PyramidTabNet: Transformer-based Table Recognition in Image-based Documents

Muhammad Umer[1], Muhammad Ahmed Mohsin[1], Adnan Ul-Hasan[2], and Faisal Shafait[1,2]

[1] School of Electrical Engineering and Computer Science (SEECS),
National University of Sciences and Technology (NUST), Islamabad, Pakistan
[2] Deep Learning Laboratory, National Center of Artificial Intelligence (NCAI),
Islamabad, Pakistan
{mumer.bee20seecs, mmohsin.bee20seecs, adnan.ulhassan,
faisal.shafait}@seecs.edu.pk

Abstract. Table detection and structure recognition is an important component of document analysis systems. Deep learning-based transformer models have recently demonstrated significant success in various computer vision and document analysis tasks. In this paper, we introduce PyramidTabNet (PTN), a method that builds upon Convolutionless Pyramid Vision Transformer to detect tables in document images. Furthermore, we present a tabular image generative augmentation technique to effectively train the architecture. The proposed augmentation process consists of three steps, namely, clustering, fusion, and patching, for the generation of new document images containing tables. Our proposed pipeline demonstrates significant performance improvements for table detection on several standard datasets. Additionally, it achieves performance comparable to the state-of-the-art methods for structure recognition tasks.

Keywords: deep learning · image transformer · image processing · data augmentation · table augmentation · table segmentation · table detection · structure recognition

1 Introduction

Table detection and structure recognition are crucial tasks that have numerous applications in fields such as data extraction, document summarization, and information retrieval. Tables play a significant role in presenting data in a structured format, and they are frequently used in a variety of document types, such as research papers, reports, and financial statements. The automatic detection and recognition of tables and their structures is a complex task that requires the integration of several computer vision and pattern recognition techniques.

The most common approach to table detection is to use supervised machine learning techniques [2, 12, 14, 22, 24, 28, 31, 33] to learn to identify tables based on features extracted from the input images. These features may include visual features, such as the presence of lines and boxes, as well as layout features, such as the position and size of elements within the document. These supervised machine-learning approaches have shown to be effective at detecting tables, but they often rely on large high-quality datasets and can be sensitive to variations in the input images.

Once a table has been detected, the next step is often to recognize its structure and extract its contents. This can be a complex task, as tables can vary significantly in terms of their layout, formatting, and content. Some approaches to table structure recognition involve analyzing the visual layout of the table, such as the presence and positioning of lines, boxes, and other visual elements. Other approaches may involve analyzing the semantic structure of the table, such as the relationships between different cells and the meaning of their contents.

Detecting and recognizing tables in document images is challenging due to various obstacles. These include inconsistent table structures, poor image quality, complex backgrounds, data imbalance, and insufficient annotated data. Inconsistent table structures result from varying layouts and structures, making it difficult for models to identify tables accurately. Poor image quality, such as blurring, distortion, and low resolution, can also impede recognition as well.

Overall, deep learning techniques present a promising solution for the accurate detection and recognition of tables in document images. However, current techniques are faced with a major challenge, which is the bias arising from image variability. This bias is a result of the complex and diverse nature of document analysis, despite the use of large amounts of training data. As such, there is a need for further research to overcome this challenge and improve the accuracy of document analysis techniques.

In this work, we have developed an end-to-end approach for table recognition in scanned document images that leverages the performance of convolution-less Pyramid Vision Transformer [35]. We also propose and integrate a novel tabular image generative augmentation technique to ensure that different types of table structures are uniformly learned by the model to address the challenges posed by the variability in table appearance and complex backgrounds.

The results of our approach demonstrate that it outperforms many of the recent works in this field, such as HybridTabNet [22], CasTabDetectoRS [14], and Document Image Transformer [18], on a variety of benchmarks. This serves as a measure of the effectiveness of our approach, as well as the value of the data augmentation techniques that we have incorporated into our method.

The rest of the paper is organized as follows: Section 2 provides the literature review and current advancements in table recognition using both CNNs and transformers and an overview of novel data augmentation pipelines. Section 3 provides an explanation of the proposed architecture; each component of the model is briefed in-depth. Section 4 provides an overview of the utilized datasets, along with the data augmentation techniques employed. Section 5 provides a comparison of our architecture against the current state-of-the-art along with its analysis. Section 6 concludes this paper along with ideas for future work and further enhancements.

2 Related Work

Table understanding is an important aspect of document image analysis. Deep learning-based approaches have been increasingly exploited to improve the generalization capabilities of table detection systems. This section aims to provide a brief overview of some of these methods.

Among the initial deep learning approaches, Gilani et al. [12] proposed a technique for table detection in which, document images are first subjected to pre-processing before being fed into an RPN. This network is designed to identify regions in the image that are likely to contain tables and later detected using a CNN. Arif and Shafait [2] introduced a method to enhance table detection by utilizing foreground and background features. The technique takes advantage of the fact that most tables contain numeric data, and utilizes color coding to differentiate between textual and numeric information within the table.

Traditional CNNs have a fixed receptive field, making table recognition challenging when tables are present in varying sizes and orientations. Deformable convolution [6], on the other hand, adapts its receptive field to the input, enabling the network to accommodate tables of any layout through customization of the receptive field. Employing a unique combination of Deformable CNN and Faster R-CNN, Siddiqui et al. [31] presented a novel strategy for table detection in documents that leverages the ability to recognize tables with any arrangement.

Qasim et al. [25] proposed using graph neural networks for table recognition tasks, combining the benefits of convolutional neural networks and graph networks for dealing with the input structure. Khan et al. [16] propose a deep learning solution for table structure extraction in document images, using a bi-directional GRU to classify inputs. Khan et al. [17] presented TabAug, a novel table augmentation approach that involves modifying table structure by replicating or deleting rows and columns. TabAug showed improved efficiency compared to conventional methods and greater control over the augmentation process.

Prasad et al. [24] proposed CascadeTabNet, a table detection system built upon Cascade Mask R-CNN HRNet framework and enhanced by transfer learning and image manipulation techniques. Nazir et al. [22] presented HybridTabNet, a pipeline comprising two stages: the first stage extracts features using the ResNeXt-101 network, while the second stage uses a Hybrid Task Cascade (HTC) to localize tables within the document images.

Zheng et al. [36] proposed Global Table Extractor (GTE), a technique that detects tables and recognizes cell structures simultaneously, using any object detection model. GTE-Table, a new training method, is used to improve table detection by incorporating cell placement predictions. Raja et al. [27] presented a novel object-detection-based deep learning model that is designed for efficient optimization and accurately captures the natural alignment of cells within tables. The author proposed a unique rectilinear graph-based formulation to enhance structure recognition to capture long-range inter-table relationships.

Image transformers have garnered a lot of popularity in computer vision and image processing tasks and recently, transformer-based models have been employed for document analysis as well. Smock et al. [32] utilized DEtection

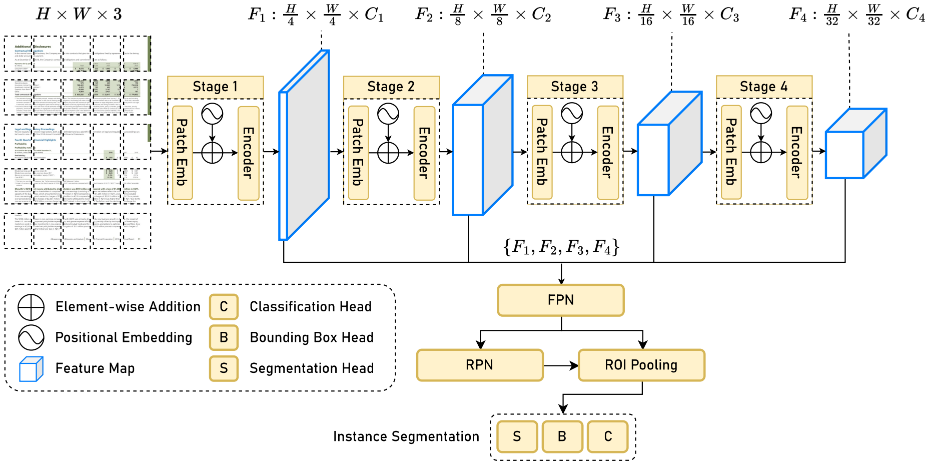


Fig. 1. Model architecture of PyramidTabNet – A convolution-less Pyramid Vision Transformer backbone is attached to a vanilla implementation of the Cascade Mask R-CNN framework to detect the instances and bounding boxes of document tables and their structural components.

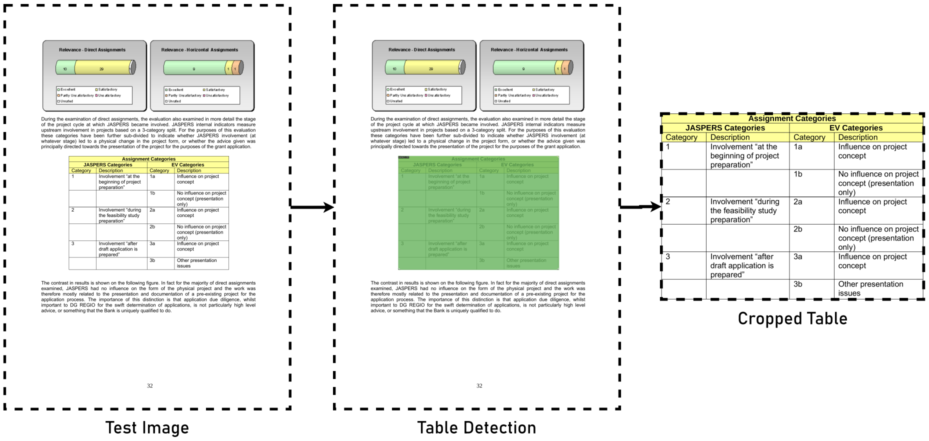
TRansformer (DETR) [4] framework for table detection and structure recognition tasks. Document Image Transformer [18] proposed by Xu et al. utilized large-scale unlabeled images for document analysis tasks and achieved state-of-the-art results in document classification as well as table recognition.

Leveraging the success of transformers in the field of document analysis, we integrate a convolution-less transformer backbone with a vanilla Cascade Mask R-CNN framework. The following section provides a comprehensive examination of the end-to-end pipeline for table detection and structure recognition, including a demonstration of how inputs are processed through the architecture.

3 PyramidTabNet: Methodology

3.1 Architecture

Building upon the superiority of Transformer models demonstrated by the Pyramid Vision Transformer (PVT) [34] in dense prediction tasks, PyramidTabNet utilizes the updated PVT v2 architecture [35] with a 3×3 depth-wise convolution in its feed-forward network. The document image is first divided into non-overlapping patches and transformed into a sequence of patch embeddings. These embeddings are then infused with positional information and processed through multiple stages of PVT to form the backbone of PyramidTabNet. The output of the PVT v2 stage is reshaped to feature maps F_1, F_2, F_3 and F_4 with strides of 4, 8, 16, and 32 respectively. Lastly, the feature pyramid $\{F_1, F_2, F_3, F_4\}$ is forwarded to a vanilla Cascade Mask R-CNN [3] framework to perform instance segmentation as shown in Figure 1.



Assignment Categories			
JASPERS Categories		EV Categories	
Category	Description	Category	Description
1	Involvement "at the beginning of project preparation"	1a	Influence on project concept
		1b	No influence on project concept (presentation only)
2	Involvement "during the feasibility study preparation"	2a	Influence on project concept
		2b	No influence on project concept (presentation only)
3	Involvement "after draft application is prepared"	3a	Influence on project concept
		3b	Other presentation issues

Fig. 2. Table detection pipeline – Instances of tables are detected on an input image and cropped to extract the tabular region.

The proposed end-to-end pipeline can be categorized into two phases: table detection and structure recognition. The details of each stage are discussed in the following sections, along with an exemplary forward pass of the input through the architecture.

Table Detection In a single feed-forward pass of the input image (a document) to the model, the table detection module detects all instances of tables in the input and performs bounding box regression. The detected bounding box coordinates, in $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ format, are then used to extract all the tables from the input, and intermediately saved. The bounding boxes are also post-processed to exclude any overlapping detections and undergo sequential expansion to align with the nearest table contour. Figure 2 shows the table detection pipeline on a sample image from ICDAR 2013 [13].

Table Structure Recognition Detected tables are then passed on to the structure recognition stage, which detects all instances of table columns, table column headers, as well as cells. Overlapping bounding boxes of detected cells are merged into a single cell on the basis of the region area. The cells spanning multiple adjacent cells along its horizontal projection are marked as row identifiers and assigned an identity to capture the row number. Using the structural information of columns and the intersection of the generated cell projections with the detected columns, the row structure is inferred. The overall table structure is further processed by classifying the presence of cells in column and column headers, and the predicted structure is written to an XML file in the same format as in other state-of-the-art methods. Figure 3 shows the table structure recognition pipeline on the extracted table from the detection pipeline.

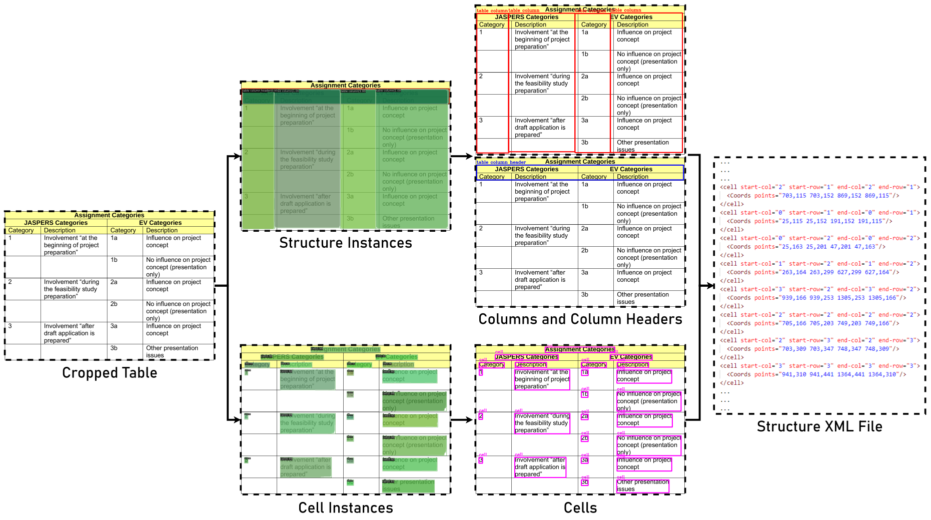


Fig. 3. Table structure recognition pipeline – Instances of cells, columns, and column headers are detected and the row structure is inferred based on the intersection of cell projections with the columns. The overall table structure is inferred based on the positional relation of the cells with the detected columns and generated rows.

3.2 Augmentation Strategy

In this section, we describe the augmentation techniques utilized in our proposed architecture, supplementing the data-hungry nature of transformers.

K-Means Clustering K-means clustering is an unsupervised learning algorithm that is used to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. In the context of table images, the k-means method can be used to group images in the form of vectors based on visual similarity, thus reducing the overall variation in data that is fused together.

Fusion Splicing of distinct tables in horizontal and vertical fashion followed by concatenation is collectively termed *fusion* in this paper. Vertical and horizontal lines are detected using probabilistic Hough lines transform [7]. The median horizontal and vertical set of line points in a sorted Hough lines array is selected as the cutoff point to achieve horizontal and vertical splices of a table image, respectively. The resultant images are then fed into an image resizing-concatenation pipeline to generate a new table.

Figure 4 shows an exemplar pipeline of fusion. A batch of images ($n = 2$) is randomly sampled from clusters formed in section 3.2 and the median horizontal contour is selected as the cutoff point after detection of all possible lines, followed by cropping the image to achieve the maximum area. Cropped images are then

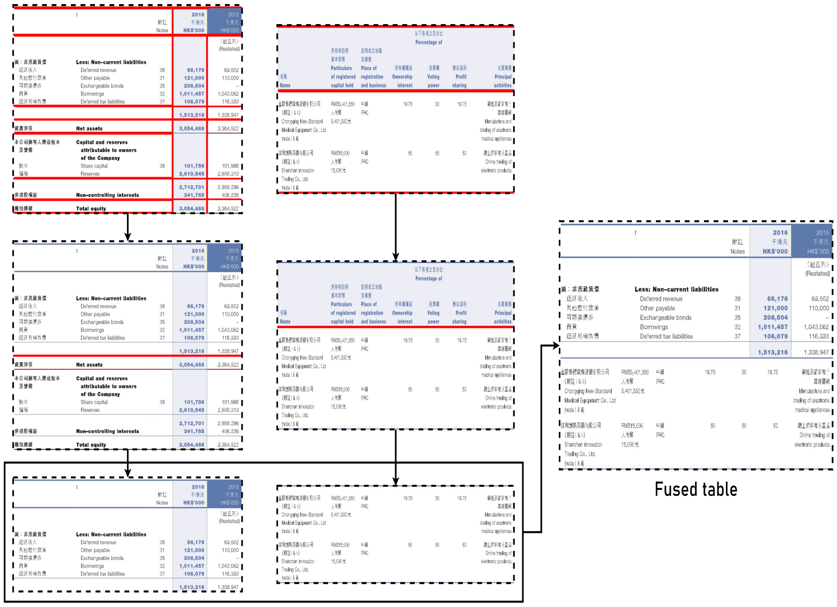


Fig. 4. Generation of new table images – Vertical and horizontal contours are detected on two randomly sampled table images from generated clusters. Tables are cropped to the median positional contour and adjacently joined to produce a new table image.

resized along the horizontal axis so that they match in width before they are concatenated to produce a new table image.

Patching Augmented tables generated as a result of fusion are lastly patched onto existing dataset images. Figure 5 shows an exemplary pipeline of patching. An image is randomly sampled from the training data along with its inverted mask. The center of the largest area in the inverted mask is the point on which a randomly sampled fusion-generated table is pasted on. Collectively, we refer to this process as patching in this paper.

4 Experiments

In this section, we start with an introduction to the datasets utilized to demonstrate the efficacy of our architecture, followed by an introduction to the data augmentation techniques employed in our method. Later, we analyze the results on these datasets and compare them with state-of-the-art methods.

4.1 Datasets

In this section, we will discuss the datasets that are commonly used and publicly available for table detection and table structure recognition.

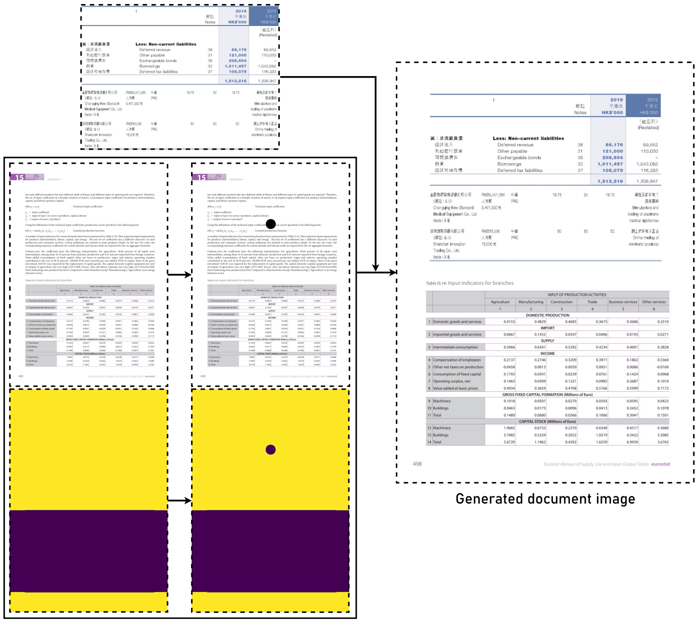


Fig. 5. Patching of table images – A document image along with its semantic information is fed into the pipeline and the best patch point is computed for the fused table to be pasted on in order to generate a new training sample.

ICDAR 2013 The ICDAR 2013 [13] dataset consists of 150 tables, with 75 from EU documents and 75 from US Government documents. The tables are defined by their rectangular coordinates on the page and can span multiple pages. The dataset includes two sub-tasks: identifying the location of tables and determining their structure. In our experiments, we will only utilize the dataset for structure recognition.

ICDAR 2017-POD ICDAR 2017-POD [11] is a widely used dataset for evaluating various table detection methods. It is significantly larger than the ICDAR 2013 table dataset, containing a total of 2,417 images that include figures, tables, and formulae. This dataset is typically divided into 1,600 images with 731 tabular areas for training and 817 images with 350 tabular regions for testing.

ICDAR 2019 cTDaR The cTDaR [10] datasets provide separate tracks, both for table detection and table structure recognition. Track A, which targets the task of table detection, is further divided into archival documents and modern documents. In this paper, we focus on modern documents where table annotations are provided for each image. The modern subset consists of 600 training and 240 test images, which contain a broad variety of PDF files. Variability in the images is further enhanced by supplementing English documents with Chinese documents, both of various formats, including scanned document images, digitally composed documents, etc.

Marmot The Marmot [8] dataset is a collection of 2,000 PDF pages that includes a net balance of positive and negative samples. It features a diverse range of table types, including ruled and non-ruled, horizontal and vertical, and tables that span multiple columns. The dataset also includes tables that are found inside columns. For evaluation, we utilized the corrected version of this dataset, as in [28], which contains 1,967 images.

UNLV The UNLV [29] dataset is a widely recognized collection of document images in the field of document analysis, which includes a total of almost 10,000 images. However, only a subset of 427 images contains tables, and in the experiments, only those images with tabular information were used.

TableBank TableBank [19] proposed an approach for automatically creating a dataset using weak supervision, which generates high-quality labeled data for a diverse range of domains such as business documents, official filings, research papers, and others, making it highly useful for large-scale table recognition tasks. This dataset is comprised of 417,234 labeled tables, along with their corresponding original documents from various domains.

PubLayNet PubLayNet [37] is a high-quality dataset designed for document layout analysis. The dataset is composed of 335,703 training images, 11,245 validation images, and 11,405 testing images. For table detection, only those images containing at least one table were used, resulting in a total of 86,460 images. The evaluation metrics used in this dataset follow the COCO [20] evaluation protocol, rather than precision, recall, and F1-score, as is used in other table detection datasets.

4.2 Settings

The experiments were implemented in PyTorch v1.11.0 and were conducted on Google Colaboratory platform with a P100 PCIE GPU of 16 GB GPU memory, Intel® Xeon® CPU @ 2.30GHz, and 12.72 GB of RAM. The MMDetection toolbox was used to implement the proposed architecture.

Hyperparameters play a crucial role in determining the performance of a deep learning model. They are adjustable settings that are not learned from the data, and must be set before training begins. The choice of hyperparameters can significantly impact the performance of a model, and their optimization is often necessary to achieve the best results. In this paper, we selected hyperparameters based on prior knowledge and empirical studies. This approach has been shown to be effective in selecting reasonable values for the hyperparameters and can save time and resources compared to exhaustive search methods.

The model was optimized using the AdamW algorithm with a batch size of 1 over 180,000 iterations. The learning rate was decayed using a linear decay schedule, with the initial learning rate, betas, epsilon, and weight decay set to 1e-4/1.4, (0.9, 0.999), 1e-8, and 1e-4, respectively. Further studies can be conducted to explore other hyperparameter settings to achieve even better results.

The data augmentation techniques described in Section 3.2 were implemented in conjunction with the augmentation policies used to train the DETR [4] architecture. Two auto-augmentation policies were adopted during the training phase. The first policy rescaled the shorter side of each image to a random number in the set {480, 512, 544, 576, 608, 640, 672, 704, 736, 768} while maintaining the aspect ratio. The second policy rescaled the image to a random number in the set {400, 500, 600} before applying an absolute range cropping window of size (384, 600). During the testing phase, the longer side of each image was rescaled to 1024 while maintaining the aspect ratio.

5 Results & Analysis

In this section, we discuss the results of the proposed architecture on the datasets introduced in Section 4.1 and compare them with state-of-the-art methods. We also evaluate the efficacy of our augmentation pipeline by training the proposed architecture using different augmentation methodologies.

To assess the effectiveness of the proposed tabular image generative augmentation pipeline, the proposed architecture was trained using three distinct methodologies:

1. **Non-Augmented (NA):** Training images are fed into the transformer without any modifications.
2. **Standard (S):** Standard augmentation techniques such as variations in brightness, exposure, contrast, jitter, etc. combined with strategies employed in DETR [4].
3. **Generative (G) (ours):** Our proposed augmentation pipeline, which consists of sequential clustering, patching, and fusion to generate new images in combination with strategies employed in DETR [4].

5.1 Table Detection

Table 1 presents a summary of the table detection results on various datasets. It does not include table detection performance comparison on PubLayNet and ICDAR 2019 cTDaR as they follow different evaluation criteria and are presented separately. The model is initially trained on a conglomerate of training images of PubLayNet, TableBank, and ICDAR 2019 cTDaR dataset. Additionally, the document images generated by our augmentation technique are also included in the initial training state. Following the strategy of other state-of-the-art methods, we fine-tune these weights on training images of each of the table detection datasets for the computation of respective evaluation metrics.

Table 1. Table detection performance comparison summary – All metrics are computed using models fine-tuned on the training samples of respective datasets – NA: Non-Augmented, S: Standard, G: Generative.

Dataset	Method	Precision	Recall	F1-Score
ICDAR 2017 POD @ $IoU = 0.8$	CDeC-Net [1]	89.9	96.9	93.4
	DeepTabStR [30]	96.5	97.1	96.8
	YOLOv3 [31]	97.8	97.2	97.5
	HybridTabNet [22]	87.8	**99.3**	93.2
	PyramidTabNet (NA)	95.3	94.7	95.0
	PyramidTabNet (S)	97.8	97.1	97.4
	PyramidTabNet (G)	**99.8**	**99.3**	**99.5**
Marmot @ $IoU = 0.5$	DeCNT [31]	94.6	84.9	89.5
	CDeC-Net [1]	77.9	94.3	86.1
	HybridTabNet [22]	88.2	91.5	89.8
	CasTabDetectoRS [14]	96.5	**95.2**	95.8
	PyramidTabNet (NA)	92.7	91.1	91.9
	PyramidTabNet (S)	94.6	93.3	93.9
	PyramidTabNet (G)	**97.7**	94.9	**96.3**
UNLV @ $IoU = 0.5$	DeCNT [31]	91.0	94.6	92.8
	CDeC-Net [1]	91.5	97.0	94.3
	HybridTabNet [22]	**96.2**	96.1	**95.6**
	CasTabDetectoRS [14]	92.8	96.4	94.6
	PyramidTabNet (NA)	89.4	93.2	91.3
	PyramidTabNet (S)	90.7	95.6	93.1
	PyramidTabNet (G)	92.1	**98.2**	95.1
TableBank (LaTeX & Word) @ $IoU = 0.5$	Li et al. [19]	90.4	95.9	93.1
	CascadeTabNet [24]	95.7	94.4	94.3
	HybridTabNet [22]	95.3	97.6	96.5
	CasTabDetectoRS [14]	98.2	97.4	97.8
	PyramidTabNet (NA)	94.4	94.1	94.2
	PyramidTabNet (S)	96.5	95.6	96.0
	PyramidTabNet (G)	**98.9**	**98.2**	**98.5**

All the metrics in Table 1 are computed at the same IoU threshold for a single multi-cell row (same dataset). In the evaluation of the ICDAR 2017-POD (Page Object Detection) dataset, we achieved an F1-score of 99.5 on the detection of the table class at 0.8 IoU threshold, pushing further the state-of-the-art metrics. It should be noted that the results reported are after the inclusion of post-processing techniques, as also observed in the original competition.

Table 2. Table detection performance comparison on PubLayNet – Evaluation metrics follow the same protocol as in the COCO [20] detection challenge – NA: Non-Augmented, S: Standard, G: Generative.

Method	$AP^{0.5:0.95}$	$AP^{0.75}$	$AP^{0.95}$
CDeC-Net [1]	96.7	-	-
RobusTabNet [21]	97.0	97.8	92.0
DiT-L (Cascade) [18]	97.8	-	-
PyramidTabNet (NA)	94.6	95.4	92.7
PyramidTabNet (S)	96.9	97.6	94.3
PyramidTabNet (G)	**98.1**	**98.8**	**96.4**

On the Marmot dataset, our model achieves the highest precision of 97.7 and F1-score of 96.3 at 0.5 IoU threshold. The direct comparison of our results with CasTabDetectoRS [14] and HybridTabNet [22] on the Marmot dataset proves that we have pronounced the new state-of-the-art.

On the UNLV dataset, we achieved the highest recall of 98.2 at 0.5 IoU threshold, indicating that our method correctly identified the highest number of tables in the dataset. However, a decrease in precision was observed in comparison to the performance on other datasets. We attribute this to the presence of a large proportion of low-quality document images in the UNLV dataset. Our model, which was trained on a diverse set of modern document images, may not have the ability to fine-tune on the UNLV dataset as well as it does on other modern datasets.

On the TableBank dataset, we achieved the highest precision, recall, and F1-score over a 0.5 IoU threshold, achieving the new state-of-the-art. We evaluate the TableBank dataset on both of its parts, LaTeX and Word document image subsets, as we believe it signifies the robustness of the proposed method to different types of modern document images.

The results of table detection on the PubLayNet dataset are shown in Table 2. Utilizing the same evaluation protocol as the COCO detection challenge [20], our method achieved the highest AP of 96.4 at 0.95 IoU threshold and a value of 98.1 for precision averaged over IoUs from 0.5 to 0.95 in steps of 0.05. These results further push the state-of-the-art and demonstrate the fine-grained object detection capabilities of our method.

The results of table detection on the ICDAR 2019 cTDaR dataset are shown in Table 3. As the number of samples in this dataset is relatively small, it aims to evaluate the few-short learning capabilities of models under low-resource scenarios. In Table 3, Our model performs better than the current baselines on all fronts, while observing an increase of 0.9% in weighted F1-score over the recent cascaded DiT-L [18], pushing further the state-of-the-art. It is worth noting that, like DiT, metrics of IoU@{0.9} are significantly performant, indicating that the proposed architecture has better fine-grained object detection capabilities.

Table 3. Table detection performance comparison on ICDAR 2019 cTDaR – F1-scores are computed at different IoU thresholds and the weighted F1-score is used to determine the overall performance – NA: Non-Augmented, S: Standard, G: Generative.

Method	$IoU_{0.6}$	$IoU_{0.7}$	$IoU_{0.8}$	$IoU_{0.9}$	$wF1$
CascadeTabNet [24]	94.3	93.4	92.5	90.1	90.1
HybridTabNet [22]	95.3	94.2	93.3	92.7	92.8
CDeC-Net [1]	95.9	95.6	95.0	91.5	94.3
GTE [36]	96.9	96.9	95.7	91.9	95.1
TableDet [9]	-	-	95.5	89.5	94.0
DiT-L (Cascade) [18]	97.9	97.2	97.0	93.9	96.3
PyramidTabNet (NA)	95.8	95.6	94.4	92.3	94.3
PyramidTabNet (S)	97.0	96.9	96.3	94.1	95.9
PyramidTabNet (G)	**98.7**	**98.7**	**98.0**	**94.5**	**97.2**

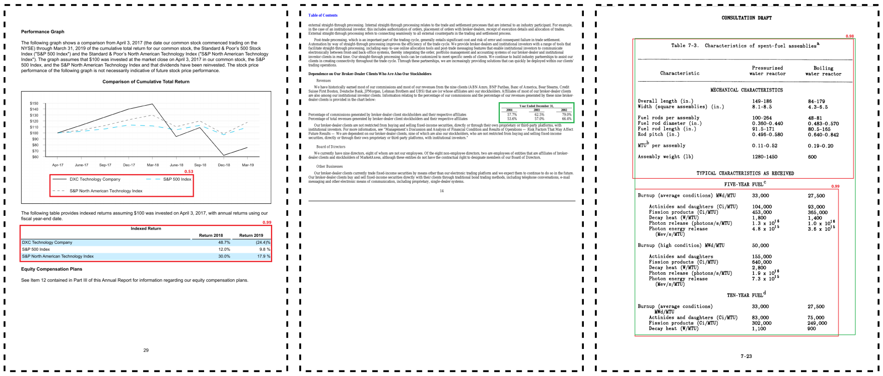
5.2 Structure Recognition

To evaluate the efficacy of our model on table structure recognition relative to other state-of-the-art methods, each cell is marked into its corresponding location in the detected table. The location of each cell is represented by [*start row*, *end row*, *start column*, *end column*, *box coordinates*]. We employ the same IoU threshold as in other methods to compute precision, recall, and F1-score.

The results of structure recognition on the ICDAR 2013 dataset are shown in Table 4. We achieved an F1-score of 93.8, lacking just behind GuiderTSR [15], which proved the efficacy of estimation of viable anchors for the detection of rows and columns over naively applied object detection algorithms.

Table 4. Table structure recognition results on ICDAR 2013 – Performance of fine-tuned models is compared without post-processing techniques as is done in the current state-of-the-art method – NA: Non-Augmented, S: Standard, G: Generative.

Method	Precision	Recall	F1-Score
SPLERGE [33]	**96.9**	90.1	93.4
TableNet [23]	92.2	89.9	91.0
GraphTSR [5]	88.5	86.0	87.2
TabStruct-Net [26]	92.7	91.1	91.9
GuidedTSR [15]	93.6	94.4	**94.2**
PyramidTabNet (NA)	91.1	90.4	90.7
PyramidTabNet (S)	91.4	92.6	92.0
PyramidTabNet (G)	92.3	**95.3**	93.8



Performance Graph

The following graph shows a comparison from April 3, 2017 (the date our common stock commenced trading on the NYSE) through March 31, 2019 of the cumulative total return for our common stock, the Standard & Poor's 500 Stock Index ("S&P 500 Index") and the Standard & Poor's North American Technology Index ("S&P North American Technology Index"). The graph assumes that $100 was invested at the market close on April 3, 2017 in our common stock, the S&P 500 Index, and the S&P North American Technology Index and that dividends have been reinvested. The stock price performance of the following graph is not necessarily indicative of future stock price performance.

Comparison of Cumulative Total Return

The following table provides indexed returns assuming $100 was invested on April 3, 2017, with annual returns using our fiscal year-end date.

Indexed Return	Return 2018	Return 2019
DXC Technology Company	48.7%	(24.4)%
S&P 500 Index	12.0%	9.8 %
S&P North American Technology Index	30.0%	17.9 %

Equity Compensation Plans

See Item 12 contained in Part III of this Annual Report for information regarding our equity compensation plans.

29

Table of Contents

external straight-through processing. Internal straight-through processing relates to the trade and settlement processes that are internal to an industry participant. For example, in the case of an institutional investor, this includes authorization of orders, placement of orders with broker-dealers, receipt of execution details and allocation of trades. External straight-through processing refers to connecting seamlessly to all external counterparts in the trading and settlement process.

Post-trade processing, which is an important part of the trading cycle, generally entails significant cost and risk of error and consequent failure in trade settlement. Automation by way of straight-through processing improves the efficiency of the trade cycle. We provide broker-dealers and institutional investors with a range of tools that facilitate straight-through processing, including easy-to-use online allocation tools and post-trade messaging features that enable institutional investors to communicate electronically between front-and back-office systems, thereby integrating the order, portfolio management and accounting systems of our broker-dealer and institutional investor clients in real time. Our straight-through processing tools can be customized to meet specific needs of clients. We continue to build industry partnerships to assist our clients in creating connectivity throughout the trade cycle. Through these partnerships, we are increasingly providing solutions that can quickly be deployed within our clients' trading operations.

Dependence on Our Broker-Dealer Clients Who Are Also Our Stockholders

Revenues

We have historically earned most of our commissions and most of our revenues from the nine clients (ABN Amro, BNP Paribas, Banc of America, Bear Stearns, Credit Suisse First Boston, Deutsche Bank, JPMorgan, Lehman Brothers and UBS) that are (or whose affiliates are) our stockholders. Affiliates of most of our broker-dealer clients are also among our institutional investor clients. Information relating to the percentage of our commissions and the percentage of our revenues generated by these nine broker-dealer clients is provided in the chart below:

	Year Ended December 31,		
	2004	2003	2002
Percentage of commissions generated by broker-dealer client stockholders and their respective affiliates	57.7%	62.5%	79.0%
Percentage of total revenues generated by broker-dealer client stockholders and their respective affiliates	53.6%	57.0%	66.4%

Our broker-dealer clients are not restricted from buying and selling fixed-income securities, directly or through their own proprietary or third-party platforms, with institutional investors. For more information, see "Management's Discussion and Analysis of Financial Condition and Results of Operations — Risk Factors That May Affect Future Results — We are dependent on our broker-dealer clients, nine of which are also our stockholders, who are not restricted from buying and selling fixed-income securities, directly or through their own proprietary or third-party platforms, with institutional investors."

Board of Directors

We currently have nine directors, eight of whom are not our employees. Of the eight non-employee directors, two are employees of entities that are affiliates of broker-dealer clients and stockholders of MarketAxess, although these entities do not have the contractual right to designate members of our Board of Directors.

Other Businesses

Our broker-dealer clients currently trade fixed-income securities by means other than our electronic trading platform and we expect them to continue to do so in the future. Our broker-dealer clients buy and sell fixed-income securities directly with their clients through traditional bond trading methods, including telephone conversations, e-mail messaging and other electronic means of communication, including proprietary, single-dealer systems.

14

CONSULTATION DRAFT

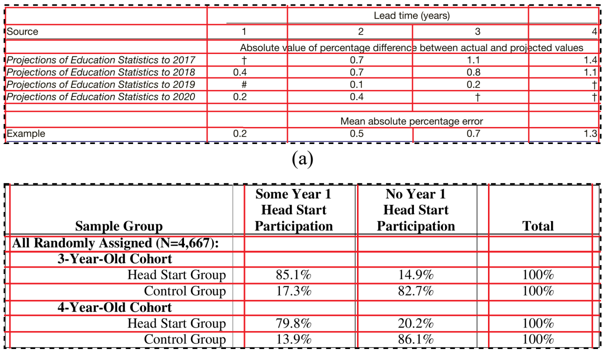
Table 7-3. Characteristics of spent-fuel assemblies[a]

Characteristic	Pressurized water reactor	Boiling water reactor
MECHANICAL CHARACTERISTICS		
Overall length (in.)	149-186	84-179
Width (square assemblies) (in.)	8.1-8.5	4.3-6.5
Fuel rods per assembly	100-264	48-81
Fuel rod diameter (in.)	0.360-0.440	0.483-0.570
Fuel rod length (in.)	91.5-171	80.5-165
Rod pitch (in.)	0.496-0.580	0.640-0.842
MTU[b] per assembly	0.11-0.52	0.19-0.20
Assembly weight (lb)	1280-1450	600
TYPICAL CHARACTERISTICS AS RECEIVED		
FIVE-YEAR FUEL[c]		
Burnup (average conditions) MWd/MTU	33,000	27,500
Actinides and daughters (Ci/MTU)	104,000	93,000
Fission products (Ci/MTU)	453,000	365,000
Decay heat (W/MTU)	1,800	1,400
Photon release (photons/s/MTU)	1.3×10^{16}	1.0×10^{16}
Photon energy release (Mev/s/MTU)	4.8×10^{15}	3.6×10^{15}
Burnup (high condition) MWd/MTU	50,000	
Actinides and daughters	155,000	
Fission products (Ci/MTU)	640,000	
Decay heat (W/MTU)	2,800	
Photon release (photons/s/MTU)	1.9×10^{16}	
Photon energy release (Mev/s/MTU)	7.3×10^{15}	
TEN-YEAR FUEL[d]		
Burnup (average conditions) MWd/MTU	33,000	27,500
Actinides and daughters (Ci/MTU)	83,000	75,000
Fission products (Ci/MTU)	302,000	249,000
Decay heat (W/MTU)	1,100	900

7-23

(a) (b) (c)

Fig. 6. Table detection examples with (a) incorrect detection, (b) unidentified table, (c) partial detection.



		Lead time (years)		
Source	1	2	3	4
	Absolute value of percentage difference between actual and projected values			
Projections of Education Statistics to 2017	†	0.7	1.1	1.4
Projections of Education Statistics to 2018	0.4	0.7	0.8	1.1
Projections of Education Statistics to 2019	#	0.1	0.2	†
Projections of Education Statistics to 2020	0.2	0.4	†	†
		Mean absolute percentage error		
Example	0.2	0.5	0.7	1.3

(a)

Sample Group	Some Year 1 Head Start Participation	No Year 1 Head Start Participation	Total
All Randomly Assigned (N=4,667):			
3-Year-Old Cohort			
Head Start Group	85.1%	14.9%	100%
Control Group	17.3%	82.7%	100%
4-Year-Old Cohort			
Head Start Group	79.8%	20.2%	100%
Control Group	13.9%	86.1%	100%

(b)

Fig. 7. Structure recognition examples with (a) column error (b) row error.

5.3 Analysis

In this section, we analyze the detection outputs of the proposed architecture and provide potential reasons for the incorrectly detected tables and their structural components. It provides valuable insights into the strengths and limitations of the model and will be useful for guiding future improvements to the model.

The three common types of errors in table detection are depicted in Figure 6. In Figure 6a, our model mistakenly identifies the figure legend as a table due to the presence of the x axis label above it. Conversely, in Figure 6b, the model fails to detect a table that is very small in relation to the overall image size. Figure 6c showcases the instance when the table size encompasses the entire image and our model fails to detect it as a whole, instead recognizing it in parts. This behavior is attributed to the use of patching augmentation techniques, which ensure the presence of a minimum of two tables in a single document image.

The error types in structure recognition are illustrated in Figure 7. The figure depicts the issues that arise from under-identified rows or over-identified columns. As shown in Figure 7a, the failure of our model to detect the second column header cell leads to a partially broken structure, demonstrating the critical dependence of our structure recognition pipeline on correctly identifying all column header cells. In Figure 7b, post-processing techniques result in the merging of two cells from separate rows, leading to an extra row in the end table structure. Despite this, post-processing techniques only counteract the model predictions for a limited number of test images. Thus, it is retained as the final stage of our proposed method after empirical evaluation.

6 Conclusion & Future Work

In this paper, we present PyramidTabNet, an end-to-end approach to table detection and structure recognition in image-based documents based on convolutionless image transformers. To make up for the data-hungry nature of transformers, PyramidTabNet employs a tabular image generative augmentation technique, resulting in an architecture with fine-grained object detection capabilities. Consequently, and through experimental results, we have shown that PyramidTabNet outperforms several strong baselines in the task of table detection, especially at a high IoU threshold, and achieves competitive and comparable performance on table structure recognition tasks.

For future work, we will study the effects of training PyramidTabNet on even larger datasets to further push the state-of-the-art results on table recognition. We are also exploring the effects of integrating AI image upscaling on detected table images to improve the evaluation metrics on the task of table structure recognition, however, with an added latency overhead.

References

1. Agarwal, M., Mondal, A., Jawahar, C.: CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 9491–9498. IEEE (2021)
2. Arif, S., Shafait, F.: Table Detection in Document Images using Foreground and Background Features. In: 2018 20th Digital Image Computing: Techniques and Applications (DICTA). pp. 1–8. IEEE (2018)
3. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving Into High Quality Object Detection. In: 2018 Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6154–6162. IEEE (2018)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection With Transformers. In: 2020 16th European Conference on Computer Vision (ECCV). pp. 213–229. Springer (2020)
5. Chi, Z., Huang, H., Xu, H.D., Yu, H., Yin, W., Mao, X.L.: Complicated Table Structure Recognition. arXiv preprint arXiv:1908.04729 (2019)
6. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable Convolutional Networks. In: 2017 16th International Conference on Computer Vision (ICCV). pp. 764–773. IEEE (2017)

7. Duan, D., Xie, M., Mo, Q., Han, Z., Wan, Y.: An Improved Hough Transform for Line Detection. In: 2010 International Conference on Computer Application and System Modeling (ICCASM). vol. 2, pp. 354–357 (2010)
8. Fang, J., Tao, X., Tang, Z., Qiu, R., Liu, Y.: Dataset, Ground-Truth and Performance Metrics for Table Detection Evaluation. In: 2012 10th IAPR International Workshop on Document Analysis Systems (DAS). pp. 445–449 (2012)
9. Fernandes, J., Simsek, M., Kantarci, B., Khan, S.: TableDet: An End-to-End Deep Learning Approach for Table Detection and Table Image Classification in Data Sheet Images. In: Neurocomputing. vol. 468, pp. 317–334. Elsevier (2022)
10. Gao, L., Huang, Y., Déjean, H., Meunier, J.L., Yan, Q., Fang, Y., Kleber, F., Lang, E.: ICDAR 2019 Competition on Table Detection and Recognition (cTDaR). In: 2019 16th International Conference on Document Analysis and Recognition (ICDAR). pp. 1510–1515 (2019)
11. Gao, L., Yi, X., Jiang, Z., Hao, L., Tang, Z.: ICDAR 2017 Competition on Page Object Detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 1417–1422 (2017)
12. Gilani, A., Qasim, S.R., Malik, I., Shafait, F.: Table Detection Using Deep Learning. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 771–776. IEEE (2017)
13. Göbel, M., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 Table Competition. In: 2013 12th International Conference on Document Analysis and Recognition (ICDAR). pp. 1449–1453 (2013)
14. Hashmi, K.A., Pagani, A., Liwicki, M., Stricker, D., Afzal, M.Z.: CasTabDetectoRS: Cascade Network for Table Detection in Document Images With Recursive Feature Pyramid and Switchable Atrous Convolution. In: Journal of Imaging. vol. 7, p. 214. MDPI (2021)
15. Hashmi, K.A., Stricker, D., Liwicki, M., Afzal, M.N., Afzal, M.Z.: Guided Table Structure Recognition Through Anchor Optimization. In: IEEE Access. vol. 9, pp. 113521–113534. IEEE (2021)
16. Khan, S.A., Khalid, S.M.D., Shahzad, M.A., Shafait, F.: Table Structure Extraction with Bi-Directional Gated Recurrent Unit Networks. In: 2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 1366–1371. IEEE (2019)
17. Khan, U., Zahid, S., Ali, M.A., Ul-Hasan, A., Shafait, F.: TabAug: Data Driven Augmentation for Enhanced Table Structure Recognition. In: 2021 16th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 2, pp. 585–601. Springer (2021)
18. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.: DiT: Self-Supervised Pre-training for Document Image Transformer. In: 2022 30th ACM International Conference on Multimedia (ACM MM). pp. 3530–3539 (2022)
19. Li, M., Cui, L., Huang, S., Wei, F., Zhou, M., Li, Z.: TableBank: Table Benchmark for Image-Based Table Detection and Recognition. In: 2020 12th Language Resources and Evaluation Conference (LREC). pp. 1918–1925 (2020)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: 2014 European Conference on Computer Vision (ECCV). pp. 740–755. Springer (2014)
21. Ma, C., Lin, W., Sun, L., Huo, Q.: Robust Table Detection and Structure Recognition from Heterogeneous Document Images. In: Pattern Recognition. vol. 133, p. 109006. Elsevier (2023)

22. Nazir, D., Hashmi, K.A., Pagani, A., Liwicki, M., Stricker, D., Afzal, M.Z.: Hybridtabnet: Towards Better Table Detection in Scanned Document Images. In: Applied Sciences. vol. 11, p. 8396. MDPI (2021)
23. Paliwal, S.S., Vishwanath, D., Rahul, R., Sharma, M., Vig, L.: TableNet: Deep Learning Model for End-To-End Table Detection and Tabular Data Extraction from Scanned Document Images. In: 2019 15th International Conference on Document Analysis and Recognition (ICDAR). pp. 128–133. IEEE (2019)
24. Prasad, D., Gadpal, A., Kapadni, K., Visave, M., Sultanpure, K.: CascadeTabNet: An Approach for End-to-End Table Detection and Structure Recognition from Image-Based Documents. In: 2020 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 572–573 (2020)
25. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking Table Recognition Using Graph Neural Networks. In: 2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 142–147. IEEE (2019)
26. Raja, S., Mondal, A., Jawahar, C.: Table Structure Recognition Using Top-Down and Bottom-Up Cues. In: 2020 16th European Conference on Computer Vision (ECCV). pp. 70–86. Springer (2020)
27. Raja, S., Mondal, A., Jawahar, C.: Visual Understanding of Complex Table Structures from Document Images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2299–2308 (2022)
28. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: 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. 1, pp. 1162–1167 (2017)
29. Shahab, A., Shafait, F., Kieninger, T., Dengel, A.: An Open Approach Towards The Benchmarking of Table Structure Recognition Systems. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. pp. 113–120 (2010)
30. Siddiqui, S.A., Fateh, I.A., Rizvi, S.T.R., Dengel, A., Ahmed, S.: DeepTabStR: Deep Learning Based Table Structure Recognition. In: 2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 1403–1409 (2019)
31. Siddiqui, S.A., Malik, M.I., Agne, S., Dengel, A., Ahmed, S.: DeCNT: Deep Deformable CNN for Table Detection. In: IEEE Access. vol. 6, pp. 74151–74161. IEEE (2018)
32. Smock, B., Pesala, R., Abraham, R.: PubTables-1M: Towards Comprehensive Table Extraction from Unstructured Documents. In: 2022 Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4634–4642 (2022)
33. Tensmeyer, C., Morariu, V.I., Price, B., Cohen, S., Martinez, T.: Deep Splitting and Merging for Table Structure Decomposition. In: 2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 114–121. IEEE (2019)
34. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction Without Convolutions. In: 2021 17th International Conference on Computer Vision (ICCV). pp. 568–578. IEEE (2021)
35. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: PVT v2: Improved Baselines With Pyramid Vision Transformer. In: Computational Visual Media. vol. 8, pp. 415–424. Springer (2022)

36. Zheng, X., Burdick, D., Popa, L., Zhong, X., Wang, N.X.R.: Global Table Extractor (GTE): A Framework for Joint Table Identification and Cell Structure Recognition using Visual Context. In: 2021 Winter Conference on Applications of Computer Vision (WACV). pp. 697–706 (2021)
37. Zhong, X., Tang, J., Yepes, A.J.: PubLayNet: Largest Dataset Ever for Document Layout Analysis. In: 2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 1015–1022. IEEE (2019)