

Towards End-to-End Semi-Supervised Table Detection with Deformable Transformer

Tahira Shehzadi*^{1,2,4}[0000-0002-7052-979X], Khurram Azeem Hashmi^{1,2,4}[0000-0003-0456-6493], Didier Stricker^{1,2,4},
 Marcus Liwicki³, and Muhammad Zeshan Afzal^{1,2,4}[0000-0002-0536-6867]

¹ Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany

² Mindgarage, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany

³ Department of Computer Science, Luleå University of Technology, 971 87 Luleå, Sweden

⁴ German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany
 {tahira.shehzadi@dfki.de}

Abstract. Table detection is the task of classifying and localizing table objects within document images. With the recent development in deep learning methods, we observe remarkable success in table detection. However, a significant amount of labeled data is required to train these models effectively. Many semi-supervised approaches are introduced to mitigate the need for a substantial amount of label data. These approaches use CNN-based detectors that rely on anchor proposals and post-processing stages such as NMS. To tackle these limitations, this paper presents a novel end-to-end semi-supervised table detection method that employs the deformable transformer for detecting table objects. We evaluate our semi-supervised method on PubLayNet, DocBank, ICADR-19 and TableBank datasets, and it achieves superior performance compared to previous methods. It outperforms the fully supervised method (Deformable transformer) by +3.4 points on 10% labels of TableBank-both dataset and the previous CNN-based semi-supervised approach (Soft Teacher) by +1.8 points on 10% labels of PubLayNet dataset. We hope this work opens new possibilities towards semi-supervised and unsupervised table detection methods.

Keywords: Semi-Supervised Learning · Deformable Transformer · Table Analysis · Table Detection.

1 Introduction

A visual summary is the main aspect of different applications in document analysis, such as recognizing graphical components in the visualization pipeline and summarizing the content of a document. As a result, localizing and detecting graphical items such as tables will be an important action in the analysis and summary of the document. Due to the increase in the number of documents, manually retrieving the table data is no longer practical. Automated processes offer efficient, reliable, and successful solutions for manual tasks. Previously, optical character recognition [1,2] and rule-based [3,4,5] table detection approaches were used to identify and locate them. Then, some automated methods [6,7,8] have been suggested to detect tables. However, these approaches are often rule-based because the documents have a set structure or dimension [9]. Moreover, they cannot generalize to a new table structure, such as borderless tables. Later on, deep learning methods were used by researchers to identify them [10,11,12,13], and shows that machine-learning approaches are more effective than traditional methods [14].

Deep learning approaches [15,16,17,18,19,20] do not rely on rules and can accurately generalize the problem. However, deep learning models take a considerable quantity of labeled data for training. These supervised methods achieve impressive results on public benchmarks, and their performance cannot be translated into industrial applications unless similar large-scale annotated datasets exist in that domain. It is potentially error-prone and time-consuming to generate this data manually or via other pre-processing approaches. Therefore, it is important to develop a semi-supervised approach due to concerns about the availability of labeled training data, which shifts the problem from a supervised to a semi-supervised setting. Recently, semi-supervised learning-based methods are introduced in computer vision containing two detectors. The first detector provides pseudo labels for unlabeled data. The second detector trains using pseudo labels generated by the first detector and a small percentage of label data and provides final predictions. Both detectors update each other during training. This approach has been described in several works, including [21,22,23,24]. In most cases, the first detector is not strong enough, which can negatively impact the pseudo-labeling process. Moreover, previous semi-supervised approaches used CNN-based networks [11] that depend on anchors to generate region

proposals and post-processing stages such as Non-Maximal suppression (NMS) to reduce the number of overlapping predictions.

To address these limitations, this paper proposes a semi-supervised table detection approach that employs the deformable transformer [25]. It generates pseudo-labels for unlabeled data and then trains the detector using them and a small quantity of label data in each iteration. This approach aims to improve the pseudo-label generation procedure by iteratively refining the pseudo-labels and the detector. It involves training in two modules. The teacher module contains a pseudo-labeling framework. The student module is the final detection network that uses pseudo-labels and a small quantity of label data. The teacher module is simply an Exponential Moving-Average (EMA) of the student module, which ensures that the pseudo-label generation and detection modules are constantly updating each other. Unlike other pseudo-labeling methods, we propose the idea of employing the deformable transformer that allows completing the pseudo-labeling process without needing object proposals and post-processing steps as Non-maximal suppression (NMS). Another benefit is having a dynamic effective receptive field to adapt for tables of different sizes and scales in the input image. This allows the network to effectively detect tables of varying sizes and orientations, making it more robust and versatile. Additionally, this framework has a reinforcing effect, providing that the Teacher model consistently monitors the Student model. In this paper, we show through empirical evidence that this semi-supervised table detection approach that uses a deformable transformer can produce results comparable to CNN-based approaches without needing object proposals and post-processing steps such as Non-maximal suppression (NMS).

In summary, the main contributions of the paper are as follows:

- We present an end-to-end semi-supervised table detection method that employs the deformable transformer and allows completing the pseudo-labeling process without needing object proposals and post-processing steps such as Non-maximal suppression (NMS).
- We formulate the problem of table detection as an object detection problem and leverage the potential of deformable detection transformer for this task. To the best of our knowledge, this work is the first that exploits the transformer-based method in a semi-supervised setting.
- We perform an exhaustive evaluation on four different datasets, PubLayNet, DocBank, ICDAR-19 and TableBank, and produce results comparable to CNN-based semi-supervised approaches without needing object proposals process and post-processing steps such as NMS.

2 Related Work

Table detection is an essential task for document image analysis. Many researchers have proposed different approaches for detecting tables containing arbitrary structures in document images. Previously, most presented approaches used custom rules or relied on extra meta-data input to deal with table detection tasks [26,27,28,29]. Recently, researchers employed statistical methods [30] and deep learning approaches to make the table detection systems more generalizable [15,31,32,33]. This section gives a detailed summary of these techniques and an overview of the CNN-based semi-supervised object detection methods.

2.1 Rule-based Approaches

To the best of our knowledge, Itonori et al. [26] presented a table detection approach for the first time on document images. This method represents the table as a text block that uses specified rules. Later, [28] introduced a table detection approach that works on horizontal and vertical lines. Pyreddy et al. [34] proposed a procedure that extracts tabular regions from the text using custom heuristics. Pivk et al. [35] presented a system that transforms HTML format table documents into logical forms. It introduces an appropriate tabular layout employed for extracting tables. Hu et al. [36] presented a table detection approach that relies on white regions and vertically connected elements in document images. Readers can find a complete overview of these rule-based methods in [3,4,5,37,38]. Though rule-based approaches perform fine on document images with matching table formats, these methods can not provide generic solutions. Therefore, systems with more generalizable abilities are needed to solve table detection tasks on document data.

2.2 Learning-based Approaches

Cesarini et al. [39] presented a supervised learning system for detecting table objects in document images. It converts a document image into an MXY tree model and labels the blocks as tables confined in horizontal and vertical lines. Hidden Markov Models [40,41] and the SVM classifier with traditional heuristics [42] are applied to document images for table detection. Though these machine learning approaches performed better than ruled-based approaches on documents, these methods need additional information, such as ruling lines. Deep Learning-based approaches outperformed traditional approaches in accuracy and efficiency. These methods are categorised into object detection, semantic segmentation, and bottom-up approaches.

Semantic segmentation-based Approaches. These approaches [43,44,45,46] consider table detection a segmentation task and apply available semantic segmentation networks to generate segmentation masks on the pixel level and then combine table regions to provide final table detection. These methods performed better than traditional approaches on several benchmark datasets [47,48,49,50,51,52,53]. Yang et al. [43] presented a fully convolutional network (FCN) [54] for page object segmentation, which combines linguistic and visual features to enhance segmentation results for table and other page object detection. He et al. [44] presented a multi-scale FCN that provides segmentation masks table/text areas and their related contours and then refined to get final table blocks.

Bottom-up Approaches These approaches consider table detection as a graph-labeling task and define graph nodes as page objects and graph edges connection between page objects. Li et al. [55] extracted line areas using the classic layout analysis approach, then used two CNN-CRF networks to categorise them into four categories: text, figure, formula and table and then provided a prediction of the corresponding cluster for pair of line areas. Holecek et al. [56] and Riba et al. [57] considered text areas as nodes, formed a graph to determine the design per document and then employed graph-neural networks for node-edge classification. These approaches rely on specific assumptions, such as text line boxes as an extra input.

Object Detection-based Approaches. Detecting tables from document images can be represented as an object detection task, with table objects treated as natural objects. Hao et al. [58] and Yi et al. [59] applied R-CNN for detecting tables, but the performance of these approaches still relies on heuristic rules as in previous methods. Later, more efficient single-stage object detectors like RetinaNet [60] and YOLO [61] and two-stage object detectors like Fast R-CNN [10], Faster R-CNN [11], Mask R-CNN [62], and Cascade Mask R-CNN [63] were applied for other document objects such as figures and formulas detection in document images [64,65,66,67,68,69,70,15,16,17]. Furthermore, [65,69,71] applied different image transformation approaches, such as coloration and dilation, to improve the results further. Siddiqui et al. [72] combined deformable-convolution and RoI-Pooling [73] into Faster R-CNN to provide a more efficient network for geometrical modifications. Agarwal et al. [70] used a composite network [74] as a backbone with deformable convolution to increase the performance of two-stage Cascade R-CNN. These CNN-based object detectors have a few heuristic stages, like proposals generating step and post-processing steps such as non-maximal suppression (NMS). Our semi-supervised approach considers detection a set prediction task, eliminating the anchor generation and post-processing stages such as NMS and providing a simpler and more efficient detection pipeline.

2.3 Semi-supervised Object Detection

Semi-supervised learning approaches in object detection are divided into two types: consistency-based approaches [75,76] and pseudo-label generation-based approaches [77,78,79,80,81,82,83]. Our method falls into the pseudo-label type. Previous work [77,78] combined prediction results from varied data augmentation techniques to produce pseudo-labels for unlabeled data, while [79] trained a SelectiveNet to generate the pseudo-labels. In [79], a box from unlabeled data was placed onto labeled data and evaluated localization consistency on the labeled images. However, this method requires a very complex detection procedure due to the modification of the image. STAC [82] presented to perform strong augmentation on the data for pseudo-label generation and weak augmentation for model training. We propose an end-to-end semi-supervised table detection method that employs the deformable transformer. Similar to other pseudo-label generation approaches [77,78,79,82,83], it follows a multi-level training mechanism. It effectively avoids the need for anchors generation stage and post-processing steps such as Non-Maximal suppression (NMS).

3 Methodology

First, we revisit Deformable DETR, a modern transformer-based object detector, in Section 3.1. Later, we explain the proposed semi-supervised learning mechanism and its pseudo-label generation module in Sections 3.2.

3.1 Revisiting Deformable DETR

Deformable DETR [25] contains a Transformer encoder-decoder network that considers object detection as a set-predictions task. It uses Hungarian loss and avoids overlapped predictions for ground-truth bounding boxes through bipartite matching. It eliminates the need for hand-crafted elements such as anchors and post-processing stages such as Non-maximal suppression (NMS) used in CNN-based object detectors. Deformable DETR is an extension of the DETR [84] architecture that addresses some of the limitations of DETR, such as slow training convergence and poor performance on small objects. Deformable DETR introduces deformable convolutions into the architecture, which allows for more flexible modeling of object shapes and better handling of objects of varying scales. This can lead to improved performance, particularly on small objects, and faster convergence during training. Here, we provide an overview of the encoder-decoder network, Multi-scale Feature processing and attention mechanism of deformable DETR. Figure 1 shows all modules of the deformable transformer, including multi-scale features and encoder-decoder network.

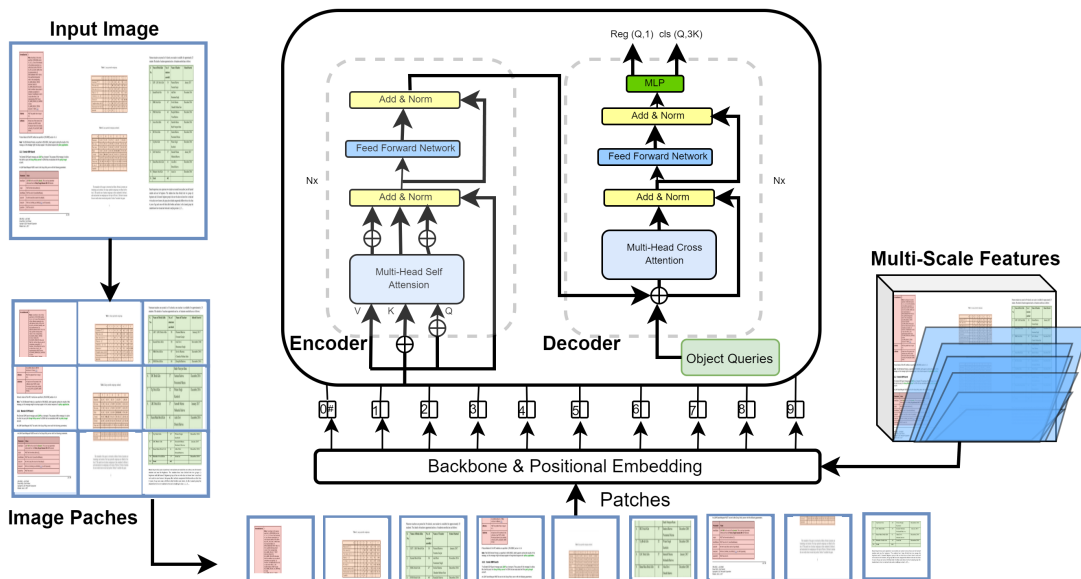


Fig. 1: Illustration of the deformable transformer employed in semi-supervised table detection method. We split the input image into small equal-sized patches, add position embeddings, and feed the resulting patches along with input multi-scale features to the transformer encoder. In the decoder, We use object queries as reference points and provide bounding boxes predictions and class labels as the final output.

Transformer Encoder. The CNN backbone (ResNet-50) extracts the input feature maps $f_m \in \mathbb{R}^{h_i \times w_i \times c_i}$. The spatial dimensional feature maps are converted into one-dimensional $z_m \in \mathbb{R}^{h_i \times w_i \times d_1}$ feature maps as the transformer encoder network takes input as a sequence. This one-dimensional vector is fed as input along with positional embeddings [85,86] to the transformer encoder network, which further transforms them into features for object queries. Every layer of the encoder module contains an attention network and a feed-forward network (FFN) where query and key values are the pixels of feature maps. Readers can refer to [87] for a detailed explanation of transformer.

Transformer Decoder. The decoder network takes the output of the encoder features and N number of object queries as input. It contains two attention types self-attention and cross-attention. The self-attention module finds

the connection between object queries. Here both key and query matrices contain object queries. The cross-attention module extracts feature using object queries from the input feature map. Here key matrix contains the feature maps provided by the encoder module, and the query matrix is the object queries fed as input to the decoder. After the attention modules, feed-forward networks (FFN) and linear projection layers are added as the prediction head. The linear projection layer predicts class labels, while FFN provides final bounding-box coordinate values.

Deformable Attention Module. The attention module in the DETR network considers all spatial locations of the input feature map, which makes the training convergence slower. However, a deformable DETR can solve this issue using the deformable convolution-based [73,88] attention network and multiscale input features [89,90]. It considers only a few sample pixels near a reference pixel, whatever the size of input features, as illustrated in Figure 2. The query matrix takes only a small set of keys, which resolves the slow training convergence issue of DETR. Readers can refer to [25] for a detailed explanation of Deformable DETR.

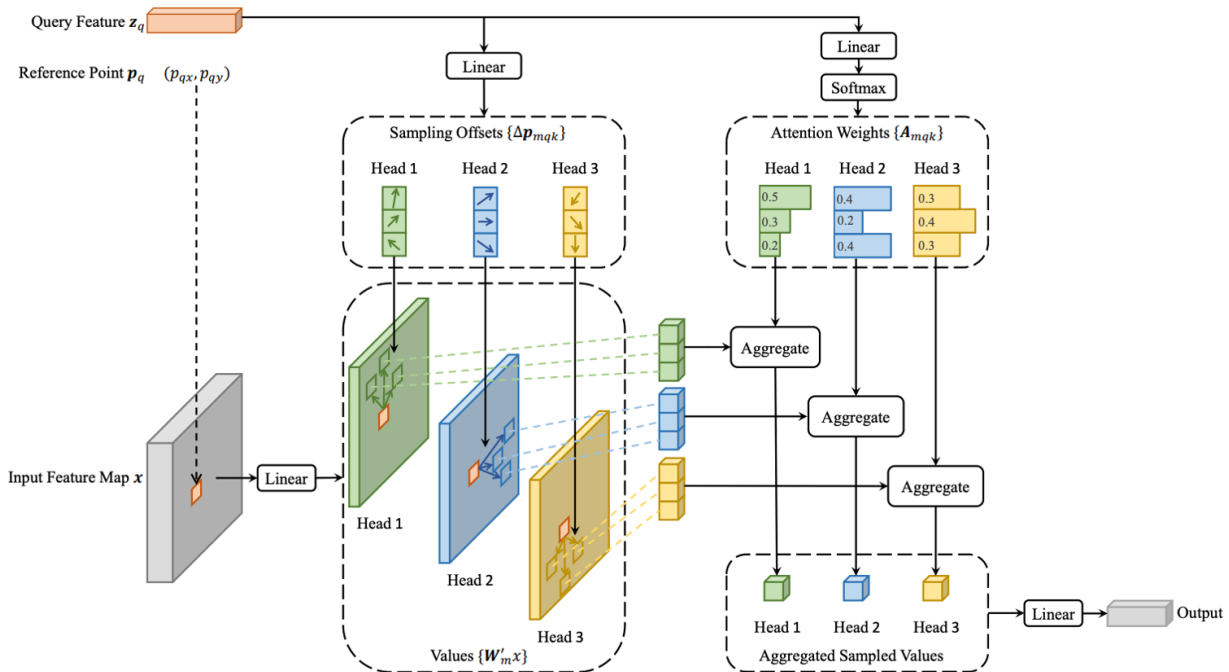


Fig. 2: Deformable Attention network. It considers only a few sample pixels near a reference pixel, whatever the size of input features. The query matrix takes only a small set of keys, which resolves the slow training convergence issue of DETR. (image from [25]).

3.2 Semi-Supervised Deformable DETR

In this subsection, we describe the learning mechanism of our proposed semi-supervised approach that employs the Deformable transformer and then explain the pseudo-labeling strategy. Semi-supervised Deformable-DETR is a unified learning approach that uses fully labeled and unlabeled data for object detection. It contains two modules a student module and a teacher module. The training data has two data types label data and unlabeled data. The student module takes both labeled and unlabeled images as input where strong augmentation is applied on unlabeled data while both (strong and weak augmentation) is applied on label data. The student module is trained using detection losses of labeled and unlabeled data through pseudo-boxes. The unlabeled data contains two groups of pseudo boxes for providing class labels and their bounding boxes. The teacher module only takes unlabeled images as input after applying weak augmentation. Figure 3 presents a summary of proposed pipeline. The teacher module feeds prediction

results to the pseudo-labeling framework to get pseudo-labels. Then, the student module uses these pseudo-labels for supervised training. Here, weak augmentation on unlabeled data is used for the teacher module to generate more precise pseudo-labels. Strong augmentation on unlabeled data is used for the student module to have more challenging learning. The student module also takes a small percentage of labeled images with strong and weak augmentation as input. The student module s_m is optimized with the total loss as follows:

$$\mathcal{L}^{s_m} = \sum_n \mathcal{L}(x_j^{l,s_a}, y_j^{l,s_a}) + \mathcal{L}(x_j^{l,w_a}, y_j^{l,w_a}) + \sum_n \mathcal{L}(x_j^{u,s_a}, y_j^{t_m}) \quad (1)$$

Where s_a represents strong augmentation, w_a represents weak augmentation. x_j^{l,s_a} is the strong augmented input image and its corresponding label is y_j^{l,s_a} . The term x_j^{l,w_a} is the weak augmented input image and its corresponding label is y_j^{l,w_a} . For the labeled images, strong and weak augmentations are also applied for learning, and are fed to the student module. The term x_j^{u,s_a} represents unlabeled strong augmented image fed to student module and the term $y_j^{t_m}$ is the pseudo-label from teacher module. Here, \mathcal{L} is the weighted sum of classification (class labels) and regression (bounding box) loss as follows:

$$\mathcal{L} = \alpha_1 \mathcal{L}^{reg} + \alpha_2 \mathcal{L}^{cls} \quad (2)$$

Where α_1 and α_2 are the weight values, the teacher-student modules are initialized randomly at the start of training. During training, the student module continuously updates the teacher module with an Exponential Moving-Average (EMA) [91] strategy. Pseudo-label generation for image classification tasks is easy, considering probability distribution as Pseudo-labels. In contrast, object detection tasks are more complicated as an image may include numerous objects, and annotation contains object location and class label. The CNN-based object detectors use anchors as object proposals and remove redundant boxes by post-processing steps such as non-maximal suppression (NMS). In contrast, transformers use attention mechanisms and object queries. Figure 4 shows sample points and attention weights from multi-scale deformable attention feature maps for both student and teacher networks. Its training complexity is $O(N_q c_i^2 + \min(h_i w_i c_i^2, N_q k c_i^2) + 5N_q k c_i + 3N_q c_i p_s k)$. This takes into account the computation of the sampling coordinate offsets and attention weights, as well as the bilinear interpolation and weighted sum in the attention mechanism. N_q is the number of query elements, c_i is the channel dimension, k is the kernel size, p_s is the number of sampling points, and $h_i w_i$ is the height and width of the feature map. In our experiments, $p_s = 8$, $k \leq 4$ and $c_i = 256$ by default, thus $5k + 3p_s k < c_i$ and the complexity is of $O(2N_q c_i^2 + \min(h_i w_i c_i^2, N_q k c_i^2))$. When used in the DETR encoder with $N_q = h_i w_i$, the complexity of the deformable attention module is $O(h_i w_i c_i^2)$, which scales linearly with the spatial size. When used in the DETR decoder with $N_q = N$ (the number of object queries), the complexity becomes $O(N k c_i^2)$, which is independent of the spatial size as attention is focused on the object queries.

Training The semi-supervised network is trained in two steps: a) train the student module independently on labeled data and generate pseudo-labels by teacher module; b) combine training of both modules to provide final prediction results.

Pseudo-Labeling Framework We used a simple framework to provide pseudo-labels for unlabeled data at the output of the teacher module, as applied in SSOD [92]. Usually, object detectors give confidence score vector $s_k \in [0, 1]^{C_i}$ for every provided bounding box b_k . A simple approach to provide pseudo-labels is to just thresholding these scores. In a simple pseudo-labeling filter, pseudo-labels can be formed by providing a threshold to the confidence value $s_k^{c_k}$ of the ground-truth class c_k . If the prediction value is not greater than the confidence value for a ground-truth class, the highest prediction value is considered the pseudo-label. Inspired by DETR [84], we develop the pseudo-label assignment task as a bipartite matching task between the teacher module predictions and the generated semi-labels. Specifically, the permutation of K elements is as follows:

$$\hat{\sigma} = \arg \min_{\sigma \in N} \sum_k^{N_i} \mathcal{L}_{match}(y_k, \hat{y}(k)), \quad (3)$$

Where $\mathcal{L}_{match}(y_k, \hat{y}(k))$ is the match-cost between teacher labels and ground-truth semi-labels as follows:

$$\mathcal{L}_{match}(y_k, \hat{y}(k)) = -\mathbb{1}_{\{c_k \neq \phi\}} \hat{p}_{\sigma(k)}(c_k) + \mathbb{1}_{\{c_k \neq \phi\}} \mathcal{L}_{bbox}(b_k, \hat{b}_{\hat{\sigma}}(k)) \quad (4)$$

The Pseudo-Labeling framework is applied to the predictions of teacher module $\hat{y}(k)$ where $\hat{y}(k) = \{\hat{y}^{class}, \hat{y}^{bbox}\}$ is the prediction, with \hat{y}^{class} and \hat{y}^{bbox} represent the class and box values, respectively. Here, $\hat{y}^{cls} = [v_1, \dots, v_N]^T \in \mathbb{R}^{N \times C_i}$ and $\hat{y}^{bbox} = [\hat{b}_1, \dots, \hat{b}_N]^T \in \mathbb{R}^{N \times 4}$, where v_N is the output vector (before the softmax), \hat{b}_N the related bounding-box prediction, and N is the object queries provided as input to the transformer decoder. y_k represents pseudo-labels

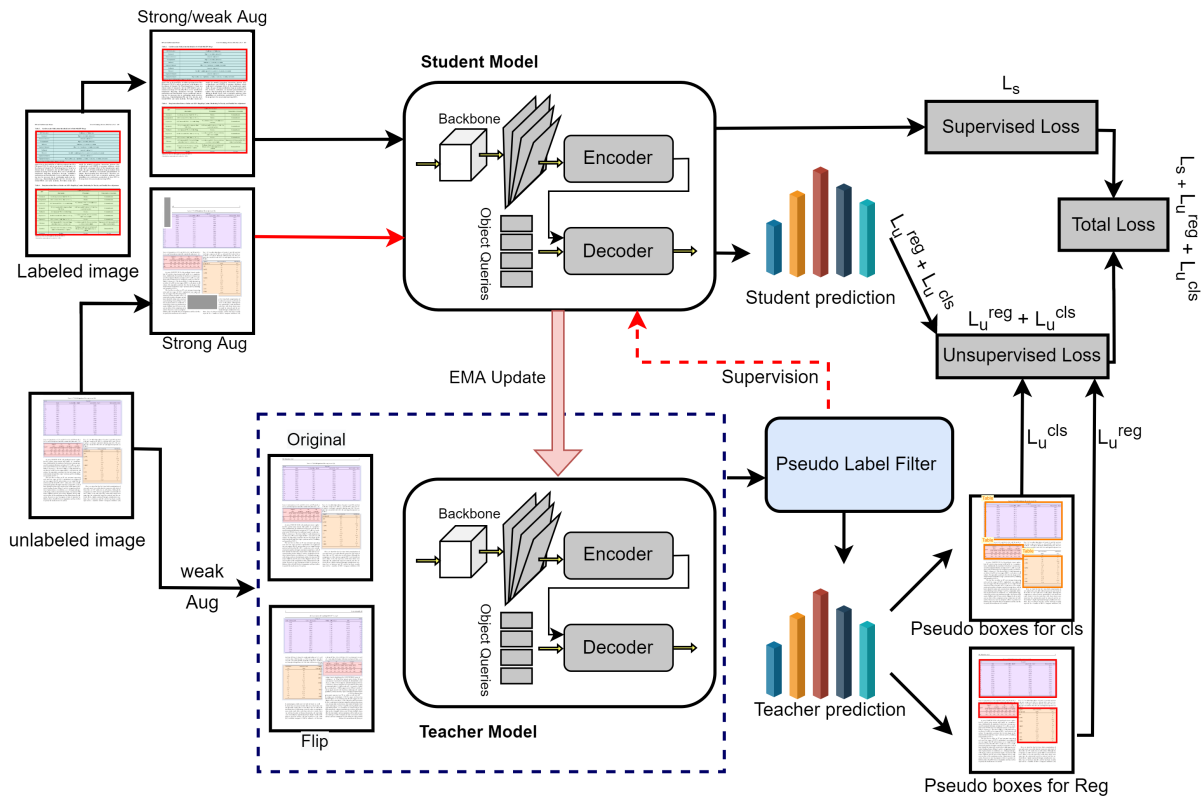


Fig. 3: Our proposed semi-supervised approach that employs Deformable transformer [25]. (1) The training data has two data types label data and unlabeled data. (2) It contains two modules a student module and a teacher module. (3) The teacher module only takes unlabeled images as input after applying weak augmentation. (4) After applying strong augmentation on unlabeled data type, the student module takes both labeled and unlabeled images as input. (5) During training, the student module continuously updates the teacher module with an Exponential Moving-Average (EMA) [91] strategy.

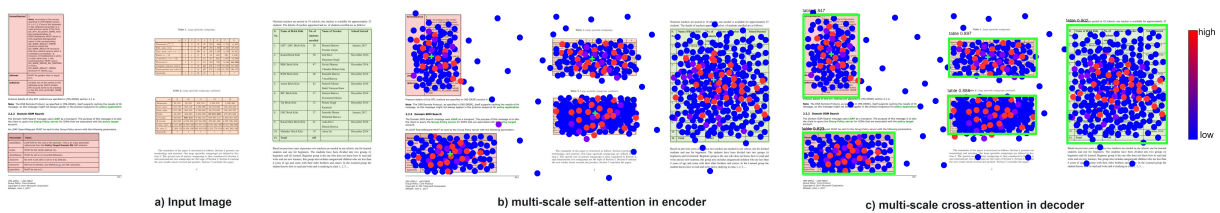


Fig. 4: Visualization of the sample points and attention weights from multi-scale deformable attention feature maps. Each sample point is denoted as a circle whose color represents its relative attention weight value. The reference points are the object queries taken as input in the encoder, represented by the green plus sign. In the decoder, the final bounding boxes are represented as green rectangles, and the class label and its confidence value are shown on the upper side in black text.

generated from confidence-score. The optimal selection is allowed with the Hungarian match mechanism [84,93], giving pseudo-labels $\{(b_k, c_k)\}$. This approach to select matching between the teacher module’s prediction and semi-labels generated by providing threshold works in the same way as the heuristic selection rules used for matching proposals [11] or anchors [89] with ground-truth objects in CNN-based object detectors. The main difference is that it determines one-to-one matching without duplicates. The second stage calculates the loss function, the Hungarian loss for all pair matching in the last stage. We define the loss similar to the previous object detector’s losses as a linear combination of a negative log-likelihood for class label and a bounding box as follows:

$$\mathcal{L}_H(y, \hat{y}) = \sum_{i=1}^N [-\log \hat{p}_{\hat{\sigma}(k)}(c_k) + \mathbf{1}_{\{c_k \neq \phi\}} \mathcal{L}_{box}(b_k, \hat{b}_{\hat{\sigma}(k)})] \quad (5)$$

Here, $b_k \in \mathbb{R}^4$ is the pseudo-bounding box, and c_k is the pseudo-class label. $\hat{\sigma}$ is the matching determined in the previous stage. In training, we reduce the weight of log probability by ten times when c_k for class imbalance. This mechanism is similar to the Faster R-CNN training strategy to balance proposals by sub-sampling [11].

4 Experimental Setup

4.1 Datasets

TableBank: TableBank [52] is the second-largest dataset in the document analysis domain for the table recognition problem. The dataset has 417,000 document images annotated through the arXiv database crawling procedure. The dataset features tables from three categories of document images: LaTeX images (253,817), Word images (163,417), and a combination of both (417,234). It also includes a dataset for recognizing the structures of the table. In our experiment, We only used the dataset for table detection from TableBank.

PubLayNet: PubLayNet [48] is a large public dataset with 335,703 images in the training set, 11,240 in the validation set, and 11,405 in the test set. It includes annotations such as polygonal segmentation and bounding boxes of figures, lists titles, tables, and text of images from research papers and articles. The dataset was evaluated using the coco analytic technique [94]. In our experiment, we only used 102,514 of the 86,460 table annotations.

DocBank: DocBank [95] is a large dataset of over 5,000 annotated document images from various sources designed to train and evaluate tasks such as text classification, entity recognition, and relation extraction. It includes annotations of title, author name, affiliation, abstract, body text, etc.

ICDAR-19: The competition for Table Detection and Recognition (cTDaR) [47] is organized at ICDAR in 2019. For the table detection task (TRACK A), two new datasets (modern and historical) are introduced in the competition. For direct comparison against the prior state-of-the-art [69], we provide results on the modern datasets with an IoU threshold ranging from 0.5–0.9.

4.2 Evaluation Criteria

We use some evaluation metrics to analyze the performance of our semi-supervised table detection approach that employs the deformable transformer. This section defines the employed evaluation metrics as precision, Recall, and F1-score. The Precision [96] is the fraction of actual instances as True Positives among the predicted instances as False Positives and True Positives). The Recall [96] is the fraction of actual instances as True Positives that were retrieved (True Positives + False Negatives). The F1-score [96] is the harmonic mean of Precision and Recall. We compute the intersection over union(IoU) by performing the intersection divided by the union for the region of the ground-truth box A_g and the formed bounding box A_p .

$$IoU = \frac{area(A_g \cap A_p)}{area(A_g \cup A_p)} \quad (6)$$

IoU estimates that either a detected table object is a false positive or a true positive. We find the average precision(AP) by a precision-recall (PR) curve following the context of MS COCO [94] evaluation. It is the area under the PR curve, calculated using the following equation:

$$AP = \sum_{k=1}^N (Re_{k+1} - Re_k) P_{intr}(Re_{k+1}) \quad (7)$$

Where Re_1, Re_2, \dots, Re_k represent the recall parameter. The mean average precision (mAP) is often used to evaluate the performance of detection methods. It is calculated by taking the mean of average precision for all classes in a dataset. The mAP can be affected by changes in the performance of individual classes due to class mapping, which is a limitation of this metric. We set the intersection over union (IoU) threshold values at 0.5 and 0.95. The mAP is calculated as follows:

$$mAP = \frac{1}{S} \sum_{s=1}^S AP_s \quad (8)$$

Where S represents total classes.

4.3 Implementation Details

We use the Deformable DETR [25] with a ResNet-50 [97] backbone pre-trained on the ImageNet [98] dataset as our detection framework for evaluating the usefulness of the semi-supervised approach. We perform training on PubLayNet, ICDAR-19, DocBank and all three splits of the TableBank dataset. We use 10%, 30% and 50% of labeled data and the rest as unlabeled data. The threshold value for pseudo-labeling is set at 0.7. We set the training epochs to 150 for all experiments with the learning rate reduced by a factor of 0.1 at the 120th epoch. We follow [92,25] to apply strong augmentation as horizontal flip, resize, remove patches, crop, grayscale and Gaussian blur. We use horizontal flipping to apply weak augmentation. The value N for the number of queries to the input of the decoder of Deformable DETR is set to 30 as it gives the best results. Unless otherwise specified, we evaluated the results using the mAP (AP50:95) metrics. All models are trained with a batch size of 16, using the same hyperparameters as Deformable DETR [25]. The weight α_1 is 2 and α_2 is 5 to balance the classification loss (L_{cls}) and regression loss (L_{box}). To make the training faster, we set the height and width of the input image to 600 pixels. We employ the standard size of 800 pixels for comparison with other approaches.

5 Results and Discussion

5.1 TableBank

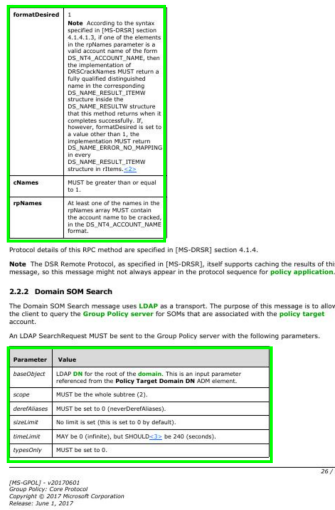
In this subsection, we provide the experimental results on all splits of the TableBank dataset on different percentages of label data. We also compare the transformer-based semi-supervised approach with previous deep learning-based supervised and semi-supervised approaches. Furthermore, we give results on 10% TableBank-both data split for all IoU threshold values. Table 1 provides the results of semi-supervised approach that employs deformable transformer for TableBank-latex, TableBank-word, and TableBank-both data splits on 10%, 30% and 50% label data and the rest as unlabeled data. It shows that the TableBank-both data split has the highest AP_{50} value of 95.8%, TableBank-word has 93.5%, and TableBank-both has 92.5% at 10% label data.

The qualitative analysis of semi-supervised learning for the TableBank-both data split is shown in Figure 5. Part (b) of Figure 5 has a matrix with a similar structure as rows and columns, and the network detects the matrix as a table giving false positive detection results. Here, incorrect detection results indicate where the network fails to provide correct detection of table regions. Table 2 gives the results of this semi-supervised approach on different IoU threshold values for all splits of the TableBank dataset on 10% label data and the rest as unlabeled data. A visual comparison of Precision, Recall and F1-Score of semi-supervised network that employs deformable transformer with ResNet-50 backbone on different IoU threshold values on 10% labeled dataset of TableBank-both data split is shown in Figure 6.

Comparisons with Previous supervised and semi-supervised approaches Table 3 compares the deep learning-based supervised and semi-supervised networks on the ResNet-50 backbone. We also compare supervised deformable DETR trained on 10%, 30% and 50% TableBank-both data split label data with our semi-supervised approach that employs deformable transformer. It shows that our attention mechanism-based semi-supervised approach provides comparable results without using proposal generation process and post-processing steps such as Non-maximal suppression (NMS).

Table 1: Performance of the semi-supervised approach that employs deformable transformer for TableBank-latex, TableBank-word, and TableBank-both data splits on different percentages of label data. Here, mAP represents mean AP at the IoU threshold range of (50:95), AP_{50} indicates AP at the IoU threshold of 0.5, and AP_{75} denotes AP at the IoU threshold of 0.75. AR_L indicates average recall for large objects.

Dataset	Label-percent	mAP	AP^{50}	AP^{75}	AR_L
TableBank-word	10%	80.5	92.5	87.7	87.1
	30%	88.3	95.7	93.1	92.1
	50%	91.5	96.7	95.2	94.5
TableBank-latex	10%	63.7	93.5	71.6	74.3
	30%	82.8	96.4	93.4	89.0
	50%	85.3	96.2	94.4	91.4
TableBank-both	10%	84.2	95.8	93.1	90.1
	30%	86.8	97.0	94.1	91.5
	50%	91.8	96.9	95.6	93.3



(a) True Positives

(b) True Positive and False Positive

(c) False Negative

Fig. 5: Semi-supervised table detection results that employs deformable transformer on TableBank-both data split. Green color represents true positives, blue denotes false negatives and red shows false positives. Here, (a) indicates true positive detection results, (b) shows true positive and false positive detection results, and (c) gives false negative detection results.

Table 2: The performance comparison of semi-supervised network that employs deformable transformer with ResNet-50 backbone on different IoU threshold values on 10% labeled dataset of TableBank-both data split.

Method	IoU	Precision	Recall	F1-score
Semi-Supervised Deformable-DETR+ResNet-50 10% labels	0.5	95.8	90.5	93.1
	0.6	94.6	90.5	92.5
	0.7	93.3	90.3	91.8
	0.8	91.8	89.8	90.8
	0.9	89.1	87.2	88.1

Interaction between the initial reaction rate (in n, P, K_0) and the final one (n^*, P^*, K_0^*) is written as:

$$R_{int}^{n,K_0} = \frac{R_{int}^{n,K_0}}{R_{int}^{n,K_0}} = \frac{R_{int}^{n,K_0}}{R_{int}^{n,K_0}} \quad (10)$$

Here,

$$\Xi = \frac{1}{2} \ln \left(\frac{R_{int}^{n,K_0}}{R_{int}^{n,K_0}} \right) = \frac{1}{2} \ln \left(\frac{R_{int}^{n,K_0}}{R_{int}^{n,K_0}} \right) = \frac{1}{2} \ln \left(\frac{R_{int}^{n,K_0}}{R_{int}^{n,K_0}} \right) \quad (11)$$

in which

$$A_{int} = (d_{int}^{n,K_0})_{int}^{n,K_0}, B_{int} = (d_{int}^{n,K_0})_{int}^{n,K_0} \quad (12)$$

From Eq. (10), for both the inter- and intra-valley exchange interactions, their initial only reaction elements between the bright reaction states, and hence both the inter- and intra-valley S-II exchange interaction can only occur the bright reaction transition. By considering the large splitting of the valence bands, the intra-valley depolarization channel due to the intra-valley S-II exchange interaction is nearly forbidden, and hence only the inter-valley S-II exchange interaction can contribute to the valley depolarization.

B. Valley depolarization due to the inter-valley $s-h$ exchange interaction

1. Model and KSSE

From Sec. II A, we conclude that only the inter-valley $s-h$ exchange interaction can cause the valley depolarization effect. For the A-section, instead, the $s-h$ exchange interaction includes the L and S-II part to the two-energy degenerate bright reaction rates, i.e., (L, P, K) and (L, P, K) . By referring to [12], (L, P, K) is the reaction "spin" space, with the effective magnetic field reading

$$B(P) = (-\text{Cos}(1) \frac{P^2 - P_0^2}{2P_0} + 2 \text{Cos}(1) \frac{P_0^2 - P_0^2}{2P_0}) \quad (13)$$

Obviously, the L-R (S-I) part of the exchange interaction acts as an in-plane P-dependent crystal magnetic field. With the effective magnetic field, the inter-valley A-section dynamics can be described by the KSSE as:

$$i \partial_t \langle P, \sigma | \rho | P, \sigma \rangle = \Delta_{int} \langle P, \sigma | \rho | P, \sigma \rangle \quad (14)$$

In these equations, $\langle P, \sigma | \rho | P, \sigma \rangle$ represent the 2×2 density matrices of A-section with center-of-mass momentum P at time t , in which the diagonal elements $\langle P, \sigma | \rho | P, \sigma \rangle$ describe the A-section distribution functions and the off-diagonal elements $\langle P, \sigma | \rho | P, \sigma' \rangle$ represent the "spin" coherence. In the coherent space, the relevant term is given by

$$\Delta_{int} \langle P, \sigma | \rho | P, \sigma' \rangle = -\frac{1}{2} \langle \sigma | \sigma' \rangle \Delta_{int} \langle P, \sigma | \rho | P, \sigma' \rangle \quad (15)$$

where $|\sigma\rangle, |\sigma'\rangle$ denotes the momentum. The scattering term $\Delta_{int} \langle P, \sigma | \rho | P, \sigma' \rangle$ include the inter-coupling scattering, intra-coupling scattering and valley depolarization scattering. Here, for simplicity, we only include the intra-coupling scattering²⁰ which is written as:

$$\Delta_{int} \langle P, \sigma | \rho | P, \sigma' \rangle = \sum_{P'} W_{P, P'} \langle P' | \rho | P' \rangle - \langle P | \rho | P \rangle \quad (16)$$

Here, $W_{P, P'}$ represents the momentum scattering rate. By using the KSSE, one obtains the evolution of the bright population $P(t) = \sum_{\sigma} \langle P, \sigma | \rho | P, \sigma \rangle$, with $\dot{P} = \Delta_{int} \langle P, \sigma | \rho | P, \sigma \rangle$ being the theory of the A-section. According to the pump-probe experiment^{21,22}, the initial condition is set to be

$$\rho_{int}(P, 0) = \alpha_{int} \exp \left[-\frac{1}{2} (P - P_{center})^2 \right] \delta(P) \quad (17)$$

and $\rho_{int}(P, 0) = 0$. Here, $\langle P | \rho | P \rangle (2\pi)^{-1}$ is the steady-state energy with δ being the reaction effective mass, P_{center} is the energy of pulse center in reference to the band minimum and $\alpha_{int} = A_{int} / A_{int}$, with A_{int} denoting the pulse width:

$$\alpha_{int} = \frac{1}{\sum_{\sigma} \int_{-\infty}^{\infty} \left[\langle P | \rho | P \rangle + \langle P | \rho | P \rangle \right] (2\pi)^{-1}} \quad (18)$$

with $\langle P | \rho | P \rangle$ being the density of A-section with "spin" σ after excitation. In the PL experiment of the pump-probe experiment, according to the classical optical valley selection rule, we set $\alpha_{int, \sigma} = \alpha_{int}$ and $\alpha_{int, \sigma} = 0$.

B. Results

In this part, we look into the current valley polarization experiments in monolayer MoS₂ with a-coupled parabolic, the inter-valley valley polarization measurement²³ of the pump-probe experiment^{22,24} and the steady-state PL polarization measurement^{25,26}. These theoretical calculations are summarized below based on the KSSE Eq. (14). The material parameters in our computation are listed in Table II.

TABLE II: Material parameters used in the computation.

Parameter	Value
$\hbar v_F$	3.47 eV
$\hbar v_{ph}$	0.27 eV
$\hbar v_{ph}^*$	0.07 eV
$\hbar v_{ph}^*$	0.07 eV
$\hbar v_{ph}^*$	0.07 eV
$\hbar v_{ph}^*$	0.07 eV

manufacturing plants in Jakarta have been relocated but most of them were just relocated to Bogor, Depok, Tangerang, and Bekasi in peripheral areas.

TABLE II: Material parameters used in the computation.

Industrial origin	2010	2011	2012
agriculture	9.88	9.64	9.78
mining and quarrying	1.47	1.16	1.15
electricity, gas and water	3.51	3.48	3.15
construction	4.33	4.76	4.77
trade, hotel and restaurants	7.68	8.22	7.28
transportation and communication	7.27	7.69	7.49
education, health and social services	14.73	15.37	15.86
information and communication	4.28	7.28	7.21
services	6.38	7.02	7.38
other activities regional product	6.28	10.09	4.73
other activities regional product	6.42	10.17	6.28

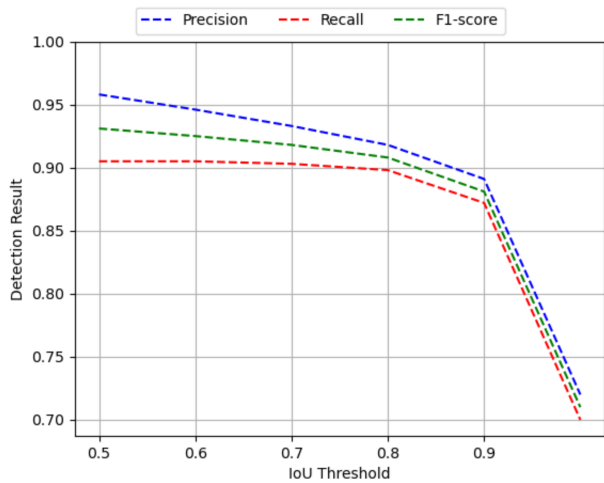
Source: BPS Jakarta Province Central Bureau of Statistics, 2012.

The rapid growth is due to the increased revenue of the country as a result of the successful implementation of development programs in various fields, particularly the manufacturing sector in the form of large-scale manufacturing and export-oriented, the tourism industry and export crops. Growth in the services sector, trade and non-manufacturing industries has also increased dramatically following the growth of the industrial base. Migration to large cities and production centers, it is unavoidable, it has also increased to meet the demand for labor supply.

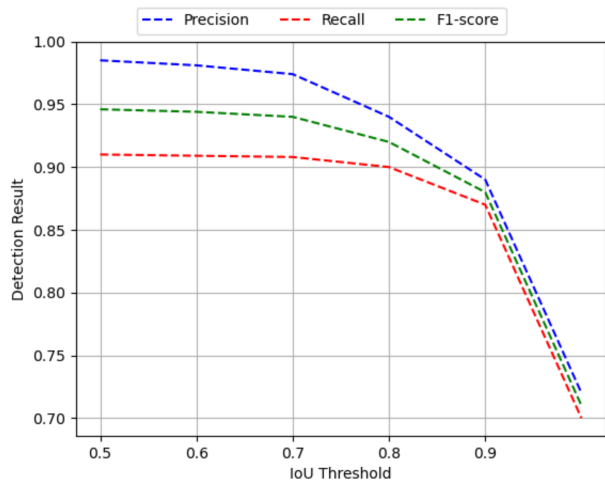
This urban economic structure provides a logical consequence of the increasing demand for the development of physical infrastructure, facilities which in turn has led to increased demand for a new model of an integrated public transportation system. We argue, developing integrated public transportation as part of the ideology. It is inevitable that the developments will carried out, in addition to further spur further growth in the country's economy and increase the employment. It has also put pressure significantly on transportation infrastructure and facilities which in most cases, not yet designed properly or to serve and accommodate the burden of the additional traffic generated by the presence of previous implicit congestion.

Table 3: Performance comparison of previous supervised and semi-supervised approaches. Supervised Deformable DETR and Faster R-CNN network trained on just 10%, 30% and 50% data of TableBank-both dataset while semi-supervised networks used 10%, 30% and 50% TableBank-both dataset as labeled and rest as unlabeled data using ResNet-50 backbone. Here, all results are represented on $mAP(0.5 : 0.95)$. The best threshold values are shown in bold.

Method	Approach	Detector	10%	30%	50%
Ren et al. [11]	supervised	Faster R-CNN	80.1	80.6	83.3
Zhu et al. [25]	supervised	Deformable DETR	80.8	82.6	86.9
STAC [82]	semi-supervised	Faster R-CNN	82.4	83.8	87.1
Unbiased Teacher [92]	semi-supervised	Faster R-CNN	83.9	86.4	88.5
Humble Teacher [99]	semi-supervised	Faster R-CNN	83.4	86.2	87.9
Soft Teacher [100]	semi-supervised	Faster R-CNN	83.6	86.8	89.6
Our	semi-supervised	Deformable DETR	84.2	86.8	91.8



(a) TableBank-both



(a) PubLayNet-table

Fig. 6: A visual comparison of Precision, Recall and F1-Score of semi-supervised network that employs deformable transformer with ResNet-50 backbone on different IoU threshold values on 10% labeled dataset of TableBank-both data split and PubLayNet table class dataset. Here, blue indicates precision results on different IoU threshold values, red shows recall results on different IoU threshold values, and green represents F1-score results on different IoU threshold values.

5.2 PubLayNet

In this subsection, we discuss the experimental results on PubLayNet table class dataset on different percentages of label data. We also compare the transformer-based semi-supervised approach with previous deep learning-based supervised and semi-supervised approaches. Furthermore, we give results on 10% PubLayNet dataset for all IoU threshold values. Table 4 provides the results of the semi-supervised approach that employs deformable transformer for PubLayNet table class on the different percentages of label data and rest as unlabeled data. Here, AP_{50} value is 98.5%, 98.8%, and 98.8% for 10%, 30% and 50% label data, respectively.

Table 4: Performance results for PubLayNet table class dataset. Here, mAP represents mean AP at the IoU threshold range of (50:95), AP_{50} indicates AP at the IoU threshold of 0.5 and AP_{75} denotes AP at the IoU threshold of 0.75. AR_L indicates average recall for large objects.

Dataset	Label-percent	mAP	AP^{50}	AP^{75}	AR_L
PubLayNet	10%	88.4	98.5	97.3	91.0
	30%	90.3	98.8	97.5	93.2
	50%	92.8	98.8	97.3	96.0

Table 5: The performance comparison of semi-supervised network that employs deformable transformer with ResNet-50 backbone on different IoU threshold values on 10% PubLayNet labeled Dataset.

Method	IoU	Precision	Recall	F1-score
Semi-Supervised	0.5	98.5	91.0	94.6
	0.6	98.1	90.9	94.4
Deformable-DETR 10% labels	0.7	97.4	90.8	94.0
	0.8	94.0	90.0	92.0
	0.9	89.0	87.0	88.0

Furthermore, our semi-supervised network is trained on different IoU threshold values on 10% of labeled PubLayNet Dataset. Table 5 gives the results of the semi-supervised approach on different IoU threshold values for PubLayNet table class on 10% label data and the rest as unlabeled data. A visual comparison of Precision, Recall and F1-score of the semi-supervised network that employs the deformable transformer network with ResNet-50 backbone on different IoU threshold values on 10% labeled dataset of PubLayNet table class is shown in Figure 6. Here, blue indicates precision results on different IoU threshold values on different IoU threshold values, red shows recall results, and green represents F1-score results on different IoU threshold values.

Table 6: Performance comparison of previous supervised and semi-supervised approaches. Deformable-DETR and Faster R-CNN trained on just 10%, 30% and 50% table data while semi-supervised networks used 10%, 30% and 50% PubLayNet dataset as labeled and rest as unlabeled data. Here, all results are represented on AP_{50} at the IoU threshold of 0.5. The best threshold values are shown in bold.

Method	Approach	Detector	10%	30%	50%
Ren et al. [11]	supervised	Faster R-CNN	93.6	95.6	95.9
Zhu et al. [25]	supervised	Deformable DETR	93.9	96.2	97.1
STAC [82]	semi-supervised	Faster R-CNN	95.8	96.9	97.8
Unbiased Teacher [92]	semi-supervised	Faster R-CNN	96.1	97.4	98.1
Humble Teacher [99]	semi-supervised	Faster R-CNN	96.7	97.9	98.0
Soft Teacher [100]	semi-supervised	Faster R-CNN	96.5	98.1	98.5
Our	semi-supervised	Deformable DETR	98.5	98.8	98.8

Comparisons with Previous supervised and semi-supervised approaches Table 6 compares the deep learning-based supervised and semi-supervised networks on PubLayNet table class using ResNet-50 backbone. We also compare

supervised deformable-DETR trained on 10%, 30% and 50% PubLayNet table class label data with our semi-supervised approach that employs the deformable transformer. It shows that our semi-supervised approach provides comparable results without using proposal and post-processing steps such as Non-maximal suppression (NMS).

5.3 DocBank:

In this subsection, we discuss the experimental results on DocBank dataset on different percentages of label data. We compare the transformer-based semi-supervised approach with previous CNN-based semi-supervised approach in Table 7.

Table 7: Performance comparison of previous semi-supervised approach and our Deformable-DETR based semi-supervised approach on DocBank dataset. Here, all results are represented on $mAP(0.5 : 0.95)$.

Method	Approach	Detector	10%	30%	50%
Soft Teacher [100]	semi-supervised	Faster R-CNN	72.3	74.4	81.5
Our	semi-supervised	Deformable DETR	82.5	84.9	87.1

Furthermore, we also compare our semi-supervised approach on different percentages of label data with previous table detection and document analysis approaches for different datasets TableBank, PubLayNet, and DocBank in Table 8. Although we cannot directly compare our semi-supervised approach with previous supervised document analysis approaches. However, we can observe that even with 50% label data, we achieve comparable results with previous supervise approaches.

Table 8: Performance comparison of previous supervised approaches for document analysis. Our semi-supervised network uses 10%, 30% and 50% label data and rest as unlabeled data. Here, all results are represented on $mAP(0.5 : 0.95)$.

Method	Approach	Labels	TableBank	PubLayNet	DocBank
CDeC-Net [70]	supervised	100%	96.5	97.8	-
CasTabDetectoRS [32]	supervised	100%	95.3	-	-
Faster R-CNN [48]	supervised	100%	-	90	86.3
VSR [101]	supervised	100%	-	95.69	87.6
Our	semi-supervised	10%	84.2	88.4	82.5
Our	semi-supervised	30%	86.8	90.3	84.9
Our	semi-supervised	50%	91.8	92.8	87.1

5.4 ICDAR-19

We also evaluate our method for table detection on the Modern Track A portion of the table detection dataset from the cTDaR competition at ICDAR 2019. We summarize the quantitative results of our approach at different percentages of label data and compare it with previously supervised table detection approaches in Table 9. We evaluate results at higher IoU thresholds of 0.8 and 0.9. For a direct comparison with previous table detection approaches, we also evaluate our approach on 100% label data. Our approach achieved a precision of 92.6% and a recall of 91.3% on the IoU threshold of 0.9 on 100% label data.

Table 9: Performance comparison between the proposed semi-supervised approach and previous state-of-the-art results on the dataset of ICDAR 19 Track A (Modern).

Method	Approach	IoU=0.8			IoU=0.9		
		Recall	Precision	F1-Score	Recall	Precision	F1-Score
TableRadar [47]	supervised	94.0	95.0	94.5	89.0	90.0	89.5
NLPR-PAL [47]	supervised	93.0	93.0	93.0	86.0	86.0	86.0
Lenovo Ocean [47]	supervised	86.0	88.0	87.0	81.0	82.0	81.5
CascadeTabNet [69]	supervised	-	-	92.5	-	-	90.1
CDeC-Net [70]	supervised	93.4	95.3	94.4	90.4	92.2	91.3
HybridTabNet [33]	supervised	93.3	92.0	92.8	90.5	89.5	90.2
Our	semi-supervised (50%)	71.1	82.3	76.3	66.3	76.8	71.2
Our	supervised (100%)	92.1	94.9	93.5	91.3	92.6	91.9

5.5 Ablation Study

In this section, we validate the key design elements. Unless otherwise stated, all the ablation studies are conducted using a ResNet-50 backbone with 30% labeled images from the PubLayNet dataset.

Pseudo-Labeling confidence threshold In Section 3.2, the threshold value (referred to as the confidence threshold) plays an important role in determining the balance between the accuracy and quantity of the generated pseudo-labels. As this threshold value increases, fewer examples will pass the filter, but they will be of higher quality. Conversely, a smaller threshold value will result in more examples passing but with a higher likelihood of false positives. The impact of various threshold values, ranging from 0.5 to 0.9, is presented in Table 10. The optimal threshold value was determined to be 0.7 based on the results.

Table 10: Performance comparison using different Pseudo-labeling confidence threshold values. The best threshold values are shown in bold.

Threshold	AP	AP ⁵⁰	AP ⁷⁵
0.5	86.9	91.6	90.1
0.6	89.5	98.1	95.7
0.7	90.3	98.8	97.5
0.8	89.4	97.2	95.3
0.9	87.9	96.3	94.5

Table 11: Performance comparison using different numbers of learnable queries to the decoder input. Here, best performance results are shown in bold.

N	AP	AP ⁵⁰	AP ⁷⁵
3	61.4	69.7	62.6
30	90.3	98.8	97.5
50	89.4	90.3	85.4
100	88.4	89.7	83.9
300	78.5	94.7	90.2

Influence of Learnable queries Quantity In our analysis, we investigate the impact of varying the number of queries fed as input in the decoder of deformable DETR. Figure 7 compares prediction results by varying the number of object queries fed as input in the decoder of deformable DETR. The optimal performance is attained when the number of queries N is set to 30; deviating from this value results in a decrease in performance. Table 11 presents and analyzes the result for varying object query quantities. Choosing a small value for N could result in the model failing to identify particular objects, negatively impacting its performance. On the other hand, selecting a large value for N may cause the model to perform poorly due to overfitting, as it would incorrectly classify certain regions as objects. Moreover, training complexity $O(Nkc_i^2)$ of this semi-supervised self-attention mechanism in the decoder of student-

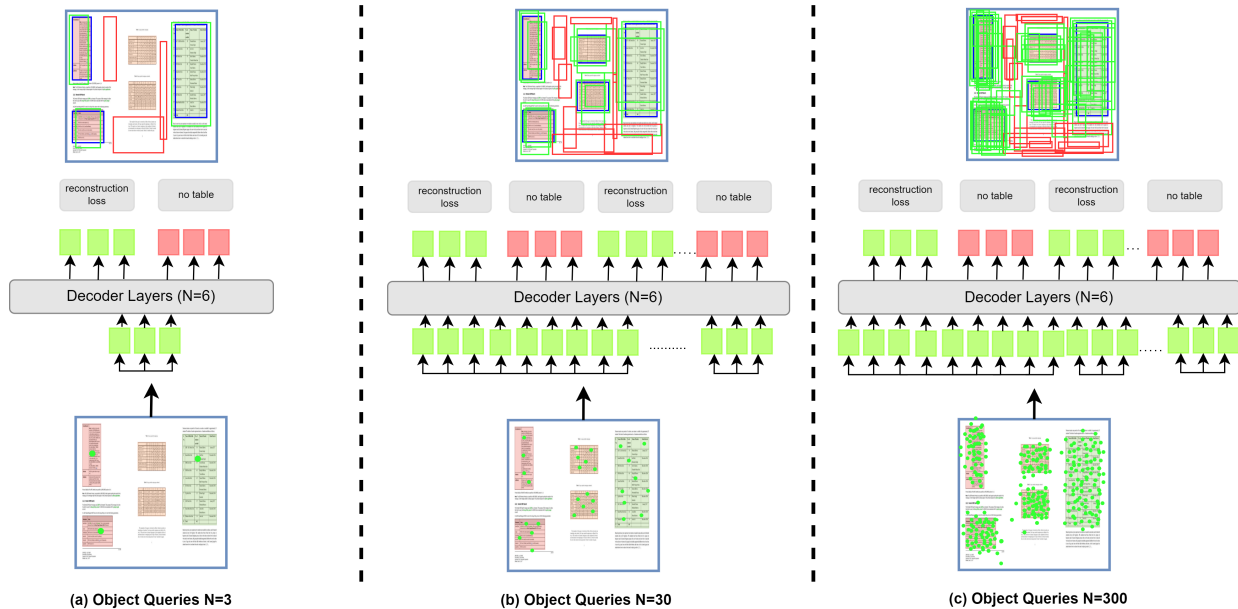


Fig. 7: Comparison of performance by variation of the number of object queries fed as input in the decoder of deformable DETR. Here, (a) takes $N=3$ object queries as input, (b) contains $N=30$ object queries as input, and (c) has $N=300$ object queries as input. The optimal performance is achieved by selecting the number of queries N to 30; deviating from this value results in a decrease in performance. Here, blue rectangles denote ground truth (GT), green rectangles indicate object class, and red rectangles show background class.

teacher module depends on the number of object queries and is subsequently improved as complexity is reduced by minimizing the number of object queries.

6 Conclusion

This paper introduces a semi-supervised approach that employs the deformable transformer for table detection in document images. The proposed method mitigates the need of large-scale annotated data and simplifies the process by integrating the pseudo-label generation framework into a streamlined mechanism. The simultaneous generation of pseudo-labels leads to a dynamic process known as the "flywheel effect", where one model continually improves the pseudo-boxes produced by the other model as the training progresses. The pseudo-class labels and pseudo-bounding boxes are improved in this framework using two distinct modules named student and teacher. These modules update each other by the EMA function to provide precise classification and bounding box predictions. The results indicate that this approach surpasses the performance of supervised models when applied to labeling ratios of 10%, 30%, and 50% on TableBank all splits and the PubLayNet training data. Furthermore, when trained on the 10% labeled data of PubLayNet, the model performed comparably to the current CNN-based semi-supervised baseline. In future, we aim to investigate the impact of the proportion of annotated data on the ultimate performance and develop models that function effectively with a minimal quantity of labeled data. Additionally, we intend to employ the transformer-based semi-supervised learning mechanism for table structure recognition task.

References

1. Z. Zhao, M. Jiang, S. Guo, Z. Wang, F. Chao, and K. C. Tan, "Improving deep learning based optical character recognition via neural architecture search," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–7.
2. D. Van Strien, K. Beelen, M. C. Ardanuy, K. Hosseini, B. McGillivray, and G. Colavizza, "Assessing the impact of ocr quality on downstream nlp tasks," 2020.

3. B. Coiasnon and A. Lemaitre, "Recognition of tables and forms," in *Handbook of Document Image Processing and Recognition*, 2014.
4. R. Zanibbi, D. Blostein, and J. R. Cordy, "A survey of table recognition," *Document Analysis and Recognition*, vol. 7, no. 1, pp. 1–16, 2004.
5. A. M. Jorge, L. Torgo *et al.*, "Design of an end-to-end method to extract information from tables," *International Journal of Document Analysis and Recognition (IJ DAR)*, vol. 8, no. 2, pp. 144–171, 2006.
6. J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage pdf documents via visual separators and tabular structures," in *2011 International Conference on Document Analysis and Recognition*, 2011, pp. 779–783.
7. J. Chen and D. Lopresti, "Table detection in noisy off-line handwritten documents," in *2011 International Conference on Document Analysis and Recognition*, 2011, pp. 399–403.
8. K. A. Hashmi, R. Bymana Ponnappa, S. S. Bukhari, M. Jenckel, and A. Dengel, "Feedback learning: Automating the process of correcting and completing the extracted information," in *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, vol. 5, 2019, pp. 116–121.
9. R. Saha, A. Mondal, and C. V. Jawahar, "Graphical object detection in document images," *CoRR*, vol. abs/2008.10843, 2020. [Online]. Available: <https://arxiv.org/abs/2008.10843>
10. R. B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015. [Online]. Available: <http://arxiv.org/abs/1504.08083>
11. S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: <http://arxiv.org/abs/1506.01497>
12. J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, vol. abs/1612.08242, 2016. [Online]. Available: <http://arxiv.org/abs/1612.08242>
13. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.
14. T. Orosz, R. Vági, G. M. Csányi, D. Nagy, I. Üveges, J. P. Vadász, and A. Megyeri, "Evaluating human versus machine learning performance in a legaltech problem," *Applied Sciences*, vol. 12, no. 1, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/1/297>
15. S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 1162–1167.
16. M. Minouei, K. A. Hashmi, M. R. Soheili, M. Z. Afzal, and D. Stricker, "Continual learning for table detection in document images," *Applied Sciences*, vol. 12, no. 18, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/18/8969>
17. K. A. Hashmi, D. Stricker, M. Liwicki, M. N. Afzal, and M. Z. Afzal, "Guided table structure recognition through anchor optimization," *CoRR*, vol. abs/2104.10538, 2021. [Online]. Available: <https://arxiv.org/abs/2104.10538>
18. K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Cascade network with deformable composite backbone for formula detection in scanned document images," *Applied Sciences*, vol. 11, no. 16, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/16/7610>
19. S. Sinha, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Rethinking learnable proposals for graphical object detection in scanned document images," *Applied Sciences*, vol. 12, no. 20, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/20/10578>
20. S. Naik, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Investigating attention mechanism for page object detection in document images," *Applied Sciences*, vol. 12, no. 15, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/15/7486>
21. K. Wang, X. Yan, D. Zhang, L. Zhang, and L. Lin, "Towards human-machine cooperation: Self-supervised sample mining for object detection," *CoRR*, vol. abs/1803.09867, 2018. [Online]. Available: <http://arxiv.org/abs/1803.09867>
22. P. Tang, C. Ramaiah, R. Xu, and C. Xiong, "Proposal learning for semi-supervised object detection," *CoRR*, vol. abs/2001.05086, 2020. [Online]. Available: <https://arxiv.org/abs/2001.05086>
23. P. K. Rhee, E. Erdenee, S. D. Kyun, M. U. Ahmed, and S. Jin, "Active and semi-supervised learning for object detection with imperfect data," *Cognitive Systems Research*, vol. 45, pp. 109–123, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389041716301127>
24. Q. Xie, Z. Dai, E. H. Hovy, M. Luong, and Q. V. Le, "Unsupervised data augmentation," *CoRR*, vol. abs/1904.12848, 2019. [Online]. Available: <http://arxiv.org/abs/1904.12848>
25. X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: deformable transformers for end-to-end object detection," *CoRR*, vol. abs/2010.04159, 2020. [Online]. Available: <https://arxiv.org/abs/2010.04159>
26. K. Itonori, "Table structure recognition based on textblock arrangement and ruled line position," in *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, 1993, pp. 765–768.
27. S. Tupaj, Z. Shi, C. H. Chang, and H. Alam, "Extracting tabular information from text files," *EECS Department, Tufts University, Medford, USA*, vol. 1, 1996.

28. S. Chandran and R. Kasturi, "Structural recognition of tabulated data," in *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR '93)*, 1993, pp. 516–519.
29. Y. Hirayama, "A method for table structure analysis using dp matching," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 2, 1995, pp. 583–586 vol.2.
30. T. G. Kieninger, "Table structure recognition based on robust block segmentation," in *Document Recognition V*, D. P. Lopresti and J. Zhou, Eds., vol. 3305, International Society for Optics and Photonics. SPIE, 1998, pp. 22 – 32. [Online]. Available: <https://doi.org/10.1117/12.304642>
31. S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, "Decnt: Deep deformable cnn for table detection," *IEEE Access*, vol. 6, pp. 74 151–74 161, 2018.
32. K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Castabdetectors: Cascade network for table detection in document images with recursive feature pyramid and switchable atrous convolution," *Journal of Imaging*, vol. 7, 2021.
33. D. Nazir, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Hybridtabnet: Towards better table detection in scanned document images," *Applied Sciences*, vol. 11, no. 18, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/18/8396>
34. P. Pyreddy and W. B. Croft, "Tintin: a system for retrieval in text tables," in *Digital library*, 1997.
35. A. Pivk, P. Cimiano, Y. Sure, M. Gams, V. Rajkovič, and R. Studer, "Transforming arbitrary tables into logical form with tartar," *Data & Knowledge Engineering*, vol. 60, no. 3, pp. 567–595, 2007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169023X06000620>
36. J. Hu, R. S. Kashi, D. P. Lopresti, and G. Wilfong, "Medium-independent table detection," in *Document Recognition and Retrieval VII*, D. P. Lopresti and J. Zhou, Eds., vol. 3967, International Society for Optics and Photonics. SPIE, 1999, pp. 291 – 302. [Online]. Available: <https://doi.org/10.1117/12.373506>
37. S. Khushro, A. Latif, and I. Ullah, "On methods and tools of table detection, extraction and annotation in pdf documents," *Journal of Information Science*, vol. 41, no. 1, pp. 41–57, 2015.
38. D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy, "Table-processing paradigms: a research survey," *International Journal of Document Analysis and Recognition (IJ DAR)*, vol. 8, no. 2, pp. 66–86, 2006.
39. F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," in *2002 International Conference on Pattern Recognition*, vol. 3, 2002, pp. 236–240 vol.3.
40. A. C. e. Silva, "Learning rich hidden markov models in document analysis: Table location," in *2009 10th International Conference on Document Analysis and Recognition*, 2009, pp. 843–847.
41. A. Silva, "Parts that add up to a whole: a framework for the analysis of tables," *Edinburgh University, UK*, 2010.
42. T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "Learning to detect tables in scanned document images using line information," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1185–1189.
43. X. Yang, M. E. Yümer, P. Asente, M. Kralej, D. Kifer, and C. L. Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural network," *CoRR*, vol. abs/1706.02337, 2017. [Online]. Available: <http://arxiv.org/abs/1706.02337>
44. D. He, S. Cohen, B. Price, D. Kifer, and C. L. Giles, "Multi-scale multi-task fcn for semantic page segmentation and table detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 254–261.
45. I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," *CoRR*, vol. abs/1804.06236, 2018. [Online]. Available: <http://arxiv.org/abs/1804.06236>
46. S. Paliwal, V. D. R. Rahul, M. Sharma, and L. Vig, "Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images," *CoRR*, vol. abs/2001.01469, 2020. [Online]. Available: <http://arxiv.org/abs/2001.01469>
47. L. Gao, Y. Huang, H. Déjean, J.-L. Meunier, Q. Yan, Y. Fang, F. Kleber, and E. Lang, "Icdar 2019 competition on table detection and recognition (ctdar)," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 1510–1515.
48. X. Zhong, J. Tang, and A. J. Yepes, "Publaynet: largest dataset ever for document layout analysis," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, Sep. 2019, pp. 1015–1022.
49. A. Mondal, P. Lipps, and C. V. Jawahar, "IIIT-AR-13K: A new dataset for graphical object detection in documents," *CoRR*, vol. abs/2008.02569, 2020. [Online]. Available: <https://arxiv.org/abs/2008.02569>
50. M. C. Göbel, T. Hassan, E. Oro, and G. Orsi, "Icdar 2013 table competition," *2013 12th International Conference on Document Analysis and Recognition*, pp. 1449–1453, 2013.
51. L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "Icdar2017 competition on page object detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 1417–1422.
52. M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li, "Tablebank: A benchmark dataset for table detection and recognition," 2019.

53. B. Smock, R. Pesala, and R. Abraham, "PubTables-1M: Towards comprehensive table extraction from unstructured documents," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 4634–4642.
54. J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1411.4038, 2014. [Online]. Available: <http://arxiv.org/abs/1411.4038>
55. X.-H. Li, F. Yin, and C.-L. Liu, "Page object detection from pdf document images by deep structured prediction and supervised clustering," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 3627–3632.
56. M. Holecek, A. Hoskovec, P. Baudis, and P. Klinger, "Line-items and table understanding in structured documents," *CoRR*, vol. abs/1904.12577, 2019. [Online]. Available: <http://arxiv.org/abs/1904.12577>
57. P. Riba, L. Goldmann, O. R. Terrades, D. Rusticus, A. Fornés, and J. Lladós, "Table detection in business document images by message passing networks," *Pattern Recognition*, vol. 127, p. 108641, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320322001224>
58. L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for pdf documents based on convolutional neural networks," *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pp. 287–292, 2016.
59. X. Yi, L. Gao, Y. Liao, X. Zhang, R. Liu, and Z. Jiang, "Cnn based page object detection in document images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 230–235.
60. T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: <http://arxiv.org/abs/1708.02002>
61. Y. Fang, B. Liao, X. Wang, J. Fang, J. Qi, R. Wu, J. Niu, and W. Liu, "You only look at one sequence: Rethinking transformer in vision through object detection," *CoRR*, vol. abs/2106.00666, 2021. [Online]. Available: <https://arxiv.org/abs/2106.00666>
62. K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: <http://arxiv.org/abs/1703.06870>
63. Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," *CoRR*, vol. abs/1712.00726, 2017. [Online]. Available: <http://arxiv.org/abs/1712.00726>
64. N. D. Vo, K. Nguyen, T. V. Nguyen, and K. Nguyen, "Ensemble of deep object detectors for page object detection," in *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, ser. IMCOM '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: <https://doi.org/10.1145/3164541.3164644>
65. A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017, pp. 771–776.
66. Y. Huang, Q. Yan, Y. Li, Y. Chen, X. Wang, L. Gao, and Z. Tang, "A yolo-based table detection method," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 813–818.
67. X. Zheng, D. Burdick, L. Popa, and N. X. R. Wang, "Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context," *CoRR*, vol. abs/2005.00589, 2020. [Online]. Available: <https://arxiv.org/abs/2005.00589>
68. R. Saha, A. Mondal, and C. V. Jawahar, "Graphical object detection in document images," *CoRR*, vol. abs/2008.10843, 2020. [Online]. Available: <https://arxiv.org/abs/2008.10843>
69. D. Prasad, A. Gadpal, K. Kapadni, M. Visave, and K. Sultanpure, "Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents," *CoRR*, vol. abs/2004.12629, 2020. [Online]. Available: <https://arxiv.org/abs/2004.12629>
70. M. Agarwal, A. Mondal, and C. V. Jawahar, "Cdec-net: Composite deformable cascade network for table detection in document images," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 9491–9498.
71. S. Arif and F. Shafait, "Table detection in document images using foreground and background features," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, 2018, pp. 1–8.
72. S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, "Decnt: Deep deformable cnn for table detection," *IEEE Access*, vol. 6, pp. 74 151–74 161, 2018.
73. J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR*, vol. abs/1703.06211, 2017. [Online]. Available: <http://arxiv.org/abs/1703.06211>
74. Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, and H. Ling, "Cbnet: A novel composite backbone network architecture for object detection," *CoRR*, vol. abs/1909.03625, 2019. [Online]. Available: <http://arxiv.org/abs/1909.03625>
75. J. Jeong, S. Lee, J. Kim, and N. Kwak, "Consistency-based semi-supervised learning for object detection," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/file/d0f4dae80c3d0277922f8371d5827292-Paper.pdf>
76. P. Tang, C. Ramaiah, R. Xu, and C. Xiong, "Proposal learning for semi-supervised object detection," *CoRR*, vol. abs/2001.05086, 2020. [Online]. Available: <https://arxiv.org/abs/2001.05086>

77. I. Radosavovic, P. Dollár, R. B. Girshick, G. Gkioxari, and K. He, "Data distillation: Towards omni-supervised learning," *CoRR*, vol. abs/1712.04440, 2017. [Online]. Available: <http://arxiv.org/abs/1712.04440>
78. B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking pre-training and self-training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 3833–3845. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/27e9661e033a73a6ad8cefcde965c54d-Paper.pdf>
79. Y. Li, D. Huang, D. Qin, L. Wang, and B. Gong, "Improving object detection with selective self-supervised self-training," *CoRR*, vol. abs/2007.09162, 2020. [Online]. Available: <https://arxiv.org/abs/2007.09162>
80. T. Shehzadi, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Mask-aware semi-supervised object detection in floor plans," *Applied Sciences*, vol. 12, no. 19, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/19/9398>
81. G. Kallempudi, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "Toward semi-supervised graphical object detection in document images," *Future Internet*, vol. 14, no. 6, 2022. [Online]. Available: <https://www.mdpi.com/1999-5903/14/6/176>
82. K. Sohn, Z. Zhang, C. Li, H. Zhang, C. Lee, and T. Pfister, "A simple semi-supervised learning framework for object detection," *CoRR*, vol. abs/2005.04757, 2020. [Online]. Available: <https://arxiv.org/abs/2005.04757>
83. K. Wang, X. Yan, D. Zhang, L. Zhang, and L. Lin, "Towards human-machine cooperation: Self-supervised sample mining for object detection," *CoRR*, vol. abs/1803.09867, 2018. [Online]. Available: <http://arxiv.org/abs/1803.09867>
84. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*. Springer, 2020, pp. 213–229.
85. N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, and A. Ku, "Image transformer," *CoRR*, vol. abs/1802.05751, 2018. [Online]. Available: <http://arxiv.org/abs/1802.05751>
86. I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention augmented convolutional networks," *CoRR*, vol. abs/1904.09925, 2019. [Online]. Available: <http://arxiv.org/abs/1904.09925>
87. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>
88. X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convnets v2: More deformable, better results," *CoRR*, vol. abs/1811.11168, 2018. [Online]. Available: <http://arxiv.org/abs/1811.11168>
89. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.
90. Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," *CoRR*, vol. abs/1811.04533, 2018. [Online]. Available: <http://arxiv.org/abs/1811.04533>
91. A. Tarvainen and H. Valpola, "Weight-averaged consistency targets improve semi-supervised deep learning results," *CoRR*, vol. abs/1703.01780, 2017. [Online]. Available: <http://arxiv.org/abs/1703.01780>
92. Y. Liu, C. Ma, Z. He, C. Kuo, K. Chen, P. Zhang, B. Wu, Z. Kira, and P. Vajda, "Unbiased teacher for semi-supervised object detection," *CoRR*, vol. abs/2102.09480, 2021. [Online]. Available: <https://arxiv.org/abs/2102.09480>
93. H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
94. T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: <http://arxiv.org/abs/1405.0312>
95. M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, and M. Zhou, "Docbank: A benchmark dataset for document layout analysis," *CoRR*, vol. abs/2006.01038, 2020. [Online]. Available: <https://arxiv.org/abs/2006.01038>
96. D. M. W. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *CoRR*, vol. abs/2010.16061, 2020. [Online]. Available: <https://arxiv.org/abs/2010.16061>
97. C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016. [Online]. Available: <http://arxiv.org/abs/1602.07261>
98. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>
99. Y. Tang, W. Chen, Y. Luo, and Y. Zhang, "Humble teachers teach better students for semi-supervised object detection," *CoRR*, vol. abs/2106.10456, 2021. [Online]. Available: <https://arxiv.org/abs/2106.10456>
100. M. Xu, Z. Zhang, H. Hu, J. Wang, L. Wang, F. Wei, X. Bai, and Z. Liu, "End-to-end semi-supervised object detection with soft teacher," *CoRR*, vol. abs/2106.09018, 2021. [Online]. Available: <https://arxiv.org/abs/2106.09018>

101. P. Zhang, C. Li, L. Qiao, Z. Cheng, S. Pu, Y. Niu, and F. Wu, "VSR: A unified framework for document layout analysis combining vision, semantics and relations," *CoRR*, vol. abs/2105.06220, 2021. [Online]. Available: <https://arxiv.org/abs/2105.06220>