Robust Table Detection and Structure Recognition from Heterogeneous Document Images

Chixiang Ma^{a,1,*}, Weihong Lin^b, Lei Sun^b, Qiang Huo^b

^aDepartment of EEIS, University of Science and Technology of China, Hefei, 230026, China ^bMicrosoft Research Asia, Beijing, 100080, China

Abstract

We introduce a new table detection and structure recognition approach named RobusTabNet to detect the boundaries of tables and reconstruct the cellular structure of each table from heterogeneous document images. For table detection, we propose to use CornerNet as a new region proposal network to generate higher quality table proposals for Faster R-CNN, which has significantly improved the localization accuracy of Faster R-CNN for table detection. Consequently, our table detection approach achieves state-of-the-art performance on three public table detection benchmarks, namely cTDaR TrackA, PubLayNet and IIIT-AR-13K, by only using a lightweight ResNet-18 backbone network. Furthermore, we propose a new split-and-merge based table structure recognition approach, in which a novel spatial CNN based separation line prediction module is proposed to split each detected table into a grid of cells, and a Grid CNN based cell merging module is applied to recover the spanning cells. As the spatial CNN module can effectively propagate contextual information across the whole table image, our table structure recognizer can robustly recognize tables with large blank spaces and geometrically distorted (even curved) tables. Thanks to these two techniques, our table structure recognizion approach achieves state-of-the-art performance on three public benchmarks, including SciTSR, PubTabNet and cTDaR TrackB2-Modern. Moreover, we have further demonstrated the advantages of our approach in recognizing tables with complex structures, large blank spaces, as well as geometrically distorted or even curved shapes on a more challenging in-house dataset.

Keywords: Table detection, Table structure recognition, Corner detection, Spatial CNN, Grid CNN, Split-and-merge

1. Introduction

Tables are a prevalent means of representing and communicating structured data, which are widely used in diverse types of documents including financial statements, scientific papers, invoices, purchasing orders, etc. With the explosive growth of the number of documents, automatic table detection and structure recognition techniques are eagerly desired to reconstruct tables from document images, which can facilitate many downstream applications, such as information retrieval [1] and question answering [2]. The aim of table detection is to detect the boundaries of tables, while the aim of table structure recognition (TSR) is to reconstruct the cellular structure of each detected table, i.e., identifying the coordinates of each cell bounding box as well as its row and column spanning information. Both table detection and structure recognition are unsolved problems due to the following challenges. First, tables in documents may have complex structures and diverse styles (erratic use of ruling lines). For example, in financial reports, some borderless tables may have complex hierarchical header structures, contain many empty or spanning cells, or have extremely large/small blank spaces between neighboring columns. Some neighboring tables may be very close to each other, making it hard to determine whether they should be merged or not. In invoices, tables may have different sizes, e.g., some line-item tables may only contain two rows and some others may span multiple pages. Second, tables cells may contain diverse contents, ranging from a single character to a set of more complex page objects such as paragraphs, tables, figures, formulas, etc. Third, some background

^{*}Corresponding author.

Email addresses: chixiangma@gmail.com (Chixiang Ma),

weihlin@microsoft.com (Weihong Lin), lsun@microsoft.com (Lei Sun), qianghuo@microsoft.com (Qiang Huo)

¹This work was done when Chixiang Ma was an intern in MMI Group, Microsoft Research Asia, Beijing, China.

objects in documents like figures, graphics, flow charts and structurally laid out texts, may have similar textures as tables, which poses another challenge for reduction of false alarms. In forms, some tables may be embedded in other more complex tabular objects (e.g., nested tables), which makes table boundaries ambiguous. Moreover, many camera-captured document images are of poor image quality, and tables contained in them may be distorted (even curved) or contain artifacts or noises, which makes table detection and structure recognition even more difficult.

In recent years, the success of deep learning in various computer vision applications has motivated researchers to explore deep neural networks for detecting tables and recognizing table structures from document images. These deep learning based table detection and structure recognition approaches have substantially outperformed traditional rule or statistical machine learning based methods in terms of both accuracy and capability [3]. Most deep learning based table detection approaches (e.g., [4-11]) treat table as a specific object and borrow various CNN-based object detection and segmentation frameworks, like Faster R-CNN [12], Mask R-CNN [13], and Cascade R-CNN [14], to solve the table detection problem. With the help of some effective techniques like more powerful backbone networks and deformable convolution operations, these CNN based table detection methods, especially CDeC-Net [11], have achieved superior performance on many public table detection benchmark datasets. Despite this, the localization accuracy of these methods is still far from satisfactory. For instance, although CDeC-Net has leveraged the Cascade R-CNN model [14] to improve table detection accuracy, its detection accuracy still drops a lot when the Intersection-over-Union (IoU) threshold is increased from 0.5 to 0.9 during evaluation (Table VIII in [11]). As the localization accuracy of table detection will significantly affect the performance of the following TSR task, more effective techniques to improve the localization accuracy of these CNN based table detection methods are still desired. For table structure recognition, deep learning based methods (e.g., [8, 15-24]) have already made great progress towards recognizing tables with complex structures and diverse styles. Recent best performing table structure recognition approaches, like TabStruct-Net [22] and LGPMA [23], typically use CNNbased object detection or segmentation models like Mask R-CNN to detect table cells first, then adopt some cell grouping/clustering algorithms to predict row/column relationships between the detected cells. Although these methods have achieved very high

accuracy on benchmark datasets like SciTSR [25] and PubTabNet [26], they still cannot be directly applied to geometrically distorted or even curved tables as they rely on an assumption that tables are axis-aligned. In some real-world application scenarios like the "Insert data from picture" feature² in Excel, document images may be captured by mobile cameras. In these camera-captured images, it is inevitable that tables are geometrically distorted. However, existing benchmark datasets haven't taken this important scenario into account as images in these datasets are either captured by scanners or converted from digital PDF files. Thus, more research is needed to find out new table structure recognition approaches robust to geometrically distorted or even curved tables.

In this paper, we propose a new table detection and structure recognition approach named RobusTabNet to overcome the abovementioned challenges. For table detection, we use CornerNet [27] as a new region proposal network for Faster R-CNN, which generates table proposals by detecting and grouping corner points. With these corner-based high quality region proposals, our approach achieves superior performance even with a very lightweight backbone network, i.e., ResNet-18 [28]. For TSR, we present a new split-and-merge based approach and propose two effective techniques to significantly improve its capability. First, we propose a novel spatial CNN [29] based separation line prediction module to split each detected table into a grid of cells. As the spatial CNN can effectively propagate contextual information across the whole table image, our separation line prediction algorithm can improve the robustness of our table structure recognizer to tables with large blank spaces and distorted or even curved shapes. Second, we propose a Grid CNN based cell merging module to recover the wrongly split cells, especially the spanning cells. In this module, the whole table is compactly represented as a grid so that a simple CNN based cell merging module can achieve higher accuracy than Relation Network or Graph Convolutional Network (GCN) based methods. With these new techniques, the proposed RobusTabNet has achieved state-of-the-art performance on both table detection (cTDaR TrackA [30], PubLayNet [31] and IIIT-AR-13K [32]) and structure recognition (SciTSR [25], PubTabNet [26] and cTDaR TrackB2-Modern [30]) public benchmarks. We have further validated the robustness of our approach to tables with complex

²https://support.microsoft.com/en-us/office/ insert-data-from-picture-3c1bb58d-2c59-4bc0-b04a-\ a671a6868fd7

structures, large blank spaces, as well as distorted or even curved shapes on a more challenging in-house dataset.

The main contributions of this paper are as follows:

- We present a new table detector by using Corner-Net as a new region proposal network for Faster R-CNN to achieve high table localization accuracy. Compared with RPN [12], the percentage of well-localized proposals (IoU>0.9) in the positive samples (IoU>0.7) from CornerNet is much higher, which contributes to better end-to-end table detection performance.
- We present a new split-and-merge based table structure recognizer, which is robust to geometrically distorted or even curved tables. To this end, a new spatial CNN based separation line prediction approach is proposed to robustly predict curvilinear separation lines from distorted or even curved tables, while a Grid CNN module is proposed to recover spanning cells efficiently and effectively.
- Our proposed table extraction approach, RobusTabNet, has achieved state-of-the-art performance on both table detection (cTDaR TrackA, PubLayNet and IIIT-AR-13K) and structure recognition (SciTSR, PubTabNet and cTDaR TrackB2-Modern) public benchmarks.

2. Related work

2.1. Table detection

2.1.1. Traditional methods

Rule-based methods are among the earliest approaches for locating tables inside documents. These methods usually exploit visual clues (e.g., text-block arrangement [33], or horizontal and vertical lines [34– 36]), keywords [37, 38], or formal templates [39] to detect tables in particular scenarios. We refer readers to [40, 41] for a more detailed summarization of these conventional approaches. Rule-based methods usually require extensive manual efforts to design heuristic rules and tune hyper-parameters. To reduce the dependence on heuristics, lots of statistical machine learning based approaches have been proposed, e.g., [42, 43]. Although these methods have improved table detection accuracy significantly, they still rely on handcrafted features, which limit their generalization ability. A comprehensive review of these statistical machine learning based methods can be found in [44].

2.1.2. Deep learning based methods

With the rapid development of deep learning, numerous CNN based table detection methods have been proposed and outperformed traditional methods by a big margin in terms of both accuracy and capability. These methods can be roughly classified into three categories: object detection based methods, semantic segmentation based methods, and bottom-up methods.

Object detection based methods. These methods adapt state-of-the-art top-down object detection or instance segmentation frameworks to solve the table detection problem. Initially, Hao et al. [4], Yi et al. [45], and Oliveira et al. [46] adopted R-CNN [47] for table detection first, but the performance of these methods was limited by the traditional region proposal generation methods, which relied on the heuristic rules and handcrafted features. Later, more advanced object detectors, like Fast R-CNN [48], Faster R-CNN [12], YOLO [49], RetinaNet [50], Mask R-CNN [13], Cascade Mask R-CNN [14], were explored by Vo et al. [5], Gilani et al. [6], Schreiber et al. [15], Huang et al. [7], Zheng et al. [8], Saha et al. [9], Prasad et al. [10] and Agarwal et al. [11], to detect tables (as well as other page objects like figures and formulas) from document images, respectively. The accuracy of these detectors for table detection could be improved further by adding some effective techniques. For example, Gilani et al. [6], Arif et al. [51], and Prasad et al. [10] proposed to use image transformation techniques, e.g., distance transforms, coloration and dilation, to enhance input document images or augment the used training sets so that additional clues could be provided to the detectors. Siddiqui et al. [52] incorporated deformable convolution and deformable RoI Pooling operations [53] into Faster R-CNN to make the model more robust to geometric transformations. Agarwal et al. [11] employed a more powerful backbone network, i.e., a composite backbone network [54] with deformable convolution filters, to push the accuracy of Cascade Mask R-CNN further. Although this method achieved state-of-the-art performance on several benchmark datasets (e.g., [30-32, 55-57]), it suffered from high computation complexity and memory usage. То improve the localization accuracy, Sun et al. [58] adopted Faster R-CNN to detect table boxes and the corresponding corner boxes simultaneously and used a post-processing algorithm to adjust table boundaries according to the detected corners. However, the corner boxes are manually predefined small boxes, and the size has no explicit meaning, which leads to higher miss detection rate for corners [58].

Semantic segmentation based methods. These methods (e.g., [59-62]) treat table detection as a semantic segmentation problem and leverage existing semantic segmentation frameworks like FCN [63] to predict a pixel-level segmentation mask first, and then group table pixels into tables. Yang et al. [59] proposed a multimodal FCN for page segmentation to detect tables and other page objects, in which both visual features from images and linguistic features from the content of underlying texts are leveraged to improve segmentation accuracy. He et al. [60] proposed a multi-scale multi-task FCN to predict two sets of segmentation masks for text-block/table/figure regions and their corresponding contours first. After refined by a conditional random field (CRF) model, these segmentation masks are then input to a post-processing module to obtain table regions. Kavasidis et al. [61] proposed a saliency-based FCN performing multi-scale reasoning on visual cues followed by a fully connected CRF for localizing tables and charts in digital/digitized documents.

Bottom-up methods. Most bottom-up methods model each document image as a graph, where each node represents a page object (e.g., word, text-line) and each edge represents a neighboring relationship between two page objects, and then formulate table detection as a graph labeling problem. Li et al. [64] used traditional layout analysis methods to generate line regions first, then applied two hybrid CNN-CRF models to classify them into four classes (text, formula, table, figure) and predict whether each pair of line regions belong to a same cluster, respectively. After that, regions belonging to the same class and the same cluster were merged to get page objects. Riba [65] and Holeček et al. [66] took text et al. regions (words or text-lines) as nodes and generated a visibility or neighborhood graph to represent the underlying structure of each input document first, then used graph neural networks to perform node and edge classification. After that, connected subgraphs where the nodes are classified as table were extracted as tables. Recently, Li et al. [67] proposed to consider document layout analysis as a text-based sequence labeling problem and leveraged pre-trained language models to classify each word into a pre-defined page object category, including table. These bottom-up methods depend on certain assumptions like availability of accurate word/text-line bounding boxes as additional inputs.

2.2. Table structure recognition

2.2.1. Traditional methods

Early table structure recognition methods were mainly based on handcrafted features and heuristic rules. These methods (e.g., [33, 68–71]) are mostly applied to simple table structures or specific data formats, such as PDF files. To reduce the dependence on heuristics, a few statistical machine learning based methods were proposed later, e.g., [72]. A comprehensive review of these traditional methods can be found in [44]. These traditional methods usually make strong assumptions about table layouts and rely on domain-specific heuristics, which limit their generalization ability.

2.2.2. Deep learning based methods

Recently, there is a trend to leverage deep learning models to solve the TSR problem. These methods can be roughly divided into three categories: row/column extraction based methods, image-to-markup generation based methods and bottom-up methods.

Row/column extraction based methods. These methods usually adopt object detection or semantic segmentation frameworks to detect rows and columns from a table image first, then intersect the detected rows and columns to generate a grid of cells. DeepDeSRT [15] is the first to apply FCN based semantic segmentation models to the TSR task. They adopted two FCN models to segment tables into rows and columns first, and then used post-processing algorithms to deal with spurious detection fragments as well as severed and conjoined structures. However, this vanilla FCN based row/column segmentation method cannot robustly predict complete segmentation masks for rows and columns when tables contain large blank spaces [16, 17]. To alleviate this problem, Siddiqui et al. [16] and Tensmeyer et al. [17] pooled features along rows and columns of pixels on some intermediate feature maps, which enabled their FCN models to leverage much wider contextual information to improve row/column segmentation accuracy. Instead of relying on FCN, Khan et al. [18] proposed to use sequential models like bi-directional gated recurrent unit networks (GRU) to scan pre-processed table images from topto-bottom and left-to-right to identify row and column separators. Siddiqui et al. [19] proposed to formulate the problem of row/column identification in a tabular structure as an object detection problem instead of a semantic segmentation problem, and leveraged three object detection models, namely deformable Faster R-CNN, deformable R-FCN and deformable FPN, to detect the bounding boxes of rows and columns from tables directly. Hashmi et al. [20] adopted another object detection model, i.e., Mask R-CNN with optimized anchors, to further improve row/column detection accuracy. All the abovementioned methods, except Tensmeyer et al. [17], didn't take spanning cells into consideration and can only recover the basic grid structures of tables. To deal with spanning cells, Tensmeyer et al. [17] presented the SPLERGE method, which used a Split model to produce the grid structure of an input table first, and then used a Merge model to predict which grid elements should be merged to recover spanning cells. Differing from this two-stage paradigm, Zou et al. [73] proposed a onestage approach to segmenting the real row and column separators directly to avoid over-splitting spanning cells. Although these methods have achieved promising results on some benchmark datasets, e.g., [16, 30, 55], they cannot be directly applied to distorted or even curved tables as they rely on an implicit assumption that tables are axis-aligned.

Image-to-markup generation based methods. These methods treat table recognition as an image-tomarkup generation problem and adopt existing imageto-markup models to directly convert each source table image into target presentational markup that fully describes its structure and cell contents. Deng et al. [74] constructed a new dataset TABLE2LATEX-450K and proposed to use an attentional encoder-decoder model to convert tables into LaTeX source codes. Li et al. [57] defined a set of HTML tags to describe table structures only and presented a new table benchmark dataset known as TableBank. Zhong et al. [26]introduced another large scale table benchmark dataset PubTabNet, which contains 568k table images with corresponding structured HTML representation, and introduced an attention-based encoder-dual-decoder architecture to recognize table structures and cell contents simultaneously. These methods rely on a large amount of training data and still struggle with big and complex tables [26, 57].

Bottom-up methods. One group of bottom-up methods [21, 25, 75, 76] treat words or cell contents as nodes in a graph and use graph neural networks to predict whether each sampled node pair is in a same cell, row, or column. These methods rely on an assumption that the bounding boxes of words or cell contents are available as additional inputs, which are not easy to obtain from table images directly. To eliminate this assumption, another group of methods [8, 22, 23] proposed to detect the bounding boxes of table cells directly. After cell detection, Zheng et al.



Figure 1: An outline of our table extraction approach.

[8] and Qiao et al. [23] designed some rules to cluster cells into rows and columns. In order to improve the accuracy of both cell detection and cell clustering, Raja et al. [22] introduced a novel loss function that modeled the inherent alignment of cells in the cell detection network, and a graph-based problem formulation to build associations between the detected cells. However, this method still fails to handle tables containing a large number of empty cells and distorted tables.

3. Methodology

3.1. Overview

Our table extraction approach, RobusTabNet, consists of two deep learning models, i.e., a table detector and a table structure recognizer. For each input image, we first use our table detector to detect all tables within it and crop them from the original image. Then, each cropped table image is resized to an appropriate resolution and fed into the table structure recognizer to reconstruct its cellular structure. Finally, the recognition results are mapped back onto the original image. An outline of our approach and the expected outputs are shown in Fig. 1. Details of our table detector and structure recognizer will be introduced in Section 3.2 and Section 3.3, respectively.

3.2. CornerNet-FRCN based table detector

Existing CNN-based table detection methods typically use RPN to generate table proposals. We find that the percentage of well-localized table proposals (IoU>0.9) in the positive samples (IoU>0.7) generated by RPN is not high enough, which is an important reason for the unsatisfactory localization accuracy of these table detectors (see analysis in Section 5.3.2). To address this issue, we propose to use CornerNet to detect the top-left and bottom-right corners of all table bounding boxes first and then group each pair of top-left and bottom-right corners to obtain table proposals. As table corners can be precisely inferred from ruling lines and alignment of cell contents in tables, the positive proposals (IoU>0.7) generated by our approach will be



Figure 2: Overall architecture of our CornerNet-FRCN based table detection approach.

of higher localization accuracy, which can improve the localization accuracy of our table detector effectively. After that, we use a simple Fast R-CNN module to reject non-table proposals and refine the bounding boxes of remaining positive proposals further.

The overall architecture of our approach is illustrated in Fig. 2. There are three core modules: 1) A CNN backbone network that is responsible for computing a shared convolutional feature map; 2) A CornerNet based region proposal generation module, which detects the top-left and bottom-right corners of the tables and enumerates all the potential table proposals; 3) A Fast R-CNN (FRCN) module, which is used to prune non-table proposals and refine the bounding boxes of remaining table proposals. For the sake of efficiency, a ResNet-18 network with dilations in "Conv5" is used as the backbone network, and the stride of the output feature map, named Dilated-C5, is 16 pixels. We further use a 1×1 convolutional layer to reduce the channel dimension of Dilated-C5 from 512 to 64 for computational efficiency.

3.2.1. CornerNet as region proposal network

CornerNet [27] detects an object as a pair of keypoints, i.e., the top-left and bottom-right corners of the bounding box. It uses a convolutional network to predict two sets of heatmaps to represent the locations of the top-left and bottom-right corners of different object categories respectively, as well as an embedding vector for each detected corner such that the distance between the embeddings of two corners from the same object is small. To produce tighter bounding boxes, the network also predicts offsets to slightly adjust the locations of the corners. With the predicted heatmaps, embeddings and and q_j be a pixel on the feature map and raw image with the coordinates of (p_i^x, p_i^y) and (q_j^x, q_j^y) , respectively. We define that p_i is corresponding to q_j if

$$p_i^x = \left\lfloor \frac{q_j^x}{s} \right\rfloor and \ p_i^y = \left\lfloor \frac{q_j^y}{s} \right\rfloor, \tag{1}$$

where *s* denotes the stride of the feature map. If p_i is corresponding to q_j and q_j is a top-left corner point, the detection module will give p_i a "top-left corner" label

offsets, a simple post-processing algorithm is applied to

obtain the final bounding boxes. However, Duan et al.

[77] find that the performance of the abovementioned

corner grouping method is restricted by its relatively

weak ability of referring to the global information of an object. Therefore, in this work, we abandon the

embedding vectors and adopt CornerNet as a new region

proposal network for Faster R-CNN by detecting and

As illustrated in Fig. 2, we append a 3×3

convolutional layer with the stride of 1 on Dilated-

C5 to generate a new feature map Dilated-C5', on

which two sibling branches are attached for detecting

top-left and bottom-right corners, respectively. Taking

the top-left corner detection branch as an example, we

first use a top-left corner pooling module, composed

of a top pooling and a left pooling operator [27], to

aggregate context information. The context enhanced

feature map is fused with the original feature map in a

residual connection manner. Then, a detection module

is attached to this feature map and performs dense perpixel prediction of top-left corners. Specifically, let p_i

exhaustively grouping corner points.

and predict the corresponding offset Δ_i defined by

$$\Delta_i = \left(\frac{q_j^x}{s} - \left\lfloor \frac{q_j^x}{s} \right\rfloor, \frac{q_j^y}{s} - \left\lfloor \frac{q_j^y}{s} \right\rfloor\right),\tag{2}$$

to adjust the location of the corner to compensate for the quantization error caused by network downsampling. As depicted in Fig. 2, the detection module contains two parallel branches, a 3×3 convolutional layer followed by a 1×1 convolutional layer in each branch, for corner/non-corner classification and corner offset regression, respectively. Furthermore, if a pair of false corner detections are close to the corresponding ground truth corner locations, they can still produce a box that highly overlaps with the ground-truth box. Therefore, during training, we reduce the penalty of the negative locations within a radius r of the positive location, and the radius r is determined by the size of the ground-truth box. We refer readers to [27] for more details about the selection of r. Once r is determined, the amount of penalty reduction is given by an unnormalized 2D Gaussian, $e^{-\frac{x^2+y^2}{2\sigma^2}}$, whose center is at the positive location and σ is set as r/3.

To generate table proposals, we first apply nonmaximal suppression (NMS) by using a 3×3 max pooling layer on the corner heatmaps. Then top-*K* top-left and bottom-right corners are extracted from the heatmaps, which are further filtered by a score threshold, C_{th} . The locations of the remaining corners are adjusted by the corresponding predicted offsets. Then, we take all the valid combinations, i.e., the *x* and *y* coordinates of the top-left corner are smaller than that of the bottom-right corner, as table proposals, so that a high recall rate is retained. After that, we use the standard NMS algorithm with an IoU threshold of 0.7 to remove redundant proposal boxes.

3.2.2. Fast R-CNN

Given the extracted region proposals, we adopt a Fast R-CNN module to reject negative (non-table) proposals and refine the bounding boxes of positive (table) proposals. As shown in Fig. 2, for each proposal, we first adopt an RoI Align algorithm [13] to extract a $7 \times 7 \times 64$ feature descriptor from the proposal box on the *Dilated-C5* feature map. Then, it is fed into two 1,024-d fully connected (*fc*) layers (each followed by a ReLU activation function) before the final table/non-table classification and bounding box regression layers. During training, a proposal is assigned a positive label if it has an IoU over 0.7 with any ground-truth bounding box, or a negative label if it has IoU lower than 0.5 for all ground-truth bounding boxes. An online hard



Figure 3: Flowchart of our table structure recognition approach.

example mining (OHEM) method is adopted to select an equal number of hard positive and hard negative samples to train the Fast R-CNN module.

3.3. Split-and-merge based table structure recognizer

After table detection, each detected table is cropped from the raw image and resized to an appropriate size to ensure that there is enough inter-line spacing for separation line prediction. Then, each resized table image is fed into a table structure recognizer to reconstruct its cellular structure. The flowchart of our table structure recognizer is shown in Fig. 3. Given a cropped table image, a spatial CNN based separation line prediction module is used to predict a row separator mask and a column separator mask first. Then, a connected component analysis (CCA) based line generation algorithm is used to extract all row and column separation lines from the predicted separator masks, which are intersected to generate a grid of cells. After that, a Grid CNN based cell merging module is adopted to merge wrongly split cells into spanning cells. The separation line prediction module and cell merging module share a same CNN backbone network and are trained jointly. For the sake of efficiency, we adopt the Feature Pyramid Network (FPN) [78], which is built on top of ResNet-18, as the backbone network. Details of the separation line prediction and cell merging modules are described in Section 3.3.1 and 3.3.3, respectively.

3.3.1. Spatial CNN based separation line prediction

Some important visual clues like ruling lines and alignment of cell contents provide useful hints to indicate whether a separation line exists at a position within a table. However, the ResNet-FPN backbone cannot embed such useful visual clues into the features of pixels in large blank regions of borderless tables effectively, because each feature vector on the output convolutional feature map only contains local context information extracted from its effective receptive field. Based on such convolutional feature map, it is hard for the following separation line segmentation module to predict separation lines from large blank spaces in borderless tables robustly, because the feature vector



Figure 4: Overall architecture of our spatial CNN based separation line prediction module.

of each pixel in these blank regions does not contain enough information to determine whether a separator line passes through this pixel or not. To address this issue, we propose to use spatial CNN modules [29] to enhance the feature representation of each pixel on the convolutional feature map by propagating contextual information across the whole feature map in left-right or top-bottom directions.

The overall architecture of our spatial CNN based separation line prediction module is depicted in Fig. 4. Given an input image $X \in R^{H \times W \times 3}$, we adopt the FPN backbone network to generate a shared convolutional feature map $P_2 \in R^{\frac{H}{4} \times \frac{W}{4} \times C}$, where C represents the number of channels and is set to 64 in our experiments. Then, two parallel semantic segmentation branches are attached to P_2 to predict a row separator mask \hat{S}^{row} and a column separator mask \hat{S}^{col} , respectively. Taking the row separation line prediction branch as an example, we add a 3×3 convolutional layer and three repeated down-sampling blocks, each composed of a sequence of a 1×2 max-pooling layer, a 3×3 convolutional layer and a ReLU activation function, after P_2 sequentially to generate a down-sampled feature map $P'_{2} \in R^{\frac{H}{4} \times \frac{W}{32} \times C}$, which is taken as the input of two cascaded spatial CNN modules. The first spatial CNN module divides the feature map into $\frac{W}{32}$ slices along the width direction, which are denoted as $S^w = \{s_i^w \in R^{\frac{H}{4} \times 1 \times C} | i \in N, i = w\}$ 1, 2, ..., $\frac{W}{32}$ } then propagates the information from the leftmost slice s_1^w to the rightmost slice $s_{W/32}^w$ with convolution operators. Specifically, the leftmost slice s_1^w is convolved by a convolution kernel with the kernel size of 9×1 (9 and 1 represent kernel height and

width respectively) and its output feature map is merged with its right slice s_2^w by element-wise addition. This procedure is done iteratively so that the information can be propagated from the leftmost slice to the rightmost slice effectively. The second spatial CNN module uses the same method to propagate the information from the rightmost slice $s_{W/32}^w$ to the leftmost slice s_1^w . In this way, each pixel in the output feature map can leverage the structural information from both sides to enhance its feature representation ability. Finally, this contextenhanced feature map is up-sampled by a factor of 4 with a bilinear interpolation operation to generate an output feature map $P_{out} \in R^{H \times \frac{W}{8} \times C}$, on which a 1×1 convolutional layer followed by a sigmoid activation function is attached to predict a row separator mask $\hat{S}^{row} \in R^{H \times \frac{W}{8} \times 1}$. The architecture of the column separation line prediction branch is similar to the row separation line prediction branch, except that the downsampling is performed along the height direction and the two spatial CNN modules propagate information from the topmost slice to the bottommost slice and from the bottommost slice to the topmost slice, respectively.

To generate the ground-truth (GT) row and column separator masks, the row and column separation lines of each table as well as the bounding boxes of textlines in each cell are annotated (Fig. 5(a)). Following SPLERGE [17], we calculate the GT separator masks by maximizing the size of the separation regions without intersecting any non-spanning cell contents, as shown in Fig. 5(b). Specifically, for each annotated row separation line, we move it upwards and downwards respectively until it touches a text box that belongs to



Figure 5: Illustration of the ground-truth generation for table structure recognition. (a) Annotated text boxes and separation lines; (b) Expanded separation lines; (c) Ground-truth cell boxes, including spanning cells; (d) If a pair of neighboring shrunk cells (blue boxes) detected by the split model are assigned to a same ground-truth cell box, we will give this pair a positive label, otherwise a negative label, to train the cell merging module. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

a non-spanning cell to obtain its corresponding row separation region. The similar procedure can also be applied to generate column separation regions. After this step, if the thickness of a separation region is less than 8 pixels, we will further expand it to make sure that its thickness is at least 8 pixels.

3.3.2. Cell generation

After separation line prediction, we first binarize the predicted row and column separator heatmaps with a classification score threshold, S_{th} . Then, we extract the connected components (CCs) from the segmentation masks, which represent detected separators. With these CCs, we can extract all row and column separation lines as well as their corresponding line thicknesses. Taking row separation line generation as an example, we first apply the *findContours* method in OpenCV [79] to the binarized row separator mask to obtain the contours of row CCs. Then for each row CC, we use a polynomial curve fitting algorithm to fit a function y = f(x) from its contour points, which approximates the center line of this row CC. To compute the corresponding line thickness, a vertical scan-line is used to traverse the related row CC mask from left to right with a stride of 8 pixels. On each scanned pixel column, a line segment can be obtained by intersecting the scan-line with the upper and lower boundaries of the row CC. By averaging the lengths of all the line segments, we can estimate the line thickness of the separation line, denoted by lw. Then, we translate the fitted separation



Figure 6: A schematic view of the Grid CNN based cell merging module.

line y = f(x) upwards and downwards respectively to generate two border lines, $y = f(x) + \frac{hw}{2}$ and $y = f(x) - \frac{hw}{2}$. Similarly, the column separation lines can also be generated by rotating the column separator heatmap with 90 degrees before running this algorithm. Finally, we intersect all the translated row lines with column lines to calculate all intersection points, from which we can extract all the shrunk cell boxes, e.g., the blue box in Fig. 6, and arrange them in a grid manner.

3.3.3. Grid CNN based cell merging

Based on the compact grid representation, we introduce a Grid CNN module to aggregate context information effectively with several stacked convolution layers to improve cell merging accuracy. As shown in Fig. 6, we first use an RoI Align algorithm to extract a $7 \times 7 \times 64$ feature descriptor from the bounding box of each cell, which is then fed into a 2-hidden-layer fully connected (fc) neural network with 512 nodes at each layer to generate a 512-d feature vector. Assume that the detected cells are arranged in M rows and Ncolumns, then the corresponding feature vectors will construct a new feature map $F_{grid} \in R^{M \times N \times 512}$. Each pixel in F_{grid} corresponds to a cell generated by the split model. F_{grid} is then convolved by three 3 × 3 convolutional layers to obtain an enhanced feature map $F'_{grid} \in \mathbb{R}^{M \times N \times 512}$. Finally, we use a relation network $\begin{bmatrix} g^{r/u} \\ [80] \end{bmatrix}$ to predict whether each pair of adjacent pixels on F'_{grid} , which corresponds to each pair of adjacent cells in the input table image, should be merged or not. Here, we only consider 4-adjacency neighborhood relations to construct relational pairs.

The architecture of the relation network is shown in the blue dashed box in Fig. 6. Given a pair of adjacent pixels in \mathbf{F}'_{grid} , p_i and p_j , whose corresponding feature vectors are \mathbf{F}'_{grid} and $\mathbf{F}'_{grid}^{p_j}$ and corresponding cell bounding boxes are b_i and b_j respectively, we extract a feature representation \mathbf{x}_{ij} to encode the appearance compatibility and spatial compatibility of their corresponding cells. Specifically, \mathbf{x}_{ij} is constructed by concatenating the appearance features $\mathbf{F}'_{grid}^{p_i}$ and $F_{grid}^{'p_j}$ and the spatial compatibility feature I_{ij} of b_i and b_j , i.e., $x_{ij} = [F_{grid}^{'p_i}; I_{ij}; F_{grid}^{'p_j}]$. The spatial compatibility feature I_{ij} is used to measure the relative scale and location relationships between b_i and b_j . Following Zhang et al. [80], let b_{ij} denote the union bounding box of b_i and b_j , then I_{ij} is defined as an 18-d vector concatenating three 6-d vectors, which indicate the box delta of b_i and b_j , b_i and b_{ij} , b_j and b_{ij} , respectively. Given two bounding boxes $b_i = \{x^i, y^i, w^i, h^i\}$ and $b_j = \{x^j, y^j, w^j, h^j\}$, their box delta is defined as $\Delta(b_i, b_j) = (I_x^{ij}, I_y^{ij}, I_w^{ij}, I_h^{ji}, I_x^{ji}, I_y^{ji})$ where each dimension is given by

$$t_{x}^{ij} = (x^{i} - x^{j})/w^{i}, \quad t_{y}^{ij} = (y^{i} - y^{j})/h^{i},$$

$$t_{w}^{ij} = \log(w^{i}/w^{j}), \quad t_{h}^{ij} = \log(h^{i}/h^{j}), \quad (3)$$

$$t_{x}^{ji} = (x^{j} - x^{i})/w^{j}, \quad t_{y}^{ji} = (y^{j} - y^{i})/h^{j}.$$

A binary classifier is applied on the feature representation to predict whether each pair of cells should be merged or not. It is implemented with a 2-hidden-layer MLP with 512 nodes at each hidden layer and a sigmoid activation node at its output layer. Note that in the inference stage, each pair of cells is predicted twice for the inputs x_{ij} and x_{ji} , and the maximum value is taken as the final merging score for this pair of cells.

In the training stage, we use detected cells from the split model to generate positive and negative relational pairs for training the cell merging module. As illustrated in Fig. 5(c-d), given all the ground-truth cell boxes, a detected cell box b_{det} is assigned to a ground-truth cell box b_{gt} if the following condition is satisfied, i.e.,

$$\frac{Area(\boldsymbol{b}_{det} \cap \boldsymbol{b}_{gt})}{Area(\boldsymbol{b}_{det})} > 0.5, \tag{4}$$

where $Area(b_{det})$ and $Area(b_{det} \cap b_{gt})$ denote the area of b_{det} and the area of the overlap between b_{det} and b_{gt} , respectively. Then, each detected cell is paired with each of its 4-connected cells to construct candidate relational pairs. If two cells in a relational pair are assigned to a same ground-truth cell box, we give this relational pair a positive label, otherwise a negative label. During training, we ignore all the negative relational pairs that contain cells not assigned to any ground-truth cell, and then adopt an OHEM method to select hard samples to train the cell merging module.

4. Loss Function

4.1. Table detection

Loss for CornerNet based region proposal network. There are two sibling output layers for each corner detection branch, i.e., a corner/non-corner classification layer and an offset regression layer. The multi-task loss function can be defined as follows:

$$\mathcal{L}_{corner} = \frac{1}{N_t} \sum_i \mathcal{L}_{det}(c_i, c_i^*) + \frac{1}{N_c} \sum_j \mathcal{L}_{off}(t_j, t_j^*), \quad (5)$$

where N_t and N_c denote the number of tables and the number of corners in a mini-batch respectively, c_i and c_i^* are the predicted and "ground-truth" labels for the *i*-th pixel on the heatmap, c_i^* has been augmented with the unnormalized Gaussians to reduce the penalty around the ground-truth locations, $\mathcal{L}_{det}(c_i, c_i^*)$ is a variant of focal loss as in [27] for classification tasks, t_j and t_j^* are predicted and ground-truth 2-d coordinate offsets defined by Eq. 1 and Eq. 2 for the *j*-th corner, $\mathcal{L}_{off}(t_j, t_j^*)$ is a Smooth- L_1 loss[12] for regression tasks.

Loss for Fast R-CNN. There are two sibling output layers for the Fast R-CNN module, i.e., a table/nontable classification layer and a quadrilateral bounding box regression layer. The multi-task loss function is defined as follows:

$$\mathcal{L}_{frcn} = \frac{1}{N} \sum_{i} \mathcal{L}_{cls}(k_i, k_i^*) + \frac{1}{N_{fg}} \sum_{j} \mathcal{L}_{reg}(b_j, b_j^*), \quad (6)$$

where N is the number of sampling region proposals (including N_{fg} positive ones), k_i and k_i^* are predicted and ground-truth labels for the *i*-th sampling region proposal respectively, $\mathcal{L}_{cls}(k_i, k_i^*)$ is a cross-entropy loss for classification tasks, b_j and b_j^* are predicted and ground-truth 8-d normalized coordinate offsets as stated in [81] for the *j*-th positive region proposal, $\mathcal{L}_{reg}(b_j, b_j^*)$ is an L_1 loss for regression tasks.

Total loss for table detector. With the definitions of \mathcal{L}_{corner} and \mathcal{L}_{frcn} , the training loss for the table detector can be defined as follows:

$$\mathcal{L}_{detector} = \lambda_{corner} \cdot \mathcal{L}_{corner} + \mathcal{L}_{frcn}, \tag{7}$$

where λ_{corner} is a loss-balancing parameter, and we set $\lambda_{corner} = 0.2$.

4.2. Table structure recognition

Loss for spatial CNN based separation line prediction. There are two branches in the separation line prediction module for row and column separator prediction, respectively. The total loss of this module is the sum of the losses of two branches. Let N_{row} and N_{col} denote the number of sampling pixels for row and column separator prediction branch respectively, $\{R_i, C_j\}$ and $\{R_i^*, C_j^*\}$ be the predicted and ground-truth labels for the *i*-th sampling pixel on the row

separator heatmap and the *j*-th sampling pixel on the column separator heatmap respectively, and $\mathcal{L}(R_i, R_i^*)$ and $\mathcal{L}(C_j, C_j^*)$ be the binary cross-entropy loss for classification tasks. Based on these definitions, the loss function for the separation line prediction module can be defined as follows:

$$\mathcal{L}_{split} = \frac{1}{N_{row}} \sum_{i} \mathcal{L}(R_i, R_i^*) + \frac{1}{N_{col}} \sum_{j} \mathcal{L}(C_j, C_j^*).$$
(8)

Loss for Grid CNN based cell merging. Let N_p be the number of selected relational pairs for cell merging, r_i and r_i^* be the predicted and ground-truth labels for the *i*-th relational pair, and $\mathcal{L}(r_i, r_i^*)$ be a binary crossentropy loss for classification tasks. The loss function for the cell merging module is defined as follows:

$$\mathcal{L}_{merge} = \frac{1}{N_p} \sum_i \mathcal{L}(r_i, r_i^*).$$
(9)

Total loss for table structure recognizer. With the definitions of \mathcal{L}_{split} and \mathcal{L}_{merge} , the training loss for the table structure recognizer can be defined as follows:

$$\mathcal{L}_{recognizer} = \mathcal{L}_{split} + \mathcal{L}_{merge}.$$
 (10)

5. Experiments

5.1. Datasets and evaluation protocols

We conduct comprehensive experiments on three table detection benchmark datasets, including cTDaR TrackA [30], PubLayNet [31] and IIIT-AR-13K [32], and three table structure recognition datasets, including SciTSR [25], PubTabNet [26] and cTDaR TrackB2-Modern [30], to evaluate the performance of our table detection and structure recognition approaches, respectively. We follow the evaluation protocols defined by the authors to make our results comparable to the ones reported by other methods. Moreover, to demonstrate the advantage of our TSR approach in dealing with geometrically distorted tables, we have also collected a much more challenging in-house dataset which contains many distorted or even curved tables.

cTDaR TrackA [30] contains both historical and modern document images. The historical subset contains hand-drawn tables and handwritten texts, including 600 images for training and 199 images for testing. The modern subset contains printed PDF documents, including 600 images for training and 240 images for testing. It adopts the weighted average (WAvg.) F1-score as evaluation metric, which is calculated with the IoU thresholds of 0.6, 0.7, 0.8 and 0.9. **PubLayNet** [31] is a high-quality dataset for document layout analysis, which contains 335,703 images for training, 11,245 images for validation and 11,405 images for testing. We use it for table detection performance evaluation, and only use the images containing at least a table for model training (86,460 images). Since the annotations of the testing set are not released, we only report results on the validation set. The COCO evaluation protocol is used as the evaluation metric of this dataset.

IIIT-AR-13K [32] is introduced for graphical object detection in annual reports, which contains 9,333 images for training, 1,955 images for validation and 2,120 images for testing. This dataset is used for table detection performance evaluation only. The PASCAL VOC evaluation protocol is used as the evaluation metric of this dataset.

SciTSR [25] contains 12,000 training images and 3,000 testing images cropped from scientific papers. To evaluate the performance of different methods on complicated tables, authors also extract all the 716 complicated tables from the test set as a test subset, called SciTSR-COMP. The adjacency relation-based evaluation metric, which is used in ICDAR-2013 table competition [55], is employed as the evaluation metric of this dataset.

PubTabNet [26] contains 500,777 training images, 9,115 validating images and 9,138 testing images. This dataset contains a large number of three-lines tables with empty or spanning cells. Since the annotations of testing set are not released, we only report results on the validation set. The authors proposed a new Tree-Edit-Distance-based Similarity (TEDS) metric for table recognition task, which can identify both table structure recognition and OCR errors. Some recent works [8, 22, 23] have proposed a modified TEDS metric, denoted as TEDS-Struct, to evaluate table structure recognition accuracy only by ignoring OCR errors. We also use the TEDS-Struct metric to evaluate our table structure recognition approach on this dataset.

cTDaR TrackB2-Modern [30] contains no images for training, but 100 images with annotations are provided as testing data. To evaluate our approach on this dataset, we manually labeled the structures of tables in the cTDaR TrackA modern subset, which contains 600 training images. The annotations will be released publicly to facilitate future research in this area. It has been checked that there is no overlap between the 600 training images and the 100 testing images. The adjacency relation-based metric³ is used as the

³https://github.com/cndplab-founder/ctdar_measurement_tool

evaluation metric of this dataset. During evaluation, the convex hull of the content is used to represent a cell. Note that both table region detection and table structure recognition have to be done on this dataset.

Private Dataset. Our in-house dataset is composed of 9,000 training images and 700 testing images. Most images in this dataset are captured by cameras so that many tables in this dataset are skewed or even curved. Sample images in this dataset are shown in Fig. 10 and Fig. 11. We use the same adjacency relation-based metric as cTDaR TrackB to evaluate our table structure recognition approach on this dataset.

5.2. Implementation Details

For both table detector and structure recognizer, the weights of ResNet-18 related layers are initialized with a pre-trained ResNet-18 model for the ImageNet classification task. The weights of newly added layers are initialized with a Gaussian distribution of mean 0 and standard deviation 0.01. The models are optimized by a standard SGD algorithm with a momentum of 0.9 and weight decay of 0.0005. Unless otherwise specified, all the models are trained for 15K iterations with a base learning rate of 0.032, which is divided by 10 at 10K and 13K iterations, respectively. The table detection model for PubLayNet is trained with a 3× training schedule because of the larger amount of data. The TSR models for SciTSR and PubTabNet are trained for 12 epochs. Besides, we apply synchronized batch normalization across multiple GPUs to stabilize the training.

We implement our approach based on PyTorch⁴ v1.6.0 and conduct experiments on a workstation with 8 Nvidia V100 GPUs. In each training iteration, we sample 4 images for each GPU. In the training phase of our table detector, since the number of positive proposals is small, we add some synthesized samples by introducing random jittering to ground-truth boxes. Then, for each image, we select a mini-batch of 32 hard positive and 32 hard negative proposals for the FRCN detector. We adopt a multi-scale training strategy during training. While keeping the aspect ratio, the shorter side of each selected training image is randomly rescaled to a number in {320, 416, 512, 608, 704, 800}. Moreover, when training our table detector on cTDaR TrackA, we also rotate training images by a random angle in $\{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ with $\pm 5^\circ$ angle jitter for data augmentation. In the training phase of our table structure recognizer, we use the cropped table images for training and the shorter side of each

selected training image is randomly rescaled to a number in {416, 512, 608, 704, 800} while keeping the aspect ratio. For each image, we sample a minibatch of 1,024 row/column separator pixels and 1,024 background pixels for each separation line prediction branch. Furthermore, we select a minibatch of 64 hard positive and 64 hard negative cell pairs for the cell merging module.

In the testing phase of table detection, the shorter side of each testing image is rescaled to be 512 pixels with the longer side not exceeding 1,024 pixels. We set the number of selected corners (top-K) as 100 with a corner score threshold C_{th} as 0.3. We apply the standard NMS algorithm with an IoU threshold of 0.3 on the detected tables to suppress redundant detections. For TSR testing, we rescale the longer side of each cropped table image to be 1,024 pixels while keeping the aspect ratio, except for the SciTSR dataset where the cropped images are not resized. The binarization score threshold S_{th} is set as 0.8. The grid cells from the split model are merged based on the merging scores with a threshold of 0.8. Abovedmentioned hyper-parameters are tuned on our in-house dataset, and we directly apply them to other datasets without further tuning.

5.3. Experiments on table detection

5.3.1. Comparisons with prior arts

We compare our table detection approach with other most competitive methods on cTDaR TrackA, PubLayNet and IIIT-AR-13K. All the results of our approach are based on single-model and single-scale testing. The results are listed in Table 1, Table 2 and Table 3. On cTDaR TrackA, our approach achieves the best WAvg. F1-score of 94.9%, outperforming other methods by a notable margin. Furthermore, it is noted that our approach has achieved the best F1scores at higher IoU thresholds, e.g., 92.9% vs. 91.5% with the IoU threshold at 0.9, which demonstrates the superiority of our approach on high precision table localization. On PubLayNet, our model with the ResNet-18 backbone network can even substantially outperform the Mask R-CNN model with the ResNeXt-101 backbone network by improving the $AP^{0.5:0.95}$ from 96.0% to 97.0%, and significantly improving the AP^{0.95} from 81.4% to 92.0%. Similarly, on IIIT-AR-13K, our model can also substantially outperform the Mask R-CNN model with the ResNet-101 backbone network by improving the AP from 97.6% to 98.2% on the validation set and from 96.5% to 97.7% on the testing set, respectively. To push the table detection performance of the Cascade Mask R-CNN

⁴https://pytorch.org/

Table 1: Table detection performance comparison on ICDAR2019 cTDaR TrackA. * indicates that the results are from [30]

Mathada	IoU	J@0.6	(%)	IoU	J@0.7	(%)	IoU	0.8	(%)	IoU	WAvg.		
Wiethous	Р	R	F1	Р	R	F1	Р	R	F1	Р	R	F1	F1(%)
Applica-robots*	90.3	90.1	90.2	88.4	88.1	88.2	82.6	82.4	82.5	54.6	54.4	54.5	77.0
ABC Fintech*	87.4	78.5	82.7	86.3	77.5	81.7	84.1	75.5	79.6	76.8	69.0	72.7	78.6
Lenovo Ocean*	91.8	90.2	90.1	90.8	89.2	90.0	88.5	87.0	87.7	82.9	81.5	82.2	87.7
NLPR-PAL*	97.1	97.5	97.3	96.0	96.4	96.2	93.6	94.0	93.8	86.5	86.9	86.7	92.9
TableRadar*	97.6	96.4	97.0	96.6	95.4	96.0	95.8	93.2	95.1	90.8	89.7	90.2	94.2
CDeC-Net[11]	98.0	93.9	95.9	97.7	93.6	95.6	97.1	93.0	95.0	93.4	89.5	91.5	94.3
RPN+FRCN	97.8	94.8	96.3	97.3	94.3	95.7	96.3	93.3	94.7	92.4	89.5	90.9	94.1
Ours (CornerNet+FRCN)	98.4	94.0	96.1	98.2	93.9	96.0	97.7	93.3	95.4	95.0	90.8	92.9	94.9

Table 2: Table detection performance comparison on the validation set of PubLayNet. * indicates that the results are from [31].

Methods	Backbone	$AP^{0.5:0.95}$	$AP^{0.75}$	$AP^{0.95}$
Faster R-CNN*	ResNeXt-101	95.4	97.8	77.8
Mask R-CNN*	ResNeXt-101	96.0	97.8	81.4
CDeC-Net[11]	Dual ResNeXt-101	96.7	-	-
RPN+FRCN	ResNet-18	96.0	97.5	87.0
Ours (CornerNet+FRCN)	ResNet-18	97.0	97.8	92.0

Table 3: Table detection performance comparison on IIIT-AR-13K. * indicates that the results are from [32].

Methods	Backhone	V	Validatio	n Set(%)		Testing	g Set(%)		
Methous	Dackbolle	Р	R	F1	AP	 Р	R	F1	AP	
Faster R-CNN*	ResNet-101	95.7	92.6	94.2	95.5	95.1	92.3	93.7	93.9	
Mask R-CNN*	ResNet-101	98.2	96.6	97.4	97.6	97.1	97.1	97.1	96.5	
Ours (CornerNet+FRCN)	ResNet-18	98.6	98.3	98.5	98.2	99.0	97.8	98.4	97.7	



Figure 7: Qualitative results of our table detector. (a-b) are from cTDaR TrackA, (c) is from PubLayNet, and (d) is from IIIT-AR-13K.

framework on public benchmarks, Agarwal et al. [11] employed a more powerful backbone network, i.e., dual backbone ResNeXt-101 with deformable convolution filters. However, its performance is still inferior to ours on cTDaR TrackA and PubLayNet. The superior performance achieved on these public benchmark datasets shows the effectiveness and robustness of our approach. Some qualitative results of our approach on these datasets are presented in Fig. 7.

5.3.2. Ablation study

Table 4, which shows that although these two methods can achieve a similar recall rate at the IoU threshold of 0.7, the CornerNet based method can significantly outperform RPN under a higher IoU threshold 0.9, i.e., 97.8% vs. 89.3%. Then, we further evaluate the end-to-end performance and the comparison results are given in the last two rows of Table 2. We can find that the performance of RPN based table detector is inferior to our CornerNet based detector, which shows that the quality of the proposals generated by CornerNet is better than RPN. To reveal the relation between proposal quality and end-to-end detection accuracy, we further compute the maximum IoU between each proposal and all the GT boxes, and the corresponding statistical results are shown in Fig. 8. We find that there are more proposals from RPN within the IoU range of (0.7, 0.9]. As the proposals in this range will also be taken as positive samples during the training of Fast R-

CornerNet vs. RPN for table proposal generation. To compare the proposed CornerNet based table proposal generation algorithm with RPN [12], we evaluate their recall rates with top-50 proposals on PubLayNet first. The quantitative results are given in



Table 4: Table proposal generation quality comparison on PubLayNet.

Figure 8: The distribution of the IoU between proposals and GT boxes, where the x-axis represents the ranges of IoU, and the y-axis represents the ratio of the number of proposals in the corresponding IoU range between the two methods.

CNN, these low quality proposals will also have high classification scores and survive from the NMS step, which will degrade the end-to-end performance when evaluating at high IoU thresholds. Compared with RPN, the percentage of well-localized proposals (IoU>0.9) in the positive samples (IoU>0.7) from CornerNet is much higher (96.3% vs. 48.1%), which contributes to better end-to-end table detection performance. These experimental results demonstrate the superiority of our CornerNet based table proposal generation algorithm for achieving higher localization accuracy and better end-to-end table detection results.

5.4. Experiments on table structure recognition

5.4.1. Comparisons with prior arts

We compare our table structure recognition approach with other most competitive methods on SciTSR, PubTabNet and cTDaR TrackB2-Modern. On SciTSR and SciTSR-COMP (see Table 5), our approach has achieved state-of-the-art performance with the best F1score of 99.3% and 98.7% on the full testing set and the complicated subset, respectively. Moreover, our approach shows negligible performance degradation on the complicated subset, which demonstrates its robustness to tables with complex structures. Similarly, on PubTabNet (see Table 6), our approach has also achieved the best TEDS-Struct score of 97.0%. It is noted that, the recent best performing method LGPMA [23] (the winner of ICDAR 2021 Competition on

Table 5: TSR performance comparison on SciTSR and SciTSR-COMP. * indicates that the results are from [25].

Mathada	Sc	iTSR(9	%)	S	SciTS	R-CON	<u>AP(%)</u>
wiethous	Р	R	F1		Р	R	F1
Adobe*	93.0	78.4	85.1	9	0.1	71.7	79.8
DeepDeSRT[15]*	90.6	88.7	89.0	8	36.3	83.1	84.6
Tabby[70]*	92.6	92.0	92.1	8	39.2	87.2	88.2
TabStruct-Net[22]	92.7	91.3	92.0	9	0.9	88.2	89.5
GraphTSR[25]	95.9	94.8	95.3	9	96.4	94.5	95.5
SEM[82]	97.7	96.5	97.1	9	6.8	94.7	95.7
LGPMA[23]	98.2	99.3	98.8	9	97.3	98.7	98.0
Ours	99.4	99.1	99.3	9	9.0	98.4	98.7

Table 6: TSR performance comparison on the validation set of PubTabNet.

Methods	TEDS(%)	TEDS-Struct(%)
EDD[26]	88.3	-
TabStruct-Net[22]	-	90.1
GTE[8]	-	93.0
LGPMA[23]	94.6	96.7
Ours	-	97.0

Scientific Literature Parsing Task B [83]) has leveraged an important task constraint, namely tables are axisaligned, to achieve higher accuracy. So, it cannot be directly applied to distorted tables. Our approach doesn't rely on such kind of assumptions but still achieves higher accuracy. On cTDaR TrackB2-Modern, our table detector and table structure recognizer are combined together to conduct end-to-end evaluation. Since the outputs of our approach are cell boxes rather than convex hulls of cell contents, for the sake of fair comparison, we use the same text detection algorithm as CascadeTabNet [10] to detect texts in each image and then assign them to table cells if 80% of a text box is located in a cell box. As shown in Table 7, our approach surpasses previous methods by a large margin. Some qualitative results of our approach on these datasets are presented in Fig. 9.

To further validate the robustness of our approach to distorted or even curved table images, we conducted experiments on the in-house dataset and compared our table structure recognizer with SPLERGE. As shown in Table 8, our approach outperforms SPLERGE significantly by improving the WAvg. F1-score from 63.8% to 94.6%. Some qualitative results of our approach on this challenging dataset are presented in Fig. 10, from which we can observe that our table structure recognizer can work robustly under various challenging conditions such as tables without ruling lines, tables with empty or spanning cells and distorted or even curved shapes.

Table 7: TSR Performance comparison on ICDAR2019 cTDaR TrackB2-Modern. * indicates that the results are from [30].

Mathada	Iol	J@0.6((%)	Iol	J@0.7((%)		Ιοι	J@0.8(%)	Ic	U@0.9	(%)	WAvg.
Methods	Р	R	F1	Р	R	F1	P		R	F1	P	R	F1	F1(%)
Zou et al.[73]	18.8	10.1	13.1	-	-	-	1.	7	0.9	1.2	-	-	-	-
NLPR-PAL*	32.2	42.1	36.5	26.9	35.1	30.5	17.	2	22.5	19.5	3.1	4.0	3.5	20.6
CascadeTabNet[10]	49.9	39.0	43.8	40.3	31.5	35.4	21.	6	16.9	19.0	4.1	3.2	3.6	23.2
GTE[8]	-	-	38.5	-	-	-	-		-	-	-	-	-	24.8
Ours	76.4	76.8	76.6	71.3	71.6	71.4	58.	1	58.4	58.3	25.7	25.8	25.8	55.3

	Table 8: TSR performance comparison on the in-house dataset.													
Methods	Iol	J@0.6((%)	Iol	J@0.7(7(%) IoU@0.8(%)			Iol	IoU@0.9(%)				
Wiethous	Р	R	F1	Р	R	F1	Р	R	F1	Р	R	F1	F1(%)	
SPLERGE [17]	75.9	55.2	63.9	75.8	55.1	63.8	75.7	55.0	63.7	75.7	55.0	63.7	63.8	
Ours	94.9	94.5	94.7	94.8	94.4	94.6	94.8	94.4	94.6	94.7	94.3	94.5	94.6	



Figure 9: Qualitative results of our table structure recognition approach. (a-b) are from SciTSR, (c-d) are from PubTabNet, (e-f) are cropped from cTDaR TrackB2-Modern.

	Search Search					and the second second											
			52		a so of heaters	Marchine	Comprises	Brar	One of	e m			High Po	-			- East
Int	S 21	MAR INCO	64.9	Contra -	# 04000 milli	110 1	Louistania	No 32	Freder	Square A	2	(Perinter and)	200 Eres	Prin.	Box!	Creat	Fork Albert
ing the second	3 1923	196/7	8.00		1 01.54	IM24 HA	Administration and Area	100		IM.P*		Laid of practs					1 01
las in	N 583	with.	100	tides statistical	1 001	112 13	Annual and a state of the state	2		an.e		Real or sold lines					1 8
-	34 4755.94	March		Bodarstarpard	ы 6.0	440			3	e4		how with make hand					3 F
		741	2'	Erw .	or with the and dense believe	denits in the statemy living						Comprises	(bear	Date	the set	fan d	-
	Sum of Span	EAST TREAT				TE Analysia at your	And the second second second	10 2	1 1	0 1	1		10 YE 31 10 27 K	Max D	FICE INC	1-Sile-Ci	
-			-	la -t	-	Martin	Canada	Bear	Dans a		-	and a second second	Sec. Po				1000
	# 26	14	-		a Samid Square	and the second	-Long Lange	la sy	Tables	-	-	(Perint Ind)	Acri Comp 1308 atta	24	Ref.	Creat	Park Shert
(QAN)	1 10.3	19(1)	100	Seal Street	1 202.34	1454 115	Advant in report to	10		234.7**	_	INCOME.	*1 41		743	1.7	
	8 20.9	3100	910	CAN PROVIDE	661	10 10	Conception of the local division of the loca	-		297		Personal Property in which the real Property is not in the left of				-3	4
	14 495.91	MOLIN		Annal an an an and	M 6.0	440 53					-	Report of the local			1.0		1 1
	1 I.	211	2	Ere Colorador Califo	as with the and denor interve	denirs in the average to value			-		-	Comprises	Bray	Dam	bu and	fand	-
	Sum of Span	SEE MODE				12 Analysis strategy	And the second second	10 2	1 1	0 11	1.		10 H 10 H 10	Net	PICE ME	1-mark	
-						Mass Survey	Constant	Brier	Descri		-		High Po				T-10
	# 21	14	10.00	in a constant	# 268.0 March	1000	Los de ma	No.24		Spon S	-	Comparison Orbectment Intellight	Box Com 1901 574	Putt	Boal		Park Mari
10000	3 (0.3)	100.15	22.2	ALC: Y	1 1119	114.54	And in case of	12	-	114.7*	-	Loss of particular			144	1.1	
	8 643	32140	908	TANK (story)	641	10 10	Bull St. (20)			inc	4	Difference and Real					1 1
-	19 275.91	301.00		Acceleration Accel	H 4111	185 18				44 - 7		Administration in column					11-1
1000		940	2	100	at with the confidence improve	I limits in the average & case			-			Canada	200	-			

Figure 10: Qualitative results on the in-house dataset. 1^{st} row: original images; 2^{nd} row: results from SPLERGE [17]; 3^{rd} row: results from our table structure recognizer.

5.4.2. Ablation study

Influence of kernel width in spatial CNN modules. We investigate the influence of the kernel width in the four spatial CNN modules to TSR accuracies. The experimental results on the manually labeled cTDaR modern subset are shown in Table 9. Here, the kernel width determines the number of pixels from which a pixel could receive messages directly. Consistent with the observations in [29], increasing the kernel width improves the performance up to a saturation point (k = 9), and then the performance slightly decreases. Therefore, we set the kernel width as 9 for all the other experiments.

Effectiveness of spatial CNN based separation line

Г	able 9: Ablation study	of the	kernel	width	for spati	al CNN	module	s
	Kernel Width (k)	1	3	5	7	9	11	
	WAvg. F1-score (%)	93.9	94.8	8 95.	3 95.8	3 96.0	95.7	

prediction. We compare our spatial CNN based message passing method with two previously used methods in the TSR field, i.e., projection networks [17] and Bi-GRU [18]. Moreover, as self-attention operations are also known to be good at aggregating global context information, we also select a representative one, i.e., criss-cross attention [84], for comparison. All models are trained with the same hyper-parameters for fair comparison and tested on our challenging in-house dataset. We have also implemented a baseline model, i.e., removing the spatial CNN modules directly from our TSR model. The quantitative results of these variants are given in Table 10 and some qualitative results are shown in Fig. 11. The experimental results show that the performance of other message passing methods are obviously inferior to our spatial CNN based method, especially for tables with large blank spaces or curved tables, which can demonstrate the effectiveness of our spatial CNN based separation line prediction method.

Effectiveness of Grid CNN based cell merging. We further compare the proposed Grid CNN based cell merging method with other two visual relationship prediction based methods [85], which are based on relation network and GCN respectively, on the in-house dataset to demonstrate the effectiveness of Grid CNN for cell merging. The experimental results are listed in Table 11, from which we can find that the cell merging module can significantly improve the performance of our TSR model (90.9% vs. 94.6%). Moreover, the

Table 10: Comparison of different message passing methods.

Message Passing	Iol	J@0.6(%)	Iol	J@0.7((%)	Iol	J@0.8((%)	Iol	IoU@0.9(%)			
Methods	Р	R	F1	Р	R	F1	Р	R	F1	Р	R	F1	F1(%)	
No message passing	92.9	92.2	92.5	92.8	92.1	92.4	92.7	92.0	92.4	92.7	91.9	92.3	92.4	
Projection Networks [17]	93.8	92.6	93.2	93.7	92.5	93.1	93.6	92.5	93.0	93.5	92.4	93.0	93.0	
Bi-GRU (2 layers) [18]	93.7	92.7	93.2	93.7	92.6	93.1	93.6	92.5	93.1	93.6	92.5	93.0	93.1	
CC Attention [84]	94.1	93.7	93.9	94.1	93.6	93.8	94.0	93.5	93.8	94.0	93.5	93.7	93.8	
Spatial CNN (Proposed)	94.9	94.5	94.7	94.8	94.4	94.6	94.8	94.4	94.6	94.7	94.3	94.5	94.6	

Table 11: Comparison of different cell merging methods. IoU@0.8(%) Cell Merging IoU@0.6(%) IoU@0.7(%) IoU@0.9(%) WAvg. Methods P F1(%) Ρ R F1 P R F1 R F1 Р R F1 No cell merging 91.1 91.0 90.9 90.2 90.9 90.9 91.7 90.5 91.6 90.4 91.5 90.3 91.5 Relation Network [85] 93.5 93.1 93.3 93.4 93.0 93.2 93.3 93.0 93.1 93.3 92.9 93.1 93.2 94.2 GCN [85] 94.0 94.1 94.1 93.9 94.0 94.1 93.8 94.0 94.0 93.8 93.9 94.0 Grid CNN (Proposed) 94.9 94.5 94.7 94.8 94.4 94.6 94.8 94.4 94.6 94.7 94.3 94.5 94.6



Figure 11: Some comparison examples from different message passing methods for separation line prediction.

last three rows show that the proposed Grid CNN based cell merging method is more effective than the relation network based (93.2% vs. 94.6%) and the GCN based (94.0% vs. 94.6%) methods. Based on our observations, due to the grid arrangement of cell features, Grid CNN can leverage context information effectively with several stacked convolution layers to improve cell merging accuracy, leading to improved robustness to tables with hierarchical spanning cells.

5.5. Limitations of our approach

Although the proposed RobusTabNet shows superior capability in most scenarios as demonstrated in the previous experiments, it still has some limitations. For example, our current table detector still struggles with nearby tables, and our table structure recognizer is not robust enough to cells with multi-line contents. Some failure examples are presented in Fig. 12. Furthermore,



Figure 12: Some typical failure cases, including the detection of nearby tables and the structure recognition of cells with multi-line contents.

our TSR approach will fail on some extremely dense tables, because the predicted segmentation masks of nearby separation lines could be overlapped. Note that these difficulties are common challenges for other stateof-the-art methods. Finding effective solutions to these problems will be our future work. Moreover, since the tables in existing datasets are mostly with black lines/letters and white backgrounds, the effectiveness and generalization ability of our approach on tables with different types of backgrounds, text fonts and line colors need to be studied in the future.

6. Conclusion and future work

In this paper, we introduce a new table detection and structure recognition approach named RobusTabNet to extract tables from heterogeneous document images. For table detection, we use CornerNet as a new region proposal network for Faster R-CNN, which can leverage more precise corner points generated from heatmaps to improve table localization accuracy. For table structure recognition, we propose two effective techniques to significantly improve the capability of the split-andmerge paradigm, i.e., spatial CNN based separation line prediction and Grid CNN based cell merging. As the spatial CNN can effectively propagate contextual information across the whole table image, improved robustness can be achieved to tables with large blank spaces and curved tables. Moreover, as the whole table is compactly represented as a grid, a simple but effective Grid CNN can be used to achieve excellent cell merging accuracy. Consequently, the proposed RobusTabNet has achieved state-of-the-art performance on both table detection (cTDaR TrackA, PubLayNet and IIIT-AR-13K) and structure recognition (SciTSR, PubTabNet and cTDaR TrackB2-Modern) public benchmarks. We have further validated the robustness of our approach to tables with complex structures, large blank spaces, as well as distorted or even curved shapes on a more challenging in-house dataset.

For future work, we will study how to leverage header analysis techniques to disambiguate nearby tables. Furthermore, we will also explore how to incorporate textual information into our Grid CNN module to improve the robustness of our table structure recognizer to cells with multi-line contents. To achieve more robust structure recognition of dense tables, we will study effective technologies for adaptive scaling. As for latency reduction, we will explore an end-to-end solution for table extraction, where the table detector and the table structure recognizer can share a same backbone network.

References

- Y. Liu, K. Bai, P. Mitra, C. L. Giles, Tableseer: Automatic table metadata extraction and searching in digital libraries, in: Proceedings of the 7th ACM/IEEE-CS Joint Conference on Digital Libraries, 2007, pp. 91–100.
- [2] H. Sun, H. Ma, X. He, W.-t. Yih, Y. Su, X. Yan, Table cell search for question answering, in: Proceedings of the International Conference on World Wide Web, 2016, pp. 771–782.
- [3] K. A. Hashmi, M. Liwicki, D. Stricker, M. A. Afzal, M. A. Afzal, M. Z. Afzal, Current status and performance analysis of table recognition in document images with deep neural networks, IEEE Access 9 (2021) 87663–87685.
- [4] L. Hao, L. Gao, X. Yi, Z. Tang, A table detection method for pdf documents based on convolutional neural networks, in: Proceedings of the IAPR International Workshop on Document Analysis Systems, IEEE, 2016, pp. 287–292.
- [5] N. D. Vo, K. Nguyen, T. V. Nguyen, K. Nguyen, Ensemble of deep object detectors for page object detection, in: Proceedings of the International Conference on Ubiquitous Information Management and Communication, 2018, pp. 1–6.
- [6] A. Gilani, S. R. Qasim, I. Malik, F. Shafait, Table detection using deep learning, in: Proceedings of the International

Conference on Document Analysis and Recognition, Vol. 1, IEEE, 2017, pp. 771–776.

- [7] Y. Huang, Q. Yan, Y. Li, Y. Chen, X. Wang, L. Gao, Z. Tang, A yolo-based table detection method, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 813–818.
- [8] X. Zheng, D. Burdick, L. Popa, X. Zhong, N. X. R. Wang, Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context, in: Proceedings of the IEEE Winter conference on Applications of Computer Vision, 2021, pp. 697–706.
- [9] R. Saha, A. Mondal, C. Jawahar, Graphical object detection in document images, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 51–58.
- [10] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, K. Sultanpure, Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 572–573.
- [11] M. Agarwal, A. Mondal, C. Jawahar, Cdec-net: Composite deformable cascade network for table detection in document images, in: Proceedings of the IEEE International Conference on Pattern Recognition, IEEE, 2021, pp. 9491–9498.
- [12] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
- [13] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
- [14] Z. Cai, N. Vasconcelos, Cascade r-cnn: High quality object detection and instance segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (5) (2019) 1483– 1498.
- [15] S. Schreiber, S. Agne, I. Wolf, A. Dengel, S. Ahmed, Deepdesrt: Deep learning for detection and structure recognition of tables in document images, in: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 1, IEEE, 2017, pp. 1162–1167.
- [16] S. A. Siddiqui, P. I. Khan, A. Dengel, S. Ahmed, Rethinking semantic segmentation for table structure recognition in documents, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 1397– 1402.
- [17] C. Tensmeyer, V. I. Morariu, B. Price, S. Cohen, T. Martinez, Deep splitting and merging for table structure decomposition, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 114–121.
- [18] S. A. Khan, S. M. D. Khalid, M. A. Shahzad, F. Shafait, Table structure extraction with bi-directional gated recurrent unit networks, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 1366– 1371.
- [19] S. A. Siddiqui, I. A. Fateh, S. T. R. Rizvi, A. Dengel, S. Ahmed, Deeptabstr: Deep learning based table structure recognition, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 1403–1409.
- [20] K. A. Hashmi, D. Stricker, M. Liwicki, M. N. Afzal, M. Z. Afzal, Guided table structure recognition through anchor optimization, IEEE Access 9 (2021) 113521–113534.
- [21] S. R. Qasim, H. Mahmood, F. Shafait, Rethinking table recognition using graph neural networks, in: Proceedings of the International Conference on Document Analysis and Recognition, 2019, pp. 142–147.

- [22] S. Raja, A. Mondal, C. Jawahar, Table structure recognition using top-down and bottom-up cues, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 70–86.
- [23] L. Qiao, Z. Li, Z. Cheng, P. Zhang, S. Pu, Y. Niu, W. Ren, W. Tan, F. Wu, Lgpma: Complicated table structure recognition with local and global pyramid mask alignment, in: Proceedings of the International Conference on Document Analysis and Recognition, Springer, 2021, pp. 99–114.
- [24] X.-H. Li, F. Yin, X.-Y. Zhang, C.-L. Liu, Adaptive scaling for archival table structure recognition, in: Proceedings of the International Conference on Document Analysis and Recognition, Springer, 2021, pp. 80–95.
- [25] Z. Chi, H. Huang, H.-D. Xu, H. Yu, W. Yin, X.-L. Mao, Complicated table structure recognition, arXiv preprint arXiv:1908.04729.
- [26] X. Zhong, E. ShafieiBavani, A. Jimeno Yepes, Image-based table recognition: Data, model, and evaluation, in: Proceedings of the European Conference on Computer Vision, Springer, 2020, pp. 564–580.
- [27] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 734–750.
- [28] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- [29] X. Pan, J. Shi, P. Luo, X. Wang, X. Tang, Spatial as deep: Spatial cnn for traffic scene understanding, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
- [30] L. Gao, Y. Huang, H. Déjean, J.-L. Meunier, Q. Yan, Y. Fang, F. Kleber, E. Lang, Icdar 2019 competition on table detection and recognition (ctdar), in: Proceedings of the International Conference on Document Analysis and Recognition, 2019, pp. 1510–1515.
- [31] X. Zhong, J. Tang, A. Jimeno Yepes, Publaynet: Largest dataset ever for document layout analysis, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 1015–1022.
- [32] A. Mondal, P. Lipps, C. Jawahar, Iiit-ar-13k: A new dataset for graphical object detection in documents, in: Proceedings of the IAPR International Workshop on Document Analysis Systems, Springer, 2020, pp. 216–230.
- [33] T. Kieninger, A. Dengel, The t-recs table recognition and analysis system, in: Proceedings of the IAPR International Workshop on Document Analysis Systems, Springer, 1998, pp. 255–270.
- [34] B. Gatos, D. Danatsas, I. Pratikakis, S. J. Perantonis, Automatic table detection in document images, in: Proceedings of the IEEE International Conference on Pattern Recognition and Image Analysis, Springer, 2005, pp. 609–618.
- [35] T. Hassan, R. Baumgartner, Table recognition and understanding from pdf files, in: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 2, IEEE, 2007, pp. 1143–1147.
- [36] T. T. Anh, N. In-Seop, K. Soo-Hyung, A hybrid method for table detection from document image, in: Proceedings of the IAPR Asian Conference on Pattern Recognition, IEEE, 2015, pp. 131– 135.
- [37] S. Tupaj, Z. Shi, C. H. Chang, H. Alam, Extracting tabular information from text files, EECS Department, Tufts University, Medford, USA 1.
- [38] G. Harit, A. Bansal, Table detection in document images using header and trailer patterns, in: Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, 2012, pp. 1–8.

- [39] Y. Wang, I. T. Phillipst, R. Haralick, Automatic table ground truth generation and a background-analysis-based table structure extraction method, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2001, pp. 528–532.
- [40] R. Zanibbi, D. Blostein, J. Cordy, A survey of table recognition: Models, observations, transformations, and inferences, International Journal of Document Analysis and Recognition 7 (1) (2004) 1–16.
- [41] D. W. Embley, M. Hurst, D. Lopresti, G. Nagy, Tableprocessing paradigms: A research survey, International Journal of Document Analysis and Recognition 8 (2) (2006) 66–86.
- [42] F. Cesarini, S. Marinai, L. Sarti, G. Soda, Trainable table location in document images, in: Proceedings of the IEEE International Conference on Pattern Recognition, Vol. 3, IEEE, 2002, pp. 236–240.
- [43] A. C. e. Silva, Learning rich hidden markov models in document analysis: Table location, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2009, pp. 843–847.
- [44] A. C. e. Silva, A. M. Jorge, L. Torgo, Design of an end-to-end method to extract information from tables, International Journal of Document Analysis and Recognition 8 (2) (2006) 144–171.
- [45] X. Yi, L. Gao, Y. Liao, X. Zhang, R. Liu, Z. Jiang, Cnn based page object detection in document images, in: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 1, IEEE, 2017, pp. 230–235.
- [46] D. A. B. Oliveira, M. P. Viana, Fast cnn-based document layout analysis, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1173–1180.
- [47] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
- [48] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440– 1448.
- [49] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
- [50] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980– 2988.
- [51] S. Arif, F. Shafait, Table detection in document images using foreground and background features, in: DICTA, IEEE, 2018, pp. 1–8.
- [52] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, S. Ahmed, Decnt: Deep deformable cnn for table detection, IEEE Access 6 (2018) 74151–74161.
- [53] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 764–773.
- [54] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, H. Ling, Cbnet: A novel composite backbone network architecture for object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11653–11660.
- [55] M. Göbel, T. Hassan, E. Oro, G. Orsi, Icdar 2013 table competition, in: Proceedings of the International Conference on Document Analysis and Recognition, 2013, pp. 1449–1453.
- [56] L. Gao, X. Yi, Z. Jiang, L. Hao, Z. Tang, Icdar2017 competition on page object detection, in: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 1,

IEEE, 2017, pp. 1417-1422.

- [57] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Z. Li, Tablebank: Table benchmark for image-based table detection and recognition, in: Proceedings of the Language Resources and Evaluation Conference, 2020, pp. 1918–1925.
- [58] N. Sun, Y. Zhu, X. Hu, Faster r-cnn based table detection combining corner locating, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 1314–1319.
- [59] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, C. Lee Giles, Learning to extract semantic structure from documents using multimodal fully convolutional neural networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 5315–5324.
- [60] D. He, S. Cohen, B. Price, D. Kifer, C. L. Giles, Multiscale multi-task fcn for semantic page segmentation and table detection, in: Proceedings of the International Conference on Document Analysis and Recognition, Vol. 1, IEEE, 2017, pp. 254–261.
- [61] I. Kavasidis, C. Pino, S. Palazzo, F. Rundo, D. Giordano, P. Messina, C. Spampinato, A saliency-based convolutional neural network for table and chart detection in digitized documents, in: Proceedings of the International Conference on Image Analysis and Processing, Springer, 2019, pp. 292–302.
- [62] S. S. Paliwal, D. Vishwanath, R. Rahul, M. Sharma, L. Vig, Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 128–133.
- [63] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
- [64] X.-H. Li, F. Yin, C.-L. Liu, Page object detection from pdf document images by deep structured prediction and supervised clustering, in: Proceedings of the IEEE International Conference on Pattern Recognition, IEEE, 2018, pp. 3627– 3632.
- [65] P. Riba, L. Goldmann, O. R. Terrades, D. Rusticus, A. Fornés, J. Lladós, Table detection in business document images by message passing networks, Pattern Recognition 127 (2022) 108641.
- [66] M. Holeček, A. Hoskovec, P. Baudiš, P. Klinger, Table understanding in structured documents, in: Proceedings of the International Conference on Document Analysis and Recognition Workshops, Vol. 5, IEEE, 2019, pp. 158–164.
- [67] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, M. Zhou, Docbank: A benchmark dataset for document layout analysis, in: Proceedings of the International Conference on Computational Linguistics, 2020, pp. 949–960.
- [68] A. Laurentini, P. Viada, Identifying and understanding tabular material in compound documents, in: Proceedings of the IEEE International Conference on Pattern Recognition, IEEE COMPUTER SOCIETY PRESS, 1992, pp. 405–405.
- [69] K. Itonori, Table structure recognition based on textblock arrangement and ruled line position, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 1993, pp. 765–768.
- [70] A. Shigarov, A. Mikhailov, A. Altaev, Configurable table structure recognition in untagged pdf documents, in: Proceedings of the 2016 ACM Symposium on Document Engineering, 2016, pp. 119–122.
- [71] R. Rastan, H.-Y. Paik, J. Shepherd, Texus: A unified framework for extracting and understanding tables in pdf documents, Information Processing & Management 56 (3) (2019) 895–918.

- [72] Y. Wang, I. T. Phillips, R. M. Haralick, Table structure understanding and its performance evaluation, Pattern recognition 37 (7) (2004) 1479–1497.
- [73] Y. Zou, J. Ma, A deep semantic segmentation model for imagebased table structure recognition, in: Proceedings of the IEEE International Conference on Signal Processing, Vol. 1, IEEE, 2020, pp. 274–280.
- [74] Y. Deng, D. Rosenberg, G. Mann, Challenges in endto-end neural scientific table recognition, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 894–901.
- [75] Y. Li, Z. Huang, J. Yan, Y. Zhou, F. Ye, X. Liu, Gfte: Graphbased financial table extraction, in: Proceedings of the IEEE International Conference on Pattern Recognition, Springer, 2021, pp. 644–658.
- [76] W. Xue, Q. Li, D. Tao, Res2tim: Reconstruct syntactic structures from table images, in: Proceedings of the International Conference on Document Analysis and Recognition, IEEE, 2019, pp. 749–755.
- [77] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 6569–6578.
- [78] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 2117–2125.
- [79] G. Bradski, The opencv library, Dr. Dobb's Journal: Software Tools for the Professional Programmer 25 (11) (2000) 120–123.
- [80] J. Zhang, M. Elhoseiny, S. Cohen, W. Chang, A. Elgammal, Relationship proposal networks, in: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 5678–5686.
- [81] Z. Zhong, L. Sun, Q. Huo, An anchor-free region proposal network for faster r-cnn-based text detection approaches, International Journal on Document Analysis and Recognition 22 (3) (2019) 315–327.
- [82] Z. Zhang, J. Zhang, J. Du, F. Wang, Split, embed and merge: An accurate table structure recognizer, Pattern Recognition 126 (2022) 108565.
- [83] A. Jimeno Yepes, P. Zhong, D. Burdick, Icdar 2021 competition on scientific literature parsing, in: Proceedings of the International Conference on Document Analysis and Recognition, Springer, 2021, pp. 605–617.
- [84] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, W. Liu, Ccnet: Criss-cross attention for semantic segmentation, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 603–612.
- [85] C. Ma, L. Sun, Z. Zhong, Q. Huo, Relatext: Exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks, Pattern Recognition 111 (2021) 107684.