Supplementary Material: 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

1 Organization

We organize our supplementary paper in the following manner:

- Section 2 contains additional details about our methodology including the post-processing step to obtain the final XML output.
- Section 3.1 contains information about the datasets used for experiments and explains the ground-truth unification process.
- Section 3.2 briefly explains various previously published methods in the space of table structure recognition including the input modalities used for predictions.
- Section 3.3 talks about the very specific details of implementation for training our model.
- Section 3.4 explains the evaluation method in greater detail which will be useful for reproducing our results. It explains various assumptions and reasons for choosing the specific metrics.
- Section 3.5 lists out various challenges encountered for evaluating and comparing our method with the existing ones, and also talks about the ways in which we handled them in our work.
- Section 4 presents additional quantitative and qualitative results for a detailed analysis.

2 TabStruct-Net

Our TabStruct-Net is a data-driven and an end-to-end trainable architecture for the prediction of table structure from a given table image, that combines top-down and bottom-up methods. As a first step, the input table image is broken down into individual cells using the cell detection network of the TabStruct-Net. We call this as the top-down step of the process. After detecting individual cells, the next step is to obtain the entire table structure by building relevant row and column associations between the detected cells. This is done using the structure recognition network of the TabStruct-Net and we call this as the bottom-up step of the process.

2.1 Post-processing to Get XML Output

After the cell bounding boxes along with the row and column adjacency matrices are obtained, an XML file is generated using an heuristic based algorithm. It works as follows:

- For row assignments, sort all bounding boxes by their $start_y$ coordinates, and initialize a row belonging list for every cell.
- Assign a row belonging index (starting from 0) to the cell c_i and assign the same row index to all other cells that are connected to c_i in the row adjacency matrix.
- Increment the row index and repeat the above step until all the cells are assigned at least one row belonging index.
- For each cell, SR is the minimum of indexes in the row belonging list, and ER is the maximum of indexes in the row belonging list.
- Similarly, for column assignments, sort all bounding boxes by their $start_x$ coordinates, and initialize a column belonging list for every cell.
- Assign a column belonging index (starting from 0) to the cell c_i and assign the same column index to all other cells that are connected to c_i in the column adjacency matrix.
- Increment the column index and repeat the above step until all the cells are assigned at least one column belonging index.
- For each cell, SC is the minimum of indexes in the row belonging list, and EC is the maximum of indexes in the column belonging list.

We use Tesseract [43] to extract the content of every predicted cell. Once SR, ER, SC and EC values (referred to as cell spanning values) and its content are obtained for every predicted cell, an XML file is created with these cell spanning values along with bounding box coordinates (top-left and bottom-right of the cell) and its content.

3 Experiments

3.1 Dataset

We use various benchmark table structure recognition datasets — SciTSR [14], SciTSR-COMP [14], ICDAR-2013 table recognition [18], ICDAR-2019 cTDaR archival [19], UNLV [28], Marmot extended [12], TableBank [11] and PubTabNet [13] datasets for extracting structure information of tables. Statistics of these datasets are listed in Table 1 (main paper).

Our TabStruct-Net makes an assumption that all cells belonging to the same column are aligned with respect to x coordinates and cells belonging to the same row are aligned with respect to y coordinates. SciTSR[14], SciTSR-COMP [14] and ICDAR-2013 [18] datasets have ground truth bounding boxes at the level of cell's content (box is the smallest rectangular block that encapsulates the cell's content). To handle this, we expand the bounding boxes of every cell in a row and column to get maximum sized content-level box in a particular row and column.

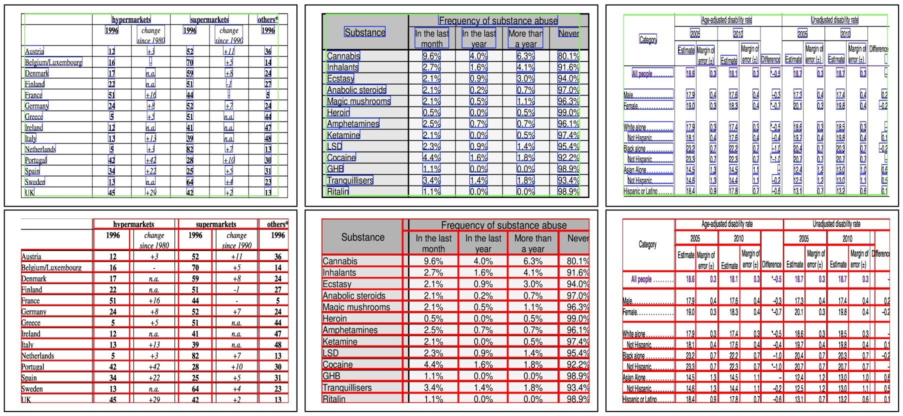

Fig. 1. Ground truth unification. content-level bounding boxes are given in ground truth as shown in **First Row**. We make content-level bounding boxes into cell-level bounding boxes as shown in **Second Row**.

3.2 Baseline Methods

We compare the performance of our TabStruct-Net against seven following benchmark methods.

DeepDeSRT [7]: This method leverages semantic segmentation approach to localize each row and column from the table image. This method outputs table as a grid-like structure and fails to identify multiple row and multiple column spanning cells. Since no code is available, we implement our own version of this method. Since this method works by predicting every row and column, the SciTSR training dataset is pre-processed to obtain row and column level coordinates before training.

TableNet [12]: This method uses semantic segmentation approach to extract table and column masks, and segments rows by identifying words present in different columns (extracted using Tesseract OCR [43]) that occur at the same horizontal level. For comparison against other methods, we directly use the results reported by the authors.

GraphTSR [14]: This method consists of edge-to-vertex and vertex-to-edge graph attention blocks to compute vertex and edge representations in a latent space, and finally classify each edge as 'horizontal', 'vertical' or 'no-relation'. It uses absolute and relative positions of every cell extracted from the PDF to compute initial vertex and edge features.

SPLERGE [10]: This method comprises of two deep learning networks that perform splitting and merging operations in sequence to predict fine grid-like table structure and to predict merged cells which span multiple rows/columns. Split method shows an improved performance when additional PDF extracted meta-features are provided along with the table image. For the split model (to obtain the basic grid of the table), authors pre-process the ground truth by maximizing the row and column separator regions without intersecting any non multiple row or column spanning cell. For the merge model (to identify cells that span multiple rows or columns), the authors prepare the ground truth by identifying grid elements that span multiple cells and set the merging probability in the respective directions. Further, for evaluating this method on ICDAR-2013 [18] dataset, the authors realized that merge method did not work with a good performance, and hence, introduced the following specific heuristics to merge cells instead:

– Merge cells where separators pass through text.
– Merge adjacent blank columns with cells that have been formed by merging many cells.
– Merge adjacent blank rows with content cells.
– Split columns that have a consistent white-space gap between vertically aligned text.

DGCNN* [9]: Authors formulate the problem as a graph learning problem to predict whether every pair of words belongs to the same cell, row and/or column or not. Their architecture consists of a visual network, an interaction network and a classification network. For evaluation purposes, table image along with word-level bounding boxes is provided as inputs.

Bi-directional GRU [15]: Given the table image, this method uses two bi-directional GRUs to establish row and column boundaries in a context driven manner. This method however fails to localize multiple row and/or multiple column spanning cells.

Image-to-Text [11]: This method utilizes an Image-to-Markup model to predict a markup-like output of a given table image. It consists of a CNN based encoder to compute visual features and an LSTM based decoder to produce markup output.

3.3 Implementation Details

Our TabStruct-Net model[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

[1] Our code will be available publicly

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 [48] dataset. 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. Further, for generation of the row/column adjacency matrices, we use 2400 as the maximum number of vertices keeping in mind dense tables. Since every input table may contain hundreds of table cells, training can be a time consuming process.

To achieve faster training, we employ a 2-stage training process. In the first step, we use 2014 anchors and 512 RoIs. With this setting, the model is able to learn high and low level features but resulted in a large number of false negatives. To combat this, network is trained with 3072 anchors and 2048 RoIs. This significantly reduces the number of false negatives. For the first step, we train a total of 30 epochs, for the first 8, we train all FPN and subsequent layers, for the next 15, we train FPN + last 4 layers of ResNet-101 and for the last 7 epochs, we train all the layers of the model. For the second step, we train a total of 10 epochs, for the first 3, we train all FPN and subsequent layers, for the next 4, we train FPN + last 4 layers of ResNet-101 and for the last 3 epochs, we train all the layers of the model. 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.

3.4 Evaluation Measures

Details of Evaluation Criteria: For comparison against most of the existing methods, we use the precision, recall and F1 score [14,18,28]. Before evaluating performance of structure recognition, it is important to understand the specific cases in which detected cells are taken into consideration for structure recognition:

- We consider a detected cell to be a true positive if it overlaps with the ground truth cell bounding box is more than 0.5.
- During evaluation of structure recognition, cells that have no content (i.e., empty or blank cells) are discarded. It means that adjacency relations that involve a blank cell are not taken into consideration.

To evaluate the performance of structure recognition, adjacency relations between every cell (with content) are generated with their horizontal and vertical neighbors. This predicted relation list is then compared with the ground truth list to generate precision, recall and F1 measures.

As per [18], this method accounts for the standard evaluation measures for table structure recognition for the following reasons:

- It provides for a simple way to account for errors in the scenarios of complex table layouts containing blank cells, and cells that span multiple rows and/or columns.
- It accounts for evaluation of both physical as well as logical structure prediction methods as it is not dependent on the bounding box coordinates information.

Further, we present our results on both micro-averaged and macro-averaged precision, recall and F1 scores which are defined as following:

- **Micro-averaged:** In this case, the confusion matrix parameters are gathered across all the data points collectively in the test dataset to compute precision, recall and F1 scores.
- **Macro-averaged:** In this case, the confusion matrix parameters are gathered individually and then averaged across all the documents in the test dataset to compute precision, recall and F1 scores.

3.5 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]).

For the unification of fair comparison for table structure recognition, we divide all inconsistencies into several levels - (i) inconsistency with respect to input modalities, (ii) inconsistency with respect to annotation levels, (iii) inconsistency with respect to representation of table structure, (iv) inconsistency with respect to evaluation methods, and (v) inconsistency with respect to way of computing evaluation scores.

Inconsistency with respect to input modalities: Section 3.2 describes that many methods for table structure recognition work with table images alone [7,10,15], several other assume additional information in the form of meta-features or bounding boxes around every word or cell-content extracted from the PDFs [9,12,14]. This makes it difficult to compare these methods under a unified scenario. To take care of this problem, we define two different experimental setups - (a) **Setup-A (S-A)** where only table image is used as an input to the structure recognition model and (b) **Setup-B (S-B)** where table image along with additional meta-features such as low-level content bounding boxes are used as an input to the structure recognition model. We present our results under both the experimental setups for a thorough comparison of our work against most of the recent methods in this space. To achieve this, we train our model for cell detection as well as structure recognition collectively for S-A. For evaluation in S-B, instead of predicting cell bounding boxes from the image, we use the table image and the low-level bounding box information from OCR or ground truth to be able to directly and fairly compare our method.

Inconsistency with respect to annotation levels: It is important to note that training of TabStruct-Net assumes cell-level bounding boxes in a way that all cells that (a) have the same SR indices having same $y1$ coordinates, (b) have the same SC indices having same $x1$ coordinates, (c) have the same ER indices having same $y2$ coordinates, and (d) have the same EC indices having same $x2$ coordinates. This assumption is necessary for our alignment loss function to work properly. However, different datasets for physical table structure recognition have ground truth annotations defined in different ways. UNLV and ICDAR-2019 archival datasets have ground truth annotated at the cell-level. SciTSR[14] and ICDAR-2013 [18] datasets have ground truth annotation defined at the content-level (cells' bounding box is the smallest rectangle that covers entire content of the cell). To be able to use those for training, we pre-process the ground-truth to obtain cell-level bounding boxes (as explained in Section 3.1). Please note that this pre-processing step is only done for the training process. Similarly, ground-truth bounding boxes of the synthetic dataset proposed in [9] are provided at the word-level. To obtain cell-level bounding boxes, we use the ground-truth cell adjacency matrix and word-level bounding boxes to obtain content-level bounding boxes. During the testing time in S-A, however, to compute if a detected cell is a true positive, we use the original ground-truth bounding boxes (either at cell-level or content-level), and not the pre-processed ones. Similarly while testing in S-B, we use the original content-level or cell-level bounding boxes as the additional input instead of the pre-processed ones. This ensures consistency while comparing our methods against previously published ones.

Inconsistency with respect to representation of table structure: We broadly classify table structure methods into two categories - (a) physical structure predicting methods that predict cell-level bounding boxes along with their associations [7,10,15] and (b) logical structure predicting methods [9,11,12,14] that predict only cell associations. In our work, we standardize our representation as described in Section 3.5, which allows us to directly compare methods in both the experimental setups. To compare the results of TabStruct-Net on logical structure prediction, we generate the mark-up string from the post-processed XML output of TabStruct-Net in the same format as TableBank [11] and PubTabNet [13] by extracting only the structure without cells' coordinates and content.

Inconsistency with respect to evaluation methods: While most of the previously published methods for table structure recognition use precision, recall and F1 scores as described in [18], there are some inconsistencies in this aspect as well. In [9], authors use true positive rate (TPR), false positive rate (FPR) and absolute accuracy on the predicted adjacency matrix to compute performance. In order to standardize evaluation with [9], we infer neighboring cell relations from their output to ensure consistency. Further, [11] use BLEU scores to compare their output with the ground truth. Since our method generates and XML output from the model's predictions, we bring our output to the same format as [11] to ensure a direct and fair comparison on the TableBank dataset [11].

Inconsistency with respect to way of computing evaluation scores: To fairly compare TabStruct-Net against previous methods, we list both micro as well as macro (document) - averaged results on all the test datasets. However, it is important to note that for TableBank [11] and PubTabNet [13] datasets, where we evaluate our results on the markup output of the model, we only consider macro(document)-averaged results.

4 Results on Table Structure Recognition

4.1 Micro-averaged Results

Tables 1-4 show the micro-averaged results of various methods for structure recognition on multiple datasets. From the tables, it can be observed that our method outperforms previously published works under multiple kinds of experimental settings. Further, it is important to note that the tables use an IoU threshold of 0.5 to identify true positive cells for experiment setup S-A. We also show the precision, recall and F1 measures on various IoU thresholds to better interpret the performance of the cell detection module of TabStruct-Net.

Method	Training		Exp.	P↑	R↑	F1↑
	Dataset	#Images	Setup			
DeepDeSRT [7]	ICDAR-2013-partial	0.124K	S-A	0.959	0.874	0.914
SPLERGE [10]	ICDAR-2013-partial	0.124K	S-A	0.917	0.911	0.914
Bi-directional GRU [15]	ICDAR-2013-partial	0.124K	S-A	**0.969**	**0.901**	**0.934**
TabStruct-Net (our)	ICDAR-2013-partial	0.124K	S-A	0.928	0.903	0.915
TabStruct-Net (our)	SciTSR + ICDAR-2013-partial	12.124K	S-A	0.930	0.908	0.919
TableNet [12]	Marmot extended	1.016K	S-B	0.931	0.900	0.915
GraphTSR [14]	SciTSR + ICDAR-2013-partial	12.124K	S-B	0.854	0.891	0.872
DGCNN [9]	SciTSR + ICDAR-2013-partial	12.124K	S-B	0.986	0.990	0.988
TabStruct-Net (our)	ICDAR-2013-partial	0.124K	S-B	**0.991**	0.989	0.990
TabStruct-Net (our)	SciTSR + ICDAR-2013-partial	12.124K	S-B	**0.991**	**0.993**	**0.992**

Table 1. Comparison of results for physical structure recognition on ICDAR-2013-partial dataset. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score and **#Images:** indicates number of table images in the training set.

Method	Training		Exp.	P↑	R↑	F1↑
	Dataset	#Images	Setup			
NLPR-PAL [19]	cTDaR	0.6K	S-A	0.720	0.770	0.745
DGCNN [9]	cTDaR	0.6K	S-A	0.785	0.751	0.768
DGCNN [9]	SciTSR	12.0K	S-A	0.552	0.519	0.535
DGCNN [9]	cTDaR + SciTSR	12.6K	S-A	0.803	0.778	0.790
SPLERGE [10]	cTDaR	0.6K	S-A	0.774	0.783	0.778
SPLERGE [10]	SciTSR	12.0K	S-A	0.559	0.572	0.565
SPLERGE [10]	cTDaR + SciTSR	12.6K	S-A	0.792	0.800	0.796
TabStruct-Net (our)	cTDaR	0.6K	S-A	0.803	0.768	0.785
TabStruct-Net (our)	SciTSR	12.0K	S-A	0.595	0.572	0.583
TabStruct-Net (our)	cTDaR + SciTSR	12.6K	S-A	**0.822**	**0.787**	**0.804**

Table 2. Comparison of results for physical structure recognition on ICDAR-2019 (cTDaR) archival dataset. For comparison against DGCNN[9], we use the cell bounding boxes detected from TabStruct-Net for a fair comparison. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score and **#Images:** indicates number of table images in the training set.

Method	Exp. Setup	P↑	R↑	F1↑
DeepDeSRT [7]	S-A	0.554	0.529	0.541
SPLERGE [10]	S-A	0.795	0.776	0.785
TabStruct-Net (our)	S-A	**0.849**	**0.828**	**0.839**
GraphTSR [14]	S-B	0.763	0.786	0.774
DGCNN[9]	S-B	0.921	0.898	0.909
TabStruct-Net (our)	S-B	**0.992**	**0.994**	**0.993**

Table 3. Comparison of results for physical structure recognition on UNLV-partial dataset. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score. All models are trained on SciTSR and fine-tuned on UNLV-partial datasets.

Method	Exp.	Evaluation on					
	Setup	SciTSR			SciTSR-COMP		
		P↑	R↑	F1↑	P↑	R↑	F1↑
DeepDeSRT [7]	S-A	0.906	0.887	0.890	0.863	0.831	0.846
SPLERGE [10]	S-A	0.922	**0.915**	0.918	**0.911**	0.880	**0.895**
TabStruct-Net (our)	S-A	**0.927**	0.913	**0.920**	0.909	**0.882**	**0.895**
GraphTSR [14]	S-B	0.959	0.948	0.953	0.964	0.945	0.955
DGCNN [9]	S-B	0.970	0.981	0.976	0.963	0.974	0.969
TabStruct-Net (our)	S-B	**0.989**	**0.993**	**0.991**	**0.981**	**0.987**	**0.984**

Table 4. Comparison of results for physical structure recognition on SciTSR and SciTSR-COMP datasets. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score. All the models are trained on SciTSR dataset.

4.2 Average Results on Markup Output

Tables 5-6 present compare our results for logical structure prediction from the table image on TableBank and PubTabNet dataset, respectively. The scores are obtained by averaging the score for every table across all the tables in the evaluation dataset. From the tables, it can be inferred that despite trained with a much smaller set of data, our model achieves better performance than [13]. Direct comparison, however would not be fair because of the use of different input modalities for training.

Method	**Training Set**			**Experimental**	**BLEU↑**		
	Dataset	**Type**	**#Images**	**Setup**	**Word**	**Latex**	**Both**
Image-to-Text [11]	TableBank	Word	55.866K	S-A	0.751	0.673	0.7138
Image-to-Text [11]	TableBank	Latex	87.597K	S-A	0.405	0.765	0.582
Image-to-Text [11]	TableBank	Both	144.493K	S-A	0.712	0.765	0.738
TabStruct-Net (our)	SciTSR	Image	12K	S-A	**0.914**	**0.937**	**0.916**

Table 5. Comparison of results for logical structure recognition on TableBank dataset.

Method	**Experimental Setup**	**Training Dataset**	**#Images**	**TEDS↑**
Acrobat Pro [13]	S-A	-	-	0.537
WYGIWYS [13]	S-A	PubTabNet	399K	0.786
EDD [13]	S-A	PubTabNet	399K	0.883
TabStruct-Net (our)	S-A	SciTSR [14]	12K	**0.901**

Table 6. Comparison of results for logical structure recognition on PubTabNet dataset [13]. **TEDS:** indicates averaged tree edit distance based similarity [13].

We present our results on the output XML file that contains — (a) bounding box coordinates, (b) start and end row indices, (c) start and end column indices, and (d) content for every predicted cell given the table image. To evaluate our method, we compare this XML against the ground-truth prepared in the same format using BLEU, CIDRr and ROUGE scores as presented in Table 7. The table also compares our results against DGCNN [**?**] when cells detected from TabStruct-Net are provided as the input to their model.

Training Set	Evaluation Set	Model	BLEU↑	CIDEr↑	ROUGE↑
SciTSR	SciTSR	DGCNN	0.774	0.8	0.782
		TabStruct-net	0.833	0.848	0.839
SciTSR	SciTSR-COMP	DGCNN	0.769	0.795	0.774
		TabStruct-net	0.826	0.837	0.830
SciTSR + UNLV-partial	UNLV-partial	DGCNN	0.721	0.744	0.729
		TabStruct-net	0.804	0.826	0.813
SciTSR	ICDAR-2013	DGCNN	0.756	0.773	0.762
		TabStruct-net	0.815	0.831	0.821
SciTSR + ICDAR-2013-partial	ICDAR-2013-partial	DGCNN	0.772	0.801	0.78
		TabStruct-net	0.829	0.845	0.834

Table 7. Results comparison of various methods for table structure recognition on various datasets.

4.3 Macro-Averaged Results

Tables 8-12 show the macro-averaged results of various methods for structure recognition on multiple datasets. From the tables, it can be observed that our method outperforms previously published works under multiple kinds of experimental settings. Further, it is important to note that the tables use an IoU threshold of 0.5 to identify true positive cells for experiment setup S-A. It can be inferred from the tables that the macro-averaged numbers follow the same trend as the micro-averaged results.

Method	Training		Experimental	P↑	R↑	F1↑
	Dataset	#Images	Setup			
DeepDeSRT [7]	SciTSR	12K	S-A	0.603	0.591	0.597
SPLERGE [10]	Private [10]	83K	S-A	**0.914**	**0.897**	**0.905**
TabStruct-Net (our)	SciTSR	12K	S-A	0.883	0.871	0.877
TableNet [12]	Marmot Extended	1K	S-B	-	-	-
GraphTSR [14]	SciTSR	12K	S-B	0.819	0.855	0.837
SPLERGE [10]	Private [10]	83K	S-B	0.932	0.917	0.924
DGCNN [9]	SciTSR	12K	S-B	0.959	0.971	0.965
TabStruct-Net (our)	SciTSR	12K	S-B	**0.961**	**0.973**	**0.967**

Table 8. Comparison of results for physical structure recognition on ICDAR-2013 dataset. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score and **#Images:** indicates number of table images in the training set. SPLERGE [10] is the **best performing model** (with post-processing) on ICDAR-2013 dataset in **S-A**. TabStruct-Net is the **best performing model** on ICDAR-2013 dataset in **S-B**.

Method	Training		Exp.	P↑	R↑	F1↑
	Dataset	#Images	Setup			
DeepDeSRT [7]	ICDAR-2013-partial	0.124K	S-A	0.853	0.795	0.823
Bi-directional GRU [15]	ICDAR-2013-partial	0.124K	S-A	-	-	-
TabStruct-Net (our)	ICDAR-2013-partial	0.124K	S-A	0.907	0.889	0.898
TabStruct-Net (our)	SciTSR + ICDAR-2013-partial	12.124K	S-A	**0.922**	**0.896**	**0.909**
TableNet [12]	Marmot extended	1.016K	S-B	-	-	-
GraphTSR [14]	SciTSR + ICDAR-2013-partial	12.124K	S-B	0.846	0.879	0.862
DGCNN [9]	SciTSR + ICDAR-2013-partial	12.124K	S-B	0.978	0.984	0.981
TabStruct-Net (our)	ICDAR-2013-partial	0.124K	S-B	**0.985**	0.986	0.985
TabStruct-Net (our)	SciTSR + ICDAR-2013-partial	12.124K	S-B	**0.985**	**0.989**	**0.987**

Table 9. Comparison of results for physical structure recognition on ICDAR-2013-partial dataset. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score and **#Images:** indicates number of table images in the training set. TabStruct-Net is the **best performing model** on ICDAR-2013-partial dataset in both **S-A** and **S-B**.

Method	Training		Exp.	P↑	R↑	F1↑
	Dataset	#Images	Setup			
NLPR-PAL [19]	cTDaR	0.6K	S-A	-	-	-
DGCNN [9]	cTDaR	0.6K	S-A	0.704	0.649	0.675
DGCNN [9]	SciTSR	12.0K	S-A	0.427	0.395	0.410
DGCNN [9]	cTDaR + SciTSR	12.6K	S-A	0.728	0.672	0.699
TabStruct-Net (our)	cTDaR	0.6K	S-A	0.729	0.667	0.697
TabStruct-Net (our)	SciTSR	12.0K	S-A	0.483	0.458	0.470
TabStruct-Net (our)	cTDaR + SciTSR	12.6K	S-A	**0.754**	**0.691**	**0.721**

Table 10. Comparison of results for physical structure recognition on ICDAR-2019 (cTDaR) archival dataset. For comparison against DGCNN[9], we use the cell bounding boxes detected from TabStruct-Net for a fair comparison. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score and **#Images:** indicates number of table images in the training set. TabStruct-Net is the **best performing model** on ICDAR-2019 cTDaR archival dataset in both **S-A** and **S-B**.

Method	**Exp. Setup**	**P↑**	**R↑**	**F1↑**
DeepDeSRT [7]	S-A	0.469	0.426	0.446
SPLERGE [10]	S-A	0.748	0.709	0.728
TabStruct-Net (our)	S-A	**0.803**	**0.775**	**0.788**
GraphTSR [14]	S-B	0.727	0.744	0.735
DGCNN* [9]	S-B	0.887	0.858	0.872
TabStruct-Net (our)	S-B	**0.975**	**0.981**	**0.978**

Table 11. Comparison of results for physical structure recognition on UNLV-partial dataset. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score. All models are trained on SciTSR and fine-tuned on UNLV-partial datasets. TabStruct-Net is the **best performing model** on UNLV-partial dataset in both **S-A** and **S-B**.

Method	**Exp. Setup**	**Evaluation on**					
		SciTSR			**SciTSR-COMP**		
		P↑	**R↑**	**F1↑**	**P↑**	**R↑**	**F1↑**
DeepDeSRT [7]	S-A	0.872	0.854	0.863	0.806	0.779	0.792
SPLERGE [10]	S-A	0.907	0.892	0.899	0.884	0.857	0.870
TabStruct-Net (our)	S-A	**0.910**	**0.897**	**0.903**	**0.889**	**0.864**	**0.876**
GraphTSR [14]	S-B	0.936	0.931	0.934	0.943	0.925	0.934
DGCNN* [9]	S-B	0.956	0.965	0.960	0.947	0.955	0.951
TabStruct-Net (our)	S-B	**0.963**	**0.977**	**0.970**	**0.955**	**0.969**	**0.962**

Table 12. Comparison of results for physical structure recognition on SciTSR and SciTSR-COMP datasets. **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score. All the models are trained on SciTSR dataset. TabStruct-Net is the **best performing model** on SciTSR and SciTSR-COMP datasets in both **S-A** and **S-B**.

4.4 Qualitative Results of Cell Detection

Figures 2-3 demonstrate some qualitative results of cell detection on all the evaluation datasets. From the figures, it can be seen that our model is able to work in the presence of archival table images, multiple row/column spanning cells, varied table layouts and multiple line spanning cells. This indicates the robustness of our method under multiple kind of table images.

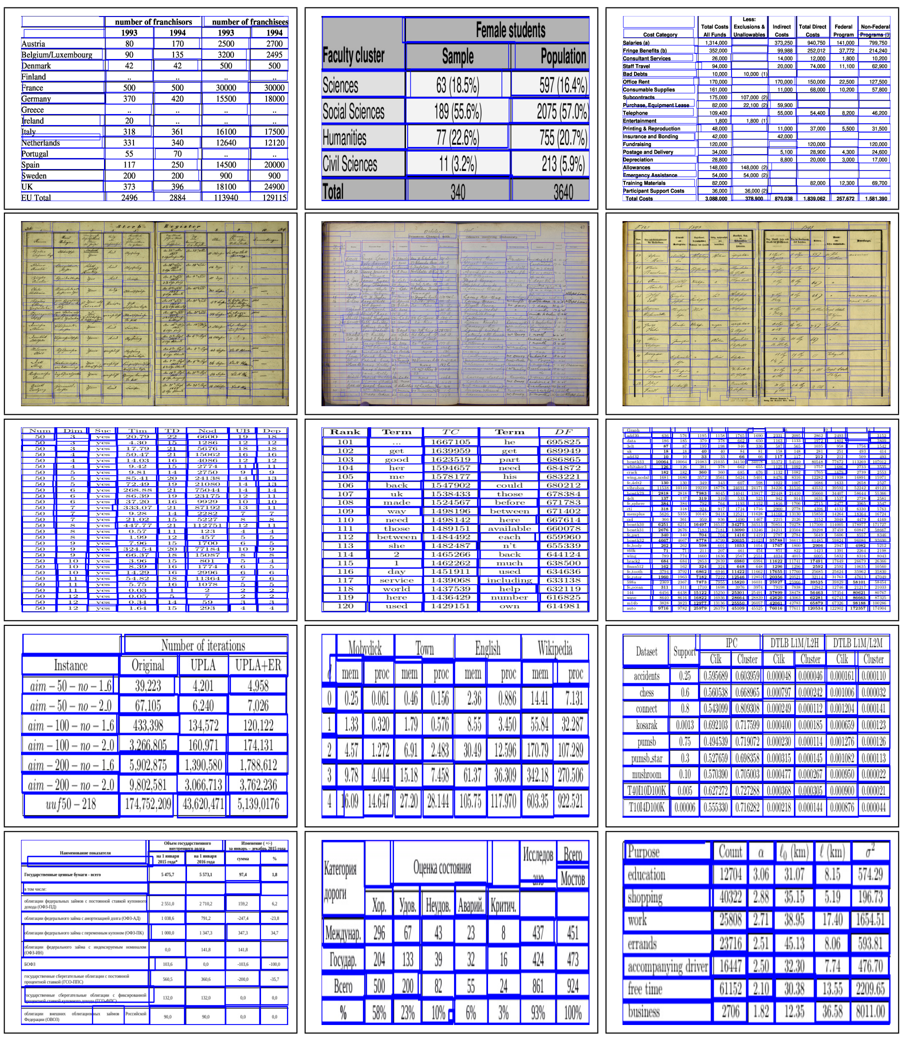

Fig. 2. Sample intermediate cell detection results of TabStruct-Net on table images of ICDAR-2013 (in **First Row**), ICDAR-2019 (in **Second Row**), SciTSR (in **Third Row**), SciTSR-COMP (in **Fourth Row**) and TableBank (in **Fifth Row**) datasets.

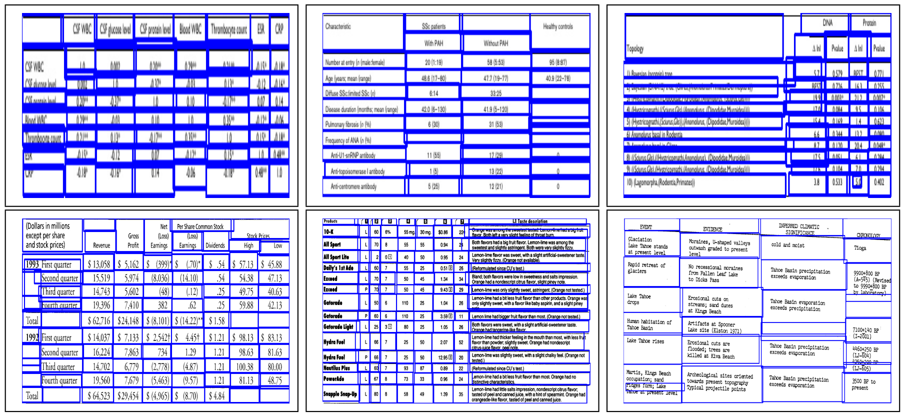

Fig. 3. Sample intermediate cell detection results of TabStruct-Net on table images of PubTabNet (in **First Row**) and UNLV (in **Second Row**) datasets.

4.5 Qualitative Results of Structure Recognition

Figures 4-10 demonstrate some qualitative results of structure recognition on all the evaluation datasets. From the figures, it can be seen that our model is able to work in the presence of archival table images, multi row/column spanning cells, varied table layouts and multiple line spanning cells. This indicates the robustness of our method under multiple kind of table images.

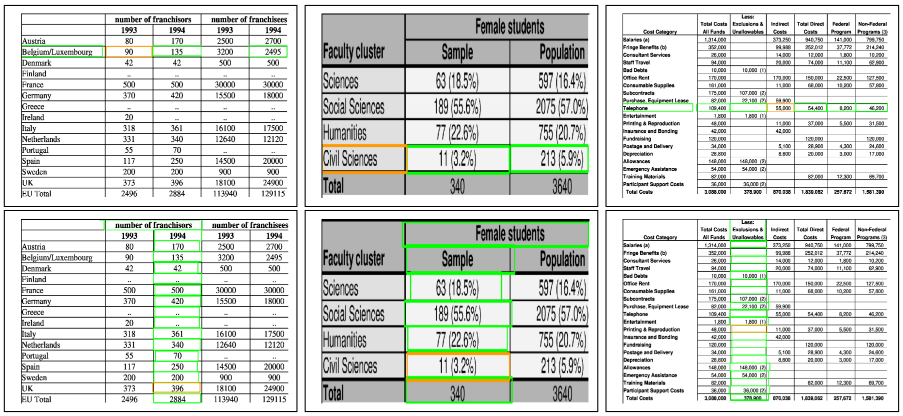

Fig. 4. Sample structure recognition output of TabStruct-Net on table images of ICDAR-2013 dataset. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

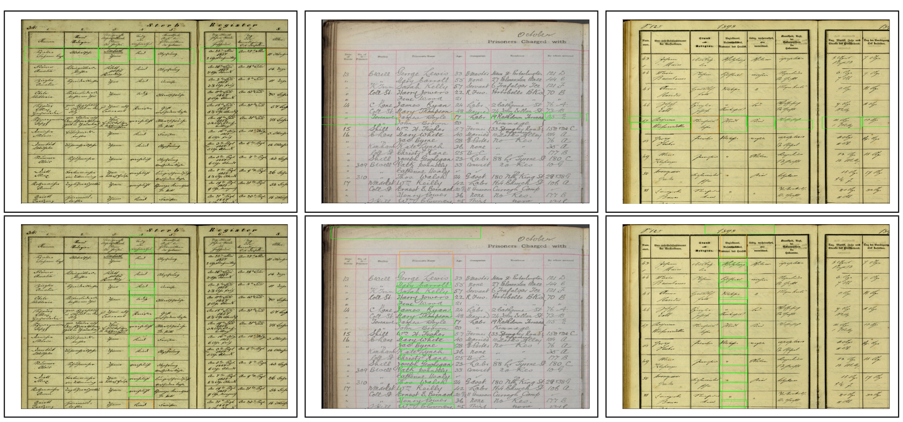

Fig. 5. Sample structure recognition output of TabStruct-Net on table images of ICDAR-2019 dataset. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

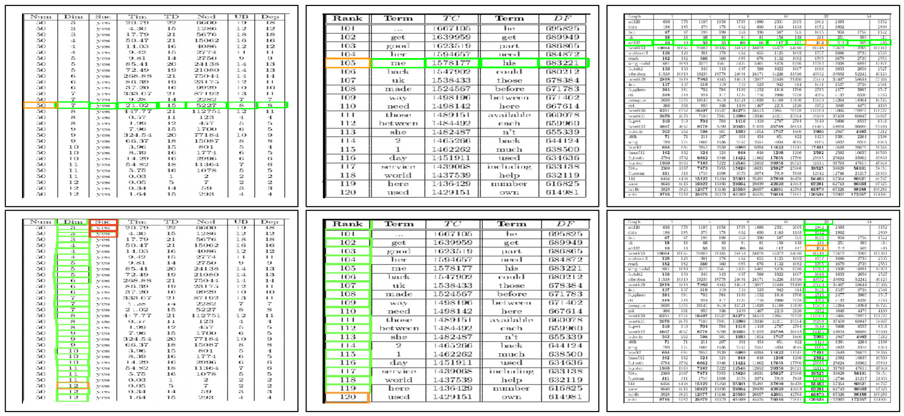

Fig. 6. Sample structure recognition output of TabStruct-Net on table images of SciTSR dataset. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

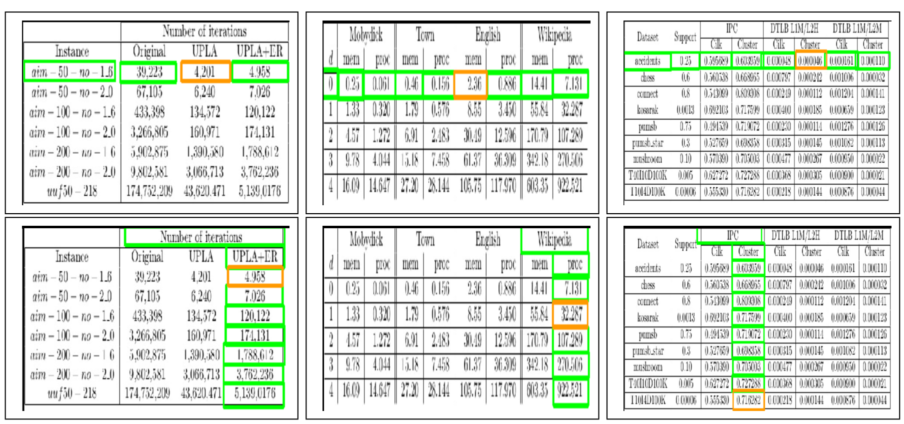

Fig. 7. Sample structure recognition output of TabStruct-Net on table images of SciTSR-COMP dataset. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

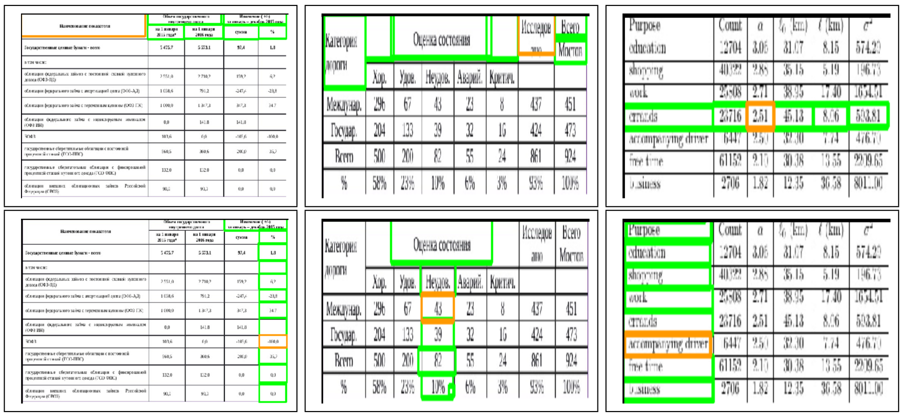

Fig. 8. Sample structure recognition output of TabStruct-Net on table images of TableBank dataset. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

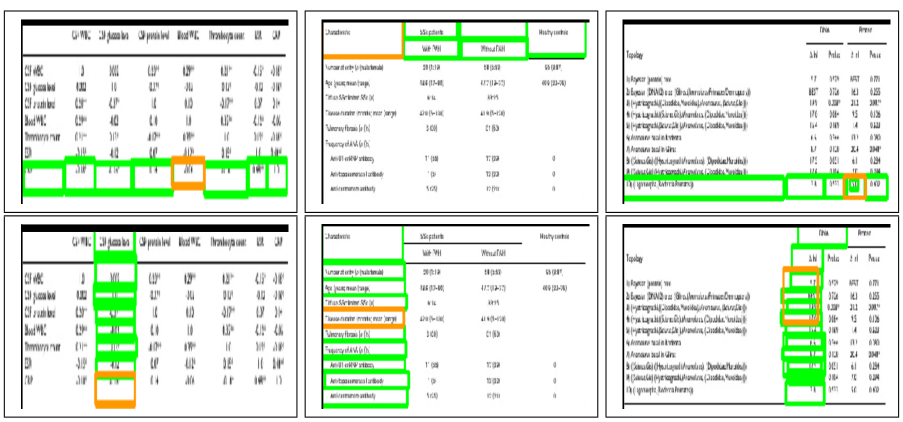

Fig. 9. Sample structure recognition output of TabStruct-Net on table images of PubTabNet dataset. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

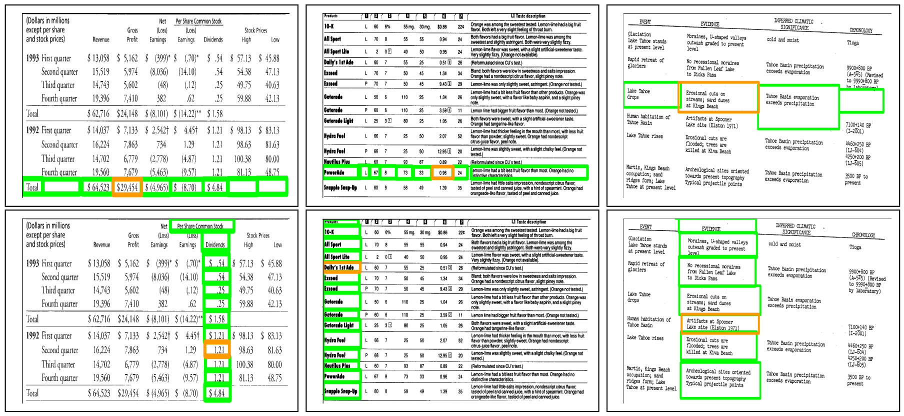

Fig. 10. Sample structure recognition output of TabStruct-Net on table images of UNLV dataset. **First Row:** prediction of cells which belong to the same row. **Second Row:** prediction of cells which belong to the same column. Cells marked with orange colour represent the examine cells and cells marked with green colour represent those which belong to the same row/column of the examined cell.

4.6 Failure Examples

Figure 11 shows some failure cases of our model in presence of empty spaces along both horizontal and vertical axes.

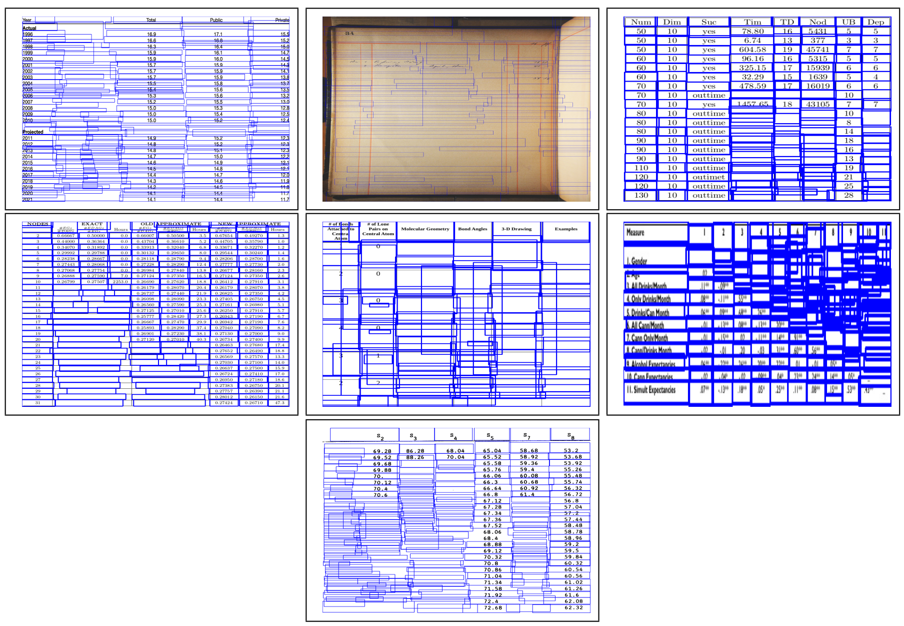

Fig. 11. Sample intermediate cell detection results of TabStruct-Net on table images of ICDAR-2013, ICDAR-2019 cTDaR, SciTSR, SciTSR-COMP, TableBank, PubTabNet and UNLV datasets illustrate failure of TabStruct-Net.

4.7 Robustness of TabStruct-Net

CD Network	SR Network	IoU TH	CD Scores P↑	CD Scores R↑	CD Scores F1↑	SR Scores P↑	SR Scores R↑	SR Scores F1↑
Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	0.5	**0.942**	**0.948**	**0.945**	**0.933**	**0.915**	**0.924**
		0.6	0.937	0.941	0.939	0.930	0.908	0.919
		0.7	0.828	0.831	0.829	0.800	0.791	0.795
		0.8	0.651	0.670	0.660	0.638	0.624	0.631
		0.9	0.314	0.336	0.325	0.291	0.284	0.287

Table 13. Physical structure recognition results on ICDAR-2013-partial dataset for varying IoU thresholds to demonstrate TabStruct-Net's robustness. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **TH:** indicates IoU threshold value, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.

CD Network	SR Network	IoU	CD Scores			SR Scores		
		TH	P↑	R↑	F1↑	P↑	R↑	F1↑
Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	0.5	**0.865**	**0.857**	**0.861**	**0.864**	**0.842**	**0.853**
		0.6	0.84	0.836	0.838	0.822	0.787	0.804
		0.7	0.694	0.681	0.687	0.641	0.625	0.633
		0.8	0.454	0.428	0.441	0.404	0.376	0.389
		0.9	0.201	0.153	0.174	0.175	0.138	0.154

Table 14. Physical structure recognition results on ICDAR-2019 dataset for varying IoU thresholds to demonstrate TabStruct-Net's robustness. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **TH:** indicates IoU threshold value, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.

CD Network	SR Network	IoU	CD Scores			SR Scores		
		TH	P↑	R↑	F1↑	P↑	R↑	F1↑
Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	0.5	**0.871**	**0.879**	**0.875**	**0.864**	**0.842**	**0.853**
		0.6	0.858	0.864	0.861	0.849	0.828	0.839
		0.7	0.751	0.773	0.762	0.735	0.711	0.723
		0.8	0.595	0.622	0.608	0.558	0.532	0.545
		0.9	0.214	0.237	0.225	0.173	0.148	0.160

Table 15. Physical structure recognition results on UNLV-partial dataset for varying IoU thresholds to demonstrate TabStruct-Net's robustness. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **TH:** indicates IoU threshold value, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.

CD Network	SR Network	IoU TH	CD Scores P↑	CD Scores R↑	CD Scores F1↑	SR Scores P↑	SR Scores R↑	SR Scores F1↑
Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	0.5	**0.939**	**0.944**	**0.941**	**0.930**	**0.922**	**0.926**
		0.6	0.932	0.938	0.935	0.927	0.913	0.920
		0.7	0.808	0.820	0.814	0.793	0.775	0.784
		0.8	0.639	0.652	0.645	0.618	0.594	0.606
		0.9	0.297	0.324	0.310	0.271	0.258	0.264

Table 16. Physical structure recognition results on SciTSR dataset for varying IoU thresholds to demonstrate TabStruct-Net's robustness. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **TH:** indicates IoU threshold value, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.

4.8 Ablation Study

ES	CD Network	SR Network	CD Scores			SR Scores		
			P↑	R↑	F1↑	P↑	R↑	F1↑
S-A	Mask R-CNN	DGCNN	0.835	0.843	0.839	0.885	0.864	0.874
	Mask R-CNN	DGCNN+P2	0.837	0.846	0.841	0.887	0.865	0.876
	Mask R-CNN	DGCNN+P2+LSTM	0.840	0.848	0.844	0.903	0.889	0.896
	Mask R-CNN+TD+BU	DGCNN	0.898	0.900	0.899	0.889	0.882	0.885
	Mask R-CNN+TD+BU	DGCNN+P2	0.902	0.905	0.903	0.893	0.885	0.889
	Mask R-CNN+TD+BU	DGCNN+P2+LSTM	0.924	0.928	0.926	0.911	0.896	0.903
	Mask R-CNN+TD+BU+AL	DGCNN	0.920	0.924	0.922	0.915	0.892	0.903
	Mask R-CNN+TD+BU+AL	DGCNN+P2	0.924	0.927	0.925	0.918	0.894	0.906
	Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	**0.937**	**0.941**	**0.939**	**0.930**	**0.908**	**0.919**
S-B	-NA-	DGCNN	-NA-	-NA-	-NA-	0.986	0.990	0.988
	-NA-	DGCNN+P2	-NA-	-NA-	-NA-	0.987	0.990	0.989
	-NA-	DGCNN+P2+LSTM	-NA-	-NA-	-NA-	**0.991**	**0.993**	**0.992**

Table 17. Ablation study for physical structure recognition on ICDAR-2013-partial dataset. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.

ES	CD Network	SR Network	CD Scores			SR Scores		
			P↑	R↑	F1↑	P↑	R↑	F1↑
S-A	Mask R-CNN	DGCNN	0.770	0.752	0.761	0.744	0.706	0.725
	Mask R-CNN	DGCNN+P2	0.774	0.761	0.767	0.751	0.718	0.734
	Mask R-CNN	DGCNN+P2+LSTM	0.797	0.785	0.791	0.775	0.750	0.762
	Mask R-CNN+TD+BU	DGCNN	0.775	0.761	0.768	0.751	0.713	0.732
	Mask R-CNN+TD+BU	DGCNN+P2	0.781	0.768	0.774	0.756	0.721	0.738
	Mask R-CNN+TD+BU	DGCNN+P2+LSTM	0.803	0.790	0.796	0.782	0.754	0.768
	Mask R-CNN+TD+BU+AL	DGCNN	0.821	0.814	0.817	0.797	0.748	0.772
	Mask R-CNN+TD+BU+AL	DGCNN+P2	0.823	0.818	0.820	0.800	0.753	0.776
	Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	**0.840**	**0.836**	**0.838**	**0.822**	**0.787**	**0.804**
S-B	-NA-	DGCNN	-NA-	-NA-	-NA-	0.904	0.889	0.896
	-NA-	DGCNN+P2	-NA-	-NA-	-NA-	0.932	0.921	0.927
	-NA-	DGCNN+P2+LSTM	-NA-	-NA-	-NA-	**0.975**	**0.958**	**0.966**

Table 18. Ablation study for physical structure recognition on ICDAR-2019 dataset. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.

ES	CD Network	SR Network	CD Scores			SR Scores		
			P↑	R↑	F1↑	P↑	R↑	F1↑
S-A	Mask R-CNN	DGCNN	0.835	0.843	0.839	0.795	0.764	0.779
	Mask R-CNN	DGCNN+P2	0.837	0.846	0.841	0.812	0.788	0.800
	Mask R-CNN	DGCNN+P2+LSTM	0.840	0.848	0.844	0.838	0.821	0.829
	Mask R-CNN+TD+BU	DGCNN	0.837	0.845	0.841	0.797	0.766	0.781
	Mask R-CNN+TD+BU	DGCNN+P2	0.840	0.849	0.844	0.815	0.790	0.802
	Mask R-CNN+TD+BU	DGCNN+P2+LSTM	0.844	0.851	0.847	0.841	0.823	0.832
	Mask R-CNN+TD+BU+AL	DGCNN	0.847	0.855	0.851	0.802	0.775	0.788
	Mask R-CNN+TD+BU+AL	DGCNN+P2	0.853	0.860	0.856	0.823	0.797	0.810
	Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	**0.858**	**0.864**	**0.861**	**0.849**	**0.828**	**0.839**
S-B	-NA-	DGCNN	-NA-	-NA-	-NA-	0.921	0.898	0.909
	-NA-	DGCNN+P2	-NA-	-NA-	-NA-	0.950	0.935	0.942
	-NA-	DGCNN+P2+LSTM	-NA-	-NA-	-NA-	**0.992**	**0.994**	**0.993**

Table 19. Ablation study for physical structure recognition on UNLV-partial dataset. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.

ES	CD Network	SR Network	CD Scores			SR Scores		
			P↑	R↑	F1↑	P↑	R↑	F1↑
S-A	Mask R-CNN	DGCNN	0.896	0.900	0.898	0.888	0.874	0.881
	Mask R-CNN	DGCNN+P2	0.904	0.907	0.905	0.892	0.879	0.885
	Mask R-CNN	DGCNN+P2+LSTM	0.911	0.915	0.913	0.903	0.894	0.898
	Mask R-CNN+TD+BU	DGCNN	0.901	0.909	0.905	0.893	0.880	0.886
	Mask R-CNN+TD+BU	DGCNN+P2	0.905	0.917	0.911	0.896	0.882	0.889
	Mask R-CNN+TD+BU	DGCNN+P2+LSTM	0.918	0.924	0.921	0.905	0.898	0.902
	Mask R-CNN+TD+BU+AL	DGCNN	0.908	0.919	0.913	0.908	0.894	0.901
	Mask R-CNN+TD+BU+AL	DGCNN+P2	0.921	0.926	0.923	0.913	0.901	0.907
	Mask R-CNN+TD+BU+AL	DGCNN+P2+LSTM	**0.932**	**0.938**	**0.935**	**0.927**	**0.913**	**0.920**
S-B	-NA-	DGCNN	-NA-	-NA-	-NA-	0.970	0.981	0.976
	-NA-	DGCNN+P2	-NA-	-NA-	-NA-	0.973	0.986	0.979
	-NA-	DGCNN+P2+LSTM	-NA-	-NA-	-NA-	**0.989**	**0.993**	**0.991**

Table 20. Ablation study for physical structure recognition on SciTSR dataset. **ES:** indicates Experimental Setup, **CD:** indicates Cell Detection, **SR:** indicates Structure Recognition, **P2:** indicates using visual features from P2 layer of the FPN instead of using separate convolution blocks, LSTM**:** indicates use of LSTMs to model visual features along center-horizontal and center-vertical lines for every cell, TD+BU**:** indicates use of Top-Down and Bottom-Up pathways in the FPN, **AL:** indicates addition of alignment loss as a regularizer to TabStruct-Net, **P:** indicates precision, **R:** indicates recall, **F1:** indicates F1 Score.