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 [1-2]. 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 [3-6]. 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 [7-8]. It can be represented in the form of either physical...
Fig. 1. The figure depicts the problem of recognizing table structure from its image. This opens up many applications including information retrieval, graphical representation and digitizing for editing.

Table structure recognition is a challenging problem due to complex structures and high variability in table layouts. 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 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 that need to be addressed to solve the problem using visual cues.

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...
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.

Through table detection has observed significant success \cite{11, 25, 26, 27, 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 \cite{14}, scitsr-comp \cite{13}, icdar-2013 table recognition \cite{15}, icdar-2019 (cTDar) archival \cite{19}, and unlv \cite{29}. Further, we extend the comparative analysis of the proposed work for logical structure recognition on TableBank \cite{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-
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 [6,7,8,9,10,11,12,13,14,15,16,17]. 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 [21,22,23,24,34,35,36]. 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 [37,38,9,10,11,12,13,14,15,16,37,38,39,40,41,42,43] 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 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, Scitrsr, Scitrsr-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
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:

\[
L_1 = \sum_{r \in SR} \sum_{c_i, c_j \in r} ||y_{1c_i} - y_{1c_j}||^2_2, \quad L_2 = \sum_{r \in ER} \sum_{c_i, c_j \in r} ||y_{2c_i} - y_{2c_j}||^2_2, \quad L_3 = \sum_{c \in SC} \sum_{c_i, c_j \in c} ||x_{1c_i} - x_{1c_j}||^2_2, \quad L_4 = \sum_{c \in EC} \sum_{c_i, c_j \in c} ||x_{2c_i} - x_{2c_j}||^2_2
\]

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 \(c\); \(x_{1c_i}\), \(y_{1c_i}\), \(x_{2c_i}\) and \(y_{2c_i}\) 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:

\[ L = L_{\text{box}} + L_{\text{cls}} + L_{\text{mask}} + L_{\text{align}} + L_{\text{gnn}}, \]

where \( L_{\text{box}} \), \( L_{\text{cls}} \) and \( L_{\text{mask}} \) are bounding box regression loss, classification loss and mask loss, respectively defined in Mask R-CNN \[47\], \( L_{\text{align}} \) is alignment loss which is modeled as a regularizer (defined in Eq. 1) and \( L_{\text{gnn}} \) 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.

| Dataset  | SciTsr | SciTsr-COMP 2013 | ICDAR-2013 -partial | ICDAR-2019 | UNLV partial | Marmot extended | TableBank | PubTabNet |
|----------|--------|-----------------|-------------------|-------------|-------------|----------------|-----------|-----------|
| Train    | 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 [11] 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 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

\(^1\) Our code is available at [https://github.com/sachinraja13/TabStructNet.git](https://github.com/sachinraja13/TabStructNet.git)
Table 2. shows the performance of our Tabstruct-Net for physical table structure recognition on various benchmark datasets.

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

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 Splerger with a 3% average F1 score. Splerger 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 Setup | P↑  | R↑  | F1↑ |
|-------------------------|-------------|--------------------|-----|-----|-----|
| DeepDesRT [7]           | scitsr      | 12K                | 0.631 | 0.619 | 0.625 |
| Splerge [10]            | scitsr      | 12K                | 0.883 | 0.875 | 0.879 |
| Split+Heuristic [10]    | Private [10]| 83K                | 0.938 | 0.922 | 0.930 |
| Tabstruct-Net (our)     | scitsr      | 12K                | 0.915 | 0.897 | 0.906 |
| TableNet [12]           | Marmot Extended | 1K            | 0.922 | 0.899 | 0.910 |
| GraphTSR [13]           | scitsr      | 12K                | 0.858 | 0.860 | 0.872 |
| Split-PDF [10]          | Private [10]| 83K                | 0.920 | 0.913 | 0.916 |
| Split-PDF +Heuristic [10]| Private [10]| 83K                | 0.959 | 0.946 | 0.953 |
| DGCNN [9]               | scitsr      | 12K                | 0.972 | 0.983 | 0.977 |
| Tabstruct-Net (our)     | scitsr      | 12K                | 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 [9,10,12]. 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 Threshold | CD Scores | SR Scores |
|------------------|------------|---------------|-----------|-----------|
|                  |            | TH | P↑ | R↑ | F1↑ | TH | P↑ | R↑ | F1↑ | TH | P↑ | R↑ | F1↑ |
| Mask R-CNN+TD+BU+AL | DGCNN+P2+LSTM | 0.5 | 0.9350 | 0.9420 | 0.9380 | 0.9270 | 0.9110 | 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.

### 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.
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.

**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., Spaminato, 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., Baudší, 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)
Appendix A: 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.

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 [53] 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.

### Appendix B: Experiments

#### Dataset

We use various benchmark table structure recognition datasets — scTSR [14], scTSR-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.

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. scTSR [14], scTSR-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.

#### Baseline Methods

We compare the performance of our TabStruct-Net against seven following benchmark methods.
Table Structure Recognition using Top-Down and Bottom-Up Cues

Fig. 9. 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**.

**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 [53]) 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** [12]: 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.

**Implementation Details**

Our TabStruct-Net model 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 [53] 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.

Evaluation Measures

Details of Evaluation Criteria: For comparison against most of the existing methods, we use the precision, recall and F1 score [14,18,29]. 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.

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 \( y_1 \) coordinates, (b) have the same \( SC \) indices having same \( x_1 \) coordinates, (c) have the same \( ER \) indices having same \( y_2 \) coordinates, and (d) have the same \( EC \) indices having same \( x_2 \) 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, [12] 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 [12] 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 averaged results on 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 micro averaged results.

**Appendix C: Results on Table Structure Recognition**

**Micro-averaged Results**

Tables [7,10] 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 0.6 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.
Table 7. 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        | Setup | #Images | P↑  | R↑  | F1↑  |
|-------------------------|-----------------|-------|---------|-----|-----|------|
| DeepDesRT               | ICDAR-2013-partial | 0.124K | S-A     | 0.959 | 0.874 | 0.914 |
| SPRLEGE                 | ICDAR-2013-partial | 0.124K | S-A     | 0.917 | 0.911 | 0.914 |
| Bi-directional GRU      | 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          | 12.124K | S-A     | 0.930 | 0.908 | 0.919 |
|                         | + ICDAR-2013-partial |       |         |      |      |      |
| TableNet                | Marmot extended | 1.016K | S-B     | 0.931 | 0.900 | 0.915 |
| GraphTSR                | sciTSR          | 12.124K | S-B     | 0.854 | 0.891 | 0.872 |
|                         | + ICDAR-2013-partial |       |         |      |      |      |
| DGCNN                   | sciTSR          | 12.124K | S-B     | 0.986 | 0.990 | 0.988 |
|                         | + ICDAR-2013-partial |       |         |      |      |      |
| Tabstruct-Net (our)     | ICDAR-2013-partial | 0.124K | S-B     | 0.991 | 0.989 | 0.990 |
| Tabstruct-Net (our)     | sciTSR          | 12.124K | S-B     | 0.991 | 0.993 | 0.992 |
|                         | + ICDAR-2013-partial |       |         |      |      |      |

Table 8. 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                  | Training        | Setup | #Images | P↑  | R↑  | F1↑  |
|-------------------------|-----------------|-------|---------|-----|-----|------|
| NLPR-PAL                | cTDAr           | 0.6K  | S-A     | 0.720 | 0.770 | 0.745 |
| DGCNN                   | cTDAr           | 0.6K  | S-A     | 0.785 | 0.751 | 0.768 |
| DGCNN                   | sciTSR          | 12.0K | S-A     | 0.552 | 0.519 | 0.535 |
| DGCNN                   | cTDAr + sciTSR  | 12.6K | S-A     | 0.803 | 0.778 | 0.790 |
| SPRLEGE                 | cTDAr           | 0.6K  | S-A     | 0.774 | 0.783 | 0.778 |
| SPRLEGE                 | sciTSR          | 12.0K | S-A     | 0.559 | 0.572 | 0.565 |
| SPRLEGE                 | 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 9. 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. Setup | \( P \uparrow \) | \( R \uparrow \) | \( F1 \uparrow \) |
|--------------------|------------|------------------|------------------|------------------|
| 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        | \textbf{0.849}   | \textbf{0.828}   | \textbf{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        | \textbf{0.992}   | \textbf{0.994}   | \textbf{0.993}   |

Table 10. 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.

| Method             | Exp. Setup | Evaluation on | SciTSR | SciTSR-COMP |
|--------------------|------------|---------------|--------|-------------|
|                    |            | \( P \uparrow \) | \( R \uparrow \) | \( F1 \uparrow \) | \( P \uparrow \) | \( R \uparrow \) | \( F1 \uparrow \) |
| DeepDesRT \[7\]    | S-A        | 0.906         | 0.887  | 0.890       | 0.363         | 0.831         | 0.846         |
| SPLERGE \[10\]     | S-A        | 0.922         | \textbf{0.915} | 0.918      | \textbf{0.911} | 0.880         | \textbf{0.895} |
| Tabstruct-Net (our)| S-A        | \textbf{0.927} | 0.913  | \textbf{0.920} | 0.909         | \textbf{0.882} | \textbf{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        | \textbf{0.989} | \textbf{0.993} | \textbf{0.991} | 0.981         | \textbf{0.987} | \textbf{0.984} |

Average Results on Markup Output

Tables \[11\][12] 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 Setup | BLEU \( \uparrow \) |
|--------------------|--------------|---------------------|---------------------|
|                    | Dataset Type | \#Images             | Word Latex Both      |
| Image-to-Text \[12\]| Tablebank   | 55.866K              | S-A                 | 0.751 0.673 0.7138 |
| Image-to-Text \[12\]| Tablebank   | 87.597K              | S-A                 | 0.405 0.765 0.582 |
| Image-to-Text \[12\]| Tablebank   | 144.493K             | S-A                 | 0.712 0.765 0.738 |
| Tabstruct-Net (our)| scitsr      | 12K                  | S-A                 | \textbf{0.914} \textbf{0.937} \textbf{0.916} |

Table 11. Comparison of results for logical structure recognition on TableBank dataset.

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,
| Method                        | Experimental Setup | Training Dataset | #Images | TEDS |
|-------------------------------|--------------------|------------------|---------|------|
| Acrobat Pro                  | S-A                | -                | -       | 0.537|
| wygwiws                      | S-A                | PubTabNet        | 399K    | 0.786|
| EDD                          | S-A                | PubTabNet        | 399K    | 0.883|
| Tabstruct-Net (our)          | S-A                | scitsr           | 12K     | 0.901|

Table 12. Comparison of results for logical structure recognition on PubTabNet dataset [13]. TEDS: indicates averaged tree edit distance based similarity [13].

| Training Set   | Evaluation Set | Model  | BLEU | CIDEr | ROUGE |
|----------------|----------------|--------|------|-------|-------|
| scitsr         | scitsr         | DGCNN  | 0.774| 0.8   | 0.782|
| scitsr         | scitsr-comp    | DGCNN  | 0.769| 0.795 | 0.774|
| scitsr + UNLV-partial | UNLV-partial | DGCNN  | 0.721| 0.744 | 0.729|
| scitsr         | icdar-2013     | DGCNN  | 0.756| 0.773 | 0.762|
| scitsr + icdar-2013-partial | icdar-2013-partial | DGCNN  | 0.772| 0.801 | 0.78  |

Table 13. Results comparison of various methods for table structure recognition on various datasets.

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, CIDEr and ROUGE scores as presented in Table 13. The table also compares our results against DGCNN [9] when cells detected from Tabstruct-Net are provided as the input to their model.
Table Structure Recognition using Top-Down and Bottom-Up Cues

Qualitative Results of Cell Detection

Figures 10-11 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.

![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.](image-url)
Fig. 11. Sample intermediate cell detection results of TabStruct-Net on table images of PubTabNet (in First Row) and UNLV (in Second Row) datasets.

Qualitative Results of Structure Recognition

Figures 12-18 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. 12. 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. 13. 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. 14. 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. 15. 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. 16. 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. 17. 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. 18. Sample structure recognition output of Tabstruct-Net on table images of UNL 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.

### 6.1 Failure Examples

Figure 19 shows some failure cases of our model in presence of empty spaces along both horizontal and vertical axes.
Fig. 19. Sample intermediate cell detection results of Tabstruct-Net on table images of ICDAR-2013, ICDAR-2019 CTdar, scTSR, scTSR-COMP, Tablebank, PubTabNet and UNLV datasets illustrate failure of Tabstruct-Net.
6.2 Robustness of TabStruct-Net

| CD Network          | SR Network          | IoU TH | CD Scores P↑ R↑ F1↑ | SR Scores P↑ R↑ F1↑ |
|---------------------|---------------------|--------|---------------------|---------------------|
| Mask R-CNN+TD+BU+ALDGCNN+P2+LSTM | 0.5                 | 0.942 0.9480.9450.9330.9150.924 |
|                     | 0.6                 | 0.937 0.9410.9390.9300.9080.919 |
|                     | 0.7                 | 0.828 0.8310.8290.8000.7910.795 |
|                     | 0.8                 | 0.651 0.6700.6600.6380.6240.631 |
|                     | 0.9                 | 0.314 0.3360.3250.2910.2840.287 |

Table 14. 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 |          |           |
| Mask R-CNN+TD+BU+AL   | DGCNN+p2+lstm        | 0.5| 0.865    | 0.857    |
|                       |                      | 0.6| 0.84     | 0.836    |
|                       |                      | 0.7| 0.694    | 0.681    |
|                       |                      | 0.8| 0.454    | 0.428    |
|                       |                      | 0.9| 0.201    | 0.153    |
|                       |                      |    | 0.861    | 0.860    |
|                       |                      |    | 0.840    | 0.822    |
|                       |                      |    | 0.820    | 0.787    |
|                       |                      |    | 0.810    | 0.804    |
|                       |                      |    | 0.83   | 0.641    |
|                       |                      |    | 0.625    | 0.633    |
|                       |                      |    | 0.404    | 0.376    |
|                       |                      |    | 0.389    | 0.138    |
|                       |                      |    | 0.154    | 0.154    |

Table 15. 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 |          |           |
| Mask R-CNN+TD+BU+AL   | DGCNN+p2+lstm        | 0.5| 0.871    | 0.879    |
|                       |                      | 0.6| 0.858    | 0.864    |
|                       |                      | 0.7| 0.751    | 0.773    |
|                       |                      | 0.8| 0.595    | 0.622    |
|                       |                      | 0.9| 0.214    | 0.237    |
|                       |                      |    | 0.870    | 0.875    |
|                       |                      |    | 0.860    | 0.861    |
|                       |                      |    | 0.850    | 0.849    |
|                       |                      |    | 0.830    | 0.828    |
|                       |                      |    | 0.840    | 0.829    |
|                       |                      |    | 0.860    | 0.861    |
|                       |                      |    | 0.735    | 0.735    |
|                       |                      |    | 0.711    | 0.711    |
|                       |                      |    | 0.558    | 0.558    |
|                       |                      |    | 0.532    | 0.532    |
|                       |                      |    | 0.545    | 0.545    |
|                       |                      |    | 0.173    | 0.173    |
|                       |                      |    | 0.148    | 0.148    |

Table 16. 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.
### Table 17. Physical structure recognition results on scit2r dataset for varying IoU thresholds to demonstrate Tabstruct-Net’s robustness.

| CD Network                   | SR Network               | IoU TH | CD Scores | SR Scores |
|------------------------------|--------------------------|--------|-----------|-----------|
|                              |                          | 0.5    | 0.939    | 0.940     | 0.941     | 0.930     | 0.920     | 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     |
| Mask r-cnn+TD+BU+AL          | DGCNN+P2+LSTM            | 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     |

**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.

### 6.3 Ablation Study
| ES | CD Network | SR Network | CD Scores | SR Scores |
|----|------------|------------|-----------|-----------|
|    |            |            | P↑ R↑ F1↑ | P↑ R↑ F1↑ |
|    |            |            | 0.835 0.843 0.839 | 0.885 0.864 0.874 |
|    |            |            | 0.837 0.846 0.841 | 0.887 0.865 0.876 |
|    |            |            | 0.840 0.848 0.844 | 0.903 0.889 0.896 |
|    | Mask R-CNN | DGCNN      |            |           |
| S-A| Mask R-CNN | DGCNN+P2   | 0.898 0.900 0.899 | 0.889 0.882 0.885 |
|    | Mask R-CNN | DGCNN+P2+LSTM| 0.902 0.905 0.903 | 0.893 0.885 0.889 |
|    | Mask R-CNN | DGCNN+P2+LSTM| 0.924 0.928 0.926 | 0.911 0.896 0.903 |
|    | Mask R-CNN+TD+BU | DGCNN | 0.920 0.924 0.922 | 0.915 0.892 0.903 |
|    | Mask R-CNN+TD+BU | DGCNN+P2 | 0.924 0.927 0.925 | 0.918 0.894 0.906 |
|    | Mask R-CNN+TD+BU | DGCNN+P2+LSTM | 0.9370.9410.9390.9300.9080.919 | |
|    | Mask R-CNN+TD+BU+AL | DGCNN+P2+LSTM | 0.986 0.990 0.988 | 0.987 0.990 0.989 |
| S-B|          |            | 0.9910.9930.992 | |

**Table 18.** 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↑ |
|    |            |            | 0.770 0.752 0.761 | 0.744 0.706 0.725 |
|    |            |            | 0.774 0.761 0.767 | 0.751 0.718 0.734 |
|    |            |            | 0.797 0.785 0.791 | 0.775 0.750 0.762 |
|    | Mask R-CNN | DGCNN      |            |           |
| S-A| Mask R-CNN | DGCNN+P2   | 0.775 0.761 0.768 | 0.751 0.713 0.732 |
|    | Mask R-CNN | DGCNN+P2+LSTM| 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+P2+LSTM | 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 | 0.8400.8360.8380.8220.7870.804 | |
| S-B|          |            | 0.904 0.889 0.896 | 0.932 0.921 0.927 |
|    |          |            | 0.9750.9580.966 | |

**Table 19.** 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 |
| S-A | Mask R-CNN | DGCNN+P2 | 0.837 0.846 0.841 | 0.812 0.788 0.800 |
| S-A | Mask R-CNN | DGCNN+P2+LSTM | 0.840 0.848 0.844 | 0.838 0.821 0.829 |
| S-A | Mask R-CNN+TD+BU | DGCNN | 0.837 0.845 0.841 | 0.797 0.766 0.781 |
| S-A | Mask R-CNN+TD+BU | DGCNN+P2 | 0.840 0.849 0.844 | 0.815 0.790 0.802 |
| S-A | Mask R-CNN+TD+BU | DGCNN+P2+LSTM | 0.844 0.851 0.847 | 0.841 0.823 0.832 |
| S-A | Mask R-CNN+TD+BU+AL | DGCNN | 0.847 0.855 0.851 | 0.802 0.775 0.788 |
| S-A | Mask R-CNN+TD+BU+AL | DGCNN+P2 | 0.853 0.860 0.856 | 0.823 0.797 0.810 |
| S-A | Mask R-CNN+TD+BU+AL | DGCNN+P2+LSTM | 0.85808640861084908280839 |  |
| S-B | -NA- | DGCNN | -NA- -NA- -NA- | 0.921 0.898 0.909 |
| S-B | -NA- | DGCNN+P2 | -NA- -NA- -NA- | 0.950 0.935 0.942 |
| S-B | -NA- | DGCNN+P2+LSTM | -NA- -NA- -NA- | 0.9920.9940.993 |

Table 20. 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 |
| S-A | Mask R-CNN | DGCNN+P2 | 0.904 0.907 0.905 | 0.892 0.879 0.885 |
| S-A | Mask R-CNN | DGCNN+P2+LSTM | 0.911 0.915 0.913 | 0.903 0.894 0.898 |
| S-A | Mask R-CNN+TD+BU | DGCNN | 0.901 0.909 0.905 | 0.893 0.880 0.886 |
| S-A | Mask R-CNN+TD+BU | DGCNN+P2 | 0.905 0.917 0.911 | 0.896 0.882 0.889 |
| S-A | Mask R-CNN+TD+BU | DGCNN+P2+LSTM | 0.918 0.924 0.921 | 0.905 0.898 0.902 |
| S-A | Mask R-CNN+TD+BU+AL | DGCNN | 0.908 0.919 0.913 | 0.908 0.894 0.901 |
| S-A | Mask R-CNN+TD+BU+AL | DGCNN+P2 | 0.921 0.926 0.923 | 0.913 0.901 0.907 |
| S-A | Mask R-CNN+TD+BU+AL | DGCNN+P2+LSTM | 0.9320.9380.9350.9270.9130.920 |  |
| S-B | -NA- | DGCNN | -NA- -NA- -NA- | 0.970 0.981 0.976 |
| S-B | -NA- | DGCNN+P2 | -NA- -NA- -NA- | 0.973 0.986 0.979 |
| S-B | -NA- | DGCNN+P2+LSTM | -NA- -NA- -NA- | 0.9890.9930.991 |

Table 21. 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.