Rethinking Table Structure Recognition Using Sequence Labeling Methods

Yibo Li[1,2], Yilun Huang[1], Ziyi Zhu[1], Lemeng Pan[3], Yongshuai Huang[3],
Lin Du[3], Zhi Tang[1], and Liangcai Gao[1](✉)

[1] Wangxuan Institute of Computer Technology, Peking University, Beijing, China
{yiboli,huangyilun,1800012988,tangzhi,glc}@pku.edu.cn
[2] Center for Data Science, Peking University, Beijing, China
[3] Huawei AI Application Research Center, Huawei, China
{panlemeng,huangyongshuai,dulin09}@huawei.com

Abstract. Table structure recognition is an important task in document analysis and attracts the attention of many researchers. However, due to the diversity of table types and the complexity of table structure, the performances of table structure recognition methods are still not well enough in practice. Row and column separators play a significant role in the two-stage table structure recognition and a better row and column separator segmentation result can improve the final recognition results. Therefore, in this paper, we present a novel deep learning model to detect row and column separators. This model contains a convolution encoder and two parallel row and column decoders. The encoder can extract the visual features by using convolution blocks; the decoder formulates the feature map as a sequence and uses a sequence labeling model, bidirectional long short-term memory networks (BiLSTM) to detect row and column separators. Experiments have been conducted on PubTabNet and the model is benchmarked on several available datasets, including PubTabNet, UNLV ICDAR13, ICDAR19. The results show that our model has a state-of-the-art performance than other strong models. In addition, our model shows a better generalization ability. The code is available on this site (www.github.com/L597383845/row-col-table-recognition).

Keywords: Table structure recognition · Encoder-decoder · Row and column separators segmentation · Sequence labeling model

1 Introduction

Table structure recognition is an important task in document analysis. Tables are commonly found in research papers, books, invoices, and financial documents. A Table contains structured information with the arrangement of rows and columns. The manual extraction of structured information is often a tedious and time-consuming process. Therefore, extracting table structure automatically has attracted many researchers' attention in recent years.

J. Lladós et al. (Eds.): ICDAR 2021, LNCS 12822, pp. 541–553, 2021.
https://doi.org/10.1007/978-3-030-86331-9_35

However, due to the diversity of table types and the complexity of table structure, table structure recognition is not a easy problem to solve. Different background colors of the table, the absence of some ruling lines, and the existence of row-span or column-span cells are all challenges to the table structure recognition. Besides, text content of the cell may be centered, left-aligned, or right-aligned, which influences some methods' performance.

In the past few decades, many table recognition methods have been proposed. These proposed methods can be divided into two categories, end-to-end methods and two-stage methods. Chris et al. [18] and Saqib et al. [10] proposed that table structure recognition could be divided into splitting stage and merging stage. The splitting stage predicts the row and column separators and the merging stage predicts which grid elements should be merged to recover cells that span multiple rows or columns. Chris et al. [18] presented Split, which uses a convolution network with novel projection pooling to get the rows and columns segmentation. But this model is not stable in some datasets. Saqib et al. presented BiGRU, which uses image pre-processing and two BiGRU networks to get row and column separators segmentation. However, the performance of BiGRU is limited to the result of the pre-processing stage.

In this paper, we propose a novel encoder-decoder model to recognize row and column separators. The encoder uses a convolution block that contains several dilated convolutions to extract the visual feature. A $H \times W$ image will be converted to a $H \times W$ feature map after the encoder. The feature map is formulated as a sequence of length H or a sequence of length W, so the row and column separators segmentation task can be solved by sequence labeling methods. The encoder is followed by two parallel BiLSTM branches for 1) Segmentation of the row separators and 2) Segmentation of the column separators. Besides, the results of different sequence labeling methods have been compared in the experiments.

We have trained our model on the PubTabNet-Train [25] dataset and evaluated its performance on several available datasets, including PubTabNet-Val, UNLV [16], ICDAR13 [5] and ICDAR19 [4], demonstrating that our approach outperforms other methods in row and column separators segmentation marginally. The code will be publicly released to GitHub.

In summary, the primary contributions of this paper are as follows:

1) The segmentation of row and column separators is firstly formulated as a sequence labeling problem.
2) A unified encoder-decoder architecture for the segmentation of both row and column separators simultaneously using convolutional and recursive neural networks is proposed in this paper and achieved a state-of-the-art performance on the publicly available PubTabNet, UNLV, ICDAR13, and ICDAR19 table structure recognition datasets.

The rest of the paper is organized as follows: Sect. 2 provides the overall of the related work on table structure segmentation. Section 3 describes the details of our method. Section 4 outlines the details about four datasets, how to use the annotation, and the evaluation metric. Section 5 will outline experiment details and results. Finally, the conclusion and future work are presented in Sect. 6.

2 Related Work

2.1 Table Structure Recognition

In the beginning, some researchers used heuristics-based methods to solve this task. T-Recs system [3,11,12], proposed by Kieninger et al. is one of the earliest works to extract tabular information. This system takes words bounding boxes as input. It groups words into columns by their horizontal ruling lines and divides them into cells based on column margins. Then, Wang et al. [20] proposed a system that relied on probability optimization to tackle the table structure understanding problem. Jianying et al. [8] used hierarchical clustering for column detection. Their system uses lexical and spatial criteria to classify headers of tables. Chen et al. [1] used Min-Cut/Max-Flow algorithm to decompose tabular structures into 2-D grids of table cells.

With the development of deep learning, neural networks have been successfully applied in various tasks [22–24]. Thus, some works try to utilize neural networks to solve the table structure recognition. Currently, most of them are two-stage methods and can be divided into cell detection and extraction of cell relationships. Shah et al. [14] used OCR technology to get the bounding box of cells and used CNN to extract the feature of every cell. Then a graph neural network was applied to classify the relationship between two cells. Devashish et al. [13] proposed an automatic table recognition method for interpretation of tabular data in document images, CascadeTabNet. It is a Cascade mask Region-based CNN High-Resolution Network (Cascade mask R-CNN HRNet) based model that detects the regions of tables and recognizes the structural body cells from the detected tables at the same time. Then it uses a rule-based method to recover the table structure. TabStruct-Net, proposed by Sachin et al. [15], is an approach for table structure recognition that combines cell detection and interaction modules to localize the cells and predicts their row and column associations with other detected cells. Structural constraints are incorporated as additional differential components to the loss function for cell detection. Shoaib et al. [17] proposed DeepTabStR, which uses a deformable convolution to detect rows, columns and cells at the same time. But it doesn't perform well on tables that span rows or columns.

Chris et al. [10] and Saqib et al. [18] proposed that table structure recognition could be divided into splitting stage and merging stage. The splitting stage predicts the row and column separators and the merging stage predicts which grid elements should be merged to recover cells that span multiple rows or columns. The former uses a convolution network with novel projection pooling to get the rows and columns segmentation, and the latter uses two BiGRU networks to solve it.

2.2 Sequence Labeling

Sequence labeling is a classical problem in the field of natural language processing (NLP), various neural models have been introduced to solve it. The proposed neural models usually contain three components: word embedding layer, context encoder layer, and decoder layer. In this paper, the convolutional block will be used to extract the feature map which can be regarded as the word embedding; the linear layer followed by the softmax layer is the decoder layer. Therefore, the detailed infomation will be introduced in the following paragraphs.

Recurrent Neural Networks (RNNs) are widely employed in NLP tasks due to their sequential characteristic, which is aligned well with the language. Specifically, bidirectional long short-term memory network (BiLSTM) [7] is one of the most widely used RNN structures. BiLSTM and Conditional Random Fields (CRF) were applied to sequence labeling tasks in [9]. Owing to BiLSTM's high power to learn the contextual representation of sequence, it has been adopted by the majority of sequence labeling models as the encoder. Cho et al. [2] proposed Gated Recurrent Unit (GRU) to solve Machine Translation problem. As a variant of LSTM, GRU combines forget gate and input gate into a single update gate. The GRU model is simpler than the standard LSTM model and is a very popular RNN structure.

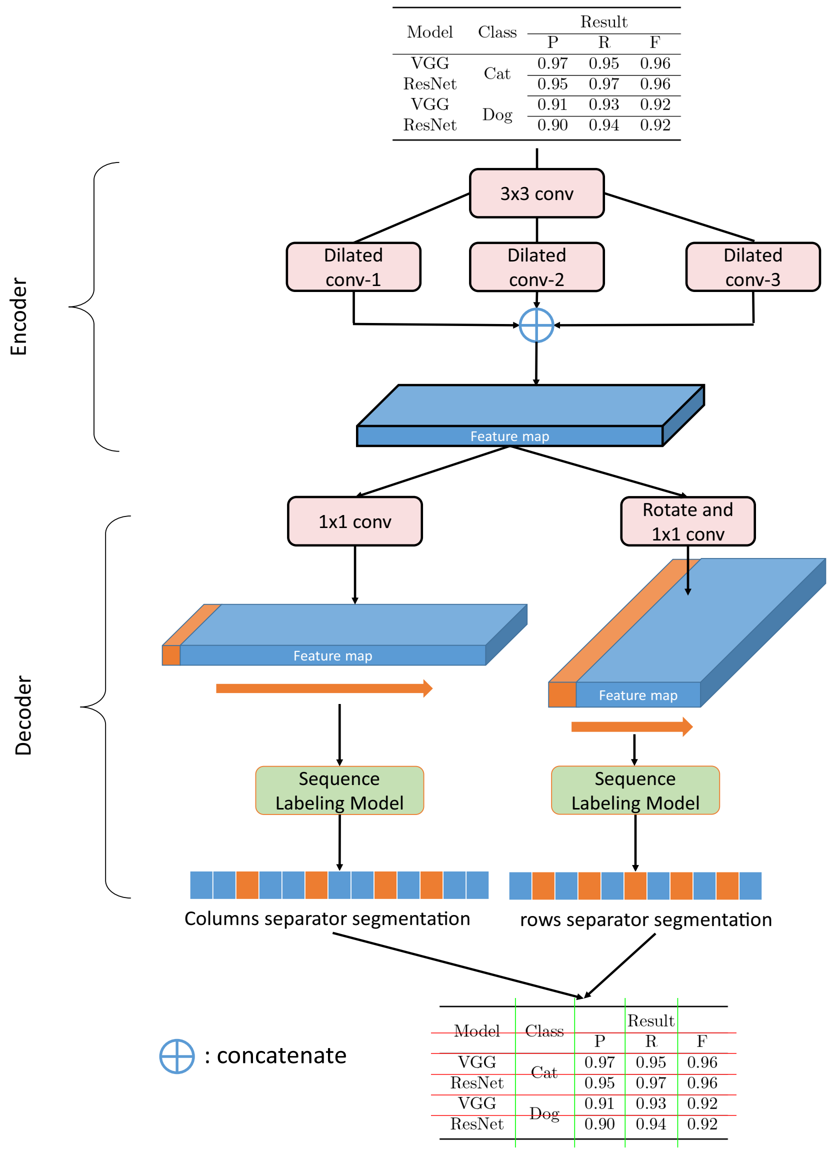
Recently, Transformer began to prevail in various NLP tasks, like machine translation, language modeling and language pretraining models [19]. The Transformer encoder adopts a fully-connected self-attention structure to model the long-range context, which is the weakness of RNNs. Moreover, Transformer has better parallelism ability than RNNs. Some researchers began to use Transformer to solve sequence labeling tasks. Guo et al. [6] tested the performance of transformer on sequence labeling firstly. Then, Yan et al. [21] proved that Transformer lost the directivity of the sequence and proposed an Adapting Transformer Encoder to improve the performance of Transformer on solving sequence labeling problems.

3 Method

We formulate the problem of table row and column separators segmentation as a sequence labeling problem. An $H \times W$ image can be regarded as a sequence of length H with W features when segmenting the row separators or as a sequence of length W with H features when segmenting the column-separators. The proposed model, illustrated in Fig. 1 which contains an encoder and two paralleled decoders. This model takes a table image as input and outputs the basic row separators and column separators segmentation.

3.1 Encoder

The encoder is responsible for extracting the visual features of the image. There is a common 3×3 convolution at the beginning. Then we use several 3×3



Model	Class	Result		
		P	R	F
VGG	Cat	0.97	0.95	0.96
ResNet		0.95	0.97	0.96
VGG	Dog	0.91	0.93	0.92
ResNet		0.90	0.94	0.92

Model	Class	Result		
		P	R	F
VGG	Cat	0.97	0.95	0.96
ResNet		0.95	0.97	0.96
VGG	Dog	0.91	0.93	0.92
ResNet		0.90	0.94	0.92

Fig. 1. The architecture of our method

dilation convolution operators with $[1, 2, 5]$ dilated rates and $[1, 2, 5]$ padding size to enlarge the receptive field of each pixel. After this, a concatenation operator is applied to fusion features. Using such a model, an image can be converted to a feature map. The encoder can be represented as:

$$F = \varphi(x) \tag{1}$$

where x represents the input whose size is $H \times W$, φ denotes the encoder and F represents the $H \times W \times C$ feature map.

3.2 Decoder

The decoder contains a convolution block and a sequence labeling model. The convolution block uses a convolution operator with a 1×1 kernel. This block is used to fusion feature and compress the feature to one channel. It can be presented as:

$$S = Conv(F). \tag{2}$$

where S represents the feature whose shape is $H \times W$. Then S can be regarded as a sequence $[s_0, s_1, s_2, \cdots, s_{H-1}]$ with feature of length W or a sequence $[s_0, s_1, s_2, \cdots, s_{W-1}]$ with feature of length H. A sequence labeling model (SLM) is followed to label separators. The output of SLM is a sequence which has the same length as input sequence. It can be calculated as:

$$Result = SLM([s_0, s_1, s_2, \cdots, s_{H-1}]) \tag{3}$$

We use BiLSTM as the sequence labeling model. The output of a LSTM cell h_t can be represented as:

$$h_t = LSTMCell(h_{t-1}, s_t) \tag{4}$$

Therefore, for the input $[s_0, s_1, s_2, \cdots, s_{H-1}]$, the output of LSTM is $[h_0, h_1, h_2, \cdots, h_{H-1}]$. To get the left feature and right feature, the bi-direction LSTM is applied. Then a Linear Layer and a information operator are used to get the final sequence labeling result. It can be represented as:

$$y = BiLSTM(S) = [h_0, h_1, h_2, \cdots, h_{H-1}] \tag{5}$$

$$Result = Softmax(Linear(y)) \tag{6}$$

Where y represents the result of BiLSTM whose size is $H \times 2 * hidden\_size$. Because there are two labels, "Separator" and "No-Separator", *Result* represents the sequence labeling result with the size of $H \times 2$.

4 Data Preparation and Evaluation Metric

We conduct experiments on PubTabNet [25] and evaluate our model on ICDAR13 [5], ICDAR19 [4], and UNLV [16]. However, the annotation of these datasets is not suitable for the method. Therefore, we convert the annotations to the appropriate format. And we use the evaluation metric represent in [10]. The details are as follows:

4.1 Data Preparation

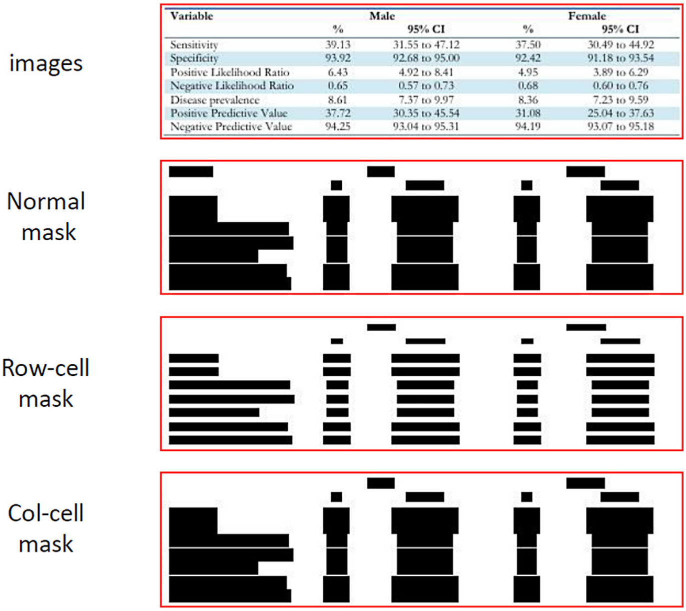
Four datasets are used in the experiments, including PubTabNet, UNLV, ICDAR13 and ICDAR19. The summary of these datasets is on Table 1.

Table 1. Summary of datasets used in our experiments

DataSet	Usage	Numbers
PubTabNet (training set)	Train	500,777
PubTabNet (validation set)	Test	9,115
UNLV	Test	557
ICDAR13	Test	225
ICDAR19	Test	145

Images in PubTabNet are extracted from the scientific publications included in the PubMed Central Open Access Subset (commercial use collection). It contains heterogeneous tables in both image and HTML format. The HTML representation encodes both the structure of the tables and the content in each table cell. Position (bounding box) of table cells is also provided to support more diverse model designs. There are $500,777$ images in the training set and $9,115$ images in the validation set. We decode the cell information including bounding box, row and column position with row and column spans from the HTML and JSON annotation. Then we compute the index of row-separators and column-separators from the cell information. When obtaining the row separator of the table, for an image of size $[H, W]$, we create a matrix M of the same size and set all the values in it to 1. Then, set the corresponding position of the cells that does not span rows to 0. When obtaining the column-separators of the table, we will set the corresponding position of the cell that does not span columns to 0. The index of all 1 rows and columns in the M matrix is the index of the row and column separator. A visualized result is shown in Fig. 2. However, considering that the bounding box of the cell marked by PubTabNet has some overlaps (the lower bound of the upper cell is greater than the previous one of the lower cell in the adjacent two rows of cells), so when we calculate the row separator, we will reduce the lower bound of every cell.

The ICDAR13 and ICDAR19 datasets present the table position, cell bounding boxes and their [“$start - row$”,“$start - col$”,“$end - col$”,“$end - row$”]. The ICDAR13 dataset includes a further collection of EU and US documents. The modern dataset in ICDAR19 comes from different kinds of PDF documents such as scientific journals, forms, financial statements, etc. They are both documents images. In order to adapt to this method, we extract the image of a single table from the document images. We compute the row-span and column-span of every cell and get the ground truth by using the same algorithm as the PubTabNet. There are 225 table images in the ICDAR13 dataset and 145 table images in the ICDAR19 dataset.



Variable	Male		Female	
	%	95% CI	%	95% CI
Sensitivity	39.13	31.55 to 47.12	37.50	30.49 to 44.92
Specificity	93.92	92.68 to 95.00	92.42	91.18 to 93.54
Positive Likelihood Ratio	6.43	4.92 to 8.41	4.95	3.89 to 6.29
Negative Likelihood Ratio	0.65	0.57 to 0.73	0.68	0.60 to 0.76
Disease prevalence	8.61	7.37 to 9.97	8.36	7.23 to 9.59
Positive Predictive Value	37.72	30.35 to 45.54	31.08	25.04 to 37.63
Negative Predictive Value	94.25	93.04 to 95.31	94.19	93.07 to 95.18

Fig. 2. The mask of cells for row and column separators

The UNLV dataset consists of a variety of documents, including technical reports, business letters, newspapers, and magazines. The dataset contains 2,889 scanned documents, 403 of which contain tables. It presents the coordinate of the table and its basic row and column separators. Thus, we can extract the table in the images directly by using the annotation. There are 557 table images in total.

4.2 Evaluation Metric

Various researchers have used different evaluation metrics ranging from simple precision and recall to more complex evaluation algorithms. In this paper, we use the methodology proposed by Shahab et al. [16]. Saqib et al. [10] detailed the methodology when benchmarking the row or column segmentation. This metric contains six values, including Correct Detections, Partial Detections, Missed Detections, Over Segmented Detections, Under Segmented Detections, and False Position Detections. The Correct Detections shows the total number of ground truth segments that have a large intersection with a detected segment and the detected segment does not have a significant overlap with any other ground truth segment. It is the most important one among the six measures.

5 Experiments and Result

Table 2. The row and column segmentation results on PubTabNet dataset by using different sequence labeling models.

	SLM	Accuracy %					
		Correct	Partial	Missed	Over-Seg	Under-Seg	False-Pos
Row	BiLSTM (ours)	**95.23**	0.63	0.03	1.93	2.49	2.25
	BiGRU	93.26	0.73	0.05	3.14	2.77	3.17
	Transformer [19]	44.79	8.15	0.33	29.10	20.48	2.22
	Ada transformer [21]	89.39	0.49	0.01	5.12	2.75	4.61
Column	BiLSTM (ours)	**97.39**	0.37	0.01	1.32	1.23	0.87
	BiGRU	95.26	0.85	0.01	2.61	1.93	1.10
	Transformer	33.11	8.46	0.12	17.73	17.21	6.24
	Ada transformer	91.21	1.12	0.00	5.84	2.47	0.93

Table 3. The results of evaluating our method and other models on four datasets. The following benchmasrk is for row segmentation.

Test dataset	Model	Accuracy %					
		Correct	Partial	Missed	Over-seg	Under-seg	False-pos
PubTabNet (Val Set)	BiGRU [10]	56.93	3.51	0.03	17.38	21.95	7.23
	Split [18]	84.07	2.56	0.02	7.01	4.95	4.68
	Ours	**95.23**	0.63	0.03	1.93	2.49	2.25
UNLV	BiGRU	26.02	3.77	0.23	28.18	22.13	5.60
	Split	48.47	3.26	0.23	26.19	17.06	5.07
	Ours	**54.80**	2.28	0.22	14.24	16.42	5.29
ICDAR13	BiGRU	25.56	2.34	0.05	23.71	30.53	5.86
	Split	55.68	5.68	0.00	23.78	23.42	10.36
	Ours	**57.78**	3.60	0.00	27.55	27.98	7.46
ICDAR19	BiGRU	29.59	4.43	0.62	32.55	32.07	5.86
	Split	8.74	4.76	2.83	54.69	48.29	2.76
	Ours	**70.43**	3.85	0.52	14.06	9.31	4.32

The sequence labeling model is used in our method to get the row and column separators segmentation. We compare the performance of different sequence labeling models on PubTabNet, including BiLSTM, BiGRU, Transformer Encoder, Adaptive Transformer Encoder. The results of these sequence labeling models are on Table 2. Comparing with other SLMs, the BiLSTM model achieves the best results in correct row and column detections. On the other hand, the value of other measures is the minimum among these SLMs.

To verify the effectiveness of this method, the PubtabNet training set is used to train our model and two baseline models (BiGRU [10], Split [18]). And all table images have been resized to 600 × 600. The comparison between different models on four table recognition datasets are shown in Table 3 and Table 4.

Table 4. The results of evaluating our method and other models on four datasets. The following benchmasrk is for column segmentation.

Test dataset	Model	Accuracy %					
		Correct	Partial	Missed	Over-seg	Under-seg	False-pos
PubTabNet (Val Set)	BiGRU [10]	60.98	6.98	0.28	25.08	12.08	1.56
	Split [18]	88.35	1.36	0.00	8.61	2.91	0.96
	Ours	**97.39**	0.37	0.01	1.32	1.23	0.87
UNLV	BiGRU	34.24	18.96	4.11	33.80	25.99	1.66
	Split	60.53	9.65	0.05	20.86	16.77	2.99
	Ours	**71.85**	5.69	0.00	13.39	12.28	3.62
ICDAR13	BiGRU	32.46	7.30	1.09	36.16	26.16	5.78
	Split	71.65	10.24	0.00	16.67	17.34	0.40
	Ours	**90.63**	2.56	0.00	4.93	4.60	0.81
ICDAR19	BiGRU	38.21	14.33	2.95	37.19	26.09	0.00
	Split	12.22	7.81	1.44	48.41	42.94	6.16
	Ours	**81.09**	3.10	0.00	10.74	7.03	2.57

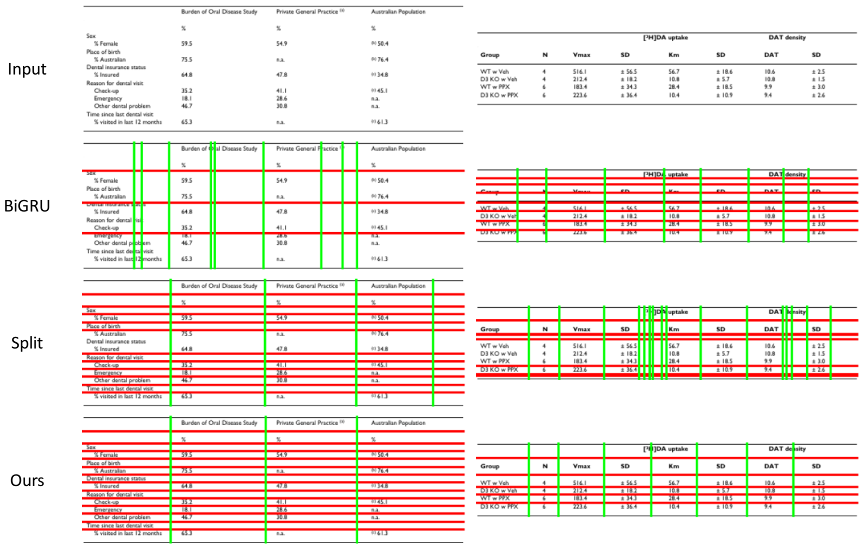


Fig. 3. Two samples of the results. The green lines represent the column separators and the red lines represent the row separators. (Color figure online)

Through the results, we can find the following things. When using BiGRU, it is obvious that there is a large quantity of Over Segmentation Detections and Under Segmentation Detections contributing to the lowest correct detections. And when evaluating the performance on UNLV, ICDAR13, and ICDAR19 datasets, the Correct Detections will be much lower. The Split has excellent performance on PubTabNet, UNLV, and ICDAR13 datasets. However, this model fails when testing on the ICDAR19 dataset. Our model achieves state-of-the-art performance than other models.

No matter on which data set, the Correct Detections of our model is much higher than BiGRU and Split. Two samples are represented on Fig. 3. It is obvious that the Over Segmentation Detection will occur when the blank space between two rows or two columns is large using BiGRU and Split. And when two rows or two columns are too close, BiGRU and Split will have higher Under Segmentation detections with lower Correct Detections. Our model solves these problems and achieves state-of-the-art performance in four datasets.

6 Conclusion

This paper proposes a novel encoder-decoder architecture for row and column separators segmentation. It uses a convolutional block as an encoder to extract feature. The decoder formulates the problem of labeling row and column separators as a sequence labeling problem and uses two parallel SLMs. In this paper, we make a comparison with the performances of different sequence labeling models. Besides, our model is evaluated on several available table recognition datasets. Our method achieves state-of-the-art performance on row and column segmentation and has a better generalization ability than other proposed methods.

In the future, a graph neural network will be applied to merge the basic cells to recognize the accurate table structure.

Acknowledgement. This work is supported by the projects of National Key R&D Program of China (2019YFB1406303) and National Natural Science Foundation of China (No. 61876003), 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. Chen, J., Lopresti, D.P.: Model-based tabular structure detection and recognition in noisy handwritten documents. In: 2012 International Conference on Frontiers in Handwriting Recognition, ICFHR 2012, pp. 75–80 (2012)
2. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, pp. 1724–1734 (2014)
3. Dengel, A., Kieninger, T.: A paper-to-HTML table converting system. In: Proceedings of Document Analysis Systems, pp. 356–365 (1998)
4. Gao, L., et al.: ICDAR 2019 competition on table detection and recognition (CTDAR). In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, pp. 1510–1515 (2019)
5. Göbel, M.C., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 table competition. In: 12th International Conference on Document Analysis and Recognition, ICDAR 2013, pp. 1449–1453 (2013)
6. Guo, Q., Qiu, X., Liu, P., Shao, Y., Xue, X., Zhang, Z.: Star-transformer. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, pp. 1315–1325 (2019)

7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
8. Hu, J., Kashi, R.S., Lopresti, D.P., Wilfong, G.T.: Table structure recognition and its evaluation. In: Document Recognition and Retrieval VIII, 2001. SPIE Proceedings, vol. 4307, pp. 44–55 (2001)
9. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991 (2015)
10. Khan, S.A., Khalid, S.M.D., Shahzad, M.A., Shafait, F.: Table structure extraction with bi-directional gated recurrent unit networks. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, pp. 1366–1371 (2019)
11. Kieninger, T., Dengel, A.: The T-Recs table recognition and analysis system. In: Lee, S., Nakano, Y. (eds.) Document Analysis Systems: Theory and Practice, Third IAPR Workshop, DAS 1998. vol. 1655, pp. 255–269 (1998)
12. Kieninger, T., Dengel, A.: Table recognition and labeling using intrinsic layout features. In: International Conference on Advances in Pattern Recognition, pp. 307–316 (1999)
13. Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: CascadeTab-Net: an approach for end to end table detection and structure recognition from image-based documents. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, pp. 2439–2447 (2020)
14. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table recognition using graph neural networks. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, pp. 142–147 (2019)
15. Raja, S., Mondal, A., Jawahar, C.V.: Table structure recognition using top-down and bottom-up cues. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12373, pp. 70–86. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_5
16. Shahab, A., Shafait, F., Kieninger, T., Dengel, A.: An open approach towards the benchmarking of table structure recognition systems. In: The Ninth IAPR International Workshop on Document Analysis Systems, DAS 2010. pp. 113–120 (2010)
17. Siddiqui, S.A., Fateh, I.A., Rizvi, S.T.R., Dengel, A., Ahmed, S.: DeepTabStR: deep learning based table structure recognition. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, pp. 1403–1409 (2019)
18. Tensmeyer, C., Morariu, V.I., Price, B.L., Cohen, S., Martinez, T.R.: Deep splitting and merging for table structure decomposition. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, pp. 114–121 (2019)
19. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS) 2017, pp. 5998–6008 (2017)
20. Wang, Y., Phillips, I.T., Haralick, R.M.: Table structure understanding and its performance evaluation. Pattern Recognit. **37**(7), 1479–1497 (2004)
21. Yan, H., Deng, B., Li, X., Qiu, X.: TENER: adapting transformer encoder for named entity recognition. CoRR abs/1911.04474 (2019)
22. Yan, Z., Ma, T., Gao, L., Tang, Z., Chen, C.: Persistence homology for link prediction: an interactive view. arXiv preprint arXiv:2102.10255 (2021)
23. Yuan, K., He, D., Jiang, Z., Gao, L., Tang, Z., Giles, C.L.: Automatic generation of headlines for online math questions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 9490–9497 (2020)

24. Yuan, K., He, D., Yang, X., Tang, Z., Kifer, D., Giles, C.L.: Follow the curve: arbitrarily oriented scene text detection using key points spotting and curve prediction. In: 2020 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2020)
25. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: data, model, and evaluation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 564–580. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_34