This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Treading Towards Privacy-Preserving Table Structure Recognition

Sachin Raja
IIIT Hyderabad
sachin.raja@research.iiit.ac.in

Ajoy Mondal
IIIT Hyderabad
ajoy.mondal@iiit.ac.in

C V Jawahar
IIIT Hyderabad
jawahar@iiit.ac.in

Abstract

We present TabGuard, a privacy-preserving framework for an end-to-end secure Table Structure Recognition. TabGuard masks all the contents of the table locally and utilizes the masked table image for structure recognition. Our method is simple yet effective for detecting table cells while preserving the inherent table alignment characteristics to reconstruct tables. Our approach benefits from inductive bias, expressed through an approximated table grid which helps alleviate challenges in the detection of cells that are small or have extreme aspect ratios. Experimental results demonstrate that our solution not only establishes a new state-of-the-art on several benchmark datasets but also effectively addresses long-standing challenges associated with dense tables having complex layouts. We make our code publically available at https://github.com/sachinraja13/TabGuard.

1. Introduction

Table Structure Recognition (TSR) is a pivotal component in document analysis and data extraction. It is formally defined as transforming an image of a table into a machine readable format, where its layout and locality information is encoded into a predefined format [7, 26, 42, 57, 64, 72]. Tables from data-sensitive sectors such as finance and healthcare often contain sensitive and confidential information highlighting privacy concerns. Since most deep-learning solutions require GPU computations, hosting a client-server system becomes inevitable. While an on-premise server can aid data security, it still poses a risk of unauthorised access within the organization. Consequently, we present TabGuard (shown in Figure 1) which treads the first step towards privacy-preserving Table Structure Recognition (TSR). Client and server communication requires transfer of lightweight fixed size masked table images and text contours as JSON to the server and resulting table structure as JSON back to the client in single or batch mode. Asynchronous communication and easy horizontal and vertical scalability ensure minimal latency. We believe that ensuring privacy can aid data acquisition from content-sensitive domains such as health records, invoices, and legal and insurance documents, which have different layout characteristics compared to academic datasets [7, 56, 72].

We follow a top-down and bottom-up strategy for TSR which requires accurate detection of table cells. Convolution-based methods like Faster R-CNN [17, 51] rely on hyperparameters for anchor size, aspect ratio, and stride to generate base anchors, which are filtered using non-maximal suppression (NMS) based on Region Proposal Network (RPN) predictions. The detection performance heavily depends on the overlap of anchor boxes with ground truth objects; poor overlap can lead to false negatives. This is worsened for objects with extreme sizes and aspect ratios, as seen for a few cases in the COCO dataset [13, 50]. Specifically, detecting table cells is challenging due to their variable sizes and layouts within the same table, and high-density tables often result in numerous false negatives due to NMS filtering of anchor boxes. Further, recent advances in object detection including DETR and its enhancements [4, 61, 73] have been shown to face challenges in detecting small and densely packed objects [61, 70, 73]. This is because the cross-attention mapping between decoder queries and the encoder output is hard to learn when the variations in the number of expected objects and their sizes vary significantly across images in the dataset.

Consequently, we utilize convolution-based Faster R-CNN augmented by table-specific alignment and continuity losses [48, 49]. However, instead of relying on anchor generation using sizes and aspect ratio hyperparameters, we first approximate the table grid using text localization. This coarse table grid allows for dynamic anchor generation spanning every text region in the table image ensuring accurate and fast convergence for detection of table cells. Once table cells are accurately detected and are well-aligned, identifying their structure by assigning row and column-spanning indices to each cell becomes straightforward. We achieve this using a simple post-processing step. For text localization, instead of using OCR tools, (which tend to produce false negatives fail in case of tables that have row/column separators), we rely on the fundamental

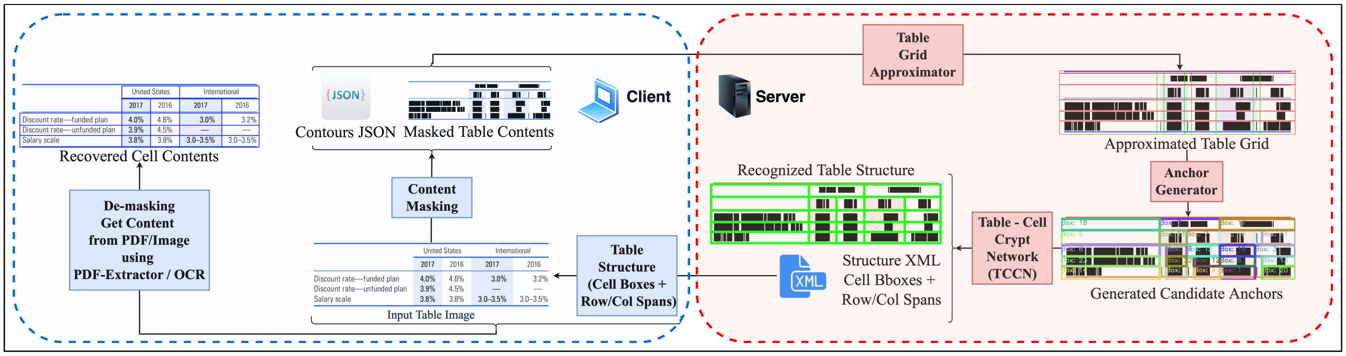


Figure 1. Overview of *TabGuard*. Content of the table is only seen by the client. TSR API server sees images with content masked.

contrast variations that exist between text and background regions of a document image. Since our text extraction does not depend on any deep-learning framework, it allows for efficient content masking on any commodity/handheld device as the precursor to structure recognition which can proceed in a completely secure manner. Overall, our contributions can be summarized as follows:

- To our knowledge, we introduce the first privacy-preserving, end-to-end framework for table structure recognition, *TabGuard*. Our comparison study demonstrates that *TabGuard* achieves state-of-the-art performance, effectively addressing challenges like high cell density and extreme aspect ratios, while ensuring data privacy and cross-domain robustness.
- We present a fast, resource-efficient, and OCR-free language-agnostic algorithm to mask out all the content present in the table image.
- To detect table-cells, we propose Table Cell Crypt Network (TCCN), a simplified Faster R-CNN [51] without the anchor generator and region proposal networks. Instead, we generate dynamic table-specific anchors using our *Table Grid Approximator (TGA)*.

2. Related Work

Literature in Table structure recognition (TSR) can primarily be classified into (i) Top-Down and Bottom-Up methods and (ii) Image-to-Sequence methods. Deep learning models involve three semantic modules: a *feature extractor* like ResNet [18], an *encoder* such as a graph neural network or a Region Proposal Network (RPN) [51] with alignment networks like Multi-Scale RoI Align [51], and a *decoder* which could be a graph neural network [25, 29, 46, 52, 65], a cell classifier and regressor [41, 44, 48, 49, 53], or a transformer [58] or LSTM [19]-based decoder [22, 26, 40].

Top-Down and Bottom-Up Methods: These methods [34, 36, 49, 57] start by breaking the table into a grid structure (top-down) and then establish inter-cell relationships (bottom-up). Techniques like DeepDeSRT [53] and TableNet [42] use FCNs [33] for segmenting rows and columns. SPLERGE [57] and Zhang et al. [69] focus on splitting grids and merging spanning cells. Khan et al. [23] and RobusTabNet [37] predict separator lines using RNNs and spatial CNNs. TSRFormer [27] utilizes line regression for table separation. Other methods [7, 20, 45, 52] leverage graph neural networks for cell or word relationships, while approaches like [30, 48, 49] combine Mask R-CNN [17] and DGCNN [45] for cell detection and adjacency prediction. TabStructNet [49] and NCGM [29] use multimodal features for complex scenarios. Shen et al. [55] and LORE [63] propose row and column projections and cascade regression frameworks respectively. GrabTab [28] adopts a progressive deliberation principle. However, these methods struggle with densely packed tables with numerous rows and empty cells due to size and aspect ratio issues.

Image-to-Sequence Methods: These methods [22, 40, 72] encode visual features into a fixed-size representation and use attention mechanisms to decode them into HTML or LaTeX sequences. Li et al. [26] use an encoder-decoder model with attention for structure prediction. Deng et al. [10] and EDD [72] employ LSTM decoders and dual attention mechanisms. TableFormer [40] and VAST [22] use transformer decoders for simultaneous structure and cell bounding box prediction. DRCC [54] utilizes a semi-autoregressive two-step approach for row and column decoding, while TableVLM [6] integrates multimodal pre-training tasks. These models are typically parameter-heavy and inefficient for dense tables, with minor output errors leading to significant structural inconsistencies.

Privacy Preserving Object Detection. Key approaches in this space include homomorphic encryption [1, 9, 24, 66], which enables computations on encrypted data, allowing secure cloud-based object detection without exposing raw images. Federated learning [31, 32, 68] facilitates collaborative model training across decentralized devices, protecting data by keeping it localized while achieving com-

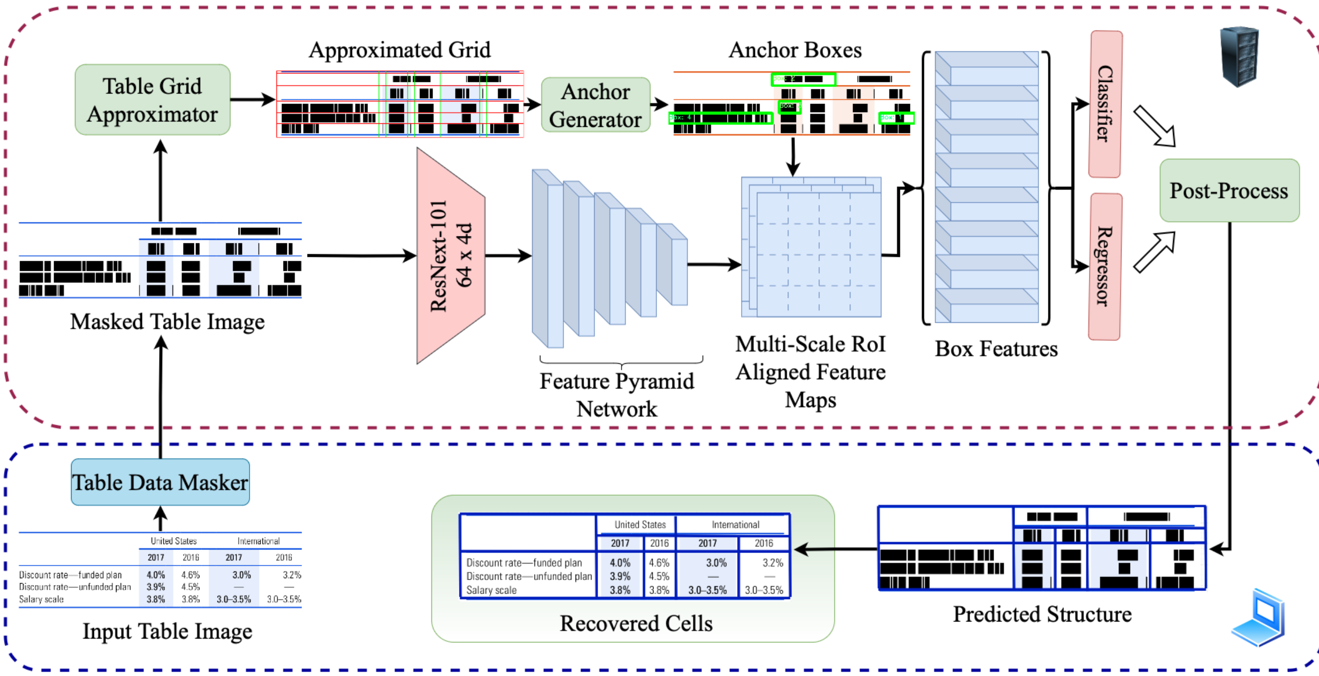


Figure 2. Architecture of *TabGuard*. Client masks content and interacts with the TSR API server for end-to-end secure table reconstruction.

petitive accuracy. Differential privacy [3, 15, 38, 59] integrates noise into the training process, providing formal privacy guarantees while preserving model performance. Secure Multi-Party Computation [2,5,60] allows multiple parties to collaboratively perform object detection without revealing their data, ensuring privacy in real-time. Lastly, lightweight, on-device methods such as those using MobileNet SSD [8, 21, 67] and elliptic curve cryptography offer efficient, privacy-preserving object detection tailored for mobile and IoT devices, balancing security with resource constraints. While these methods cater well to general object detection, they do not address challenges of complex structure and precision requirements for table image analysis. *To address these challenges and privacy concerns, we propose a robust solution that integrates data masking and inductive bias through an approximated table grid.*

3. TabGuard

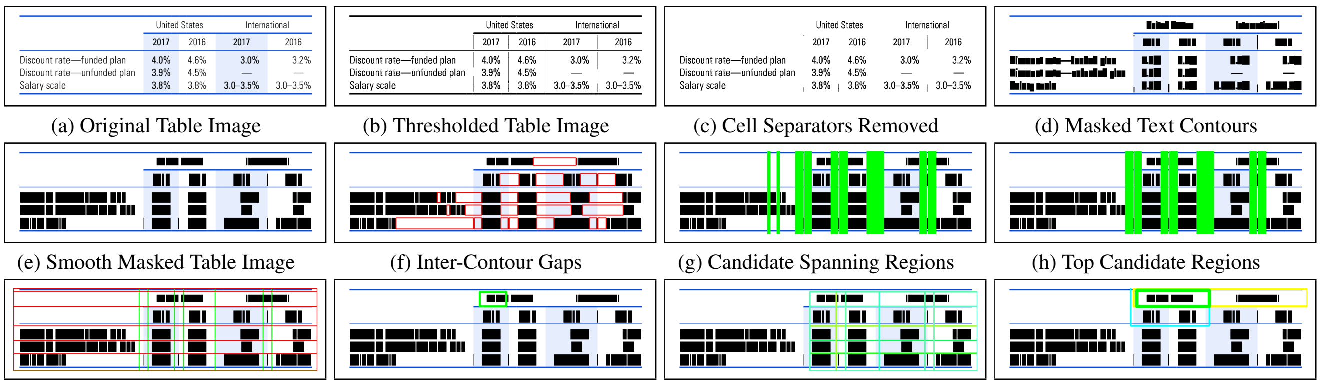
TabGuard, as shown in Figure 2, comprises: (i) a client-side content masking algorithm, (ii) a server-side *Table Grid Approximator (TGA)* that provides inductive bias and generates candidate anchors for table cell detection, (iii) a server-side *Table Cell Crypt Network (TCCN)* with a ResNext-101 64×4d backbone and specialized loss functions for table structure recognition, followed by a dataset-agnostic post-processor to refine bounding boxes and assign row/column indices, and (iv) a client-side content and structure aggregator to produce an end-to-end digitized table. Given the original table image I_t and the masked table image I_{mt}, I_{mt} is used as input to *TCCN*, which encodes the table layout into an XML format containing bounding box coordinates ($[X^i_{left}, Y^i_{top}, X^i_{right}, Y^i_{bottom}]$) and row/column spanning indices ($[R^i_{start}, C^i_{start}, R^i_{end}, C^i_{end}]$) for each predicted cell TC^i ($i \in \{0, 1, \ldots, n\}$).

Algorithm 1: Algorithm to Mask Table Content.

Given a table image I_t, this algorithm generates a masked table image I_{mt} by masking all content with black rectangular contours.

1. Apply the popular **Projection Profile algorithm** [43] for skew estimation in I_t and correct the skew accordingly by the estimated angle.
2. Convert the image to grayscale and apply **Gaussian adaptive thresholding** to binarize it.
3. Remove horizontal and vertical line segments using **Probabilistic Hough Line Transform** to remove any row/column separators.
4. Identify connected components in the resulting binary image to find text contours, representing the boundaries of text segments.
5. Sort contours by X-end and Y-end coordinates to obtain the final contours for masking the image content.

Content Masking: We propose masking of table's content as a precursor to structure recognition using a masking algorithm, which is language-agnostic and caters to all styles of tables without making any explicit assumptions. It analyzes the color distribution within the image to identify the regions of content and mask them with blacked out boxes using standard algorithms from the popular OpenCV library. Steps and Visualizations of the table masking can be



(a) Original Table Image (b) Thresholded Table Image (c) Cell Separators Removed (d) Masked Text Contours

(e) Smooth Masked Table Image (f) Inter-Contour Gaps (g) Candidate Spanning Regions (h) Top Candidate Regions

(i) Approximated Table Grid (j) Example Grid Cell (k) All Grid-Cell Anchors (l) Top K Grid-Cell Anchors

Figure 3. Steps (a) through (e) visualize the table's content masking algorithm. Steps (f) through (i) visualize the steps of table grid approximation. (j) through (l) show the anchor generation process for an example grid-cell.

referenced from Algorithm 1 and Figure 3. *The importance of our masking algorithm is highlighted by the fact that the OCR bounding boxes (DocTR [39]) across FinTabNet and SciTSR cover 93.7% of the total token area, while our algorithm covers 99.86% of the total content area. Runtime complexity of Algorithm 1 is $O(H \times W)$, where H and W are height and width of the masked table image respectively.*

Approximating the Table Grid: Given the masked table image, we find an approximation of the table's structural grid completely unsupervised. We use the distribution of text contours and inter-contour spacing to identify candidate regions for row (or line) and column separators. Initially, we consider all inter-contour spacing as prospective column separators and gaps between all vertically overlapping contours as prospective row separators. We greedily remove column separators based on the distribution of their width and the number of contours they intersect until a good quantifiable grid is obtained. Details and visualization of the algorithm can be referenced from Algorithm 2 and Figure 3. *Runtime complexity of Algorithm 2 is $O(n \ \log n)$, where n is the number of text contours. It uses* ***Interval Tree*** *data structure to identify overlaps in an optimized manner.*

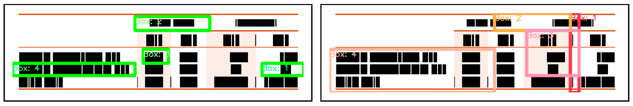
Generation of Anchor Boxes: The table grid approximated thus far is the most granular candidate grid of the table, which will not have any cues about multi-row/multi-column spanning cells. Further, the grid would generally have some false positive row and column separators, dividing the grid into smaller cells. Therefore, we add another step that merges left-to-right and vertically top-to-bottom adjacent grid cells for generating anchor boxes. *We assume that a table would have a maximum of 50 columns, the largest a cell could span column-wise. Similarly, we assume that vertically, a cell would not span more than 20 consecutive adjacent lines of the table*. We merge the adjacent boxes for every grid cell to cover all possible combinations of up to 20 row and 50 column spans. This means

Algorithm 2: Algorithm to Approximate Table's Grid.

Given a masked table image I_{mt} of height H and width W, this algorithm outputs an approximate structural grid with n_r rows and n_c columns in an unsupervised manner.

1. Identify *lines* (rows) within the table based on Y-axis overlaps of contours & for each line, identify empty spacing between adjacent contours to obtain $empty\_regions$.
2. Stretch each empty region in $empty\_regions$ from top to bottom, scoring on two dimensions — width and the percentage of lines where the region does not intersect with any text contour. All such regions are candidates for column separators, which we term as $candidate\_spanning\_regions$.
3. Filter out those regions from $candidate\_spanning\_regions$, which intersect with any text contour in at least half the total number of lines across the entire height of the image.
4. Normalize each dimension (width, s_w and percentage of non-intersecting lines, s_l) to have 0 mean and unit standard deviation followed by 0-1 scaling.
5. Compute an aggregated score for each region in $candidate\_spanning\_regions$ as $s = \sqrt{s_w^2 + s_l^2}$.
6. Apply K-Means clustering on the aggregated score and elements with higher average cluster centroid, with an upper cap of 50 sorted by the aggregated score to find $filtered\_spanning\_regions$.
7. If the size of $filtered\_spanning\_regions$ is below 20, add elements from the other cluster in descending order of aggregated score until it reaches 20. This ensures sufficient grid granularity to minimize false negatives. Then, generate an approximate table grid using the identified lines and $filtered\_spanning\_regions$.

we have approximately 1000 grid cell anchors corresponding to one grid cell. Assuming that a table contains 2000 grid cells, it means the total number of anchors is in the order of two million. Next, we employ a filtering mechanism for anchor boxes based on a score derived from a combination of simple geometrical features. We first normalize the features — intersections with text contours and inter-line gaps, number of empty lines, number of grid-row and grid-column spans, presence of empty lines above and be-



(a) Good Candidate Anchors (b) Bad Candidate Anchors

Figure 4. Sample good and bad anchors. Good anchors have high overlap with ground-truth cells, contrary to bad anchors, which may have intersections with text regions.

low, etc., by applying zero mean and unit variance scaling, followed by rescaling to a 0-1 range. This is then used to train a linear regression model that predicts the IoU (Intersection over Union) overlap with the nearest true table-cell. By excluding the bias term in the model, we ensure that it focuses solely on the relevant features, promoting consistent detection performance across different table structures. We select the top 5 scoring anchor boxes for each grid cell, collectively serving as the anchor boxes for table cell detection. With a maximum grid size assumed to be 2000, this approach effectively limits the number of anchor boxes to 10,000. The grid cells from Algorithm 2, divided by row and column separators, span the entire table and are well-aligned with adjacent ones. Each grid-cell is then merged with adjacent ones to form candidate anchors. We retain the top 10 scoring anchors per grid cell, ensuring comprehensive coverage and alignment. *Across FinTabNet and SciTSR test datasets, our anchors cater for 99.92% of ground-truth cell boxes with an IoU threshold @ 0.75.* Algorithm 3 lists the steps for anchor generation in detail and last row of Figure 3 visualizes the steps. Figure 4 illustrates the best and worst scoring anchor boxes corresponding to four randomly selected sample table image grid cells. *Runtime complexity of Algorithm 2 is $O(k \times m)$, where k and m are the number of anchors per grid-cell and number of grid-cells, respectively.*

Algorithm 3: Algorithm to Generate Candidate Anchors

Given a masked table image I_{mt} and a coarse grid from TGA, this algorithm outputs candidate anchors for TCCN in an unsupervised manner.

1. For each grid cell, recursively merge with 50 adjacent cells horizontally and 20 vertically, generating 1000 possible anchors per grid-cell.
2. Ensure at least one anchor per grid cell spans the entire image width and at least 20 vertical lines to cover various spanning scenarios.
3. Extract features for each anchor, including intersections with text contours, number of empty lines, inter-line gaps, lines spanned, presence of empty lines above/below, and the box's dimensions.
4. Normalize feature values to have 0 mean and unit variance followed by 0-1 scaling and use linear regression without bias to learn feature weights, targeting the highest IoU with ground-truth cells.
5. At test time, select top 10 scoring anchors per grid cell, ensuring alignment and coverage of all regions, including empty cells.

Cells Detection and Structure Recognition: After generating the anchor boxes, we employ our *Table Cell Crypt Network (TCCN)*, which is an enhancement of Fast R-CNN architecture [12] with a ResNext-101 64×4d backbone [62], to predict the coordinates of the table cells. With pre-generated anchors in place, the need for a Region Proposal Network (RPN) is eliminated. We maintain a ratio of 1:1 for positive to negative anchors, determined using an IoU overlap threshold of 0.5. These anchors are then multi-scale Region of Interest (RoI)-aligned with the feature pyramid to extract box features for accurately locating table cells. Alongside the standard bounding box regressor and classifier, which utilize L1 and Cross-Entropy losses, respectively, our network is augmented with alignment and continuity losses, as proposed in [48, 49]. However, instead of modeling them as L2 losses, we employ smooth L1 losses for better performance. Both alignment ($\mathcal{L}_r$ in Eq. 1) and continuity loss ($\mathcal{L}_c$ in Eq. 2) functions help *TCCN* to make precise predictions with the desired spatial characteristics. These losses are added to standard classification and regression losses for a comprehensive training objective.

$$\begin{aligned}
\mathcal{L}_{RS} &= \forall r \in R^{start} \sum_{i,j \in \, row \; r} ||Y^i_{top} - Y^j_{top}||^1_1, \\
\mathcal{L}_{RE} &= \forall r \in R^{end} \sum_{i,j \in \, row \; r} ||Y^i_{bottom} - Y^j_{bottom}||^1_1, \\
\mathcal{L}_{CS} &= \forall c \in C^{start} \sum_{i,j \in \, col \; c} ||X^i_{left} - X^j_{left}||^1_1, \\
\mathcal{L}_{CE} &= \forall c \in C^{end} \sum_{i,j \in \, col \; c} ||X^i_{right} - X^j_{right}||^1_1, \\
\mathcal{L}_{align} &= \mathcal{L}_{RS} + \mathcal{L}_{RE} + \mathcal{L}_{CS} + \mathcal{L}_{CE} \qquad (1)
\end{aligned}$$

$$\begin{aligned}
\mathcal{L}_r &= \sum_{i,j \in \, cells} ||Y^i_{top} - Y^j_{bottom}||^1_1 \cdot I\big(R^i_{start} == R^j_{end} + 1\big), \\
\mathcal{L}_c &= \sum_{i,j \in \, cells} ||X^i_{left} - X^j_{right}||^1_1 \cdot I\big(C^i_{start} == C^j_{end} + 1\big) \\
\mathcal{L}_{cont} &= \mathcal{L}_r + \mathcal{L}_c \qquad (2)
\end{aligned}$$

Post-processing: Subsequently, we employ simple dataset-agnostic post-processing[1] to refine cell boundaries and assign row/column spanning indices. It relies on cell overlaps as the sole criterion for assigning row and column indices to every predicted cell. The structure predictions from *TCCN* and postprocessor are sent back to the client, which uses a PDF extractor or OCR tools to map content to each cell based on coordinate alignment. This ensures an end-to-end privacy-driven table reconstruction.

4. Experiments and Results

Implementation: We resize all images to a resolution of 1024×1024. Anchors having an IoU overlap of more than 0.6 with a ground truth box are used as positive and others as negative samples for training in a balanced manner. Our

[1] Details of postprocessing are in the supplementary material.

Method	CAR-F1			Struct-TEDS		TEDS		AP_{50}	
	IC-13	*SciTSR*	*cTDaR*	*FTN*	*PTN*	*FTN*	*PTN*	*FTN*	*PTN*
GraphTSR [7]	87.2	95.3	-	-	-	-	-	-	-
SPLERGE [57]	95.0	92.6	-	-	-	-	-	-	-
LGPMA [47]	95.3	98.8	-	-	96.7	-	94.6	-	-
TSRFormer [27]	-	**99.6**	-	-	97.5	-	-	-	-
CascadeTabNet [44]	-	-	43.8	-	-	-	-	-	-
GTE [71]	93.5	-	45.9	91.0	93.0	-	-	-	-
TGRNet [65]	66.7	-	82.8	-	-	-	-	-	-
GuidedTSR-AO [16]	95.46	-	-	-	-	-	-	-	-
SEM [69]	-	-	-	-	-	-	93.7	-	-
EDD [72]	-	-	-	90.6	89.9	-	88.3	-	79.2
TableFormer [40]	-	-	-	96.8	96.8	-	93.6	-	82.1
MTL-TabNet [35]	-	-	-	**98.8**	97.9	-	-	-	96.7
TabStructNet [49]	90.6	92.0	58.3	-	-	-	90.1	-	-
VAST [22]	96.5	99.5	58.6	98.6	97.2	**98.2**	96.3	-	94.8
NCGM [29]	98.8	98.8	85.3	-	95.4	-	-	-	-
LORE [63]	98.9	98.7	88.3	-	**98.1**	-	-	-	-
GridFormer [36]	-	99.3	-	98.6	97.0	-	95.8	-	-
Faster RCNN* [51]	84.2	85.3	33.4	78.8	80.3	76.1	77.4	71.5	72.6
RetinaNet*	83.6	86.2	32.4	77.3	80.1	75.4	77.1	70.8	72.2
YOLO v9 †	89.6	90.2	40.3	83.6	86.1	81.8	83.5	77.4	78.7
Deformable-DETR* [73]	92.2	93.9	61.4	91.7	92.4	-	-	87.3	89.1
Anchor-DETR* [61]	95.4	96.8	70.2	94.9	95.6	-	-	91.0	92.3
RetinaNet †,*	98.6	99.0	85.3	96.8	97.2	95.1	95.6	93.6	93.8
$TabGuard^{cell}$	**99.2**	99.1	**89.9**	97.8	**98.1**	97.1	**97.3**	**95.7**	**96.2**
$TabGuard^{content}$	**99.2**	99.2	NA	98.1	**98.3**	97.1	**97.3**	**95.9**	**96.4**

Table 1. Comparison using CAR-F1 scores at IoU=0.5 on IC-13, SciTSR, cTDaR datasets; and using S-TEDS and TEDS on FTN and PTN datasets. Training and testing environments for each test dataset is consistent across methods for fairness. Method M^* includes alignment and continuity losses [48, 49], and $M^\dagger$ uses anchors from TGA. *TabGuard* has been trained and tested using masked table images. $TabGuard^{cell}$ evaluates cell-level bounding boxes, and $TabGuard^{content}$ evaluates content-level bounding boxes, ensuring fair comparison across different environments. Since *TabGuard* generates rectangular bounding boxes, it does effectively handle misaligned or curved tables. Therefore, we opt not to compare our method on the WTW dataset.

model can be trained on a single NVIDIA 1080TI GPU with a batch size of 2. Regularization parameters corresponding to alignment and continuity losses are set to 0.01. We smooth out the cell boxes by identifying overlaps along X and Y axes, and the final coordinates are translated to PDF coordinates to extract content[2].

Datasets and Evaluation: We use FinTabNet [71] (FTN) and SciTSR [7] datasets for training. We evaluate TabGuard on FinTabNet and PubTabNet [72] (PTN) datasets using Tree Edit Distance Similarity (TEDS) [72] and Structural TEDS (S-TEDS, ignoring cell content). For ICDAR-2013 [14] (IC-13), SciTSR, and ICDAR-2019 [11] (cTDaR) datasets we use F1 score on Cell Adjacency Relations (CAR-F1) for evaluation [14]. Since *TCCN* has alignment & continuity constraints, we split the horizontal & vertical gaps between every adjacent pair of cells equally and extend their boundaries to ensure proper alignment. We consistently use IoU of 0.5 for all evaluations. We evaluate TabGuard on both cell-level and content-level bounding boxes. To obtain content-level boxes, we identify masked contours within a predicted cell and accordingly identify the predicted content bounding boxes within each table cell.

[2]Additional details in the supplementary material.

Method/IoU	**0.6**	**0.7**	**0.8**	**0.9**	**W.Avg**
NLPR-PAL	0.37	0.31	0.20	0.04	0.21
CascadeTabNet	0.44	0.35	0.19	0.04	0.23
TabGuard	0.86	0.73	0.31	0.07	0.446

Table 2. Comparison on IC19 Track B2 on varying IoU thresholds.

Comparative Study We assess the performance of our method on scanned and cropped table images and table images extracted from PDF documents. To ensure consistency in comparisons, we additionally present our results using content-level bounding boxes, as depicted in the last row of Table 1. To derive content-level boxes, we identify masked contours within a predicted cell and utilize

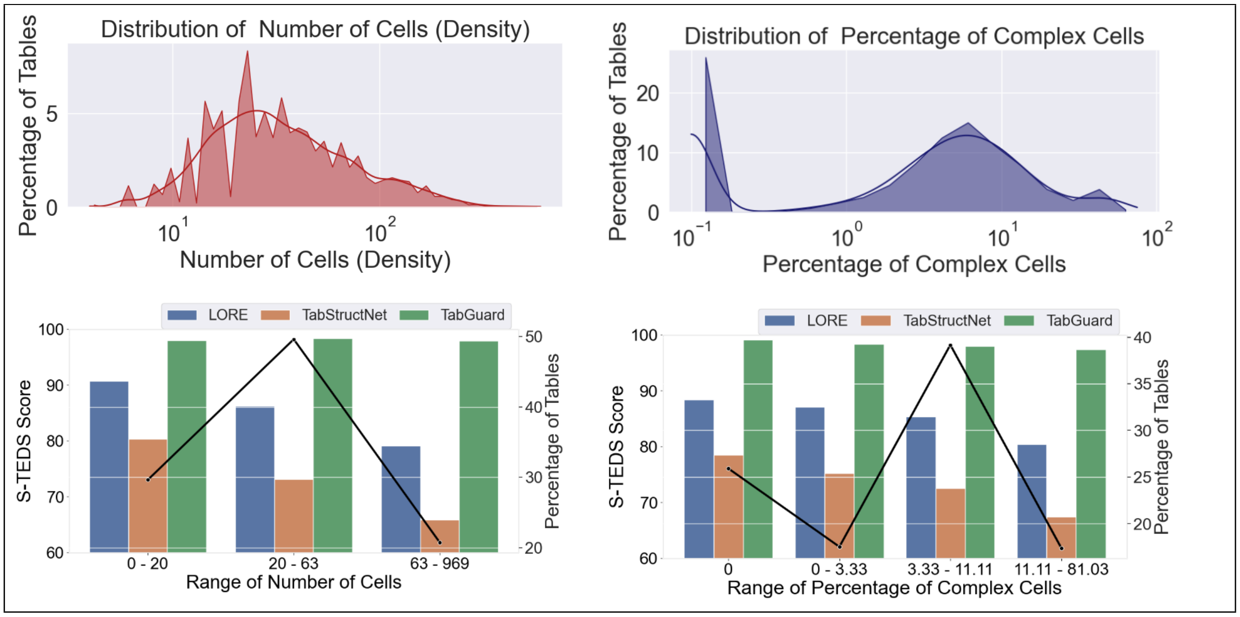


Figure 5. Top-Left and Top-Right plots indicate log-scale distribution of table density with respect to number of cells and tables with varying complexity (multi-column/row/line) of cells. Bottom-Left plot compares performances of LORE [63], TabStructNet [49] and *TabGuard* with against varying table densities. Bottom-Left plot compares performances of LORE [63], TabStructNet [49] and *TabGuard* with against varying table complexities. All distributions and performances are measured on FinTabNet-Test dataset with S-TEDS evaluation metric with masked table images. Black line in the second row shows the linear-scale dataset distribution.

these coordinates to determine each table cell's expected content-level bounding boxes. As indicated in Table 1, our method surpasses all prior approaches on the ICDAR-2013 and cTDaR datasets, achieving a margin of 0.3% and 1.3% Cell Adjacency Relation F1 scores, respectively. Regarding the ICDAR-2013 dataset, we follow a consistent evaluation protocol employed by [29, 49, 63], where a partial dataset was utilized for fine-tuning and evaluation respectively. In case of the cTDaR dataset, we compute the results using an IoU threshold of 0.5, following the approach of the competitive baseline method GTE. Table 1 presents the weighted average F1 score. On the SciTSR, our method surpasses the performance of most prior methods, achieving an F1 score within a 0.5% range of SoTA while maintaining content privacy. TabGuard achieves state-of-the-art S-TEDS and TEDS scores on PubTabNet. Nonetheless, we observe that some images in the FinTabNet dataset have incorrect annotations, particularly those containing multi-row spanning cells[3]. Interestingly, our model generates the correct structure for such cases compared to the original annotation. Additionally, we report the performance of cell detection using average precision (AP) at an IoU threshold of 0.5 on the PubTabNet and FinTabNet datasets. Table 2 compares the performance of TabGuard on ICDAR-2019 Track B2 dataset on varying IoU thresholds. It is crucial to highlight that input to our solution for all datasets are masked table images. Our findings show that the predominant characteristic in identifying the table layout is the location of content rather than the content itself.

Comparison in Privacy Preserving Scenario: To evaluate our method against the current state-of-the-art, we fine-tune LORE [63] and TabStructNet [49] on masked table images from the FTN-train dataset using their respective open-source implementations. While it might seem intuitive that masking content should enhance the performance of all existing methods, we observe from Table 3 that the performance of [47, 49, 63] decreases notably when trained and tested on masked images. The incorrect cases primarily arise in cells spanning multiple lines and where the inter-contour gap is more comprehensive than average within the same cell. We attribute the superior performance of *TabGuard* on masked images to the anchors generated using the *Table Grid Approximator*, which provides additional cues to the model to aid in table structure recognition.

Ablation Study We perform a series of ablation experiments to validate the efficacy of our proposed modules. Table 4 compares use of differential privacy, OCR based content-masking and our contour based content masking for ensuring privacy preservation. For a fair ablation study, we use Faster RCNN as the fixed architecture augmented by alignment and continuity losses [48, 49] without using anchors from TGA. Table 5 illustrates that our model, trained on masked images, can proficiently process images with un-

[3]Details of such instances are available in our supplementary material.

Method	Train Mask	Test Mask	S-TEDS
LORE [63]	✗	✗	96.7
LORE [63]	✗	✓	71.1
LORE [63]	✓	✓	85.2
TabStructNet [49]	✗	✗	89.8
TabStructNet [49]	✗	✓	64.5
TabStructNet [49]	✓	✓	72.8
TabGuard	✗	✗	93.6
TabGuard	✗	✓	91.4
TabGuard	✓	✓	98.2

Table 3. Presents a comparison of training and testing variations using masked and unmasked table images for *TabGuard* against LORE [63]. We maintain consistency by utilizing the FTN-Train and FTN-Test datasets for training and evaluation. When testing *TabGuard* on original images, OCR bounding boxes are employed for grid approximation and anchor generation.

Method	*CAR-F1*			S-TEDS	
	IC-13	*Sci-C*	*IC-19*	*FTN*	*PTN*
Faster RCNN	81.3	81.7	31.1	76.5	79.4
Faster RCNN^{DP}	74.1	74.6	21.5	71.2	72.6
Faster RCNN^{OCR}	82.4	82.8	27.6	77.4	78.9
Faster RCNN^{CM}	84.2	85.3	33.4	78.8	80.3

Table 4. Impact of privacy strategy. DP, OCR and CM indicate additional, differential privacy, OCR content masking and contours based content masking. alignment and continuity losses are used for all. TGA and custom anchors are not used for fairness.

masked content during testing. For the unmasked regions, we utilize DocTR [39] OCR to acquire cell-level bounding boxes employed in grid approximation and anchor generation. The findings also indicate that performance remains consistent across training and testing domains, especially when the majority of the table's content is masked. Table 6 shows the effectiveness of our TGA and anchor generation for table cells detection. The reduced number of anchor boxes reduces the search space for the global optimum. It improves optimization performance by 2.5 times[4]. Prior approximation and generation of anchor boxes also allow for better performance in case of unseen table styles in a cross-domain setup. Table 6 highlights the fact that switching the network backbone from ResNext-101 to ResNet-18 leads to small impact on performance while significantly reducing the number of parameters.

Dense and Complex Tables: To study the effectiveness of our method on densely packed and complex (multi-row/multi-column and multi-line spanning cells) tables, we analyze the comprehensively analyze characteristic distributions and compare our method against LORE [63] and TabStructNet [49]. For fairness, we use open-source implementations of the two methods, fine-tune them using masked images from the FinTabNet-Train dataset, and evaluate them on the FinTabNet-Test dataset. Figure 5 demonstrates our method's superiority in privacy-preserving scenarios across varying table densities and complexities.

[4]Qualitative examples and details on anchors distribution and optimization are in the supplementary material.

Train Dataset	Test Dataset	% Content Masked				
		100%	75%	50%	25%	0%
SciTSR	SciTSR	99.1	98.3	96.5	94.2	92.5
SciTSR	FTN	96.4	93.2	92.9	89.3	86.4
FTN	SciTSR	98.8	97.1	95.9	94.1	92.3
FTN	FTN	98.2	97.6	96.4	95.1	93.7

Table 5. Impact of content masking on domain adaptation. All quantitative scores are measured in terms of CAR-F1 scores.

BackBone	Anchors	#Model Params	CAR-F1 Score
ResNet -18	RPN	30.3M	88.3
	TGA	29.0M	97.1
ResNxet-50 32x4d	RPN	40.2M	89.7
	TGA	41.5M	97.5
ResNxet-101 64x4d	RPN	98.6M	91.1
	TGA	99.9M	98.2

Table 6. Illustrates the impact of inductive bias and backbones through training and testing on the FinTabNet dataset. CAR-F1 scores on Cell Detection at the IoU threshold of 0.5 are reported.

Impact and Limitations: TabGuard is specifically tailored for scanned images of cropped tables, focusing on extracting table structures in a controlled 2D environment. It does not extend to scenarios where tables are captured using camera devices, with challenges such as curvature, perspective distortions, or 3D effects. Moreover, our approach does not support tables with complex embedded entities, such as nested tables, graphs, or images, which may require more advanced parsing techniques. However, in our experiments with skewed images—where tables appear slightly rotated, we found that applying skew correction as a preprocessing step successfully addressed minor misalignments, however it may not be sufficient for more severe distortions or images with significant perspective changes.

5. Conclusion

Through our simple yet effective solution, we take a step towards privacy-preserving table structure recognition. We show that using prior in the form of an approximated grid structure can significantly improve performance. Experimental results show that TabGuard[5] achieves state-of-the-art performance on benchmark datasets, and can effectively tackle dense and complex tables, agnostic of it's content.

[5]This work is supported by MeitY, Government of India.

References

[1] Ahmad Al Badawi, Chao Jin, Jie Lin, Chan Fook Mun, Sim Jun Jie, Benjamin Hong Meng Tan, Xiao Nan, Khin Mi Mi Aung, and Vijay Ramaseshan Chandrasekhar. Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1330–1343, 2020. 2

[2] Tianyu Bai, Song Fu, and Qing Yang. Privacy-preserving object detection with secure convolutional neural networks for vehicular edge computing. *Future Internet*, 14(11):316, 2022. 3

[3] Zhiqi Bu, Jialin Mao, and Shiyun Xu. Scalable and efficient training of large convolutional neural networks with differential privacy. *Advances in Neural Information Processing Systems*, 35:38305–38318, 2022. 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229, 2020. 1

[5] Imen Chakroun, Tom Vander Aa, Roel Wuyts, and Wilfried Verarcht. Privacy-preserving multi-party machine learning for object detection. In *2021 IEEE Global Conference on Artificial Intelligence and Internet of Things (GCAIoT)*, pages 7–13. IEEE, 2021. 3

[6] Leiyuan Chen, Chengsong Huang, Xiaoqing Zheng, Jinshu Lin, and Xuan-Jing Huang. TableVLM: Multi-modal pre-training for table structure recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 2437–2449, 2023. 2

[7] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. Complicated table structure recognition. *arXiv*, 2019. 1, 2, 6

[8] Yu-Chen Chiu, Chi-Yi Tsai, Mind-Da Ruan, Guan-Yu Shen, and Tsu-Tian Lee. Mobilenet-ssdv2: An improved object detection model for embedded systems. In *2020 International conference on system science and engineering (ICSSE)*, pages 1–5. IEEE, 2020. 3

[9] Kuan-Yu Chu, Yin-Hsi Kuo, and Winston H Hsu. Real-time privacy-preserving moving object detection in the cloud. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 597–600, 2013. 2

[10] Yuntian Deng, David Rosenberg, and Gideon Mann. Challenges in end-to-end neural scientific table recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 894–901, 2019. 2

[11] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. ICDAR 2019 competition on table detection and recognition (cTDaR). In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1510–1515, 2019. 6

[12] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 5

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 1

[14] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. ICDAR 2013 table competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1449–1453, 2013. 6

[15] Aditya Golatkar, Alessandro Achille, Yu-Xiang Wang, Aaron Roth, Michael Kearns, and Stefano Soatto. Mixed differential privacy in computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8376–8386, 2022. 3

[16] Khurram Azeem Hashmi, Didier Stricker, Marcus Liwicki, Muhammad Noman Afzal, and Muhammad Zeshan Afzal. Guided table structure recognition through anchor optimization. *IEEE Access*, 9:113521–113534, 2021. 6

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017. 1, 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[20] Martin Holeček, Antonín Hoskovec, Petr Baudiš, and Pavel Klinger. Line-items and table understanding in structured documents. *arXiv preprint arXiv:1904.12577*, 2019. 2

[21] Andrew G Howard. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[22] Yongshuai Huang, Ning Lu, Dapeng Chen, Yibo Li, Zecheng Xie, Shenggao Zhu, Liangcai Gao, and Wei Peng. Improving table structure recognition with visual-alignment sequential coordinate modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11134–11143, 2023. 2, 6

[23] Saqib Ali Khan, Syed Muhammad Daniyal Khalid, Muhammad Ali Shahzad, and Faisal Shafait. Table structure extraction with Bi-directional Gated Recurrent Unit networks. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1366–1371, 2019. 2

[24] Keonhyeong Kim and Im Young Jung. Secure object detection based on deep learning. *Journal of Information Processing Systems*, 17(3):571–585, 2021. 2

[25] Eunji Lee, Jaewoo Park, Hyung Il Koo, and Nam Ik Cho. Deep-learning and graph-based approach to table structure recognition. *Multimedia Tools and Applications*, 81(4):5827–5848, 2022. 2

[26] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. TableBank: Table benchmark for image-based table detection and recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1918–1925, 2019. 1, 2

[27] Weihong Lin, Zheng Sun, Chixiang Ma, Mingze Li, Jiawei Wang, Lei Sun, and Qiang Huo. TSRFormer: Table structure recognition with transformers. In *Proceedings of the 30th

ACM International Conference on Multimedia*, pages 6473–6482, 2022. 2, 6
[28] Hao Liu, Xin Li, Mingming Gong, Bing Liu, Yunfei Wu, Deqiang Jiang, Yinsong Liu, and Xing Sun. Grab what you need: Rethinking complex table structure recognition with flexible components deliberation. *arXiv preprint arXiv:2303.09174*, 2023. 2
[29] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, and Bo Ren. Neural Collaborative Graph Machines for table structure recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4533–4542, 2022. 2, 6, 7
[30] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, Bo Ren, and Rongrong Ji. Show, read and reason: Table structure recognition with flexible context aggregator. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1084–1092, 2021. 2
[31] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13172–13179, 2020. 2
[32] Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Federated learning-powered visual object detection for safety monitoring. *AI Magazine*, 42(2):19–27, 2021. 2
[33] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 2
[34] Rujiao Long, Wen Wang, Nan Xue, Feiyu Gao, Zhibo Yang, Yongpan Wang, and Gui-Song Xia. Parsing table structures in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 944–952, 2021. 2
[35] Nam Tuan Ly and Atsuhiro Takasu. An end-to-end local attention based model for table recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 20–36, 2023. 6
[36] Pengyuan Lyu, Weihong Ma, Hongyi Wang, Yuechen Yu, Chengquan Zhang, Kun Yao, Yang Xue, and Jingdong Wang. Gridformer: Towards accurate table structure recognition via grid prediction. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 7747–7757, 2023. 2, 6
[37] Chixiang Ma, Weihong Lin, Lei Sun, and Qiang Huo. Robust table detection and structure recognition from heterogeneous document images. *arXiv preprint arXiv:2203.09056*, 2022. 2
[38] Yunlong Mao, Shanhe Yi, Qun Li, Jinghao Feng, Fengyuan Xu, and Sheng Zhong. Learning from differentially private neural activations with edge computing. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 90–102. IEEE, 2018. 3
[39] Mindee. doctr: Document text recognition. https://github.com/mindee/doctr, 2021. 4, 8
[40] Ahmed Nassar, Nikolaos Livathinos, Maksym Lysak, and Peter Staar. Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4614–4623, 2022. 2, 6
[41] Manabu Ohta, Ryoya Yamada, Teruhito Kanazawa, and Atsuhiro Takasu. A cell-detection-based table-structure recognition method. In *ACM Symposium on Document Engineering*, pages 1–4, 2019. 2
[42] Shubham Singh Paliwal, D Vishwanath, Rohit Rahul, Monika Sharma, and Lovekesh Vig. TableNet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 128–133, 2019. 1, 2
[43] Wolfgang Postl. Detection of linear oblique structures and skew scan in digitized documents. In *Proc. Int. Conf. on Pattern Recognition*, pages 687–689, 1986. 3
[44] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. 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 (CVPRW)*, pages 572–573, 2020. 2, 6
[45] Shah Rukh Qasim, Jan Kieseler, Yutaro Iiyama, and Maurizio Pierini. Learning representations of irregular particle-detector geometry with distance-weighted graph networks. *The European Physical Journal C*, 79(7):1–11, 2019. 2
[46] Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. Rethinking table parsing using graph neural networks. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 142–147, 2019. 2
[47] Liang Qiao, Zaisheng Li, Zhanzhan Cheng, Peng Zhang, Shiliang Pu, Yi Niu, Wenqi Ren, Wenming Tan, and Fei Wu. LGPMA: Complicated table structure recognition with local and global pyramid mask alignment. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 99–114, 2021. 6, 7
[48] Sachin Raja, Ajoy Mondal, and CV Jawahar. Visual understanding of complex table structures from document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2299–2308, 2022. 1, 2, 5, 6, 7
[49] Sachin Raja, Ajoy Mondal, and C. V. Jawahar. Table structure recognition using top-down and bottom-up cues. In *European Conference on Computer Vision (ECCV)*, pages 70–86, 2020. 1, 2, 5, 6, 7, 8
[50] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271, 2017. 1
[51] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NIPS)*, 2015. 1, 2, 6
[52] Pau Riba, Anjan Dutta, Lutz Goldmann, Alicia Fornés, Oriol Ramos, and Josep Lladós. Table detection in invoice docu-

ments by graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 122–127, 2019. 2

[53] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. DeepDeSRT: Deep learning for detection and structure recognition of tables in document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1162–1167, 2017. 2

[54] Huawen Shen, Xiang Gao, Jin Wei, Liang Qiao, Yu Zhou, Qiang Li, and Zhanzhan Cheng. 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, International Joint Conferences on Artificial Intelligence Organization*, pages 1369–1377, 2023. 2

[55] Xinyi Shen, Lingjun Kong, Yunchao Bao, Yaowei Zhou, and Weiguang Liu. RCANet: A rows and columns aggregated network for table structure recognition. In *2022 3rd Information Communication Technologies Conference (ICTC)*, pages 112–116, 2022. 2

[56] Brandon Smock, Rohith Pesala, and Robin Abraham. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4634–4642, 2022. 1

[57] Christopher Tensmeyer, Vlad Morariu, Brian Price, Scott Cohen, and Tony Martinezp. Deep splitting and merging for table structure decomposition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 114–121, 2019. 1, 2, 6

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural information Processing Systems*, 30, 2017. 2

[59] Baoping Wang, Duanyang Feng, Junyu Su, and Shiyang Song. An effective federated object detection framework with dynamic differential privacy. *Mathematics*, 12(14):2150, 2024. 3

[60] Ruonan Wang, Min Luo, Qi Feng, Cong Peng, and Debiao He. Multi-party privacy-preserving faster r-cnn framework for object detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023. 3

[61] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2567–2575, 2022. 1, 6

[62] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1492–1500, 2017. 5

[63] Hangdi Xing, Feiyu Gao, Rujiao Long, Jiajun Bu, Qi Zheng, Liangcheng Li, Cong Yao, and Zhi Yu. LORE: Logical location regression network for table structure recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence 37(3), 2992-3000.*, 2023. 2, 6, 7, 8

[64] Wenyuan Xue, Qingyong Li, and Dacheng Tao. ReS2TIM: Reconstruct syntactic structures from table images. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019. 1

[65] Wenyuan Xue, Baosheng Yu, Wen Wang, Dacheng Tao, and Qingyong Li. TGRNet: A table graph reconstruction network for table structure recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1295–1304, 2021. 2, 6

[66] Ryo Yonetani, Vishnu Naresh Boddeti, Kris M Kitani, and Yoichi Sato. Privacy-preserving visual learning using doubly permuted homomorphic encryption. In *Proceedings of the IEEE international conference on computer vision*, pages 2040–2050, 2017. 2

[67] Ayesha Younis, Li Shixin, Shelembi Jn, and Zhang Hai. Real-time object detection using pre-trained deep learning models mobilenet-ssd. In *Proceedings of 2020 6th International Conference on Computing and Data Engineering*, pages 44–48, 2020. 3

[68] Peihua Yu and Yunfeng Liu. Federated object detection: Optimizing object detection model with federated learning. In *Proceedings of the 3rd international conference on vision, image and signal processing*, pages 1–6, 2019. 2

[69] Zhenrong Zhang, Jianshu Zhang, Jun Du, and Fengren Wang. Split, embed and merge: An accurate table structure recognizer. *Pattern Recognition*, 126:108565–108578, 2022. 2, 6

[70] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16965–16974, 2024. 1

[71] Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. 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 (WACV)*, pages 697–706, 2021. 6

[72] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. In *European Conference on Computer Vision (ECCV)*, pages 564–580, 2020. 1, 2, 6

[73] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 6