LGPMA: Complicated Table Structure Recognition with Local and Global Pyramid Mask Alignment[*]

Liang Qiao[1], Zaisheng Li[1], Zhanzhan Cheng[2,1], Peng Zhang[1], Shiliang Pu[1], Yi Niu[1], Wenqi Ren[1], Wenming Tan[1], and Fei Wu[2]

[1] Hikvision Research Institute, China
{qiaoliang6, lizaisheng, chengzhanzhan, zhangpeng23, pushiliang.hri, niuyi, renwenqi, tanwenming}@hikvision.com
[2] Zhejiang University, China
wufei@cs.zju.edu.cn

Abstract. Table structure recognition is a challenging task due to the various structures and complicated cell spanning relations. Previous methods handled the problem starting from elements in different granularities (rows/columns, text regions), which somehow fell into the issues like lossy heuristic rules or neglect of empty cell division. Based on table structure characteristics, we find that obtaining the aligned bounding boxes of text region can effectively maintain the entire relevant range of different cells. However, the aligned bounding boxes are hard to be accurately predicted due to the visual ambiguities. In this paper, we aim to obtain more reliable aligned bounding boxes by fully utilizing the visual information from both text regions in proposed local features and cell relations in global features. Specifically, we propose the framework of *Local and Global Pyramid Mask Alignment*, which adopts the soft pyramid mask learning mechanism in both the local and global feature maps. It allows the predicted boundaries of bounding boxes to break through the limitation of original proposals. A pyramid mask re-scoring module is then integrated to compromise the local and global information and refine the predicted boundaries. Finally, we propose a robust table structure recovery pipeline to obtain the final structure, in which we also effectively solve the problems of empty cells locating and division. Experimental results show that the proposed method achieves competitive and even new state-of-the-art performance on several public benchmarks.

Keywords: Table Structure Recognition · Aligned Bounding Box · Empty Cell.

1 Introduction

Table is one of the rich-information data formats in many real documents like financial statements, scientific literature, purchasing lists, etc. Besides the text
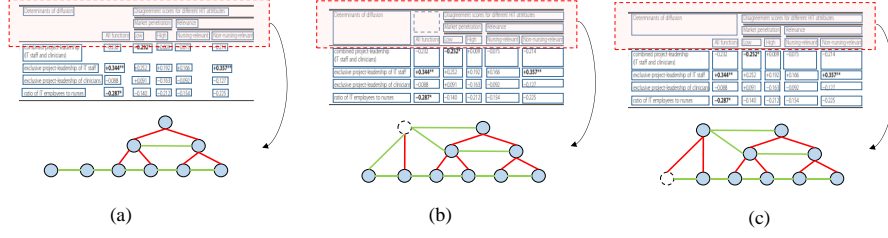

Fig. 1. (a) The visualized results without considering empty cells. (b) The ground-truth of aligned bounding boxes and nodes relations. (c) A false example due to the ambiguity between the empty cell and cross-column cell. The cells and their relations are represented as nodes and connected lines (red: vertical, green: horizontal). Empty cells are displayed in dashed circles.

content, the table structure is vital for people to do the key information extraction. Thus, table structure recognition [10,21,34,30,5,4,39] becomes one of the important techniques in current document understanding systems.

From the global perspective, early table structure recognition processes usually depend on the detection of the grid's boundaries [19,18]. However, these methods can not handle tables without grid boundaries, such as three-line tables. Though recent works [30,22,32,31] attempt to predict row/column regions or even invisible grid lines [33], they are limited to handle tables that cross span multiple rows/columns. The row/column splitting operation might also cut cells that contain text in multiple lines.

Another group of methods solves the above problems in a bottom-up way to firstly detect the text blocks' positions and then recover the bounding-boxes' relations by heuristic rules [38] or GNN(Graph Neural Networks) [29,14,2,24,26]. However, rules designed based on bounding boxes of text regions are vulnerable to handling complicated matching situations. GNN-based methods not only bring extra network cost but also depend on more expensive training cost such as the data volume. Another issue is that these methods are difficult to obtain the empty cells because they usually fall into the visual ambiguity problem with the cross-row/column cells. The prediction of empty cells directly affects the correctness of table structure, as illustrated in Figure 1(a). Moreover, how to split or merge these empty regions is still a challenging problem that cannot be neglected, because different division results will generate different editable areas when the image is transferred into digital format.

Notice that the structure of the table itself is a human-made rule-based data form. Under the situation that tables are without visual rotation or perspective transformation, if we could obtain all of the perfect aligned cell regions rather than the text regions [26], the structure inference will be easy and almost lossless, as illustrated in Figure 1(b). Nevertheless, acquiring such information is not

easy. On the one hand, the annotations of text regions [5,2,39] are much easier to get than cell regions. On the other hand, the aligned boxes are difficult to be accurately learned since there is usually no visible texture of boundaries at the region's periphery. Multi-row/column cells are easy to be confused with the empty cell regions. For example, in Figure 1(c), the network usually falls into the situation that the predicted aligned boxes are not large enough and results in the wrong cell matching. Although [26] designs an alignment loss to assist the bounding boxes learning, it only considers the relative relations between boxes and fails to capture the cell's absolute coverage area.

In this paper, we aim to train the network to obtain more reliable aligned cell regions and solve the problems of empty cell generation and partition in one model. Observing that people perceive visual information from both local text region and global layout when they read, we propose a uniform table structure recognition framework to compromise the benefits from both local and global information, called LGPMA (Local and Global Pyramid Mask Alignment) Network. Specifically, the model simultaneously learns a local Mask-RCNN-based [6] aligned bounding boxes detection task and a global segmentation task. In both tasks, we adopt the pyramid soft mask supervision [17] to help obtain more accurate aligned bounding boxes. In LGPMA, the local branch (LPMA) acquires more reliable text region information through visible texture perceptron, while the global branch (GPMA) can learn more legible spatial information of cells' range or division. The two branches help the network learn better-fused features via jointly learning and effectively refine the detected aligned bounding boxes through a proposed mask re-scoring strategy. Based on the refined results, we design a robust and straightforward table structure recovery pipeline, which can effectively locate empty cells and precisely merge them according to the guidance of global segmentation.

The major contributions of this paper are as follows: (1) We propose a novel framework called LGPMA Network that compromises the visual features from both local and global perspectives. The model makes full use of the information from the local and global features through a proposed mask re-scoring strategy, which can obtain more reliable aligned cell regions. (2) We introduce a uniform table structure recovering pipeline, including cell matching, empty cell searching, and empty cell merging. Both non-empty cells and empty cells can be located and split efficaciously. (3) Extensive experiments show that our method achieves competitive and even state-of-the-art results on several popular benchmarks.

2   Related Works

Traditional table recognition researches mainly worked with hand-crafted features and heuristic rules [8,10,34,19,18,3]. These methods are mostly applied to simple table structures or specific data formats, such as PDFs. The early techniques about table detection and recognition can be found in the comprehensive survey [37]. With the great success of deep neural network in computer vision field, works began to focus on the image-based table with more general

structures [21,30,24,9,13,36,23,14,33]. According to the basic components granularities, we roughly divide previous methods into two types: global-object-based methods and local-object-based methods.

Global-object-based methods mainly focus on the characteristics of global table components and mostly started from row/column or grid boundaries detection. Works of [30,31,32] firstly obtain the rows and columns regions using the detection or segmentation models and then intersect these two regions to obtain the grids of cells. [22] handles the table detection and table recognition tasks in an end-to-end manner by the table region mask learning and table's row/column mask learning. [33] detects the rows and columns by learning the interval areas' segmentation between rows/columns and then predicting the indicator to merge the separated cells.

There also exist some methods [13,39] that directly perceive the whole image information and output table structures as text sequence in an encoder-decoder framework. Although these methods look graceful and entirely avoid human being involved, the models are usually challenging to be trained and rely on a large amount of training data. Global-object-based methods usually have difficulties in handling various complicated table structures, such as cells spanning multiple rows/columns or containing text in multi-lines.

Local-object-based methods begin from the smallest fundamental element, cells. Given the cell-level text region annotation, the text detection task is relatively easy to finish by the general detection methods like Yolo [27], Faster R-CNN [28], etc. After that, a group of methods [36,23,38] tries to recover the cell relations based on some heuristic rules and algorithms. Another type of methods [11,2,14,24,26] treat the detected boxes as nodes in a graph and attempt to predict the relations based on techniques of Graph Neural Networks [29]. [14] predicts the relations between nodes in three classes (the horizontal connection, the vertical connection, no connection) using several features such as visual features, text positions, word embedding, etc. [2] adopts graph attention mechanism to enhance the predicting accuracy. [24] alleviates the problem of large graph nodes numbers by the pair sampling strategy. The above three works [14,2,24] also published new table datasets for this research area. Since there is no empty cell detected, local-object based-methods usually fall into empty cell ambiguity.

In this paper, we try to compromise the advantages of both global and local features. Based on the local detection results, we integrate the global information to refine the detected bounding boxes and provide a straightforward guide for empty cell division.

3   Methodology

3.1   Overview

We propose the model LGPMA, whose overall workflow is shown in Figure 2.

The model is built based on the existing Mask-RCNN [6]. The bounding box branch directly learns the detection task of aligned bounding boxes for non-
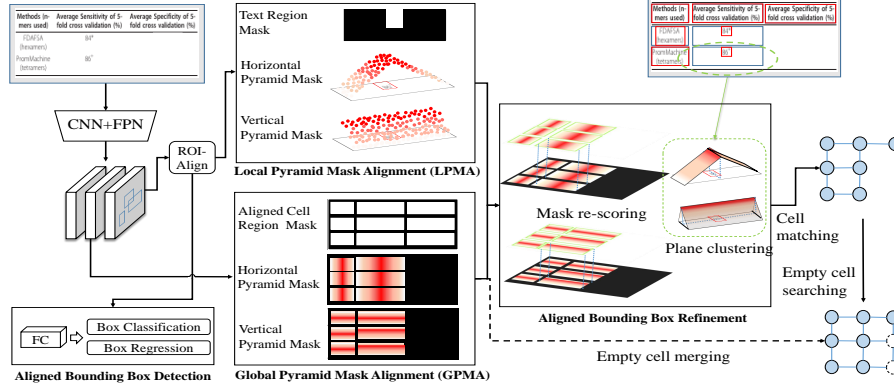
Fig. 2. The workflow of LGPMA. The network simultaneously learns a local aligned bounding boxes detection task (LPMA) and a global segmentation task (GPMA). We adopt the pyramid mask learning mechanisms in both branches and use a mask re-scoring strategy to refine the predicted bounding boxes. Finally, the table structure can be uniformly recovered by a pipeline, including cell matching, empty cell searching, and empty cell merging.

empty cells. The network simultaneously learns a Local Pyramid Mask Alignment (LPMA) task based on the local feature extracted by the RoI-Align operation and a Global Pyramid Mask Alignment (GPMA) task based on the global feature map.

In LPMA, in addition to the binary segmentation task that learns the text region mask, the network is also trained with the pyramid soft mask supervision in both horizontal and vertical directions.

In GPMA, the network learns a global pyramid mask for all aligned bounding boxes of non-empty cells. To obtain more information about empty cell splitting, the network also learns the global binary segmentation task that considers both non-empty and empty cells.

A pyramid mask re-scoring module is then adopted to refine the predicted pyramid labels. The accurate aligned bounding boxes can be obtained by the process of plane clustering. Finally, a uniform structure recovering pipeline containing cell matching, empty cell searching, empty cell merging is integrated to obtain the final table structure.

3.2 Aligned Bounding Box Detection

The difficulty of accurate text region matching mainly comes from the covered range gap between text regions and the real cell regions. Real cell regions may contain empty spaces for row/column alignment, especially for those cells crossing span multiple rows/columns. Inspired by [26,36], with the annotations of text
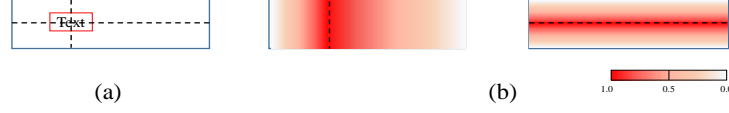
Fig. 3. (a) shows the original aligned bounding box (blue) and text region box (red). (b) shows the pyramid mask labels in horizontal and vertical direction, respectively.

regions and row/column indices, we can easily generate the aligned bounding box annotations according to the maximum box height/width in each row/column. The regions of aligned bounding boxes approximately equal to that of real cells. For the table images in print format and without visual rotation or perspective transformation, if we could obtain the aligned cell regions and assume there is no empty cell, it is easy to infer the cell relations according to the coordinate overlapping information in horizontal and vertical directions.

We adopt Mask-RCNN [6] as the base model. In the bounding box branch, the network is trained based on the aligned bounding box supervision. However, the aligned bounding box learning is not easy because cells are easy to be confused with empty regions. Motivated by the advanced pyramid mask text detector [17], we find that using the soft-label segmentation may break through the proposed bounding box's limitation and provide more accurate aligned bounding boxes. To fully utilize the visual features from both local texture and global layout, we propose to learn the pyramid mask alignment information in these two folds simultaneously.

3.3   Local Pyramid Mask Alignment

In the mask branch, the model is trained to learn both a binary segmentation task and a pyramid mask regression task, which we call Local Pyramid Mask Alignment (LPMA).

The binary segmentation task is the same as the original model, in which only the text region is labeled as 1 and others are labeled as 0. The detected mask regions can be used in the following text recognition task.

For the pyramid mask regression, we assign the pixels in the proposal bounding box regions with the soft-label in both horizontal and vertical directions, as shown in Figure 3. The middle point of text will have the largest regressed target 1. Specifically, we assume the proposed aligned bounding box has the shape of $H \times W$. The top-left point and bottom right point of the text region are denoted as $\{(x_1, y_1), (x_2, y_2)\}$, respectively, where $0 \leq x_1 < x_2 \leq W$ and $0 \leq y_1 < y_2 \leq H$. Therefore, the target of the pyramid mask is in shape $\mathbb{R}^{2 \times H \times W} \in [0, 1]$, in which the two channels represent the target map of the horizontal mask and vertical mask, respectively. For every pixel (h, w), these two targets can be formed as:

$$t_h^{(w,h)} = \begin{cases} w/x_1 & w \leq x_{mid} \\ \frac{W-w}{W-x_2} & w > x_{mid} \end{cases}, \quad t_v^{(w,h)} = \begin{cases} h/y_1 & h \leq y_{mid} \\ \frac{H-h}{H-y_2} & h > y_{mid} \end{cases}, \tag{1}$$

where $0{\leq}w{<}W$, $0{\leq}h{<}H$, and $x_{mid}{=}\frac{x_1+x_2}{2}$, $y_{mid} = \frac{y_1+y_2}{2}$. In this way, every pixel in the proposal region takes part in predicting the boundaries.

3.4   Global Pyramid Mask Alignment

Although LPMA allows the predicted mask to break through the proposal bounding boxes, the local region's receptive fields are limited. To determine the accurate coverage area of a cell, the global feature might also provide some visual clues. Inspired by [40,25], learning the offsets of each pixel from a global view could help locate more accurate boundaries. However, bounding boxes in cell-level might be varied in width-height ratios, which leads to the unbalance problem in regression learning. Therefore, we use the pyramid labels as the regressing targets for each pixel, named Global Pyramid Mask Alignment (GPMA).

Like LPMA, the GPMA learns two tasks simultaneously: a global segmentation task and a global pyramid mask regression task. In the global segmentation task, we directly segment all aligned cells, including non-empty and empty cells. The ground-truth of empty cells are generated according to the maximum height/width of the non-empty cells in the same row/column. Notice that only this task learns empty cell division information since empty cells don't have visible text texture that might influence the region proposal networks to some extent. We want the model to capture the most reasonable cell division pattern during the global boundary segmentation according to the human's reading habit, which is reflected by the manually labeled annotations. For the global pyramid mask regression, since only the text region could provide the information of distinct 'mountain top,' all non-empty cells will be assigned with the soft labels similar to LPMA. All of the ground-truths of aligned bounding boxes in GPMA will be shrunk by 5% to prevent boxes from overlapping.

3.5   Optimization

The proposed network is trained end-to-end with multiple optimization tasks. The global optimization can be written as:

$$\mathcal{L} = \mathcal{L}_{rpn} + \lambda_1(\mathcal{L}_{cls} + \mathcal{L}_{box}) + \lambda_2(\mathcal{L}_{mask} + \mathcal{L}_{LPMA}) + \lambda_3(\mathcal{L}_{seg} + \mathcal{L}_{GPMA}), \quad (2)$$

where $\mathcal{L}_{rpn}, \mathcal{L}_{cls}, \mathcal{L}_{box}, \mathcal{L}_{mask}$ are the same losses with that of Mask-RCNN, which represent the region proposal network loss, the bounding box classification loss, the bounding boxes regression loss and the segmentation loss of mask in proposals, respectively. $\mathcal{L}_{seg}$ is the global binary segmentation loss that is implemented in Dice coefficient loss [20], $\mathcal{L}_{LPMA}$ and $\mathcal{L}_{GPMA}$ are the pyramid label regression losses which are optimized by pixel-wise L1 loss. $\lambda_1, \lambda_2, \lambda_3$ are weighted parameters.

3.6   Inference

The inference process can be described in two stages. We first obtain the refined aligned bounding boxes according to the pyramid mask prediction and then generate the final table structure by the proposed structure recovery pipeline.

Aligned Bounding Box Refine. In addition to the benefits generated via joint training, the local and global features also exhibit various advantages in object perceiving [35]. In our setting, we find that local features predict more reliable text region masks, while global prediction can provide more credible long-distance visual information. To compromise both levels' merits, we propose a pyramid mask re-scoring strategy to compromise predictions from LPMA and GPMA. For any proposal region with local pyramid mask prediction, we add the information that comes from the global pyramid mask to adjust these scores. We use some dynamic weights to balance the impacts from LPMA and GPMA.

Specifically, for a predicted aligned bounding box $B=\{(x_1, y_1), (x_2, y_2)\}$, we firstly obtain the bounding box of the text region mask, denoted as $B_t=\{(x'_1, y'_1), (x'_2, y'_2)\}$. Then, we can find a matched connected region $P=\{p_1,p_2,...,p_n\}$ in the global segmentation map, where $p=(x, y)$ represents a pixel. We use $P_o = \{p|x_1 \leq p.x \leq x_2, y_1 \leq p.y \leq y_2, \forall p \in P\}$ to represent the overlap region. Then the predicted pyramid label of point $(x, y) \in P_o$ can be re-scored as follows.

$$F(x) = \begin{cases} \frac{x-x_1}{x_{mid}-x_1}F_{hor}^{(L)}(x,y) + \frac{x_{mid}-x}{x_{mid}-x_1}F_{hor}^{(G)}(x,y) & x_1 \leq x \leq x_{mid} \\ \frac{x-x_2}{x_{mid}-x_2}F_{hor}^{(L)}(x,y) + \frac{x_{mid}-x}{x_{mid}-x_2}F_{hor}^{(G)}(x,y) & x_{mid} < x \leq x_2 \end{cases}, \quad (3)$$

$$F(y) = \begin{cases} \frac{y-y_1}{y_{mid}-y_1}F_{ver}^{(L)}(x,y) + \frac{y_{mid}-y}{y_{mid}-y_1}F_{ver}^{(G)}(x,y) & y_1 \leq y \leq y_{mid} \\ \frac{y-y_2}{y_{mid}-y_2}F_{ver}^{(L)}(x,y) + \frac{y_{mid}-y}{y_{mid}-y_2}F_{ver}^{(G)}(x,y) & y_{mid} < y \leq y_2 \end{cases}, \quad (4)$$

where $x_{mid} = \frac{x'_1+x'_2}{2}, y_{mid} = \frac{y'_1+y'_2}{2}, F_{hor}^{(L)}(x,y), F_{hor}^{(G)}(x,y), F_{ver}^{(L)}(x,y), F_{ver}^{(G)}(x,y)$ are the local horizontal, global horizontal, local vertical and global vertical pyramid label prediction, respectively.

Next, for any proposal region, the horizontal and vertical pyramid mask labels (corresponding to the z-coordinate) can be used to fit two planes in the 3-dimensional space, respectively. All the four planes' intersection lines with the zero plane are the refined boundaries. For example, to refine the right boundary of the aligned box, we select all pixels that $P_r = \{p|x_{mid} \leq p.x \leq x_2, p \in P_o\}$ with the refined pyramid mask prediction $F(x, y)$ to fit the plane. If we formed the plane as $ax + by + c - z = 0$, using the least square method, the problem is equal to minimize the equation of:

$$\min \sum_{y_i=y_1}^{y_2} \sum_{x_i=x_{mid}}^{x_2} (ax_i + by_i + c - F(x_i, y_i))^2, \qquad \forall p = (x_i, y_i) \in P_r. \quad (5)$$

The parameters of a, b, c can be calculated by the matrix as follows:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i y_i & \sum x_i \\ \sum x_i y_i & \sum y_i^2 & \sum y_i \\ \sum x_i & \sum y_i & ||P_o|| \end{bmatrix}^{-1} \begin{pmatrix} x_i F(x_i, y_i) \\ y_i F(x_i, y_i) \\ F(x_i, y_i) \end{pmatrix}, \quad (6)$$

where $||.||$ is the set size. Then we calculate the intersection line between the fitting plane with the plane of $z = 0$. Given that the bounding boxes are axis-
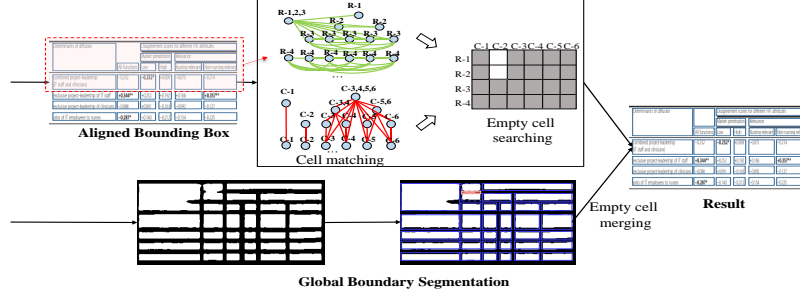
Fig. 4. The illustration of table structure recovery pipeline.

aligned, we calculate the refined x-coordinate as the average value:

$$x_{refine} = -\frac{1}{y_2 - y_1 + 1} \sum_{y_i=y_1}^{y_2} \frac{by_i + c}{a} \tag{7}$$

Similarly, we can obtain the other three refined boundaries. Notice that the refining process can optionally be conducted iteratively refer to [17].

Table Structure Recovery. Based on the refined aligned bounding boxes, the table structure recovery pipeline aims to obtain the final table structure, including three steps: *cell matching, empty cell searching and empty cell merging*, as illustrated in Figure 4.

Cell Matching. In the situation that all of the aligned bounding boxes are axis-aligned, the cells matching process is pretty simple but robust. Following the same naming convention with [14,2,24], the connecting relations can be divided into horizontal and vertical types. The main idea is that if two aligned bounding boxes has enough overlap in x/y-coordinate, we will match them in vertical/horizontal direction. Mathematically, for every two aligned bounding boxes, $\{(x_1, y_1), (x_2, y_2)\}$ and $\{(x_1', y_1'), (x_2', y_2')\}$, they will be horizontally connected if $y_1' \leq \frac{y_1+y_2}{2} \leq y_2'$ or $y_1 \leq \frac{y_1'+y_2'}{2} \leq y_2$. Similarly, they will be vertically connected if $x_1' \leq \frac{x_1+x_2}{2} \leq x_2'$ or $x_1 \leq \frac{x_1'+x_2'}{2} \leq x_2$.

Empty Cell Searching. After obtaining the relations between the detected aligned bounding boxes, we treat them as nodes in a graph, and the connected relations are edges. All of the nodes in the same row/column make up a complete subgraph. Inspired by [24], we adopt the algorithm of Maximum Clique Search [1] to find all maximum cliques in the graph. Take the row searching process as an example, every node that belongs to the same row will be in the same clique. For the cell that crosses span multiple rows, the corresponding node will appear multiple times in different cliques. After sorting these cliques by the average y-coordinate, we can easily label each node with its row index. Nodes that appear

in multiple cliques will be labeled with multiple row indices. We can easily find those vacant positions, which are corresponding to the empty cells.

Empty Cell Merging. By now, we have obtained the empty cells at the smallest level (occupies 1 row and 1 column). To merge these cells more feasibly, we first assign the single empty cells with the aligned bounding box shape as the cell's maximum height/width in the same row/column. Thanks to the visual clues learned by the global segmentation task, we can design the simple merging strategy following the segmentation result. We compute the ratio of pixels that are predicted as 1 in the interval region for every two neighbor empty cells, as the red region illustrated in Figure 4. If the ratio is larger than the preset threshold, we will merge these two cells. As we can see, the empty regions' visual ambiguity always exists, and the segmentation task can hardly be learned perfectly. That is why many segmentation-based methods [24,23,22] struggle with complicated post-processing, such as fracture completion and threshold setting. The proposed method straightforwardly adopts the original visual clue provided by global segmentation and uses pixel voting to obtain a more reliable result.

4   Experiments

4.1   Datasets

We evaluate our proposed framework on following popular benchmarks that contain the annotations of both text content bounding boxes and cell relations.

ICDAR 2013 [5]. This dataset contains 98 training samples and 156 testing samples that cropped from the PDF of government reports.

SciTSR [2]. This dataset contains 12,000 training images and 3,000 testing images cropped from PDF of scientific literature. Authors also select a subset of complicated samples that contains 2,885 training images and 716 testing images, called SciTSR-COMP.

PubTabNet [39]. It is a large-scale complicated table collection that contains 500,777 training images, 9,115 validating images and 9,138 testing images. This dataset contains a large amount of three-lines tables with multi-row/column cells, empty cells, etc.

4.2   Implementation Details

All experiments are implemented in Pytorch with 8×32 GB-Tesla-V100 GPUs. The deep features are extracted and aggregated through the backbone of ResNet-50 [7] with Feature Pyramid Network (FPN) [15]. The weights of the backbone are initialized from the pre-trained model of MS-COCO [16]. In LPMA, the model generates anchors in six different ratios $[1/20, 1/10, 1/5, 1/2, 1, 2]$ for capturing the different shapes of bounding boxes. The Non-Maximal suppression (NMS) IoU threshold of RCNN is 0.1 in the testing phase.

For all benchmarks, the model is trained by the SGD optimizer with batch-size=4, momentum=0.9, and weight-decay=1×10^{-4}. The initial learning ratio

Table 1. Results on ICDAR 2013, SciTSR, SciTSR-COMP datasets. P, R, F1 represent Precision, Recall, F1-Score, respectively. Symbol of † means pre-trained data are used.

Methods	Training Dataset	ICDAR 2013			SciTSR			SciTSR-COMP		
		P	R	F1	P	R	F1	P	R	F1
DeepDeSRT [30]	-	0.959	0.874	0.914	0.906	0.887	0.890	0.863	0.831	0.846
Split [33]	Private	0.869	0.866	0.868	-	-	-	-	-	-
DeepTabStR [31]	ICDAR 2013	0.931	0.930	0.930	-	-	-	-	-	-
Siddiqui et al. [32]	Synthetic 500k	0.934	0.934	0.934	-	-	-	-	-	-
ReS2TIM [36]	ICDAR 2013†	0.734	0.747	0.740	-	-	-	-	-	-
GTE [38]	ICDAR 2013†	0.944	0.927	0.935	-	-	-	-	-	-
GraphTSR [2]	SciTSR	0.885	0.860	0.872	0.959	0.948	0.953	0.964	0.945	0.955
TabStruct-Net [26]	SciTSR	0.915	0.897	0.906	0.927	0.913	0.920	0.909	0.882	0.895
LGPMA	SciTSR	0.930	0.977	0.953	**0.982**	**0.993**	**0.988**	**0.973**	**0.987**	**0.980**
LGPMA	ICDAR 2013†	**0.967**	**0.991**	**0.979**	-	-	-	-	-	-

of 1×10^{-2} is divided by 10 every 5 epochs. The model's training on SciTSR and PubTabNet lasts for 12 epochs, and the fine-tuning process on ICDAR 2013 lasts for 25 epochs. We also randomly scale the longer side of the input images to the lengths in the range [480, 1080] for all training processes. In the testing phase, we set the longer side of the input image as 768. We empirically set all weight parameters as $\lambda_1 = \lambda_2 = \lambda_3 = 1$.

4.3   Results on Table Structure Recognition Benchmarks

We first conduct experiments on the datasets of ICDAR 2013 and SciTSR, and the evaluation metric follows [5] (counting the micro-averaged correctness of neighboring relations). Notice that since ICDAR 2013 has very few samples, many previous works used different training or pre-training data. To be comparable to [2] and [26], our model is only trained by the training set of SciTSR. We also report the result of the model that is then fine-tuned on the training set of ICDAR2013, labeled by †. The results of DeepDeSRT on SciTSR come from [2]. The results are demonstrated in Table 1, from which we can see that the proposed LGP-TabNet vastly surpasses previous advances on these three benchmarks by 4.4%, 3.5%, 2.5%, respectively. Beside, LGPMA shows no performance decline on the complicated testing dataset SciTSR-COMP, which demonstrates its powerful ability to perceive spatial relations.

We also test our model in a more challenging benchmark of PubTabNet, whose results are demonstrated in Table 2. Since the corresponding evaluation metric of TEDS [39] considers both table structure and text content, we simply adopt an attention-based model [12] to recognize the text recognition. In the results, our method surpasses the previous SOTA by 1.6 in TEDS. We also report the results that only considers the table structure, denoted as TEDS-Struc. The performance gap between TEDS-Struc and TEDS mainly comes from the recognition error and annotation ambiguities.

Table 2. Results on PubTabNet. TESDS-Struc only considers the table structures.

Methods	Training Dataset	Tesing Dataset	TEDS (All)	TEDS-Struc. (All)
EDD [39]	PTN-train	PTN-val	88.3	-
TabStruct-Net [26]	SciTSR	PTN-val	90.1	-
GTE [38]	PTN-train	PTN-val	93.0	-
LGPMA (ours)	PTN-train	PTN-val	**94.6**	**96.7**

Fig. 5. Visualization results on ICDAR2013, SciTSR, PubTabNet. Green boxes are detected aligned bounding boxes, and blue boxes are empty cells generated by the proposed table structure recovery pipeline.

Visualization Results. We demonstrate some of the visualization results in Figure 5, in which the green boxes denote the predicted non-empty boxes, and blue boxes are the recovered empty boxes. We can see that our model can predict cells' accurate boundaries, even for those cross span multiple rows/columns.

Figure 6 demonstrates an example that is successfully predicted due to the correct refinement. We only show the LPMA and GPMA maps in the horizontal direction, where the LPMA map is generated by overlapping all proposals' local maps. In the initially proposed bounding boxes, some boxes do not have enough breadth, which would lead to wrong matching results. After refining by LPMA and GPMA, these boundaries can reach the more feasible positions. The empty cells can also be merged feasibly according to the predicted segmentation map during the table structure recovery.

4.4 Ablation Studies

We randomly select 60,000 training images and 1,000 validation images from PubTabNet to conduct the subsequent ablation studies.

Effectiveness of Aligned Bounding Box Detection. To verify the effectiveness of the designed modules, we conduct a group of ablation experiments, as shown in Table 3. Besides the TEDS-Struc metric, we also report the text
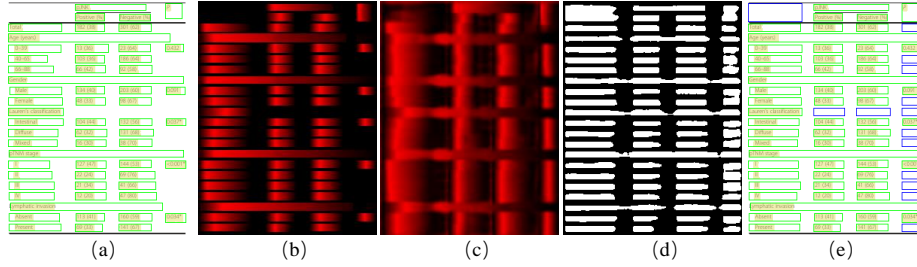
Fig. 6. Visualization of an example that is successfully refined. (a) The aligned bounding boxes before refinement. (b) LPMA (in horizontal). (c) GPMA (in horizontal). (d) Global binary segmentation. (e) Final result after refinement and empty cell merging.

Table 3. Ablation experiments on that how different modules effect the aligned bounding box detection

Models	Modules			Det of text regions			Det of non-empty aligned bounding boxes			TEDS-Struc.
	LPMA	GPMA	AL[26]	Precision	Recall	Hmean	Precision	Rrecall	Hmean	
Faster R-CNN				-	-	-	81.32	81.31	81.31	94.63
Mask R-CNN				91.71	91.53	91.62	81.83	81.82	81.83	94.65
	✓			91.92	91.66	91.79	84.29	84.10	84.20	95.22
		✓		91.98	91.50	91.74	83.48	83.18	83.33	95.04
	✓	✓		**92.27**	**91.86**	**92.06**	**85.14**	**84.77**	**84.95**	**95.53**
Mask R-CNN			✓	92.11	91.85	91.98	81.91	81.79	81.85	94.94
	✓	✓	✓	92.05	91.65	91.85	84.87	84.50	84.68	95.31

region's detection results and aligned bounding boxes, where the detection IoU threshold is 0.7, and empty cells are ignored. From the results, we can easily find that both LPMA and GPMA can vastly enhance the aligned bounding box detection performance, which is also proportional to the performance of TEDS-Struc. Although these modules are designed for aligned bounding boxes, they also slightly enhance the performance of text region detection results. The performance gap between text region detection and aligned bounding box detection again demonstrates the latter task is much more difficult and challenging.

We also evaluate the effectiveness of Alignment Loss (abbr. AL) proposed by [26]. Although solely adopting AL achieves better performance than the original Mask-RCNN, the performance is even lower than the best results of LGPMA when compromising all three modules. It means AL might bring adverse impact on LGPMA. Compared to AL, our proposed LGPMA can obtain more performance gain by 3.1% in aligned bounding box detection and 0.59 in TEDS-Struc.

Effectiveness of Table Structure Recovery. To verify the proposed table recovery pipeline's effectiveness, we conduct experiments to compare different

Table 4. Ablation experiments on different empty cells merging strategy. The bottom part shows the results given non-empty aligned bounding boxes ground-truth.

Structure recovery strategies	with non-empty box GT?	Det of empty aligned bounding boxes			Det of all aligned bounding boxes			TEDS-Struc.
		Precision	Recall	Hmean	Precision	Recall	Hmean	
Minimum empty cells	✗	56.17	63.04	59.40	82.36	82.74	82.55	95.50
Maximum empty cells	✗	23.19	10.76	14.70	82.30	77.39	79.77	92.57
Proposed LGPMA	✗	**68.12**	**72.43**	**70.21**	**82.69**	**83.17**	**82.93**	**95.53**
Minimum cells	✓	96.14	97.00	96.56	99.60	99.69	99.64	99.52
Maximum cells	✓	41.13	15.22	22.22	97.58	91.19	94.28	95.60
Proposed LGPMA	✓	**97.26**	**97.68**	**97.47**	**99.85**	**99.88**	**99.87**	**99.77**

empty cell merging strategies, as illustrated in Table 4. The evaluations contain the detection results that only consider empty cells, the detection results of all aligned bounding boxes, and TEDS-Struc. The strategy of using minimum cells means the direct results after *Empty Cell Searching*, and using maximum cells means merging all neighboring empty cells with the same height/width. From the results, we can see that our strategy using the visual information from GPMA can correctly merge many empty cells and obtain the highest performances in both detection and TED-Struc metrics. Compared with Strategy of *Minimum empty cells* , the promotion of *Empty Cell Merging by LGPMA* on TEDS-Struc is relatively small. This is because the number of empty cells minority, and most of them in this dataset are labeled in the smallest shapes. Nevertheless, our proposed empty cell merging strategy is more robust to adapt to any possibility.

Suppose the aligned bounding boxes of non-empty cells are detected perfectly, which equals the situation given ground-truths, as shown in the bottom part of Table 4. In this case, we can easily find that using the strategy of whether *Minimum cells* or the proposed *LGPMA* can almost achieve 100% accuracy in table structure recovery, and many errors come from the noisy labels. It demonstrates the robustness of our table structure recovery pipeline, and the performance mainly depends on the correctness of aligned bounding box detection.

5   Conclusion

In this paper, we present a novel framework for table structure recognition named LGPMA. We adopt the local and global pyramid mask learning to compromises advantages from both local texture and global layout information. In the inference stage, fusing the two levels' predictions via a mask re-scoring strategy, the network generates more reliable aligned bounding boxes. Finally, we propose a uniform table structure recovery pipeline to get the final results, which can also predict the feasible empty cell partition. Experimental results demonstrate our method has achieved the new state-of-the-art in three public benchmarks.

References

1. Bron, C., Kerbosch, J.: Finding all cliques of an undirected graph (algorithm 457). Commun. ACM **16**(9), 575–576 (1973)
2. Chi, Z., Huang, H., Xu, H., Yu, H., Yin, W., Mao, X.: Complicated table structure recognition. CoRR **abs/1908.04729** (2019)
3. Doush, I.A., Pontelli, E.: Detecting and recognizing tables in spreadsheets. In: IAPR. pp. 471–478 (2010)
4. Gao, L., Huang, Y., Déjean, H., Meunier, J., Yan, Q., Fang, Y., Kleber, F., Lang, E.M.: ICDAR 2019 competition on table detection and recognition (ctdar). In: ICDAR. pp. 1510–1515 (2019)
5. Göbel, M.C., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 table competition. In: ICDAR. pp. 1449–1453 (2013)
6. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. In: ICCV. pp. 2980–2988 (2017)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
8. Itonori, K.: Table structure recognition based on textblock arrangement and ruled line position. In: ICDAR. pp. 765–768 (1993)
9. Khan, S.A., Khalid, S.M.D., Shahzad, M.A., Shafait, F.: Table structure extraction with bi-directional gated recurrent unit networks. In: ICDAR. pp. 1366–1371 (2019)
10. Kieninger, T.: Table structure recognition based on robust block segmentation. In: Document Recognition V. vol. 3305, pp. 22–32 (1998)
11. Koci, E., Thiele, M., Lehner, W., Romero, O.: Table recognition in spreadsheets via a graph representation. In: IAPR. pp. 139–144 (2018)
12. Lee, C., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: CVPR. pp. 2231–2239 (2016)
13. Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: Tablebank: Table benchmark for image-based table detection and recognition. In: LREC. pp. 1918–1925 (2020)
14. Li, Y., Huang, Z., Yan, J., Zhou, Y., Ye, F., Liu, X.: GFTE: graph-based financial table extraction. In: ICPR Workshops. vol. 12662, pp. 644–658 (2020)
15. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR. pp. 936–944 (2017)
16. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV. Lecture Notes in Computer Science, vol. 8693, pp. 740–755 (2014)
17. Liu, J., Liu, X., Sheng, J., Liang, D., Li, X., Liu, Q.: Pyramid mask text detector. CoRR **abs/1903.11800** (2019)
18. Liu, Y., Bai, K., Mitra, P., Giles, C.L.: Improving the table boundary detection in pdfs by fixing the sequence error of the sparse lines. In: ICDAR. pp. 1006–1010 (2009)
19. Liu, Y., Mitra, P., Giles, C.L.: Identifying table boundaries in digital documents via sparse line detection. In: CIKM. pp. 1311–1320 (2008)
20. Milletari, F., Navab, N., Ahmadi, S.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3DV. pp. 565–571 (2016)
21. Nishida, K., Sadamitsu, K., Higashinaka, R., Matsuo, Y.: Understanding the semantic structures of tables with a hybrid deep neural network architecture. In: AAAI. pp. 168–174 (2017)
22. Paliwal, S.S., D, V., 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. pp. 128–133 (2019)

23. Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In: CVPR Workshops. pp. 2439–2447 (2020)

24. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table recognition using graph neural networks. In: ICDAR. pp. 142–147 (2019)

25. Qiao, L., Tang, S., Cheng, Z., Xu, Y., Niu, Y., Pu, S., Wu, F.: Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In: AAAI. pp. 11899–11907 (2020)

26. Raja, S., Mondal, A., Jawahar, C.V.: Table structure recognition using top-down and bottom-up cues. In: ECCV. Lecture Notes in Computer Science, vol. 12373, pp. 70–86 (2020)

27. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)

28. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NeurIPS. pp. 91–99 (2015)

29. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Networks **20**(1), 61–80 (2009)

30. 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. pp. 1162–1167 (2017)

31. Siddiqui, S.A., Fateh, I.A., Rizvi, S.T.R., Dengel, A., Ahmed, S.: Deeptabstr: Deep learning based table structure recognition. In: ICDAR. pp. 1403–1409 (2019)

32. Siddiqui, S.A., Khan, P.I., Dengel, A., Ahmed, S.: Rethinking semantic segmentation for table structure recognition in documents. In: ICDAR. pp. 1397–1402 (2019)

33. Tensmeyer, C., Morariu, V.I., Price, B.L., Cohen, S., Martinez, T.R.: Deep splitting and merging for table structure decomposition. In: ICDAR. pp. 114–121 (2019)

34. Wang, Y., Phillips, I.T., Haralick, R.M.: Table structure understanding and its performance evaluation. Pattern Recognit. **37**(7), 1479–1497 (2004)

35. Xie, E., Zang, Y., Shao, S., Yu, G., Yao, C., Li, G.: Scene text detection with supervised pyramid context network. In: AAAI. pp. 9038–9045 (2019)

36. Xue, W., Li, Q., Tao, D.: Res2tim: Reconstruct syntactic structures from table images. In: ICDAR. pp. 749–755 (2019)

37. Zanibbi, R., Blostein, D., Cordy, J.R.: A survey of table recognition. Int. J. Document Anal. Recognit. **7**(1), 1–16 (2004)

38. Zheng, X., Burdick, D., Popa, L., Wang, N.X.R.: Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context. CoRR **abs/2005.00589** (2020)

39. Zhong, X., ShafieiBavani, E., Jimeno-Yepes, A.: Image-based table recognition: Data, model, and evaluation. In: ECCV. vol. 12366, pp. 564–580 (2020)

40. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: an efficient and accurate scene text detector. In: CVPR. pp. 2642–2651 (2017)