Expert Systems With Applications 269 (2025) 126461

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Rethinking detection based table structure recognition for visually rich document images

Bin Xiao [a], Murat Simsek [a], Burak Kantarci [a,*], Ala Abu Alkheir [b]

[a] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, K1N 6N5, ON, Canada
[b] Lytica Inc., 555 Legget Drive, Ottawa, K2K 2X3, ON, Canada

ARTICLE INFO

Keywords:
Table structure recognition
Information extraction
Document processing
Object detection
Visually rich document understanding

ABSTRACT

Detection models have been extensively employed for the Table Structure Recognition (TSR) task, aiming to convert table images into structured formats by detecting table components such as Columns and Rows. However, prevailing detection-based TSR models usually cannot perform well regarding cell-level metrics, such as TEDS, and the reasons hindering their performance are not thoroughly explored. Therefore, we first examine the underlying reasons impeding these models' performance and find that the key issues are the improper problem formulation, the mismatch issue of detection and TSR metrics, the inherent characteristics of detection models, and the influence of local and long-range feature extraction. Based on these findings, we propose a tailored Cascade R-CNN based solution by introducing a new problem formulation, tuning the proposal generation, and applying deformation convolution and the proposed Spatial Attention Module. The experimental results show that our proposed model can improve the base Cascade R-CNN model by 19.32%, 11.56%, and 14.77% on the SciTSR, FinTabNet, and PubTables1M datasets regarding the structure-only TEDS, achieving state-of-the-art performance, demonstrating that our findings can serve as a valuable guide for enhancing detection-based TSR models. Our code and pre-trained models are public available[1].

1. Introduction

Portable Document Format (PDF) and scanned documents are commonly employed in business scenarios and on the internet because of their inherent readability for human users. However, these documents are typically not structured, posing a substantial obstacle to further information extraction and semantic analysis. Besides the unstructured format of these documents, tables in these documents, which are widely used to summarize critical information, often have very complex structures and layouts, making it challenging to interpret and analyze them. While some studies and tools (Mendes & Saraiva, 2017; Rastan, Paik, & Shepherd, 2019; Singer-Vine, 2022) can directly parse PDF files to extract the text content and tables, their performance remains limited in extracting the complex structures of tables, and they cannot deal with scanned documents and table images. Therefore, many studies convert PDF files into document images and apply deep learning models for Document Layout Analysis (Wu, Ma et al., 2023; Wu, Xiao et al., 2023), Table Detection (Xiao, Simsek, Kantarci, & Alkheir, 2023c; Yu, Huang, Luo, Zhang, & Lu, 2023), Table Structure Recognition (TSR) (Fernandes et al., 2023; Huang et al., 2023; Li, Yin, Dai and Liu, 2022; Ly & Takasu, 2023; Ma, Lin, Sun, & Huo, 2023; Nassar, Livathinos, Lysak, & Staar, 2022; Qiao et al., 2021; Xiao, Akkaya, Simsek, Kantarci and Alkheir, 2022; Xiao, Simsek, Kantarci and Alkheir, 2022; Zheng, Burdick, Popa, Zhong, & Wang, 2021; Zhong, ShafieiBavani, & Jimeno Yepes, 2020), and other document analysis tasks (Hu, Wang, Li, & Wang, 2021). In this study, we focus on the TSR task aiming to convert table images into structured formats, such as HTML sequences.

TSR studies can roughly be categorized into three groups based on their problem formulations: image-to-sequence models, graph-based models and detection-based models. Image-to-sequence models usually follow the encoder–decoder architecture and directly generate structured outputs, such as HTML sequences. Some image-to-sequence models (Ly & Takasu, 2023; Ye et al., 2021) also integrate the OCR task into the model to make the model end-to-end without using extra OCR tools (JaidedA, 2022; Kuang et al., 2021; Xiao, Akkaya, Simsek, Kantarci and Alkheir, 2023) to extract text contents from the images. However, since these models use auto-regressive decoders, they often suffer from error accumulation (Shen et al., 2023), and their OCR

* Corresponding author.
E-mail addresses: bxiao103@uottawa.ca (B. Xiao), murat.simsek@uottawa.ca (M. Simsek), burak.kantarci@uottawa.ca (B. Kantarci), ala_abualkheir@lytica.com (A.A. Alkheir).

[1] https://github.com/uobinxiao/CascadeTSRDet

https://doi.org/10.1016/j.eswa.2025.126461
Received 10 May 2024; Received in revised form 5 November 2024; Accepted 6 January 2025
Available online 9 January 2025

capacity usually cannot generalize well because of the limitation of training data. On the other hand, graph-based models usually use segmentation or detection methods to extract table cells, treat extract table cells as nodes of a graph, and further build the relation among the graph nodes. This graph-based definition makes it easier to deal with the scenarios in which table images are collected from the wild, such as rotated, distorted tables. However, graph-based models introduce extra complexity because they need to build extra graph models compared with detection-based models. By contrast, detection-based models are more straightforward in detecting the table components directly and post-processing the detection results with a deterministic rule-based method for reconstructing the table structure. However, detection-based methods can fail to deal with rotated and distorted samples. Besides, detection-based models usually cannot perform as well as other types of solutions regarding cell-level TSR metrics, such as TEDS (Zhong et al., 2020). Therefore, these different types of approaches have their benefits and must be selected based on the application scenarios. In this study, we focus on applying detection-based TSR models to process the table images from well-formatted documents.

There have been many studies (Fernandes et al., 2023; Hashmi, Stricker, Liwicki, Afzal, & Afzal, 2021; Siddiqui, Fateh, Rizvi, Dengel, & Ahmed, 2019; Smock, Pesala, & Abraham, 2022; Xiao, Akkaya et al., 2022) using detection models together with a post-processing method to solve the TSR task. However, existing studies either oversimplify the problem or define a multi-label detection task, which is challenging for two-stage object detectors. For example, some studies (Fernandes et al., 2023; Hashmi et al., 2021; Siddiqui et al., 2019; Xiao, Akkaya et al., 2022) do not define Column Header as detection target, making it impossible to provide information regarding the header cells. By contrast, PubTables1M (Smock et al., 2022) defines six types of components, including Table, Column, Row, Spanning cell, Column Header, and Projected Row Header, which can provide as much structure information as other types of TSR models. However, PubTables1M (Smock et al., 2022) does not consider that some Column Headers and Projected Row Headers can share identical bounding boxes with corresponding Rows, making this definition a multi-label detection task. Besides these issues of problem formulation, detection models used in these studies are trained to optimize their detection performance. However, since the complex structures of tables are processed by a post-processing step inferring defined table components, such as Columns and Rows, a model with good detection performance cannot necessarily lead to good performance in TSR metrics, such as TEDS. Moreover, some critical characteristics of detection models are not considered in the model design and problem formulation in existing studies. For example, typical two-stage detection models, such as Cascade R-CNN (Cai & Vasconcelos, 2018), are not suitable for multi-label detection tasks, while transformer-based detection models, such as DETR (Carion et al., 2020) and Sparse R-CNN (Sun, Zhang et al., 2021), can achieve promising results on multi-label detection tasks. Another example is that for two-stage detection models, regional proposal generation plays a crucial role in the model's performance, and the defined components in table images have different aspect ratios compared with common objects. At last, many studies apply deformable convolution (Dai et al., 2017) to improve the models' performance regarding detection evaluation metrics, such as COCO metric (Lin et al., 2014). However, simply applying deformable convolution can degrade the model's performance regarding the TEDS, and it is necessary to extract long-range dependencies while improving the local feature extraction. Therefore, in this study, we comprehensively revisit existing detection-based solutions and further explore the possible reasons hindering the performance of detection-based models for the TSR task. Based on our findings and analysis, we apply three simple methods to a typical two-stage detection model, Cascade R-CNN, including tuning the aspect ratios and increasing the number of region proposals in regional proposal generation, transforming the multi-label detection task into the single-label task, and introducing a Spatial Attention Module to build long-range dependencies. Fig. 1 presents the flow chart of our proposed method, including an input table image, the Object Detection Model, its outputs of six types of table components and the final results after post-processing. The details of our proposed solution are discussed in Sections 3 and 4. We conducted comprehensive experiments and the experimental results show that our proposed method can achieve state-of-the-art performance with very simple methods, demonstrating that our findings can be a guideline for further improvement of detection-based solutions.

1.1. Research objectives

As discussed, this study focuses on the detection-based solution for the TSR task, and many issues are limiting detection-based TSR solutions. Therefore, the research objectives of this study are three-fold: (1) This study explores and reveals the underlying reasons and factors impeding the performance of existing detection-based TSR solutions. (2) Based on our analysis and findings, this study builds a state-of-the-art detection-based solution for the TSR task regarding COCO metrics and structural-only TEDS. (3) This study discusses and summarizes the critical design aspects for the success of a detection-based TSR model based on the observations from the experimental results and analysis.

1.2. Contributions

The contributions of this study are four-fold:

1. We comprehensively revisit existing detection-based TSR models and explore possible reasons hindering the performance of these models, including the improper problem formulation, the mismatch issue of detection metrics and TSR metrics, the inherent characteristics of detection models, and the impact of feature extraction. Our analysis and findings can be a guideline for further improving the performance of detection-based TSR models.
2. Based on our analysis and findings, we apply three simple methods to improve Cascade R-CNN, including proposing a pseudo-class generation method to transform multi-label detection into a regular single-label detection problem, adjusting the ratio aspects and the number of regional proposals in the region proposal generation, applying the deformable convolution and introducing a Spatial Attention Module to build the long-range dependencies and context information in the backbone network.
3. We conduct extensive experiments to evaluate our proposed solution on various datasets, including SciTSR (Chi et al., 2019), FinTabNet (Zheng et al., 2021), PubTabNet (Zhong et al., 2020) and PubTables1M (Smock et al., 2022) with both detection metrics and cell-level TSR metrics. The experimental results show that our proposed solution can outperform state-of-the-art models in terms of detection and cell-level TSR metrics.
4. We further verify our analysis and findings with experiments and discuss and summarize valuable insights from the experimental results for further model design.

1.3. Article structure

Section 2 discusses related studies, including studies in Object Detection and Table Structure Recognition. Section 3 explores and discusses the reasons that hinder the performance of detection-based TSR models. Section 4 describes our proposed solution based on our analysis and findings. Section 5 shows the experiment settings and experimental results. Section 6 discusses the design aspects of the proposed method. At last, we draw our conclusion and possible directions in Section 7.

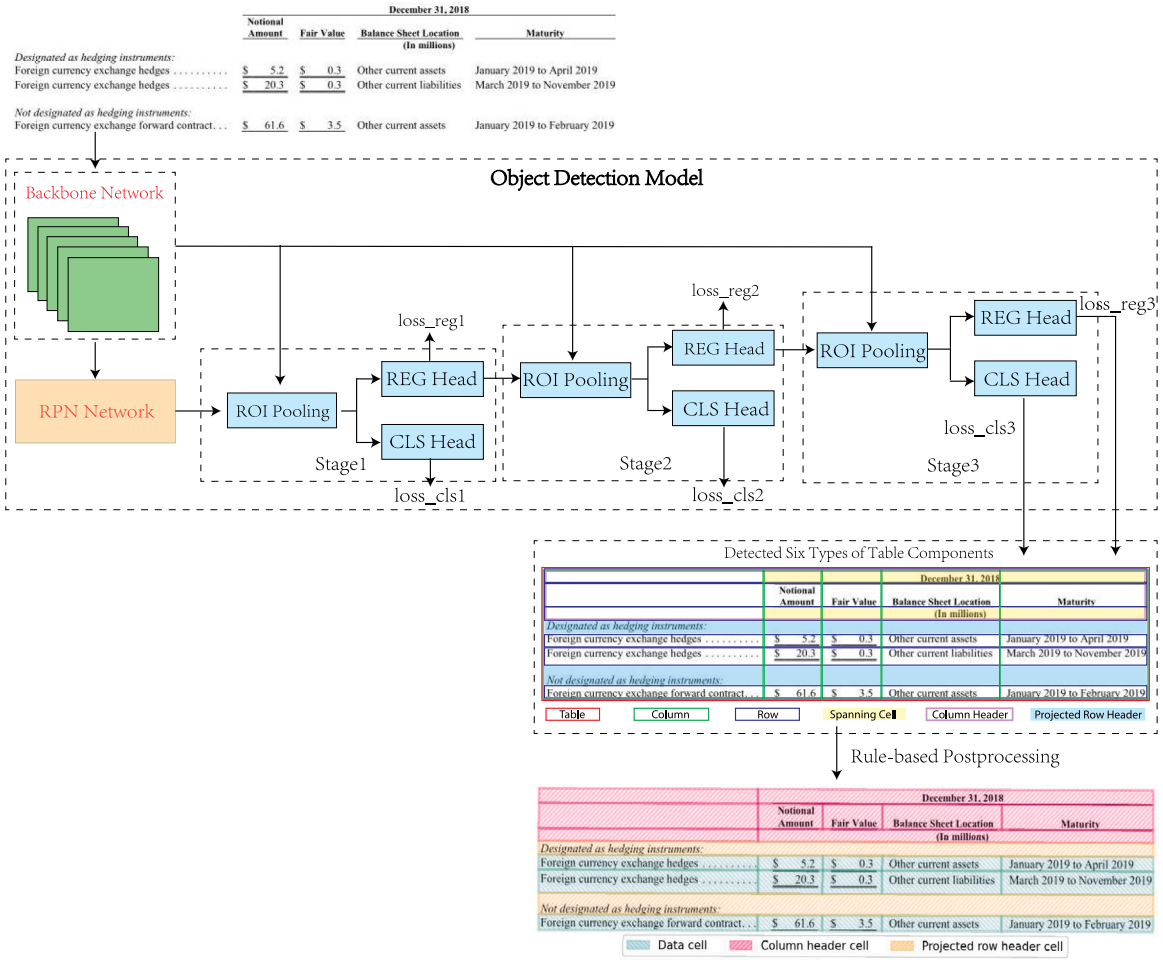


Fig. 1. Flowchart of a detection-based TSR solution.

2. Related work

2.1. Object detection models

Object Detection is a fundamental task that has been widely discussed in many studies. Since deep models have become the dominant solutions, we only discuss popular deep learning based models in this section. Based on different design perspectives, popular detection models can be categorized in different ways. One popular categorization of detectors based on the number of regression steps is to classify them into one- and two-stage detectors. Two-stage models, such as Cascade R-CNN (Cai & Vasconcelos, 2018), usually use a Region Proposal Network (RPN) to generate region proposals first and then feed the region proposals to the well-designed model to classify and regress the proposals. In the RPN network, one key parameter is the aspect ratio, which defines the height/width ratio when generating anchor boxes. Suitable aspect ratios are often close to the target objects' height/width ratio, making the regression task easier and improving the model performance. In contrast, popular one-stage models, such as FCOS (Tian, Shen, Chen, & He, 2019), YOLO series models (Bacea & Oniga, 2023; Li, Li et al., 2022; Wang, Bochkovskiy and Liao, 2023), integrate the region proposal generation and other regression and classification components into a single network. For example, YOLO series models divide the images into grids first, then classify the class of grid cells and directly predict the bounding boxes and their confidences. The simple design of one-stage detectors leads to faster training and inference time compared with two-stage detectors.

On the other hand, some studies (Sun, Zhang et al., 2021; Zhang et al., 2023) categorize the popular detectors from the perspective of Non-maximum Suppression (NMS), which is widely used to reduce the redundant predictions from the detectors. From this perspective, popular detectors can be categorized into end-to-end and none end-to-end models based on whether NMS is needed. DETR (Carion et al., 2020) is a typical end-to-end detector introducing transformer architecture (Vaswani et al., 2017), set prediction loss, and one-to-one label assignment to the object detection problem. Sparse R-CNN (Sun, Zhang et al., 2021) further refactors the DETR model and proposes to use sparse learnable regional proposals to replace dense regional proposals and utilize a dynamic instance interactive head to regress and classify the proposals in an iterative manner. Study (Sun, Jiang et al., 2021) analyzes the success of end-to-end detectors and argues that the one-to-one label assignment method in end-to-end detectors contributes to the success of the end-to-end models but is not sufficient to fully remove the NMS from the pipeline. This study further points out that the classification cost in the matching cost when applying one-to-one label assignment plays a key role in the success of these end-to-end models. Study (Zhang et al., 2023) further analyzes combinations of the label assignment methods and queries and argues that sparse queries with one-to-one label assignment can degrade the recall, and dense queries with one-to-one label assignment are hard to optimize. To address these issues, study (Zhang et al., 2023) proposes a dense distinct queries (DDQ) method to select distinct queries from dense queries using a class-agnostic NMS, achieving promising precision and recall. SQR (Chen et al., 2023) points out that the stages in DETR series detectors have different unbalanced responsibilities and proposes to collect and select intermediate queries for subsequent stages. It is worth mentioning that these end-to-end detector can easily be extended

to none end-to-end solutions by adapting many-to-one label assignments (Hong et al., 2022) and NMS. In this study, we refer to models using transformer architecture, set prediction loss, and their variations as transformer-based detection models, such as DETR, Sparse R-CNN and Deformable-DETR (Zhu et al., 2021).

2.2. Table structure recognition

There have been many studies (Adiga, Bhat, Shah, & Vyeth, 2019; Chi et al., 2019; Liu, Li et al., 2022; Schreiber, Agne, Wolf, Dengel, & Ahmed, 2017; Xiao, Akkaya et al., 2022; Zheng et al., 2021) discussing the TSR problem in recent years. As mentioned in Section 1, we can roughly categorize these solutions into image-to-sequence, detection-based, and graph-based models. Image-to-sequence based models usually define the ground truth as structured sequences, such as HTML sequences, built on the transformer architecture (Vaswani et al., 2017), and follow an encoder–decoder architecture. For instance, TableMaster (Ye et al., 2021) is a typical image-to-sequence based model that can generate HTML sequences. More specifically, TableMaster follows the architecture of MASTER (Lu et al., 2021), which is originally designed for the scene text generation following the transformer architecture (Vaswani et al., 2017), and further improved the encoder part by introducing a Multi-Aspect Global Context Attention. Besides, TableMaster has two branches designed for the HTML sequence generation and bounding box regression. Similarly, MTL-TabNet (Ly & Takasu, 2023) also follows the encoder–decoder architecture but contains three decoders for the cell box regression, cell content recognition, and HTML sequence generation, respectively. DRCC (Shen et al., 2023) argues that the error accumulation problem degrades the performance of image-to-sequence TSR models, especially when the input image is large. Therefore, DRCC proposes a two-step decoder architecture, which first decodes the input image into rows and then decodes the rows in cell sequences. VAST (Huang et al., 2023) pays more attention to the imprecise bounding boxes of table cells and proposes a Coordinate Sequence Decoder to improve the model's ability to generate accurate bounding boxes and introduces a visual-alignment loss to align the visual and structural information. To sum up, this type of method is usually based on the encoder–decoder architecture and can be trained end-to-end without using post-processing methods. Since the ground truth sequences used in image-to-sequence models usually contain information regarding spanning cells and header cells, these models can handle complex structures with spanning cells and identify header cells.

On the other hand, detection-based models usually define the problem as detecting different table components and applying a post-processing method to reconstruct table structures. DeepTabStR (Siddiqui et al., 2019) proposes to detect columns and rows to obtain the table cells. However, DeepTabStR ignores the row/column-span in the tables, which means that it cannot recover the hierarchical structures of tables. TableStrRec (Fernandes et al., 2023) extends the DeepTabStR, defining four types of table components: regular columns, irregular columns, regular rows, and irregular rows. Then, the spanning cells across multiple columns can be inferred from the difference between the regular and irregular columns when they are overlapped, and the spanning cells across multiple rows can be inferred from regular and irregular rows similarly. PubTables1M (Smock et al., 2022) is another typical detection-based approach that defines six table components: table, column, row, spanning cell, Projected Row Header, and Column Header, in which Projected Row Header and Column Header are for the function analysis, and other components can be used to reconstruct the complex table structure. Among these formulations, only the problem formulation of PubTables1M can provide as much information as image-to-sequence models because it can provide header cell information and reconstruct the complex table structure. Besides, these detection-based models need an extra deterministic rule-based post-processing method to infer the table structure from detected table components, meaning they are not end-to-end.

At last, graph-based methods usually apply either detection or segmentation methods to obtain the locations of table cells and further build the relation among table cells. For instance, TGRNet (Xue, Yu, Wang, Tao, & Li, 2021) formulates the cell location detection and cell logical location prediction jointly in a multi-task architecture, which is modularized by a segmentation based method and graph convolutional network (GCN), respectively. Similarly, TSRNet (Li, Yin et al., 2022) proposes a unified GNN-based approach modeling table detection and table structure recognition tasks together. More specifically, TSRNet also employs a semantic segmentation module to extract primitive regions, then applies k-nearest neighbors and line-of-sight neighbors to construct the graph and further classify the graph nodes and edges to filter the noise regions, merge, and build relations. In contrast, LGPMA (Qiao et al., 2021) proposes a Local Pyramid Mask Alignment Module and Global Pyramid Mask Alignment Module to localize table cells, which are formulated as detection and segmentation problems and can be implemented by MaskR-CNN (He, Gkioxari, Dollár, & Girshick, 2017). To construct the structure of the table, LGPMA further proposes a pipeline of cell matching, empty cell searching, and empty cell merging using the Maximum Clique Search algorithm and rule-based methods. Besides building graph explicitly, some studies (Nguyen, Le, Lu, Mai, & Tran, 2023; Tensmeyer, Morariu, Price, Cohen, & Martinez, 2019; Zhang, Zhang, Du, & Wang, 2022) predict the table grids or separators first, and then merge grid elements, which are also treating grid elements as graph nodes. SPLERGE (Tensmeyer et al., 2019) is a typical method following this strategy consisting of a Split Model and Merge Model, in which the Split Model consists of a Row Projection Network and a Columns Projection Network to obtain the table grid, and the Merge Model is used to merge the grid cells. Similarly, SEM (Zhang et al., 2022) employs a segmentation model to segment columns and rows and generate the table grid with a post-processing method. After the table grid is obtained, SEM introduces an Embedder network to extract and fuse the features from textual and visual modalities. A Merger network takes the fused features from Embedder as inputs to merge the grid elements. TSRFormer-DQ-DETR (Wang, Lin et al., 2023) leverages a DETR (Carion et al., 2020) based separation line prediction model, termed DQ-DETR, to predict the reference points on separation lines, followed by a Relation Network based cell Merging module to merge grid elements. RobustTabNet (Ma et al., 2023) employs a spatial network to predict Row and Column separation lines and further introduces a Grid CNN module to merge and build relations of table cells. Since these graph-based models identify the graph nodes first, defining a cell-type classification task is necessary if they want to provide information regarding header cells.

3. Rethinking detection-based TSR models

3.1. Preliminaries

Since most existing detection-based TSR models are based on two-stage and transformer-based detectors, we use Cascade R-CNN (Cai & Vasconcelos, 2018) and Sparse R-CNN (Sun, Zhang et al., 2021) as two examples of these two types of detectors and briefly review their critical designs in this section.

3.1.1. Cascade R-CNN

Cascade R-CNN (Cai & Vasconcelos, 2018) is a typical two-stage detection model containing a Backbone Network, a Region Proposal Network (RPN), and a series of Cascade Heads, as shown in Fig. 2. The RPN is the first regression step of a two-stage detection model responsible for generating region proposals. More specifically, a set of predefined anchor boxes are defined and slides across the feature map to generate the fix-length of feature vectors for the classification and regression tasks in the RPN (Ren, He, Girshick, & Sun, 2015). The classification task classifies anchor boxes into object and background, and the regression task coarsely regresses the anchor boxes to generate

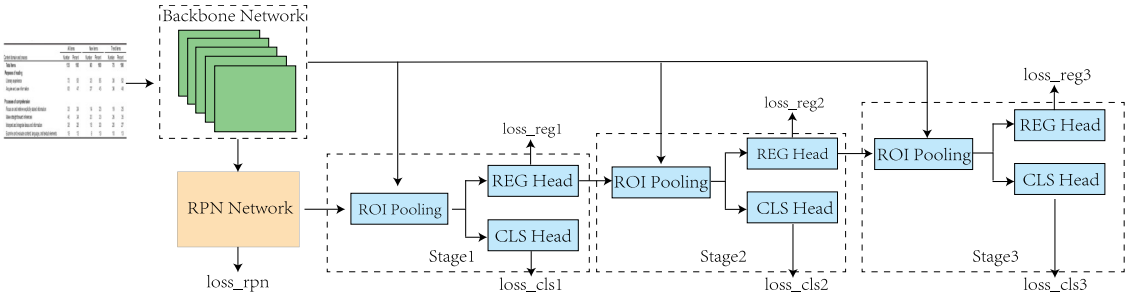


Fig. 2. Overall architecture of Cascade R-CNN.

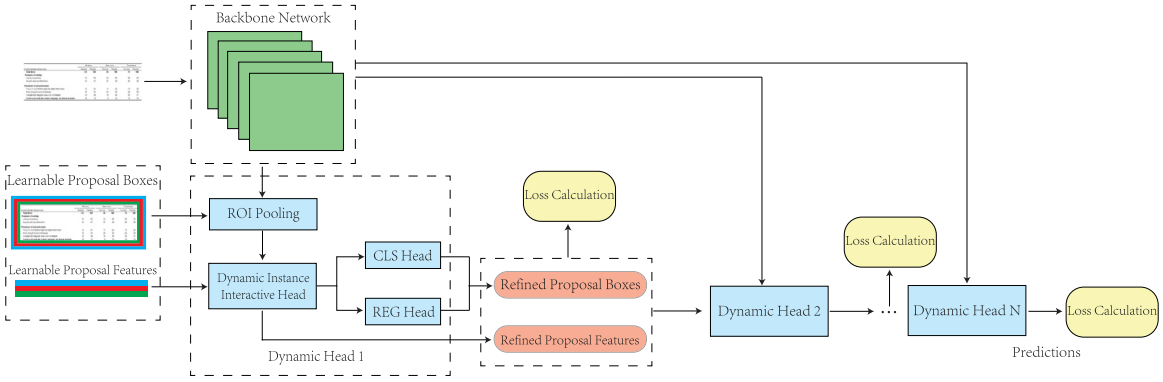


Fig. 3. Overall architecture of Sparse R-CNN.

higher-quality region proposals. Since the RPN only coarsely classifies and regresses the anchor boxes, the parameters of defining anchor boxes play a key role in the performance of the RPN, such as the number of anchor boxes, the aspect ratios of anchor boxes, and the scales of applied feature maps.

The Backbone Network is used to extract features of the input images, which is often followed by Feature Pyramid Network (FPN) (Lin et al., 2017) to extract and fuse features from different scales. The extracted features, together with the region proposals generated by the RPN, are fed into the first cascade head for the classification and regression tasks, and the regression results would be the inputs of the subsequent Cascade Head, as shown in Fig. 2.

Since there are multiple Cascade Heads, all the outputs of these Cascade Heads are used to calculate the loss. Moreover, the final loss of the model can be defined as the sum of these Cascade Heads loss and the RPN loss, as defined by Eq. (1), where N is the number of Cascade Heads. It is worth mentioning that we follow the most popular Cascade R-CNN model to show three Cascade Heads in Fig. 2. Each Cascade Head has a REG Head and a CLS Head for the regression and classification tasks, respectively. The input features of these REG Heads and CLS Heads e_{cls}, e_{reg} are extracted by applying ROI Pooling operations to the features from Backbone Network with the proposal boxes b, which can be defined by Eqs. (2) and (3) where $PROJ$, ROI_POOL, and $BACKBONE$ are the Projection layer, ROI Pooling operations, and the Backbone Network. Therefore, for a trained model, the input features of the CLS Heads e_{cls} are determined by the input image x and the proposal boxes b, meaning that a single proposal box cannot be classified into multiple classes because CLS Heads are not multi-label classifiers.

$$\mathcal{L} = \mathcal{L}_{rpn} + \sum_{i=1}^{N}(\mathcal{L}^i_{cls} + \mathcal{L}^i_{reg}) \tag{1}$$

$$e_{cls} = PROJ_{cls}(ROI\_POOL(BACKBONE(x), b)) \tag{2}$$

$$e_{reg} = PROJ_{reg}(ROI\_POOL(BACKBONE(x), b)) \tag{3}$$

3.1.2. Sparse R-CNN

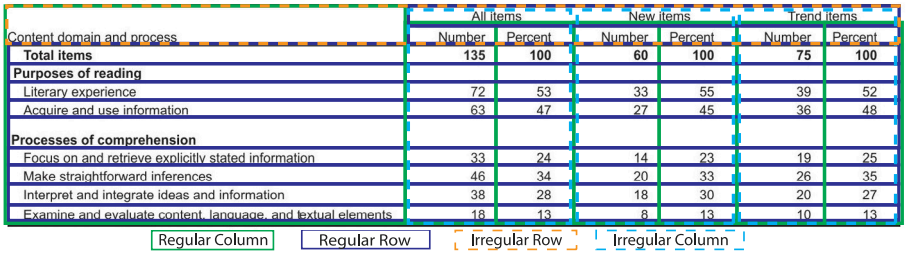
Sparse R-CNN is a popular end-to-end transformer-based detection model. Similar to Cascade R-CNN, Sparse R-CNN also employs a cascade architecture containing a series of Dynamic Heads, as shown in Fig. 3. In each Dynamic Head, an ROI Pooling layer is applied to extract features from the feature map based on the given proposal boxes, and the extracted features, together with the learnable proposal features, are fed to the Dynamic Instance Interactive Head to generate final features for the classification and regression tasks. Therefore, the features fed into CLS Head and REG Head of each Dynamic Head can be defined as Eqs. (4) and (5), where $BACKBONE$, DYN_HEAD, and $PROJ$ are the Backbone Network, Dynamic Instance Interactive Head and the Projection layer, respectively, and x, b, f are the input image, the proposal boxes and the learnable proposal features. It is worth mentioning that Sparse R-CNN does not use any RPN network to generate regional proposals. Instead, it proposes to use a set of learnable proposal boxes paired with a set of learnable features, in which learnable proposal boxes can be initialized by some pre-defined methods, such as image size initialization, random initialization, and grid initialization. Once the model is trained, the proposal boxes can be treated as an identical value, such as the box of image size, and their classification and regression results are mainly determined by their corresponding learnable proposal features f and the input image x. Therefore, for a multi-label detection problem, when objects belonging to different classes can share an identical box, the learnable proposal features can be different for these objects, making it possible for Sparse R-CNN to deal with multi-label detection tasks.

$$e_{cls} = PROJ_{cls}(DYN\_HEAD(ROI\_POOL(BACKBONE(x), b), f)) \tag{4}$$

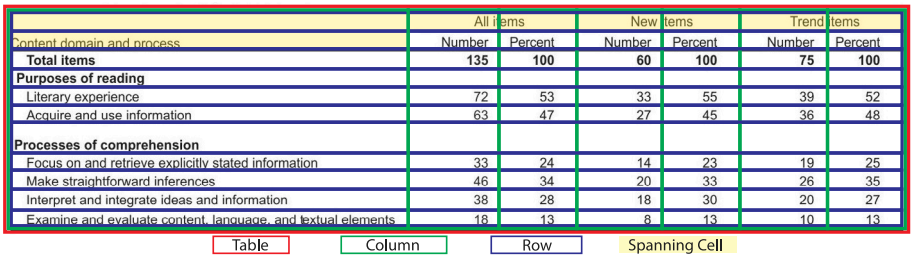
$$e_{reg} = PROJ_{reg}(DYN\_HEAD(ROI\_POOL(BACKBONE(x), b), f)) \tag{5}$$

3.2. Rethinking problem formulations

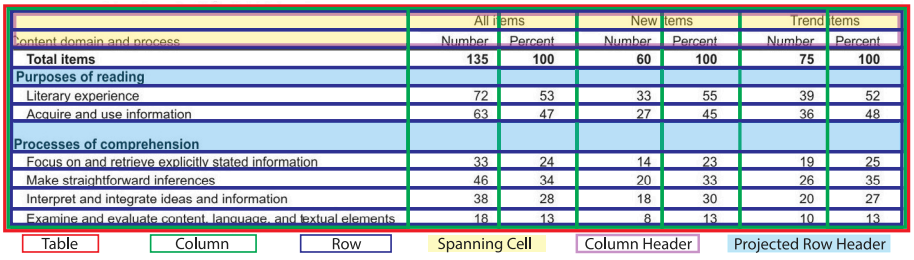
As aforementioned in Section 1, there have been many detection-based solutions (Fernandes et al., 2023; Hashmi et al., 2021; Siddiqui



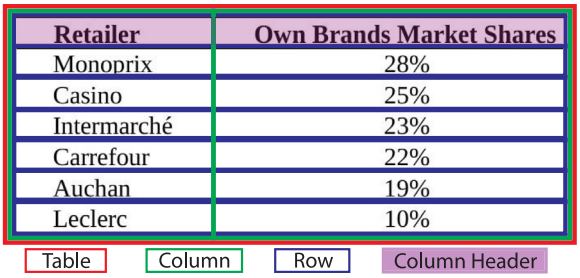
(a) Four types of defined table components in TableStrRec (Fernandes et al., 2023). This definition can infer Spanning Cells but cannot provide Column Header information. Besides, the Projected Row Headers are treated as regular Rows, which can lead to a wrong structure.



(b) Four types of defined table components in the study (Xiao et al., 2022a). This definition cannot provide Column Header information, and the Projected Row Headers are treated as regular Rows, which can lead to a wrong structure.



(c) Six types of defined table components in PubTables1M (Smock et al., 2022). The defined Projected Row Headers in this sample share identical bounding boxes with two corresponding rows.



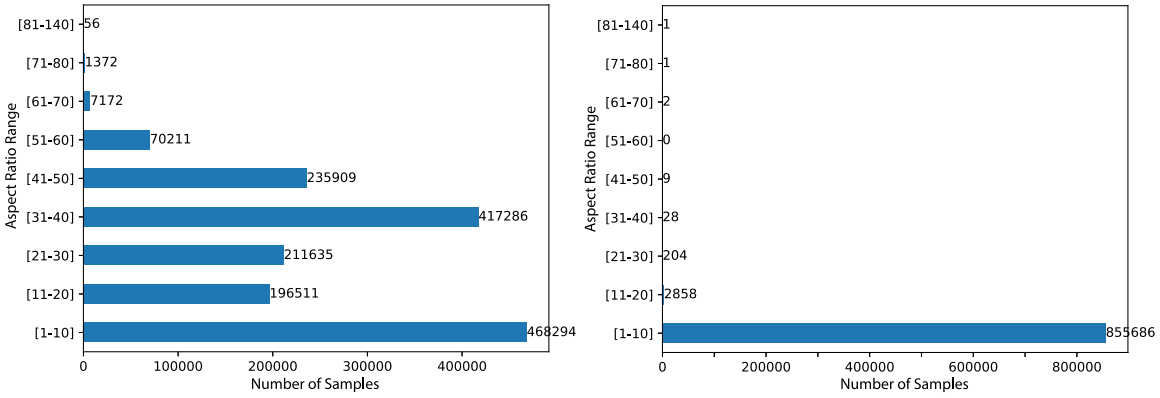
(d) Six types of defined table components in PubTables1M (Smock et al., 2022). In this sample, the defined Column Header shares an identical bounding box with a defined Row. It is worth mentioning that there are no Spanning Cell and Projected Row Header in this sample.

Fig. 4. Different problem formulations for the detection-base TSR.

et al., 2019; Smock et al., 2022; Xiao, Akkaya et al., 2022) with different problem formulations that either oversimplified the TSR task or ignored its multi-label characteristic. More specifically, following image-to-sequence TSR models, a detection-based TSR model should be able to fully reconstruct the structure of both regular and spanning table cells, as well as provide information regarding header cells. However, studies (Hashmi et al., 2021; Siddiqui et al., 2019) formulate the problem as only detecting columns and rows, making them impossible to deal with spanning cells and identify header cells. TableStrRec (Fernandes et al., 2023) further extend the formulation by defining regular column, regular row, irregular column, and irregular row so that the spanning cell can be inferred from these four types of components, as shown in Fig. 4(a). But still, this formulation cannot provide information regarding header cells, which is still over-simplified the TSR task. Study (Xiao, Akkaya et al., 2022) simplifies the formulation of in PubTables1M (Smock et al., 2022), directly detecting table, column, row, and spanning cell, as shown in Fig. 4(b), which is another formulation ignoring header cells' information. Besides, these three formulations treat the Projected Row Header as a regular row, resulting in over-simplified table structures. By contrast, PubTables1M (Smock et al., 2022) defines six types of components, including Table, Column, Row, Spanning Cell, Column Header, and Projected Row Header, as shown in Figs. 4(c) and 4(d), which can provide as

Table 1
Comparisons of different problem formulations.

Study	Detection Targets	Metrics	Outputs	Issues
Siddiqui et al. (2019) Hashmi et al. (2021)	Row/column	Precision Recall/F1	Regular cells	Information Loss
Fernandes et al. (2023)	Regular row/column Irregular row/column	F1/TEDS	Regular cells Spanning cells	Information Loss
Xiao, Akkaya et al. (2022)	Table/column Row/spanning cell	COCO	Regular cells Spanning cells	Information Loss
Smock et al. (2022)	Table/column Row/spanning cell Column header Projected row header	COCO GriTS	Header cells Spanning cells Projected row header Regular cells	Multi-label Detection
This study	Table/column Row/spanning cell Column header Projected row header	COCO TEDS	Header cells Spanning cells Projected row header Regular cells	–



(a) Aspect ratios of the FinTabNet training set.

(b) Aspect ratios of the COCO training set.

Fig. 5. Statistics of aspect ratio values of COCO and FinTabNet training sets. When an aspect ratio is less than 1, its multiplicative inverse counts the number of aspect ratios.

much structure information as image-to-sequence TSR models. However, this formulation does not consider that some Column Headers and Projected Row Headers can share identical bounding boxes with corresponding Rows. For example, as shown in Fig. 4(c), the bounding boxes of the two Projected Row Headers can also be classified as Rows. Similarly, as shown in Fig. 4(d), the Column Header's bounding box is also the Row's bounding box. Therefore, the problem definition of study (Smock et al., 2022) is a multi-label detection problem, which is must be considered when we choose and design detection models. It is worth mentioning that all these problem formulations use extracted table images as inputs. Even though many studies (Prasad, Gadpal, Kapadni, Visave, & Sultanpure, 2020; Xiao, Simsek, Kantarci, & Alkheir, 2023b; Xiao et al., 2023c) have achieved very promising performance on the Table Detection (TD) task, it is still difficult to guarantee that all the table content can be fully included in the detection results. Therefore, in practice, the detected bounding boxes of tables from the TD model are often padded with extra pixels, making it necessary to define a Table component for TSR. We summarize these problem formulations in Table 1 focusing on the detection targets, evaluation metrics, outputs after post-processing, and potential issues. The formulation in this study can avoid the issues of other formulations, including the information loss and the multi-label detection formulation.

3.3. Revisiting region proposal generation

As aforementioned in Section 3.1, the parameters of generating anchor boxes in the RPN play a key role in two-stage detection models, while transformer-based detection models, such as DETR and Sparse R-CNN, use learnable queries or proposals without the need to tune the RPN. If we choose two-stage detection models, such as Cascade R-CNN which is used in TableStrRec (Fernandes et al., 2023), we need to identify the difference between the TSR detection problem and widely discussed common object detection problem, because the default settings of detection frameworks, such as Detectron2 (Wu, Kirillov, Massa, Lo, & Girshick, 2019) and MMDetection (Chen et al., 2019) are often tuned on COCO (Lin et al., 2014) dataset. Therefore, we compare the statistics of the COCO dataset with a popular TSR dataset, FinTabNet (Zheng et al., 2021), regarding the number of objects in each image and the aspect ratios of objects. More specifically, the COCO training set contains 118 287 images and 860 001 target objects, resulting in an average of 7.27 objects in each image, while the FinTabNet training set contains 78 537 images, 1 628 298 target objects, resulting in an average of 20.73 objects in each image. Besides, the aspect ratios of objects in these two dataset are also very different, as shown in Fig. 5. The vast majority of target objects in the COCO training set have aspect ratios between 1 and 10, while objects in the FinTabNet training set have much larger aspect ratios. Therefore, we need to consider these differences when tuning the parameters of RPN if we apply a two-stage object detection model for the TSR task, such as increasing the number of region proposals and adjusting the aspect ratios of anchor boxes. On the other hand, transformer-based detection models, such as Sparse R-CNN and DETR, can alleviate the issues caused by these differences intrinsically because they use learnable queries (learnable proposals) instead of an RPN, as discussed in Section 3.1. However, increasing the number of learnable queries for each image might be also useful for transformer-based detection models because TSR datasets contain more objects than common object detection datasets.

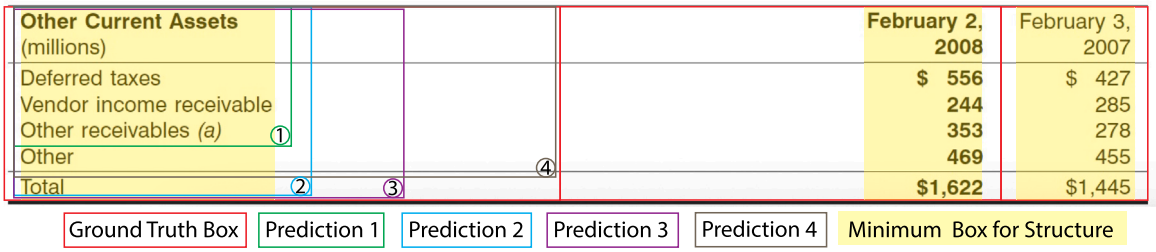


Fig. 6. A sample from the FinTabNet dataset with ground truth boxes larger than the minimum bounding boxes for table structure. We only show the annotations of Columns for simplicity.

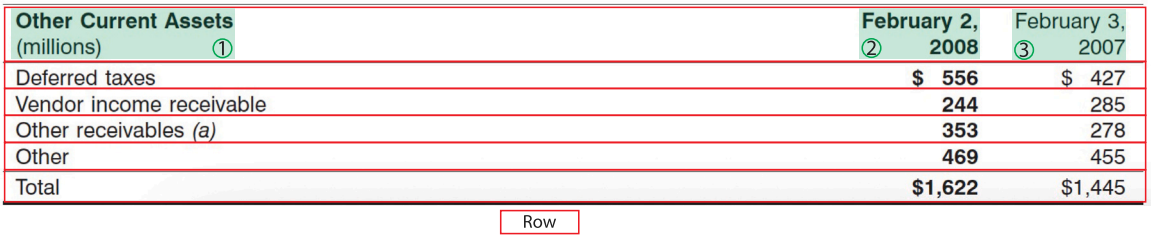


Fig. 7. A sample from the FinTabNet dataset. We only show its Row annotations for simplicity. The first Row in this Figure contains three major parts numbered 1 to 3.

3.4. Rethinking detection and TSR metrics

As mentioned in Sections 1 and 2, detection-based TSR models need a deterministic rule-based post-processing method to transform the detected table components into structured sequences. Existing studies (Fernandes et al., 2023; Hashmi et al., 2021; Siddiqui et al., 2019; Xiao, Akkaya et al., 2022) usually use the detection performance to evaluate the model performance before applying the post-processing method. However, the detection metrics are not aligned with cell-level TSR metrics. We use COCO (Lin et al., 2014) and TEDS (Zhong et al., 2020) metrics as examples for further analysis in this section. The COCO metrics employ mean Average Precision (mAP) to evaluate the model performance, which can be defined by Eq. (6) where N, $precision_i(r)$ and dr in Eq. (6) are the number of classes, and the precision at a given recall level r for class i. In practice, the precision–recall curves in COCO metrics are computed for each class at a series of IoU thresholds, and the integral of $precision_i(r)$ often is approximated by the discrete sum. The IoU score can be defined by Eq. (7), where $A \cap B$, $A \cup B$ are the intersection and union of bounding boxes A and B. In practice, in many studies, mAP is represented by AP and calculated by averaging the mean precision scores of all categories at IoU thresholds from 0.5 to 0.95 with 0.05 intervals. AP50, AP75 are the mean precision scores of all categorizes at IoU thresholds 0.5 and 0.75, respectively. Therefore, COCO metrics are IoU-based evaluation metrics. By contrast, TEDS can be defined by Eq. (8), where EditDist is the tree-edit distance, and T is the number of nodes in the tree, meaning that TEDS is not correlated with IoU scores.

$$mAP = \frac{1}{N}\sum_{i=1}^{N}\left(\int_0^1 \text{precision}_i(r)\,dr\right) \tag{6}$$

$$IoU = \frac{A \cap B}{A \cup B} \tag{7}$$

$$TEDS(T_a, T_b) = 1 - \frac{EditDist(T_a, T_b)}{max(|T_a|, |T_b|)} \tag{8}$$

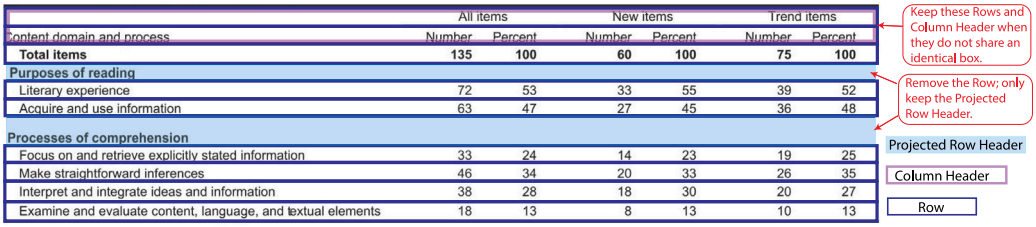
On the other hand, TSR datasets usually use a canonicalization procedure (Smock et al., 2022) or annotate the bounding boxes following the lines in tables, which makes the ground truth boxes larger than the minimum box that can recover the structure of the table. Fig. 6 shows an example from the FinTabNet dataset, whose ground truth boxes are larger than the minimum bounding boxes for table structure. Considering the four prediction boxes in Fig. 6, since the prediction 1 is smaller than the minimum box for table structure, and the prediction2 can cover all content of the minimum box for table structure and has a larger IoU with the ground truth box, prediction 2 can lead to better performance regarding both COCO and TEDS metrics than prediction 1. By contrast, prediction 3 has a larger IoU with the ground truth box than prediction 2, which can lead to better detection performance. However, when it comes to TEDS, prediction 3 cannot show any superiority compared to prediction 2, because both of them can cover the minimum box for table structure. When we compare prediction 2 and prediction 4, prediction 4 has a larger IoU with the ground truth box, making it better on detection performance, but it loses information of the row, making its performance in TEDS worse than prediction 2. Therefore, because of the definitions of COCO and TEDS metrics and the procedure of annotating datasets, a detection-based TSR model might be over-optimized towards detection performance without increasing the TEDS performance and sometimes can decrease the TEDS performance.

3.5. Rethinking feature extraction

As mentioned in Section 1, deformable convolution (Dai et al., 2017) has been applied in detection-based TSR (Fernandes et al., 2023; Siddiqui et al., 2019) and other related solutions (Mondal, Agarwal, & Jawahar, 2023; Siddiqui, Malik, Agne, Dengel, & Ahmed, 2018), demonstrating its effectiveness in improving detection performance. Deformable convolution uses a learnable grid offset to sample the grid points from the feature map and then apply the convolution operation to the sampled grid points, as defined by Eq. (9),

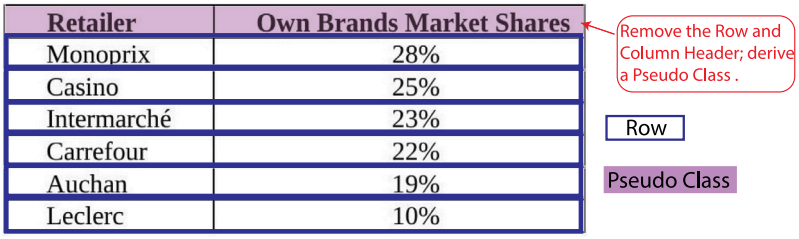
$$\mathbf{z}_{p_0} = \sum_{p_n \in R} w(p_n)x(p_0 + p_n + \Delta p_n) \tag{9}$$

where p_0 is the location on the output feature map $\mathbf{z}$, p_n is the nth grid point in grid R, and Δp_n is the nth learnable offset. Since the offset Δp_n applied to the deformable convolution is usually obtained by a regular convolution with small kernels, such as a 3 * 3 kernel, it can only improve the local feature instead of building long-range dependencies. However, building the long-range dependencies for the TSR task is important because of the characteristics of table components. More specifically, different parts of a single table component are often sparsely distributed across the table instead of a single area of compact pixels like common objects. Fig. 7 shows a sample with its Row annotations. Taking the first Row as an example, as shown in Fig. 7, it



Content domain and process	All items Number	All items Percent	New items Number	New items Percent	Trend items Number	Trend items Percent
Total items	**135**	**100**	**60**	**100**	**75**	**100**
Purposes of reading						
Literary experience	72	53	33	55	39	52
Acquire and use information	63	47	27	45	36	48
Processes of comprehension						
Focus on and retrieve explicitly stated information	33	24	14	23	19	25
Make straightforward inferences	46	34	20	33	26	35
Interpret and integrate ideas and information	38	28	18	30	20	27
Examine and evaluate content, language, and textual elements	18	13	8	13	10	13

(a) An example of our problem formulation. In this example, we remove two Rows because their bounding boxes are identical with two Projected Row Headers.



Retailer	Own Brands Market Shares
Monoprix	28%
Casino	25%
Intermarché	23%
Carrefour	22%
Auchan	19%
Leclerc	10%

(b) An example of our problem formulation. In this example, we derive a Pseudo Class because its bounding box simultaneously belongs to a Row and a Column Header.

Fig. 8. Examples of our proposed problem formulation. Since the definitions of Table, Column, and Spanning Cells are same with PubTables1M, only Row, Column Header and Projected Row Header are showed for simplicity.

mainly contains three parts, which are distributed sparsely, and there is a large space between the first part and the second part, even they all belong to a single target component. Therefore, it is important to build long-range dependencies together with improving local features, such as applying deformable convolution. And over-optimized local features, such as merely applying deformable convolution might degrade the performance regarding the TEDS.

4. Proposed method

In this section, we demonstrate how to fill the performance gap between detection-based and other types of TSR models by applying very simple methods to tailor the Cascade R-CNN model based on our analysis and findings in the previous sections. We first introduce the our problem formulation, then give the details of the proposed methods, including adjusting the parameters of the RPN, applying deformable convolution and introducing Spatial Attention Module.

4.1. Proposed problem formulation

As mentioned in Sections 3.1 and 3.2, the definition of PubTables1M (Smock et al., 2022) can provide as much information as other types of solutions and is a multi-label detection problem, which is challenging for two-stage detectors. Therefore, we follow PubTables1M to define six table components: Table, Column, Row, Spanning Cell, Projected Row Header, and Column Header, and transform the formulation into a single-class detection problem. More specifically, we remove the Rows that share their bounding boxes with the Projected Row Header, as shown in Fig. 8(a), and use a Pseudo Class to replace the Rows and Column Headers when they share identical bounding boxes, as shown in Fig. 8(b). It is worth mentioning that only the Row, Projected Row Header, and Column Header are shown because the Table, Column, and Spanning Cell are the same as PubTables1M. These two samples are also in Figs. 4(c) and 4(d), which show their original definition in PubTables1M.

Formally, the ground truth Y in PubTable1M's definition for each image is a set of tuples containing bounding boxes and their corresponding labels, as defined by Eq. (10), where b_i, c_i are the ith bounding box and its class, and values from 0 to 5 are the defined Table, Column, Row, Spanning Cell, Projected Row Header and Column Header, respectively.

$$Y = \{(b_i, c_i)\}_{i=1}^{N}, c_i \in \{0,1,2,3,4,5\}, \forall i \neq j, (c_i \neq c_j) \wedge ((b_i = b_j) \vee (b_i \neq b_j)) \tag{10}$$

By contrast, in this study, considering the observation that the defined Projected Row Headers are all Rows at the same time, we only keep the Projected Row Headers samples during the training. Since some Column Headers can share identical bounding boxes with corresponding Rows, we derive a pseudo class for these overlapped samples and remove the original overlapped samples. Therefore, during the training stage, we refactor the ground truth for each image to the regular single-label classification, as defined by Eq. (11), where values 0 to 6 are the Table, Column, Row, Spanning Cell, Projected Row Header, Column Header and the Pseudo Class, respectively. During the testing stage, the results of Project Column Header are duplicated once to generate their corresponding prediction Rows, and the results of the pseudo-class are duplicated twice to generate the corresponding prediction Rows and Headers, so that we can still follow the formulation defined by Eq. (10) to evaluate the model performance. Notably, we only apply this problem formulation to our tailored Cascade R-CNN model, and all other detection benchmark models are following the formulation of PubTables1M.

$$Y = \{(b_i, c_i)\}_{i=1}^{N}, c_i \in \{0,1,2,3,4,5,6\}, \forall i \neq j, (c_i \neq c_j) \wedge (b_i \neq b_j) \tag{11}$$

4.2. Tuning parameters of RPN

As mentioned in Sections 3.1 and 3.3, regional proposal generation is a critical step in two-stage detectors, which need to be carefully considered for the TSR problem. Therefore, we adjust Aspect Ratios and increase the number of generated regional proposals for our tailored model. More specifically, aspect ratios control the shape of the generated anchor boxes. Popular implementations of Cascade R-CNN, such as Detectron2 (Wu et al., 2019), usually use 0.5, 1.0, and 2.0 as default values, which can work well for detecting common objects, such as the objects in COCO (Lin et al., 2014) dataset. However, in the context of TSR, the range of aspect ratios is much larger because of the shape of the table components, as discussed in Section 3.3. Without proposing fancy new modules to select suitable values, we simply select the values based on the statistics of the training sets. Taking the FinTabNet dataset

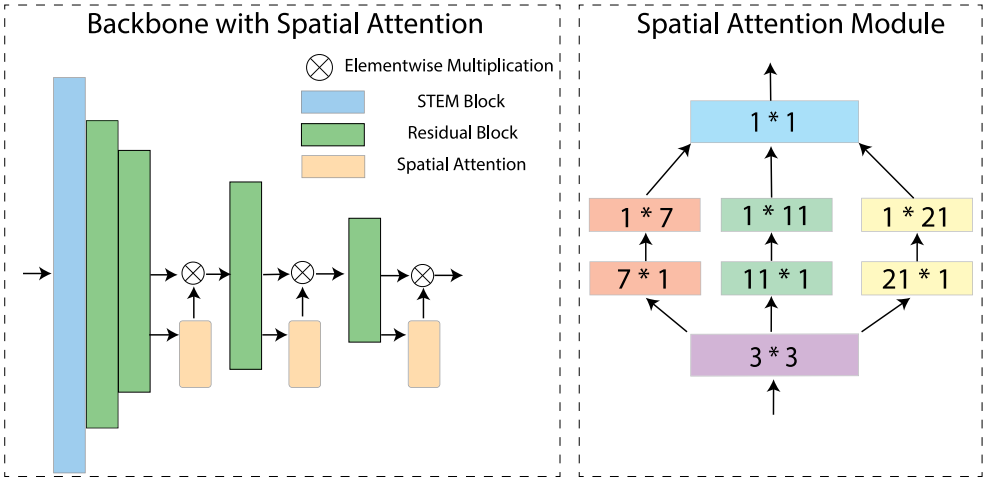


Fig. 9. Architecture of proposed Spatial Attention Module. A ResNet backbone consists of a STEM Block and four stages of Residual Block. Our proposed Spatial Attention Module are inserted between the blocks of the backbone to build long dependencies.

as an example, the aspect ratios of the defined components are shown in Fig. 5. The maximum value is 140, far larger than the popular choices in common object detection. Besides, the majority of aspect ratios in Fig. 5 are in the range between 1 and 60. Therefore, we extend this parameter for our proposed model as [0.0125, 0.025, 0.0625, 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0, 16, 40, 80]. A further detailed parameter table is provided in Section 5. It is worth mentioning that when an aspect ratio is less than 1, its multiplicative inverse is applied to count the number of aspect ratios in Fig. 5. We also did not further fine-tune this parameter through validation, which means that they might not be optimal. But this parameter has improved the model performance by around 2.7% as shown in Section 5.4. Besides, since increasing the number of proposals has been applied in existing studies (Fernandes et al., 2023), and demonstrated its effectiveness, we increase it for both the base Cascade R-CNN and our proposed model.

4.3. Spatial attention and deformable convolution

As discussed in Section 3.5, building long-range dependencies for detecting the defined components is important. Inspired by the recent studies using large convolution kernels (Ding, Zhang, Han, & Ding, 2022; Liu, Mao et al., 2022), we introduce a Spatial Attention Module for our solution, whose architecture is shown in Fig. 9. For the design of the Spatial Attention Module, we use a similar architecture with MSCA (Guo et al., 2022) containing multiple branches and large kernel convolutions and use spatial and depthwise separable convolution (Chollet, 2017; Howard et al., 2017) to reduce the number of parameters. More specifically, for the spatial separable convolution, we use a pair of $7 * 1$ and $1 * 7$ kernels to replace a typical $7 * 7$, use the pair of $11 * 1$ and $1 * 11$, and the pair of 21×1 and 1×21 to replace $11 * 11$ and $21 * 21$ kernels, respectively. For the depthwise separable convolution, we applied the convolution on each channel of the feature maps independently. Then, the outputs of the three branches are concatenated together as the input of a convolution layer with 1×1 kernel to make the channel dimension the same as the inputs. The proposed Spatial Attention Module can be easily inserted into the Backbone Network between two blocks because they do not change the feature shapes. For example, for a typical backbone network implemented by ResNet (He, Zhang, Ren, & Sun, 2016) containing a STEM block and four Residual Blocks, as shown in Fig. 9, the Spatial Attention Module can be inserted after the last three Residual Blocks to generate the spatial attention, then the spatial attention can be applied to the original outputs of each Residual Block by Element-wise Multiplication. It is worth mentioning that the Spatial Attention Module shown in Fig. 9 have independent trainable parameters, and all the feature maps are padded correspondingly to keep the size of the feature maps.

On the other hand, as discussed in Section 3.5, deformable convolution (Dai et al., 2017) can improve the local feature generation and has been demonstrated to help improve the detection performance on the document image detection tasks by many studies (Fernandes et al., 2023; Mondal et al., 2023; Siddiqui et al., 2018). Therefore, in this study, we apply the proposed Spatial Attention Module and deformable convolution to build long-range dependencies and improve local features together.

5. Experiments

5.1. Datasets and experimental settings

We utilize four datasets in this work, including SciTSR (Chi et al., 2019), FinTabNet (Zheng et al., 2021), PubTabNet (Zhong et al., 2020) and PubTables1M (Smock et al., 2022). As discussed in the study (Smock, Pesala, & Abraham, 2023), FinTabNet and SciTSR datasets contain noise annotations that harm the model performance. Therefore, we use their cleaned versions proposed in the study in Smock et al. (2023). Each image sample in these four datasets contains only a table with extra padding pixels to ensure the entire table is extracted. The SicTSR dataset is collected from academic publications containing 7453, 1034, and 2134 samples for training, validation, and testing. PubTables1M dataset is a large-scale dataset for the TSR problem collected from the PMCOA corpus, containing 758 849 training samples, 94 959 validation samples, and 93 834 testing samples. Since the PubTabNet dataset does not provide original PDF files, we cannot process it to make detection annotations. Besides, its testing is not publicly available. Therefore, we use its validation set to evaluate the model trained with the PubTable1M dataset. Following the study in Smock et al. (2023), we use the code base in Smock and Pesala (2021) to process the datasets and align the formats of these datasets. FinTabNet is also a large dataset widely used for the TSR problem, containing 78 537, 9289, and 9650 samples for training, testing, and validation. FinTabNet is collected from the annual reports of companies, making its data source different from the other datasets. Table 2 summarizes the datasets used in this study for the model evaluation.

Since the TSR problem in this study is formulated as an object detection problem, we use both detection and cell-level TSR metrics for the model evaluation. For the detection metric, we employ the widely accepted COCO metrics (Lin et al., 2014), which has been discussed in

Table 2
Summary of datasets.

Dataset	Train	Validation	Test
SciTSR (Chi et al., 2019)	7,453	1,034	2,134
FinTabNet (Zheng et al., 2021)	78,537	9,650	9,289
PubTabNet (Zhong et al., 2020)	500,777	9,115	–
PubTables1M (Smock et al., 2022)	758,849	94,959	93,834

Section 3.4. More specifically, mean Average Precision (mAP), $AP50$, $AP75$, AP_s, AP_m, AP_l, and object-specific AP scores are used as metrics, where $AP50$, $AP75$ are the APs using 0.50 and 0.75 as IoU thresholds, respectively. AP_s, AP_m, and AP_l are the APs of different target object sizes, defined by Eq. (12).

$$object\_size = \begin{cases} small & \text{if area} < 32^2 \text{ px} \\ medium & \text{if } 32^2 < \text{area} < 64^2 \text{ px} \\ large & \text{otherwise} \end{cases} \tag{12}$$

For the TSR metric, we choose structure-only Tree-Edit-Distance-Based Similarity(TEDS) (Zhong et al., 2020), which is firstly introduced in the study to overcome the drawbacks of adjacency relation metrics, and can be defined as Eq. (8) as discussed in Section 3.4. We use structure-only TEDS in the study, which only considers the HTML tags without extracting their contents to avoid the influence of OCR tools. The testing samples can be categorized into simple and complex groups based on whether they have cells spanning multiple columns and rows. Notably, for the evaluation of detection performance, we use the formulation defined in Eq. (10), which is also the problem definition of PubTables1M (Smock et al., 2022), and the results generated by our single-label detection can be easily transformed into the multi-label detection results defined by PubTables1M (Smock et al., 2022), as discussed in Section 4.1.

5.2. Implementation details and experimental results

To verify the effectiveness of our proposed solution, we include three state-of-the-art detection-based methods as benchmarks, including Cascade R-CNN (Cai & Vasconcelos, 2018), Deformable-DETR (Zhu et al., 2021) and Sparse R-CNN (Sun, Zhang et al., 2021), in which Cascade R-CNN (Cai & Vasconcelos, 2018) is also the based model of the proposed methods, Deformable-DETR and Sparse R-CNN are two state-of-the-art transformer-based detection models.

We implement Cascade R-CNN and our proposed method based on the Detection2 (Wu et al., 2019), the Deformable-DETR based on detrex (Ren et al., 2023), and the Sparse R-CNN with their official codebase. For the Deformable-DETR and Sparse R-CNN, we use their default parameters. For the Cascade R-CNN baseline, we align the number of regional proposals and the batch normalization method to the TSRDet, as shown in Table 3. All these detection models are using ResNet50 (He et al., 2016) pre-trained with ImageNet (Deng et al., 2009) as the backbone network. We also re-train TableMaster (Ye et al., 2021) with the FinTabNet dataset based on its official code base. We term our proposed method with TSRDet for fast reference. For the implementation of the proposed TSRDet, aspect ratios in the anchor box generation are set as [0.0125, 0.025, 0.0625, 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0, 16, 40, 80], and other key parameters are summarized in Table 3. Notably, to calculate the structure-only TEDS, we use the scripts provided by study in Smock and Pesala (2021) to generate the HTML sequences from the detected components, and all benchmark models, except our proposed model, are using the original definition of PubTables1M, which treats all table components independently with its multi-label detection setting. All the models are trained with 240, 120, and 60 epochs for the SciTSR, FinTabNet, and PubTables1M datasets, respectively. We also used regularization methods to mitigate overfitting issues, including data augmentation, weight decay (Krogh & Hertz, 1991), batch normalization (Ioffe, 2015), and gradient clipping (Pascanu, 2013). Specifically, for data augmentation, we employ a ResizeShortestEdge (Wu et al., 2019) method, which scales the shortest edge of an image while maintaining the original aspect ratio. We defined a range of shortest edge targets as [80, 160, 320, 640, 672, 704, 736, 768, 800, 1000], and the targets are randomly selected during the training to augment each input image, while we set the shortest edge target as 800 during testing. The Weight Decay is a parameter applying the L2 regularization for model parameters, which is set to 0.0001 in our implementation. We set the maximum allowed value for the gradient clipping as 1.0 for the L2 gradient norm. Finally, the batch normalization layers reside in the backbone network.

The experimental results regarding the structure-only TEDS and COCO metrics are shown in Tables 4 and 5, which can demonstrate the superiority of the proposed solution. For the SciTSR dataset, the proposed TSRDet can improve the baseline Cascade R-CNN by 19.32% regarding the structure-only TEDS, outperforming Deformable-DETR and Sparse R-CNN. When it comes to COCO metrics, the mAP of the proposed TSR is as good as Deformable-DETR, outperforming other benchmark models. Similarly, the proposed TSRDet can also outperform benchmark models regarding both COCO metrics and structure-only TEDS on the FinTabNet and PubTables1M datasets. Fig. 10 shows a prediction sample and its generated HTML sequence after post-processing, demonstrating the capacities of the proposed solution.

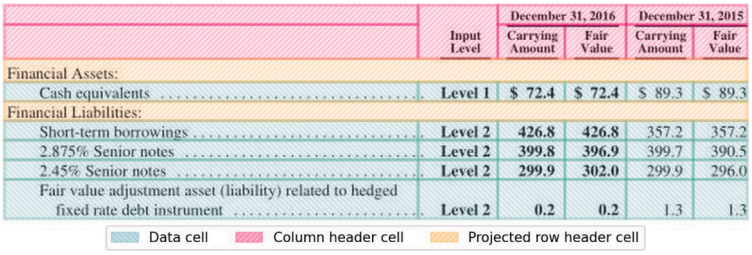
5.3. Comparison with non-detection-based models

As discussed in Sections 1 and 2, image-to-sequence and graph-based models are another two types of solutions for the TSR problem. Therefore, we compare our proposed solution with these two types of solutions in this section. Specifically, for the image-to-sequence models, EDD (Zhong et al., 2020), TableFormer (Nassar et al., 2022), TableMaster (Ye et al., 2021), VAST (Huang et al., 2023) and MTL-TabNet (Ly & Takasu, 2023) are included. TSRFormer-DQ-DETR (Wang, Lin et al., 2023) and RobustTabNet (Ma et al., 2023) are two state-of-the-art models following the pipeline of detecting separation lines and then merging cell grids, which can be treated as a graph-based model as discussed in Section 2. TSRNet (Li, Yin et al., 2022) is also a graph-based methods which detect table cells first, then applies GNN to build the relations among the detected cells. Since FinTabNet and PubTabNet are the most widely used datasets for these non-detection-based models, we report the experimental results of these two datasets in Tables 6 and 7. It is worth mentioning that the PubTabNet dataset does not provide original PDF files, making it hard to generate detection annotations. Therefore, the model performance reported in Table 7 is calculated using the model trained with the PubTable1M dataset. Although the PubTable1M and PubTabNet datasets have misalignments regarding the ground truth HTML sequences, the proposed method still shows competitive performance compared with other state-of-the-art methods, as shown in Table 7. Our proposed method can also outperform non-detection-based models regarding structure-only TEDS on the FinTabNet dataset, as shown in Table 6.

Despite the superior performance of the FinTabNet and PubTabNet datasets, the proposed detection-based solution has some inherent limitations compared to the other two types of solutions. First, it is not suitable for dealing with tables with irregular shapes, such as distorted tables, because the predicted boxes of a typical object detection model are rectangles. Second, a detection-based model needs to work with an OCR tool to extract the text content from a table image, while image-to-sequence models usually combine the text extraction task with the TSR task. Therefore, our proposed solution can be better than other types of solutions when tables are well-formatted without distortions and OCR tools or PDF parsing libraries can easily extract their text content.

Table 3
Key training parameters of the proposed model. MAX_ITER and STEPS are for the FinTabNet dataset as examples.

Parameter	Value	Description
RESNETS.NORM	nnSyncBN	Batch Normalization for the Backbone Network
MAX_ITER	112,500	Total number of mini-batch
STEPS	84,375	The mini-batch to apply the learning rate schedule
SCHEDULER	MultiStepLR	The scheduler to change the learning rate
NMS_THRESH	0.9	Non-maximum suppression threshold
PRE_NMS_TOPK_TRAIN	4000	RPN proposals to keep before applying NMS in training
PRE_NMS_TOPK_TEST	2000	RPN proposals to keep before applying NMS in testing
POST_NMS_TOPK_TRAIN	4000	RPN proposals to keep after applying NMS in training
POST_NMS_TOPK_TEST	2000	RPN proposals to keep after applying NMS in testing
DEFORM_ON_PER_STAGE	[True, True, True, True]	Whether to use deformable convolution in backbone stages



	Input Level	December 31, 2016 Carrying Amount	December 31, 2016 Fair Value	December 31, 2015 Carrying Amount	December 31, 2015 Fair Value
Financial Assets:					
Cash equivalents	Level 1	$ 72.4	$ 72.4	$ 89.3	$ 89.3
Financial Liabilities:					
Short-term borrowings	Level 2	426.8	426.8	357.2	357.2
2.875% Senior notes	Level 2	399.8	396.9	399.7	390.5
2.45% Senior notes	Level 2	299.9	302.0	299.9	296.0
Fair value adjustment asset (liability) related to hedged fixed rate debt instrument	Level 2	0.2	0.2	1.3	1.3

(a) A prediction sample after post-processing.

```
<table>
  <thead>
      <th></th>
      <th rowspan="2"></th>
      <th colspan="2"></th>
      <th colspan="2"></th>
  </thead>
  <thead>
      <th></th>
      <th></th>
      <th></th>
      <th></th>
      <th></th>
  </thead>
  <tr><td colspan="6"></td></tr>
  <tr>
      <td></td> <td></td> <td></td> <td></td><td></td> <td></td>
  </tr>
  <tr><td colspan="6"></td></tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
</table>
```

(b) The generated HTML sequence after post-processing.

Fig. 10. A sample of prediction result from the FinTabNet testing set.

5.4. Ablation study

In this section, we conduct experiments on the FinTabNet dataset to demonstrate the effectiveness of our applied methods, including using the proposed single-label detection formulation, tuning parameters of RPN, applying the deformable convolution and spatial attention. It is worth mentioning that tuning parameters of RPN includes increasing the number of proposals and adjusting the aspect rations. Since other

Table 4
Experimental results on the SciTSR, FinTabNet and PubTables1M datasets with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells.

Dataset	Model	TEDS-struc.(%)		
		Sim.	Com.	All
SciTSR	Cascade R-CNN	77.31	84.74	79.09
	Deformable-DETR	98.17	94.59	97.30
	Sparse R-CNN	**99.08**	95.92	98.30
	TSRDet(Ours)	98.59	**97.88**	**98.41**
FinTabNet	Cascade R-CNN	82.17	92.50	87.49
	Deformable-DETR	98.08	97.54	97.81
	Sparse R-CNN	98.36	97.91	98.13
	TSRDet(Ours)	**99.08**	**99.02**	**99.05**
PubTables1M	Cascade R-CNN	82.73	85.21	83.78
	Deformable-DETR	97.54	93.14	95.73
	Sparse R-CNN	99.04	95.90	97.72
	TSRDet(Ours)	**99.19**	**97.66**	**98.55**

studies have successfully applied the effectiveness of increasing the number of proposals, we applied it to both the Cascade R-CNN baseline and the proposed TSRDet, as discussed in Sections 3.3 and 5.2. Therefore, we only discuss the impact of adjusting the aspect ratios for tuning parameters of RPN in this section.

The experimental results are shown in Tables 8 and 9, in which Asp_Ratio Tuning, Single_Label, DEFORM, and S_Attn are shorts for applying aspect ratio tuning, single label formulation, deformable convolution, and spatial attention, respectively. Even though Cascade R-CNN baseline can reach 95.06% regarding the mAP, its overall structure-only TEDS only reaches 82.70%. After tuning the aspect ratios for the anchor generation, the structure-only TEDS is increased to 90.23%, even though the mAP is only increased from 95.06% to 95.54%. Applying deformable convolution without other methods can improve the detection performance significantly but lead to a worse structure-only TEDS if we compare Ablation 1 and the Cascade R-CNN baseline. Ablation 3 and Ablation 4 show that transforming the multi-label detection formulation into single-label formulation can significantly improve the performance, and also make deformable convolution improve the model performance. And applying both deformable convolution and spatial attention together can further improve the model performance from the results of Ablation 4, 5 and TSRDet, as shown in Table 8. On the other hand, when it comes to detection metrics, applying deformable convolution always brings performance improvements from the results of Ablation 1 and Ablation 4, which can verify our analysis on the mismatch of detection metrics and cell-level metrics in Section 3.4.

6. Discussions and analysis

Sections 5.2 and 5.4 have demonstrated the effectiveness of our proposed methods. In this section, we further discuss some observations from the experimental results and how these observations verify our analysis in Section 3.

6.1. Multi-label detection

As discussed in Sections 3.1 and 3.2, multi-label detection tasks are difficult for two-stage detection models, but transformer-based detection models with learnable proposals can deal with multi-label detection tasks. Besides, the problem formulation of PubTables1M is a multi-label task, making it difficult for two-stage detection models. The experimental results from Section 5.2 can demonstrate our analysis. For example, as shown in Table 4, on the SciTSR dataset, the performance of Deformable-DETR and Sparse R-CNN are 97.30% and 98.30% regarding the structure-only TEDS, which are very close to the performance of proposed TSRDet (98.41%) and far better than the Cascade R-CNN baseline (79.09%). Notably, as mentioned in Section 5, all the models, except our proposed TSRDet, are using the multi-label detection setting. Therefore, two transformer-based detection models show promising results in the multi-label detection setting. Similarly, the experiments on the FinTabNet and PubTables1M datasets also show similar results. For example, the structure-only TEDS performance of Sparse R-CNN, TSRDet, and Cascade R-CNN baseline are 97.72%, 98.55%, and 83.78% on the PubTables1M dataset, 98.13, 99.05, and 87.49 on the FinTabNet dataset.

6.2. The misalignment of metrics

The experimental results in Sections 5.2 and 5.4 show the misalignment of COCO and TEDS metrics many times. For example, in Table 5, both the Deformable-DETR and our proposed TSRDet can reach 96.28% regarding mAP on the SciTSR dataset, which is better than that of Sparse R-CNN. However, when it comes to structure-only TEDS, as shown in Table 4, both TSRDet and Sparse R-CNN can perform better than Deformable-DETR on the SciTSR dataset. Similar results also appear in the experiments on the FinTabNet dataset. As shown in Table 5, the mAP of Sparse-RCNN and Deformable-DETR are 96.38% and 96.68%, while their structure-only TEDS are 97.81% and 98.13%. More similar results can be found in the results of the ablation study, such as Ablation 1 and Ablation 3, as shown in Tables 8 and 9. To further verify our discussion in Section 3.4, we show the prediction results of Ablation 1 and 3 in Fig. 11. As discussed in Section 3.4, COCO metrics are relied on IoU scores, while TEDS is not. Therefore, as shown in Fig. 11(a), Ablation 1 with deformable convolution can better fit the extra white areas to improve the detection performance, but it cannot improve the TEDS compared with Ablation 3 whose result is shown in Fig. 11(b). It is worth mentioning that the ground truth of the sample in Fig. 11 has been shown in Fig. 6.

6.3. Deformable convolution and spatial attention

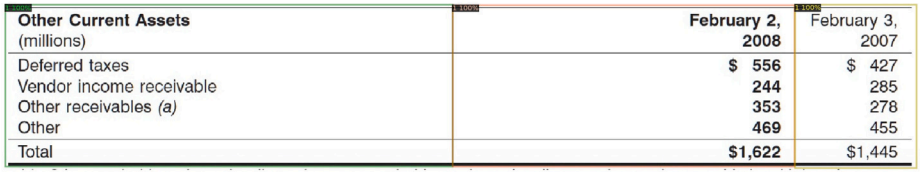
As discussed in Sections 3.4 and 3.5, both generating good local features and building long-range dependencies are essential for a detection-based TSR model, and deformable convolution can improve the local feature generation but has the risk leading to the over-optimization to the detection performance. The ablation study's experimental results can somewhat demonstrate our analysis. Considering the performance of Ablation 1 with deformable convolution in Tables 8 and 9, its TEDS is 84.35%, lower than the Cascade R-CNN baseline (87.49%), but its mAP is improved from 95.23% to 97.22%. These results not only show the misalignment of COCO and TEDS metrics but also demonstrate that merely improving local features can make the model fit empty spaces better, as shown in Fig. 11(a), but does not help alleviate the multi-label detection issue. Therefore, deformable convolution needs to be applied with other methods. On the other hand, our proposed Spatial Attention Module can improve the mAP and structure-only TEDS simultaneously if we compare the performance of Ablation 3 and 4, and also can be used with deformable convolution together to improve the structure-only TEDS further, as shown in Table 8, demonstrating the effectiveness of building long-range dependencies.

6.4. Analysis of the generalization capacities

In this section, we conduct extra experiments in a cross-dataset setting to explore the generalization capacities of the trained models. Specifically, since we mainly use SciTSR, FinTabNet and PubTables1M datasets in the previous sections, in this section, we set up a cross-dataset setting using the training set from one of these three datasets to train the model and the testing sets of the rest of datasets to evaluate the model. As shown in Table 10, all three models show significant performance degradation in the cross-dataset setting, which is caused

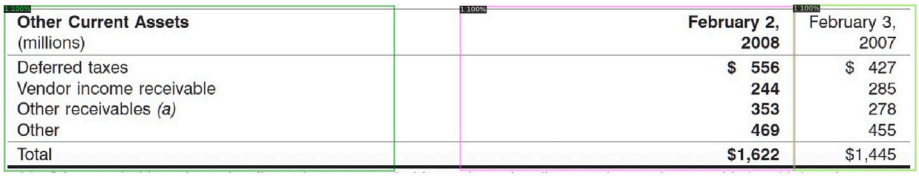
Table 5
Experimental results with Mean Average Precision (mAP).

Dataset	Model	mAP	$AP50$	$AP75$	AP_s	AP_m	AP_l	Table	Column	Row	Spanning cell	Projected row header	Column header
SciTSR	Cascade R-CNN	93.89	95.27	94.80	95.81	93.89	92.96	98.96	98.63	96.33	88.58	83.80	97.01
	Deformable-DETR	96.28	97.39	97.01	96.75	96.55	96.07	98.96	98.63	97.26	93.84	90.86	98.15
	Sparse R-CNN	94.78	96.17	95.48	95.49	95.07	90.08	98.98	98.30	97.93	88.06	86.92	98.49
	TSRDet(Ours)	96.28	96.79	96.57	99.01	96.42	95.65	98.97	99.25	98.57	95.30	87.06	98.50
FinTabNet	Cascade R-CNN	95.23	97.53	96.90	87.32	95.31	93.08	99.00	96.69	96.96	84.43	96.63	97.64
	Deformable-DETR	96.68	98.42	97.98	75.17	95.53	95.58	99.00	97.55	96.95	91.91	96.62	98.04
	Sparse R-CNN	96.38	98.37	97.69	62.11	96.22	95.86	99.01	97.79	97.84	88.39	97.29	97.97
	TSRDet(Ours)	97.50	98.33	98.09	91.60	97.40	97.15	99.01	98.83	97.99	94.62	96.61	97.93
PubTables1M	Cascade R-CNN	93.40	95.38	94.76	85.75	93.32	92.57	99.01	98.76	87.56	82.18	95.81	97.11
	Deformable-DETR	94.82	97.43	96.79	78.33	92.55	94.48	98.99	97.89	95.84	85.04	95.43	95.74
	Sparse R-CNN	96.46	98.14	97.60	84.25	95.73	96.45	99.00	98.42	98.03	87.85	97.91	97.57
	TSRDet(Ours)	97.72	98.26	98.04	94.76	97.43	97.33	99.01	98.99	98.41	94.21	97.88	97.85



Other Current Assets (millions)	**February 2, 2008**	February 3, 2007
Deferred taxes	**$ 556**	$ 427
Vendor income receivable	**244**	285
Other receivables *(a)*	**353**	278
Other	**469**	455
Total	**$1,622**	$1,445

(a) A sample prediction result of the Ablation 1 model. The mAP and structure-only TEDS are 97.22 and 84.35, respectively. We only include the columns' predictions for simplicity.



Other Current Assets (millions)	**February 2, 2008**	February 3, 2007
Deferred taxes	**$ 556**	$ 427
Vendor income receivable	**244**	285
Other receivables *(a)*	**353**	278
Other	**469**	455
Total	**$1,622**	$1,445

(b) A sample prediction result of the Ablation 3 model. The mAP and structure-only TEDS are 95.51 and 96.95, respectively. We only include the columns' predictions for simplicity.

Fig. 11. Comparison of results from Ablation1 and Ablation3 models. Even though Ablation 1 can achieve better detection performance, its performance regarding structure-only TEDS is much lower than that of Ablation 3 model.

Table 6
Experimental results on the FinTabNet dataset with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells.

Model	TEDS-struc.(%)		
	Sim.	Com.	All
EDD (Zhong et al., 2020)	88.40	92.08	90.60
TableFormer (Nassar et al., 2022)	97.50	96.00	96.80
TableMaster (Ye et al., 2021)	98.36	98.28	98.32
VAST (Huang et al., 2023)	–	–	98.63
MTL-TabNet (Ly & Takasu, 2023)	99.07	98.46	98.79
TSRFormer-DQ-DETR (Wang, Lin et al., 2023)	–	–	98.40
TSRDet(Ours)	**99.08**	**99.02**	**99.05**

Table 7
Experimental results on PubTabNet validation set with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells. The proposed model is trained with PubTable1M dataset, while the benchmark models are trained with PubTabNet dataset.

Model	TEDS-struc.(%)		
	Sim.	Com.	All
EDD (Zhong et al., 2020)	91.10	88.70	89.90
RobustTabNet (Ma et al., 2023)	–	–	97.00
TSRNet (Li, Yin et al., 2022)	–	–	95.64
VAST (Huang et al., 2023)	–	–	97.23
TableFormer (Nassar et al., 2022)	98.50	95.00	96.75
MTL-TabNet (Ly & Takasu, 2023)	99.05	96.66	97.88
TSRDet(Ours)	96.99	94.99	96.58

by the domain gap among the three datasets. On the other hand, the models trained with the FinTabNet and PubTables1M datasets show promising performance on the SciTSR testing set, especially for the model trained with the FinTabNet dataset. Moreover, the model trained with the FinTabNet dataset also performs better on the PubTables1M testing set than the model trained with the SciTSR dataset. Considering the scale of these datasets, as mentioned in Table 2, the diversity of samples and the model performance reported in Table 10, we conclude that the model trained with the SciTSR dataset has the worst generalization capacities. The model trained with the FinTabNet dataset can show better generalization capacities than the model trained with the PubTables1M dataset, even though PubTables1M is much larger.

6.5. Analysis of the failed cases

In this section, we visualize some failed cases from the FinTabNet testing set and discuss possible underlying reasons. The model used in this section is trained and tested with the FinTabNet dataset. As discussed in previous sections, even though our proposed method can achieve promising results on the FinTabNet dataset, it can sometimes fail in some edge cases. For example, as shown in Fig. 12, the failed prediction is caused by the missing prediction of the Column Header, whose Column Header is a special case that lies out of the table box, as shown in Fig. 12(b). Fig. 13 shows another example which failed to predict the data cells. For this example, as shown in Fig. 13(b), the appearance of these data cells is texted in paragraphs, which is also different from common cases. Fig. 14 shows another edge case which classifies a regular row as a projected row header. This regular row is divided into two lines, making its appearance very similar to

Table 8
Ablation study results on FinTabNet dataset with structure-only TEDS score. Asp_Ratio Tuning, Single_Label, DEFORM, and S_Attn are shorts for applying aspect ratio tuning, single-label formulation, deformable convolution, and spatial attention.

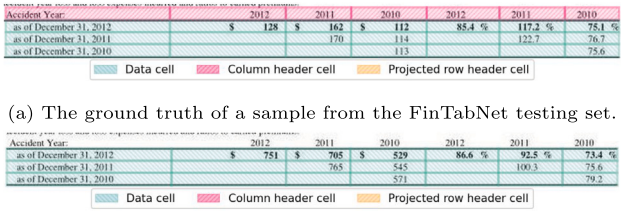
Model	Asp_Ratio tuning	Single_Label	DEFORM	S_Attn	TEDS-struc.(%)		
					Sim.	Com.	All
Cascade R-CNN					82.17	92.50	87.49
Ablation 1			✓		81.45	87.11	84.35
Ablation 2	✓				84.27	95.80	90.23
Ablation 3	✓	✓			95.17	98.63	96.95
Ablation 4	✓	✓	✓		96.44	99.14	97.83
Ablation 5	✓	✓		✓	96.95	98.75	97.88
TSRDet(Ours)	✓	✓	✓	✓	99.08	99.02	99.05

Table 9
Ablation study results regarding mean Average Precision (mAP). The model names are aligned with models in Table 8.

Model	mAP	$AP50$	$AP75$	AP_s	AP_m	AP_l	Table	Column	Row	Spanning Cell	Projected row header	Column header
Cascade R-CNN	95.23	97.53	96.90	87.32	95.31	93.08	99.00	96.69	96.96	84.43	96.63	97.64
Ablation 1	97.22	98.03	97.90	90.11	96.72	96.76	99.00	98.95	96.16	94.98	96.01	98.19
Ablation 2	95.54	97.54	96.91	87.43	95.79	94.04	99.00	97.04	97.64	84.84	96.67	98.02
Ablation 3	95.51	97.56	96.94	88.43	95.52	93.76	99.00	97.31	97.87	84.74	96.82	97.28
Ablation 4	97.83	98.37	98.13	91.91	97.65	97.58	99.00	98.96	98.33	95.78	96.98	97.93
Ablation 5	96.97	97.84	97.58	90.32	96.88	96.21	99.00	98.83	98.03	91.97	96.58	97.37
TSRDet(Ours)	97.50	98.33	98.09	91.60	97.40	97.15	99.01	98.83	97.99	94.62	96.61	97.93

Table 10
Experimental results in the cross-dataset setting with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells.

Training set	Testing set	TEDS-struc.(%)		
		Sim.	Com.	All
SciTSR	SciTSR	98.59	97.88	98.41
	FinTabNet	77.39	78.73	78.08
	PubTables1M	59.51	59.96	59.70
FinTabNet	SciTSR	96.75	93.77	96.03
	FinTabNet	99.08	99.02	99.05
	PubTables1M	76.78	76.78	76.78
PubTables1M	SciTSR	91.02	93.19	91.54
	FinTabNet	81.99	79.40	80.66
	PubTables1M	99.19	97.66	98.55



(a) The ground truth of a sample from the FinTabNet testing set.

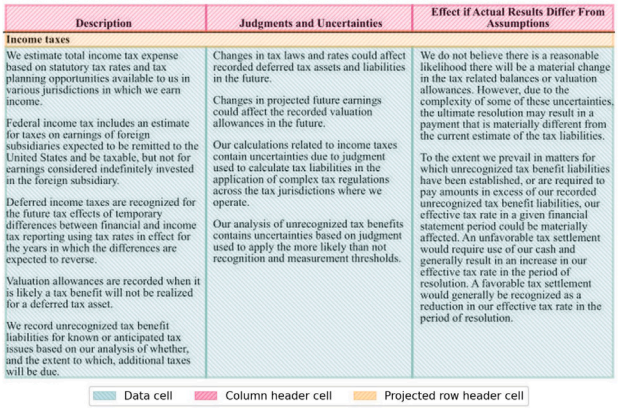
(b) The failed prediction of a sample from the FinTabNet testing set.

Fig. 12. A failed prediction example from the FinTabNet testing set.

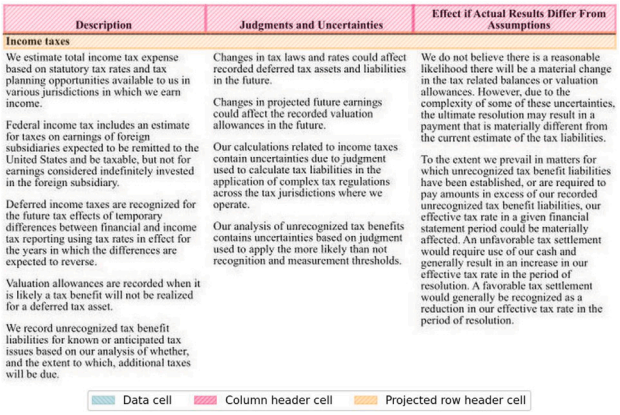
the projected row header. Besides these examples from the FinTabNet dataset, we include more cases in the Appendix. From the visualization of failed cases, we conclude that even with promising structure-only TEDS, we still need to pay more attention to the edge cases when applying our proposed models.

6.6. Other observations

Besides the observations discussed in previous sections, the experimental results also show other phenomena that can be helpful in our model design. One observation is that Cascade R-CNN has better detection performance on small objects than Sparse R-CNN. For example, on the FinTabNet dataset, APs of Sparse R-CNN is only 62.11%, while the Cascade R-CNN baseline and our proposed TSRDet reach 87.32% and 91.60%. This phenomenon might be caused by their methods of generating regional proposals. As discussed in Section 3.1, Cascade R-CNN uses RPN to generate regional proposals, which regress and classify anchor boxes, and the anchor boxes are generated by sliding the pre-defined boxes with different aspect ratios and sizes on the feature map of multiple scales. Therefore, Cascade R-CNN uses a dense



(a) The ground truth of a sample from the FinTabNet testing set.



(b) The failed prediction of a sample from the FinTabNet testing set.

Fig. 13. A failed prediction example from the FinTabNet testing set.

	Year Ended December 31,				
	2008	2007	2006	2005	2004
	(In thousands, except per share amounts)				
Income statement data					
Total revenue	$2,025,267	$1,962,159	$1,650,549	$1,232,480	$956,636
Cost of operations (exclusive of amortization and depreciation disclosed separately below)[1]	1,342,039	1,304,631	1,095,929	833,283	641,067
General and administrative[1]	82,804	80,898	91,815	88,797	75,819
Depreciation and other amortization	68,527	59,688	48,499	40,545	37,369
Amortization of purchased intangibles	67,291	67,323	40,926	23,004	13,415
Loss on sale of assets	1,052	16,045	—	—	—
Merger costs	3,053	12,349	—	—	—
Total operating expenses	1,564,766	1,540,934	1,277,169	985,629	767,670
Operating income	460,501	421,225	373,380	246,851	188,966
Fair value loss on interest rate derivative	—	—	—	—	808
Interest expense, net	63,648	69,381	40,722	13,905	6,651
Income from continuing operations before income taxes	396,853	351,844	332,658	232,946	181,507
Provision for income taxes	153,454	137,403	126,261	86,318	67,560
Income from continuing operations	243,399	214,441	206,397	146,628	113,947
Loss from discontinued operations, net of taxes	(26,006)	(50,380)	(16,792)	(7,883)	(11,576)
Net income	$ 217,393	$ 164,061	$ 189,605	$ 138,745	$102,371
Income from continuing operations per share—basic	$ 3.40	$ 2.74	$ 2.59	$ 1.78	$ 1.41
Income from continuing operations per share—diluted	$ 3.31	$ 2.65	$ 2.53	$ 1.73	$ 1.36
Net income per share—basic	$ 3.04	$ 2.09	$ 2.38	$ 1.69	$ 1.27
Net income per share—diluted	$ 2.95	$ 2.03	$ 2.32	$ 1.64	$ 1.22
Weighted average shares used in computing per share amounts—basic	71,502	78,403	79,735	82,208	80,614
Weighted average shares used in computing per share amounts—diluted	73,640	80,811	81,686	84,637	84,040

Data cell Column header cell Projected row header cell

(a) The ground truth of a sample from the FinTabNet testing set.

	Year Ended December 31,				
	2008	2007	2006	2005	2004
	(In thousands, except per share amounts)				
Income statement data					
Total revenue	$2,025,267	$1,962,159	$1,650,549	$1,232,480	$956,636
Cost of operations (exclusive of amortization and depreciation disclosed separately below)[1]	1,342,039	1,304,631	1,095,929	833,283	641,067
General and administrative[1]	82,804	80,898	91,815	88,797	75,819
Depreciation and other amortization	68,527	59,688	48,499	40,545	37,369
Amortization of purchased intangibles	67,291	67,323	40,926	23,004	13,415
Loss on sale of assets	1,052	16,045	—	—	—
Merger costs	3,053	12,349	—	—	—
Total operating expenses	1,564,766	1,540,934	1,277,169	985,629	767,670
Operating income	460,501	421,225	373,380	246,851	188,966
Fair value loss on interest rate derivative	—	—	—	—	808
Interest expense, net	63,648	69,381	40,722	13,905	6,651
Income from continuing operations before income taxes	396,853	351,844	332,658	232,946	181,507
Provision for income taxes	153,454	137,403	126,261	86,318	67,560
Income from continuing operations	243,399	214,441	206,397	146,628	113,947
Loss from discontinued operations, net of taxes	(26,006)	(50,380)	(16,792)	(7,883)	(11,576)
Net income	$ 217,393	$ 164,061	$ 189,605	$ 138,745	$102,371
Income from continuing operations per share—basic	$ 3.40	$ 2.74	$ 2.59	$ 1.78	$ 1.41
Income from continuing operations per share—diluted	$ 3.31	$ 2.65	$ 2.53	$ 1.73	$ 1.36
Net income per share—basic	$ 3.04	$ 2.09	$ 2.38	$ 1.69	$ 1.27
Net income per share—diluted	$ 2.95	$ 2.03	$ 2.32	$ 1.64	$ 1.22
Weighted average shares used in computing per share amounts—basic	71,502	78,403	79,735	82,208	80,614
Weighted average shares used in computing per share amounts—diluted	73,640	80,811	81,686	84,637	84,040

Data cell Column header cell Projected row header cell

(b) The failed prediction of a sample from the FinTabNet testing set.

Fig. 14. A failed prediction example from the FinTabNet testing set.

Method	Con-local	ETE-hung	R-FCN	f-localized	our method
AP	45.4	78.4	84.8	85.3	**88.1**

Data cell Column header cell Projected row header cell

(a) The ground truth of a sample from the SciTSR testing set.

Method	Con-local	ETE-hung	R-FCN	f-localized	our method
AP	45.4	78.4	84.8	85.3	**88.1**

Data cell Column header cell Projected row header cell

(b) The failed prediction of a sample from the SciTSR testing set.

Fig. A.15. A failed prediction example from the SciTSR testing set.

Value	x (loudness)	y (rudeness)
0.0	no sound	undetectable
0.1	whisper	indirect request: hint
0.2	urgent whisper	preference
0.3	subdued speech	query
0.4	speaking voice	direct request: suggestion
0.5	authoritative tone	obligation
0.6	loud voice	command
0.7	yell	generic foul words
0.8	shout	targeted offense: eg. ethnic slur
0.9	scream	
1.0	shriek	threat of physical violence

Data cell Column header cell Projected row header cell

(a) The ground truth of a sample from the SciTSR testing set.

Value	x (loudness)	y (rudeness)
0.0	no sound	undetectable
0.1	whisper	indirect request: hint
0.2	urgent whisper	preference
0.3	subdued speech	query
0.4	speaking voice	direct request: suggestion
0.5	authoritative tone	obligation
0.6	loud voice	command
0.7	yell	generic foul words
0.8	shout	targeted offense: eg. ethnic slur
0.9	scream	
1.0	shriek	threat of physical violence

Data cell Column header cell Projected row header cell

(b) The failed prediction of a sample from the SciTSR testing set.

Fig. A.16. A failed prediction example from the SciTSR testing set.

proposal generation method (Sun, Zhang et al., 2021) with more region proposals, meaning that Cascade R-CNN can use the parameters of RPN to generate more high-quality small region proposals. By contrast, Sparse R-CNN uses sparse learnable regional proposals to replace dense proposals generated by the RPN, which can avoid parameter tuning of RPN but limit its performance on small objects. Another interesting observation is that the baseline Cascade R-CNN can work better on complex tables than simple tables, which is very different from other benchmark models. This phenomenon is caused by the fact that the spanning cells in complex tables are usually in the Column Row Headers, which can alleviate the multi-label detection issue. For example, Figs. 4(c) and 4(d) show two samples from PubTables1M dataset, in which the former is a complex table and the latter is a simple table. Because of the existence of Spanning Cells in Fig. 4(c), the Column Header does not share its bounding box with any rows, which avoids multi-label detection. By contrast, the sample in Fig. 4(d) does not contain any Spanning Cell, making its Column Header share its bounding box with a Row, which is the challenging multi-label detection to Cascade R-CNN. As comparisons, Deformable-DETR and Sparse R-CNN can deal with multi-label detection, and their performance on simple tables is better than complex tables regarding the structure-only TEDS, as shown in Table 4.

DataSet	# Training quads	# Test quads
Labeled data		
WSJ	20,801	3,097
NYTC	0	293
WKP	0	381
Unlabeled data		
WKP	100,000	4,473,072

Data cell Column header cell Projected row header cell

(a) The ground truth of a sample from the SciTSR testing set.

DataSet	# Training quads	# Test quads
	Labeled data	
WSJ	20,801	3,097
NYTC	0	293
WKP	0	381
	Unlabeled data	
WKP	100,000	4,473,072

Data cell Column header cell Projected row header cell

(b) The failed prediction of a sample from the SciTSR testing set.

Fig. A.17. A failed prediction example from the SciTSR testing set.

6.7. Summary of insights

In this section, we summarize the key insights and critical design aspects for a detection-based TSR model. It is worth mentioning that the rationale behind these insights, along with the experiments validating them, has been thoroughly discussed in Sections 3, 5, and 6. First, a detection-based TSR solution needs to define the target table components properly to provide full table structural information. Some studies (Fernandes et al., 2023; Hashmi et al., 2021; Siddiqui et al., 2019; Xiao, Akkaya et al., 2022) over-simply the target detection components without Headers and Projected Row Headers, making them cannot fully recover the complex table structures. Furthermore, the problem formulation should align with the capacities of the employed detection model. For instance, PubTable1M (Smock et al., 2022) defines six types of target table components to fully reconstruct the complex table structures. However, it presents a multi-label detection definition, posing challenges for two-stage detection models. Therefore, in this study, we further develop the formulation of PubTable1M by introducing a pseudo-class to transform multi-label detection to regular single-label detection, as discussed in Section 4.1. Thirdly, existing studies usually employ COCO metrics to evaluate the detection-based TSR models. However, COCO metrics are insufficient for evaluating the TSR models, because ground truth boxes are often exceed the minimum bounding boxes required for capturing table structures, as discussed in Section 3.4, and models may optimize towards accommodating easier components with additional spaces, rather than effectively identifying challenging components. Hence, this study incorporates structure-only TEDS for model evaluation and introduces a Spatial Attention Module. The module is designed to establish long-range dependencies, enhancing the model's ability to explore and address challenging components effectively. Fourthly, two-stage and transformer-based detection models have different capacities in the context of the TSR task. In this study, we leverage Cascade R-CNN and Sparse R-CNN as illustrative examples to highlight their differing capacities. More specifically, Sparse R-CNN excels in handling multi-label detection tasks without the need for tuning the region proposals because of its utilization of sparse learnable proposals, as discussed in Sections 3.1 and 3.3. By contrast, Cascade R-CNN cannot deal with multi-label detection tasks and needs to carefully tune the parameters of proposal generation because of the aspect ratios of

Table 3: Characteristics of the RTT patient and control brain samples used in this study.

Sample	Source	Uid	Sex	Age	Mutation			
					nt change	aa change	domain affected	type of mutation
RTT1	HBTRC	4315	F	11	c.763C>T	R255X	TRD-NLS	Nonsense
RTT3	HBTRC	4422	F	12	c.808C>T	R270X	TRD-NLS	Nonsense
RTT4	TICHR	NB	F	18	c.473C>T	T158M	MBD	Missense
RTT5	TICHR	KB	F	11	c.316C>T	R106W	MBD	Missense
RTT6	TICHR	BC	F	21	c.808C>T	R270X	TRD-NLS	Nonsense
RTT9	BCM	93-244	F	4	c.750insC	P251fs	TRD	Frameshift/Truncation
CON4	NSWTRC	88210	F	43				
CON5	NSWTRC	88295	F	42				
CON6	NSWTRC	88365	F	31				
CON7	NSWTRC	88304	F	43				
CON8	NSWTRC	9092	F	46				
CON10	NSWTRC	12862	F	52				

Abbreviations: HBTRC = The Harvard Brain Tissue Resource Center, Boston, USA; NSWTRC = The New South Wales Tissue Resource Centre, Sydr Australia; BCM = Baylor College of Medicine, Houston, USA; TICHR = Telethon Institute for Child Health Research, Perth, Australia; Uid = uniq

Data cell Column header cell Projected row header cell

(a) The ground truth of a sample from the PubTables1M testing set.

Table 3: Characteristics of the RTT patient and control brain samples used in this study.

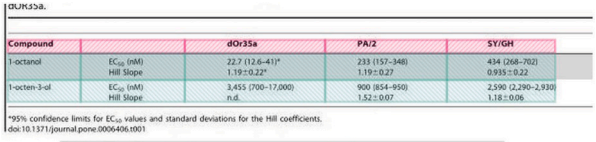
Sample	Source	Uid	Sex	Age	Mutation			
					nt change	aa change	domain affected	type of mutation
RTT1	HBTRC	4315	F	11	c.763C>T	R255X	TRD-NLS	Nonsense
RTT3	HBTRC	4422	F	12	c.808C>T	R270X	TRD-NLS	Nonsense
RTT4	TICHR	NB	F	18	c.473C>T	T158M	MBD	Missense
RTT5	TICHR	KB	F	11	c.316C>T	R106W	MBD	Missense
RTT6	TICHR	BC	F	21	c.808C>T	R270X	TRD-NLS	Nonsense
RTT9	BCM	93-244	F	4	c.750insC	P251fs	TRD	Frameshift/Truncation
CON4	NSWTRC	88210	F	43				
CON5	NSWTRC	88295	F	42				
CON6	NSWTRC	88365	F	31				
CON7	NSWTRC	88304	F	43				
CON8	NSWTRC	9092	F	46				
CON10	NSWTRC	12862	F	52				

Abbreviations: HBTRC = The Harvard Brain Tissue Resource Center, Boston, USA; NSWTRC = The New South Wales Tissue Resource Centre, Sydr Australia; BCM = Baylor College of Medicine, Houston, USA; TICHR = Telethon Institute for Child Health Research, Perth, Australia; Uid = uniq

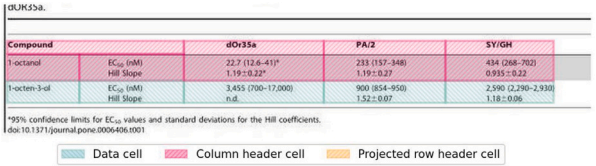
Data cell Column header cell Projected row header cell

(b) The failed prediction of a sample from the PubTables1M testing set.

Fig. A.18. A failed prediction example from the PubTables1M testing set.

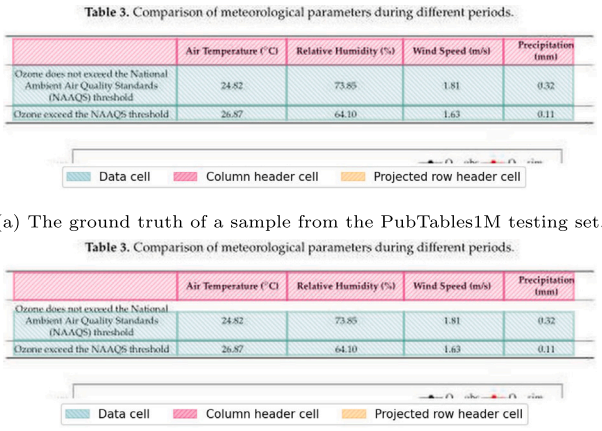


(a) The ground truth of a sample from the PubTables1M testing set.



(b) The failed prediction of a sample from the PubTables1M testing set.

Fig. A.19. A failed prediction example from the PubTables1M testing set.



(a) The ground truth of a sample from the PubTables1M testing set.

(b) The failed prediction of a sample from the PubTables1M testing set.

Fig. A.20. A failed prediction example from the PubTables1M testing set.

defined table components, as discussed in Sections 3.1, 3.2 and 3.3. Additionally, Cascade R-CNN demonstrates superior performance on small objects compared to Sparse R-CNN, partially attributed to its dense and tunable proposal generation, as illustrated in Section 5.2. At last, while enhancing local feature extraction, such as employing deformable convolution, often leads to improved detection performance, it may not necessarily translate to enhanced TSR performance. It is necessary to build long-range dependencies, as discussed in Section 3.5. To sum up, it is imperative to ensure proper alignment between the problem formulation, capacities of detection models, evaluation metrics, and feature extraction in the context of a detection-based TSR solution. Our proposed Cascade R-CNN can be a demonstrative application of these insights in designing an effective detection-based TSR model.

7. Conclusion and future work

In this study, we first revisit existing detection-based TSR solutions and analyze the critical design aspects for a successful detection-based TSR model, including the problem formulation, the characteristics of detection models, and the characteristics of TSR tasks. Our analysis can be a guideline for improving the performance of a detection-based model. To demonstrate our analysis and findings, we propose TSRDet by applying simple methods to tailor the Cascade R-CNN, which can outperform different types of state-of-the-art models, including image-to-sequence and graph-based models. Even though we only applied very simple methods to a two-stage detection model, there should be other methods to further improve the model based on our analysis. For example, vision transformers can be considered to build long-range dependencies. Transformer-based detection models, such as Sparse R-CNN, can also be considered as base models with the benefits of dealing with multi-label detection tasks and learnable proposals. Besides, since the proposed method is detection-based and focuses on well-formatted, visually rich documents, one major limitation is that it may fail to deal with irregular tables, such as rotated and distorted tables. Integrating instance segmentation with detection models can be another direction to deal with irregular tables because instance segmentation can handle irregular shapes and be guided by bounding boxes.

CRediT authorship contribution statement

Bin Xiao: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization . **Murat Simsek:** Writing – review & editing, Validation, Investigation, Formal analysis, Conceptualization. **Burak Kantarci:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition. **Ala Abu Alkheir:** Writing – review & editing, Resources, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by Mathematics of Information Technology and Complex Systems (Mitacs) Accelerate Program and Lytica Inc.

Appendix. More failed prediction cases

In this section, we include more examples of failed predictions from the SciTSR and PubTables1M datasets. It is worth mentioning the low resolution of some figures is caused by the original images from the datasets. Fig. A.16 shows an example of failing to predict the boundary of two rows. Specifically, as shown in Fig. A.16(a), the texts “targeted offense: eg”. and “ethnic slur” should be in the same cell, but predicted as two separate cells in Fig. A.16(b). Figs. A.15 and A.17 are another two examples from the SciTSR dataset, failing to classify the projected row headers and table headers. The model trained with the PubTables1M dataset also sometimes fails to predict the table headers correctly, as shown in Figs. A.18 and A.19. Besides, sometimes its predicted bounding box can be smaller than the ideal box, leading to the loss of information as shown in Fig. A.20. Therefore, we need to consider the edge cases when applying our proposed models, even though they can show promising performance with structure-only TEDS.

Data availability

Publicly available datasets.

References

Adiga, D., Bhat, S. A., Shah, M. B., & Vyeth, V. (2019). Table structure recognition based on cell relationship, a bottom-up approach. In *Proceedings of the international conference on recent advances in natural language processing* RANLP 2019, (pp. 1–8). INCOMA Ltd.

Bacea, D.-S., & Oniga, F. (2023). Single stage architecture for improved accuracy real-time object detection on mobile devices. *Image and Vision Computing*, *130*, Article 104613.

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *2018 IEEE/CVF conference on computer vision and pattern recognition* (pp. 6154–6162).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision* (pp. 213–229). Springer.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., et al. (2019). MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155.

Chen, F., Zhang, H., Hu, K., Huang, Y.-K., Zhu, C., & Savvides, M. (2023). Enhanced training of query-based object detection via selective query recollection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 23756–23765). IEEE.

Chi, Z., Huang, H., Xu, H.-D., Yu, H., Yin, W., & Mao, X.-L. (2019). Complicated table structure recognition. arXiv preprint arXiv:1908.04729.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1251–1258). IEEE.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., et al. (2017). Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 764–773). IEEE.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

Ding, X., Zhang, X., Han, J., & Ding, G. (2022). Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11963–11975). IEEE.

Fernandes, J., Xiao, B., Simsek, M., Kantarci, B., Khan, S., & Alkheir, A. A. (2023). Tablestrrec: framework for table structure recognition in data sheet images. *International Journal on Document Analysis and Recognition (IJDAR)*, 1–19.

Guo, M.-H., Lu, C.-Z., Hou, Q., Liu, Z., Cheng, M.-M., & Hu, S.-M. (2022). Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems*, *35*, 1140–1156.

Hashmi, K. A., Stricker, D., Liwicki, M., Afzal, M. N., & Afzal, M. Z. (2021). Guided table structure recognition through anchor optimization. *IEEE Access*, *9*, 113521–113534.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969). IEEE.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778). IEEE.

Hong, Q., Liu, F., Li, D., Liu, J., Tian, L., & Shan, Y. (2022). Dynamic sparse r-cnn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4723–4732). IEEE.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Hu, P., Wang, W., Li, Q., & Wang, T. (2021). Touching text line segmentation combined local baseline and connected component for uchen tibetan historical documents. *Information Processing & Management*, *58*(6), Article 102689.

Huang, Y., Lu, N., Chen, D., Li, Y., Xie, Z., Zhu, S., et al. (2023). Improving table structure recognition with visual-alignment sequential coordinate modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11134–11143). IEEE.

Ioffe, S. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

JaidedA, I. (2022). Easyocr. https://github.com/JaidedAI/EasyOCR.git.

Krogh, A., & Hertz, J. (1991). A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems*, *4*.

Kuang, Z., Sun, H., Li, Z., Yue, X., Lin, T. H., Chen, J., et al. (2021). Mmocr: A comprehensive toolbox for text detection, recognition and understanding. arXiv preprint arXiv:2108.06543.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., et al. (2022). Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

Li, X.-H., Yin, F., Dai, H.-S., & Liu, C.-L. (2022). Table structure recognition and form parsing by end-to-end object detection and relation parsing. *Pattern Recognition*, *132*, Article 108946.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125). IEEE.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September (2014) 6-12, proceedings, Part V 13* (pp. 740–755). Springer.

Liu, H., Li, X., Liu, B., Jiang, D., Liu, Y., & Ren, B. (2022). Neural collaborative graph machines for table structure recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4533–4542). IEEE.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11976–11986). IEEE.

Lu, N., Yu, W., Qi, X., Chen, Y., Gong, P., Xiao, R., et al. (2021). Master: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*, *117*, Article 107980.

Ly, N. T., & Takasu, A. (2023). 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 - volume 5: VISAPP* (pp. 626–634). SciTePress.

Ma, C., Lin, W., Sun, L., & Huo, Q. (2023). Robust table detection and structure recognition from heterogeneous document images. *Pattern Recognition*, *133*, Article 109006.

Mendes, J., & Saraiva, J. (2017). Tabula: A language to model spreadsheet tables. arXiv preprint arXiv:1707.02833.

Mondal, A., Agarwal, M., & Jawahar, C. (2023). Dataset agnostic document object detection. *Pattern Recognition*, *142*, Article 109698.

Nassar, A., Livathinos, N., Lysak, M., & Staar, P. (2022). Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4614–4623). IEEE.

Nguyen, N. Q., Le, A. D., Lu, A. K., Mai, X. T., & Tran, T. A. (2023). Formerge: Recover spanning cells in complex table structure using transformer network. In *International conference on document analysis and recognition* (pp. 522–534). Springer.

Pascanu, R. (2013). On the difficulty of training recurrent neural networks. arXiv preprint arXiv:1211.5063.

Prasad, D., Gadpal, A., Kapadni, K., Visave, M., & Sultanpure, K. (2020). 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* (pp. 572–573). IEEE.

Qiao, L., Li, Z., Cheng, Z., Zhang, P., Pu, S., Niu, Y., et al. (2021). Lgpma: Complicated table structure recognition with local and global pyramid mask alignment. In *International conference on document analysis and recognition* (pp. 99–114). Springer.

Rastan, R., Paik, H.-Y., & Shepherd, J. (2019). Texus: A unified framework for extracting and understanding tables in pdf documents. *Information Processing & Management*, *56*(3), 895–918.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, *28*.

Ren, T., Liu, S., Li, F., Zhang, H., Zeng, A., Yang, J., et al. (2023). Detrex: Benchmarking detection transformers. arXiv preprint arXiv:2306.07265.

Schreiber, S., Agne, S., Wolf, I., Dengel, A. R., & Ahmed, S. (2017). Deepdesrt: Deep learning for detection and structure recognition of tables in document images. *Vol. 01*, In *2017 14th IAPR international conference on document analysis and recognition* (pp. 1162–1167).

Shen, H., Gao, X., Wei, J., Qiao, L., Zhou, Y., Li, Q., et al. (2023). Divide rows and conquer cells: Towards structure recognition for large tables. In *Proceedings of the thirty-second international joint conference on artificial intelligence* IJCAI-23, (pp. 1369–1377). International Joint Conferences on Artificial Intelligence Organization.

Siddiqui, S. A., Fateh, I. A., Rizvi, S. T. R., Dengel, A., & Ahmed, S. (2019). Deeptabstr: Deep learning based table structure recognition. In *2019 international conference on document analysis and recognition* ICDAR, (pp. 1403–1409). IEEE.

Siddiqui, S. A., Malik, M. I., Agne, S., Dengel, A., & Ahmed, S. (2018). Decnt: Deep deformable cnn for table detection. *IEEE Access*, *6*, 74151–74161.

Singer-Vine, J. (2022). Pdfplumber. https://github.com/jsvine/pdfplumber.git.

Smock, B., & Pesala, R. (2021). Table transformer. https://github.com/microsoft/table-transformer.

Smock, B., Pesala, R., & Abraham, R. (2022). Pubtables-1 m: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4634–4642). IEEE.

Smock, B., Pesala, R., & Abraham, R. (2023). Aligning benchmark datasets for table structure recognition. In G. A. Fink, R. Jain, K. Kise, & R. Zanibbi (Eds.), *Document analysis and recognition - ICDAR 2023* (pp. 371–386). Cham: Springer Nature Switzerland.

Sun, P., Jiang, Y., Xie, E., Shao, W., Yuan, Z., Wang, C., et al. (2021). What makes for end-to-end object detection? In *International conference on machine learning* (pp. 9934–9944). PMLR.

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., et al. (2021). Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14454–14463). IEEE.

Tensmeyer, C., Morariu, V. I., Price, B., Cohen, S., & Martinez, T. (2019). Deep splitting and merging for table structure decomposition. In *2019 international conference on document analysis and recognition* ICDAR, (pp. 114–121). IEEE.

Tian, Z., Shen, C., Chen, H., & He, T. (2019). Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9627–9636). IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. CVPR, IEEE.

Wang, J., Lin, W., Ma, C., Li, M., Sun, Z., Sun, L., et al. (2023). Robust table structure recognition with dynamic queries enhanced detection transformer. *Pattern Recognition*, *144*, Article 109817.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

Wu, X., Ma, T., Du, X., Hu, Z., Yang, J., & He, L. (2023). Drfn: A unified framework for complex document layout analysis. *Information Processing & Management*, *60*(3), Article 103339.

Wu, X., Xiao, L., Du, X., Zheng, Y., Li, X., Ma, T., et al. (2023). Cross-domain document layout analysis using document style guide. *Expert Systems with Applications*, Article 123039.

Xiao, B., Akkaya, Y., Simsek, M., Kantarci, B., & Alkheir, A. A. (2022). Efficient information sharing in ict supply chain social network via table structure recognition. In *GLOBECOM 2022-2022 IEEE global communications conference* (pp. 4661–4666). IEEE.

Xiao, B., Akkaya, Y., Simsek, M., Kantarci, B., & Alkheir, A. A. (2023). Multi-modal ocr system for the ict global supply chain. In *ICC 2023-IEEE international conference on communications* (pp. 3096–3101). IEEE.

Xiao, B., Simsek, M., Kantarci, B., & Alkheir, A. A. (2022). Handling big tabular data of ict supply chains: a multi-task, machine-interpretable approach. In *GLOBECOM 2022-2022 IEEE global communications conference* (pp. 504–509). IEEE.

Xiao, B., Simsek, M., Kantarci, B., & Alkheir, A. A. (2023b). Revisiting table detection datasets for visually rich documents. arXiv preprint arXiv:2305.04833.

Xiao, B., Simsek, M., Kantarci, B., & Alkheir, A. A. (2023c). Table detection for visually rich document images. *Knowledge-Based Systems*, *282*, Article 111080.

Xue, W., Yu, B., Wang, W., Tao, D., & Li, Q. (2021). Tgrnet: A table graph reconstruction network for table structure recognition. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 1295–1304). IEEE.

Ye, J., Qi, X., He, Y., Chen, Y., Gu, D., Gao, P., et al. (2021). Pingan-vcgroup's solution for icdar 2021 competition on scientific literature parsing task b: Table recognition to html. arXiv preprint arXiv:2105.01848.

Yu, F., Huang, J., Luo, Z., Zhang, L., & Lu, W. (2023). An effective method for figures and tables detection in academic literature. *Information Processing & Management*, *60*(3), Article 103286.

Zhang, S., Wang, X., Wang, J., Pang, J., Lyu, C., Zhang, W., et al. (2023). Dense distinct query for end-to-end object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7329–7338). IEEE.

Zhang, Z., Zhang, J., Du, J., & Wang, F. (2022). Split, embed and merge: An accurate table structure recognizer. *Pattern Recognition*, *126*, Article 108565.

Zheng, X., Burdick, D., Popa, L., Zhong, X., & Wang, N. X. R. (2021). 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* (pp. 697–706). IEEE.

Zhong, X., ShafieiBavani, E., & Jimeno Yepes, A. (2020). Image-based table recognition: data, model, and evaluation. In *European conference on computer vision* (pp. 564–580). Springer.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). Deformable detr: Deformable transformers for end-to-end object detection. In *International conference on learning representations*.