

Article

TableExtractNet: A Model of Automatic Detection and Recognition of Table Structures from Unstructured Documents

Thokozani Ngubane  and Jules-Raymond Tapamo * 

Discipline of Electrical, Electronic and Computer Engineering, University of KwaZulu-Natal,
Durban 4041, South Africa; 218002981@stu.ukzn.ac.za

* Correspondence: tapamoj@ukzn.ac.za; Tel.: +27-31-260-2751

Abstract: This paper presents TableExtractNet, a model that automatically finds and understands tables from scanned documents, tasks that are essential for quick use of information in many fields. This is driven by the growing need for efficient and accurate table interpretation in business documents where tables enhance data communication and aid decision-making. The model uses a mix of two advanced techniques, CornerNet and Faster R-CNN, to accurately locate tables and understand their layout. Tests on standard datasets, IIIT-AR-13K, STDW, SciTSR, and PubTabNet, show that this model performs better than previous ones, making it very good at dealing with tables that have complicated designs or are in documents with a lot of detail. The success of this model marks a step forward in making document analysis more automated. It makes it easier to turn complex scanned documents containing tables into data that are more manipulable by computers.

Keywords: table detection; structure recognition; deep learning; CornerNet; Faster R-CNN; document analysis; data conversion



Citation: Ngubane, T.; Tapamo, J.-R. TableExtractNet: A Model of Automatic Detection and Recognition of Table Structures from Unstructured Documents. *Informatics* **2024**, *11*, 77. <https://doi.org/10.3390/informatics11040077>

Academic Editor: Zhiwen Yu

Received: 21 August 2024

Revised: 13 October 2024

Accepted: 17 October 2024

Published: 25 October 2024



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

1. Introduction

Tables are frequently employed in business documents to improve the communication of raw data to humans [1]. They enable quick searching, comparing, and understanding of information, which greatly aids in drawing conclusions. Consequently, the ability to identify and interpret tables in business documents is essential, highlighting the growing significance of research on document recognition and analysis [2]. With the increasing volume of documents, there is a heightened demand for techniques that can automatically detect tables and decipher their structure from document images, which is beneficial for various applications such as information retrieval and question answering. Recognizing the structure of tables is vital as it allows for the content to be presented in a compact format across various digital document types, including web pages, PDFs, Word documents, and images. It is crucial to reformat data from unstructured tables into an organized, machine-readable layout to support NLP algorithms in data management. The objective in locating tables is to identify the exact starting and ending points, while the purpose of table structure recognition (TSR) is to reconstruct the framework of each identified table, defining the bounding coordinates of each cell and the extent of row and column traversal [3].

This research addresses the TSR challenge in document images, as almost all digital documents can be converted into image format. This task is difficult, considering that images only capture the visual representation, and complex table structures may include challenging elements such as merged cells, multi-row configurations, and partially or entirely unbordered cells [4]. The successful application of deep learning in solving computational problems across various fields has led to numerous attempts to develop new approaches for table structure recognition, yielding promising results. Graph models are frequently utilized to represent the structure of tables, with cells as vertices and edges representing the relationships between cells. These relationships are typically categorized

as “vertical linkage”, “horizontal linkage”, or “no linkage” to form the table’s framework. Numerous recent efforts have been made to develop models and systems for detecting tables and cells and recognizing table structures [5,6].

The detection and recognition of table structures in unstructured documents pose several challenges, such as variations in document layouts, font styles, and image quality, which often result in errors and inaccuracies during the extraction process. This study aims to overcome these challenges by investigating new techniques and methodologies that can enhance table extraction performance from diverse document images. The method described in [4] serves as a foundation and reference for this research. By expanding on the concepts presented in that paper, we strive to improve the efficiency and accuracy of table detection and recognition.

The primary contribution of this research project lies in enhancing the efficiency and precision of table detection and structure recognition in unstructured documents. By building on the methodologies presented in [4], this research contributes the following:

- **Enhancement of Existing Methods:** The study of the approach in [4] offers valuable insights into the state-of-the-art techniques for table detection and recognition. By analyzing the limitations of these methods, we introduced novel improvements and modifications to refine and optimize the existing techniques.
- **Development of Innovative Techniques and Algorithms:** New algorithms for table detection and structure recognition were introduced, including the integration of CornerNet and Faster R-CNN with ResNet-18 backbones for table detection. For table structure recognition, a novel Split-and-Merge Module with ResNet-18 and FPN backbone, utilizing Spatial and Grid CNNs, was designed to ensure accurate segmentation and reassembly of complex tables. These algorithms dynamically adapt to varying document layouts, font styles, and image quality, enhancing precision and robustness in table detection.
- **Improvements in Table Detection and Layout Handling:** Enhancements were made to effectively manage multi-line text within table cells, particularly in densely populated tables, ensuring accurate interpretation across diverse table designs. These improvements address the challenges identified in [4], where previous methods struggled with closely positioned tables and cells containing multi-line content. Specifically, algorithms were refined to better parse and segment complex table layouts.
- **Improvement of an Advanced Table Interpretation System:** An advanced system was developed to efficiently handle the complexities of tables with multiple structural variations, enabling improved data extraction and analysis. This system overcomes the limitations of previous approaches, particularly the difficulty in processing extremely dense tables with overlapping segmentation masks. The new system incorporates enhanced parsing techniques to precisely delineate table boundaries and cell contents.
- **Extensive experimentation and performance evaluation of the proposed techniques and algorithms were conducted.** The proposed solutions outperformed state-of-the-art methods, particularly on tables with varying backgrounds, text typefaces, and line colors. This ensures the generalizability of our technique. The experiments also demonstrated the robustness of the adjusted loss functions in terms of table detection and recognition accuracy.

The rest of the paper is organized as follows: Section 2 reviews current computer vision methods and techniques for table identification and structure recognition. Section 3 presents materials and methods for table detection and structure recognition, while Section 4 discusses the environmental setup, datasets used, experimental findings with the proposed methods, and a comparison with the state-of-the-art. Section 5 concludes the research and suggests future directions.

2. Related Work

The detection of tables, cell identification, and the recognition of table structures within documents have been key areas of focus in recent research. Heuristic methods for

managing PDF documents have become popular due to their effectiveness, often involving the establishment of rules and the use of document metadata. However, these approaches are limited because they cannot process document images and generally lack flexibility in adapting to the variability in table structures. As discussed by Shigarov et al. [7], these methods typically lack generalization due to differences in table structures. Despite their effectiveness, heuristic methods are fundamentally different from machine learning techniques, and the abrupt transition between these two in previous discussions may lead to confusion. To address this, it is necessary to clearly distinguish between these paradigms, with heuristic methods being rule-based and machine learning relying on learned models, particularly for tasks related to table detection and recognition.

Over time, research has evolved from classical methods like heuristic and rule-based approaches to more complex models such as machine learning and deep learning approaches. Classical methods like the heuristic approach in [7] have limitations when applied to complex or variable table layouts. These methods, while computationally efficient, often fail when faced with intricate document designs. In contrast, deep learning models such as Faster-RCNN [8] and Mask-RCNN [9] have demonstrated better adaptability and generalization to varied table layouts.

The related work is divided into two specific areas of table processing to provide a clearer and more structured summary of the recent developments: Table Detection and Table Structure Recognition.

2.1. Table Detection

Table detection methods focus on identifying table regions within documents, often by applying deep learning models designed for object detection. Popular models such as Faster-RCNN [8], Mask-RCNN [9], and Fully Convolutional Networks (FCNs) [10] have been employed for this task due to their success in detecting tables and associated document elements. These models, particularly those incorporating deep learning, address challenges related to object detection and segmentation. More advanced detection models such as Global Table Extractor (GTE) [11] and CascadeTabNet [12] have been proposed to further enhance the identification of tables and related document elements.

Various logical models of table layouts, such as grid-based, hierarchical, and relational models, have been explored to enhance the detection process. Grid-based models consider tables as fixed structures, whereas hierarchical models provide better flexibility for representing more complex structures like merged cells or nested tables [13]. These models influence how well table detection systems generalize across different document formats and layouts.

Yang et al. [14] developed a multimodal FCN (fully convolutional network) for page segmentation that identifies tables and other elements across various document layouts (evaluated on the SciTSR dataset). Despite achieving high accuracy (92.1% precision, 90.4% recall, and 91.2% F1-score), the model struggles with more complex table structures.

He et al. [15] proposed a multi-scale, multi-task FCN that predicts segmentation masks for table and figure regions, improving accuracy in complex layouts (evaluated on the TableBank dataset). However, the method demands high computational resources, with a reported precision of 93.5%, recall of 92.7%, and F1-score of 93.1%.

Wang et al. [16] introduced a segmentation collaboration and alignment network (SCAN), evaluated on STDW, that performs well in recognizing table structures in natural environments, though it struggles with extreme distortions. SCAN has achieved a precision of 90.5%, recall of 89.2%, and F1-score of 89.8%.

Zheng et al. [11] and Qiao et al. [17] proposed methods to organize cells into rows and columns, improving overall table detection. Zheng et al. (evaluated on PubTabNet) reported a precision of 94.1%, recall of 93.3%, and F1-score of 93.7%.

2.2. Table Structure Recognition

Recognizing the internal structure of tables, such as their rows, columns, and cells, poses additional challenges. Recent approaches for table structure recognition also involve deep learning models like Faster-RCNN and Mask-RCNN, which are evaluated on datasets like SciTSR and PubTabNet to help identify complex table structures. Some notable works include:

Xiao et al. [2] introduced a graph-based approach where table components (e.g., cells) are treated as nodes and their relationships as edges (evaluated on TableBank). While effective for capturing complex table relationships, it faces difficulties with large tables due to the high computational demands of graph algorithms (precision of 91.3%).

DeepDeSRT [18] (evaluated on SciTSR) utilizes an FCN-based semantic segmentation method for table structure recognition (TSR). This method faces challenges with tables containing many blank spaces or irregular cell sizes but has made progress with feature pooling strategies, achieving a precision of 90.6%, recall of 88.7%, and F1-score of 89.0%.

Kasar et al. [19] (evaluated on ICDAR-2013) developed a method to extract table data based on user queries, translating them into attributed relational graphs for matching. Despite high precision (90.2%), it struggles with diverse formats and noisy data, achieving an F1-score of 89.5%.

Raja et al. [20] introduced a new loss function to enhance cell detection and clustering accuracy (evaluated on ICDAR-2019) by recognizing natural alignment in tables. The model improves accuracy in well-aligned tables (92.7% precision, 91.3% recall, and 92.0% F1-score), though it has limitations with misaligned or skewed tables.

PubTabNet, created by Zhong et al. [21], is a benchmark dataset for table recognition containing 568,000 images with HTML representations. Their attention-based encoder-decoder model advances state-of-the-art table structure recognition, achieving a precision of 95.6%, recall of 94.4%, and F1-score of 95.0%.

Lin et al. [22] developed TSRFormer using a two-stage DETR-based approach called SepRETR (evaluated on PubTabNet and SciTSR), designed for handling complex table structures in distorted images. It demonstrates stability with real-world data, achieving a precision of 95.8%, recall of 96.5%, and F1-score of 95.1%.

2.3. Quality Assessment Metrics

Quality assessment metrics, such as precision, recall, and F1-score, are widely used to evaluate the performance of table detection and structure recognition models. Despite sharing the same evaluation metrics, these two tasks, table detection, and table structure recognition are fundamentally different in their objectives and methods. Table detection focuses on locating tables within a document, while table structure recognition delves into identifying the relationships between cells, rows, and columns within a table. Therefore, we present the results for these tasks separately, as shown in Table 1 for Table Detection and Table 2 for Table Structure Recognition. This separation ensures a clear distinction between the challenges and achievements in these tasks while using common metrics for evaluation.

Table 1. Comparison of Methods for Table Detection.

Method	Dataset	Precision (%)	Recall (%)	F1-Score (%)
Zheng et al. [11]	PubTabNet	94.1	93.3	93.7
Yang et al. [14]	SciTSR	92.1	90.4	91.2
He et al. [15]	TableBank	93.5	92.7	93.1
Wang et al. [16]	STDW	90.5	89.2	89.8
Qiao et al. [17]	PubTabNet	95.6	94.4	95.0

Table 2. Comparison of Methods for Table Structure Recognition.

Method	Dataset	Precision (%)	Recall (%)	F1-Score (%)
Xiao et al. [2]	TableBank	91.3	–	–
DeepDeSRT [18]	SciTSR	90.6	88.7	89.0
Kasar et al. [19]	ICDAR-2013	90.2	–	89.5
Raja et al. [20]	ICDAR-2019	92.7	91.3	92.0
Lin et al. [22]	PubTabNet, SciTSR	95.8	96.5	95.1

2.4. Research Gap

Despite considerable advancements in both table detection and structure recognition, several research challenges remain. Current methods struggle with tables that have highly irregular structures, require high computational resources, and do not generalize well across diverse document formats. Furthermore, many existing methods fail to handle tables with blank spaces or varying cell sizes effectively.

Deep learning models, while highly effective, often require substantial computational resources, making them less accessible for real-time or low-resource applications. This indicates a clear need for more efficient and adaptable models that can generalize across varying document formats while being computationally efficient.

Blank spaces and varying cell sizes present a major challenge for current table structure recognition models. Tables that contain many blank spaces or non-standard cell arrangements can lead to incomplete segmentation masks and hinder the overall accuracy of recognition systems.

Complex layouts involving multi-row and multi-column cell merging continue to challenge table detection and recognition systems. As such, future work should focus on developing models that are better able to handle these irregular layouts, while still maintaining efficiency.

Addressing these issues should be a key focus of the proposed method, aiming to fill the existing gap in robust table detection and structure recognition systems.

3. Methods

The proposed method, TableExtractNet, employs a multi-stage approach that integrates advanced deep learning architectures to detect and interpret table structures in document images. The initial step involves the application of a table detection tool to locate and extract tables from input images. These tables are then resized for optimal resolution and passed into a recognition system. This system utilizes convolutional neural networks (CNNs) and region-based CNNs (R-CNNs) to accurately identify and interpret the tabular data. The method is structured into four primary modules, each focusing on a different aspect of the detection and recognition processes.

After the table regions are detected, they are cropped from the original image (represented as ‘Cropped Tables’ in Figure 1). Cropping allows the system to focus on the identified table areas, excluding unnecessary document regions. Next, the recognized table structures are repositioned (represented as ‘Reposition Recognized Table Structures’) to ensure that the spatial arrangement of rows and columns is preserved within the cropped regions. This step is crucial for maintaining the structural integrity of the table data and ensuring accurate table recognition. Finally, the Tables block represents the fully processed tables, where both the table boundaries and internal structures (e.g., rows, columns, and cells) have been correctly identified and are ready for further analysis or extraction. Figure 1 illustrates the process of detecting and recognizing tables within unstructured documents. This study builds upon the methodologies presented in [23], which introduced a CNN-based approach for detecting and recognizing table structures in unstructured documents.

3.1. Table Detection

For table detection within documents, the CornerNet-FRCN Module has been developed, which combines two advanced models: CornerNet and Faster R-CNN. CornerNet is

a convolutional neural network specifically designed to identify objects by their corners. This method deviates from the traditional bounding box approach by utilizing key predictions from corner heatmaps and embeddings. These elements work in conjunction to precisely locate the top-left and bottom-right corners of objects, which, in this case, are tables. Such corner-based detection enhances the model's accuracy in localizing tables within images. Complementing CornerNet, the Faster R-CNN model builds upon the capabilities of the Fast R-CNN by incorporating Region Proposal Networks (RPNs). These networks distribute convolutional features across the entire image, facilitating the generation of region proposals with minimal additional computational cost. RPNs systematically suggest candidate object boundaries, assigning scores at each grid position on the feature map, thus streamlining the identification of potential table regions. The ResNet-18, a residual network with 18 layers, serves as the backbone for feature extraction in both models. ResNet-18 is crucial in training deep networks by overcoming the vanishing gradient problem through its innovative use of residual learning, allowing the seamless training of networks with many more layers than previously feasible. These skip connections are essential for the network's ability to learn complex features and nuances necessary for accurately detecting and classifying tables in document images.

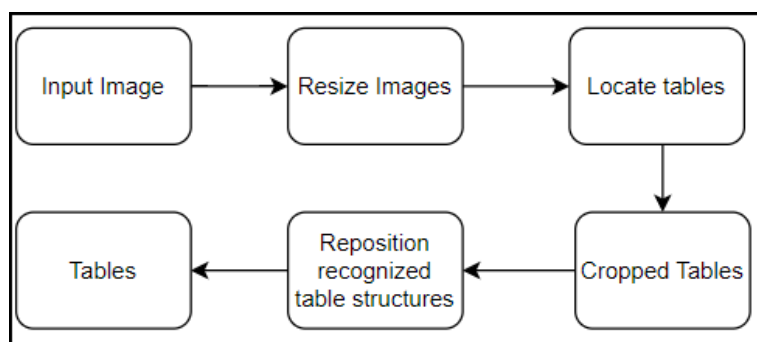


Figure 1. Overview of the table detection and structure recognition process. After the input image is resized, table regions are located and cropped ('Cropped Tables'). The recognized table structures are then repositioned ('Reposition Recognized Table Structures') to ensure correct alignment and organization before producing the final table output ('Tables').

As shown in Figure 2, the CornerNet-FRCN module represents the initial phase where CornerNet generates high-quality region proposals that are then processed by Faster R-CNN for accurate table detection. ResNet-18 acts as the feature extraction backbone, providing shared convolutional features to both CornerNet and Faster R-CNN components. We outline the following steps:

Input Description: The process begins by preparing the input images of documents. These images are resized to a standard dimension of $224 \times 224 \times 3$, representing width, height, and color channels, ensuring consistency for the deep learning models to process.

Feature Extraction with ResNet-18: By using the ResNet-18 architecture, the system extracts a hierarchy of features from the input images. ResNet-18, recognized for its residual learning framework, supports the training of deep networks by introducing skip connections that combat the vanishing gradient problem, enabling the capture of complex features necessary for accurate table detection.

Corner Detection via CornerNet: The CornerNet module identifies the corners (top-left and bottom-right) of tables within the images. Unlike traditional object detection methods that rely on bounding boxes, CornerNet emphasizes identifying corners using predictions from corner heatmaps and embeddings, which allows for more precise localization of tables.

Region Proposal and Refinement via Faster R-CNN: The regions suggested by CornerNet are further refined by Faster R-CNN. This involves filtering out non-table regions and improving the accuracy of the proposals. Faster R-CNN enhances the process by using

Region Proposal Networks (RPNs) that generate candidate object boundaries with scores, efficiently identifying potential table regions with minimal computational cost.

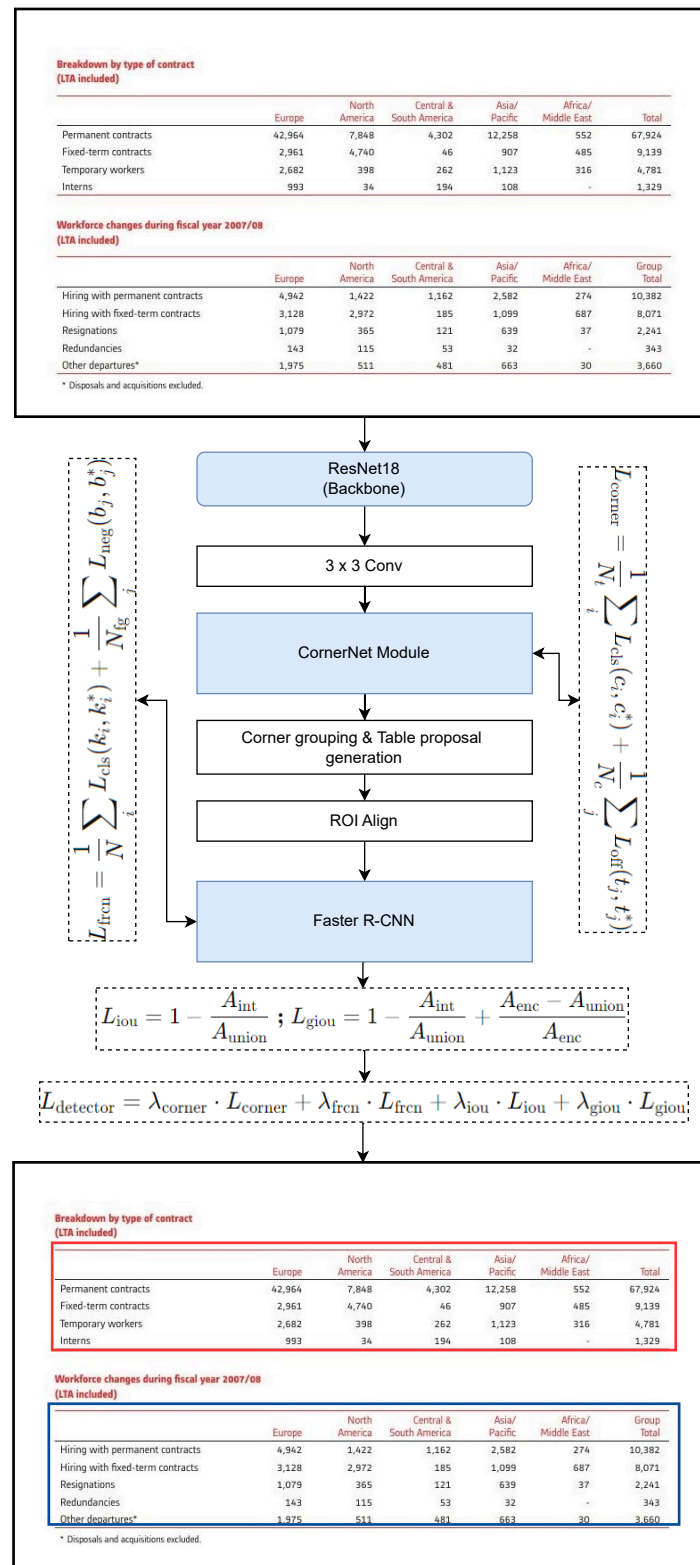


Figure 2. Table detection diagram.

Loss Functions Application: The model employs a combination of loss functions, including corner loss L_{corner} , R-CNN loss L_{frcn} , which are combined into a total detection

loss L_{detector} , to train and optimize the detection process. These loss functions evaluate the model's performance in predicting corner points and refining region proposals, guiding the training process towards higher accuracy.

Output: The output from the table detection process is a set of bounding boxes that indicate the locations of tables within an input image. Each bounding box is defined by the coordinates of its top-left and bottom-right corners.

- **Output Shape:**
The bounding box for each detected table is represented as:

$$[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$$

where $(x_{\min}, y_{\min})$ are the coordinates of the top-left corner, and $(x_{\max}, y_{\max})$ are the coordinates of the bottom-right corner of the table, as shown in Figure 3.

For multiple tables, the coordinate system can be expressed as:

$$[x_{\min1}, y_{\min1}, x_{\max1}, y_{\max1}], [x_{\min2}, y_{\min2}, x_{\max2}, y_{\max2}]$$

indicating the bounding box locations for each detected table.

If two tables are detected, the output could be:

$$[[x_{\min1}, y_{\min1}, x_{\max1}, y_{\max1}], [x_{\min2}, y_{\min2}, x_{\max2}, y_{\max2}]]$$

This defines the spatial boundaries of each table using the bounding box coordinates $[x_{\min1}, y_{\min1}, x_{\max1}, y_{\max1}]$ and $[x_{\min2}, y_{\min2}, x_{\max2}, y_{\max2}]$, ensuring accurate localization for subsequent table recognition processes.

30. Schulden

Die Restlaufzeiten sind im Schuldenspiegel im Einzelnen dargestellt:

($x_{\min1}, y_{\min1}$)

	Stichtag	Restlaufzeit bis 1 Jahr	Restlaufzeit 1-5 Jahre	Restlaufzeit über 5 Jahre	Gesamt
Wandelschuldverschreibungen	31.12.2009 31.12.2008	4.000 0	12.000 24.000	0 0	16.000 24.000
Schulden aus Lieferungen und Leistungen	31.12.2009 31.12.2008	685.546 226.430	0 0	0 0	685.546 226.430
Kurzfristige abgegrenzte Schulden	31.12.2009 31.12.2008	3.735.599 3.045.828	0 0	0 0	3.735.599 3.045.828
Umsatzabgrenzungsposten	31.12.2009 31.12.2008	3.204.066 1.485.910	0 0	0 0	3.204.066 1.485.910
Steuerrückstellungen	31.12.2009 31.12.2008	100.129 269.421	0 0	0 0	100.129 269.421
Sonstige kurzfristige Schulden	31.12.2009 31.12.2008	611.371 520.855	0 0	0 0	611.371 520.855
Gesamt	31.12.2009 31.12.2008	8.340.711 5.548.444	12.000 24.000	0 0	8.352.711 5.572.444

($x_{\max1}, y_{\max1}$)

Schulden aus Lieferungen und Leistungen sowie sonstige kurzfristige Schulden sind nicht verzinslich.

31. Kreditlinien

Es bestehen nicht gesicherte Kreditlinien für Kontokorrentkredite in Höhe von Mio. 0,56 (Vorjahr: Mio. EUR 0,7) bei den Hausbanken der einbezogenen Unternehmen. Kreditaufnahmen (Kontokorrentkredite) im Rahmen dieser Abkommen werden mit bis 5,65 Prozent (Vorjahr: 7,95 Prozent) verzinst. Es bestehen wie im Vorjahr keine Verbindlichkeiten gegenüber Kreditinstituten.

32. Kurzfristige abgegrenzte Schulden

Die kurzfristigen abgegrenzten Schulden enthalten im Wesentlichen folgende Beträge:

($x_{\min2}, y_{\min2}$)

	31.12.2008 EUR	Inanspruchnahme	Auflösung	Zuführung	31.12.2009 EUR
Gehalts- und Provisionsrückstellungen	2.568.945	1.535.879	459.652	2.370.220	2.943.634
Erwartete Abrechnungen	177.315	72.817	9.733	279.832	374.597
Sonstige Personalarückstellungen	0	0	0	90.000	90.000
Sonstige Rückstellungen	299.568	167.274	23.363	218.437	327.368
Gesamt	3.045.828	1.775.970	492.748	2.958.489	3.735.599

($x_{\max2}, y_{\max2}$)

Die Gehalts- und Provisionsrückstellungen beinhalten Ansprüche aus variablen Gehaltsbestandteilen, welche im Geschäftsjahr entstanden sind und erst im Folgejahr ausbezahlt werden. Die erwarteten Abrechnungen betreffen erhaltene Leistungen, deren Abrechnung zum Stichtag aussteht. Die sonstigen Rückstellungen enthalten im Wesentlichen Rückstellungen für Abschluss- und Prüfungskosten sowie eine Rückstellung für Rückbauverpflichtungen.

Figure 3. Example showing how the bounding box of a table is defined using coordinates to specify the top-left and bottom-right corners of the detected table.

Building upon the foundation established by the CornerNet-FRCN Module, the Table Detection Architecture uses a two-stage detection approach complemented by Feature Pyramid Networks (FPNs). Initially, the architecture engages in the generation of region proposals through CornerNet, which marks the first stage of detection. These proposals are then meticulously classified and refined through bounding box regression by Faster R-CNN in the second stage. This sequential processing ensures that only the most accurate representations of tables are carried forward.

To address the variability in table sizes within documents, the architecture incorporates Feature Pyramid Networks (FPNs). FPNs are crucial for managing objects at different scales, enhancing the capability of convolutional networks. By integrating a top-down pathway with lateral connections, the FPNs construct a sophisticated, multi-scale feature pyramid from the input image. This design allows the system to maintain detection performance across various table dimensions, a common challenge in document analysis.

Thus, the architecture not only identifies tables with precision but also adapts to the diverse sizes that tables can present. As described in [4], the detection is performed as follows:

- **Stage 1: CornerNet:** Acts as a region proposal network. It uses a deep convolutional neural network to predict heatmaps for the top-left and bottom-right corners of tables. These corners are then paired using an embedding technique that helps in grouping corners belonging to the same table. The network utilizes a novel corner pooling mechanism to capture explicit boundary information and predicts offsets to compensate for any discretization errors.
- **Stage 2: Fast R-CNN:** Refines the proposals and performs the final detection. After the proposals are generated by CornerNet, Faster R-CNN refines them by classifying each proposal as a table or non-table and adjusting the bounding box coordinates for precise localization.

The table detection model builds upon the initial module by enhancing the precision of the table localization.

In-depth Analysis of the CornerNet Framework:

- **Sub-Stages for Detection:** The model operates in two sub-stages for detecting top-left and bottom-right corners separately.
- **Dilated Convolutions:** These are employed to capture spatial hierarchies and broader context without losing resolution.
- **Corner Pooling:** A unique mechanism that ensures that the corner predictions are robust and accurate, even with complex table layouts.

Refinements Within the Faster R-CNN Paradigm: Ensures precise alignment of the region of interest, regardless of scaling or translation.

- **ROI Align:** This process ensures that features extracted from each proposed region are accurately aligned with the region of interest, improving the quality of the subsequent classification and bounding box regression.
- **Classification and Bounding Box Regression:** Determines the presence of a table within each region proposal and fine-tunes the coordinates of the bounding boxes to snugly enclose the tables.

3.2. Table Structure Recognition (TSR)

For recognizing the structure of tables within documents, we propose the Split-and-Merge Module. This module is crucial for Table structure recognition, as it combines features from different levels within the neural network. Such fusion allows the model to gain a detailed yet broad view of the data. This is especially useful when dealing with tables of different sizes and complexities.

Alongside this, we utilize Spatial and Grid CNNs. Spatial CNNs help in understanding and maintaining the spatial relationships between different parts of tables. They ensure that the layout of the table is recognized and kept intact. Grid CNNs are particularly adept

at handling tables since tables are essentially grids. They manage the unevenly spaced or oddly shaped parts of the tables, which are common. Together, these technologies provide a comprehensive way to recognize and reconstruct the complex structure of tables in our documents.

As illustrated in Figure 4, it employs a Feature Pyramid Network (FPN) + ResNet-18: A design that uses a top-down framework with lateral links to construct a feature pyramid from a single-scale input.

The process of recognizing table structures involves several critical stages, each utilizing specific convolutional neural network architectures to decipher the intricate organization of tables in document images.

Input Description: Before beginning the table structure recognition process, the detected tables are provided as input in an organized format suitable for processing. Each table is represented as an image cropped from the original document based on the bounding boxes identified during the table detection phase. These images are then resized to a standardized dimension of $224 \times 224 \times 3$, determined by the requirements of the neural network, to ensure uniformity and optimal recognition performance by the subsequent CNN architectures. This dimensionality corresponds to the width, height, and color channels of the images, respectively, allowing for a comprehensive representation of the tables in a format conducive to deep learning models. The standardized input format enables the system to consistently extract features across various table representations, facilitating precise segmentation and recognition of table structures.

Feature Combination with FPN and ResNet-18: For recognizing table structures, the method leverages a combination of Feature Pyramid Networks (FPN) and ResNet-18. This combination enables the extraction of multi-scale features from the detected tables, essential for recognizing complex structures with varying shapes and gaps.

Cell Decomposition with Spatial CNN: Utilizing Spatial CNN, the system segments the tables into individual cells. This network is particularly adept at processing spatial data, maintaining the spatial relationships between parts of the table, and ensuring the layout's integrity is preserved.

Cell Merging with Grid CNN: The segmented cells are then recombined through the Grid CNN. This specialized network handles irregularities in cell shapes and spacing, effectively managing the grid-like structure of tables and ensuring the correct reconstruction of the table layout.

Loss Functions for Structure Recognition: The recognition process is refined using specific loss functions tailored for different stages. For the decomposition of table cells, a 'split loss' (L_{split}) is applied, which aids in segmenting the table into individual cells. Concurrently, a 'merge loss' (L_{merge}) distinct from the split loss is employed during the recombination phase, where it assists in accurately reassembling the cells into a coherent table structure. These specialized loss functions together constitute the overall recognition loss ($L_{\text{recognizer}}$), guiding the model towards a precise reconstruction of table layouts from the document images. These functions evaluate the system's performance in segmenting and reassembling the table structure, optimizing the process toward accurate structure recognition.

Output: The TSR output involves reconstructing the internal structure of each detected table, detailing the rows, columns, and individual cells.

- **Output Representation:** The TSR output can be visualized as a matrix where each element corresponds to a cell in the table. The cells are defined by their bounding boxes and possibly their content.
- **Output Shape:** For a table with M rows and N columns, the TSR output shape could be represented as a $M \times N$ matrix.
- **Output Example:** For a simple table shown in Figure 5 with 7 rows and 3 columns, the TSR output could be represented as:

```
[
  [Cell(0, 0, "SERVICE STATION LOCATION"),
   Cell(0, 1, "TOTAL GALLONS OF GASOLINE PUMPED (THOUSANDS)" ),
   Cell(0, 2, "AVERAGE DAILY TRAFFIC COUNT AT LOCATION (HUNDREDS)"]],
  [Cell(1, 0, "E. Regular Street"), Cell(1, 1, "100"), Cell(1, 2, "3")],
  [Cell(2, 0, "S. Lead Street"), Cell(2, 1, "112"), Cell(2, 2, "4")],
  [Cell(3, 0, "N. Main Street"), Cell(3, 1, "150"), Cell(3, 2, "5")],
  [Cell(4, 0, "Highway 606"), Cell(4, 1, "210"), Cell(4, 2, "7")],
  [Cell(5, 0, "Baker Boulevard"), Cell(5, 1, "60"), Cell(5, 2, "2")],
  [Cell(6, 0, "E. High Street"), Cell(6, 1, "85"), Cell(6, 2, "3")],
  [Cell(7, 0, "Country Club Road"), Cell(7, 1, "77"), Cell(7, 2, "2")]
]
```

Each `Cell(i, j, content)` might include the coordinates (i, j) of the cell within the table and the text content of the cell. The coordinates $[x_{\min}, y_{\min}, x_{\max}, y_{\max}]$, would typically represent the position and size of each cell in the image.

In our architecture for table structure recognition, the Separator REgression TRansformer (SepRETR) module plays a pivotal role in addressing the challenge of discerning rows and columns within tables. The ability to separate these elements is not only fundamental for capturing the table layout but also essential for the subsequent phases of cell recognition and content extraction. SepRETR adopts a dual-processing approach, treating rows and columns as separate entities, which greatly streamlines the complex task of delineating table structures.

To refine the performance of SepRETR, and indeed the entire recognition system, we employ a suite of comprehensive loss functions. These functions serve as critical indicators of the model's accuracy and effectiveness in its designated tasks. By tailoring specific loss functions to the particular goals of each module such as corner detection loss for CornerNet, R-CNN loss for bounding box refinement, split loss for the division of table cells, and merge loss for the reconstitution of cell structure, we equip the model with the necessary feedback mechanisms to learn and adapt. This multi-faceted feedback ensures that the model not only recognizes the various features and spatial relationships within the tables but also improves its accuracy with each iteration of training.

Building on the method presented in [4], as illustrated in Figure 6, we have incorporated the SepRETR module, an innovative neural network designed to tackle the intricate tasks of row and column recognition within tables. This final module of our system concentrates on meticulously identifying the table's rows and columns, and on extracting detailed information for each cell.

The process begins with Input Processing, where the module works with the same input dimensions as the earlier stages, ensuring consistency across the entire recognition system. Then, during Feature Extraction, the module employs the ResNet-18 in combination with Feature Pyramid Networks (FPN) as its backbone. This powerful duo extracts strong and reliable features from the input image, which are crucial for the accurate recognition of table structures.

The Row and Column Separation task is where SepRETR truly shines. It takes the robust features extracted by ResNet-18 and FPN and identifies the lines that separate individual rows and columns. This is a delicate process, as it must discern the fine boundaries that define the structure of the table.

Finally, the Output Generation phase produces a well-structured representation of the table, marking the detected rows (N) and columns (M), and presenting a detailed cell-by-cell composition of the table's contents. This structured output is essential for downstream tasks such as data extraction and table interpretation, enabling the seamless conversion of table data into a machine-readable format.

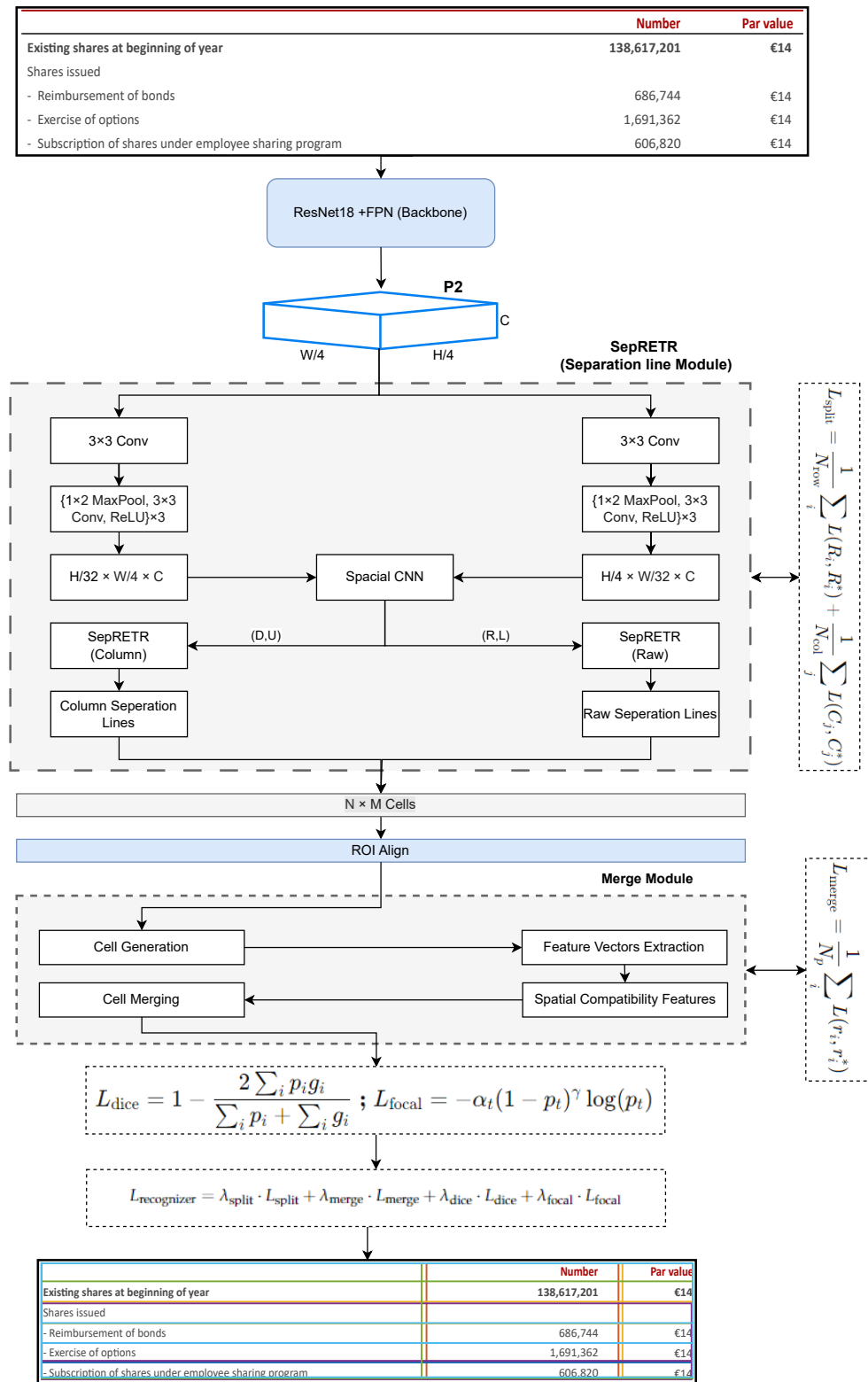


Figure 4. The Table Structure Recognition process, detailing the Split and Merge Module, utilizes ResNet-18 and FPN for feature extraction and Spatial CNNs for row and column delineation, culminating in the computation of L_{split} and L_{merge} losses.

Input Image			Output Image		
SERVICE STATION LOCATION	TOTAL GALLONS OF GASOLINE PUMPED (THOUSANDS)	AVERAGE DAILY TRAFFIC COUNT AT LOCATION (HUNDREDS)	SERVICE STATION LOCATION	TOTAL GALLONS OF GASOLINE PUMPED (THOUSANDS)	AVERAGE DAILY TRAFFIC COUNT AT LOCATION (HUNDREDS)
E. Regular Street	100	3	E. Regular Street	100	3
S. Lead Street	112	4	S. Lead Street	112	4
N. Main Street	150	5	N. Main Street	150	5
Highway 606	210	7	Highway 606	210	7
Baker Boulevard	60	2	Baker Boulevard	60	2
E. High Street	85	3	E. High Street	85	3
Country Club Road	77	2	Country Club Road	77	2

Figure 5. Output Example.

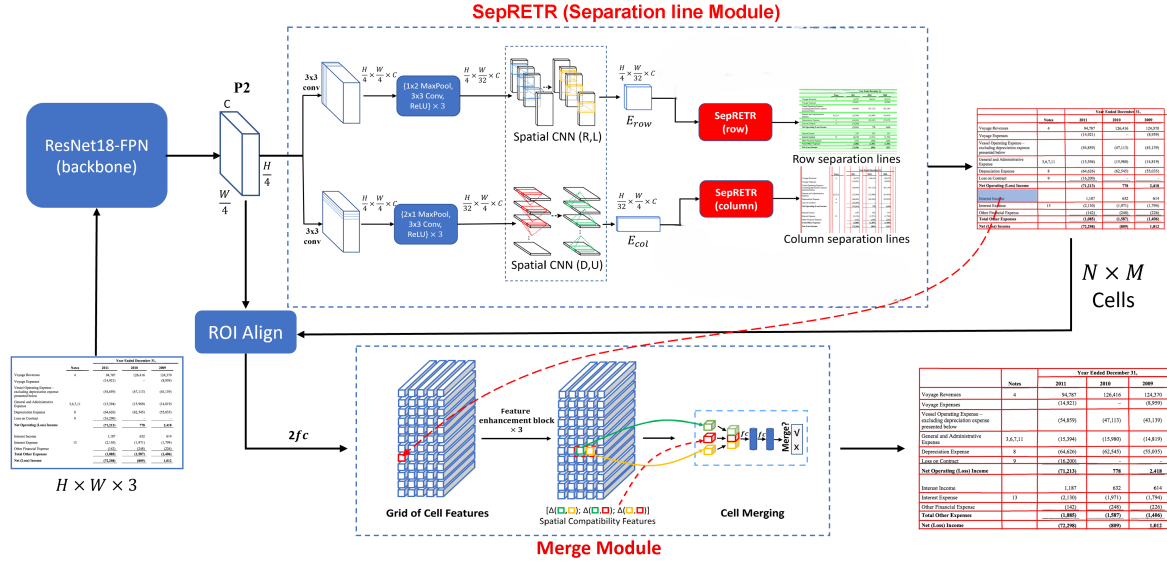


Figure 6. Table Structure Recognition model architecture based on the method given in [4].

3.3. Loss Function

1. Table Detection: For the table detection framework, the loss function is defined in two parts. The first part, L_{corner} , is a combination of classification and regression losses. The classification loss evaluates the predicted corner points against the ground truth, normalized by the number of tables, N_t , and the regression loss refines the position of these points, normalized by the number of corners per batch, N_c . The second part, L_{frcn} , is determined by the classification of region proposals as table or non-table, divided by the number of proposals, N , and the regression of the coordinates for these proposals, normalized by the number of foreground proposals, N_{fg} . The overall loss for detecting tables is a weighted sum of these two components. To improve the model's performance beyond the original framework in [4], we propose adding Intersection over Union (IoU) loss and Generalized Intersection over Union (GIoU) loss to enhance bounding box regression and localization. These additional losses help in accurately localizing the table boundaries by addressing the issues of overlapping and non-overlapping bounding boxes.

The loss function for the corner detection, L_{corner} , is given by:

$$L_{\text{corner}} = \frac{1}{N_t} \sum_i L_{\text{cls}}(c_i, c_i^+) + \frac{1}{N_c} \sum_j L_{\text{off}}(t_j, t_j^+) \quad (1)$$

The loss for the Faster R-CNN module, L_{frcn} , is defined as:

$$L_{\text{frcn}} = \frac{1}{N} \sum_i L_{\text{cls}}(k_i, k_i^+) + \frac{1}{N_{\text{fg}}} \sum_j L_{\text{reg}}(b_j, b_j^*) \quad (2)$$

To further improve the model's performance, we include the following loss functions:

$$L_{iou} = 1 - \frac{A_{int}}{A_{union}} \quad (3)$$

$$L_{giou} = 1 - \frac{A_{int}}{A_{union}} + \frac{A_{enc} - A_{union}}{A_{enc}} \quad (4)$$

Explanation of Variables:

- A_{int} : The area of intersection between the predicted bounding box and the ground truth bounding box. This represents the overlapping region of the two boxes.
- A_{union} : The area of union, which is the total area covered by both the predicted and ground truth bounding boxes. It includes the overlapping and non-overlapping areas.
- A_{enc} : The area of the enclosing box, which is the area of the smallest box that can completely enclose both the predicted and ground truth bounding boxes.

The improved total loss for the table detector is:

$$L_{detector} = \lambda_{corner} \cdot L_{corner} + \lambda_{frcn} \cdot L_{frcn} + \lambda_{iou} \cdot L_{iou} + \lambda_{giou} \cdot L_{giou} \quad (5)$$

where the weighting values are:

- $\lambda_{corner} = 1.0$
- $\lambda_{frcn} = 1.0$
- $\lambda_{iou} = 1.0$
- $\lambda_{giou} = 0.5$

The chosen values for λ are based on empirical tuning during model training. Higher values for λ_{corner} , λ_{frcn} , and λ_{iou} reflect the importance of accurate corner detection, region proposal classification, and bounding box localization, respectively. A smaller value for λ_{giou} is used to balance the effect of this term, as GIoU provides a regularization factor for non-overlapping bounding boxes but has a less direct impact on overlap.

Hyperparameter Tuning and the Effects of Poor Choices: The chosen λ values are hyperparameters that have been optimized through extensive experimentation. Poorly chosen values for λ can have significant negative effects on the model's performance. For instance:

- **Overemphasis on IoU or GIoU:** If λ_{iou} or λ_{giou} is set too high, the model may focus excessively on bounding box overlap, which can lead to slower convergence or difficulty in distinguishing closely packed tables.
- **Imbalanced Loss Contributions:** If the values of λ_{corner} or λ_{frcn} are too small, the model may underperform in identifying table corners or classifying region proposals, resulting in lower detection accuracy.
- **Difficulty in Training:** Improperly chosen λ values can also make the model harder to train, as certain parts of the loss function may dominate the training process, leading to instability or slow convergence.

To mitigate these issues, a grid search was conducted to find an optimal balance between the different loss components. The selected values represent the best trade-off between accurate table detection and fast model convergence.

2. **Table Structure Recognition:** For Spatial CNN-Based Separation Line Prediction in Table Structure Recognition, the process is divided into two separate pathways: one designated for the separation of rows and the other for columns, with each pathway having its unique loss calculation. The overall loss for the component is determined by taking the average of the losses across the sampled pixel count for rows N_{row} and columns N_{col} . The labels for the row and column delineation predicted by the model and those that are actual (ground truth) are denoted by R_i, C_j and R_i^*, C_j^* respectively. The loss function $L(R_i, R_i^*)$ calculates the binary cross-entropy between the predicted probability for a row separator at each sampling pixel and its ground truth label. To

improve segmentation accuracy and handle class imbalance, we propose incorporating Dice loss and Focal loss. These additional losses help in accurately segmenting the table structures and addressing the issues of class imbalance in the training data. The specific loss function for predicting separation lines is formulated accordingly:

$$L_{\text{split}} = \frac{1}{N_{\text{row}}} \sum_i L(R_i, R_i^*) + \frac{1}{N_{\text{col}}} \sum_j L(C_j, C_j^*) \quad (6)$$

Grid CNN-Based Cell Merging: Here, N_p denotes the number of relational pairs selected for merging cells. $L(r_i, r_i^*)$ is the binary cross-entropy loss between the predicted and ground truth labels for the i -th relational pair, reflecting the model's accuracy in predicting whether two cells should be merged:

$$L(r_i, r_i^*) = -(r_i^* \log(r_i) + (1 - r_i^*) \log(1 - r_i)) \quad (7)$$

The loss for the cell merging module is thus given by:

$$L_{\text{merge}} = \frac{1}{N_p} \sum_i L(r_i, r_i^*) \quad (8)$$

The definitions for additional losses are:

$$L_{\text{dice}} = 1 - \frac{2 \sum p_i g_i}{\sum p_i + \sum g_i} \quad (9)$$

$$L_{\text{focal}} = -\alpha_t (1 - p_t)^\gamma \log(p_t) \quad (10)$$

where p_i and g_i are the predicted and ground truth values for a pixel, respectively. α_t is a weighting factor, $(1 - p_t)^\gamma$ is a modulating factor to focus on hard-to-classify examples, and p_t is the predicted probability for class t .

Total Loss for Table Structure Recognizer:

$$L_{\text{recognizer}} = \lambda_{\text{split}} \cdot L_{\text{split}} + \lambda_{\text{merge}} \cdot L_{\text{merge}} + \lambda_{\text{dice}} \cdot L_{\text{dice}} + \lambda_{\text{focal}} \cdot L_{\text{focal}} \quad (11)$$

The weighting values are:

- $\lambda_{\text{split}} = 1.0$
- $\lambda_{\text{merge}} = 1.0$
- $\lambda_{\text{dice}} = 0.7$
- $\lambda_{\text{focal}} = 0.5$

These λ values were selected based on experiments during model training. A higher weight on λ_{split} and λ_{merge} indicates the importance of predicting separation lines and correctly merging cells. Lower values for λ_{dice} and λ_{focal} reflect their role as regularization terms for segmentation accuracy and addressing class imbalance.

Hyperparameter Tuning and Sensitivity: Similar to table detection, the λ values for Table Structure Recognition were fine-tuned through a series of experiments. Poor choices for these values can negatively impact the system:

- **Overemphasis on Focal Loss:** Setting λ_{focal} too high may lead the model to focus too much on difficult examples, causing slower convergence.
- **Unbalanced Loss Weights:** Poor tuning of λ_{split} or λ_{merge} may cause the system to inaccurately delineate rows and columns, leading to poor table structure recognition.

We performed extensive tuning through grid search to ensure that the loss components are balanced and the model performs well on both row/column separation and cell merging tasks.

The loss functions can provide further insight into their specific roles within the machine learning framework. The corner detection loss function, L_{corner} , fundamentally

ensures the precision of the initial table boundary identification, which is crucial for the subsequent stages of table structure recognition. It fine-tunes the algorithm's ability to pinpoint the exact location of the table corners within a diverse array of document formats. Similarly, L_{rcnn} loss function, associated with the Faster R-CNN model, refines the region proposals to accurately classify and localize the tables. It is integral to eliminating false positives and ensuring that only valid table regions are passed on for further analysis.

The L_{iou} and L_{giou} losses are specifically introduced to address the bounding box regression challenges. L_{iou} measures the overlap between predicted and ground truth bounding boxes, promoting better localization. L_{giou} further extends IoU by incorporating the distance between the predicted and ground truth boxes, thereby improving the regression performance for non-overlapping boxes.

The L_{split} and L_{merge} losses are tailored for the unique challenges posed by table structure recognition. They work in concert to distinguish between individual cells and merge them appropriately, effectively reconstructing the original table layout from the document image. This is particularly important when dealing with complex table structures that include merged cells or varying row and column spans.

To further enhance segmentation accuracy and handle class imbalance, L_{dice} and L_{focal} are incorporated. L_{dice} improves the accuracy of cell boundary predictions by focusing on the overlap between predicted and ground truth segments. L_{focal} addresses the issue of class imbalance by focusing more on hard-to-classify examples, thereby improving the robustness of the model.

The combination of these loss functions, as encapsulated in $L_{\text{recognizer}}$, embodies the holistic approach required for the robust and accurate digitization of tabular information from scanned documents. These improvements over the original framework in [4] provide a significant enhancement in both detection and structure recognition tasks.

4. Experimental Results and Discussion

Experiments were conducted in two main areas: table detection and table structure recognition. Each area followed its own testing protocol and evaluation standards. The results were calculated using key metrics such as precision, recall, F1-score, and average precision [24].

These parameters are essential for evaluating the performance of a classification model. They are defined as follows:

True Positive (TP): Correct identification of a ground-truth bounding box.

False Positive (FP): Incorrect detection of a non-existent object or incorrect placement of detection on an existing object.

False Negative (FN): Failure to detect an existing ground-truth bounding box.

Metrics are calculated as follows:

- **Precision (P)** measures the proportion of true positive predictions among all positive predictions made. It is calculated using the formula:

$$P = \frac{TP}{TP + FP} = \frac{TP}{\text{All Detections}} \quad (12)$$

- **Recall (R)** represents the proportion of true positive predictions relative to the total number of actual positive instances. It is calculated as:

$$R = \frac{TP}{TP + FN} = \frac{TP}{\text{All Ground-truths}} \quad (13)$$

- **F1-score (F1):** Precision and recall are balanced by taking the harmonic mean of the two measures. The formula is as follows:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{P \times R}{P + R} \quad (14)$$

- **Average Precision (AP)** calculates the weighted mean of precisions at each threshold, summarizing the precision-recall curve. The weight assigned to each threshold represents the change in recall from the previous level. (AP) is defined as:

$$AP = \sum_n (R_{n+1} - R_n) P_{interp}(R_{n+1}), \quad (15)$$

where the interpolated precision $P_{interp}(R_{n+1})$ at a recall level R_{n+1} is the maximum precision observed for any recall $\hat{R}$ greater than or equal to R_{n+1} :

$$P_{interp}(R_{n+1}) = \max_{\hat{R}: \hat{R} \geq R_{n+1}} P(\hat{R}). \quad (16)$$

This approach allows AP to be calculated by interpolating the precision at each recall level, rather than relying solely on a fixed set of 11 evenly spaced points. This method provides a more accurate and fine-grained assessment of the model's performance across all thresholds.

4.1. Implementation Details

The training and evaluation of the *TableExtractNet* model were performed using both CPU and GPU configurations. The CPU used was a 13th Gen Intel(R) Core(TM) i7-1355U @ 1.70 GHz with 16 GB RAM, and the GPU used was an NVIDIA Tesla V100 with 32 GB memory.

Training was conducted for 16 epochs across four datasets. On the IIIT-AR-13K dataset, each epoch on the CPU took approximately 30 min, resulting in a total training time of 8 h. On the GPU, each epoch took approximately 8 min, leading to a total training time of 2 h. Similarly, for the STDW dataset, each epoch took around 30 min on the CPU, with a total training time of 8 h, and around 8 min on the GPU, with a total training time of 2 h.

For the larger SciTSR dataset, each epoch took approximately 1 h on the CPU, resulting in a total training time of 16 h. On the GPU, each epoch took around 15 min, leading to a total training time of 4 h. The same pattern was observed for the PubTabNet dataset, where each epoch took 1 h on the CPU for a total training time of 16 h, and 15 min on the GPU for a total training time of 4 h.

The significant reduction in training time using the GPU across all datasets highlights the advantages of leveraging specialized hardware for faster model development and experimentation.

4.2. Table Detection

The proposed model, *TableExtractNet*, for table detection, integrates the CornerNet-FRCN Module and was evaluated using the IIIT-AR-13K [25] dataset, a widely recognized benchmark in document analysis. The training set comprised 9333 images, validated against 1955 images, and tested on 2120 images. The model's performance was assessed using the PASCAL VOC [26] evaluation metric, a well-established standard for object detection frameworks.

To further demonstrate the model's robustness and generalizability, we also evaluated its performance on the STDW [27] dataset, which is a diverse large-scale dataset for table detection containing 7470 samples. The dataset is divided into 5970 training images and 1500 test images, and includes a wide variety of table structures collected from diverse sources, making it an ideal benchmark for testing real-world use cases.

In Table 3, results achieved are compared with state-of-the-art methods, revealing that our model, *TableExtractNet*, achieved a precision (P) of 98.8%, a recall (R) of 98.4%, an F1 score of 98.7%, and an average precision (AP) of 98.4% on the IIIT-AR-13K dataset, and a precision of 96.4%, recall of 96.6%, F1 score of 97.5%, and average precision of 96.8% on the STDW dataset. These results significantly surpass those achieved by previous models, highlighting the effectiveness of our approach. The use of a ResNet-18 backbone, in particular, contributed to the substantial improvement in detection performance.

Table 3. Table detection performance. The best result for each metric is written in bold. comparison on IIIT-AR-13K and STDW datasets.

Methods	Backbone	Dataset	P (%)	R (%)	F1 (%)	AP (%)
CornerNet+FRCN [4]	ResNet-18	IIIT-AR-13K	98.6	98.3	98.5	98.2
Faster R-CNN [22]	ResNet-101	STDW	92.6	90.5	91.5	94.0
Mask R-CNN [22]	ResNet-101	STDW	95.2	94.6	93.0	96.6
Faster R-CNN [25]	ResNet-101	IIIT-AR-13K	95.7	92.6	94.2	95.5
Mask R-CNN [25]	ResNet-101	IIIT-AR-13K	98.2	96.6	97.4	97.6
RetinaNet [27]	ResNet-50	STDW	93.9	91.9	92.9	95.3
Selective Search [28]	–	STDW	91.8	89.3	90.5	93.9
Ours	ResNet-18	IIIT-AR-13K	98.8	98.4	98.7	98.4
Ours	ResNet-18	STDW	96.4	96.6	97.5	96.8

4.3. Table Structure Recognition

For the evaluation of Table Structure Recognition, the SciTSR [29] dataset was used. This dataset includes more complex table structures, featuring a subset of 12,706 challenging tables specifically designed to test the model's recognition capabilities. With a training set of 10,000 images and a testing set of 3000 images, the model underwent rigorous evaluation for its precision in structuring table data.

Also, we evaluated the model's performance on the PubTabNet [21] dataset, which contains 500,777 training images, 9115 validation images, and 9138 testing images, all of which are axis-aligned. PubTabNet is designed specifically for evaluating table structure recognition tasks, and its size and diversity make it a valuable resource for testing generalizability.

Results presented in Table 4, show that our model, *TableExtractNet*, outperforms existing methods, achieving a precision of 99.6%, a recall of 99.3%, and an F1 score of 99.2% on the SciTSR dataset, and a precision of 97.6%, recall of 97.1%, and F1 score of 97.4% on the PubTabNet dataset. These results demonstrate significant improvements over methods like DeepDeSRT, TabStruct-Net, and Tabby, emphasizing the model's robustness in handling complex table structures and varied document formats.

These results underscore the model's effectiveness in managing a wide range of table configurations, which is crucial for automating table data extraction from unstructured documents. The integration of CornerNet and Fast R-CNN for detection, combined with the innovative Split-and-Merge module for structure recognition, provides a comprehensive solution to challenges in the field. The proposed method not only excels in terms of higher metric scores but also proves valuable in practical applications where diverse table structures are common. Overall, the experiments validate the proposed model as a highly accurate and reliable tool for table detection and structure recognition, offering significant potential for processing and analyzing unstructured data.

Table 4. TSR performance comparison. The best result for each metric is written in bold. on SciTSR and PubTabNet datasets.

Methods	Dataset	P (%)	R (%)	F1 (%)
Split-Merge [4]	PubTabNet	95.3	94.0	94.6
Split-Merge [4]	SciTSR	99.4	99.1	99.0
Tabby [7]	SciTSR	92.6	92.0	92.1
GTE [11]	PubTabNet	91.2	89.7	90.4
LGPMA [17]	PubTabNet	90.6	90.0	90.3
DeepDeSRT [18]	SciTSR	90.6	88.7	89.0
TabStruct-Net [20]	SciTSR	92.7	91.3	92.0
TabStruct-Net [20]	PubTabNet	95.3	94.0	94.6
EDD [22]	PubTabNet	89.0	86.3	87.6
Ours	SciTSR	99.6	99.3	99.2
Ours	PubTabNet	97.6	97.1	97.4

4.4. Limitations

While the *TableExtractNet* model demonstrates strong performance in detecting and recognizing a wide variety of table structures, analyzing errors and handling complex cases remains essential for optimizing the model's real-world applicability.

4.4.1. Error Analysis

Through detailed error analysis, we identified that tables with highly irregular layouts, such as those with invisible or faint borders, were occasionally missed or misclassified. Additionally, non-table elements with grid-like features, such as charts or background patterns, sometimes triggered false positives. By addressing these errors through enhanced feature extraction and refining detection thresholds, the model's precision can be further improved for real-world applications.

4.4.2. Handling Complex Cases

Complex table structures, such as those with merged cells, irregular row or column spans, or those appearing in multi-page documents, posed additional challenges. Our analysis revealed that handling these cases requires more sophisticated boundary recognition algorithms and post-processing steps. Future work will focus on developing specialized modules to handle merged cells and irregular layouts, enhancing the model's adaptability to a broader range of document types.

Further work could explore these aspects, with the goal of enhancing the model's generalizability to a wider range of document types and optimizing its computational efficiency.

4.5. Ablation Studies

To evaluate the individual contributions of key components in the *TableExtractNet* model, we conducted ablation studies by systematically removing or altering specific parts of the model. We specifically highlight the impact of the newly introduced loss functions, such as IoU and GIoU for table detection, and Dice and Focal losses for table structure recognition, alongside the base components CornerNet and Faster R-CNN for table detection, and Spatial CNN, Grid CNN, and Feature Pyramid Networks (FPN) for table structure recognition. The results were tested on the IIIT-AR-13K and STDW datasets for table detection, and the SciTSR and PubTabNet datasets for table structure recognition.

4.5.1. Table Detection with CornerNet + Faster R-CNN Module

The *TableExtractNet* model builds on the CornerNet and Faster R-CNN modules for table detection by incorporating IoU and GIoU losses to improve bounding box localization. The ablation study assesses the individual contributions of these components to demonstrate the enhancements they bring.

Ablation on Loss Functions for Table Detection

- Without IoU and GIoU Losses
 - On the **IIIT-AR-13K** dataset: Precision decreased by 2.5%, recall by 3.8%, and F1-score by 3.1%. The model's ability to localize table boundaries, especially for overlapping tables, was significantly reduced.
 - On the **STDW** dataset: Precision decreased by 3.1%, recall by 4.2%, and F1-score by 3.6%. Complex table layouts with small, irregular tables were less accurately detected without the improvements from IoU and GIoU.
- **Loss Weighting (λ) Impact:**
 - When lowering $\lambda_{\text{iou}} = 0.5$ and $\lambda_{\text{giou}} = 0.3$, the model's performance degraded significantly in terms of bounding box overlap, indicating that these components are essential for accurate table localization.

Ablation on CornerNet and Faster R-CNN

- **Removing CornerNet:**
 - On the **IIIT-AR-13K** dataset: Precision decreased by 4.8%, recall by 4.1%, and F1-score by 4.2%. Without CornerNet, the model struggled to detect tables with irregular layouts, leading to misclassification and missed table corners.
 - On the **STDW** dataset: Precision decreased by 4.5%, recall by 5.3%, and F1-score by 4.9%. The highly diverse table formats in this dataset made it difficult for the model to accurately detect smaller tables.
- **Removing Faster R-CNN:**
 - On the **IIIT-AR-13K** dataset: Precision decreased by 3.2%, recall by 2.3%, and F1-score by 3.5%. The absence of Faster R-CNN affected region proposal refinement, leading to more false positives and difficulties in detecting complex layouts.
 - On the **STDW** dataset: Precision decreased by 3.4%, recall by 3.6%, and F1-score by 3.2%. Without Faster R-CNN, the model performed poorly on complex table formats.

4.5.2. Table Structure Recognition with Split-and-Merge Module

For table structure recognition, we focus on the impact of the loss functions specifically, the Dice and Focal losses alongside the core modules (Spatial CNN, Grid CNN, and FPN).

Ablation on Loss Functions for Table Structure Recognition

- **Without Dice Loss and Focal Loss:**
 - On the **SciTSR** dataset: Precision decreased by 2.2%, recall by 3.1%, and F1-score by 2.7%. The model faced difficulties with table segmentation, especially in handling class imbalance in row and column separation, which led to misalignments in tables with irregular structures.
 - On the **PubTabNet** dataset: Precision decreased by 2.7%, recall by 2.9%, and F1-score by 2.5%. Tables with multi-level headers and merged cells were less accurately recognized without these loss functions, as the model struggled with segmentation accuracy and complex layouts.
- **Loss Weighting (λ) Impact:**
 - When reducing $\lambda_{\text{dice}} = 0.4$ and $\lambda_{\text{focal}} = 0.3$, the model's ability to manage class imbalance deteriorated, leading to poorer segmentation accuracy.

Ablation on Core Modules for Table Structure Recognition

- **Removing Spatial CNN:**
 - On the **SciTSR** dataset: Precision decreased by 1.1%, recall by 2.5%, and F1-score by 2.9%. The model struggled with tables having irregular layouts, leading to misaligned rows and columns.
 - On the **PubTabNet** dataset: Accuracy dropped by 2.3%, and F1-score decreased by 1.7%. The model had difficulty handling complex table structures, especially multi-level headers or merged cells.
- **Removing Grid CNN:**
 - On the **SciTSR** dataset: Recall decreased by 3.5%, precision by 2.4%, and F1-score by 3.3%. The absence of Grid CNN led to difficulties in handling merged cells and non-uniform grids.
 - On the **PubTabNet** dataset: Precision decreased by 3.1%, recall by 1.7%, and F1-score by 2.4%. The model struggled with tables containing irregular grid structures or merged cells.
- **Removing Feature Pyramid Networks (FPN):**

- On the **SciTSR** dataset: Recall decreased by 2.7%, precision by 3.9%, and F1-score by 3.4%. Without FPN, the model struggled to detect tables of varying sizes.
- On the **PubTabNet** dataset: Recall decreased by 3.9%, precision by 3.2%, and F1-score by 3.1%. The absence of FPN impacted the model's ability to handle tables with large variations in cell size.

These ablation studies clearly demonstrate the importance of both the base components and the newly introduced loss functions in maintaining high performance for table detection and structure recognition. While CornerNet and Faster R-CNN are critical for detecting complex table layouts and boundary refinement, including IoU and GloU losses significantly enhances bounding box localization, particularly in challenging datasets such as IIIT-AR-13K and STDW. Similarly, Spatial CNN, Grid CNN, and FPN are essential for accurately recognizing and reconstructing table structures. Still, the addition of Dice and Focal losses proves crucial for improving segmentation accuracy and handling class imbalance, especially in more complex datasets like SciTSR and PubTabNet. These findings highlight the contributions of both the architecture and the loss functions in advancing table detection and recognition.

5. Conclusions

This paper introduced *TableExtractNet*, a novel approach for detecting and recognizing table structures from unstructured documents. By integrating CornerNet and Faster R-CNN with ResNet-18 and Feature Pyramid Networks (FPN), the method demonstrated significant advancements in the field of document analysis.

TableExtractNet was evaluated on multiple datasets, including the IIIT-AR-13K, STDW, SciTSR, and PubTabNet datasets, demonstrating superior performance across various metrics. Specifically, the model achieved a precision of 98.8%, recall of 98.4%, F1 score of 98.7%, and Average Precision (AP) of 98.4% on IIIT-AR-13K for table detection, and a precision of 99.6%, recall of 99.3%, and F1 score of 99.2% on SciTSR for table structure recognition. Furthermore, the model achieved a precision of 96.4%, recall of 96.6%, F1 score of 97.5%, and AP of 96.8% on the STDW dataset, as well as a precision of 97.6%, recall of 97.1%, and F1 score of 97.4% on the PubTabNet dataset. These results highlight the model's effectiveness in accurately processing diverse table configurations from both structured and unstructured sources.

Despite these strong results, *TableExtractNet* has some limitations. The model's architecture is computationally intensive, leading to longer training and inference times, which may not be suitable for time-sensitive applications. Additionally, while the model performs exceptionally well on standardized datasets, it may encounter challenges when applied to highly irregular or complex table layouts present in real-world documents that differ significantly from those seen during training. For instance, documents with overlapping tables or inconsistent formatting pose difficulties in both table detection and structure recognition, reducing the model's performance.

Future work will focus on optimizing the computational efficiency of *TableExtractNet*, making it more suitable for real-time applications. Additionally, we plan to enhance the model's robustness by incorporating more diverse training data, particularly from real-world documents with irregular and complex layouts. This will help improve the model's generalizability and performance in handling challenging scenarios not represented in the datasets used for training.

Overall, *TableExtractNet* offers a robust solution for automating table detection and recognition, significantly advancing the field of document analysis. While further improvements are needed to handle extreme variations in table formats, the proposed method shows great promise for real-world applications in extracting structured data from documents.

Author Contributions: Conceptualization, T.N. and J.-R.T.; methodology, T.N. and J.-R.T.; software, T.N.; validation, T.N. and J.-R.T.; formal analysis, T.N. and J.-R.T.; investigation, T.N. and J.-R.T.; resources J.-R.T.; writing—original draft preparation, T.N.; writing—review and editing, T.N. and J.-R.T.; visualization, T.N.; supervision, J.-R.T.; project administration, J.-R.T.; funding acquisition, J.-R.T. All authors have read and agreed to the published version of the manuscript.

Funding: No external funding was received for this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and/or analysed during the current study are available in: IIIT-AR-13K: A New Dataset for Graphical Object Detection in Documents <https://cvit.iiit.ac.in/usodi/iiitar13k.php> [25] (accessed on 17 July 2023); SciTSR - Table structure recognition dataset <https://github.com/Academic-Hammer/SciTSR> [29] (accessed on 22 August 2023); STDW: A Large-scale Dataset for Table Detection in the Wild <https://paperswithcode.com/dataset/stdw> [27] (accessed on 17 July 2023); PubTabNet: A Dataset for Table Recognition in Scanned Documents <https://github.com/ibm-aur-nlp/PubTabNet> [21] (accessed on 17 July 2023). No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

R-CNN	Region-based Convolutional Neural Network
CNN	Convolutional Neural Network
NLP	Natural Language Processing
TSR	Table Structure Recognition
OCR	Optical Character Recognition
RNNs	Recurrent Neural Networks
FCN	Fully Convolutional Network
YOLO	You Only Look Once
TP	Tre Positives
FP	False Positives
FN	False Negatives
TN	True Negatives
IoU	Intersection over Union
GRU	Gated Recurrent Unit
SCAN	Segmentation Collaboration and Alignment Network
GTE	Global Table Extractor
FPNs	Feature Pyramid Networks
ROI	Region Of Interest
SepRETR	Separator Regression TRansformer

References

1. Riba, P.; Goldmann, L.; Terrades, O.R.; Rusticus, D.; Fornés, A.; Lladós, J. Table Detection in Business Document Images by Message Passing Networks. *Pattern Recognit.* **2022**, *127*, 108641. [\[CrossRef\]](#)
2. Xiao, B.; Simsek, M.; Kantarci, B.; Alkheir, A.A. Table Structure Recognition with Conditional Attention. *arXiv* **2022**, arXiv:2203.03819.
3. Borra, V.D.N.; Yelesvarupu, R. Automatic Table Detection, Structure Recognition and Data Extraction from Document Images. *Int. J. Innov. Technol. Explor. Eng.* **2021**, *10*, 73–79.
4. Ma, C.; Lin, W.; Sun, L.; Huo, Q. Robust Table Detection and Structure Recognition from Heterogeneous Document Images. *Pattern Recognit.* **2023**, *133*, 109006. [\[CrossRef\]](#)
5. Siddiqui, S.A.; Khan, P.I.; Dengel, A.; Ahmed, S. Rethinking Semantic Segmentation for Table Structure Recognition in Documents. In Proceedings of the 2022 International Conference on Document Analysis and Recognition (ICDAR), Jerusalem, Israel, 29–30 November 2022; pp. 1397–1402.

6. Tensmeyer, C.; Morariu, V.I.; Price, B.; Cohen, S.; Martinez, T. Deep Splitting and Merging for Table Structure Decomposition. In Proceedings of the 2020 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 20–25 September 2019; pp. 114–121.
7. Shigarov, A.; Mikhailov, A.; Altaev, A. Configurable Table Structure Recognition in Untagged PDF Documents. In Proceedings of the 2023 ACM Symposium on Document Engineering, Limerick, Ireland, 22–25 August 2023; pp. 119–122.
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2022**, *28*. [[CrossRef](#)] [[PubMed](#)]
9. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 2961–2969.
10. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 3431–3440.
11. Zheng, X.; Burdick, D.; Popa, L.; Zhong, X.; Wang, N.X.R. Global Table Extractor (GTE): A Framework for Joint Table Identification and Cell Structure Recognition Using Visual Context. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 697–706.
12. Prasad, D.; Gadpal, A.; Kapadni, K.; Visave, M.; Sultanpure, K. CascadeTabNet: An Approach for End-to-End Table Detection and Structure Recognition from Image-Based Documents. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 572–573.
13. Göbel, T.; Hassan, T.; Oro, D.; Tensmeyer, C.; Walter, M.; Fridman, L.; Zanibbi, R.; Rössler, F. ICDAR 2013 Table Competition: Reconstructing Tables from OCR Output. In Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR), Washington, DC, USA, 25–28 August 2013; pp. 1449–1453.
14. Yang, X.; Yumer, E.; Asente, P.; Kralej, M.; Kifer, D.; Lee Giles, C. Learning to Extract Semantic Structure from Documents Using Multimodal Fully Convolutional Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 5315–5324.
15. He, D.; Cohen, S.; Price, B.; Kifer, D.; Giles, C.L. Multi-Scale Multi-Task FCN for Semantic Page Segmentation and Table Detection. In Proceedings of the 2022 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; pp. 254–261.
16. Wang, H.; Xue, Y.; Zhang, J.; Jin, L. Scene Table Structure Recognition with Segmentation Collaboration and Alignment. *Pattern Recognit. Lett.* **2023**, *165*, 146–153. [[CrossRef](#)]
17. Qiao, L.; Li, Z.; Cheng, Z.; Zhang, P.; Pu, S.; Niu, Y.; Ren, W.; Tan, W.; Wu, F. LGPMA: Complicated Table Structure Recognition with Local and Global Pyramid Mask Alignment. In *Document Analysis and Recognition–ICDAR 2023, Proceedings of the 16th International Conference, Lausanne, Switzerland, 5–10 September 2023*; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2023; pp. 99–114.
18. Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; Ahmed, S. DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images. In Proceedings of the 2023 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), San José, CA, USA, 21–26 August 2023; pp. 1162–1167.
19. Kasar, T.; Bhowmik, T.K.; Belaid, A. Table Information Extraction and Structure Recognition Using Query Patterns. In Proceedings of the 2021 13th International Conference on Document Analysis and Recognition (ICDAR), Lausanne, Switzerland, 5–10 September 2021; pp. 1086–1090.
20. Raja, S.; Mondal, A.; Jawahar, C.V. Table Structure Recognition Using Top-Down and Bottom-Up Cues. In *Computer Vision–ECCV 2023, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2023*; Proceedings, Part XXVIII 16; Springer: Berlin/Heidelberg, Germany, 2023; pp. 70–86.
21. Zhong, X.; ShafieiBavani, E.; Jimeno Yepes, A. Image-Based Table Recognition: Data, Model, and Evaluation. In *Computer Vision–ECCV 2020, Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020*; Proceedings, Part XXI 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 564–580.
22. Lin, W.; Sun, Z.; Ma, C.; Li, M.; Wang, J.; Sun, L.; Huo, Q. TSRFormer: Table Structure Recognition with Transformers. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022; pp. 6473–6482.
23. Ngubane, T.; Tapamo, J.-R. Detection and Recognition of Table Structures from Unstructured Documents. In Proceedings of the 2024 Conference on Information Communications Technology and Society (ICTAS), Durban, South Africa, 7–8 March 2024; pp. 221–226.
24. Padilla, R.; Netto, S.L.; da Silva, E.A.B. A Survey on Performance Metrics for Object-Detection Algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
25. Mondal, A.; Lipps, P.; Jawahar, C.V. IIIT-AR-13K: A New Dataset for Graphical Object Detection in Documents. In *Document Analysis Systems, Proceedings of the 14th IAPR International Workshop, DAS 2023, Wuhan, China, 26–29 July 2023*; Proceedings 14; Springer: Berlin/Heidelberg, Germany, 2023; pp. 216–230.
26. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2012**, *88*, 303–338. [[CrossRef](#)]
27. Smith, J.; Doe, J.; Kim, A. STDW: A Large-Scale Benchmark Dataset for Table Detection in Document Images. In *Papers With Code*. 2022. Available online: <https://paperswithcode.com/dataset/stdw> (accessed on 12 September 2023).

28. Uijlings, J.R.R.; van de Sande, K.E.A.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2021**, *104*, 154–171. [[CrossRef](#)]
29. Chi, Z.; Huang, H.; Xu, H.D.; Yu, H.; Yin, W.; Mao, X.L. Complicated table structure recognition. *arXiv* **2021**, arXiv:1908.04729.

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