Improving Table Structure Recognition with Visual-Alignment Sequential Coordinate Modeling

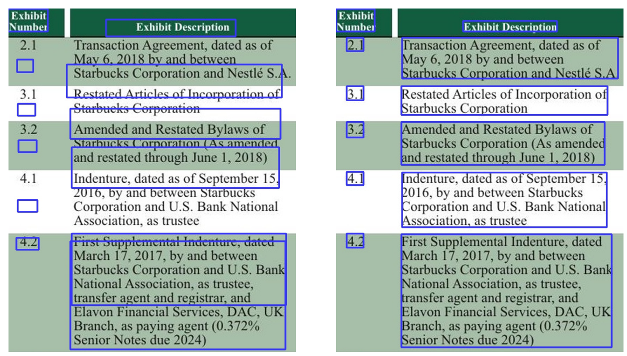
Yongshuai Huang[*,1] Ning Lu[*,1] Dapeng Chen[1]
Yibo Li[2] Zecheng Xie[1] Shenggao Zhu[1] Liangcai Gao[2] Wei Peng[1]
[1] Huawei Technologies Ltd. [2] Peking University
{huangyongshuai1,luning12,chendapeng8,xiezecheng1,zhushenggao,peng.wei1}@huawei.com
{yiboli,gaoliangcai}@pku.edu.cn

Abstract

Table structure recognition aims to extract the logical and physical structure of unstructured table images into a machine-readable format. The latest end-to-end image-to-text approaches simultaneously predict the two structures by two decoders, where the prediction of the physical structure (the bounding boxes of the cells) is based on the representation of the logical structure. However, the previous methods struggle with imprecise bounding boxes as the logical representation lacks local visual information. To address this issue, we propose an end-to-end sequential modeling framework for table structure recognition called ***VAST****. It contains a novel coordinate sequence decoder triggered by the representation of the non-empty cell from the logical structure decoder. In the coordinate sequence decoder, we model the bounding box coordinates as a language sequence, where the left, top, right and bottom coordinates are decoded sequentially to leverage the inter-coordinate dependency. Furthermore, we propose an auxiliary visual-alignment loss to enforce the logical representation of the non-empty cells to contain more local visual details, which helps produce better cell bounding boxes. Extensive experiments demonstrate that our proposed method can achieve state-of-the-art results in both logical and physical structure recognition. The ablation study also validates that the proposed coordinate sequence decoder and the visual-alignment loss are the keys to the success of our method.*

1. Introduction

Tables are an essential medium for expressing structural or semi-structural information. Table structure recognition, including recognizing a table's logical and physical structure, is crucial for understanding and further editing a visual table. The logical structure represents the row-column relation of cells and the spanning information of a cell. The physical structure contains not only the logical structure but also the bounding box or content of the cells, focusing on the exact locations in the image.



(a) TableFormer (Baseline) (b) VAST (Ours)

Figure 1. Visualization comparison of the bounding box predicted by TableFormer and VAST. Our results are more accurate, which is vital for downstream content extraction or table understanding tasks. The image is cropped from the table with id 7285, which comes from FinTabNet.

Table recognition can be implemented by an end-to-end encoder-decoder paradigm. Such methods excel at predicting the logical structure but usually produce less accurate physical structures, *i.e.*, bounding boxes of cells or cell contents. However, the bounding box accuracy is essential to downstream tasks, such as text information extraction or table QA. This work designs the sequential coordinate decoding and enforces more visual information to produce more accurate bounding boxes.

In the coordinate sequence decoder, the start embedding of the non-empty cell is the representation from the HTML sequence decoder. The representation usually contains a more global context of the table and has fewer local visual details. Because the local visual appearance is vital for pre-

*Equal contribution.

dicting accurate coordinates, we align the representation of non-empty cells from the HTML sequence decoder with the visual features from the CNN image encoder. In particular, a visual-alignment loss is designed to maximize the cosine similarity of the paired visual-HTML representation in the image. In summary, our contributions are threefold.

- We propose a coordinate sequence decoder to significantly improve the table's physical structure accuracy upon an end-to-end table recognition system.
- We introduce a visual-alignment loss between the HTML decoder and coordinate sequence decoder. It enforces the representation from the HTML decoding module contains more detailed visual information, which can produce better bounding boxes for the non-empty cells.
- We develop an end-to-end sequential modeling framework for table structure recognition, the comparison experiments prove that our method can achieve state-of-the-art performance and the ablation experiments show the effectiveness of our method.

2. Related Work

The recent deep learning approaches have shown excellent performance on table structure recognition tasks. These methods can be divided into three categories: methods based on splitting and merging, methods based on detection and classification, and image-to-text generation methods.

Methods based on splitting and merging. These methods consist of two stages. The first stage detects rows and columns, then splits the table into multiple basic text blocks through the intersection of rows and columns; the second stage merges text blocks to restore the structure.

Several works focus on splitting the rows and columns better. For example, DeepDeSRT [34] and TableNet [26] adjusted FCN from the semantic segmentation to segment rows and columns. DeepTabStR [36] applied deformable convolution to Faster R-CNN [33], FPN [17], and R-FCN [4], which has a wider receptive field to capture the table line this can split accurate table rows and columns. Khan *et al*. [12] and Li *et al*. [15] used a bi-directional gated recurrent unit network to identify the pixel-level row and column separators. Inspired by DETR, TSRFormer [18] formulated table separation line prediction as a line regression problem and they proposed a separator regression transformer to predict separation lines from table images directly.

Several merging methods have been developed to recognize tables containing cells that span rows or columns. The SPLURGE method [40] proposed the idea of table splitting and merging. They designed a merging model to merge cells span multiple columns or rows. To achieve a more accurate merged result, [45] fuse both visual and semantic features to produce grid-level features. RobusTabNet [24] proposed a spatial CNN-based separation line prediction module to split the table into a grid of cells, and a Grid CNN-based cell merging module was applied to recover the spanning cells. TRUST [9] introduced an end-to-end transformer-based query-based splitting module and vertex-based merging module. The splitting module is used to extract the features of row/column separators, and the row/column features are further fed into the vertex-based merging module to predict the linking relations between adjacent basic cells.

Methods based on detection and classification. The basic idea of this method is first to detect the cells and then classify the row and column relationships between the cells. A graph can be constructed based on the cell and connection to obtain the table structure.

For the irregular layout table, a good cell detection result could effectively improve the accuracy of table recognition, [21, 27, 30, 46] were committed to improving the accuracy of cell detection. Some other researchers aimed to classify the cell relationship to construct table structure [3], [29], [16], [43]. They utilized ground truth or OCR results to get text blocks. Then they regarded text blocks as vertexes to construct a graph and used the graph-based network to classify the relationship between cells.

The most recent approaches put cell detection and cell relation classification into one network. TableStructNet [31] and FLAG-NET [20] both utilized Mask R-CNN [11] network to obtain the region of cells and cell visual features. They both utilized the DGCNN architecture in [28] to model the interaction between geometrically neighboring detected cells. Hetero-TSR [19] proposed a novel Neural Collaborative Graph Machines (NCGM) that leverages modality interaction to boost the multimodal representation for complex scenarios. Lee *et al*. [13] formulated tables as planar graphs, and they first obtained cell vertex confidence maps and line fields. After that, they reconstruct the table structure by solving a constrained optimization problem.

Methods based on image-to-text generation. These methods treat the structure of the table (HTML or latex, etc.) as a sequence, and adopt the end-to-end image-to-text paradigm to recognize the table structure.

Deng *et al*. [6] used the classic IM2MAKEUP framework [5] to recognize the logical structure of the table, where a CNN was designed to extract visual features, and an LSTM with an attention mechanism was used to generate the latex code of the table. Zhong *et al*. [47] tried to generate the logical structure and the cell content with an encoder-dual-decoder (EDD) architecture. In the decoding stage, they used two attention-based recurrent neural networks, one was responsible for decoding the table structure code, and the other was responsible for decoding the

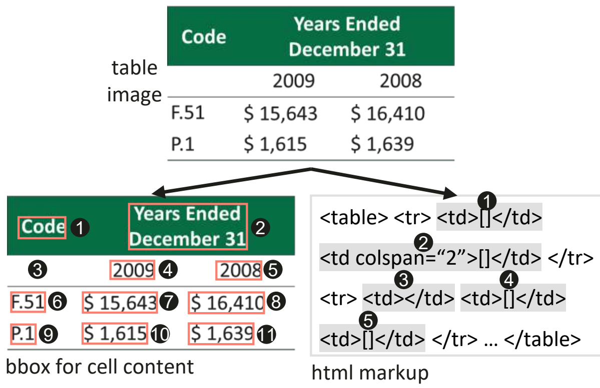


Figure 2. Visualization of table HTML markup and cells. Cell ❶ is a spanning cell that spans two columns, and cell ❸ is an empty cell with no content. '[]' refers to the content of the cell.

content. TableMaster [44] and TableFormer [25] leveraged the transformer decoder to improve the decoder of EDD. In addition, they used the regression decoder to predict the bounding box instead of the content. Since the lack of local visual information, the bounding boxes predicted by these methods were less accurate. In this paper, we treat the bounding box prediction as a coordinate sequence generation task, and cooperate with visual alignment loss to produce more accurate bounding boxes.

3. Task Definition

For a given table image, our goal is to predict its logical structure and physical structure end-to-end. Specifically, the logical structure refers to the HTML of the table, and the physical structure refers to the bounding box coordinates of all non-empty cells. We use $S = [s^1, \dots, s^T]$ to indicate the tokenized HTML sequence, where T is the length of sequences and s is a token of predefined HTML tags. We define $B = \{\mathbf{b}^1, \dots, \mathbf{b}^N\}$ is the set of sequences of all non-empty cells, where $\mathbf{b} = (x_{\text{left}}, y_{\text{top}}, x_{\text{right}}, y_{\text{bottom}})$, is a sequence of non-empty cell bounding box coordinates and each coordinate is discretized into an integer. An example of HTML for a table and content bounding boxes of non-empty cells is shown in Fig. 2.

4. Methodology

Our framework consists of three modules: a CNN image encoder, an HTML sequence decoder and a coordinate sequence decoder. Given a table image, we extract the feature map through the CNN image encoder. The feature map will be fed into the HTML sequence decoder and the coordinate sequence decoder to produce a HTML sequence and bounding boxes of the non-empty cells, respectively. The representation of non-empty cells from the HTML sequence decoder will trigger the coordinate sequence decoder. To enforce the local visual information of the representation, visual-alignment loss is employed during training. The model architecture is illustrated in Fig. 3.

4.1. CNN Image Encoder

We use a modified ResNet [23] equipped with multi-aspect global content attention as the CNN image encoder. The resulted image feature map is C4, which is from the output of the last convolutional layer of the 4-th stage. The input of the encoder is a RGB image with a size of $H \times W \times 3$. The output of the encoder is feature map $\mathbf{M}$ with a size $\frac{H}{16} \times \frac{W}{16} \times d$.

4.2. HTML Sequence Decoder

The logical structure of a table contains information such as the number of cells, rows, columns, adjacencies, spanning, etc. In this paper, we use HTML to represent the logical structure of a table. The ground truth HTML of table logical structure is tokenized into structural tokens. As in the work [44], we use merged label to represent a non-spanning cell to reduce the length of HTML sequence. Specifically, we use *<td></td>* and *<td>[]</td>* to denote empty cells and non-empty cells, respectively. For spanning cells, the HTML is tokenized to *<td*, *colspan="n"* or *rowspan="n"*, *>* and *</td>*. We use the first token *<td* to represent a spanning cell.

As shown in Fig. 3, the HTML sequence decoder is a transformer with a stack of $N = 3$ identical layers. The memory keys and values are the flattened feature map $\mathbf{M}$ added with the positioning encoding. The queries are shifted structure tokens. The output of the transformer is a HTML sequence, which is decoded by auto-regression. The output of the t-th step is a distribution: $p(s_t|\mathbf{M}, s_{1:t-1})$. In training, we employ the cross-entropy loss:

$$\mathcal{L}_s = -\log p(S^*|\mathbf{M}) = -\sum_{t=2}^{n} \log p(s_t^*|s_{1:t-1}^*, \mathbf{M}), \qquad (1)$$

where S^* is the ground truth HTML of the target table. The start token s_1^* or s_1 is a fixed token *<sos>* in both training and testing phrase.

4.3. Coordinate Sequence Decoder

For coordinate prediction, we cascade coordinate sequence decoder after HTML sequence decoder. The decoder is triggered by a non-empty cell s_i^{nc}. The left, top, right and bottom coordinates are decoded one element at a time. In particular, each of the continuous corner coordinates is uniformly discretized into an integer between $[0, n_{\text{bins}}]$. In the decoder, we utilize the embedding of the previously predicted coordinates to predict the latter coordinate, which inject contextual information into the prediction of the next coordinate. The procedure of the coordinate sequence decoder is also illustrated in Fig. 3.

Table 1. The public datasets for table structure recognition. "PDF" refers to multiple input modalities, such as images, text, etc., which can be extracted from PDF. "CAR" indicates cell adjacency relationship. "Det" indicates the evaluation of detection. "Cell BBox" and "Content BBox" refer to the bounding box of cells and content, respectively. "IC19B2H' and "IC19B2M" stand for "ICDAR2019 TrackB2 historical" and "ICDAR2019 TrackB2 Modern" respectively.

Dataset	#Samples			Input Modality	Cell Content	Cell BBox	Content BBox	Metric
	Train	Val	Test					
Logical Structure Recognition								
TABLE2LATEX-450K [7]	447K+	9,322	9,314	Image	✓	✗	✗	BLEU
TableBank [14]	130K+	10,000	5000	Image	✗	✗	✗	BLEU
PubTabNet [47]	500K+	9,115	10,000	Image	✓	✗	✓	TEDS
FinTabNet [46]	92K	10,635	10,656	PDF	✓	✗	✓	TEDS
Physical Structure Recognition								
UNLV [35]	-	-	558	Image	✗	✓	✗	Det
ICDAR2013 [10]	-	-	156	PDF	✓	✗	✓	CAR
IC19B2H [8]	-	-	190	Image	✗	✓	✗	CAR
IC19B2M [8]	-	-	145	Image	✗	✗	✓	CAR
SciTSR [2]	12K	-	3,000	PDF	✓	✗	✓	CAR
WTW [21]	10K+	-	3,611	Image	✗	✓	✗	CAR
TUCD [32]	-	-	4,500	Image	✗	✓	✗	CAR
PubTables-1M [38]	758K+	94,959	93,834	PDF	✓	✓	✓	GriTS

Table 2. Comparision on the FinTabNet and PubTabnet. "PTN + FTN" means training on PubTabNet and finetuning on FinTabNet.

Methods	Training Dataset	S-TEDS	TEDS
FinTabNet			
Det-Base [46]	PTN	41.57	-
GTE [46]	PTN + FTN	91.02	-
EDD [47]	PTN	90.60	-
TableFormer [25]	FTN	96.80	-
VAST	FTN	**98.63**	**98.21**
PubTabNet			
TabStructNet [31]	SciTSR		90.10
FLAG-Net [20]	SciTSR	-	95.10
NCGM [19]	SciTSR	-	95.40
GTE [46]	PTN	93.01	-
RobustTabNet [24]	PTN	97.00	-
LGPMA [30]	PTN	96.70	94.60
SEM [45]	PTN	-	93.70
EDD [47]	PTN	89.90	88.30
TableMaster [44]	PTN	96.04	96.16
TableFromer [25]	PTN	96.75	93.60
TSRFormer [18]	PTN	**97.50**	-
TRUST [9]	PTN	97.10	96.20
VAST	PTN	97.23	**96.31**

NCGM, FLAG-Net, etc., were tested on a randomly selected samples from the test set and did not release their split. Thus they are not directly comparable. For the fairness of the comparison, we only compare with methods that report their results on the ICDAR2013 full test dataset. As shown in Tab. 4, our VAST outperforms all previous methods with the best F1-score of 96.52% when trained with FinTabNet and 95.72% when trained with SciTSR.

Table 3. Comparison of content bounding box detection (Det) results on PubTabNet.

Methods	Dataset	AP_{50} (%)
EDD + BBox [25]	PTN	79.2
TableFormer [25]	PTN	82.1
VAST	PTN	**94.8**

On IC19B2M, we report the results with the IoU thresholds of 0.5 and 0.6 as the competitive baseline method GTE [46]. The **WAvg.F1** score is the weighted average value of F1 scores under each threshold. As shown in Tab. 5, VAST achieves the highest F1-score at the IoU threshold of 0.5 and 0.6, outperforming GTE by 12% and 13.2%, respectively. Compared with CascadeTabNet, when the IoU threshold is set to 0.6, VAST surpasses it by 7.9%, even though it used their own labeled ICDAR2019 dataset for training. Inherently, for the overall average F1 (WAvg.F1), VAST achieves the best score of 58.6%.

PubTables-1M is the most challenging benchmark dataset with 93834 samples for evaluation. As shown in Tab. 6, we report the results on $\mathrm{Acc}_{\mathrm{Cont}}$, $\mathrm{GriTS}_{\mathrm{Top}}$, $\mathrm{GriTS}_{\mathrm{Cont}}$ and $\mathrm{GriTS}_{\mathrm{Loc}}$. The scores of VAST in $\mathrm{Acc}_{\mathrm{Top}}$,

Table 4. Comparison of cell adjacency relation (CAR) score on the SciTSR and ICDAR2013 datasets.

Methods	Training Dataset	P (%)	R (%)	F1 (%)
SciTSR				
GraphTSR [2]	SciTSR	95.90	94.80	95.30
TabStructNet [31]	SciTSR	92.70	91.30	92.00
LGPMA [30]	SciTSR	98.20	99.30	98.80
SEM [45]	SciTSR	97.70	96.52	97.11
RobustTabNet [24]	SciTSR	99.40	99.10	99.30
FLAG-Net [20]	SciTSR	99.70	99.30	99.50
NCGM [19]	SciTSR	99.70	**99.60**	**99.60**
TSRFormer [18]	SciTSR	99.70	**99.60**	**99.60**
VAST	SciTSR	**99.77**	99.26	99.51
ICDAR2013				
GraphTSR [2]	SciTSR	88.50	86.00	87.20
TabStructNet [31]	SciTSR	91.50	89.70	90.60
CycleCenterNet [21]	WTW	**95.50**	88.30	91.70
LGPMA [30]	SciTSR	93.00	97.70	95.30
GTE [46]	FTN	92.72	94.41	93.50
VAST	SciTSR	93.84	**97.68**	**95.72**
VAST	FTN	95.29	**97.79**	**96.52**

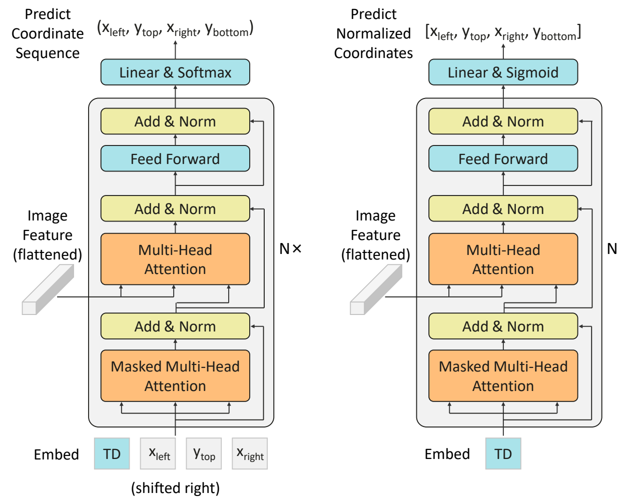
Table 5. Comparison of cell adjacency relation (CAR) F1-score (%) on the IC19BM. "IC19 †" refers to the manually annotated ICDAR2019 dataset in CascadeTabNet [27].

Methods	Training Dataset	IoU 0.5	IoU 0.6	WAvg.F1
NLPR-PAL [8]	-	-	36.5	36.5
CascadeTabNet [27]	IC19 †	-	43.8	43.8
GTE [46]	FTN	54.8	38.5	45.9
VAST	FTN	**66.8**	**51.7**	**58.6**

Table 6. Comparison of GriTS (%) score on PubTables-1M

Methods	$\mathrm{Acc}_{\mathrm{Cont}}$	$\mathrm{GriTS}_{\mathrm{Top}}$	$\mathrm{GriTS}_{\mathrm{Cont}}$	$\mathrm{GriTS}_{\mathrm{Loc}}$
FasterRCNN [25]	10.39	86.16	85.38	72.11
DETR [25]	81.38	98.45	98.46	**97.81**
VAST	**90.11**	**99.22**	**99.14**	94.99

$\mathrm{GriTS}_{\mathrm{Top}}$, $\mathrm{GriTS}_{\mathrm{Cont}}$ are 90.11%, 99.22% and 99.14% respectively, achieving the current state-of-the-art performance. The $\mathrm{GriTS}_{\mathrm{Loc}}$ score of VAST is lower than that of DETR because DETR uses the bounding box of the content contained in the cell to adjust the predicted bounding box of the cell.



(a) CSD (b) RD

Figure 4. Architecture of Coordinate Sequence Decoder (CSD) and Regression Decoder (RD). 'TD' indicates the representation of the non-empty cell from HTML Sequence Decoder. To simplify, position encoding is omitted.

Table 7. Ablation studies for structure recognition on FinTabNet test set and IC19B2M. "RD" and "CSD" indicate regression decoder and coordinate sequence decoder, respectively. "VA" refers to visual alignment loss.

Exp	Modules RD	Modules CSD	Modules VA	FinTabNet S-TEDS	FinTabNet AP	IC19B2M WAvg.F1
#1	✓			98.22	87.3	42.5
#2		✓		98.48	95.6	52.1
#3		✓	✓	98.63	96.2	58.6

5.3. Ablation Study

We conduct a set of ablation experiments to verify the effectiveness of our proposed modules. We use FinTabNet for training, and then test on the FinTabNet test set and IC19B2M. The results are in Tab. 7, where the S-TEDS scores for logical structure and detection AP (MS COCO AP at IoU=.50:.05:.95) and WAvg.F1 scores for non-empty cells are reported.

Effectiveness of coordinate sequence decoder. To validate the effectiveness of the Coordinate Sequence Decoder (CSD), we follow TableFormer [25] and TableMaster [44] to implement a Regression Decoder (RD) module, as shown in 4. The difference between the CSD and RD lies in the output header and loss function: 1) By using a Softmax activation function, CSD generates the discrete coordinate sequence (x_{left}, y_{top}, x_{right}, y_{bottom}) one element at a time, which can consume the previously generated coordinate as additional input when generating the next. RD uses the Sigmoid activation function to output the normal-

nition (ICDAR), pages 128–133. IEEE Computer Society, 2019. 2

[27] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, page 2439–2447. IEEE, Jun 2020. 2, 7

[28] Shah Rukh Qasim, J. Kieseler, Y. Iiyama, and Maurizio Pierini. Learning representations of irregular particle-detector geometry with distance-weighted graph networks. *The European Physical Journal C*, 79:1–11, 2019. 2

[29] Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. Rethinking table recognition using graph neural networks. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 142–147, 2019. 2

[30] Liang Qiao, Zaisheng Li, Zhanzhan Cheng, Peng Zhang, Shiliang Pu, Yi Niu, Wenqi Ren, Wenming Tan, and Fei Wu. LGPMA: complicated table structure recognition with local and global pyramid mask alignment. In *2021 International Conference on Document Analysis and Recognition (ICDAR)*, volume 12821, pages 99–114, 2021. 2, 6, 7

[31] Sachin Raja, Ajoy Mondal, and C. V. Jawahar. Table structure recognition using top-down and bottom-up cues. In *Computer Vision – ECCV 2020*, pages 70–86, 2020. 2, 6, 7

[32] Sachin Raja, Ajoy Mondal, and C V Jawahar. Visual understanding of complex table structures from document images. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2543–2552. IEEE Computer Society, jan 2022. 6

[33] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 39, pages 1137–1149, 2015. 2

[34] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1162–1167, 2017. 2

[35] Asif Shahab, Faisal Shafait, Thomas Kieninger, and Andreas Dengel. An open approach towards the benchmarking of table structure recognition systems. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, DAS '10, page 113–120. Association for Computing Machinery, 2010. 6

[36] Shoaib Ahmed Siddiqui, Imran Ali Fateh, Syed Tahseen Raza Rizvi, Andreas R. Dengel, and Sheraz Ahmed. Deeptabstr: Deep learning based table structure recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1403–1409, 2019. 2

[37] Brandon Smock, Rohith Pesala, and Robin Abraham. Grits: Grid table similarity metric for table structure recognition, 2022. 5

[38] Brandon Smock, Rohith Pesala, and Robin Abraham. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4634–4642, June 2022. 5, 6

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. 5

[40] Chris Tensmeyer, Vlad I. Morariu, Brian L. Price, Scott D. Cohen, and Tony Martinez. Deep splitting and merging for table structure decomposition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 114–121, 2019. 2

[41] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, July 2018. 4

[42] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 5

[43] Wenyuan Xue, Qingyong Li, and Dacheng Tao. Res2tim: Reconstruct syntactic structures from table images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 749–755, 2019. 2

[44] Jiaquan Ye, Xianbiao Qi, Yelin He, Yihao Chen, Dengyi Gu, Peng Gao, and Rong Xiao. Pingan-vcgroup's solution for icdar 2021 competition on scientific literature parsing task b: Table recognition to html, 2021. 3, 6, 7

[45] Zhenrong Zhang, Jianshu Zhang, Jun Du, and Fengren Wang. Split, embed and merge: An accurate table structure recognizer. *Pattern Recognition*, 126:108565, 2022. 2, 6, 7

[46] X. Zheng, D. Burdick, L. Popa, X. Zhong, and N. 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 (WACV)*, pages 697–706. IEEE Computer Society, jan 2021. 2, 5, 6, 7

[47] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: Data, model, and evaluation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 564–580, 2020. 2, 5, 6