




Image-Based Relation Classification Approach for Table Structure Recognition

Koji Ichikawa^(✉) 

The Japan Research Institute, Limited, Tokyo, Japan
ichikawa.koji@jri.co.jp

Abstract. In recent years, the use of tabular data has become a major area of research and development. However, the number of tables structured in a machine-readable format is still limited. A major challenge that is encountered when using tabular data is converting the table information in a free-format document into a structured format. Unlike markup languages such as HTML, XML, and JSON, free-format documents such as PDF, Word, Excel, and images generally have no tags or separators. Therefore, the table structure should be recognized from the positional information of the table elements. A major approach of table structure recognition is to classify the relationship between each pair of bounding boxes of the table elements. Recent works have achieved significant improvements by applying graph convolutional networks (GCNs) to the graph structure of the bounding boxes. However, fully recognizing a complex table structure is still a major challenge, owing to the presence of spanning cells. In this study, we propose a novel, simple image-based approach to this relation classification task. Our model efficiently exploits information such as the geometry of the table elements and ruled lines through an image cropping strategy based on the pairs of bounding boxes. We evaluate our approach on two real-world table datasets by comparing four baselines including two state-of-the-art GCN approaches. We observe that our approach significantly outperforms the baseline in the exact matching ratio for tables by up to 6.7%.

Keywords: Table structure recognition · Image recognition · Relation classification

1 Introduction

In recent years, table information retrieval has garnered substantial attention. In several cases, table data describe, explain, or complement key statements in the document; therefore, they can be utilized for various natural language processing tasks, such as question answering systems [16, 30, 34], constructing or augmenting a knowledge base [4, 22, 23], and fact-checking [1]. In particular, tables that are contained in free-format documents such as PDF, Word, Excel, and images are often critical for the above tasks, e.g., experimental data in papers; financial

performance in financial reports; and statistics in public documents, invoices, and ledgers.

However, the amount of table data available for machines is still limited; a major reason for this is that extracting the tables and modifying them into a machine-readable format is still a great challenge. This difficulty arises because free-format documents do not have tags or separators for tables similar to markup languages such as HTML, XML, and JSON; therefore, even after identifying the location of the table [6, 9, 24, 25, 29], it is necessary to structure it to a machine-readable format.

Specifically, the main issue is parsing the table elements to the machine-readable table format. Table elements can be extracted using a PDF content-stream analyzer or an optical character reader. However, these tools only provide a bounding box position for each table element. To obtain machine-readable table data, it is essential to parse these bounding boxes into a structured table format. This task is often known as table structure recognition, and is the main subject of this study.

For table structure recognition, the following difficulties prevent a simple pre-defined rule strategy: (1) the presence of spanning cells; (2) the width and height of the bounding box must vary. For instance, an intuitive approach would be to construct a parsing rule based on the relative positions of the bounding boxes; i.e., if two or more bounding boxes are aligned on a single vertical line, these boxes may belong to a single column. This rule-based approach sometimes works, especially for a simple table. In practice, however, most tables have spanning cells that belong to multiple columns or rows. Moreover, determining the box alignment is difficult because of the different widths and heights of each bounding box.

To overcome the above difficulties, recent studies have proposed deep neural network-based approaches. An earlier attempt [24] employs fully convolutional network (FCN) architecture [15] to detect the row and column regions. This approach has also been adopted in recent works [27, 28], which applied the object detection framework. The advantage of this approach is that it can naturally incorporate the table structure information, such as ruled lines or margins. However, one should take care of the mechanism through which the blank cells are joined to construct the spanning cells [31, 35], which is necessary for correctly recognizing the hierarchical structure of the table. In this paper, we refer to this approach as the detection-based approach.

Recently, relation classification approaches have been proposed in several studies [2, 14, 18, 21], wherein row and column recognition is considered as a relation classification task between a pair of bounding boxes. The advantage of this method is that a joint operation is not required for constructing spanning cells. Most studies on this approach utilize the graph structure of the table elements and employ graph convolutional networks (GCNs) [13], which successfully recognize multi-rows/columns using spanning cells. However, one major disadvantage of this approach is the difficulty of feature engineering. For instance, it is difficult

to fully utilize ruled line information using this approach. In this paper, we refer to this approach as the graph-based approach.

In this study, we adopt a novel and simple image-based relation classification approach for the table structure recognition task. Our idea is to employ an edge-based rectangle region formed by each pair of nodes as the input to a relation classifier. This rectangle contains essential information for the classification: the relative position, ruled lines, and the geometry of bounding boxes. Moreover, enlarging this edge-based region incorporates the global patterns of the table, which significantly improves the model accuracy. We stress that our approach has the advantages of both detection- and graph-based approaches, and succeeds in considerably reducing the complex design of pre-defined rules or feature engineering. Another advantage of our approach is that the data can be augmented through label-invariant operations. We propose novel label-invariant data augmentation techniques for the edge-based region, and demonstrate that they make significant contributions, especially when training with small amounts of data. In summary, our contributions are as follows.

- We propose a novel edge-based cropping strategy for table structure recognition.
- We introduce an edge region-based convolutional neural network (ER-CNN) that efficiently encodes the edge-based cropped images and positional information of the bounding boxes.
- We propose efficient data augmentation techniques for the edge-based cropped images.

We evaluate our approach on two real-world table datasets consisting of PDF and handwritten scanned images. We compare our approach with four baselines, including two state-of-the-art graph-based approaches. We observe that our approach significantly outperforms the baselines in the exact matching ratio for tables.

The remainder of this article is organized as follows. In Sect. 2, we briefly review related works. In Sect. 3, we define the problem that is the focus of this study. In Sect. 4, we introduce the motivation of our approach through observation. In Sect. 5, we describe our approach. We then present our experimental results in Sect. 6; finally, we provide a conclusion in Sect. 7.

2 Related Works

For table structure recognition, similar to the development of the table detection task [6, 9, 24, 25, 29], recent studies adopted a deep learning approach rather than pre-defined rules or heuristics [11, 26, 32] for structuring more complex tables. Several studies use the semantic segmentation or object detection methods to detect the columns and rows of a table [24, 27, 28]. The difference in our approach is that, while the approaches in previous studies are based on row and column detection, we adopt the relation classification approach and employ the edge-based cropping strategy for the classification.

Recently, other approaches based on relation classification have been proposed [2, 14, 18, 21]. In this approach, the table structure is recognized via the relationship between each cell. Most works for this approach utilize the graph structure of the table elements, considering each bounding box as nodes. In [14], the graph structure is constructed using the k -nearest neighbor (k -NN) algorithm, and features for the classification are constructed via GCN [13]. In [2], a multi-head attention mechanism is incorporated. In this study, both the node and edge features are convoluted via GCN, thereby exchanging their feature propagation. [21] also convolutes the edge feature via GCN architecture. Meanwhile, [18] adopts GravNet [17] and DGCNN [33] for graph convolution. A significant difference in our approach is that, while the previous studies mainly utilize the positional information of the table element for their input feature, we incorporate information about the ruled lines and geometry of bounding boxes by adopting CNN-based architecture and an edge-based region cropping strategy.

Our approach also relates to object detection and categorization, such as R-CNN [8], Fast R-CNN [7], and Faster R-CNN [20] in that the cropped image can be considered as a proposed object, and the relation classification corresponds to the categorization. The difference is that we determine the cropping region through the combination of the nodes, and utilize both the image and position for the model input to stress the geometry of the component.

3 Problem Setting

In our problem setting, we define dataset \mathcal{D} as a set of n tables: $\mathcal{D} \equiv \{T^1, \dots, T^n\}$, where each table T^t consists of a table image I^t , set of bounding boxes \mathcal{B}^t and set of relations \mathcal{R}^t .¹

$$T^t \equiv \{I^t, \mathcal{B}^t, \mathcal{R}^t\}. \quad (1)$$

The table image I^t has an image with $H^t \times W^t$ pixels and C channels, i.e.,

$$I^t \in [0, 1]^{H^t \times W^t \times C}. \quad (2)$$

In this study, we assume that table images are preprocessed into gray-scaled or binarized pictures with a single channel; that is, $C = 1$. Meanwhile, \mathcal{B}^t is a set of bounding boxes for each table element, i.e.,

$$\mathcal{B}^t \equiv \{b_1^t, b_2^t, \dots, b_{m^t}^t\}, \quad (3)$$

where m^t denotes the number of bounding boxes contained in table T^t . b represents a bounding box that is defined as follows:

$$b_i^t \equiv (x_{li}^t, y_{li}^t, x_{ri}^t, y_{bi}^t). \quad (4)$$

¹ Note that our approach does not require additional information such as text or captions in the table.

We let each bounding box be described as a rectangle with a four-dimensional vector (x_l, y_t, x_r, y_b) , where (x_l, y_t) and (x_r, y_b) , respectively, represents the top-left and bottom-right position of the bounding box. Coordinate x/y increases from left/top to right/bottom; x and y satisfy $0 \leq x_l < x_r < W$ and $0 \leq y_t < y_b < H$, respectively. We also refer to (x_{ci}, y_{ci}) as the coordinate of the center of b_i . In practice, bounding boxes may be split, merged, or missing owing to incomplete identification.² Although such misidentification can be mitigated by improving the box identification tool, improvements of the tool are beyond the scope of this paper. Therefore, in our problem setting, we assume that the bounding boxes are ideally provided; that is, $b \in \mathcal{B}$ has a one-to-one correspondence with the table element. Finally, \mathcal{R}^t represents a set of relations between pairs of boxes, which is determined by a set of triplets:

$$\mathcal{R}^t \equiv \{(b_i^t, b_j^t, y_{ij}^t) \mid b_i^t, b_j^t \in \mathcal{B}^t, y_{ij}^t \in \mathcal{L}\}, \quad (5)$$

where \mathcal{L} represents a set of relation labels: $\mathcal{L} = \{\text{irrelevant, row, column}\}$. Subsequently, by analogy from the graph representation, we may refer to the bounding boxes as nodes and relations between boxes as edges. Moreover, we may omit the table index t if it is clear from the context.

The relation classification approaches for the table structure recognition are used to predict y_{ij} for b_i and b_j under a given table image I and a set of bounding boxes \mathcal{B} .

4 Observations

To clarify the motivation of our approach, we provide an overview of the relationship between nodes, edges, ruled lines, and other neighbor nodes, using concrete examples.

Figure 1 shows examples of the geometry of nodes in a table. The figure shows that the relationship between the two blue boxes differs depending on the geometry of the other nodes and the ruled lines, even if the relative positions of the two nodes are the same. From the upper examples in Fig. 1, most relationships can be inferred by observing the inner area of the two nodes. In (1), we can infer that a column relationship between the two blue nodes is allowed, whereas this is inappropriate in (2) because of the presence of the intermediate cell. Meanwhile, (3), (4), and (5) show the effect of the ruled lines: (3) allows for

² More specifically, the noise related to the identification of boxes can be classified into the following six types: box size, misalignment of box positions, mis-joining between boxes, unnecessary division of boxes, missing boxes, and presence of extra boxes. We expect that our data augmentation in Sect. 5.3 improves robustness against the first two cases. The rest of the cases, on the other hand, cannot be straightforwardly dealt with by the relation classification approach, and the accuracy is degraded by the noise. While we expect that the noise can be suppressed by state-of-the-art box identification tool, we also expect that it is possible to extend our approach to an end-to-end framework [19] to address them, which we see as an interesting future work.

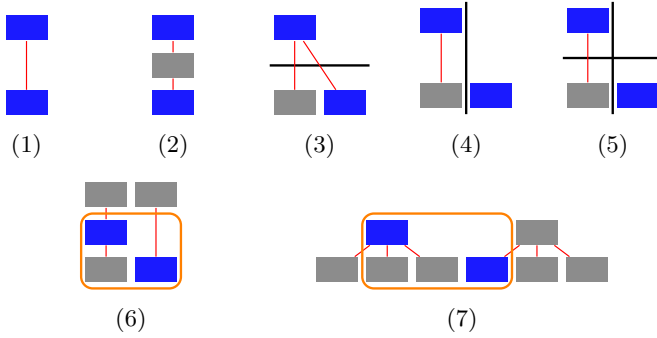


Fig. 1. Relationships between column relations and table elements. The boxes represent the bounding boxes in the table, and the two blue boxes correspond to the nodes of interest. The red lines represent the column relationships, while the thick black lines represent the table rule lines. Here, we omit the row relationships. (Color figure online)

column relationships between blue nodes, whereas (4) does not. Similarly, the combination of these ruled lines shown in (5) cuts off the column relationship between the two blue nodes.

Meanwhile, there are examples where the outer geometry influences the relationship, as shown in the lower examples in Fig. 1. In these examples, if we focus only on the inner area (orange rectangle), there could be a column relationship between the two blue nodes. However, once we increase the size of the region, such a relationship is found to be inappropriate because of the relationship with the other nodes. This observation suggests that the model should incorporate a proper range of peripheral information.

In the previous relation classification approaches, these geometrical patterns were not efficiently incorporated. This is because constructing a node or edge feature that incorporates these geometrical patterns requires hard feature engineering. Meanwhile, the image near the pair of nodes, we call it the *edge region*, naturally contains such information, which is efficiently extracted by a CNN architecture without complex feature engineering. Motivated by these observations, we propose a novel image-based relation classification approach, which is discussed in the subsequent section.³

5 Description of Our Approach

Our approach consists of two modules: the preprocessor, which extracts the information of the edge region, and the ER-CNN, which is employed as the classification model. An overview of our approach is shown in Fig. 2.

³ We note that because our approach adopt CNN architecture, the inference speed is slower than that of the graph-based approaches. However, we believe that the table structure recognition does not necessarily require as high an inference speed as object detection tasks that are intended for applications such as automated driving.

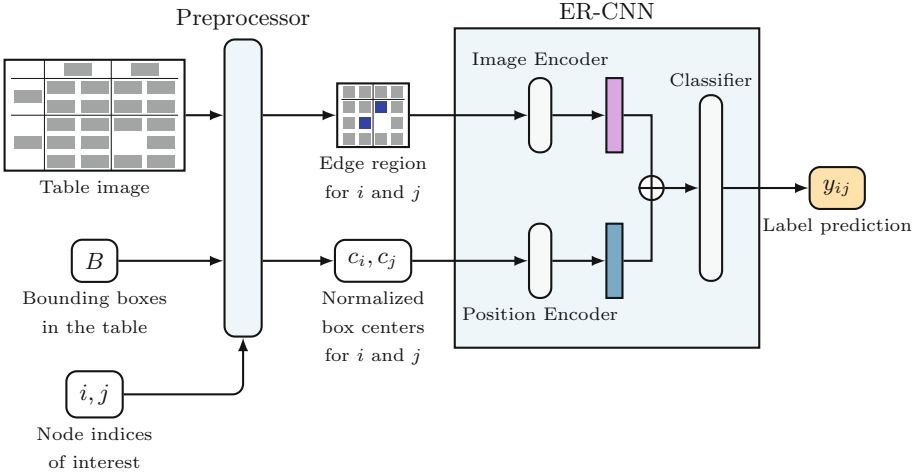


Fig. 2. Illustration of our approach.

5.1 Preprocessor

Edge-Based Cropping Region. The key idea of our approach is to employ a node pair-based image cropping strategy. Based on the observations in the previous section, we use a cropped table image with a rectangle formed by a pair of nodes as a primitive feature for the relation classification. More concretely, if node pairs have bounding boxes at $(x_{ri}, y_{ti}, x_{li}, y_{bi})$ and $(x_{rj}, y_{tj}, x_{lj}, y_{bj})$, we define the inner region of the bounding boxes as follows:

$$b_{ij}^e \equiv (\min(x_{li}, x_{lj}), \min(y_{ti}, y_{tj}), \max(x_{ri}, x_{rj}), \max(y_{bi}, y_{bj})). \tag{6}$$

This region encompasses information about the inner state between two nodes.

Another important aspect for relation classification is the outer status near the node pair. To incorporate this global information, we scale the width and height of the inner region b_{ij}^e . The edge-based cropping region is determined as follows:

$$b_{ij}^{\prime e} = (\min(x_{li}, x_{lj}) - rw_{ij}, \min(y_{ti}, y_{tj}) - rh_{ij}, \max(x_{ri}, x_{rj}) + rw_{ij}, \max(y_{bi}, y_{bj}) + rh_{ij}). \tag{7}$$

Here, w_{ij} and h_{ij} denote the width and height of b_{ij}^e , respectively, and r is a hyperparameter that defines the scale of the cropping region and we adopt $r = 1$ in this paper. If the cropping region extends outside the image, we fill in the overflow with a blank value. We cut out the rectangular region $b_{ij}^{\prime e}$ of the original table image I , which we define as I_{ij}^e , and use for crafting an input image for ER-CNN.

Crafting the Input Image of the Model. We construct the input of ER-CNN by splitting the cropped image I_{ij}^e into three channels: the channels of bounding boxes i and j , the channel of the other bounding boxes, and the channel of the other pixels, containing the noise and ruled information of the image. In the channels for the boxes, the rectangles of the boxes are filled with a constant, and the remaining area is filled with a blank value. This channel splitting helps the model to correctly recognize the table components. Finally, we resize this channel-split image to a 64×64 square shape for the input of ER-CNN.

Box Position Extraction. In addition to the cropped image, we utilize the positional information of the bounding boxes. The sizes of the bounding boxes can vary considerably depending on the lengths and styles of the original table elements. Therefore, the pixel area occupied by a bounding box can be sometimes extremely small after the cropping and resizing procedure. In such a case, the geometry of the table element cannot be extracted properly from the image alone. To cope with this problem, we explicitly input the box position into the model. Specifically, we use the box centers of b_i and b_j normalized by the size of b_{ij}^e : $\{c_i, c_j\}$, where

$$c_i \equiv \left(\frac{x_{ci} - x'_{lij}}{w'_{ij}}, \frac{y_{ci} - y'_{tij}}{h'_{ij}} \right). \quad (8)$$

Here, w'_{ij} and h'_{ij} denote the width and height of b_{ij}^e , respectively, and x'_{lij}/y'_{tij} is left/top position of b_{ij}^e .

5.2 Model

As described in Fig. 2, the ER-CNN consists of two encoders: an image encoder and a position encoder. The outputs of these encoders are concatenated, and then, passed through a final classification layer that outputs the label probability. For the backbone architecture of the image encoder, we adopt a small pre-trained residual neural network (ResNet) model (with 18 layers) [10]. The encoded vector is obtained via the first block layer output of the ResNet (with 64 channels) and subsequent two FC layers with batch normalization and rectified linear unit (ReLU) activation. Notably, one can easily exchange this module with a larger and more complex architecture. As demonstrated in the results section, our approach performs well, even with this shallow architecture.

For the position encoder, the input is a set of box positions defined by Eq. (8). Each two-dimensional coordinate is first embedded into a d -dimensional hidden space \mathcal{R}^d via transformation f : $l_i = f(c_i)$. Thereafter, the encoded position vector l_p is obtained by inputting the concatenation of l_i, l_j into function g : $l_p = h(l_i \oplus l_j)$. For the transformation functions f and g , we adopted a two-FC layer with ReLU activation. Similarly, we set up the final classification layer using two FC layers with batch normalization, ReLU activation, and a softmax function. We set the size of all hidden layers to 64 and $d = 64$.

5.3 Data Augmentation

An advantage of the image-based approach is that the amount of data can be augmented through label-invariant operations. Unlike typical image classification tasks, however, the geometry or presence of bounding boxes and ruled lines significantly affects the relation label; therefore, commonly used data augmentation, such as random crop, random erasing [36] or cutout [3], are likely to generate noisy samples.

We introduce two novel label-invariant data augmentation techniques for our approach: randomly changing the size of the bounding boxes and adding noise near the box. The former is based on the intuition that the size of the box can vary depending on the size of the characters of the table element, whereas the relationship is mostly independent of the character size. The latter incorporates noise that often occurs near the box, owing to the mismatch between the characters and bounding box. Specifically, we create the augmented images \tilde{I}_{ij}^e table-by-table according to the process in Sect. 5.1, using the randomly rescaled bounding boxes and the table image. More precisely, we first fill the original box areas with a blank value, and then place the rescaled bounding boxes. We added one augmented data per sample for our model’s training set. A new set of augmented data was generated for each training iteration

5.4 Scalability

We finally mention the scalability of our approach. If prediction is done for all combinations of boxes, $\mathcal{O}(m^2)$ computations are required in each table, which is difficult to perform for large tables. However, this computational complexity can be reduced by the fact that distant boxes are mostly irrelevant pairs. Specifically, we reduce the number of candidate pairs of boxes by using a k -NN method based on the location of the boxes [2]. This reduces the computational complexity to $\mathcal{O}(km)$, making it practically feasible. Besides, we expect that it is possible to reduce the number of actual CNN computation using techniques similar to Fast R-CNN [7], which is an interesting future work.

6 Experiments

In this section, we first review the datasets and introduce evaluation metrics. Next, we introduce the baselines and experimental details. Finally, we present the results under the three experimental settings.

6.1 Dataset

We used two real table datasets: SciTSR [2], comprising typed PDF images, and ICDAR2019 [5], comprising of handwritten scanned images.

The SciTSR dataset comprises 15,000 PDF format tables, containing bounding boxes, relationships, and table images for each table. The average number of

nodes in one table is approximately 40. Because we found that some of the table relationships were missing in the bounding box in the lower-left corner, we fixed the generation code. In addition, some tables in the dataset were out of the PDF area, which were removed by imposing a simple threshold to the maximum and minimum positions of the bounding box. After filtering, 11,134 training tables and 2,801 test tables were obtained. In the test dataset, a list of complex table IDs is provided (635 tables after the preprocessing above), and we also report the performance on this list as the complex test set.

The ICDAR2019 dataset comprises 850 (600 for training and 250 for test) scanned table images with handwritten entries. The ground truth of the table area and table structure is provided in XML format. We constructed the relations by parsing these XML files. To reduce the overlap between the boxes and ruled lines, we reduced the size of the bounding box by 50%. In addition, because the images had various background colors, we gray-scaled and binarized the images using threshold values of 80 percent quantile for each table image. Sometimes one scanned image contained multiple tables; these tables were split using XML tags. In the test set, we found that images with ID numbers greater than 10,000 had significantly different properties than the other training and test data: not handwritten, captured images, approximately one-tenth the size of the images. Most models did not perform well against this test set; therefore, we decided to separately evaluate them as in-domain and out-of-domain test set. After preprocessing, we obtained 677 tables for training, 190 tables for the in-domain, and 145 for the out-of-domain test set.⁴ The average number of nodes per table was approximately 300.

6.2 Evaluation Metrics

We adopted macro-averaged precision, recall, and F1 scores as metrics for our experiment. These metrics tend to achieve a high score in the relation classification of table structures. For instance, one misclassification for each table yields a F1 score of approximately 0.99. However, such misclassification, even at a rate of one per table, seriously degrades the performance of subsequent natural language processing tasks. Therefore, we employed an additional metric, the exact match. This metric yields 1 if the predicted rows or columns match the ground truth perfectly in each table. In our experiment, we measured the average ratio of the exact match for rows, columns, and tables (i.e., 1 if both rows and columns yield perfect matches).

6.3 Baselines

We compared our model performance with the following four baselines.

⁴ We removed one XML data that contained zero-width bounding boxes.

Rule Base. A simple, but strong baseline, for constructing the table rows and columns using a pre-defined rule. Here, we adopted the following extraction algorithm based on the overlaps of the pairs of bounding boxes.

1. Select $b_i \in \mathcal{B}$, which is located at the left/top most position.
2. Select b_j , which is located at the left/top position most in $\mathcal{B} \setminus \{b_i\}$.
3. If b_i and b_j do not overlap on the vertical/horizontal-axis and overlap on the horizontal/vertical-axis above a threshold length, we set $y_{ij} = \text{row/column}$, otherwise $y_{ij} = \text{irrelevant}$.
4. If $y_{ij} = \text{irrelevant}$, then remove b_j and go back to 2.
5. If b_j that satisfies $y_{ij} \neq \text{irrelevant}$ is found or all b_j is searched, then we assign $y_{ij'} = \text{irrelevant}$ to all the remaining $b_{j'}$, and restart this algorithm from 1 replacing \mathcal{B} with $\mathcal{B} \setminus \{b_i\}$.

In our experiments, we set the threshold value in the step 3 as 50% of the smaller height/width of b_i and b_j . Because this rule accurately identifies the row/column relationship between two distant nodes, the algorithm achieves high prediction accuracy, although it cannot structure a spanning-cell relationship.

MLP. Multi-layer perceptron (MLP) is a class of a feed-forward network consisting of input, output, and hidden layers. In this experiment, we constructed a module with three hidden layers and ReLU activations. As the input, we fed a concatenation of the pair of node features. We will describe the node feature adopted in the experiment in Sect. 6.4.

GraphTSR. [2] incorporates node and edge features for the input of the graph neural network. The author adopts a multi-head attention layer for the graph convolution. Both node and edge features are constructed based on the positions and sizes of the nodes. We adopted the same architecture and features for our experiment.

Ties. [18] shows variations in the architecture of the graph convolution mechanism for node features. From the results of the study, we adopted the DGCNN [33] module, where the node graph structure is dynamically constructed by the hidden features of the nodes. We set the number of the vertex neighbors for the DGCNN to 10. The image information was also used for the node feature by convolving the table image and sampling the CNN feature at the node position. In addition, the authors employed an edge sampling strategy to reduce the memory complexity. In our experiment, we sampled a constant number of negative samples (i.e., irrelevant edges) for each node. We set the number of the negative samplings for each node to 10.⁵

⁵ Ties also incorporates the textual information into the node features. In our experiment, we do not use the textual information.

Table 1. Performances on SciTSR dataset. P , R , and F_1 are precision, recall and F1 scores respectively. The numbers in parentheses represent standard deviations.

	Row [%]			Column [%]			Exact match [%]		
	$1 - P$	$1 - R$	$1 - F_1$	$1 - P$	$1 - R$	$1 - F_1$	Row	Column	Table
Full test set									
Rule	0.64 _(0.00)	1.17 _(0.00)	1.03 _(0.00)	1.40 _(0.00)	3.09 _(0.00)	2.40 _(0.00)	86.4 _(0.0)	73.4 _(0.0)	70.8 _(0.0)
MLP	9.13 _(0.83)	12.96 _(0.51)	13.27 _(0.34)	1.38 _(0.06)	2.74 _(0.12)	2.26 _(0.11)	21.2 _(1.1)	68.9 _(0.6)	18.9 _(1.1)
GraphTSR	0.87 _(0.07)	1.52 _(0.03)	1.33 _(0.02)	2.12 _(0.08)	1.79 _(0.15)	2.12 _(0.07)	79.0 _(0.3)	69.0 _(0.7)	64.8 _(0.5)
Ties	0.86 _(0.20)	0.93 _(0.11)	0.99 _(0.04)	1.20 _(0.22)	0.49 _(0.02)	0.89 _(0.12)	83.3 _(0.9)	82.7 _(2.1)	76.9 _(2.2)
ER-CNN	0.64 _(0.02)	0.27 _(0.04)	0.49 _(0.02)	0.80 _(0.03)	0.42 _(0.05)	0.64 _(0.01)	89.2 _(0.3)	87.0 _(0.3)	83.6 _(0.2)
Complex test set									
Rule	1.06 _(0.00)	5.13 _(0.00)	3.54 _(0.00)	3.57 _(0.00)	13.07 _(0.00)	8.94 _(0.00)	61.7 _(0.0)	11.7 _(0.0)	0.5 _(0.0)
MLP	12.54 _(1.17)	10.20 _(0.20)	12.95 _(0.65)	1.81 _(0.03)	7.86 _(0.24)	5.26 _(0.16)	9.7 _(0.9)	26.4 _(1.3)	4.6 _(0.5)
GraphTSR	1.32 _(0.21)	3.36 _(0.35)	2.52 _(0.17)	3.49 _(0.32)	4.94 _(0.23)	4.54 _(0.20)	50.2 _(1.5)	29.2 _(1.2)	23.6 _(0.6)
Ties	1.52 _(0.43)	1.20 _(0.20)	1.43 _(0.21)	2.30 _(0.53)	1.54 _(0.13)	2.02 _(0.22)	70.4 _(1.5)	60.4 _(5.0)	52.1 _(4.6)
ER-CNN	0.95 _(0.04)	0.90 _(0.13)	0.98 _(0.06)	1.41 _(0.07)	1.28 _(0.21)	1.40 _(0.07)	79.8 _(0.3)	71.3 _(1.3)	63.1 _(2.1)

Table 2. Performances on ICDAR2019 dataset.

	Row [%]			Column [%]			Exact match [%]		
	$1 - P$	$1 - R$	$1 - F_1$	$1 - P$	$1 - R$	$1 - F_1$	Row	Column	Table
In-domain test set									
Rule	1.16 _(0.00)	3.06 _(0.00)	2.14 _(0.00)	0.08 _(0.00)	0.85 _(0.00)	0.47 _(0.00)	44.2 _(0.0)	59.5 _(0.0)	41.1 _(0.0)
MLP	0.28 _(0.04)	1.32 _(0.07)	0.84 _(0.04)	0.10 _(0.02)	0.05 _(0.00)	0.08 _(0.01)	50.5 _(1.1)	74.0 _(2.5)	49.8 _(2.2)
GraphTSR	0.26 _(0.11)	1.85 _(0.04)	1.10 _(0.07)	0.08 _(0.03)	0.04 _(0.00)	0.06 _(0.01)	47.9 _(1.9)	78.2 _(2.2)	47.0 _(1.1)
Ties	1.23 _(0.13)	9.07 _(0.26)	6.44 _(0.26)	0.66 _(0.07)	0.23 _(0.07)	0.45 _(0.06)	26.5 _(1.1)	65.1 _(1.3)	24.7 _(1.8)
ER-CNN	0.36 _(0.11)	1.32 _(0.11)	0.87 _(0.10)	0.05 _(0.04)	0.20 _(0.09)	0.13 _(0.03)	56.8 _(0.0)	78.2 _(1.1)	56.5 _(0.6)
Out-of-domain test set									
Rule	2.81 _(0.00)	9.02 _(0.00)	6.40 _(0.00)	12.66 _(0.00)	25.93 _(0.00)	20.20 _(0.00)	37.9 _(0.0)	8.3 _(0.0)	6.2 _(0.0)
MLP	43.24 _(4.50)	92.16 _(1.03)	87.75 _(1.60)	46.83 _(4.92)	91.62 _(1.36)	86.77 _(1.95)	1.4 _(0.0)	0.0 _(0.0)	0.0 _(0.0)
GraphTSR	70.21 _(24.10)	95.28 _(3.35)	93.24 _(5.36)	53.14 _(32.10)	89.72 _(13.25)	85.32 _(17.68)	1.4 _(0.0)	0.5 _(0.8)	0.0 _(0.0)
Ties	6.52 _(2.27)	13.13 _(2.87)	10.48 _(2.80)	9.23 _(3.03)	9.52 _(2.03)	9.71 _(1.55)	17.7 _(6.3)	23.0 _(8.2)	10.6 _(3.9)
ER-CNN	2.76 _(0.91)	26.45 _(9.50)	18.22 _(7.00)	3.26 _(1.19)	11.99 _(0.16)	8.29 _(0.50)	15.2 _(7.2)	26.0 _(3.3)	10.6 _(3.4)

Table 3. Ablation study on the SciTSR dataset.

	Row [%]			Column [%]			Exact match [%]		
	$1 - P$	$1 - R$	$1 - F_1$	$1 - P$	$1 - R$	$1 - F_1$	Row	Column	Table
-DA	0.68 _(0.02)	0.32 _(0.04)	0.54 _(0.01)	0.89 _(0.09)	0.41 _(0.08)	0.67 _(0.02)	88.6 _(0.2)	86.3 _(0.2)	82.7 _(0.3)
$r = 0$	0.71 _(0.02)	0.40 _(0.03)	0.60 _(0.01)	1.13 _(0.03)	0.58 _(0.02)	0.90 _(0.02)	87.7 _(0.5)	82.6 _(0.3)	78.7 _(0.6)
$r = 0.5$	0.70 _(0.01)	0.26 _(0.00)	0.52 _(0.01)	0.85 _(0.04)	0.47 _(0.01)	0.69 _(0.02)	88.9 _(0.4)	85.2 _(0.3)	82.0 _(0.7)
$r = 1$	0.64 _(0.02)	0.27 _(0.04)	0.49 _(0.02)	0.80 _(0.03)	0.42 _(0.05)	0.64 _(0.01)	89.2 _(0.3)	87.0 _(0.3)	83.6 _(0.2)
$r = 2$	0.59 _(0.04)	0.36 _(0.05)	0.50 _(0.01)	0.83 _(0.06)	0.42 _(0.04)	0.65 _(0.03)	88.3 _(0.5)	86.5 _(0.6)	82.4 _(0.5)

6.4 Experimental Details

We split the training dataset in a ratio of 3:1; the former was used for training and the latter for validation. The training was terminated by referring to the average F1 score of the rows and columns of the validation data. We adopted the cross-entropy loss, and minimized it using the Adam optimizer [12]. For GraphTSR, and Ties, we referred to the official code, and modified or reconstructed them for our experiments, retaining their original architectures.

To construct a set of relations \mathcal{R} including $y = \text{irrelevant}$, we adopted a conventional negative sampling method: For each node i , we constructed a set of node pairs by pairing i to k -NN nodes. We assigned a row, column, or irrelevant label to each node pair by referring to the ground truth. For all baselines, we set the number of nearest neighbor nodes $k = 20$ for training and validation, and $k = 50$ for testing except for GraphTSR. For GraphTSR, we found that the same k for training and prediction yielded a higher performance, and hence we adopted $k = 20$ for prediction.

We used the following 16 node features for MLP and Ties: box positions, box centers, box width, and box height, along with these features normalized by the table size. The box center normalized by the table size was also used for the k -NN box search. For GraphTSR, the features are standardized, and the edge feature is used. We adopted the same features as the original codes.

Each approach was run three times and the average values are reported.

6.5 Performances on the Full Data

First, we tested the performance of the model when trained on all training data samples. Tables 1 and 2 summarize the results.⁶ For SciTSR dataset, we observe that our method significantly outperforms the baselines for most of the metrics.⁷ In ICDAR2019 dataset, the images are distorted and noisy, which is difficult for image-based approach. Nevertheless, our approach achieves a competitive performance with the baselines on the F1 scores and significantly outperforms on the exact matching ratio for rows and tables. We expect this accuracy to improve further with more sophisticated image preprocessing.

6.6 Ablation Studies

Next, we present the results of the ablation studies of our model on the SciTSR dataset. In this experiment, we tested the model performance by ablating the data augmentation (-DA) and changing scaling parameter r in the range of $(0, 0.5, 1, 2)$. The results are summarized in Table 3. Interestingly, even $r = 0$, which only contains information on the inner part, yields a competitive accuracy with the baselines, indicating the importance of the internal states of the node pairs. In contrast, the best accuracy was obtained at $r = 1$, and the accuracy decreased for larger r values. This is because, when the cropped area is excessively large, important information on the inner part is missing owing to the resizing procedure.

⁶ We note that the benchmark micro-F1 scores [2] of ER-CNN were 0.993, 0.990, and 0.984, for SciTSR full test set, SciTSR complex test set, and ICDAR2019 full test set, respectively, although a direct comparison may be inappropriate due to the difference in preprocessing.

⁷ We checked that the p -values of the F1 scores in SciTSR dataset were less than or close to 0.05. In addition, we performed two additional runs for GraphTSR, Ties, and ER-CNN and confirmed p -values < 0.05 for the F1 scores in SciTSR dataset.

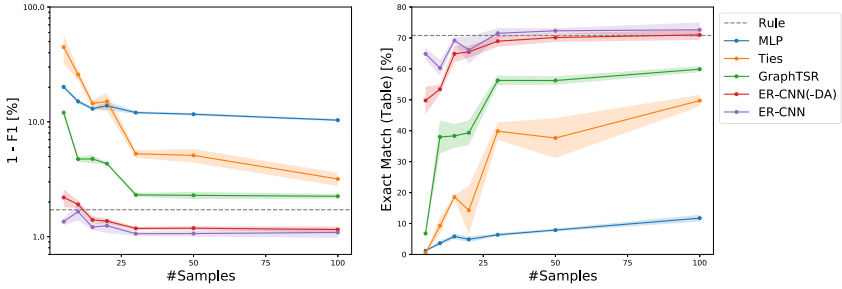


Fig. 3. Performances on the small data. The shaded regions represent the standard deviation. F1 in the left panel represents the average of the row and column F1 scores.

6.7 Performances on the Small Data

Finally, we present the model performance under a small data setting. Documents often contain sensitive information, and crowdsourcing is not always available. In such a situation, it is desirable to achieve practical accuracy with a small number of annotations. Assuming a sample size that is sufficiently small to be annotated without crowdsourcing, we sampled 5, 10, 15, 20, 30, 50 and 100 tables for training and 10 tables for validation from the SciTSR training dataset. The performance was evaluated on the same test dataset used in the full data experiment.

The results are presented in Fig. 3. In graph-based approaches (Ties and GraphTSR), the model cannot be trained well with such a small training dataset, and the performances are below that of the rule-based baseline. In contrast, our method significantly outperforms the graph-based approaches and is competitive with the rule-based baseline, even without data augmentation. With data augmentation, the performance increases further; even with five samples, the exact matching ratio is greater than 60% and with 30 samples, it is above 70%.

7 Conclusion and Discussion

In this study, we proposed a novel image-based approach for the table structure recognition task. Our model efficiently exploits information on geometry of the table elements and table ruled lines through edge-based cropped images. We evaluated our approach on two real-world table datasets, consisting of typed PDFs and handwritten scanned images, in comparison with four baselines. We have observed that our approach significantly outperforms the baselines in the exact matching ratio for tables. In addition, our experiments have confirmed that our approach works well with small amounts of data. We finally note that our approach can be easily combined with the graph convolutional architecture by exchanging the position encoder with a GCN architecture, which may help to improve robustness against noisy and distorted images.

References

1. Chen, W., et al.: Tabfact: a large-scale dataset for table-based fact verification. In: ICLR 2020
2. Chi, Z., Huang, H., Xu, H.D., Yu, H., Yin, W., Mao, X.L.: Complicated table structure recognition. arXiv preprint [arXiv:1908.04729](https://arxiv.org/abs/1908.04729)
3. Devries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint [arXiv:1708.04552](https://arxiv.org/abs/1708.04552)
4. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: KDD 2014. <https://doi.org/10.1145/2623330.2623623>
5. Gao, L., et al.: ICDAR 2019 competition on table detection and recognition (CTDAR). In: ICDAR 2019. <https://doi.org/10.1109/ICDAR.2019.00243>
6. Gilani, A., Qasim, S.R., Malik, M.I., Shafait, F.: Table detection using deep learning. In: ICDAR 2017. <https://doi.org/10.1109/ICDAR.2017.131>
7. Girshick, R.B.: Fast R-CNN. In: ICCV 2015. <https://doi.org/10.1109/ICCV.2015.169>
8. Girshick, R.B., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR 2014. <https://doi.org/10.1109/CVPR.2014.81>
9. Hao, L., Gao, L., Yi, X., Tang, Z.: A table detection method for PDF documents based on convolutional neural networks. In: DAS 2016. <https://doi.org/10.1109/DAS.2016.23>
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR 2016. <https://doi.org/10.1109/CVPR.2016.90>
11. Kieninger, T., Dengel, A.: The T-Recs table recognition and analysis system. In: Lee, S.-W., Nakano, Y. (eds.) DAS 1998. LNCS, vol. 1655, pp. 255–270. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48172-9_21
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: ICLR 2015
13. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR 2017
14. Li, Y., Huang, Z., Yan, J., Zhou, Y., Ye, F., Liu, X.: GFTE: graph-based financial table extraction. In: Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R. (eds.) ICPR 2021. LNCS, vol. 12662, pp. 644–658. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68790-8_50
15. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR 2015. <https://doi.org/10.1109/CVPR.2015.7298965>
16. Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. In: ACL 2015. <https://doi.org/10.3115/v1/p15-1142>
17. Qasim, S.R., Kieseler, J., Iiyama, Y., Pierini, M.: Learning representations of irregular particle-detector geometry with distance-weighted graph networks. Eur. Phys. J. C **79**(7), 1–11 (2019). <https://doi.org/10.1140/epjc/s10052-019-7113-9>
18. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table recognition using graph neural networks. In: ICDAR 2019. <https://doi.org/10.1109/ICDAR.2019.00031>
19. Raja, S., Mondal, A., Jawahar, C.V.: Table structure recognition using top-down and bottom-up cues. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020, Part XXVIII. LNCS, vol. 12373, pp. 70–86. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_5
20. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS 2015

21. Riba, P., Dutta, A., Goldmann, L., Fornés, A., Terrades, O.R., Lladós, J.: Table detection in invoice documents by graph neural networks. In: ICDAR 2019. <https://doi.org/10.1109/ICDAR.2019.00028>
22. Ritze, D., Lehmborg, O., Bizer, C.: Matching HTML tables to DBpedia. In: WIMS 2015. <https://doi.org/10.1145/2797115.2797118>
23. Ritze, D., Lehmborg, O., Oulabi, Y., Bizer, C.: Profiling the potential of web tables for augmenting cross-domain knowledge bases. In: WWW 2016. <https://doi.org/10.1145/2872427.2883017>
24. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: DeepDeART: deep learning for detection and structure recognition of tables in document images. In: ICDAR 2017. <https://doi.org/10.1109/ICDAR.2017.192>
25. Shafait, F., Smith, R.: Table detection in heterogeneous documents. In: DAS 2010. <https://doi.org/10.1145/1815330.1815339>
26. Shigarov, A.O., Mikhailov, A.A., Altaev, A.: Configurable table structure recognition in untagged PDF documents. In: DocEng 2016. <https://doi.org/10.1145/2960811.2967152>
27. Siddiqui, S.A., Fateh, I.A., Rizvi, S.T.R., Dengel, A., Ahmed, S.: DeepTabStR: deep learning based table structure recognition. In: ICDAR 2019. <https://doi.org/10.1109/ICDAR.2019.00226>
28. Siddiqui, S.A., Khan, P.I., Dengel, A., Ahmed, S.: Rethinking semantic segmentation for table structure recognition in documents. In: ICDAR 2019. <https://doi.org/10.1109/ICDAR.2019.00225>
29. Siddiqui, S.A., Malik, M.I., Agne, S., Dengel, A., Ahmed, S.: DeCNT: deep deformable CNN for table detection. *IEEE Access* **6**, 74151–74161 (2018). <https://doi.org/10.1109/ACCESS.2018.2880211>
30. Sun, H., Ma, H., He, X., Yih, W., Su, Y., Yan, X.: Table cell search for question answering. In: WWW 2016. <https://doi.org/10.1145/2872427.2883080>
31. Tensmeyer, C., Morariu, V.I., Price, B.L., Cohen, S., Martinez, T.R.: Deep splitting and merging for table structure decomposition. In: ICDAR 2019. <https://doi.org/10.1109/ICDAR.2019.00027>
32. Wang, Y., Phillips, I.T., Haralick, R.M.: Table structure understanding and its performance evaluation. *Pattern Recognit.* **37**(7), 1479–1497 (2004). <https://doi.org/10.1016/j.patcog.2004.01.012>
33. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* **38**(5), 146:1–146:12 (2019). <https://doi.org/10.1145/3326362>
34. Zhong, V., Xiong, C., Socher, R.: Seq2SQL: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103* (2017)
35. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.: Image-based table recognition: data, model, and evaluation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) *ECCV 2020*. LNCS, vol. 12366, pp. 564–580. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_34
36. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: *AAAI* (2020)