electronics

MDPI

Article

TSRDet: A Table Structure Recognition Method Based on Row-Column Detection

Zixuan Zhu [1], Weibin Li [1,2,*], Chenglong Yu [2], Wei Li [1] and Licheng Jiao [2]

[1] Laboratory of AI, Hangzhou Institute of Technology of Xidian University, Hangzhou 311231, China; 22173224954@stu.xidian.edu.cn (Z.Z.); 22171214936@stu.xidian.edu.cn (W.L.)
[2] School of Artificial Intelligence, Xidian University, Xi'an 710071, China; 22171110155@stu.xidian.edu.cn (C.Y.); lchjiao@mail.xidian.edu.cn (L.J.)
* Correspondence: weibinli@xidian.edu.cn

Abstract: As one of the most commonly used and important data carriers, tables have the advantages of high structuring, strong readability and strong flexibility. However, in reality, tables usually present various forms, such as Excel, images, etc. Among them, the information in the table image cannot be read directly, let alone further applied. Therefore, the research related to image-based table recognition is crucial. It contains the table structure recognition and the table content recognition. Among them, table structure recognition is the most important and difficult task because the table structure is abstract and changeable. In order to address this problem, we propose an innovative table structure recognition method, named TSRDet (Table Structure Recognition based on object Detection). It includes a row-column detection method, named SACNet (StripAttention-CenterNet) and the corresponding post-processing. SACNet is an improved version of the original CenterNet. The specific improvements include the following: firstly, we introduce the Swin Transformer as the encoder to obtain the global feature map of the image. Then, we propose a plug-and-play row-column attention module, including a channel attention module and a row-column spatial attention module. It improves the detection accuracy of rows and columns by capturing long-range row-column feature maps in the image. After completing the row-column detection, this paper also designs a simple and fast post-processing to generate the table structure based on the row-column detection results. Experimental results show that for row-column detection, SACNet has high detection accuracy, even at a high IoU threshold. Specifically, when the threshold is 0.75, its mAP of row detection and column detection still exceeds 90%, which is 91.40% and 92.73% respectively. In addition, in the comparative experiment with the existing object detection methods, SACNet's performance was significantly better than that of all others. For table structure recognition, the TEDS-Struct score of TSRDet is 95.7%, which shows competitive performance in table structure recognition, and verifies the rationality and superiority of the proposed method.

Keywords: table recognition; structure recognition; row-column detection; CenterNet; swin transformer; spatial attention

check for updates

Citation: Zhu, Z.; Li, W.; Yu, C.; Li, W.; Jiao, L. TSRDet: A Table Structure Recognition Method Based on Row-Column Detection. *Electronics* **2024**, *13*, 4263. https://doi.org/10.3390/electronics13214263

Academic Editor: George A. Tsihrintzis

Received: 16 September 2024
Revised: 18 October 2024
Accepted: 19 October 2024
Published: 30 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In today's digital information age, tables, as an important way to store structured information, have been widely used to organize, present and analyze data, whether in business, science or government. Additionally, due to some reliability and security requirements such as handwritten signatures in sensitive scenarios, the use of paper tables remains unavoidable. However, a large number of paper tables or image tables make it difficult for the information contained within them to be directly utilized by computer systems. Consequently, the development of table recognition has become crucial.

As we all know, table recognition tasks are primarily divided into three major tasks: table area detection, table structure recognition and table content recognition. Existing table

area detection methods have already achieved relatively ideal results, including methods based on object detection [1–3], methods based on image segmentation [4–6], and methods based on features extraction [7–9]. Table content recognition is mainly accomplished through optical character recognition(OCR) methods, with mainstream approaches including CRNN [10], SVTR [11], ABINet [12] and MASTER [13]. Table structure recognition is achieved by understanding the table layout and connecting the table area detection to the table content recognition. Existing table structure recognition methods can be roughly divided into three categories: object detection-based, image segmentation-based, and sequence prediction-based. Among these, the first two methods rely on visual features, such as separator lines, cells, text blocks, rows and columns. Specifically, the object detection-based method reconstructs the table structure based on the extracted visual feature, which requires extremely high positioning accuracy because if the predicted results of the object's position are offset, even if they are very small, it will cause adhesion between adjacent elements. For instance, GTE [9], TRACE [8], LORE [14]and GraphTSR [15] are methods based on cell detection, while DeepTabStR [16] is a method based on row-column detection. In addition, the segmentation-based method is actually pixel-level classification, which has a large amount of computation and consumes more resources. It specifically includes separator line-based methods such as SPLERGE [17], RobustTabNet [7], TSRFormer [18], TRUST [19], SEM [20], and SEMv2 [21], row-column-based methods such as DeepDeSRT [1] and TableFCN [22], and cell-based methods such as TGRNet [23] and TabStructNet [24]. The last method based on sequence prediction takes table images as input and outputs a table structure represented by a text markup language. It is the method least affected by the quality of table images among the three methods. Its main methods include VAST [25], WSTabNet [26], EDD [27], MTLTabNet [28] and TableMaster [29].

Through the analysis and comparison of the above methods, this study believes that the row-column-based object detection method is the best solution. First, we choose the row-column-based method because visual features can be further divided into direct visual features and indirect visual features. The former includes features such as cells, separator lines, and text blocks, which often do not have structural prior features, so their missed detection will lead to serious errors in the table structure, while the latter includes row and column features, which have structural a priori features and are the first features that the human eye will notice when observing a table. They also have cell position alignment, structuring and other features. Therefore, we believe that rows and columns should be the best choice to describe cell features. Secondly, the method based on object detection is chosen because the row and column of the table are all rectangular structures, and the object detection method is better at locating the bounding box than segmentation. However, since this type of method has too high of a requirement for detection accuracy, even slight deviations in bounding boxes can cause them to stick together. At present, this type of method has not achieved the ideal effect and there is still room for improvement.

Based on the above analysis, this study proposes a new table structure recognition method based on row-column detection, TSRDet, which consists of two parts: SACNet (StripAttention-CenterNet) and post-processing. The former obtains the bounding boxes of rows and columns by improving CenterNet by combining hierarchical feature representation with a row-column attention mechanism. It is based on CenterNet, introduces Swin Transformer as the backbone, and then combines the proposed innovative module RCAM at multiple scales. The latter takes the intersection of the obtained row-column bounding boxes and corrects them to obtain the table structure.

For the experimental results of row-column detection, this study uses the mAP value under different IoU thresholds for evaluation, while The IoU threshold is set from 0.5 to 0.9, and the value is taken every 0.05; for the experimental results of table structure recognition, this study uses TEDS-Struct and the IoU scores under different IoU thresholds for evaluation. The experimental results show that compared with existing methods, this method significantly improves the accuracy of row-column detection. In addition, it can effectively generate the correct table structure based on the detection results, thereby showing the

ability to recognize the structure of various table types, with robust stability and good application prospects. In summary, the main contributions of this study can be summarized as follows:

1. We proposed a table structure recognition pipeline based on row and column detection. It consists of two parts: SACNet (StripAttention-CenterNet) and post-processing. The former adopts an improved CenterNet that combines hierarchical feature representation with row and column attention mechanism to detect row and column bounding boxes, while the latter intersects and corrects these bounding boxes to construct the table structure.
2. We compared the performance of existing object detection models on the row and column detection task and found that methods based on center point detection achieve higher localization accuracy and are more suitable for this task than anchor-based methods. This is attributed to the use of heatmaps in center point detection methods, which predict the centers of objects, reducing the likelihood of missing smaller objects, and eliminating the need for Non-Maximum Suppression (NMS), thereby speeding up detection.
3. We proposed a plug-and-play row-column attention module, denoted as RCAM, including a channel attention module and a row-column spatial attention module. This module improves the row-column detection accuracy by capturing long-range row-column feature maps in the image.
4. We proposed a novel row-column detection model named SACNet, derived from CenterNet, featuring Swin Transformer as its backbone. RCAM is combined at multiple scales in the encoder to improve the accuracy of row and column detection.

In the subsequent chapters of this article, Section 2 briefly reviews the main methods of table structure recognition, Section 3 elaborates on the specific details and structure of our proposed method TSRDet, Section 4 introduces the experimental dataset, evaluation indicators and some equipment parameter information, and further describes three parts of experiments in detail in Section 5. The first part explains the performance of our proposed method in the two experiments of row and column detection and table recognition, the second part conducts ablation experiments on our proposed method, and the third part also trains the selected comparison model and conducts evaluation of row and column detection and table structure recognition. Finally, in Section 6, we summarize the effectiveness and contribution of the method proposed in this paper, and also point out the shortcomings, based on which we propose future work directions.

2. Related Works

In this chapter, we briefly review the main approaches to table structure recognition, which are divided into three categories, i.e., segmentation-based approaches, sequence prediction-based approaches, and detection-based approaches, each of which is represented as follows.

Segmentation-based Approaches. According to the different objects of segmentation, they are divided into separator lines-based and row-column-based. In the approach based on separator lines, Chris et al. proposed the split-merge model SPLERGE [17], which initially forecasts the basic grid structure from input images and subsequently identifies which grids need to be merged to reconstruct cells across rows and columns, thereby achieving a comprehensive table structure. Zhang et al. designed SEM [20] which decomposes the table structure recognition process into three stages. First, it conducts multilevel feature mapping utilizing ResNet34+FPN as the backbone. Next, it employs a splitter to predict the fine grid structure based on the backbone's output. Finally, the merger combines the basic grid feature representations extracted by the embedder to derive the final grid structure. RobusTabNet [7] proposed by Ma et al. utilizes a lighter backbone, ResNet18+FPN, compared with SEM. It integrates a spatial CNN-based separator line prediction module and a cell merging module to accurately predict and reconstruct the tabular structure. Lin et al. proposed TSRFormer [18] by enhancing the RobusTabNet approach. While

retaining RobusTabNet's backbone unchanged, TSRFormer incorporates Spatial CNN after the backbone to augment feature representation. Subsequently, the predicted separator lines are extracted from the table image using SepRETR. Finally, the merged cells are reconstructed by integrating the output of ROI Align and Grid CNN to derive the table structure. In the approach based on row-column, Sebastian et al. proposed DeepDeSRT [1], a method for extracting table structures based on the FCN-Xs [30]. DeepDeSRT enhances FCN-8s by incorporating skip connections from the first and second layers, resulting in FCN-2s tailored for the completion of row-column segmentation tasks. Paliwal et al. proposed TableNet [4], leveraging the interdependence between table detection and table structure tasks, it first segments the column regions, then conducts row regions extraction based on formulated segmentation rules derived from the identified table sub-regions. This method typically demonstrates good performance; however, its efficacy often hinges on image quality. Moreover, it does not directly acquire the table structure and necessitates intricate post-processing procedures. In the approach based on cells, TGRNet [23] proposed by Xue et al., initially identifies the spatial positions of cells by extracting extensive feature maps, subsequently employing graph convolutional neural networks to ascertain their respective row and column coordinates; TabStructNet [24] initially utilizes Mask RCNN to predict cell masks and subsequently computes the row-column adjacency matrix for these cells. In this matrix, cells that share the same row or column are assigned a value of 1, while all other relationships are designated a value of 0.

Sequence-based Approaches. This type of approach typically utilizes a "Single Encoder-Multi-Decoder" architecture to obtain a table structure represented by a text markup language. Due to the ability of this method to obtain table structures without post-processing and the resilience of text sequences to image distortion, many such methods have emerged in recent years. EDD [27] is a single encoder dual decoder model that captures the visual features of input table images through an encoder. It then reconstructs the table structure and recognizes the content of cells through a structure decoder As and a cell decoder Ac, respectively, and finally obtains an HTML representation of the input table image. TableMaster [29] also adopts a single encoder dual decoder architecture, where the encoder converts images into sequences and then encodes them. The dual decoders correspond to two learning tasks: regression of cell boxes and prediction of table structure sequences. They, respectively, output the positional coordinates of the cell boxes and sequences of table structures annotated with hypertext syntax rules. VAST [25] follows a similar framework, extracting feature maps through a CNN image encoder, and then obtaining HTML sequences and bounding boxes of non-empty cells through an HTML sequence decoder. This information is sent to the coordinate sequence decoder, which decodes the position information of the unit cells in a sequence from top left to bottom right. MTLTabNet [28] stands out from other methods by employing a single shared encoder and decoder to address three table recognition tasks through three decoders: Cell-box decoder, Cell content decoder and Structure decoder. These decoders predict cell positions, text content, and HTML structure tags, respectively. The cell texts are then integrated into the corresponding HTML tags to generate the final HTML code of the table image. While this method effectively identifies table structures directly, its efficacy heavily depends on the volume of trainable data, particularly for text recognition. Recognition results are further constrained by sequence length and tag numbers, leading to unsatisfactory accuracy. Additionally, this method frequently struggles to accurately pinpoint cell locations.

Detection-based Approaches. In general, these methods can be categorized into two groups: cell-based and row-column based. The former entails generating a table structure by predicting cell positions and their logical relationships. Chi et al. proposed an approach where they initially extract the content and bounding boxes of cells directly from PDF documents through preprocessing. Subsequently, they construct the table as an undirected graph based on these cells and utilize their proposed GraphTSR [15] for predicting cell relationships. This method is further enhanced with post-processing to restore the table structure. GTE [9] eliminates the need for post-processing by incorporating a dedicated

GTE-Cell unit for cell detection. The model classifies table images by detecting six graphical table lines and feeds them into a cell detection network. Subsequently, it employs post-processing techniques to correct anomalies in the detections. Notably, GTE-Cell relies on the table boundaries provided by GTE-Table to generate specific cell structures for each table. LORE [14], redefines the task of identifying cell spatial positions by prioritizing the identification of the center point of each cell. It achieves this through a combination of cell center point recognition and coordinate regression, enabling LORE to extract the corner coordinates of cells. The model utilizes a CNN backbone to extract visual features from table cells and employs two regression heads to predict their spatial and logical locations. The fundamental concept of row-column detection is to construct a table structure by predicting the positions of rows and columns. DeepTabStR [16], leveraging deformable convolution, is built upon Faster RCNN and ResNet101. It begins by generating candidate object boxes via RPN, classifies these boxes using ROI Pooling and Classifier layers, and ultimately derives a table structure by intersecting the classified row and column regions. Overall, these approaches rely on visual features and typically exhibit superior performance and faster inference compared to segmentation-based approaches. However, they demand higher detection accuracy concerning cells and rows. Even minor offsets can lead to issues such as sticking or overlapping, posing challenges and limitations, particularly in intricate table structures.

Among the prevailing methods for table structure recognition, those based on object detection are widely utilized. However, techniques focusing specifically on row-column detection remain notably scarce. Recognizing that rows and columns, as basic elements of tables, implicitly combine the structural properties and alignment properties of cells, this study proposes a new method centered on row-column detection. The details of this method are elaborated in Section 3.

3. Methods

In this chapter, we elucidate the specifics of our proposed method, named TSRDet, consisting of two primary components: SACNet (a novel row-column detection model) and the corresponding post-processing. Initially, in Section 3.1, we comprehensively outline the design blueprint of SACNet, CenterNet. Subsequently, in Section 3.2, in order to improve the model's local grasp of more global row-column information, we delve into the more powerful encoder Swin Transformer with a hierarchical structure. Following this, in Section 3.3, to further enhance the model's perception of row and column information, we propose the RCAM, which consists of a channel attention module and a row-column spatial attention module, which is a pivotal highlight of our proposed method. We then present SACNet's architecture. Finally, in Section 3.4, we introduce the post-processing for generating table structures. The overall process of the proposed method is illustrated in Figure 1.

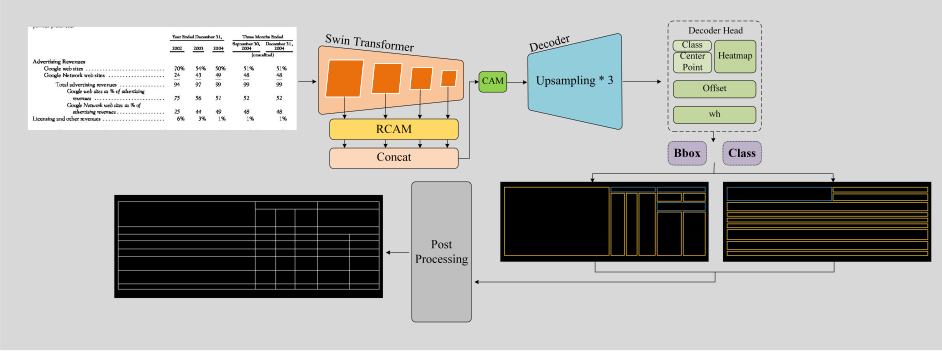


Figure 1. The procedure of TSRDet. It shows the process of our proposed method, which consists of SA-CenterNet and the corresponding post-processing. The '*' denotes the coordinates of the center points adjusted for offsets.

3.1. CenterNet

CenterNet [31], proposed by Zhou et al. in 2019, is an anchor-free object detection method that diverges from the conventional anchor-based methods utilized in popular methods such as Faster R-CNN [32], SSD [33], and the YOLO series [34]. These traditional methods typically rely on a multitude of anchors to achieve a high Intersection Over Union (IoU) with ground truth, requiring manually designed anchor boxes of specific sizes and aspect ratios.

Differing from the aforementioned methods, CenterNet performs object detection by predicting the centers, widths, and heights of objects, without the need for non-maximum suppression (NMS). This approach achieves an average detection precision of 45.1% on the MS COCO dataset [35], while maintaining a high frame rate. In terms of model architecture, the model comprises three main components: an encoder for extracting visual feature maps, a high-resolution decoder that generates expanded feature maps at an 8x resolution, and a final segment containing three detection heads that predict different types of information. The first head is responsible for predicting the categories and centers of objects, the second predicts the widths and heights of objects corresponding to their centers, and the third predicts offsets of the center points. By decoding the outputs from CenterNet according to specific rules, the categories and locations of the predicted objects can be determined. CenterNet discussed is illustrated in Figure 2. The key characteristics of this model are summarized as follows:

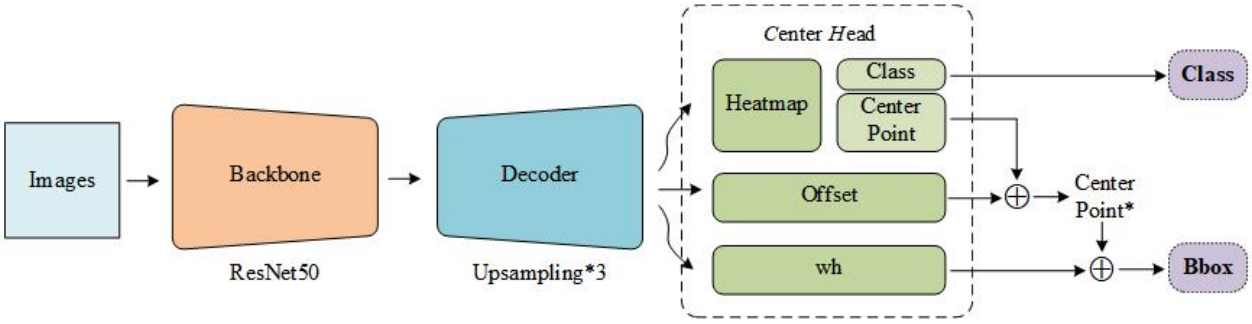


Figure 2. The architecture of CenterNet. It consists of three parts, backbone, decoder, and detection heads. The three detection heads, respectively, predict the center points and categories, the offsets, and the width and height of the objects. The '*' denotes the coordinates of the center points adjusted for offsets. From these coordinates and the dimensions of the objects, we can derive the bounding boxes of the objects.

First, CenterNet introduces an object detection method based on a single central point. Traditional anchor-based detection methods require pre-defining a large number of potential bounding box sizes and ratios, which not only increases the computational burden but also adds to the model's complexity. Meanwhile, other keypoint estimation-based object detection methods, such as CornerNet [36] and ExtremeNet [37], set two bounding box corners and the extremities (top, bottom, left, right, and center) of all objects as keypoints, respectively. This necessitates grouping and post-processing after detection. In contrast, CenterNet simplifies the process and reduces computational overhead by directly predicting the center points and dimensions (width and height) of objects, thereby enhancing efficiency.

Secondly, CenterNet employs keypoint heatmaps to denote the probable center points of objects. Each pixel in the heatmap represents the likelihood of that pixel being the object's center, thereby facilitating precise object localization. Mathematically, if (x_c, y_c) denotes the coordinates of an object's center point, then the value y at any point (x, y) on the heatmap is given by Equation (1):

$$y = \exp\left(-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}\right) \quad (1)$$

Here, σ represents the standard deviation, which scales with the object's size, modulating the spatial spread of the heatmap peak centered at (x_c, y_c). This formulation allows for a more precise localization of the center points, particularly for objects of varying sizes. The adaptive standard deviation, σ, ensures that the spatial resolution of the heatmap peak is proportional to the object's dimensions, thereby enhancing accuracy in center point detection across diverse object scales.

3.2. Swin Transformer

The migration of Transformers from NLP (Natural Language Processing) to the field of computer vision faces two main challenges. First, the scale of entities involved in the two domains differs; in NLP, the scale of entities is standardized and fixed, while in CV (Computer Vision), the scale of entities varies significantly. Second, CV requires a higher resolution than NLP, and the computational complexity of using Transformers in CV is proportional to the square of the image size, leading to excessively large computational demands.

In 2021, Liu et al. proposed an innovative visual Transformer architecture known as Swin Transformer [38], designed to enhance the efficiency and accuracy of image recognition tasks by constructing a hierarchical Transformer network. To address the aforementioned challenges, the Swin Transformer gradually merges image patches to construct a hierarchical Transformer as the network depth increases, making it a versatile visual backbone network suitable for tasks such as object detection. It adopts a hierarchical building approach commonly used in convolutional neural networks and introduces the concept of locality, performing self-attention calculations within non-overlapping windows. This approach ensures that the computational complexity is linearly proportional to the input image size, thereby improving efficiency. Additionally, it enables the model to capture broader contextual information while maintaining locality over larger regions, which is highly suitable for row-column detection tasks. The overall structure of the model is shown in Figure 3. The key features of the model are summarized as follows:

Hierarchical Transformer Network. The Swin Transformer constructs a hierarchical network that processes images at multiple scales, enhancing the model's ability to handle various object sizes and improving its applicability to different vision tasks. This hierarchical approach, inspired by the pyramidal structure common in convolutional neural networks, progressively merges adjacent patches to form feature maps at reduced resolutions, allowing for efficient high-resolution image processing.

Window self-attention mechanism. The Swin Transformer significantly reduces computational complexity by confining self-attention computations to fixed-size windows. Specifically, with a window size of $M \times M$, the complexity of self-attention decreases from $O(N^2)$ to $O(M^2N)$, where N represents the number of pixels in the image. Compared to traditional Transformer models, this method not only lowers computational complexity but also enhances the model's focus on local features by restricting the attention scope to local areas. Additionally, this window-based approach maintains computational efficiency by reducing the number of pixel pairs involved in the attention calculation.

Shifted Window Strategy. To maintain the efficiency of local window calculations while introducing cross-window information exchange, the Swin Transformer employs a shifted window mechanism. By alternating the partitioning of windows between successive Transformer layers, the Swin Transformer facilitates cross-window self-attention computation. This strategy enables the model to capture broader contextual information, maintaining locality over larger regions, while also avoiding the complexity of global attention calculations.

Enhanced Performance on Standard Benchmarks. The Swin Transformer sets new state-of-the-art performance benchmarks on several key datasets:

- **Image Classification on ImageNet-1K:** Achieved a top-1 accuracy of 87.3%, which is a notable improvement over previous models.

- **Semantic Segmentation on ADE20K:** Demonstrated a box Average Precision (AP) of 58.7 and a mask AP of 51.1, surpassing prior state-of-the-art models by margins of +2.7 box AP and +2.6 mask AP, respectively.
- **Object Detection and Instance Segmentation on COCO:** Recorded a mean Intersection-Over-Union (mIoU) of 53.5, which is an improvement of +3.2 mIoU over the best previous results.

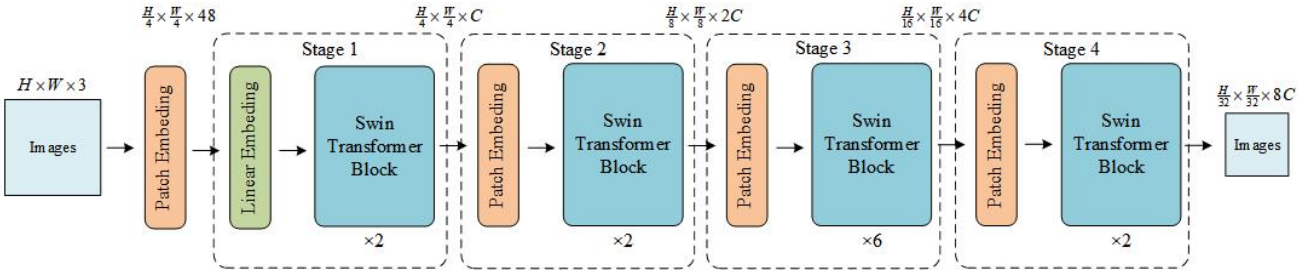


Figure 3. The architecture of Swin Transformer. It consists of four stages, each of which is a similar repetitive unit.

3.3. RCAM

According to the experimental outcomes presented by Sanghyun et al. [39], the synergy between channel attention and spatial attention mechanisms demonstrably amplifies the performance of convolutional neural networks in detection tasks. More crucially, this synergy enables more precise object localization through an enhanced attention distribution, effectively reducing the influence of extraneous elements. Recognizing the critical importance of row and column information within feature maps for precise detection tasks, we introduce the Row-Column Attention Module (RCAM). This module comprises two components: the Row-Column Spatial Attention Module (RCSAM) and the Channel Attention Module (CAM), detailed in Sections 3.3.1 and 3.3.2, respectively. Furthermore, a comprehensive exposition of the SACNet's architecture is thoroughly provided in Section 3.3.3.

3.3.1. CAM

Channel Attention Module (CAM), a channel-based attention mechanism introduced by Sanghyun et al. [39] in 2018, enhances the inter-channel relationship of features within convolutional neural networks. CAM operates by assessing the significance of different feature maps across channels, and is designed to be a plug-and-play component that can seamlessly integrate with existing network architectures, as depicted in Figure 4.

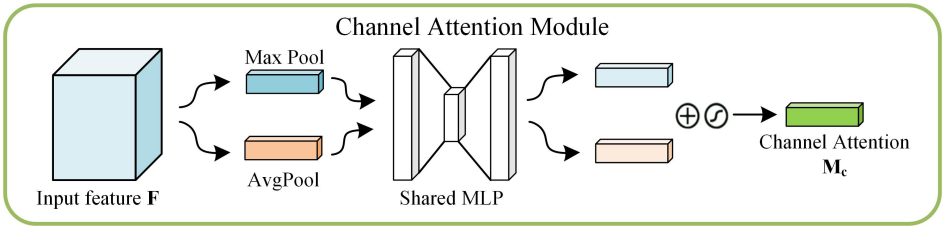


Figure 4. The Channel Attention Module. This figure shows the overall architecture of the channel attention module. It utilizes both max-pooling outputs and average-pooling outputs with a shared network.

According to the above, Figure 4 shows the processing of CAM. Specifically, Given an input feature map $F \in \mathbb{R}^{C \times H \times W}$, the process begins with the application of global Max-Pooling and AveragePooling across the channels to produce two separate context vectors. These vectors are then fed into a shared multi-layer perceptron (MLP) with a hidden layer configured as $(C/r, 1, 1)$, where r is the reduction ratio aiming to minimize parameter overhead while maintaining performance. This MLP outputs two Channel Attention Vectors,

which are combined using element-wise summation followed by a sigmoid activation to produce the final channel attention map $M_c(F)$, which can be calculated by Equation (2):

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (2)$$

This attention map is then element-wise multiplied with the original feature map F to yield a refined feature map, emphasizing informative features while suppressing less useful data.

3.3.2. RCSAM

The Row-Column Spatial Attention Module (RCSAM) proposed herein is an innovative architecture designed to enhance specific features within rows and columns of a feature map. Figure 5 illustrates the structure of RCSAM, which encompasses an input feature map, attention generation, feature recalibration, and fusion operations. The process is detailed as follows:

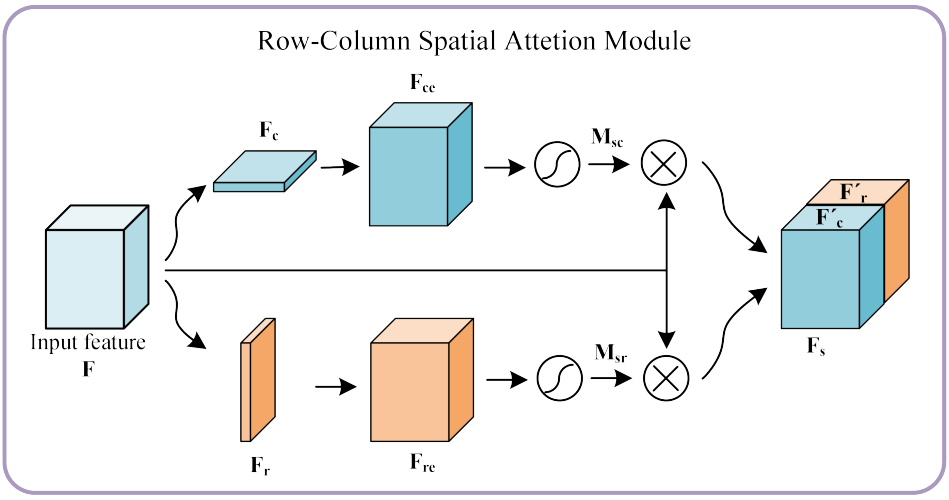


Figure 5. The Row-Column Spatial Attention Module. This figure shows the overall architecture of the row-column spatial attention module. As depicted, the module operates in primary phase: feature map input, attention generation, feature recalibration, and fusion operation.

1. **Input Feature Map**: Let the input feature map be denoted as $F \in \mathbb{R}^{C \times H \times W}$, where C, H, and W represent the number of channels, height, and width, respectively.
2. **Feature Aggregation**: Initially, the input feature map is globally aggregated along the vertical and horizontal directions using average pooling, extracting two distinct feature representations. The vertical aggregation generates a matrix $F_c \in \mathbb{R}^{C \times H \times 1}$, representing the average features for each channel along the row dimension. Similarly, horizontal aggregation produces a matrix $F_r \in \mathbb{R}^{C \times 1 \times W}$, representing the average features for each channel along the column dimension.
3. **Feature Restoration**: To restore the aggregated feature matrices to the same resolution as the input feature map, 1D convolution is applied to both F_c and F_r:
   - For F_c, a 1D convolution is applied along the column dimension, capturing row-wise features and restoring the matrix to shape $C \times H \times W$, resulting in the restored matrix F_{ce}.
   - For F_r, a 1D convolution is applied along the column dimension, capturing row-wise features and restoring the matrix to shape $C \times H \times W$, resulting in the restored matrix F_{re}.
4. **Activation and Normalization**: The restored matrices F_{ce} and F_{re} are then passed through a Sigmoid activation function, producing the normalized attention weight maps M_{sc} and M_{sr}. Here, M_{sc} represents the attention weights along the column dimension, while M_{sr} represents the attention weights along the row dimension.

5. **Feature Recalibration**: The attention maps M_{sc} and M_{sr} are applied to the original input feature map F by element-wise multiplication, resulting in recalibrated feature maps F'_c and F'_r. This step selectively enhances or diminishes features based on their importance.
6. **Fusion Operation**: The recalibrated feature maps F'_c and F'_r are concatenated along the channel dimension to produce the final output F_s, which combines the attributes of both attention mechanisms.

3.3.3. SACNet

The structure of SACNet proposed in this study is shown in Figure 6. It consists of an encoder, a decoder, and detection heads. The encoder is based on the Swin Transformer and includes four stages, each containing 2, 2, 6, and 2 Swin Transformer blocks, respectively. The output channel numbers for each stage are C, 2C, 4C, and 8C. While the spatial dimensions of the input features remain unchanged in the first stage, they are reduced by half with each subsequent stage.

Additionally, after the output of each stage in the Swin Transformer, we integrate the RCSAM module proposed in this study. The parallel outputs of the four RCSAM modules are concatenated and passed through a CAM module before being sent to the decoder. The decoder and detection heads retain the design of CenterNet. The decoder consists of three serial deconvolution layers, which double the spatial dimensions of the feature maps while reducing the number of channels. The channel numbers are fixed and sequentially decrease from the initial input to 256, 128, and 64.

Finally, SACNet outputs a heat map through three detection heads, including the objects' categories, center point coordinates, width and height, and the offsets of the center point. With this information, the bounding box of the object can be accurately determined.

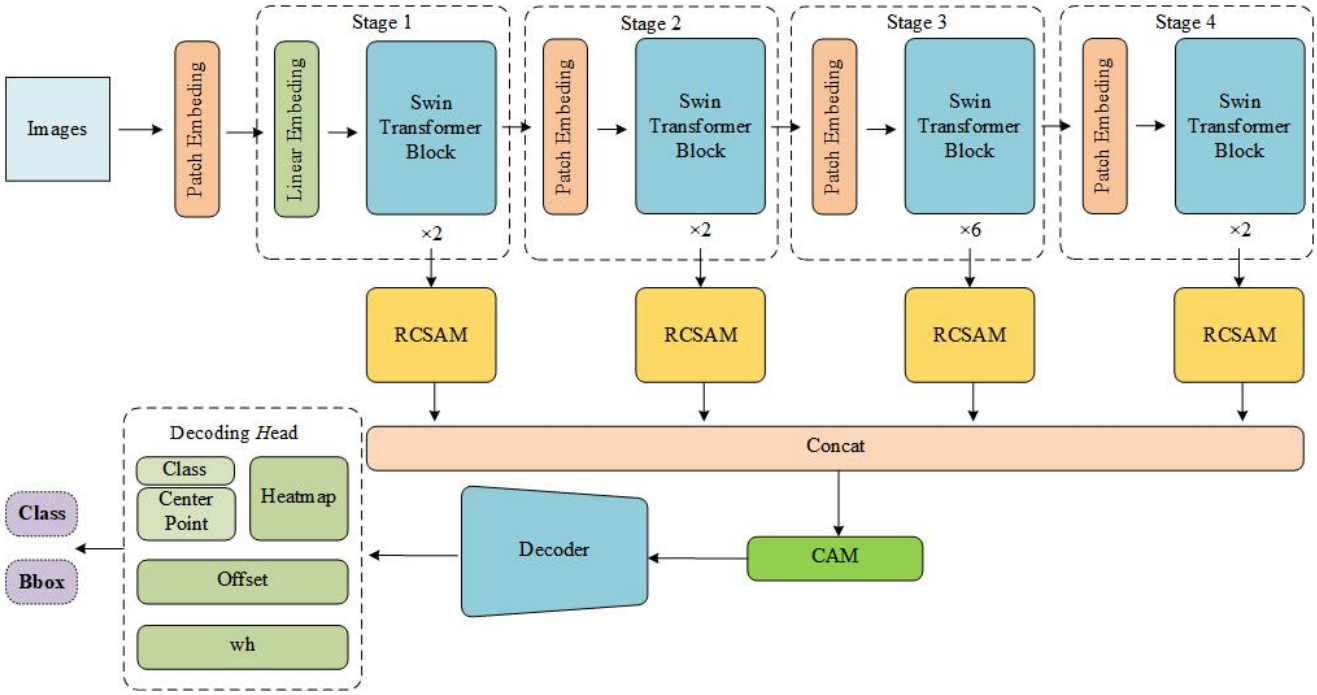


Figure 6. The overall architecture of SACNet. It consists of three parts, namely encoder, decoder, and the decoding head.

3.4. Post-Processing

Based on the above, we know that SACNet can only predict row and column bounding boxes in table images. Therefore, the objective of this section is to introduce a post-processing algorithm that utilizes image-processing techniques to derive the table structure from these bounding boxes. The specific steps are as follows:

First, we obtain the cells by performing intersection operations on the rows and columns. At this stage, the cells may appear irregular, such as misalignments and offsets within the same row or column, as shown in Figure 7. This can result in incorrect row and column position calculations. To address this issue, we apply a position adjustment operation to correct the cells and obtain a standardized table structure. Specifically, the position adjustment operation involves performing one erosion and dilation operation on the horizontal and vertical lines using convolution kernels of size $cols/scale$ and $rows/scale$, respectively, where cols and rows are the width and height of the image, and $scale$ is set to 5, thereby aligning the row and column positions. The flowchart of the post-processing algorithm is shown in Figure 7.

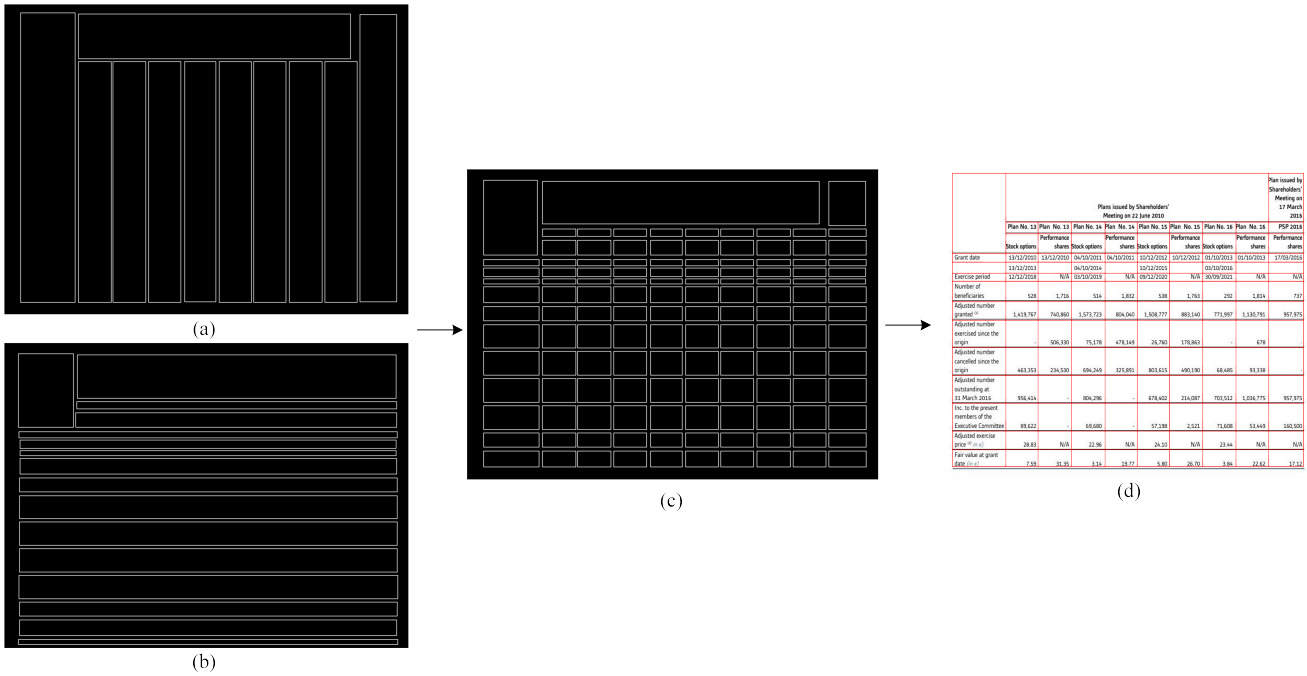


Figure 7. An example of post-processing. Among them, (**a**,**b**), respectively, show the column and row bounding boxes predicted by SACNet, where misalignment issues are evident. (**c**) displays the cells obtained after intersecting the rows and columns and applying position correction. (**d**) visualizes the resulting table structure on the original image.

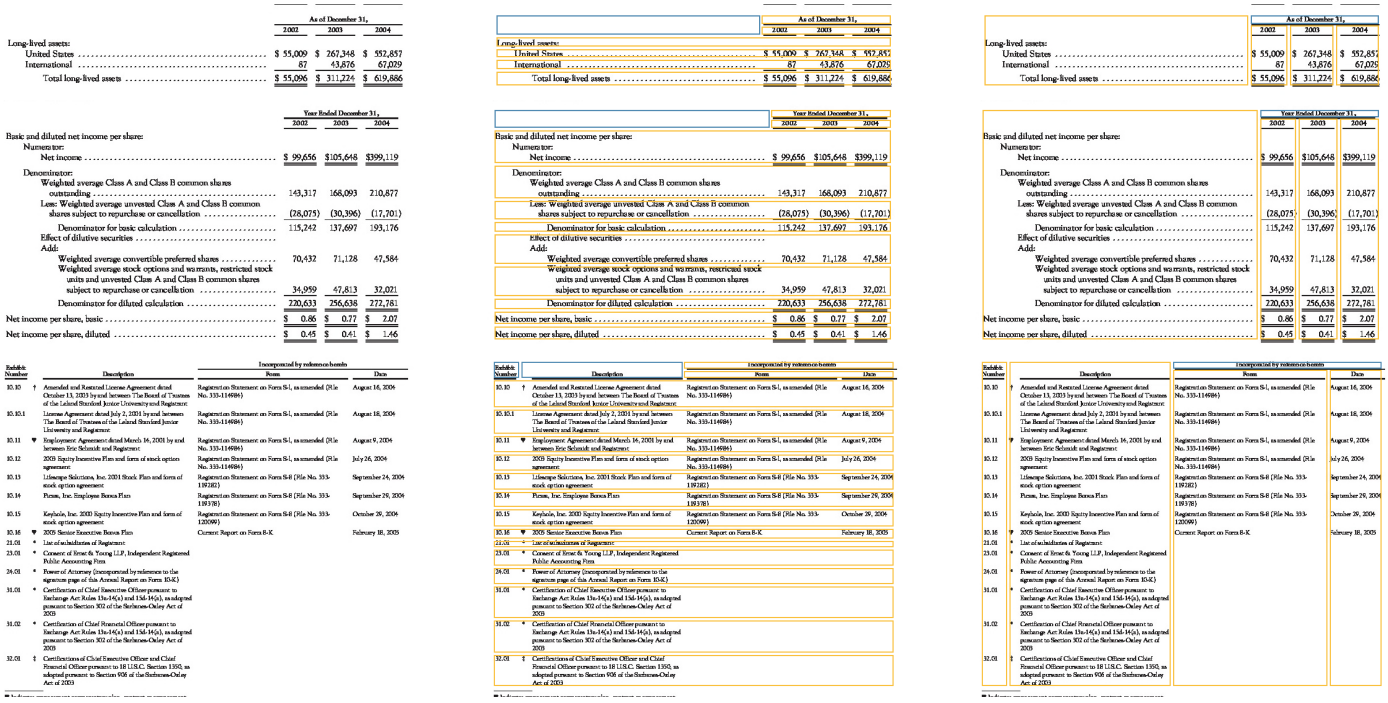
4. Experimental Preparation

In this chapter, we present the experimental setup in detail. Specifically, Section 4.1 introduces the table dataset we collected and annotated ourselves, consisting of a total of 11,555 images. In Section 4.2, we explain the evaluation metrics used in our experiments, including the mean Average Precision (mAP) for row and column detection, as well as the TEDS-Struct [27] and IoU scores for assessing table structure recognition. Finally, Section 4.3 provides a detailed overview of the hardware used and the specific parameter settings employed in our experiments.

4.1. Experimental Datasets

In existing public datasets, annotations for row and column bounding boxes are uncommon, often misaligned, and tend to overlook the presence of cells spanning multiple rows or columns. This leads to the division of table structure recognition tasks into two sub-problems: row-column recognition, and cell recognition with row and column spans, followed by post-processing to complete the structure recognition. However, this paper attempts to address the problem holistically by simultaneously predicting rows, columns, and cells with row and column spans.

Aiming at the above purposes, we collected a challenging dataset comprising 11,555 images, each containing a single table. The tables vary in form, including those with merged cells, large blank cells, wireless tables, and defective tables. Each image was carefully annotated: for the row dataset, bounding boxes were annotated across multiple rows, while for the column dataset, bounding boxes were annotated across multiple columns. Figure 8 illustrates some examples from the dataset along with their corresponding annotations. Additionally, we split the dataset into training, validation, and test sets in an 8:1:1 ratio, resulting in 9243, 1156, and 1156 images, respectively.



(**a**) Source Images (**b**) Row Bounding Box Annotations (**c**) Col Bounding Box Annotations

Figure 8. Some examples and the corresponding annotations in our collected dataset. The first col displays the table images, the second col shows the annotations for row-bounding boxes, and the third col displays the annotations for column-bounding boxes. In the second and third cols, each image is annotated in two different colors to represent different categories of objects.

4.2. Assessment Methodology

To evaluate the performance of our proposed method, we evaluate it from two different dimensions: row-column detection and table structure recognition. For the evaluation of the model's row and row detection capabilities, because this task requires high positioning accuracy, small position disturbances will cause adhesion between adjacent rows and rows, so we use mAP under different IoU thresholds for evaluation, according to Equation (3). It is calculated that the IoU threshold setting is from 0.5 to 0.9, and the value is taken every 0.05. For the evaluation of the model's table structure recognition ability, we use two evaluation indicators, TEDS-Struct [27] and IoU score, as shown in Equations (4) and (5). In particular, the calculation of the IoU score is still based on a threshold from 0.5 to 0.9, with a value every 0.05.

$$\begin{cases} \text{mAP} = \dfrac{1}{2}\displaystyle\sum_{i=1}^{2} \text{AP}_{\text{IoU}} \\ \text{AP} = \displaystyle\int_{0}^{1} p(r)\, dr \end{cases} \tag{3}$$

$$\text{TEDS-Struct(G,P)} = 1 - \frac{\text{Edit Distance}(G, P)}{\max(\|G\|, \|P\|)} \tag{4}$$

$$\begin{cases} \text{IoU} = \dfrac{\text{Area of Overlap}}{\text{Area of Union}} \\ \text{Score}_{\text{IoU}} = \dfrac{\text{Number of Correct Predictions}}{\text{Total Number of Valid Predictions}} \end{cases} \tag{5}$$

In Equation (4), TEDS represents the Tree-Edit Distance based on Similarity. G is the true sequence of table structure, while P is the predicted sequence of table structure. $\|G\|$ and $\|P\|$ represent the number of elements (nodes) in the sequences G and P, respectively. Edit Distance(G, P) refers to the minimum number of edit operations needed to transform the predicted sequence P into the true sequence G. In Equation (5), a *correct prediction* refers to a predicted region whose Intersection over Union (IoU) with the true label exceeds a certain threshold.

4.3. Implement Details

In this section, we present the implementation details of our experiments. First, in terms of training, the model was optimized using AdamW [40]. We use the Cosine decay scheme. The initial learning rate, betas and weight decay are set to 5×10^{-4} (0.9, 0.999) and 0, respectively. The total number of epochs of training is set to 100, and the batch size is 16. Secondly, in terms of the experimental environment, all experiments are implemented in PyTorch v1.12.0 and conducted on a workstation using 1 NVIDIA GeForce RTX A5000. More details are summarized in Table 1.

Table 1. The experimental environment.

Experimental Environment	Details
CPU	Intel(R)Xeon(R)Gold 6230 CPU
GPU	NVIDIA GeForce RTX A5000
Optimizer	AdamW
Total Batch Size	16
Learning Rate Schedule	The Cosine learning rate schedule
Initial Learning Rate	5×10^{-4}
Weight Decay	0

5. Experiments and Results

In this chapter, we first present the performance of SACNet in row and column detection and the overall performance of TSRDet in table structure recognition in Section 5.1. To demonstrate the effectiveness of SACNet's improvements over the baseline and the contribution of its key modules, we conduct ablation experiments in Section 5.2. Finally, in Section 5.3, we compare SACNet's row-column detection performance with five mainstream object detection models, all trained on the dataset introduced in Section 4.1, and further analyze the generated table structures.

5.1. Table Structure Recognition

In this study, we assess the recognition capabilities of our proposed method using the dataset we have compiled. We begin by employing SACNet to predict row and row bounding boxes, assessing its performance in row and row detection through mean Average Precision (mAP) across various Intersection over Union (IoU) thresholds. Subsequently, we apply the post-processing algorithm outlined in Section 3.4 to generate table structures, evaluating this method's effectiveness in recognizing table structures via IoU scores under different thresholds and TEDS-Struct. The experimental results indicate that the model achieves a TEDS-Struct score of 95.7% for table structure recognition. For a clearer presentation, the remaining evaluation metrics are listed in tables. Table 2, respectively, presents the model's mAP for row-column detection tasks, while Table 3 lists the IoU scores for recognizing table structures at various thresholds. Experimental results are depicted in Figure 9, showcasing three table images along with their corresponding bounding boxes.

Table 2. The quantitative evaluation results of row-column detection.

	Row			Column		
Minoverlap	**Class**	**AP**	**mAP**	**Class**	**AP**	**mAP**
0.5	merge normal	96.15% 97.61%	96.88%	merge normal	94.79% 99.54%	97.16%
0.55	merge normal	95.99% 97.07%	96.53%	merge normal	94.22% 99.48%	96.85%
0.6	merge normal	95.70% 96.25%	95.97%	merge normal	93.03% 99.38%	96.20%
0.65	merge normal	95.39% 94.93%	95.16%	merge normal	91.79% 99.15%	95.47%
0.7	merge normal	94.98% 92.57%	93.77%	merge normal	89.56% 98.84%	94.20%
0.75	merge normal	94.30% 88.49%	91.40%	merge normal	87.14% 98.31%	92.73%
0.8	merge normal	92.49% 82.25%	87.37%	merge normal	82.87% 97.23%	90.05%
0.85	merge normal	89.51% 74.36%	82.09%	merge normal	78.35% 94.93%	86.64%
0.9	merge normal	80.21% 53.43%	66.82%	merge normal	63.91% 88.24%	76.07%

Table 3. The quantitative evaluation results of table structure recognition.

Minoverlap	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9
IoU Score	0.963	0.959	0.955	0.946	0.932	0.912	0.862	0.796	0.589

Based on the data presented in Table 2, it is evident that SACNet can effectively and accurately predict the row-column bounding boxes in table images. Specifically, at an IoU threshold of 0.75, the mAP for row-column prediction still exceeds 91%. Furthermore, according to the data in Table 3, for the task of table structure recognition, TEDS-Struct achieves a remarkable score of 95.7%. Additionally, the IoU score reaches 0.91 at an IoU threshold of 0.75, demonstrating that our proposed TSRDet method performs well even under stringent localization accuracy requirements.

Furthermore, as can be observed from Figure 9, TSRDet achieves good results on complex tables. For instance, the three samples demonstrate their recognition capabilities when faced with complex tables that include large areas of blank cells and those without borders. The second sample also showcases its ability to recognize tables with significant column element misalignments. Overall, the method we propose is capable of effectively and robustly extracting the structures of various tables.

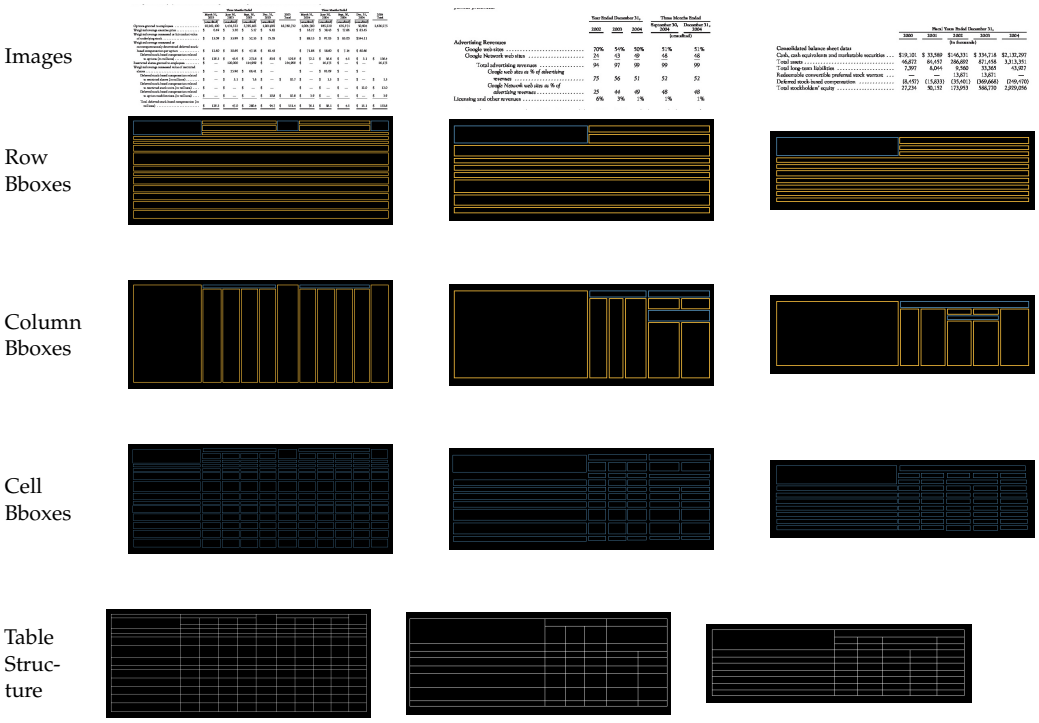


Figure 9. Some results of row-column detection and table structure recognition. Among them, the first row exposes multiple table images. The second and third rows show the row-column detection results, respectively. The fourth row displays the cell bounding box prediction results, and the last row reveals the visualization results of table structure.

5.2. The Ablation Study of SACNet

In this section, we conduct an ablation study on the architecture of the Table Structure Recognition Detector (SACNet) to verify its performance advantages. Specifically, through comparative experiments, we evaluate the contributions of the Swin Transformer (SwinT) backbone network and the RCAM to the task of detecting rows and columns in tables, thereby confirming the effectiveness of these components. Additionally, this research provides a detailed analysis of how integrating various functional modules within the standard model framework affects the performance of row-column detection, aiming to validate the efficacy of this architectural approach. To achieve these research objectives, we constructed three variant models and evaluated their performance on a custom dataset. These three models are the original CenterNet, CenterNet with the backbone replaced by Swin Transformer, and SACNet. The experimental results are presented in Table 4 and Table 5, respectively. For a clearer presentation and analysis of these data, we have displayed the results on the rows and columns using dot-line graphs, as illustrated in Figure 10.

Table 4. The ablation experimental results of row detection.

Models	**mAP_x(%)**								
	0.5	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**
Baseline	96.32	95.91	95.33	94.45	93.05	90.16	85.81	79.89	64.60
+SwinT	96.80	96.50	95.99	95.20	93.84	91.19	87.12	81.49	65.95
+SwinT+RCAM	96.92	96.63	96.23	95.29	93.94	91.40	87.37	82.09	66.82

Table 5. The ablation experimental results of col detection.

Models	mAP_x (%)								
	0.5	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**
Baseline	96.37	95.91	95.31	94.40	93.07	91.84	89.45	85.07	71.30
+SwinT	97.00	96.57	96.11	95.10	93.99	92.55	89.90	86.21	75.42
+SwinT+RCAM	97.16	96.85	96.43	95.47	94.20	92.73	90.05	86.64	76.07

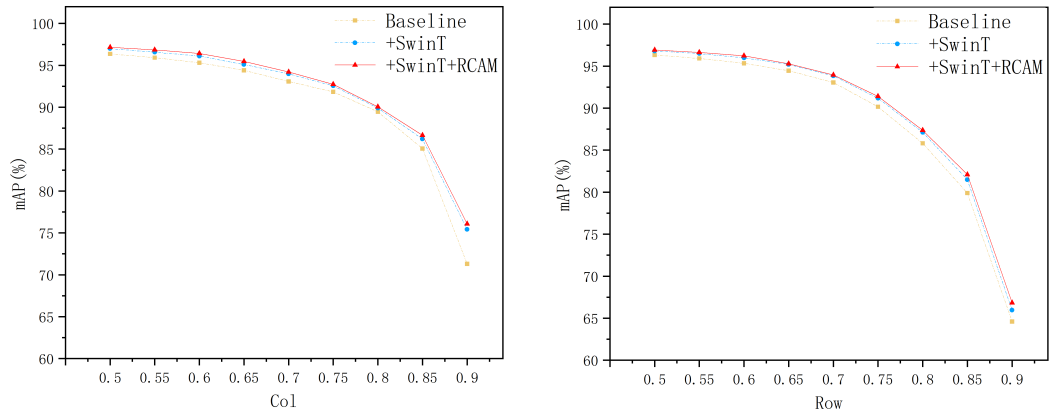


Figure 10. The visualization of ablation experiment results. Among them, left figure illustrates the mAP at different thresholds for three models in the column detection task, while right figure displays the row task. According to them, it is evident that replacing the backbone with the Swin Transformer equipped with RCAM is the optimal choice. Therefore, the structure of SACNet is reasonable.

According to the data in Tables 4 and 5, we observed that compared to the original baseline, replacing the backbone with a Swin Transformer significantly improved performance across detection thresholds from 0.5 to 0.9. At a threshold of 0.9, the mAP for column detection tasks increased from 71.30% to 75.42%, and for row detection tasks from 64.60% to 65.95%. After introducing the RCAM, the mAP further increased, reaching 76.07% for column detection and 66.82% for row detection at the threshold of 0.9. These findings confirm that the window-based self-attention and hierarchical structure in the Swin Transformer enhance the model's ability to extract global features. Additionally, the RCAM introduced at each level, with its row-column spatial attention mechanisms, allows the model to automatically select more critical feature areas, further reducing the interference of irrelevant information.

5.3. Comparison with Existing Object Detection Models

In this section, we compare the performance of TSRDet with several existing object detection models trained on our dataset. These models include YOLOv5-s [34], YOLOv8-s, YOLOx-s [41], DETR [42], and Faster R-CNN [32], and their computational complexity is listed in Table 6, measured in FLOPs (Floating Point Operations per Second). The YOLO series offers different size versions, including n, s, m, l, x, where the model size increases progressively controlled by depth, width, and max channels. In this study, the s-size versions are used. Unlike the YOLO series, DETR does not require predefined anchor priors or NMS, enabling end-to-end object detection. This Transformer-based detection network achieves state-of-the-art performance in detecting large objects. Faster R-CNN introduced the Region Proposal Network (RPN), which generates region proposals automatically through shared convolutional layers, further improving inference speed compared to Fast R-CNN. In this study, we evaluated the performance of various models and our proposed SACNet on the row and column detection task by comparing the mean average precision (mAP) values at different thresholds. The specific results are shown in Tables 7 and 8. To show these results more intuitively, Figure 11 depicts the table data in the form of a dotted line graph. In addition, we selected three table images to visualize the row and

column annotations on the original image as a benchmark. Subsequently, we plotted the row detection results, column detection results of each model, and the table structure derived after adding post-processing on the corresponding original image, as shown in Figure 12.

Table 6. The computational complexity of all models.

Models	Detr	Yolov5	Yolov8	Yolox	Faster RCNN	Ours
Total cost (GFLOPs)	31.951	46.3	12.175	65.915	909.765	63.091

Table 7. The experimental comparison results of column detection.

Models	**mAP_x (%)**								
	0.5	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**
Detr	25.09	18.40	13.18	9.50	6.61	4.31	2.85	1.87	0.70
Yolov5	94.26	90.43	85.88	81.14	74.56	65.30	57.42	50.17	43.06
Yolov8	96.26	95.13	93.73	92.08	87.84	79.17	70.19	56.75	45.25
Yolox	97.74	97.40	96.90	95.78	93.49	87.18	76.98	61.91	47.06
Faster RCNN	52.25	49.85	46.71	41.98	34.88	25.09	15.83	6.30	0.89
Ours	96.37	95.91	95.31	94.40	93.07	91.84	89.45	85.07	71.30

Table 8. The experimental comparison results of row detection.

Models	**mAP_x (%)**								
	0.5	**0.55**	**0.6**	**0.65**	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**
Detr	43.14	36.25	29.82	24.00	18.40	12.98	7.41	2.86	0.38
Yolov5	65.96	60.05	57.35	55.66	54.23	52.43	49.89	42.33	17.71
Yolov8	88.83	85.90	83.36	80.07	75.19	67.45	57.74	44.69	17.80
Yolox	97.00	96.42	94.50	91.41	85.57	76.95	64.48	49.70	22.80
Faster RCNN	12.48	11.60	10.70	9.80	8.69	7.36	5.44	2.58	0.40
Ours	96.32	95.91	95.33	94.45	93.05	90.16	85.81	79.89	64.60

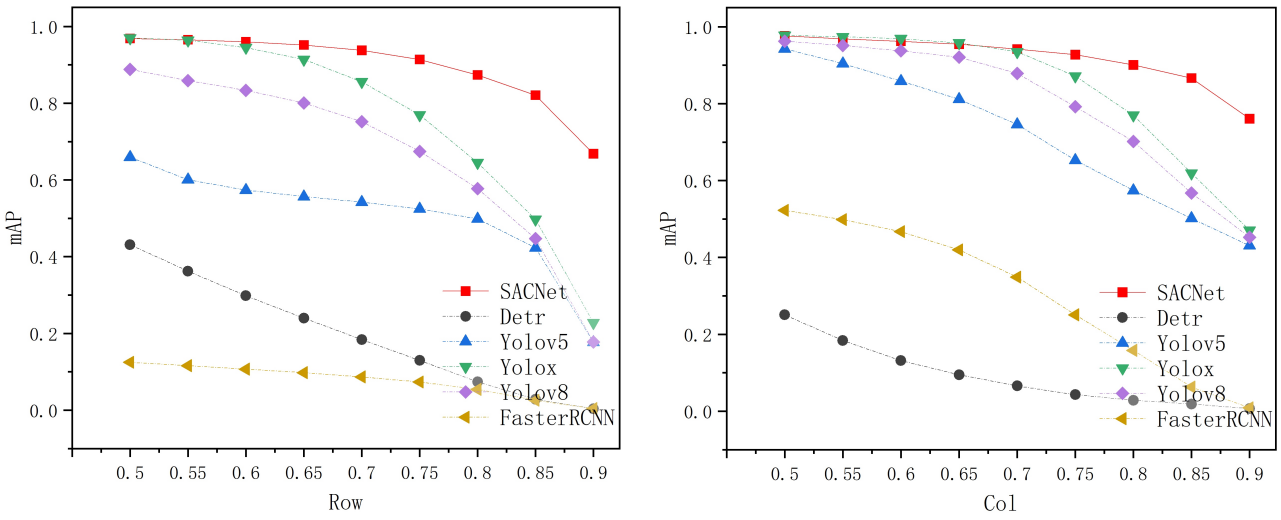
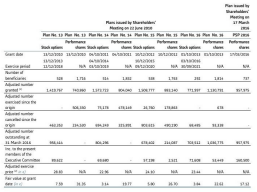
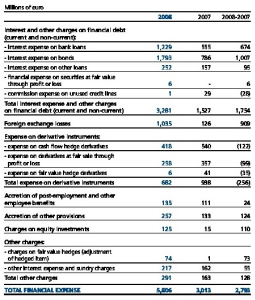


Figure 11. The Comparison of row-column detection between our method and five other models.

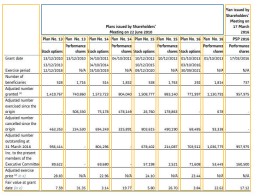
Source Images

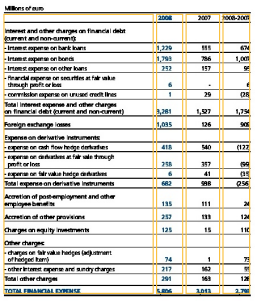






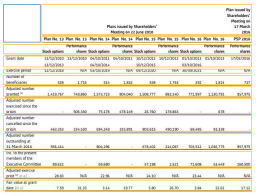
Col Label

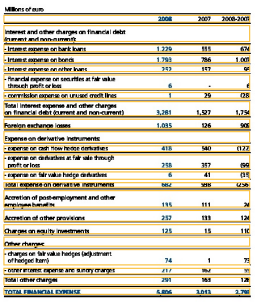






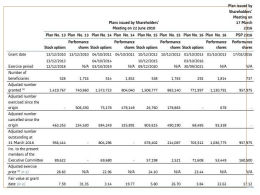
Row Label

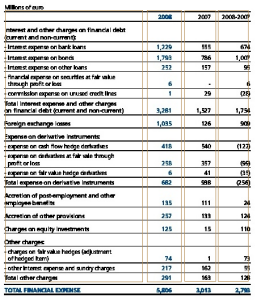


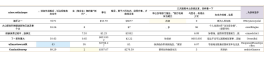




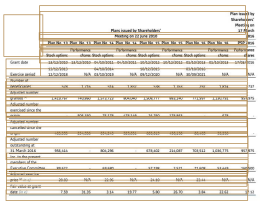
Col (DETR)







Row (DETR)



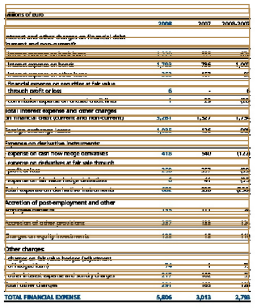
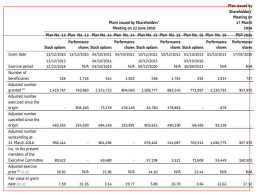
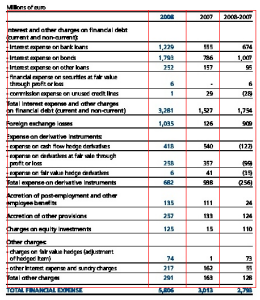




Figure 12. *Cont.*

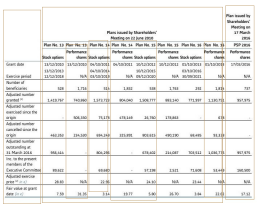
Result (DETR)

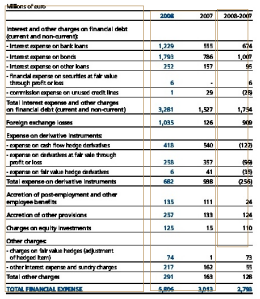






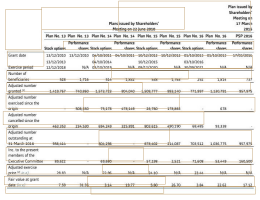
Col (Faster RCNN)

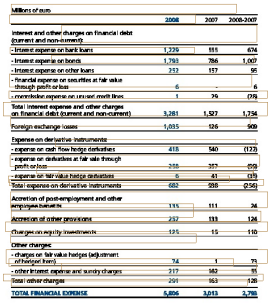






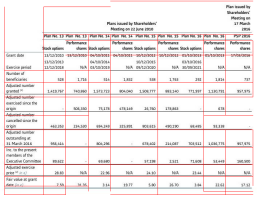
Row (Faster RCNN)

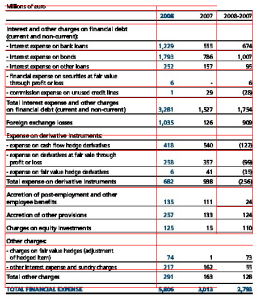






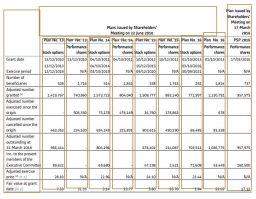
Result (Faster RCNN)







Col (Yolov5)



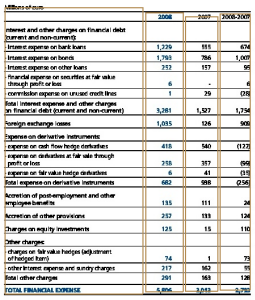
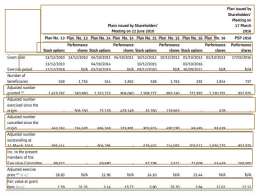
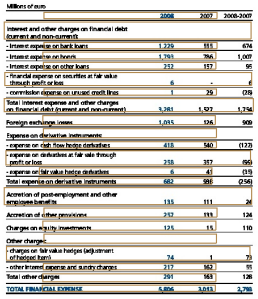




Figure 12. *Cont.*

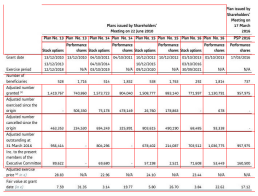
Row (Yolov5)

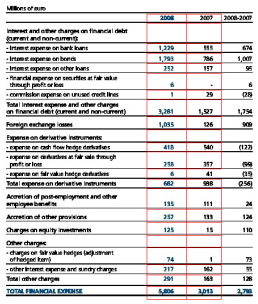






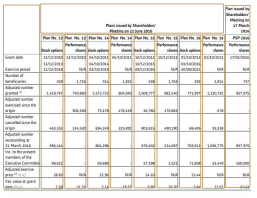
Result (Yolov5)

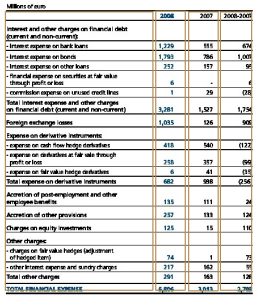






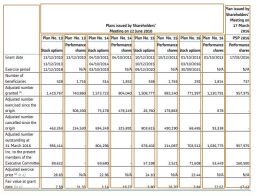
Col (Yolov8)

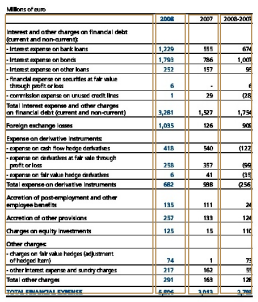






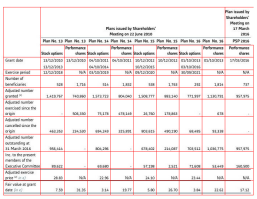
Row (Yolov8)







Result (Yolov8)



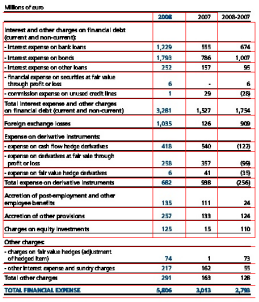
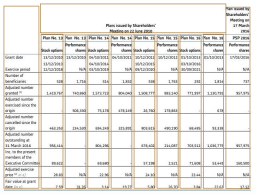
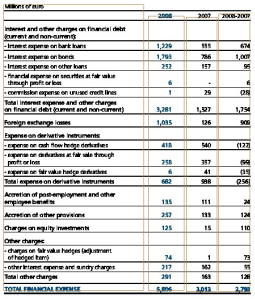




Figure 12. *Cont.*

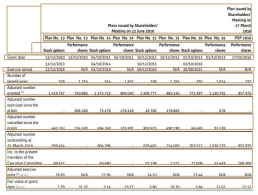
Col (Yolox)

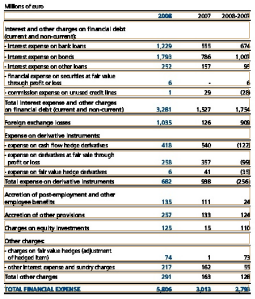






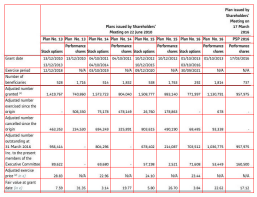
Row (Yolox)

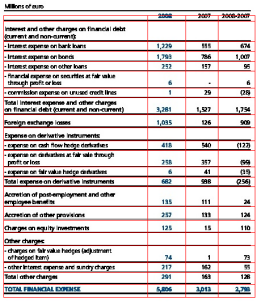






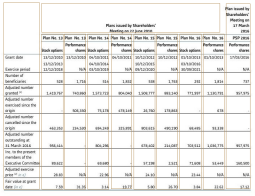
Result (Yolox)

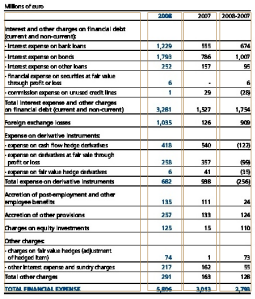






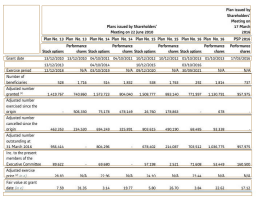
Col (Ours)







Row (Ours)



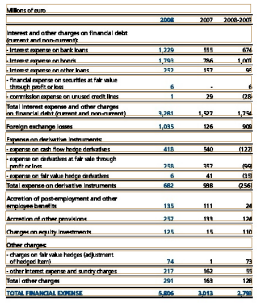
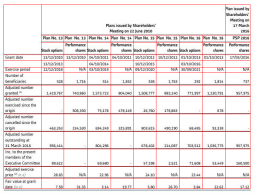




Figure 12. *Cont*.

Result (Ours)



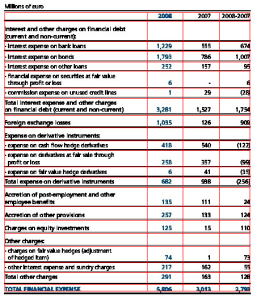




Figure 12. The comparison with existing detection models.

Based on the data in Tables 7 and 8, we can conclude that our proposed model demonstrates more consistent performance in row-column detection tasks compared to the other models. When the threshold is set to 0.75, our model maintains an mAP value of over 90%. In column detection, it outperforms YOLOv5-s by 27.43%, YOLOv8-s by 13.56%, YOLOx-s by 5.55%, DETR by 88.42%, and Faster RCNN by 67.64%. In row detection, it surpasses these five models by 38.97%, 23.95%, 14.45%, 78.42%, and 84.04%, respectively. Moreover, even at a stringent threshold of 0.9, our model still exhibits robust row-column detection performance. The mAP for column detection is 76.07%, exceeding the aforementioned models by 33.01%, 30.82%, 29.01%, 75.37%, and 75.18%. For row detection, the mAP is 66.82%, which is higher by 49.11%, 49.02%, 44.02%, 66.44%, and 66.42%, respectively. To more clearly demonstrate the improvements achieved by SACNet, we have illustrated the data from Tables 7 and 8 using dot-line graphs, as shown in Figure 11.

Additionally, as shown in Figure 12, SACNet exhibits the best detection performance, with almost no significant errors in the row or column bounding boxes. Compared to the existing models, issues such as overlapping adjacent objects and false positives/negatives are significantly reduced. Therefore, SACNet maintains superior detection performance, making TSRDet more advantageous for generating accurate table structures.

6. Discussion

This paper introduces TSRDet, a groundbreaking method for table structure recognition that leverages the capabilities of SACNet for detecting rows and columns, coupled with subsequent post-processing to elucidate the table structure. Notably, TSRDet employs the Swin Transformer and an innovative RCAM as encoders to enhance detection precision, effectively navigating the complexity and variability inherent in table structures. This methodology is particularly adept at capturing detailed and expansive feature maps, markedly boosting the accuracy of row and column detection. Experimental validation confirms TSRDet's superior performance, achieving a mean average precision (mAP) exceeding 91% at a stringent IoU threshold of 0.75. Moreover, TSRDet surpasses existing object detection models with a TEDS-Struct score of 95.7%, underscoring its competitive edge in table structure recognition. Overall, TSRDet significantly advances the field of image-based table recognition, offering a robust and precise solution for decoding complex table structures.

In addition, since the dataset used in this paper is a table image containing complex tables, such as tables with merged cells, missing lines, and wireless tables, it does not include table images with distortion and environmental factors, such as distorted tables and table images with different lighting, nor does it include images with multiple tables or multiple types of elements. This limits the application scenarios of TSRDet so that it can only process tables with complex structures but not images with too many interference factors and can only complete efficient and accurate table structure recognition after table area detection. Based on the current shortcomings, future work can be divided into the following three points:

- Further design and adaptation of methods can be conducted for distorted tables.

- More new table structure recognition methods can be studied, and efficient and useful modules can be used to further improve the universality and robustness of the method.
- Large language models can be used to further implement tasks such as fact verification and question answering of tables.

Author Contributions: All authors contributed to the study conception and design. Data collection and annotation analysis were prepared by Z.Z. and W.L. (Wei Li), methods were designed by Z.Z. and C.Y., experimental results visualization was completed by Z.Z., experimental design verification was completed by W.L. (Weibin Li), experimental supervision and leadership were completed by L.J., the first draft of the manuscript was written by Z.Z., and all authors commented on the previous version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: The support projects for this work include Xianyang City Key Research and Development Plan Project (2021ZDYF-GY-0031), Shaanxi Provincial Department and City Joint Key Project (2022GD-TSLD-61-3), Xi'an City Science and Technology Plan Project (23ZDCYTSGG0026-2022), Shanxi Coal Geology Group Co., Ltd. Scientific Research Project (SMDZ-2023CX-14), and Shaanxi Provincial Water Conservancy Development Fund Science and Technology Project (2024SLKJ-16).

Data Availability Statement: The datasets generated during and analysed during the current study are not publicly available due not authorized by the partner organization but are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; Ahmed, S. DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 1162–1167. [CrossRef]
2. 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 572–573.
3. Siddiqui, S.A.; Malik, M.I.; Agne, S.; Dengel, A.; Ahmed, S. DeCNT: Deep Deformable CNN for Table Detection. *IEEE Access* **2018**, *6*, 74151–74161. [CrossRef]
4. Paliwal, S.S.; Vishwanath, D.; Rahul, R.; Sharma, M.; Vig, L. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 128–133.
5. He, D.; Cohen, S.; Price, B.; Kifer, D.; Giles, C.L. Multi-Scale Multi-Task FCN for Semantic Page Segmentation and Table Detection. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 1, pp. 254–261. [CrossRef]
6. Reza, M.M.; Bukhari, S.S.; Jenckel, M.; Dengel, A. Table Localization and Segmentation using GAN and CNN. In Proceedings of the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), Sydney, Australia, 22–25 September 2019; Volume 5, pp. 152–157. [CrossRef]
7. Ma, C.; Lin, W.; Sun, L.; Huo, Q. Robust Table Detection and Structure Recognition from Heterogeneous Document Images. *Pattern Recognit.* **2023**, *133*, 109006. [CrossRef]
8. Baek, Y.; Nam, D.; Surh, J.; Shin, S.; Kim, S. TRACE: Table Reconstruction Aligned to Corner and Edges. In Proceedings of the International Conference on Document Analysis and Recognition, San José, CA, USA, 21–26 August 2023; pp. 472–489.
9. Zheng, X.; Burdick, D.; Popa, L.; Zhong, X.; Wang, N.X.R. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 697–706.
10. Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 2298–2304. [CrossRef] [PubMed]
11. Du, Y.; Chen, Z.; Jia, C.; Yin, X.; Zheng, T.; Li, C.; Du, Y.; Jiang, Y.G. Svtr: Scene text recognition with a single visual model. *arXiv* **2022**, arXiv:2205.00159.
12. Fang, S.; Xie, H.; Wang, Y.; Mao, Z.; Zhang, Y. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 7098–7107.
13. Lu, N.; Yu, W.; Qi, X.; Chen, Y.; Gong, P.; Xiao, R.; Bai, X. MASTER: Multi-aspect non-local network for scene text recognition. *Pattern Recognit.* **2021**, *117*, 107980. [CrossRef]

14. Xing, H.; Gao, F.; Long, R.; Bu, J.; Zheng, Q.; Li, L.; Yao, C.; Yu, Z. LORE: Logical location regression network for table structure recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 2992–3000.
15. Chi, Z.; Huang, H.; Xu, H.D.; Yu, H.; Yin, W.; Mao, X.L. Complicated table structure recognition. *arXiv* **2019**, arXiv:1908.04729.
16. Siddiqui, S.A.; Fateh, I.A.; Rizvi, S.T.R.; Dengel, A.; Ahmed, S. DeepTabStR: Deep Learning based Table Structure Recognition. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1403–1409. [CrossRef]
17. Tensmeyer, C.; Morariu, V.I.; Price, B.; Cohen, S.; Martinez, T. Deep Splitting and Merging for Table Structure Decomposition. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 114–121. [CrossRef]
18. Lin, W.; Sun, Z.; Ma, C.; Li, M.; Wang, J.; Sun, L.; Huo, Q. Tsrformer: Table structure recognition with transformers. In Proceedings of the 30th ACM International Conference on Multimedia, Lisbon, Portugal, 10–14 October 2022; pp. 6473–6482.
19. Guo, Z.; Yu, Y.; Lv, P.; Zhang, C.; Li, H.; Wang, Z.; Yao, K.; Liu, J.; Wang, J. Trust: An accurate and end-to-end table structure recognizer using splitting-based transformers. *arXiv* **2022**, arXiv:2208.14687.
20. Zhang, Z.; Zhang, J.; Du, J.; Wang, F. Split, embed and merge: An accurate table structure recognizer. *Pattern Recognit.* **2022**, *126*, 108565. [CrossRef]
21. Zhang, Z.; Hu, P.; Ma, J.; Du, J.; Zhang, J.; Yin, B.; Yin, B.; Liu, C. SEMv2: Table separation line detection based on instance segmentation. *Pattern Recognit.* **2024**, *149*, 110279. [CrossRef]
22. Siddiqui, S.A.; Khan, P.I.; Dengel, A.; Ahmed, S. Rethinking Semantic Segmentation for Table Structure Recognition in Documents. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 1397–1402. [CrossRef]
23. Xue, W.; Yu, B.; Wang, W.; Tao, D.; Li, Q. Tgrnet: A table graph reconstruction network for table structure recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Conference, 11–17 October 2021; pp. 1295–1304.
24. Raja, S.; Mondal, A.; Jawahar, C. Table structure recognition using top-down and bottom-up cues. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*; Springer: New York, NY, USA, 2020; pp. 70–86.
25. Huang, Y.; Lu, N.; Chen, D.; Li, Y.; Xie, Z.; Zhu, S.; Gao, L.; Peng, W. Improving Table Structure Recognition with Visual-Alignment Sequential Coordinate Modeling. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 11134–11143.
26. Ly, N.; Takasu, A.; Nguyen, P.; Takeda, H. Rethinking Image-Based Table Recognition Using Weakly Supervised Methods. In Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods, Lisbon, Portugal, 22–24 February 2023; SCITEPRESS-Science and Technology Publications: Lisbon, Portugal, 2023. [CrossRef]
27. Zhong, X.; ShafieiBavani, E.; Jimeno Yepes, A. Image-based table recognition: Data, model, and evaluation. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August, 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 564–580.
28. Ly, N.; Takasu, A. An End-to-End Multi-Task Learning Model for Image-based Table Recognition. In Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Lisbon, Portugal, 19–21 February 2023; SCITEPRESS-Science and Technology Publications: Setubal, Portugal, 2023. [CrossRef]
29. Ye, J.; Qi, X.; He, Y.; Chen, Y.; Gu, D.; Gao, P.; Xiao, R. PingAn-VCGroup's solution for ICDAR 2021 competition on scientific literature parsing task B: Table recognition to HTML. *arXiv* **2021**, arXiv:2105.01848.
30. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef] [PubMed]
31. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
32. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149.
33. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Lecture Notes in Computer Science*; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37. _2 [CrossRef]
34. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
35. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part V 13*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755..
36. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
37. Zhou, X.; Zhuo, J.; Krahenbuhl, P. Bottom-up object detection by grouping extreme and center points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 850–859.
38. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual Conference, 11–17 October 2021; pp. 10012–10022.

39. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
40. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
41. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
42. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.