Table Structure Recognition using Top-Down and Bottom-Up Cues

by

Sachin Raja, Ajoy Mondal, C V Jawahar

Report No: IIIT/TR/2020/-1

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2020

Table Structure Recognition using Top-Down and Bottom-Up Cues

Sachin Raja, Ajoy Mondal, and C V Jawahar

Center for Visual Information Technology,
International Institute of Information Technology, Hyderabad, India
sachinraja13@gmail.com, {ajoy.mondal,jawahar}@iiit.ac.in

Abstract. Tables are information-rich structured objects in document images. While significant work has been done in localizing tables as graphic objects in document images, only limited attempts exist on table structure recognition. Most existing literature on structure recognition depends on extraction of meta-features from the PDF document or on the optical character recognition (OCR) models to extract low-level layout features from the image. However, these methods fail to generalize well because of the absence of meta-features or errors made by the OCR when there is a significant variance in table layouts and text organization. In our work, we focus on tables that have complex structures, dense content, and varying layouts with no dependency on meta-features and/or OCR.
We present an approach for table structure recognition that combines cell detection and interaction modules to localize the cells and predict their row and column associations with other detected cells. We incorporate structural constraints as additional differential components to the loss function for cell detection. We empirically validate our method on the publicly available real-world datasets - ICDAR-2013, ICDAR-2019 (cTDaR) archival, UNLV, SciTSR, SciTSR-COMP, TableBank, and PubTabNet. Our attempt opens up a new direction for table structure recognition by combining top-down (table cells detection) and bottom-up (structure recognition) cues in visually understanding the tables.

Keywords: Document image, table detection, table cell detection, row and column association, table structure recognition.

1 Introduction

Deep neural networks have shown promising results in understanding document layouts [1–3]. However, more needs to be done for structural and semantic understanding. Among these, the problem of table structure recognition has been of high interest in the community [4–20]. Table structure recognition refers to representation of a table in a machine-readable format, where its layout is encoded according to a pre-defined standard [10–14, 17]. It can be represented in

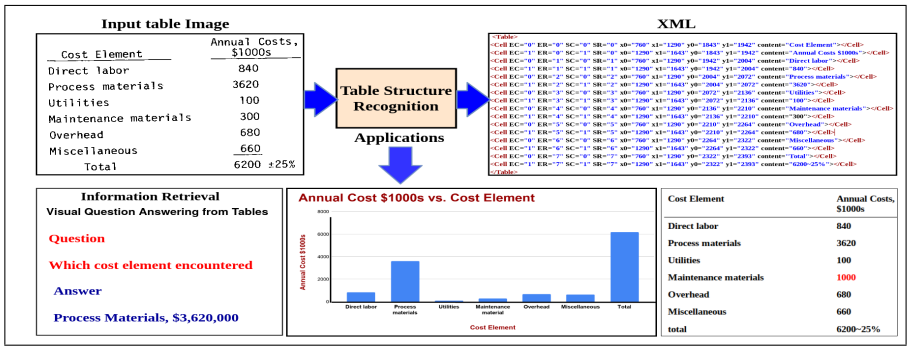

Fig. 1. The figure depicts the problem of recognizing table structure from it's image. This opens up many applications including information retrieval, graphical representation and digitizing for editing.

the form of either physical [10, 12, 14, 17] or logical formats [11, 13]. While logical structure contains every cells' row and column spanning information, physical structure additionally contains bounding box coordinates. Table structure recognition is a precursor to contextual table understanding, which has a myriad of applications in business document analysis, information retrieval, visualization, and human-document interactions, as motivated in Figure 1.

Table structure recognition is a challenging problem due to complex structures and high variability in table layouts [4–17]. Early attempts in this space are dependent on extraction of hand-crafted features and meta-data extracted from the PDFs on top of heuristic/rule-based algorithms [21–24] to locate tables and understanding tables by predicting/recognizing structures. These methods, however, fail to extend to scanned documents as they rely on meta-data information contained in the PDFs. They also make strong assumptions about the structure of the tables. Some of these methods are also dependent on textual information analysis which make them domain dependent. While textual features are useful, visual analysis becomes imperative for analysis of complex page objects. Inconsistency of size and density of tables, presence and location of table cell borders, variation in table cells' shapes and sizes, table cells spanning multiple rows and/or columns and multi-line content are some challenges (refer Figure 2 for some examples) that need to be addressed to solve the problem using visual cues [4, 5, 21–24].

We pose the table structure recognition problem as the generation of XML containing table's physical structure in terms of bounding boxes along with spanning information and, additionally, digitized content for every cell (see Figure 1). Since our method aims to predict this table structure given the table image only (without using any meta-information), we employ a two-step process — (a) *top-down:* where we decompose the table image into fundamental table objects, which are table cells using a cell detection network and (b) *bottom-up:* where we re-build the entire table as a collection of all the table cells localized from

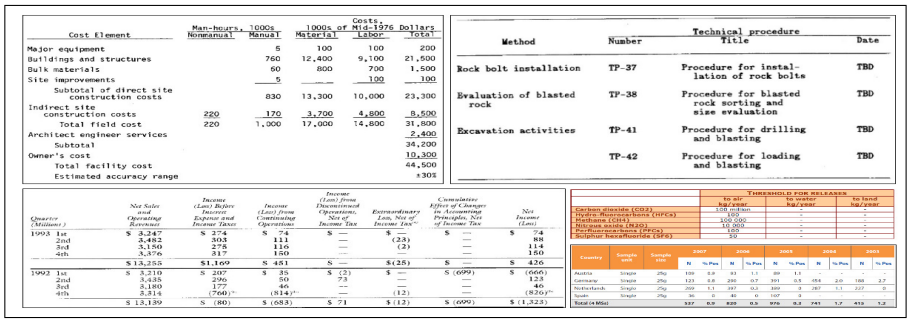

Fig. 2. Examples of complex table images from UNLV and ICDAR-2013 datasets. Complex tables are ones which contain partial or no ruling lines, multi-row/column spanning cells, multi-line content, many empty dense cells.

the top-down process, along with their row and column associations with every other cell. We represent row and column associations of table cells using row and column adjacency matrices.

Though table detection has observed significant success [11, 25–28], detection of table cells remains a challenging problem. This is because of (i) large variation in sizes and aspect ratios of different cells present in the same table, (ii) cells' inherent alignment despite high variance in text amount and text justification, (iii) lack of linguistic context in cells' content, (iv) presence of empty cells and (v) presence of cells with multi-line content. To overcome these challenges, we introduce a novel loss function that models the inherent alignment of cells in the cell detection network; and a graph-based problem formulation to build associations between the detected cells. Moreover, as detection of cells and building associations between them depend highly on one another, we present a novel end-to-end trainable architecture, termed as TabStruct-Net, for cell detection and structure recognition. We evaluate our model for physical structure recognition on benchmark datasets: SciTSR [14], SciTSR-COMP [14], ICDAR-2013 table recognition [18], ICDAR-2019 (cTDaR) archival [19], and UNLV [29]. Further, we extend the comparative analysis of the proposed work for logical structure recognition on TableBank [11] dataset. Our method sets up a new direction for table structure recognition as a collaboration of cell detection, establishing an association between localized cells and, additionally, cells' content extraction.

Our main contributions can be summarised as follows:

- We demonstrate how the top-down (cell detection) and bottom-up (structure recognition) cues can be combined visually to recognize table structures in document images.
- We present an end-to-end trainable network, termed as TabStruct-Net for training cell detection and structure recognition networks in a joint manner.
- We formulate a novel loss function (i.e., alignment loss) to incorporate structural constraints between every pair of table cells and modify Feature Pyra-

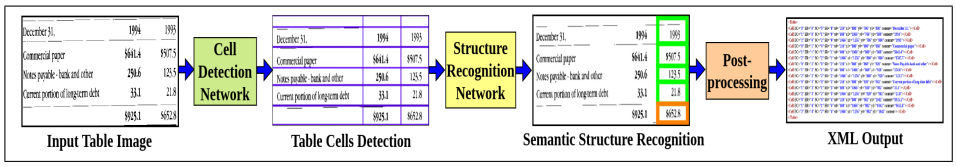

Fig. 3. Block diagram of our approach. Table detection is a precursor to table structure recognition and our method assumes that table is already localized from the input document image. The end-to-end architecture predicts cell bounding boxes and their associations jointly. From the outputs of cell detection and association predictions, XML is generated using a post-processing heuristic.

mid Network (FPN) to capture better low-level and long-range features for cell detection.
- We enhance the visual features representation for structure recognition (built on top of model [9]) through LSTM.
- We unify results from previously published methods on table structure recognition for a thorough comparison study.

2 Related Work

In the space of document images, researchers have been working on understanding equations [30, 31], figures [32, 33] and tables [6–17]. Diverse table layouts, tables with many empty cells and multi-row/column spanning cells are some challenges that make table structure recognition difficult. Research in the domain of table understanding through its structure recognition from document images dated back to the early 1990s when algorithms based on heuristics were proposed [21–24, 34–36]. These methods were primarily dependent on hand-crafted features and heuristics (horizontal and vertical ruling lines, spacing and geometric analysis). To avoid heuristics, Wang et al. [5] proposed a method for table structure analysis using optimization methods similar to the X-Y cut algorithm. Another technique based on column segmentation, header detection, and row segmentation to identify the table structure was proposed by Hu et al. [4]. These methods make strong assumptions about table layouts for a domain agnostic algorithm.

Many cognitive methods [6–12, 14–16, 37–43] have also been presented to understand table structures as they are robust to the input type (whether being scanned images or native digital). These also do not make any assumptions about the layouts, are data-driven, and are easy to fine-tune across different domains. Minghao et al. [11] proposed one class of deep learning methods to directly predict an XML from the table image using the image-to-markup model. Though this method worked well for small tables, it was not robust enough to dense and complex tables. Another set of methods is invoice specific table extraction [39, 40], which were not competent for a more generic use-cases. To

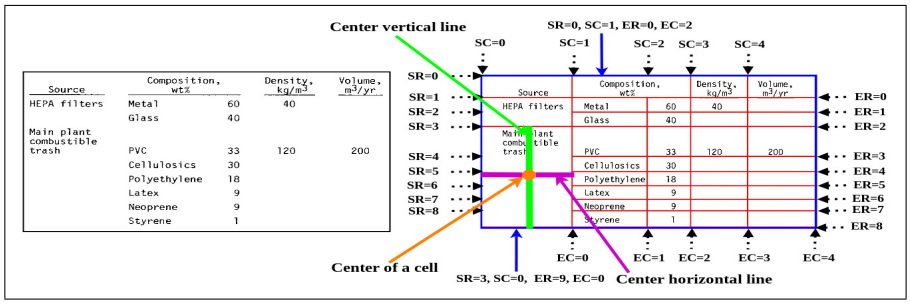

Fig. 4. Visual illustration of cell spanning information along rows and columns of a table from UNLV dataset. **Left Image:** shows original table image in UNLV and **Right Image:** illustrates ground-truth cell spanning information.

overcome this challenge, a combination of heuristics and cognitive methods has also been presented in [12]. Chris et al. [10] presented another interesting deep model, called SPLERGE, which is based on the fundamental idea of first splitting the table into sub-cells, and then merging semantically connected sub-cells to preserve the complete table structure. Though this algorithm showed considerable improvements over earlier methods, it was still not robust to skew present in the table images. Another interesting direction was presented by Vine et al. [42], where they used conditional generative adversarial networks to obtain table skeleton and then fit a latent table structure into the skeleton using a genetic algorithm. Khan et al. [15], through their GRU based sequential models, showed improvements over several CNN based methods for table structure extraction. Recently, many works have preferred a graph-based formulation of the problem as the graph is inherently an ideal data structure to model structural associativity. Qasim et al. [9] proposed a solution where they used graph neural networks to model table-level associativity between words. The authors validate their method on synthetic table images. Chi et al. [14] proposed another graph-based problem formulation and solution using a graph attention mechanism. While these methods made significant progress towards understanding complex structured tables, they made certain assumptions like availability of accurate word bounding boxes, accurate document text, etc. as additional inputs [6, 9, 14]. Our method does not make any such assumptions. We use the table image as the input and produce XML output without any other information. We demonstrate results on complex tables present in UNLV, ICDAR-2013, ICDAR-2019 cTDaR archival, SciTSR, SciTSR-COMP TableBank, and PubTabNet datasets.

3 TabStruct-Net

Our solution for table structure recognition progresses in three steps — (a) detection of table cells; (b) establishing row/column relationships between the

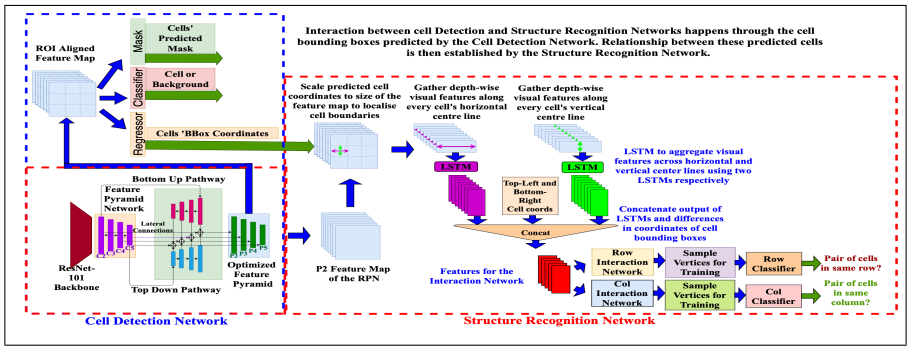

Fig. 5. Our TabStruct-Net. Modified RPN in cell detection network, which consists of both top-down and bottom-up pathways to better capture low-level visual features. P2 layer of the optimized feature pyramid is used in the structure recognition network to extract visual features.

detected cells, and (c) post-processing step to produce the XML output as desired. Figure 3 depicts the block diagram of our approach.

3.1 Top-Down: Cell Detection

The first step of our solution for table structure recognition is localization of individual cells in a table image, for which we use the popular object detection paradigm. The difference from natural scene images, however, is an inherent association between table cells. Recent success of R-CNNs [44] and its improved modifications (Fast R-CNN [45], Faster R-CNN [46], Mask R-CNN [47]) have shown significant success in object detection in natural scene images. Hence, we employ Mask R-CNN [47] for our solution with additional enhancements — (a) we augment the Region Proposal Network (RPN) with dilated convolutions [48, 49] to better capture long-range row and column visual features of the table. This improves detection of multi-row/column spanning and multi-line cells ; (b) inspired by [50], we append the feature pyramid network with a top-down pathway, which propagates high-level semantic information to low-level feature maps. This allows the network to work better for cells with varying scales; and (c) we append additional losses during the training phase in order to model the inherent structural constraints. We formulate two ways of incorporating this information — (i) through an end-to-end training of cell detection and the structure recognition networks (explained next), and (ii) through a novel alignment loss function. For the latter, we make use of the fact that every pair of cells is aligned horizontally if they span the same row and aligned vertically if they span the same column. For the ground truth, where tight bounding boxes around the cells' content are provided [18, 14, 13], we employ an additional ground truth pre-processing step to ensure that bounding boxes of cells in the same row and same column are aligned vertically and horizontally, respectively. We model these constraints

during the training in the following manner:

$$L_1 = \sum_{r \in SR} \sum_{c_i, c_j \in r} ||y1_{c_i} - y1_{c_j}||_2^2,\ L_2 = \sum_{r \in ER} \sum_{c_i, c_j \in r} ||y2_{c_i} - y2_{c_j}||_2^2$$
$$L_3 = \sum_{c \in SC} \sum_{c_i, c_j \in c} ||x1_{c_i} - x1_{c_j}||_2^2 \text{ and } L_4 = \sum_{c \in EC} \sum_{c_i, c_j \in c} ||x2_{c_i} - x2_{c_j}||_2^2$$

Here, SR, SC, ER and EC represent starting row, starting column, ending row and ending column indices as shown in Figure 4. Also, c_i and c_j denote two cells in a particular row r or column c; $x1_{c_i}$, $y1_{c_i}$, $x2_{c_i}$ and $y2_{c_i}$ represent bounding box coordinates X-start, Y-start, X-end and Y-end respectively of the cell c_i. These losses (L_1, L_2, L_3, L_4) can be interpreted as constraints that enforce proper alignment of cells beginning from same row, ending on same row, beginning from same column and ending on same column respectively. Alignment loss is defined as

$$L_{align} = L_1 + L_2 + L_3 + L_4. \tag{1}$$

3.2 Bottom-Up: Structure Recognition

We formulate the table structure recognition using graphs similar to [9]. We consider each cell of the table as a vertex and construct two adjacency matrices - a row matrix M_{row} and a column matrix M_{col} which describe the association between cells with respect to rows and columns. $M_{row}, M_{col} \in \mathbb{R}^{N_{cells} \times N_{cells}}$. $M_{row_{i,j}} = 1$ or $M_{col_{i,j}} = 1$ if cells i, j belong to the same row or column, else 0.

The structure recognition network aims to predict row and column relationships between the cells predicted by the cell detection module during training and testing. During training, only those predicted table cells are used for structure recognition which overlap with the ground truth table cells having an IoU greater than or equal to 0.5. This network has three components:

- *Visual Component:* We use visual features from P2 layer (refer Figure 5) of the feature pyramid based on the linear interpolation of cell bounding boxes predicted by the cell detection module. In order to encode cells' visual characteristics across their entire height and width, we pass the gathered P2 features for every cell along their centre horizontal and centre vertical lines using LSTM [51] to obtain the final visual features (refer Figure 5) (as opposed to visual features corresponding to cells' centroids only as in [52]).
- *Interaction Component:* We use the DGCNN architecture based on graph neural networks used in [52] to model the interaction between geometrically neighboring detected cells. It's output, termed as interaction features, is a fixed dimensional vector for every cell that has information aggregated from its neighbouring table cells.
- *Classification Component:* For a pair of table cells, the interaction features are concatenated and appended with difference between cells' bounding box coordinates. This is fed as an input to the row/column classifiers to predict row/column associations. Please note that we use the same [52] Monte Carlo based sampling to ensure efficient training and class balancing. During testing time, however, predictions are made for every unique pair of table cells.

We train the cell detection and structure recognition networks in a joint manner (termed as TabStruct-Net) to collectively predict cell bounding boxes along with row and column adjacency matrices. Further, the two structure recognition pathways for row and column adjacency matrices are put together in parallel. The visual features prepared using LSTMs for every vertex are duplicated for both the pathways, after which they work in a parallel manner. The overall empirical loss of TabStruct-Net is given by:

$$L = L_{box} + L_{cls} + L_{mask} + L_{align} + L_{gnn}, \tag{2}$$

where L_{box}, L_{cls} and L_{mask} are bounding box regression loss, classification loss and mask loss, respectively defined in Mask R-CNN [47], L_{align} is alignment loss which is modeled as a regularizer (defined in Eq. 1) and L_{gnn} is the cross-entropy loss back propagated from the structure recognition module of TabStruct-Net. The additional loss components help the model in better alignment of cells belonging to same rows/columns during training, and in a sense fine-tunes the predicted bounding boxes that makes it easier for post-processing and structure recognition in the subsequent step.

3.3 Post-Processing

Once all the cells and their row/column adjacency matrices are predicted, we create the XML interpretable output as a post-processing step. From the cell coordinates along with row and column adjacency matrix, SR, SC, ER and EC indexes are assigned to each cell, which indicate spanning of that cell along rows and columns. We use Tesseract [53] to extract the content of every predicted cell. The XML output for every table image finally contains coordinates of predicted cell bounding boxes and along with cell spanning information and its content.

4 Experiments

4.1 Datasets

We use various benchmark datasets — SciTSR [14], SciTSR-COMP [14], ICDAR-2013 table recognition [18], ICDAR-2019 (cTDaR) archival [19], UNLV [29], Marmot extended [12], TableBank [11] and PubTabNet [13] datasets for extracting structure information of tables. Statistics of these datasets are listed in Table 1.

	SciTSR	SciTSR COMP	ICDAR 2013	ICDAR-2013 -partial	ICDAR 2019	UNLV	UNLV-partial	Marmot extended	Table Bank	PubTabNet
Train	12000	12000	-	124	600	-	446	1016	145K	339K
Test	3000	716	158	34	150	558	112	-	1000	114K

Table 1. Statistics of the datasets used for our experiments.

4.2 Baseline Methods

We compare the performance of our TabStruct-Net against seven benchmark methods — DeepDeSRT [7], TableNet [12], GraphTSR [14], SPLERGE [10], DGCNN [9], Bi-directional GRU [15] and Image-to-Text [11].

4.3 Implementation Details

TabStruct-Net[1] has been trained and evaluated with table images scaled to a fixed size of 1536×1536 while maintaining the original aspect ratio as the input. While training, cell-level bounding boxes along with row and column adjacency matrices (prepared from start-row, start-column, end-row and end-column indices) are used as the ground truth. We use NVIDIA TITAN X GPU with 12 GB memory for our experiments and a batch-size of 1. Instead of using 3×3 convolution on the output feature maps from the FPN, we use a dilated convolution with filter size of 2×2 and dilation parameter of 2. Also, we use the ResNet-101 backbone that is pre-trained on MS-COCO [54] dataset. Dilated convolution blocks of filter size 7 are used in the FPN. To compute region proposals, we use 0.5, 1 and 2 as the anchor scale and anchor box sizes of 8, 16, 32, 64 and 128. LSTMs used to gather visual features have a depth of 128. The final memory state of the LSTM layers is concatenated with the cell's coordinates to prepare features for the interaction network. Further, for generation of the row/column adjacency matrices, we use 2400 as the maximum number of vertices keeping in mind dense tables. Next, features from 40 neighboring vertices are aggregated using an edge convolution layer followed by a dense layer of size 64 with ReLu activation. Since every input table may contain hundreds of table cells, training can be a time consuming process. To achieve faster training, we employ a two-stage training process. In the first stage, we use 2014 anchors and 512 RoIs, and in the second stage, we use with 3072 anchors and 2048 RoIs. During both the stages, we use 0.001 as the learning rate, 0.9 as the momentum and 0.0001 as the weight decay regularisation.

4.4 Evaluation Measures

We use various existing measures — precision, recall and F1 [14, 18, 29] to evaluate the performance of our model for recognition of physical structure of tables. For recognition of logical structure of tables, we use BLEU [55] score as used in [11] and Tree-Edit-Distance-based similarity (TEDS) [13]. Since XML is our final output for table structure recognition, we also use BLEU [55], CIDEr [56] and ROUGE [57] scores to compare generated XML and ground truth XML on spanning information and content of every cell. We first calculate these scores separately on each table and then compute both micro-averaged score and macro-averaged score as the final result. We consistently use an IoU threshold of 0.6 to compute the confusion matrix. Please note that only non-empty table cells are considered similar to [18] for the evaluation.

[1] Our code is available at https://github.com/sachinraja13/TabStructNet.git

Test Dataset	Train Dataset	S-A			S-B		
		P↑	R↑	F1↑	P↑	R↑	F1↑
ICDAR-2013	SciTSR	0.915	0.897	0.906	0.976	0.985	0.981
ICDAR-2013-partial	SciTSR	0.930	0.908	0.919	0.991	0.993	0.992
SciTSR	SciTSR	0.927	0.913	0.920	0.989	0.993	0.991
SciTSR-comp	SciTSR	0.909	0.882	0.895	0.981	0.987	0.984
UNLV-partial	SciTSR	0.849	0.828	0.839	0.992	0.994	0.993
ICDAR-2019	SciTSR	0.595	0.572	0.583	0.924	0.899	0.911
ICDAR-2019	ICDAR-2019	0.803	0.768	0.785	0.975	0.957	0.966
ICDAR-2019	SciTSR+ICDAR-2019	0.822	0.787	0.804	0.975	0.958	0.966

Table 2. shows the performance of our TabStruct-Net for physical table structure recognition on various benchmark datasets.

Test Dataset	Train Dataset	Metric	Score
TableBank-Word	SciTSR	BLEU	0.914
TableBank-LaTeX	SciTSR	BLEU	0.937
TableBank-Word+LaTeX	SciTSR	BLEU	0.916
PubTabNet	SciTSR	TEDS	0.901

Table 3. shows the performance of our TabStruct-Net for logical table structure recognition on various benchmark datasets.

4.5 Experimental Setup

One major challenge in the comparison study with the existing methods is the inconsistent use of additional information (e.g., meta-features extracted from the PDFs [10], content-level bounding boxes from ground truths [12, 14] and cell's location features generated from synthetic dataset [9]). Hence, we do experiments in two different setups

- **Setup-A (S-A):** using only table image as the input
- **Setup-B (S-B):** using table image along with additional information (e.g., cell bounding boxes) as the input. For this, instead of removing the cell detection component from the network, we ignore the predicted boxes and use the ground truth ones as input for structure recognition.

5 Results on Table Structure Recognition

Tables 2 and 3 summarize the performance of our model on standard datasets used in the space of table structure recognition.

5.1 Analysis of Results

Table 4 presents results on ICDAR-2013 dataset. In S-A, we observe that our model outperforms DeepDeSRT [7] method by a 27.5% F1 score. This is because cell coordinates for the latter are obtained by row and column intersections,

making it unable to recognize cells that span multiple rows/columns. For dense tables (small inter-row spacing), row segmentation results of DeepDeSRT combined multiple rows into one in several instances. Split+Heuristic [10] method outperforms TabStruct-Net by a small margin, however, it requires ICDAR-2013 dataset-specific cell merging heuristics and is trained on a considerably larger set of images. Therefore, a direct comparison of (Split+Heuristic) with our method is not fair. Nevertheless, comparable results of TabStruct-Net indicates its robustness to ICDAR-2013 dataset, without using any kind of dataset-specific post-processing. However, if compared under the same training environment and no post-processing, our model outperforms SPLERGE with a 3% average F1 score. SPLERGE works well for datasets where ground truth bounding boxes are annotated at the content-level instead of cell-level. This is because it allows for a wider area for a prospective prediction of a row/column separator. Further, since it is based on cell detection through the row and column separators, it is not agnostic to input image noise such as skew and rotations. This method is susceptible to dataset-specific post-processing as opposed to ours, where no post-processing is needed.

Method	**Training**		**Experimental**	**P↑**	**R↑**	**F1↑**
	Dataset	**#Images**	**Setup**			
DeepDeSRT [7]	SciTSR	12K	S-A	0.631	0.619	0.625
SPLERGE [10]	SciTSR	12K	S-A	0.883	0.875	0.879
Split+Heuristic [10]	Private [10]	83K	S-A	**0.938**	**0.922**	**0.930**
TabStruct-Net (our)	SciTSR	12K	S-A	0.915	0.897	0.906
TableNet [12]	Marmot Extended	1K	S-B	0.922	0.899	0.910
GraphTSR [14]	SciTSR	12K	S-B	0.885	0.860	0.872
Split-PDF [10]	Private [10]	83K	S-B	0.920	0.913	0.916
Split-PDF +Heuristic [10]	Private [10]	83K	S-B	0.959	0.946	0.953
DGCNN [9]	SciTSR	12K	S-B	0.972	0.983	0.977
TabStruct-Net (our)	SciTSR	12K	S-B	**0.976**	**0.985**	**0.981**

Table 4. Comparison of results for physical structure recognition on ICDAR-2013 dataset. **#Images:** indicates number of table images in the training set. **Heuristic:** indicates dataset specific cell merging rules for various models in [10].

In S-B, TabStruct-Net sets up a state-of-the-art benchmark on the ICDAR-2013 dataset, outperforming all the existing methods [9, 10, 14, 12]. It is further interesting to note that our technique outperforms Split-PDF+Heuristic model also without needing any post-processing. It is because our enhancements to the DGCNN [9] model can capture the visual characteristics of a cell across a larger span through LSTMs. We observe that our model achieves significantly improved performance when content-level bounding boxes are used instead of cell-level, which are much easier to obtain with the help of OCR tools and PDF meta-information.

CD Network	SR Network	IoU TH	CD Scores P↑	R↑	F1↑	SR Scores P↑	R↑	F1↑
Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	0.5	**0.935**	**0.942**	**0.938**	**0.927**	**0.911**	**0.919**
		0.6	0.921	0.926	0.923	0.915	0.897	0.906
		0.7	0.815	0.820	0.817	0.797	0.785	0.791
		0.8	0.638	0.653	0.645	0.629	0.615	0.622
		0.9	0.275	0.312	0.292	0.247	0.236	0.241

Table 5. Physical structure recognition results on ICDAR-2013 dataset for varying IoU thresholds to demonstrate TabStruct-Net's robustness. **ES:** Experimental Setup, **CD:** Cell Detection, **TH:** IoU threshold value, **SR:** Structure Recognition, **P2:** using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** use of Top-Down and Bottom-Up pathways in the FPN, **AL:** addition of alignment loss as a regularizer to TabStruct-Net.

Table 5 shows the performance of our technique under the varying IoU thresholds. It can be inferred from the table that our model achieves an F1 score of 79.1% on structure recognition with an IoU threshold value of as high as 0.7. For the IoU values of 0.5 and 0.6, our model's performance is 91.9% and 90.6%, respectively. It demonstrates the robustness of our model. Figures 6 and 7 display some qualitative outputs of our method on the datasets discussed in Section 4.1. Figure 8 shows some of the failure cases of cell detection by our method. It can be seen that our model fails for table images that have large amounts of empty spaces. Supplementary material has (i) more quantitative results, (ii) more qualitative examples, (iii) specific implementation details, (iv) detailed comparative analysis, IoU variation results, and ablation study on all the datasets.

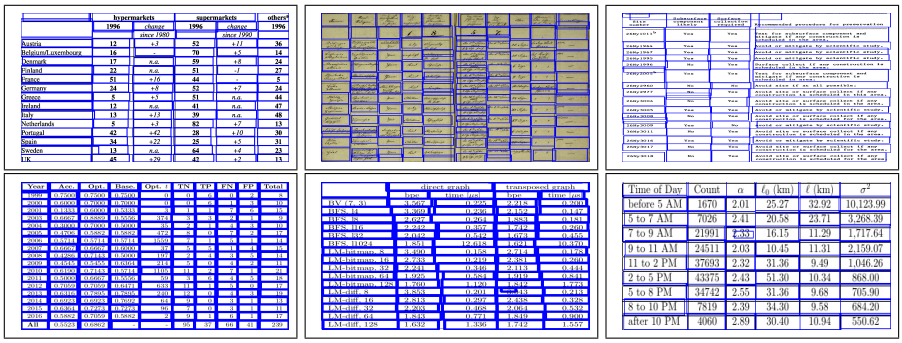

Fig. 6. Sample intermediate cell detection results of TabStruct-Net on table images of ICDAR-2013, ICDAR-2019 cTDaR and UNLV, SciTSR, SciTSR-COMP and TableBank datasets.

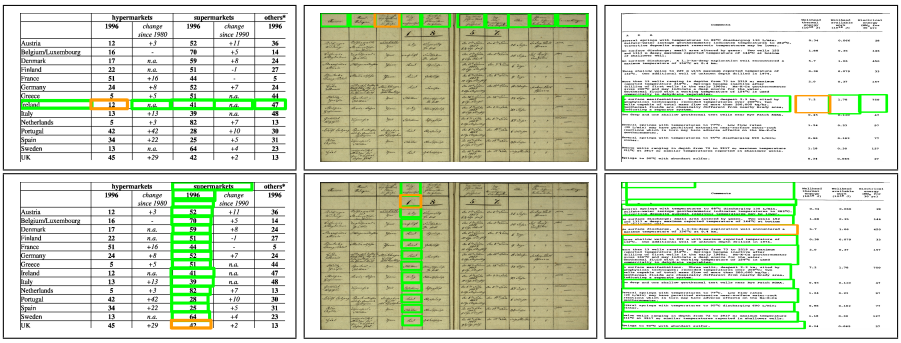

Fig. 7. Sample structure recognition output of TabStruct-Net on table images of ICDAR-2013, ICDAR-2019 cTDaR archival and UNLV datasets. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

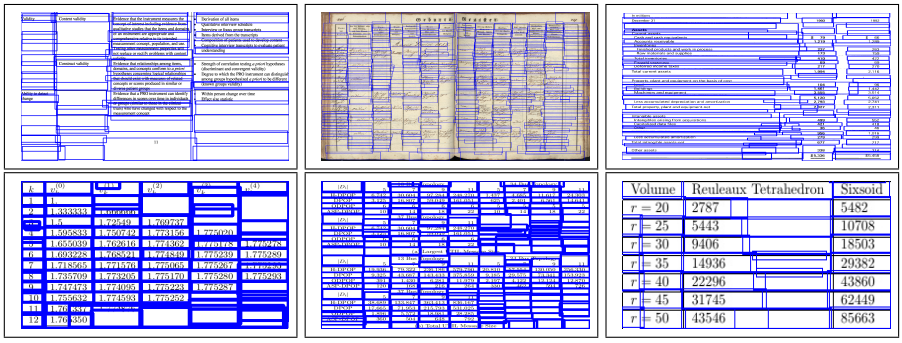

Fig. 8. Sample intermediate cell detection results of TabStruct-Net on table images of ICDAR-2013, ICDAR-2019 cTDaR, UNLV, SciTSR, SciTSR-COMP and TableBank datasets illustrate failure of TabStruct-Net.

5.2 Ablation Study

Table 6 shows the outcome of our enhancements to Mask R-CNN [47] and DGCNN [9] models for both cell detection and structure recognition networks under S-A and S-B. From the table, it can be observed that our additions to the networks result in a significant increase of 4% average F1 scores on cell detection and structure recognition tasks. The novel alignment loss, along with the use of top-down and bottom-up pathways in the FPN results in an improvement of 2.3% F1 score for cell detection and 2.4% on structure recognition. Use of LSTMs and P2 layer output to prepare visual features for structure recognition results in a 2.1% improvement of F1 scores. Interestingly, because both models are trained together

in an end-to-end fashion, cell detection's effect is also observed in the form of a 1.5% average F1 score. This empirically bolsters our claim of using an end-to-end architecture for cell detection and, in turn, structure recognition.

ES	CD Network	SR Network	CD Scores			SR Scores		
			P↑	R↑	F1↑	P↑	R↑	F1↑
S-A	Mask R-CNN	DGCNN	0.885	0.890	0.887	0.871	0.860	0.865
	Mask R-CNN	DGCNN+P2	0.886	0.892	0.889	0.877	0.863	0.870
	Mask R-CNN	DGCNN+P2+LSTM	0.898	0.904	0.901	0.885	0.879	0.882
	Mask R-CNN+TD+BU	DGCNN	0.895	0.899	0.897	0.883	0.867	0.875
	Mask R-CNN+TD+BU	DGCNN+P2	0.895	0.901	0.898	0.886	0.870	0.878
	Mask R-CNN+TD+BU	DGCNN+P2+LSTM	0.904	0.910	0.907	0.892	0.884	0.888
	Mask R-CNN+TD+BU+AL	DGCNN	0.905	0.911	0.908	0.891	0.879	0.885
	Mask R-CNN+TD+BU+AL	DGCNN+P2	0.914	0.920	0.917	0.906	0.885	0.895
	Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	**0.921**	**0.926**	**0.924**	**0.915**	**0.897**	**0.906**
S-B	-NA-	DGCNN	-NA-	-NA-	-NA-	0.972	0.983	0.977
	-NA-	DGCNN+P2	-NA-	-NA-	-NA-	0.973	0.983	0.978
	-NA-	DGCNN+P2+LSTM	-NA-	-NA-	-NA-	**0.976**	**0.985**	**0.981**

Table 6. Ablation study for physical structure recognition on ICDAR-2013 dataset. **ES:** Experimental Setup, **CD:** Cell Detection, **SR:** Structure Recognition, **P2:** using visual features from P2 layer of the FPN instead of using separate convolution blocks, **LSTM:** use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, **TD+BU:** use of Top-Down and Bottom-Up pathways in the FPN, **AL:** addition of alignment loss as a regularizer to TabStruct-Net.

6 Summary

We formulate the problem of table structure recognition as a combination of cell detection (top-down) and structure recognition (bottom-up) tasks. For cell detection, we make a modification to the RPN of original Mask R-CNN and introduce a novel alignment loss function (formulated for every pair of table cells) to enforce structural constraints. For structure recognition, we improve input representation for the DGCNN network by using LSTM, pre-trained ResNet-101 backbone and RPN of cell detection network. Further, we propose an end-to-end trainable architecture to collectively predict cell bounding boxes along with their row and column adjacency matrices to predict structure. We demonstrate our results on multiple public datasets on both digital scanned as well as archival handwritten table images. We observe that our approach fails to handle tables containing a large number of empty cells along both horizontal and vertical directions. In conclusion, we encourage further research in this direction.

7 Acknowledgment

This work is partly supported by MEITY, Government of India.

References

1. Yang, X., Yumer, E., Asente, P., Kraley, M., Kifer, D., Lee Giles, C.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In: CVPR. (2017)
2. Augusto Borges Oliveira, D., Palhares Viana, M.: Fast CNN-based document layout analysis. In: ICCV. (2017)
3. Yi, X., Gao, L., Liao, Y., Zhang, X., Liu, R., Jiang, Z.: CNN based page object detection in document images. In: ICDAR. (2017)
4. Hu, J., Kashi, R.S., Lopresti, D.P., Wilfong, G.: Medium-independent table detection. In: Document Recognition and Retrieval VII. (1999)
5. Wang, Y., Phillips, I.T., Haralick, R.M.: Table structure understanding and its performance evaluation. Pattern Recognition (2004)
6. Nishida, K., Sadamitsu, K., Higashinaka, R., Matsuo, Y.: Understanding the semantic structures of tables with a hybrid deep neural network architecture. In: AAAI. (2017)
7. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: DeepDeSRT: Deep learning for detection and structure recognition of tables in document images. In: ICDAR. (2017)
8. Bao, J., Tang, D., Duan, N., Yan, Z., Lv, Y., Zhou, M., Zhao, T.: Table-to-text: Describing table region with natural language. In: AAAI. (2018)
9. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table parsing using graph neural networks. In: ICDAR. (2019)
10. Tensmeyer, C., Morariu, V., Price, B., Cohen, S., Martinezp, T.: Deep splitting and merging for table structure decomposition. In: ICDAR. (2019)
11. Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: TableBank: Table benchmark for image-based table detection and recognition. In: ICDAR. (2019)
12. Paliwal, S.S., Vishwanath, D., Rahul, R., Sharma, M., Vig, L.: TableNet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In: ICDAR. (2019)
13. Zhong, X., ShafieiBavani, E., Yepes, A.J.: Image-based table recognition: data, model, and evaluation. arXiv (2019)
14. Chi, Z., Huang, H., Xu, H.D., Yu, H., Yin, W., Mao, X.L.: Complicated table structure recognition. arXiv (2019)
15. Khan, S.A., Khalid, S.M.D., Shahzad, M.A., Shafait, F.: Table structure extraction with Bi-directional Gated Recurrent Unit networks. In: ICDAR. (2019)
16. Siddiqui, S.A., Khan, P.I., Dengel, A., Ahmed, S.: Rethinking semantic segmentation for table structure recognition in documents. In: ICDAR. (2019)
17. Xue, W., Li, Q., Tao, D.: ReS2TIM: Reconstruct syntactic structures from table images. In: ICDAR. (2019)
18. Göbel, M., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 table competition. In: ICDAR. (2013)
19. Gao, L., Huang, Y., Déjean, H., Meunier, J.L., Yan, Q., Fang, Y., Kleber, F., Lang, E.: ICDAR 2019 competition on table detection and recognition (cTDaR). In: ICDAR. (2019)
20. Mondal, A., Lipps, P., Jawahar, C.V.: IIIT-AR-13K: a new dataset for graphical object detection in documents. In: DAS. (2020)
21. Itonori, K.: Table structure recognition based on textblock arrangement and ruled line position. In: ICDAR. (1993)

22. Green, E., Krishnamoorthy, M.: Recognition of tables using table grammars. In: Annual Symposium on Document Analysis and Information Retrieval. (1995)
23. Kieninger, T.G.: Table structure recognition based on robust block segmentation. In: Document Recognition V. (1998)
24. Tupaj, S., Shi, Z., Chang, C.H., Alam, H.: Extracting tabular information from text files. EECS Department, Tufts University, Medford, USA (1996)
25. Gilani, A., Qasim, S.R., Malik, I., Shafait, F.: Table detection using deep learning. In: ICDAR. (2017)
26. Dong, H., Liu, S., Han, S., Fu, Z., Zhang, D.: TableSense: Spreadsheet table detection with convolutional neural networks. In: AAAI. (2019)
27. Kavasidis, I., Pino, C., Palazzo, S., Rundo, F., Giordano, D., Messina, P., Spampinato, C.: A saliency-based convolutional neural network for table and chart detection in digitized documents. In: ICIAP. (2019)
28. Saha, R., Mondal, A., Jawahar, C.V.: Graphical object detection in document images. In: ICDAR. (2019)
29. Shahab, A., Shafait, F., Kieninger, T., Dengel, A.: An open approach towards the benchmarking of table structure recognition systems. In: DAS. (2010)
30. Zanibbi, R., Blostein, D., Cordy, J.R.: Recognizing mathematical expressions using tree transformation. IEEE Trans. on PAMI (2002)
31. Zhang, J., Du, J., Dai, L.: Multi-scale attention with dense encoder for handwritten mathematical expression recognition. In: ICDAR. (2018)
32. Siegel, N., Horvitz, Z., Levin, R., Divvala, S., Farhadi, A.: FigureSeer: Parsing result-figures in research papers. In: ECCV. (2016)
33. Tang, B., Liu, X., Lei, J., Song, M., Tao, D., Sun, S., Dong, F.: DeepChart: Combining deep convolutional networks and deep belief networks in chart classification. Signal Processing (2015)
34. Harit, G., Bansal, A.: Table detection in document images using header and trailer patterns. In: ICVGIP. (2012)
35. Gatos, B., Danatsas, D., Pratikakis, I., Perantonis, S.J.: Automatic table detection in document images. In: CVPR. (2005)
36. Ohta, M., Yamada, R., Kanazawa, T., Takasu, A.: A cell-detection-based table-structure recognition method. In: ACM Symposium on Document Engineering. (2019)
37. Deng, Y., Rosenberg, D., Mann, G.: Challenges in end-to-end neural scientific table recognition. In: ICDAR. (2019)
38. Adiga, D., Bhat, S.A., Shah, M.B., Vyeth, V.: Table structure recognition based on cell relationship, a bottom-up approach. In: RANLP. (2019)
39. Riba, P., Dutta, A., Goldmann, L., Fornes, A., Ramos, O., Llados, J.: Table detection in invoice documents by graph neural networks. In: ICDAR. (2019)
40. Holeček, M., Hoskovec, A., Baudiš, P., Klinger, P.: Line-items and table understanding in structured documents. arXiv (2019)
41. Deng, L., Zhang, S., Balog, K.: Table2Vec: Neural word and entity embeddings for table population and retrieval. In: SIGIR. (2019)
42. Le Vine, N., Zeigenfuse, M., Rowan, M.: Extracting tables from documents using conditional generative adversarial networks and genetic algorithms. In: IJCNN. (2019)
43. Sage, C., Aussem, A., Elghazel, H., Eglin, V., Espinas, J.: Recurrent neural network approach for table field extraction in business documents. In: ICDAR. (2019)
44. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)

45. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
46. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
47. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: CVPR. (2017)
48. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv (2015)
49. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE trans. on PAMI (2017)
50. Woo, S., Hwang, S., Jang, H.D., Kweon, I.S.: Gated bidirectional feature pyramid network for accurate one-shot detection. Machine Vision and Applications (2019)
51. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation (1997)
52. Qasim, S.R., Kieseler, J., Iiyama, Y., Pierini, M.: Learning representations of irregular particle-detector geometry with distance-weighted graph networks. arXiv (2019)
53. Smith, R.: An overview of the Tesseract OCR engine. In: ICDAR. (2007)
54. Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR (2014)
55. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: AMACL. (2002)
56. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: CIDEr: Consensus-based image description evaluation. In: CVPR. (2015)
57. Lin, C.Y.: ROUGE: A package for automatic evaluation of summaries. Text Summarization Branches Out (2004)