we make great improvement over Team HCL, there is still a clear gap compared with the top result. It's worth noting that their approaches have not been presented by academic papers. Through analysis, we find our model performs well for tables that are similar to synthetic data. However, real-world data is more complex than synthetic data. The competition dataset has greater diversity whose tables are selected from documents with different sources and languages. And a detailed description can be found in [16]. Therefore, it's quite necessary to finetune our segmentation model over abundant real-world data. And in the competition, the convex hull of the content describes a cell region while a bounding rectangle is used in our model, which also affects our results. We leave these for future work.

V. Conclusion

We have established a deep semantic segmentation model for image-based table structure recognition. In fact, the deep

TABLE IV
RESULTS ON ICDAR 2019 TABLE COMPETITION. OURS@X DENOTES OUR MODEL BASED ON X NETWORK.

Team / IoU	@IoU=0.6			@IoU=0.8		
	Precision	Recall	F1	Precision	Recall	F1
NLPR-PAL	0.3224	0.4206	0.3650	0.1722	0.2246	0.1950
Ours@U-Net	0.1879	0.1007	0.1311	0.0171	0.0092	0.0119
Ours@DeepLab	0.1496	0.0647	0.0904	0.0180	0.0078	0.0109
HCL IDARON	0.0017	0.0010	0.0013	0.0003	0.0002	0.0002

semantic segmentation model can determine the pixel-wise prediction map for any input table image with the labels of each pixel such as row separator, column separator, cell content and background. We then use relative positions between row/column separators and cell contents to assign row/column number for each cell. Unlike the other methods, our prposed model only takes one table image as the input without any additional information. Moreover, our adopted post-processing procedure is very simple without complex heuristic rules. As a result, our proposed model is suitable for various table styles, regardless of tables without ruling lines, tables with spanning cells or tables with linear perspective transform. When IoU threshold is set to 0.6 and 0.8, our proposed model can achieve 0.9769 and 0.9343 average F1 scores, respectively.

ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China under grant 2018AAA0100205.

REFERENCES

[1] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "Learning to detect tables in scanned document images using line information," in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 1185–1189.

[2] 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)*, 2016, pp. 287–292.

[3] S. F. Rashid, A. Akmal, M. Adnan, A. A. Aslam, and A. Dengel, "Table recognition in heterogeneous documents using machine learning," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 777–782.

[4] J. Fang, P. Mitra, Z. Tang, and C. L. Giles, "Table header detection and classification," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012, pp. 599–605.

[5] J. C. Handley, "Table analysis for multiline cell identification," in *Document Recognition and Retrieval VIII*, P. B. Kantor, D. P. Lopresti, and J. Zhou, Eds., vol. 4307, 2001, pp. 34–43.

[6] K. Itonori, "Table structure recognition based on textblock arrangement and ruled line position," in *Proceedings of 2nd International Conference on Document Analysis and Recognition*, 1993, pp. 765–768.

[7] S. Chandran and R. Kasturi, "Structural recognition of tabulated data," in *Proceedings of 2nd International Conference on Document Analysis and Recognition*, 1993, pp. 516–519.

[8] T. Kieninger, "Table structure recognition based on robust block segmentation," in *Document Recognition V*, vol. 3305, 1998, pp. 22–32.

[9] 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 *2017 14th IAPR International Conference on Document Analysis and Recognition*, vol. 01, 2017, pp. 1162–1167.

[10] S. R. Qasim, H. Mahmood, and F. Shafait, "Rethinking table recognition using graph neural networks," in *2019 International Conference on Document Analysis and Recognition*, 2019, pp. 142–147.

[11] Z. Chi, H. Huang, H. Xu, H. Yu, W. Yin, and X. Mao, "Complicated table structure recognition," *arXiv preprint arXiv:1908.04729*, 2019.

[12] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li, "Tablebank: Table benchmark for image-based table detection and recognition," *arXiv preprint arXiv:1903.01949*, 2019.

[13] X. Zhong, E. ShafieiBavani, and A. Jimeno-Yepes, "Image-based table recognition: data, model, and evaluation," *arXiv preprint arXiv:1911.10683*, 2019.

[14] F. Shafait, "Table ground truth for the uw3 and unlv datasets (dfki-tgt-2010)," http://tc11.cvc.uab.es/datasets/DFKI-TGT-2010_1.

[15] M. Gbel, T. Hassan, E. Oro, and G. Orsi, "Icdar 2013 table competition," in *2013 12th International Conference on Document Analysis and Recognition*, 2013, pp. 1449–1453.

[16] L. Gao, Y. Huang, H. Djean, J. Meunier, Q. Yan, Y. Fang, F. Kleber, and E. Lang, "Icdar 2019 competition on table detection and recognition (ctdar)," in *2019 International Conference on Document Analysis and Recognition*, 2019, pp. 1510–1515.

[17] S. S. Paliwal, V. D, R. Rahul, M. Sharma, and L. Vig, "Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images," in *2019 International Conference on Document Analysis and Recognition*, 2019, pp. 128–133.

[18] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, 2017.

[19] S. A. Siddiqui, P. I. Khan, A. Dengel, and S. Ahmed, "Rethinking semantic segmentation for table structure recognition in documents," in *2019 International Conference on Document Analysis and Recognition*, 2019, pp. 1397–1402.

[20] C. Tensmeyer, V. I. Morariu, B. Price, S. Cohen, and T. Martinez, "Deep splitting and merging for table structure decomposition," in *2019 International Conference on Document Analysis and Recognition*, 2019, pp. 114–121.

[21] S. A. Khan, S. M. D. Khalid, M. A. Shahzad, and F. Shafait, "Table structure extraction with bi-directional gated recurrent unit networks," in *2019 International Conference on Document Analysis and Recognition*, 2019, pp. 1366–1371.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, Proceedings, Part III*, vol. 9351, 2015, pp. 234–241.

[23] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[24] M. C. Göbel, T. Hassan, E. Oro, and G. Orsi, "A methodology for evaluating algorithms for table understanding in PDF documents," in *ACM Symposium on Document Engineering, DocEng '12*, 2012, pp. 45–48.

[25] H. Djean, J. Meunier, L. Gao, Y. Huang, Y. Fang, F. Kleber, and E. Lang, "Icdar 2019 competition on table detection and recognition (ctdar)," http://sac.founderit.com/results.html.