ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240549

RTSR: A Real-Time Table Structure Recognition Approach

Nam Quan Nguyen[a,*], **Xuan Phong Pham**[a] **and Tuan-Anh Tran**[a,b,**]

[a]Viettel Artificial Intelligence and Data Services Center, Viettel Group,
Lot D26 Cau Giay New Urban Area, Yen Hoa Ward, Cau Giay District, Hanoi, Vietnam.
[b]Faculty of Computer Science & Engineering, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM,
Ho Chi Minh City, Vietnam

Abstract. Table Structure Recognition (TSR) aims to reconstruct the logical structure of a table to understand semantic information ordered in the table. Many approaches to modeling the TSR problem have been proposed and have achieved promising results. However, most heavy models or complex post-processing approaches require much time and data consumption for inference and training progress. This paper proposes a new TSR approach called RTSR, a robust simplifying modeling. RTSR includes a lightweight backbone and a module to enhance contextual information between rows/columns. We combine two stages in a split-and-merge manner into only one step by reconstructing a table with horizontal and vertical separators. Specifically, we redesign the split stage to identify grid and spanning cells. Our RTSR can run on average at 38.1 FPS while achieving comparable performance with state-of-the-art methods on several benchmark datasets, including SciTSR, PubTabNet, FinTabNet, and WTW.

1 Introduction

Document processing automation gradually replaces the digital transformation time-consuming and error-prone manual data entry process. As one of the common elements in that process, tables organize and condense information in structural form. Table Structure Recognition (TSR) refers to reconstructing the logical structure of a table in images to machine-understandable formats, usually in logical coordinates or markup sequences. However, various layouts, arbitrary sizes, and implicit components make it challenging to get the correct structure for table reconstruction.

Due to increasing demand, TSR has gradually become a big problem and has recently received much attention. A diverse range of approaches exist to address the TSR problem, aimed at handling various tables that may originate from scanned images, photographs, or PDF documents. With deep heuristic analysis rules and computer vision techniques, [31] can extract table structure in some small datasets. This approach does not leverage the power of GPU; recent researches focus on deep learning models to accelerate processing time. [8] proposes a system entirely based on GPU processing to extract table structure in HTML tags from input images directly. Nevertheless, this approach is slow and does not achieve high Accuracy, requiring substantial data for model training. [35] employs an alternative technique using a graph neural network that utilizes textual information and layout to reconstruct tables, with text detection or recognition performed beforehand. In recent years, detecting table components and extracting table structures have become increasingly appealing approaches to numerous researchers. This method allows the TSR

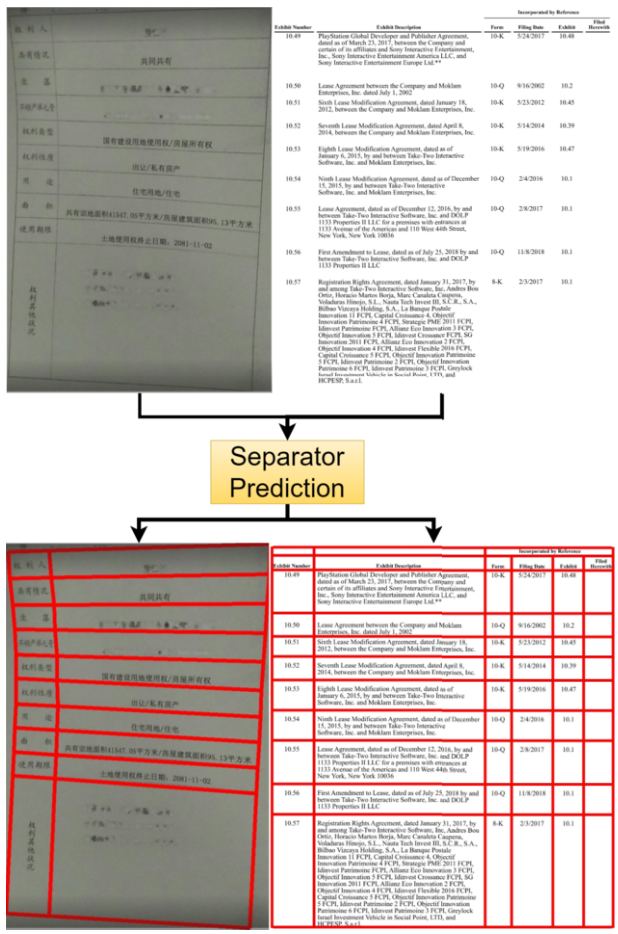


Figure 1. TSR with separator extraction approach in challenging condition. Warped table in photograph (left image). Borderless, spanning cell, empty cell, and multi-line content cell (right image).

* First author: ngnamquan@gmail.com
** Corresponding author: trtanh@hcmut.edu.vn

module to operate concurrently with text detection, reducing system latency. [30, 22, 10, 12] propose the detection of cell/column/row regions or the identification of column/row separators, which yield superior results in table structure extraction while maintaining processing efficiency. Research on accuracy has yet to progress recently, while processing time is a very challenging issue, especially in practical applications (with many users and requests). Applying table recognition in document analysis packages in a product package requires fast processing time, especially with tables with many cells.

Our research focuses on optimizing runtime while ensuring approximation accuracy. Trade-offs between processing time and accuracy are always interesting, and often, we find ourselves in a trade-off situation. With only column and row separators, Figure 1 formulates the table structure in multiple conditions such as photographs, warped images, border, borderless, empty, spanning, and multi-line content cells. In this article, we use a separation extraction approach to solve the TSR problem and propose using soft labels and faster feature aggregation in deep learning networks. It has helped the model maintain Accuracy while reducing processing time to a level approaching real-time. Our significant contributions can be summarized as follows:

- Proposed a new TSR method with approximately the same accuracy as other methods but superior processing time (in real-time).
- Realizing extensive ablation studies and analysis on the performances of our proposed method on many different challenge datasets.

2 Related Work

Early studies on TSR methods such as [6, 34, 32] strongly depended on handcrafted features and heuristics logic. These methods mainly use traditional techniques in computer vision to recognize table structures like space analysis, connected component extraction, text block arrangement, and vertical/horizontal alignment. Usually, rule-based methods require deep insight into the dataset to design heuristics logic and adjust parameters manually. These approaches have impressive performance and rational analysis in small amounts of datasets, but their generalization still needs to be improved and appropriate for diverse structure data. In recent years, many deep learning-based approaches have conspicuously outperformed traditional methods in accuracy and scalability. These approaches can be divided into three categories based on modeling the TSR problem: Table component extraction-based methods, Markup language-based methods, and Bottom-up methods.

2.1 Table component extraction based methods.

These methods focus on detecting the components that make up a table and then use post-process steps to reconstruct it. Straightforwardly, [26] directly detects cell location with object detection and then links them to get structure by graphs. The most challenging part of cell detection is empty cells, where the network will be confused when learning to empty due to explicit shape. Some methods [28, 29, 22] consider recognition of rows and columns, then intersecting them to extract table cells. [18, 16] reconstruct table structure based on vertex, edge of each cell, and logical relationship of each cell. Recently, recognizing table structure by row/column separators (Figure 1) become more efficient than row/column regions because they alleviate sensitivity to the alignment of cells. These approaches are based on the "split-and-merge" paradigm [30], which identifies the primary grid of cells and then recovers spanning cells to generate table structure. Each stage, in this manner, attracted many researchers to continue proposing ways for improvement. For the split stage, [7, 10, 37, 12, 33] focus on row/column separators extraction while [36, 20, 4] propose some ideas to improve merge grid cells into spanning cells. Besides, [19, 14] is concerned about an efficient pipeline and suggests a proposal for both stages. Row/column separator recognition is highly robust for the TSR problem, so the inference time must be considered for real-world applications.

2.2 Markup language-based methods.

Methods in this type treat TSR as an image-to-text generation problem to convert raw table images into a text sequence describing table structure and cell contents. The text sequence can be formatted in HTML tags [8, 40] or LaTeX symbols [2], and both sequences are interchangeable. An obvious limitation of these methods is that they are time and memory-consuming, especially when dealing with a complex table with many cells. To alleviate this issue, [17] proposed a new markup language called OTSL to improve performance (accuracy and inference time) by optimizing table structure representation in HTML sequence. Despite significant improvements in processing time, this work and this method, in general, still cannot meet the speed for real-life applications.

2.3 Bottom-up methods.

These approaches treat table structure as graph representation while text regions like words or cell contents as nodes. The graph neural network (GNN) is applied to predict the relationship of each sampled node pair in the same cell, row, or column. Methods [1, 24, 13] still need additional information about the text segment location and content, which can be extracted from PDF metadata or need an OCR engine. Later methods [35, 9] design text detection module inside the system and combine with GCN to create a comprehensive pipeline.

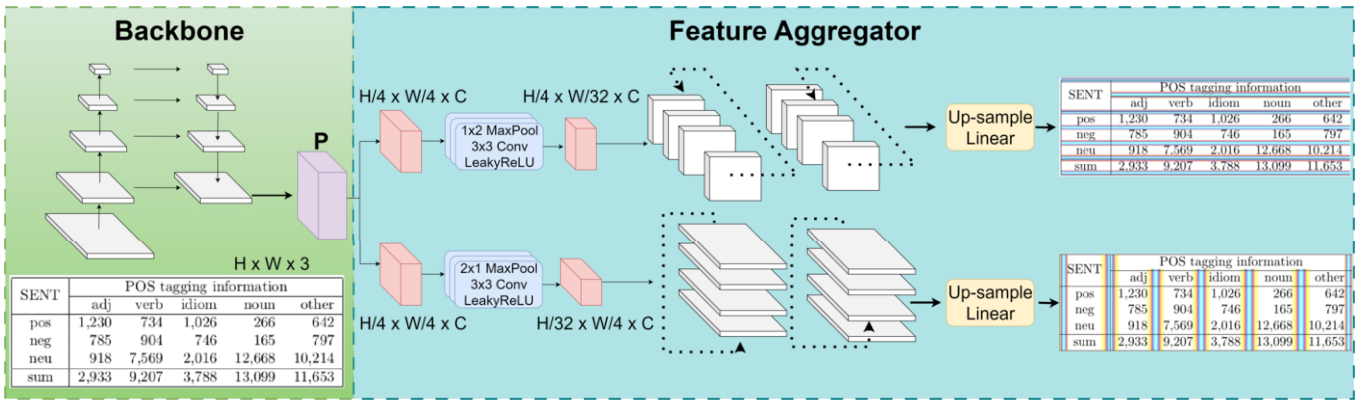
3 Method

3.1 Backbone

As Figure 2, RTSR comprises two key components: CNN backbone and Separator prediction. Firstly, the input table image $X \in R^{H \times W \times 3}$ is fed into a feature-pyramid backbone to produce feature $P \in R^{\frac{H}{4} \times \frac{W}{4} \times C}$, where C represents the number of channels and is set to 64 in our experiments. The network backbone uses Resnet18 [5] as the visual feature encoder with Feature Pyramid Network (FPN) [11] produced by the pixel decoder with resolution $1/32, 1/16, 1/8, 1/4$ of the original input image to extract multi-resolution features. Then, feature P is used to predict the probability map of row and column separators simultaneously.

3.2 Feature Aggregator

Two semantic segmentation branches are built on feature map P to enhance features and predict row separation mask S^{row} and column separation mask S^{col}. Taking the row mask S^{row} as an example, we first down-sample P eight times in a horizontal direction by conducting consecutively three times sequences of a 1×2 max-pooling layer, a 3×3 convolutional layer and a LeakyReLU activation layer. After that, feature map $F^{row} \in R^{\frac{H}{4} \times \frac{W}{32} \times C}$ is obtained as input of



SENT	POS tagging information				
	adj	verb	idiom	noun	other
pos	1,230	734	1,026	266	642
neg	785	904	746	165	797
neu	918	7,569	2,016	12,668	10,214
sum	2,933	9,207	3,788	13,099	11,653

Figure 2. The proposed architecture of RTSR.

two following feature aggregator modules RESA [38] to gather spatial information horizontally. F^{row} can be splitted into $W/32$ slides, which are denoted as $F^{row} = \{f_i \in R^{\frac{H}{4} \times 1 \times C}, i = 1, 2, ..., \frac{W}{32}\}$. In this module, the feature map is shifted simply by index calculation as follows:

$$f_i^{left} = f_{(i+\frac{L}{2^{K-k}})} \mod L \forall i = 1, 2, ..., L \quad (1)$$

$$f_i^{right} = f_{(i-\frac{L}{2^{K-k}})} \mod L \forall i = 1, 2, ..., L \quad (2)$$

where L is down-sample dimension, here $L = \frac{W}{32}$, f_i^{left} is new position of slice $i - th$ after left-shifting, K is iteration total and k is current iteration. Eq. (1) and Eq. (2) show left and right-shifted feature map formulas. The first aggregator module applies the convolution layer with kernel 9×1 followed by the ReLU activation layer to the left-shifted feature map, and its output is updated with the input feature map. This procedure is done iteratively after K times. The second aggregator module propagates information in a reversed direction with a right-shifted feature map. After finishing the passing information process, the result feature map is up-sampled by a factor of $(4, 32)$ and predicted by a linear transformation to generate an output probability map $S^{row} \in R^{H \times W \times 1}$. The loss function for predicted map S^{row} with ground-truth $\bar{S}^{row}$ is formulated as follows:

$$L^{row} = \sum_{i,j}^{H,W} (\alpha \times L_{bce}(S_{i,j}^{row}, \bar{S}_{i,j}^{row}) + (1-\alpha) \times L_{dice}(S_{i,j}^{row}, \bar{S}_{i,j}^{row})) \quad (3)$$

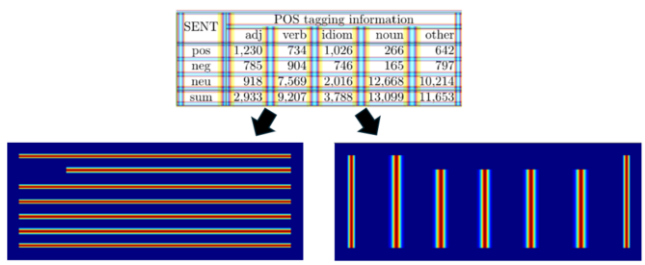
where L_{bce}, L_{dice} is the binary-cross entropy and dice loss, α is a hyper-parameter for balancing loss, we set $\alpha = 0.4$.

3.3 Label generation

For each training image, we generate the ground truth label for the row/column separators of the table. Each separator utilized is distinct from those employed in split-and-merge methodologies, guaranteeing that it avoids crossing the spanning cell.

Unlike a binary segmentation map, which labels each pixel as 0 or 1, we encode the probability of the center of the separator with a Gaussian heat map. Motivated by the advanced hybrid pyramid mask alignment in cell detection presented in [25], we find that using the soft-label segmentation obtains more accurate aligned separation regions. The utilization of soft labels assists the model in concentrating on pixels along the middle line of the separator. Consider a row separator with width D and length L as an example; the row mask is generated by calculating the 1D Gaussian distribution array with mean $\mu = \frac{D}{2}$ and covariance $\sigma^2 = 3 \times D$. Then, assign this array to the L position in the row separator region, shown in Equation (4). The final output is a row separator mask with soft label value in range $[0, 1]$, as shown in Figure 3

$$W_i(x) = \frac{1}{\sqrt{6\pi D}} e^{-\frac{(x-\frac{D}{2})^2}{6D}}, \forall i = 1, ..., L \quad (4)$$



SENT	POS tagging information				
	adj	verb	idiom	noun	other
pos	1,230	734	1,026	266	642
neg	785	904	746	165	797
neu	918	7,569	2,016	12,668	10,214
sum	2,933	9,207	3,788	13,099	11,653

Figure 3. An example of target masks using Soft label (Gaussian heat map).

4 Experiments

4.1 Dataset and Metric

This paper used four well-known datasets for the experiment and evaluation analysis.

SciTSR [30] is a large-scale table structure recognition dataset derived from scientific papers. It contains 15000 tables split into $12000/3000$ for training and testing. Additionally, the authors established a sub-dataset named SciTSR-COMP, consisting of $2,885$ and 716 extremely complex tables in the training and test sets, respectively, to enhance the challenges. As presented in [30], the metric for this dataset is the cell adjacent relationship score.

Pubtabnet [40] is a large table structure recognition dataset that extracts research papers from the medical domain. This dataset contains $500,777/9,115/9,138$ documents for training/validating and testing. Because the labeling of the testing set has yet to be released, the validation set is used for the evaluation profile. This article also proposes a measure called TED to evaluate performance. However, the OCR metric is not fair when considering table restructuring. Therefore, several modified versions, such as TEDS-Struct, have been proposed and are widely used in TSR competitions. We also use this modified metric to evaluate our approach on this dataset.

Fintabnet [39] is a large dataset containing more than $70,000$ pages with full labeling, including bounding box and cell structure from the annual reports of the S&P companies. The number of axis-aligned tables with cell bounding boxes obtained in these images is $91596/10635/10656$ as train/val/test. As proposed in FinTabNet paper [39], the TEDS-Struct is used as the evaluation metric.

WTW [15] is a different dataset, as it collects images mainly from natural scene images and focuses on bordered tabular objects. This dataset contains $10,970$ images for training and $3,611$ for testing. This dataset is well-labeled with table ID, coordinates, and row and column information. Following [27], we crop table regions from original images for training and testing, using the cell adjacency relationship ($IoU = 0.6$) [3] as the evaluation metric.

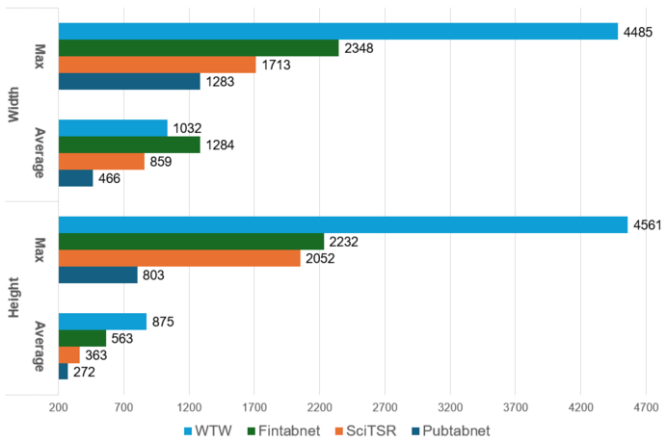


Figure 4. Image size distribution in 4 datasets: SciTSR, Pubtabnet, Fintabnet, and WTW.

4.2 Implementation details

All experiments are conducted using PyTorch 1.13.0. The training phase is on an NVIDIA V100 32GB GPU, while inference was executed on an RTX 3060. The weights of the ResNet-18 are initialized from the pre-trained model of ImageNet. We employ the AdamW algorithm for optimization, with the following hyper-parameters: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e^{-8}$, $\lambda = 5e^{-4}$. During the training phase, we resize the longer side of the table image to pre-defined values to facilitate multi-scale training while preserving the aspect ratio. We conducted statistical analysis, as depicted in Figure 4, on the size distribution of images across all four datasets to determine the optimal longer side dimension. Fintabnet and WTW datasets exhibit larger average image sizes, leading us to select higher dimensions, namely $\{1056, 1152, 1184\}$, while $\{800, 896, 928\}$ were chosen for Pubtabnet and SciTSR datasets.

In the testing phase, we resize the longer side of each image to 896 for SciTSR and Pubtabnet and 1184 for Fintabnet and WTW. The separator maps are binarized dynamically with the Otsu algorithm [21].

4.3 Experiment Results

We compare our RTSR with other state-of-the-art TSR methods on four popular datasets: SciTSR, Pubtabnet, Fintabnet, and WTW. Our approach has comparable performance for the first benchmark in Table 1 about the SciTSR dataset, and the gap with the highest method is 1.0% (in SciTSR-COMP). With a simple design in our approach, RTSR can perform at a real-time speed with an FPS average of 45.8, a significant gap from other methods. Here are some studies on processing speed, and more information about FPS needs to be provided in the report.

From an accuracy perspective with large-scale datasets, our method performs less than 1.4% and 1.8% Fintabnet and Pubtabnet (Table 2). The reason for these datasets could be that the size is large. At the same time, RTSR is lightweight, model capacity is inadequate with enormous data points, and we have not done any specific post-processing on the datasets, as shown in Figure 5, 6. However, the processing time on each image of these two data still ensures real-time performance. On the more challenging WTW dataset, our RTSR achieves acceptable performance with previous state-of-the-art, behind 2.2% compared with LORE++[16] (Table 3).

From a computational performance perspective, Table 4 presents the processing speed of RTSR across four datasets with an average of 38.1 FPS. The primary discrepancy of FPS among the datasets lies in their image resolutions and the quantity of separators within each dataset. Due to the utilization of larger image sizes in WTW and Fintabnet, their processing speeds are comparatively lower. Specifically, WTW exhibits the slowest speed, attributed to its handling of photographic images and the longer execution time required for post-processing steps.

4.4 Result analysis

Some example results depicted in Figure 5 showcase the efficacy of the proposed method across various complex layouts. Our model demonstrates proficiency in extracting table structures from bordered and borderless tables. In bordered tables, explicit borders serve as cues for separators, while in borderless, whitespace and alignment between cells play a pivotal role in recognition. Addressing challenges inherent to the TSR problem, our approach successfully handles issues such as spanning cell (Figure 5-(a-e)), warped image (Figure 5-f), form-like document with non-axis structures and empty cell (Figure 5-g).

Nonetheless, our method exhibits limitations in specific scenarios, such as the absence of spanning cells in the density table (Figure 6-a). Our baseline, which employs FPN-resnet18, is notably lightweight, resulting in numerous false positive separators, as depicted in Figure 6. Noisy separators can be mitigated through additional post-processing steps or by implementing a more robust design in the backbone. Furthermore, our approach still experiences diminished Accuracy when dealing with photographic images, particularly in cases where the background complexity increases (Figure 6-g).

4.5 Ablation Study

To clarify some performance and evaluation, we conducted multiple experiments to evaluate the effectiveness of our proposal modules.

Table 1. Performance evaluation of the nine methods on the SciTSR and SciTSR-COMP datasets (FPS means frame per second). The processing time information is taken from the publication and may differ from the experimental setup.

Method/Dataset	SciTSR			SciTSR-COMP			FPS
	Prec.	Rec.	F1.	Prec.	Rec.	F1.	
TabStruct-Net [26]	92.7	91.3	92.0	90.9	88.2	89.5	-
GraphTSR [30]	95.9	94.8	95.3	52.9	41.7	46.6	-
LGPMA [25]	98.2	99.3	98.8	97.3	98.7	98.0	-
FLAG-Net [13]	99.7	99.3	99.5	98.4	98.6	98.5	-
SEMv2 [37]	99.3	99.2	99.3	98.7	98.6	98.7	7.3
TSRFormer w/ DQ-DETR [33]	99.5	99.3	99.4	99.1	98.6	98.9	4.2
GridFormer [18]	99.4	99.0	99.2	-	-	-	-
LORE++ [16]	-	-	-	99.4	99.2	99.3	2.3
Our RTSR	99.5	99.4	99.4	98.7	98.0	98.3	45.8

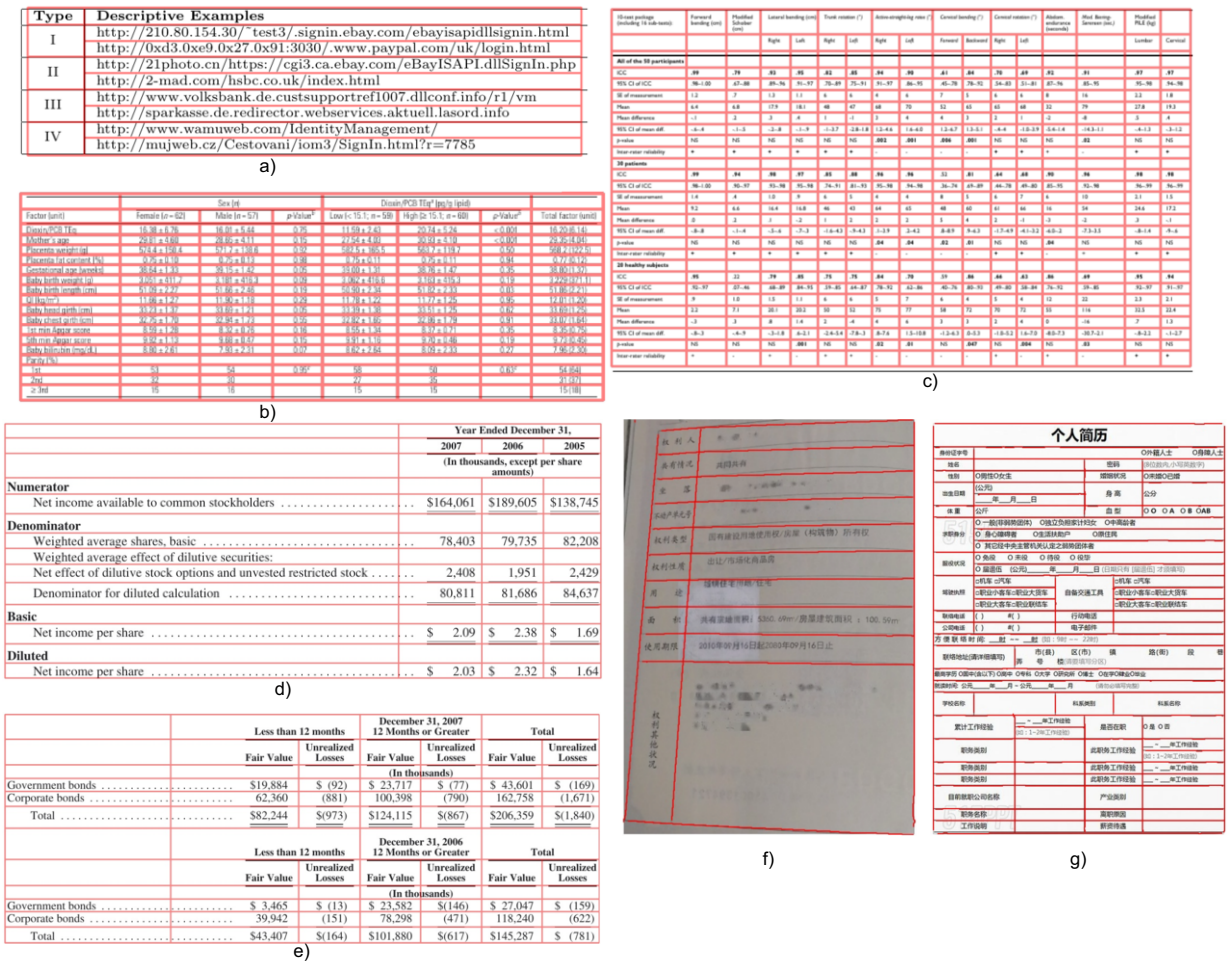


Figure 5. Example results of the RTSR on datasets, (a) SciTSR, (b-c) Pubtabnet, (d-e) Fintabnet, (f-g) WTW.

Table 2. Evaluation profile for the six methods on the Pubtabnet and Fintabnet dataset (TEDS-struct metric).

Method	Pubtabnet	Fintabnet
LGPMA [25]	96.7	-
SEMv2 [37]	97.5	92.8
RobusTabNet [19]	96.5	93.8
TSRFormer w/ DQ-DETR [33]	97.5	98.4
GridFormer [18]	97.0	98.6
Our RTSR	95.7	97.2

Table 3. Evaluation profile for the six methods on the WTW dataset.

Method	Prec.	Rec.	F1.
Cycle-CenterNet [15]	93.3	91.5	92.4
SEMv2 [37]	93.8	93.4	93.6
TSRFormer w/ DQ-DETR [33]	94.5	94.0	94.3
Grid-Former [18]	94.1	94.2	94.1
LORE++ [16]	94.5	95.9	95.1
Our RTSR	92.2	93.6	92.9

The ablation study is assessed using the SciTSR dataset.

Message Passing with Feature Shift Aggregation. The result on Table 5 indicate the efficiency of RESA module [38]. SCNN [23] needs to iterate over each point on every image dimension, con-

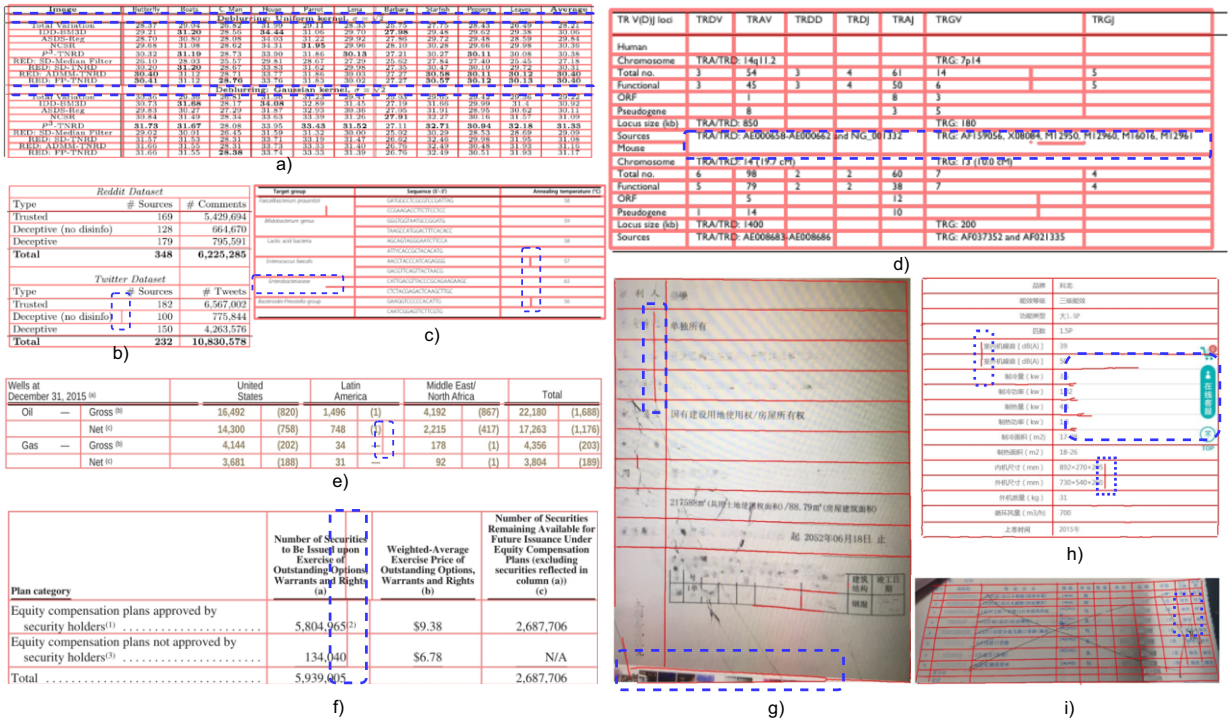


Figure 6. Example of failure cases (**blue dash box**) of the RTSR on datasets, (a-b) SciTSR, (c-d) Pubtabnet, (e-f) Fintabnet, (g-i) WTW.

Table 4. Speed performance of the RTSR on four datasets.

Dataset	SciTSR	Pubtabnet	Fintabnet	WTW	**Average**
FPS	45.8	43.2	38.0	25.5	**38.1**

Table 5. Ablation studies of Soft label and Message passing method in RTSR.

Soft	**Mes/Pass**	**SciTSR**			**FPS**	**#Param**
		Prec.	**Rec.**	**F1.**		
No	SCNN	99.29	99.00	99.10	29.4	18.4M
Yes	SCNN	99.38	99.16	99.24		
No	RESA	99.45	99.14	99.26	45.8	19.0M
Yes	RESA	99.53	99.37	99.43		

suming processing time and computation. RESA helps to reduce the computation complexity from $O(n)$ down to $O(1)$ with a predefined number of iterations. Specifically, FPS on RESA is better than SCNN by about $\times 1.5$ despite using more parameters. Besides, with the RESA module, prediction accuracy has slightly improved.

Table 6. Performance of the RTSR with different iterations in RESA.

Num iters	**Prec.**	**Rec.**	**F1.**
3	99.50	99.32	99.38
5	99.53	99.37	99.43
7	99.48	99.35	99.40

Soft label. As reported in Table 5, improving the soft label in the pipeline is not trivial. In combination with SCNN or RESA, soft label also increases the F1 score over 0.17% without more computation.

Number of iterations in RESA. In this section, we explore the effect of different iterations in RESA. Theoretically, as the iteration increases, each slice of the feature map can aggregate more information, which contributes to obtaining better performance. As shown in Table 6, the performance will improve as the iteration increases. However, more iterations lead to more computational time costs, while performance improvement is insignificant. We designate iteration 5 as our ultimate selection to make a balance between them.

5 Conclusion

This paper presents a new approach to the real-time table structure recognition problem called RTSR. We adopt a soft label and feature aggregation to increase the ability to extract and retain feature characteristics in deep learning networks. Our segmentation-based approach can reach a real-time prediction while keeping an approximation of advanced Accuracy. Experimental results demonstrate that our method has outperformed current TSR methods on time and competes in Accuracy. The following research will focus on model design to achieve better performance on large-scale datasets while ensuring real-time speed.

Acknowledgments

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

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

[2] Y. Deng, D. Rosenberg, and G. Mann. Challenges in end-to-end neural scientific table recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 894–901. IEEE, 2019.

[3] M. Göbel, T. Hassan, E. Oro, and G. Orsi. A methodology for evaluating algorithms for table understanding in pdf documents. In *Proceedings of the 2012 ACM symposium on Document engineering*, pages 45–48, 2012.

[4] Z. Guo, Y. Yu, P. Lv, C. Zhang, H. Li, Z. Wang, K. Yao, J. Liu, and J. Wang. Trust: an accurate and end-to-end table structure recognizer using splitting-based transformers. *arXiv:2208.14687*, 2022.

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

[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 (ICDAR'93)*, pages 765–768. IEEE, 1993.

[7] 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 (ICDAR)*, pages 1366–1371. IEEE, 2019.

[8] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1918–1925, 2020.

[9] X.-H. Li, F. Yin, H.-S. Dai, and C.-L. Liu. Table structure recognition and form parsing by end-to-end object detection and relation parsing. *Pattern Recognition*, 132:108946, 2022.

[10] Y. Li, Y. Huang, Z. Zhu, L. Pan, Y. Huang, L. Du, Z. Tang, and L. Gao. Rethinking table structure recognition using sequence labeling methods. In *2021 International Conference on Document Analysis and Recognition (ICDAR)*, pages 541–553. Springer, 2021.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[12] W. Lin, Z. Sun, C. Ma, M. Li, J. Wang, L. Sun, and Q. Huo. Tsrformer: Table structure recognition with transformers. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6473–6482, 2022.

[13] H. Liu, X. Li, B. Liu, D. Jiang, Y. Liu, B. Ren, and R. Ji. Show, read and reason: Table structure recognition with flexible context aggregator. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1084–1092, 2021.

[14] H. Liu, X. Li, M. Gong, B. Liu, Y. Wu, D. Jiang, Y. Liu, and X. Sun. Grab what you need: Rethinking complex table structure recognition with flexible components deliberation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 3603–3611, 2024.

[15] R. Long, W. Wang, N. Xue, F. Gao, Z. Yang, Y. Wang, and G.-S. Xia. Parsing table structures in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 944–952, 2021.

[16] R. Long, H. Xing, Z. Yang, Q. Zheng, Z. Yu, C. Yao, and F. Huang. Lore++: Logical location regression network for table structure recognition with pre-training. *arXiv preprint arXiv:2401.01522*, 2024.

[17] M. Lysak, A. Nassar, N. Livathinos, C. Auer, and P. Staar. Optimized table tokenization for table structure recognition. In *International Conference on Document Analysis and Recognition*, pages 37–50. Springer, 2023.

[18] P. Lyu, W. Ma, H. Wang, Y. Yu, C. Zhang, K. Yao, Y. Xue, and J. Wang. Gridformer: Towards accurate table structure recognition via grid prediction. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7747–7757, 2023.

[19] C. Ma, W. Lin, L. Sun, and Q. Huo. Robust table detection and structure recognition from heterogeneous document images. *Pattern Recognition*, 133:109006, 2023.

[20] N. Q. Nguyen, A. D. Le, A. K. Lu, X. T. Mai, and T. A. Tran. Formerge: Recover spanning cells in complex table structure using transformer network. In *International Conference on Document Analysis and Recognition*, pages 522–534. Springer, 2023.

[21] N. Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[22] S. S. Paliwal, D. Vishwanath, 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 (ICDAR)*, pages 128–133. IEEE, 2019.

[23] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[24] S. R. Qasim, H. Mahmood, and F. Shafait. Rethinking table recognition using graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 142–147. IEEE, 2019.

[25] L. Qiao, Z. Li, Z. Cheng, P. Zhang, S. Pu, Y. Niu, W. Ren, W. Tan, and F. Wu. Lgpma: complicated table structure recognition with local and global pyramid mask alignment. In *International conference on document analysis and recognition*, pages 99–114. Springer, 2021.

[26] S. Raja, A. Mondal, and C. Jawahar. Table structure recognition using top-down and bottom-up cues. In *2020 European Conference on Computer Vision (ECCV)*, pages 70–86. Springer, 2020.

[27] R. Rastan, H.-Y. Paik, and J. Shepherd. Texus: A unified framework for extracting and understanding tables in pdf documents. *Information Processing & Management*, 56(3):895–918, 2019.

[28] 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 (ICDAR)*, volume 1, pages 1162–1167. IEEE, 2017.

[29] B. Smock, R. Pesala, and R. Abraham. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4634–4642, 2022.

[30] 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 (ICDAR)*, pages 114–121. IEEE, 2019.

[31] T. A. Tran, H. T. Tran, I. S. Na, G. S. Lee, H. J. Yang, and S. H. Kim. A mixture model using random rotation bounding box to detect table region in document image. *Journal of Visual Communication and Image Representation*, 39:196–208, 2016.

[32] T. A. Tran, O. KangHan, I. S. Na, G. S. Lee, H. J. Yang, and S. H. Kim. A robust system for document layout analysis using multilevel homogeneity structure. *Expert Systems With Applications*, 85(285-296): 99–113, 2017.

[33] J. Wang, W. Lin, C. Ma, M. Li, Z. Sun, L. Sun, and Q. Huo. Robust table structure recognition with dynamic queries enhanced detection transformer. *Pattern Recognition*, 144:109817, 2023.

[34] Y. Wang, I. T. Phillips, and R. M. Haralick. Table structure understanding and its performance evaluation. *Pattern recognition*, 37(7):1479–1497, 2004.

[35] W. Xue, Q. Li, and D. Tao. Res2tim: Reconstruct syntactic structures from table images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 749–755. IEEE, 2019.

[36] Z. Zhang, J. Zhang, J. Du, and F. Wang. Split, embed and merge: An accurate table structure recognizer. *Pattern Recognition*, 126:108565, 2022.

[37] Z. Zhang, P. Hu, J. Ma, J. Du, J. Zhang, B. Yin, B. Yin, and C. Liu. Semv2: Table separation line detection based on instance segmentation. *Pattern Recognition*, 149:110279, 2024.

[38] T. Zheng, H. Fang, Y. Zhang, W. Tang, Z. Yang, H. Liu, and D. Cai. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3547–3554, 2021.

[39] X. Zheng, D. Burdick, L. Popa, X. Zhong, and N. X. R. Wang. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 697–706, 2021.

[40] X. Zhong, E. ShafieiBavani, and A. Jimeno Yepes. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pages 564–580. Springer, 2020.