IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 26, 2021, accepted July 26, 2021, date of publication August 9, 2021, date of current version August 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3103413

Guided Table Structure Recognition Through Anchor Optimization

KHURRAM AZEEM HASHMI[1,2,3], DIDIER STRICKER[1,2], MARCUS LIWICKI[4], (Member, IEEE), MUHAMMAD NOMAN AFZAL[5], AND MUHAMMAD ZESHAN AFZAL[1,2,3]

[1]German Research Center for Artificial Intelligence, 67663 Kaiserslautern, Germany
[2]Department of Computer Science, University of Kaiserslautern, 67663 Kaiserslautern, Germany
[3]Mindgrage, University of Kaiserslautern, 67663 Kaiserslautern, Germany
[4]Department of Computer Science, Luleå University of Technology, 971 87 Luleå, Sweden
[5]Bilojix Soft Technologies, Bahawalpur 63100, Pakistan

Corresponding author: Khurram Azeem Hashmi (Khurram_Azeem.Hashmi@dfki.de)

This work was supported in part by the European Project INFINITY under Grant 883293.

ABSTRACT This paper presents the novel approach towards table structure recognition by leveraging the guided anchors. The concept differs from current state-of-the-art systems for table structure recognition that naively apply object detection methods. In contrast to prior techniques, first, we estimate the viable anchors for table structure recognition. Subsequently, these anchors are exploited to locate the rows and columns in tabular images. Furthermore, the paper introduces a simple and effective method that improves the results using tabular layouts in realistic scenarios. The proposed method is exhaustively evaluated on the two publicly available datasets of table structure recognition: ICDAR-2013 and TabStructDB. Moreover, we empirically established the validity of our method by implementing it on the previous approaches. We accomplished state-of-the-art results on the ICDAR-2013 dataset with an average F1-measure of 94.19% (92.06% for rows and 96.32% for columns). Thus, a relative error reduction of more than 25% is achieved. Furthermore, our proposed post-processing improves the average F1-measure to 95.46% that results in a relative error reduction of more than 35%. Moreover, we surpassed the baseline results on the TabStructDB dataset with an average F1-measure of 94.57% (94.08% for rows and 95.06% for columns).

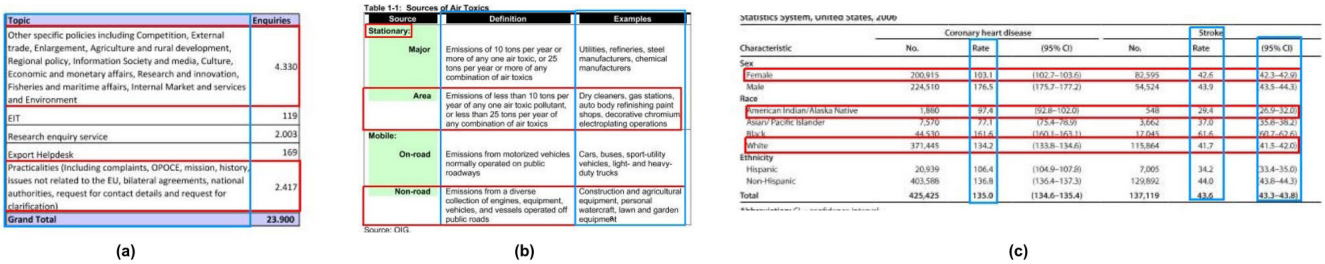
INDEX TERMS Deep neural network, Mask R-CNN, document images, object detection, anchor optimization, guided anchors, table structure recognition, table structure extraction, table understanding.

I. INTRODUCTION

In this modern age of digitization, several camera-equipped devices [1] have been operated daily to upload documents leading to expanding the need for robust systems that can extract information from raw documents images [2]. In the past, numerous approaches have advertised remarkable results in retrieving information by applying Optical Character Recognition (OCR) methods on documents [3]–[5]. One of the most appropriate ways to represent the information in documents is through tables [6]. The table contains significant facts and figures stored in a concise and organized manner [7]. These tabular structures are extensively used as a medium to convey valuable information in domains like finance, administration, research, and even historical documents [8]. Hence, automated identification of these tabular structures is a significant problem in the document analysis community [6], [9], [10].

The problem of table analysis can be explained by breaking it down into two sub-problems: The first problem is identifying the table's boundary in a document image. The second task is to recognize the structure from a tabular image [6]. The task of table detection is a complex problem because of the diversity in tabular patterns. For instance, some tables contain ruling lines while others do not have any information. It is highly probable to detect false positives while spotting a table because of having similarities between tables and charts or figures [11]. These challenges demonstrate that custom heuristics or traditional approaches are not capable of handling the problem of table detection [8]. The recent development in deep learning-based methods has exceptionally improved state-of-the-art table detection methods.

The associate editor coordinating the review of this manuscript and approving it for publication was Badri Narayan Subudhi.



(a)

Topic	Enquiries
Other specific policies including Competition, External trade, Enlargement, Agriculture and rural development, Regional policy, Information Society and media, Culture, Economic and monetary affairs, Research and innovation, Fisheries and maritime affairs, Internal Market and services and Environment	4.330
EIT	119
Research enquiry service	2.003
Export Helpdesk	169
Practicalities (Including complaints, OPOCE, mission, history, issues not related to the EU, bilateral agreements, national authorities, request for contact details and request for clarification)	2.417
Grand Total	23.900

(b)

Table 1-1: Sources of Air Toxics

Source	Definition	Examples
Stationary:		
Major	Emissions of 10 tons per year or more of any one air toxic, or 25 tons per year or more of any combination of air toxics	Utilities, refineries, steel manufacturers, chemical manufacturers
Area	Emissions of less than 10 tons per year of any one air toxic pollutant, or less than 25 tons per year of any combination of air toxics	Dry cleaners, gas stations, auto body refinishing paint shops, decorative chromium electroplating operations
Mobile:		
On-road	Emissions from motorized vehicles normally operated on public roadways	Cars, buses, sport-utility vehicles, light- and heavy-duty trucks
Non-road	Emissions from a diverse collection of engines, equipment, vehicles, and vessels operated off public roads	Construction and agricultural equipment, personal watercraft, lawn and garden equipmeNt

Source: OIG.

(c)

Statistics system, United States, 2006

Characteristic	Coronary heart disease			Stroke		
	No.	Rate	(95% CI)	No.	Rate	(95% CI)
Sex						
Female	700,915	103.1	(102.7–103.6)	82,595	42.6	42.3–42.9
Male	224,510	176.5	(175.7–177.2)	54,524	43.9	43.5–44.3
Race						
American Indian/Alaska Native	1,880	97.4	(92.8–102.0)	548	28.4	26.9–32.0
Asian/ Pacific Islander	7,570	77.1	(75.4–78.9)	3,662	37.0	35.8–38.2
Black	44,530	161.6	(160.1–163.1)	12,045	41.6	40.7–42.6
White	371,445	134.2	(133.8–134.6)	115,864	41.7	41.5–42.0
Ethnicity						
Hispanic	20,939	106.4	(104.9–107.8)	7,005	34.2	33.4–35.0
Non-Hispanic	403,586	136.8	(136.4–137.3)	129,892	44.0	43.8–44.3
Total	425,425	135.0	(134.6–135.4)	137,119	43.6	43.3–43.8

FIGURE 1. Table Structure Recognition problem definition and challenges. Red color defines the bounding box for rows, while blue denotes columns. In the figure, part(a) and part(b) represents tabular images having rows spanning multiple lines, whereas, in part(c), rows are restricted to a single line. Columns can be as wide as illustrated in part(a) and part(b) but also as narrow as shown in part(c). For the sake of clarity, only a few rows and columns are highlighted.

Several researchers have exploited deep learning algorithms to detect the tabular area [12]–[14]. Object detection algorithms have been proven to surpass the rest of the techniques and achieved almost perfect results [8], [15].

The task of table structure recognition is about detecting various cells present in the table [16]. This problem can be further dissolved into detecting rows and columns in a table. Later, the rows and columns can be combined to produce the respective cells [17]. The pre-condition for table structure recognition is the accurately detected tabular regions [18], [19]. Fig. 1 illustrates how the problem of table structure recognition is defined in our approach. Additionally, the figure depicts the challenges that exist due to the diversities in structures of rows (columns) in tabular images. Only a few rows (columns) are marked for the sake of clarity.

Several approaches have tackled the problem of table structure recognition by leveraging additional metadata extracted from the PDF files [20], [21]. However, extracting tabular structures directly from images is perplexing compared to operating over digital-born PDFs [17]. Although few considerable efforts have tackled the problem of recognizing tabular structures straight from images [13], [19], accurate structural recognition is far from achievable [18].

This paper extends the idea of treating the problem of table structure recognition as an object detection problem [18]. In object detection problems, the elementary task is to find the object in a natural scene image. In our case, we operate a document as a natural image while the rows and columns in the table are our targeted objects. While the system DeepTabStr [18] relied on memory-intensive deformable convolutions [22], our approach consists of intuitive utilization of Mask R-CNN [23] with optimized anchors.

The deep neural networks get confused in the localization of rows due to the specialized layouts of tabular structures. The ground-truth between the two publicly available table structure recognition datasets significantly differs in terms of semantics to complicate the matter even more. Fig. 2 depicts the semantic difference between the datasets of ICDAR-2013 [16] and TabStructDB [18]. The ground truth of tabular rows in TabStructDB is labeled without considering the table row's actual contents. Therefore, it is easy for the

	All companies analysed	FTSE Eurotop 100 companies analysed
Number of member states in the analysis	21	8
Number of member states where one or more of the financial companies applied the amendment	11	3

(a)

Estimation method	OLS
Adjusted R^2	0.560
Equation standard error	7.17%
Long-run restrictions (F-test)	1.90 [0.16]
LM test for serial correlation (F-test)	0.06 [0.80]
Normality test (χ^2-test)	0.34 [0.84]
White test for heteroscedasticity (F-test)	0.66 [0.76]
Sample period	1992:Q1-2006:Q4 ($T = 60$)

(b)

FIGURE 2. Visual illustration of semantic difference between the ground truth of two employed datasets. The red bounding box demonstrates the ground truth annotation for rows in tabular images. Part (a) represents the annotation scheme of ICDAR-2013 [16] whereas part(b) depicts the labelling criteria of TabStructDB [18]. It is evident that the ground truth bounding box for rows is restricted to the content in part (a), whereas there is no consideration of the content in (b).

network to work with such annotations. However, it is essential to mention that in production systems, the emphasis is on the reliable extraction of actual content. Hence, the annotation of ICDAR-2013 dataset illustrated in Fig. 2(a) is more realistic in comparison to the annotation scheme of TabStructDB (Fig. 2(b)).

When object detection algorithms are trained on realistic datasets like ICDAR-2013 [16], extra white spaces are added, or important textual information is compromised, resulting in imprecise information extraction from tables. To tackle this critical problem, we have devised a simple and effective post-processing method that can easily be incorporated to improve real-world situations. It is essential to mention that our approach outperformed the state-of-the-art even without incorporating the post-processing method. However, this optional step improves the performance further for the datasets, where recognizing the table's actual content is imperative.

In particular, the contributions of this paper are summarized as follows:

- We have treated the problem of **table structure extraction as an object detection problem** by employing the well-known Mask R-CNN model [23].
- We have implemented a novel **anchor optimization technique** in a region-based convolutional neural network that produces faster network convergence. Moreover, we generalize this method to previous approaches.
- We have introduced a simple and effective **post-processing method to remove the extra white spaces** from the predicted rows. We have demonstrated the effectiveness of this method on the ICDAR-13 dataset [16] and showed that this method is beneficial in recognizing tabular structures in realistic scenarios.
- After **extensive cross-dataset evaluations**, our proposed approach has beaten the **state-of-the-art results on the ICDAR-13 dataset** [16] by using same evaluation metrics proposed by Schreiber *et al.* [17]. Furthermore, we have also **surpassed the baseline results on the TabStructDB** dataset [18].

The rest of the paper is organized as follows: In the beginning, we discuss some of the previous work closely related to our approach in Section II; In Section III we explain our proposed approach and discuss the ideas used in the experiments; Section IV provides a brief overview about the datasets that are exploited in the proposed method; Along with a brief detail over evaluation metrics, we present our results in Section V; Finally, Section VI concludes the paper.

II. RELATED WORK

In this section, we highlight the most relevant related work in the field of table structure analysis. We have divided the contributions into pre and post-deep learning eras described in the following sections. For an exhaustive state-of-the-art overview in the closely related research area of table understanding, refer to [6], [7], [9], [10], [24]–[29].

A. TRADITIONAL APPROACHES

Kieninger and Dengel [20], [30], Kieninger [31], who are the pioneers for working in table structure extraction, tackled the problem by leveraging the traditional approaches. Their proposed system, T-Recs gathered the words into columns by calculating their horizontal ruling lines. Subsequently, the method splits horizontal lines into respective cells based on column margins.

Wangt *et al.* [32] proposed a system that can generate many table ground truths that are beneficial for table recognition systems. The author used a novel table analysis algorithm and an X-Y cut algorithm to extract table structure by detecting the respective cells. Later, another data-driven system proposed by Wang *et al.* [33] that operates on joint probability distributions and deals with both detection and structure decomposition of tables. Their algorithm was analogous to a well-known X-Y cut algorithm [34].

The problem of table structure extraction caught attention when a table structure recognition competition is organized at ICDAR in 2013 [16]. While the first part of the competition was to detect the boundary of the table, the second part of this competition was to recognize the tabular structure by reconstructing the cellular structure of a table. The system employed cell-level metrics to evaluate the performance of the systems. It is important to note that apart from one candidate, all of the participants in the competition vastly used the PDF metadata. However, poor results achieved by the pure image-based system depict that cell-level metrics are not suitable for the evaluation of image-based table analysis systems.

Another approach that leverages PDF metadata to detect the structure of tables is published by Klampfl *et al.* [21]. The system employed a blend of unsupervised learning techniques and hand-crafted heuristics to perform table structure recognition. Kasar *et al.* [35] came up with a query-based system to extract structure of the tables. The system converts the input query taken from the user into a relational graph. Then it compares the query by using a graph matching algorithm to fetch the required information.

Shigarov *et al.* [36] performed an exhaustive evaluation on various algorithms with different thresholds and custom heuristics to tackle the problem of table structure recognition. Their approach was heavily dependent on the PDF meta-data as well. Another approach relying heavily on PDF-metadata is proposed by Rastan *et al.* [37]. Along with recognizing the structure of tables, their system TEXUS can also extract the content from tabular structures.

All these techniques are heavily dependent on the meta-data available in digital-born PDFs. Since our approach works on the scanned document images, these techniques are not directly comparable with our approach.

B. DEEP LEARNING BASED APPROACHES

1) GRAPH NEURAL NETWORKS

Recently, Chi *et al.* [12] has exploited graph neural networks [13] to perform the task of table structure recognition on PDF documents. Another approach powered by graph neural networks is published by Qasim *et al.* [38]. Their model combines the capabilities of convolutional neural networks and graph neural networks to extract tabular structures. Xue *et al.* introduced a bottom-up approach by reconstructing the table structure using a cell relationship network. The system ReS^{2}TIM [39] employed a distance-based weight technique to retrieve a syntactic table structure.

2) RECURRENT NEURAL NETWORKS

Recurrent neural networks [40] have also been employed to handle the problem of table structure extraction [41], [42]. However, most of the prior approaches have utilized PDF meta-data. Since we deal with natural document images, they are not directly comparable to our approach.

3) CONVOLUTIONAL NEURAL NETWORKS

Schreiber *et al.* [17] published the first natural image-based deep learning system to the best of our knowledge, which

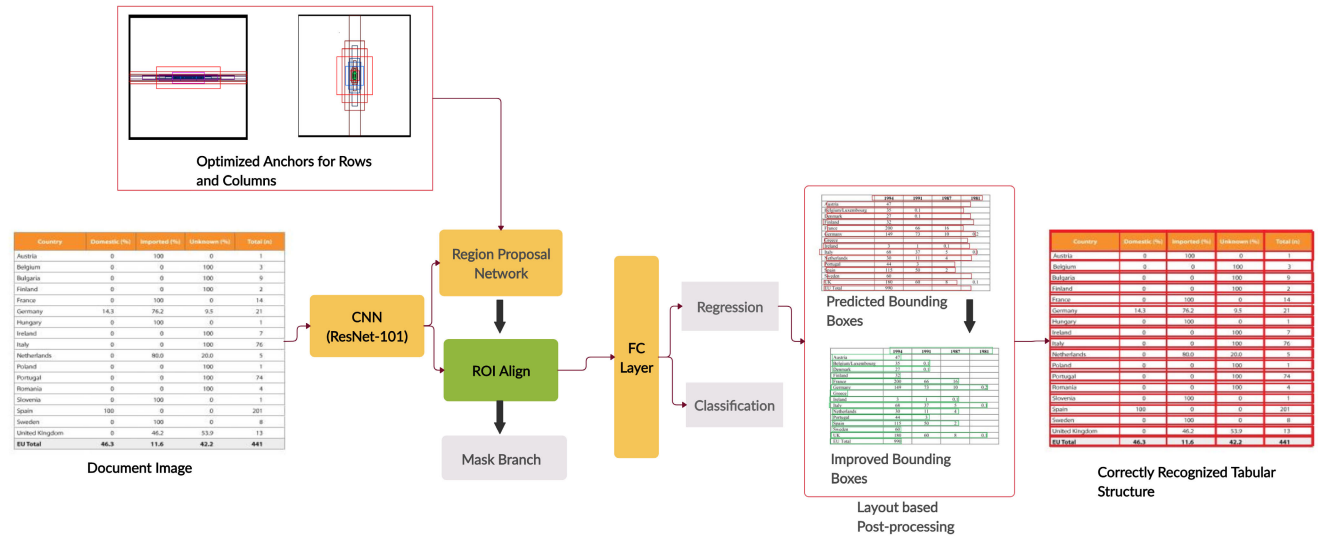


FIGURE 3. The proposed pipeline for Table Structure Recognition. Optimized anchors are given to the region proposal network of Mask R-CNN. After regressing coordinates by the network, the predicted bounding boxes for row detection are further enhanced by employing the post-processing technique.

explored the problem of table structure analysis. The system leverages the Fully Convolutional Network (FCN) [43] to segment the table into rows and columns. Later in 2019, TableNet has been proposed by Paliwal *et al.* [13]. The authors tackled the problem of table structure extraction through a semantic segmentation technique. Another approach powered by semantic segmentation to extract tabular structures from document images is published by Siddiqui *et al.* [19].

These approaches have either used semantic segmentation or FCN to solve the problem of document images. Contrarily, we have chosen to handle the task of table structure recognition as an object detection problem. Although Siddiqui *et al.* [18] has treated the table structure analysis as an object detection problem, there are various considerable differences between the two methods. The system DeepTabStr [18] has adopted Faster R-CNN [44] with deformable convolutions [22] while our proposed approach works with Mask R-CNN [23] exploiting optimized anchors to directly detect boundaries of respective rows and columns in a tabular image.

III. METHOD

We have devised the problem of table structure recognition as an object detection problem. Object detection is a famous problem in computer vision that studies how a machine recognizes objects from a natural scene image. Recent progress in deep learning has remarkably enhanced the state-of-the-art object detection systems [23], [44]. To achieve the ultimate goal of table structure recognition, we have decomposed our problem into two sub-problems. The first one is about detecting rows in tables, while the second sub-problem deals with detecting columns.

A. MODEL

We could implement our approach in two ways:

1) Separate model for both rows and columns.
2) Single combined model to handle both the problems.

Considering the diversity in the structures of rows and columns, it has been empirically established that the separate model performs better [18]. Hence, we have decided to go for two segregated models to solve the problem of table structure recognition.

1) MASK R-CNN

We have adopted Mask R-CNN [23] as our model to identify the rows and columns in a table. Mask R-CNN is one of the accurate object detection algorithms and the latest member of the group of Region-based Convolutional Neural Networks (RCNN) [45]. Mask R-CNN is a two-phase model and has shown compelling performance on the PASCAL VOC [46], and COCO [47] datasets. Researchers have leveraged the capabilities of Mask R-CNN to identify various graphical objects in document images [11].

To execute the training process of the deep neural network, it requires an extensive amount of data that we lack specifically in the domain of table structure extraction. To tackle this problem, we have exploited transfer learning capabilities in our approach. The backbone of our Mask R-CNN is a pre-trained model on ImageNet [48] dataset.

Fig. 3 illustrates the complete pipeline of our proposed approach. Analogous to Faster R-CNN [44], Mask R-CNN [23] follows the two-phase procedure with one addition. The first phase consists of the Region Proposal Network (RPN), which proposes regions of interest in a document image. The second phase deals with the classification of labels and regression of bounding boxes, including the binary masks of

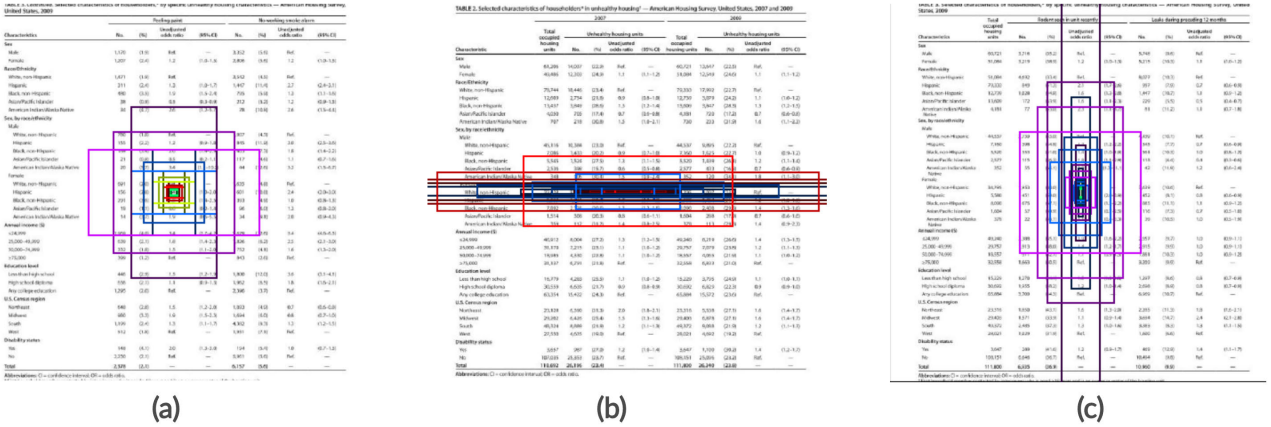


FIGURE 4. Visualization of anchors conventionally used for object detection techniques against optimized anchors used in our approach. (a) Anchors conventionally used for object detection. (b) Optimized anchors for row detection. (c) Optimized anchors for column detection. Traditional anchors are transformed into optimized anchors using the K-Means Clustering technique.

each region of interest. The loss function for of Mask R-CNN is mathematically explained in [23] as:

$$L = L_{cls} + L_{box} + L_{mask} \tag{1}$$

where L represents the sum of classification loss (L_{cls}), bounding box loss (L_{box}), and segmentation loss (L_{mask}). We will now discuss the remaining components of our proposed pipeline presented in Fig. 3.

In the first stage, the combination of ResNet-101 [49] and Feature Pyramid Network (FPN) [50] which is acting as a backbone in our case, extracts the features from the document image. These features are further propagated to Region Proposal Network (RPN). RPN is a lightweight neural network that scans some regions in an image and tries to filter out the more likely regions to contain objects. These input regions for RPN are known as *anchors*. Anchors are defined as a set of rectangular regions with a predefined set of scales and aspect ratios [44]. The RPN generates two kinds of outputs for each anchor:

1) The anchor class states whether an anchor is an object or background.
2) Bounding box refinement, which is the change in the position of the bounding box to precisely fit the object in the proposed region of interest.

B. ANCHOR OPTIMIZATION

The concept of anchors was introduced in the Faster R-CNN by Ren *et al.* [44] which is transported into Mask R-CNN [23] as well. Contrary to the hand-crafted approach of selecting anchors in Mask R-CNN, we have applied the K-means clustering technique to retrieve fine anchors as explained in the approach proposed by Redmon and Farhadi [51]. The anchors traditionally used in object detection consist of various width to height ratios to deal with objects having diverse shapes [52]. However, in detecting rows, we are aware that the anchor's width will always be equal or greater than the height of an anchor while it is the other way around for columns. Hence, anchors having customized sizes and aspect ratios will lead to better performance than anchors commonly used for object detection techniques. It is essential to mention that the euclidean distance was not used as a distance metric in our K-means clustering technique, but the following distance metric [15] is used:

$$D(box, centroid) = 1 - IoU(box, centroid) \tag{2}$$

where the *box* represents the bounding box as a data sample for clustering and *centroid* is the center of a cluster that will be the output of clustering. IoU (Intersection over Union) is an evaluation metrics which is explained in Section V. The purpose of choosing this metric over euclidean distance is that the bigger boxes will lead to more errors as compared to smaller ones which is not the main concern in our scenario [15]. The traditional anchors are given as input to the K-means clustering technique along with the training datasets of ICDAR 2013 [16], and TabStructDB [18] in order to retrieve optimized anchors for each dataset.

Fig. 4 illustrates the comparison between original anchors used for object detection and the optimized anchors for the row (column) detection. The anchor ratios (0.5, 1 and 2) are used in Fig. 4(a) while for Fig. 4(b) and Fig. 4(c), we have used four different anchor ratios (50, 25, 10 and 3) and (0.1, 0.3, 0.5 and 1) respectively. It can be perceived that optimized anchors 4(b) and 4(c) are well suited to execute the task of table structure recognition as compared to the anchors traditionally used for object detection 4(a).

It is essential to mention that RPN scans these optimized anchors on the feature maps instead of an actual document image. The proposed anchor optimization technique improves the model's performance and facilitates the network to converge faster, making our approach even more efficient. Fig. 5 illustrates how optimized anchors help the network to achieve better results in less time. Along with faster network convergence, the optimized anchors significantly

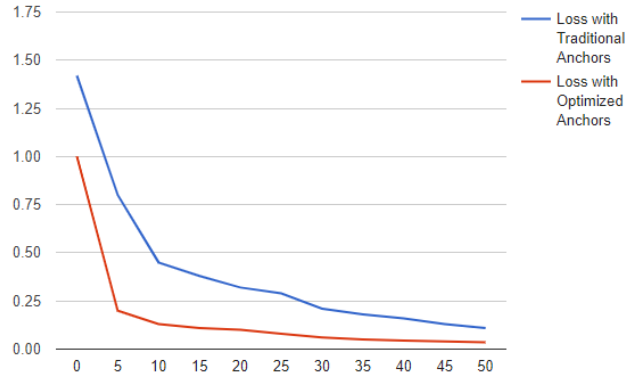


FIGURE 5. Network loss comparison between optimized anchors and traditional anchors for row detection. The blue line represents training loss with traditional anchors, whereas the red depicts the loss by incorporating optimized anchors. The X-axis and Y-axis of the graph denote the number of epochs and loss values, respectively. It is evident that the network with optimized anchors achieves a loss value of less than 0.1 right after 30 epochs, while the network with the conventional anchors cannot achieve the same loss value even after 50 epochs.

TABLE 1. Comparison of F1-measure scores for rows and column detection between conventionally used anchors and our optimized anchors. These results are achieved on the Mask R-CNN model.

Model	Row Detection	Column Detection
Traditional Anchors	0.8710	0.8920
Optimized Anchors	**0.9206**	**0.9632**

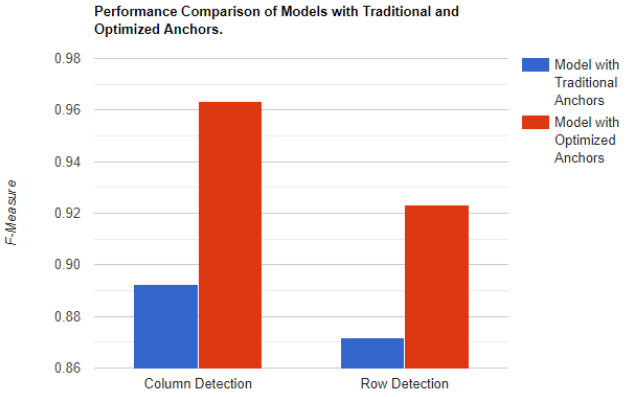


FIGURE 6. Performance comparison between the models trained with traditional anchors and optimized anchors. We have experienced a noticeable increase in the F1-measure score for both rows and columns detection after employing optimized anchors.

improved the performance of our model. The performance comparisons between the models for row and column detection employed with conventional and optimized anchors are exhibited in Fig. 6. Their F1-measure scores are summarized in Table 1.

C. LAYOUT BASED POST PROCESSING

Once the Mask R-CNN detects the rows and columns, we noticed that while the network managed to detect the columns properly, it could not recognize the precise boundaries of rows. In the case of row detection, we observed that the height of predicted bounding boxes is identical to ground truth. However, the network struggled to predict the correct width of a bounding box. The network either causes extra white spaces in the bounding box or drops some valuable information from the rows. To tackle the problem, we came up with a simple and effective post-processing algorithm that can resize the width of a bounding box based on few constraints.

We are aware that the information is written in black for most of the documents. Our proposed method improves precision and recall in two ways:

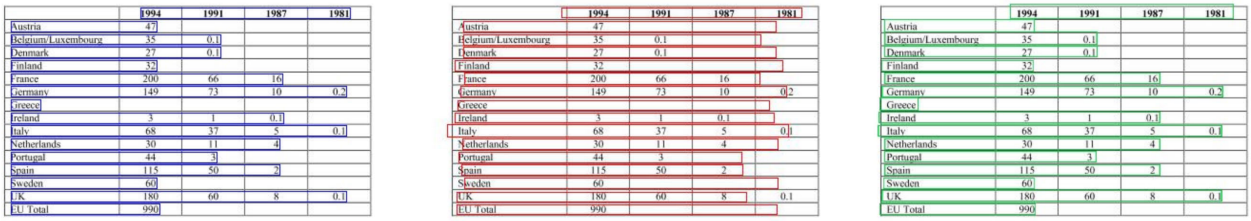
1) Incorporating the important information that was overlooked by the network by increasing the width of a bounding box close to the last set of black pixels.
2) Removing extra white spaces by decreasing the width of the bounding box to the nearest set of black pixels.

Algorithm 1 explains the pseudo-code for the proposed method. It is vital to mention that method does not work in a few cases where the text is not displayed in black pixels. However, the proposed method has shown significant improvements in achieving precise prediction for row detection, summarized in Table 2. Fig. 7 portrays the performance improvement in row detection with a simple post-processing method.

Algorithm 1 Resize the Width of Bounding Box by Identifying Black Pixels

Input: I: 2d array of predicted bounding box
Output: R: Improved bounding box

```
blackPT ← BlackPixelThreshold
Area ← ImageSpecficArea
R ← I
for xValue of R to end of image do
    {checking forward for both xmin and xmax}
    if blackPixelfound then
        Compute blackPixel count in that Area
        if blackPixelCount >= blackPT then
            xmaxofR ← xValue
        end if
    end if
end for
for xValue of R to beginning of image do
    {checking backward for both xmin and xmax}
    if blackPixelfound then
        Compute blackPixel count in that Area
        if blackPixelCount >= blackPT then
            xminofR ← xValue
        end if
    end if
end for
return R
```



(A) Ground Truth

(B) Prdicted Image without Post-processing.

(C) Improved Image with Post-processing.

FIGURE 7. Explaining through example how the IoU for row detection in document images can be further improved with simple post-processing. Detected rows in part (B) are either stretched or reduced to produce accurate boundaries, as illustrated in part (C).

TABLE 2. Performance comparison for row detection with Mask R-CNN before and after applying post-processing technique. After applying our post-processing method, we have seen a significant increase in an F1-measure score in the case of row detection in document images.

Approach	Precision	Recall	F1-measure
Before Post-processing	0.9106	0.9326	0.9206
After Post-processing	**0.9468**	**0.9452**	**0.9460**

It is necessary to emphasize that our layout-based post-processing method is optional. Our anchor optimization approach does not rely on this method to produce state-of-the-art results. However, we have decided to include this algorithm because of its highly effective performance, especially in realistic situations where tabular rows are restricted to the textual regions.

D. ADDITIONAL EXPERIMENTS

To assess the generalization capabilities of our anchor optimization approach and obtain a direct comparison with the prior literature [18], we have incorporated the anchor optimization technique in Faster R-CNN [44] and deformable Faster R-CNN.

1) FASTER R-CNN

Faster R-CNN [44] is a two-stage object detection network built upon Fast R-CNN [44] by replacing the selective search algorithm with a region proposal network. For the detailed explanation of Faster R-CNN, we refer our readers to [44]. Like the Mask R-CNN, the RPN in Faster R-CNN takes the input anchors and proposes the regions of interest. We have followed the same anchor optimization scheme in Faster R-CNN with the identical anchor scales and ratios for rows (columns) as explained in Section III-B.

2) DEFORMABLE FASTER R-CNN

Along with the conventional Faster R-CNN, we have implemented deformable Faster R-CNN to evaluate the performance of our method. The deformable Faster R-CNN leverages deformable convolutional layers, which are introduced by Dai *et al.* [22]. The neurons present in these layers can modify their receptive fields by producing additional offsets based on the previous feature maps. This enables the filters of convolutional layers to adjust to various arbitrary scales and transformations. For the detailed explanation of deformable convolutions, readers may refer to [22] and [53]. The deformable Faster R-CNN modifies the actual Faster R-CNN by replacing the conventional ROI-pooling with the deformable ROI-pooling. Furthermore, instead of the conventional ResNet-101, deformable ResNet-101 is utilized as a based network. Since we have exploited the power of transfer learning throughout our approach, the deformable ResNet-101 is trained on the ImageNet dataset [48]. Analogous to Faster R-CNN and Mask R-CNN, we have adopted the identical anchor optimization approach in our deformable Faster R-CNN.

E. HYPERPARAMETERS

We worked with the combination of ResNet-101 [49] and FPN [50] model as a backbone for both the row and columns detection. Apart from the aspect ratios anchors, the rest of the hyperparameters were identical for both models. For row detection, we have used four different anchor ratios (50, 25, 10, and 3), whereas, for columns, we have picked four different anchor ratios with (0.1, 0.3, 0.5, and 1). However, we have used five different anchor scales (16, 32, 64, 128, 256) for both networks. We trained both models for 50 epochs, where each epoch consists of 100-time steps. The maximum image size was limited to 1024×800, and the images exceeding this size were resized to the maximum dimension. We used a batch size of 2 on a single NVIDIA 1080 Ti GPU. Our model works on stochastic gradient descent, having a momentum value of 0.9 and a learning rate of 0.0001. Gradients are clipped to 5.0, and weights are decayed by 0.0001 at each epoch. In order to prevent the problem of overfitting, we have applied augmentation techniques like random rotations, Gaussian blurring, and random horizontal and vertical flips on the training dataset. We have implemented this work in Keras [54] with Tensorflow [55] as a backend.

IV. DATASETS

We have used two publicly available table structure recognition datasets to conduct the experiments. The particulars of these datasets are explained below.

A. ICDAR-2013

ICDAR-2013 [16] dataset has been used to standardize the state-of-the-art results for the task of table detection and table structure recognition [17], [19]. There are 238 pages in the dataset, out of which 156 contain tabular structures. Originally, the dataset contains labels for cells in a table. However, we have used the transformed version of the dataset [1] published by Siddiqui *et al.* [19]. The authors have converted the cell-based annotations into the corresponding labeling for rows and columns. We have used the identical test split as employed by Schreiber *et al.* [17] in order to implement a direct comparison against the similar approaches [17]–[19]. A sample tabular image is illustrated in Fig. 1.

B. TabStructDB

A Page Object Detection (POD) competition was arranged in ICDAR 2017. The task of this competition was to detect graphical page objects in documents like a table, figures, charts, and equations [56]. By leveraging this dataset, Siddiqui *et al.* [19] has published a new dataset for table structure recognition known as TabStructDB.[2] The dataset contains structural information of each table present in the ICDAR-2017 POD dataset. Each complete row has been annotated separately regardless of the textual region to maintain consistency in the dataset. Hence, making this dataset ideal for the object detection approach. The authors preserved the same dataset split to keep the coherence with the ICDAR-2017 POD dataset. The dataset comprises 731 tabular regions for training, whereas 350 tabular regions are preserved for the testing part. A sample tabular image is illustrated in Fig. 1.

V. EVALUATION

In order to compare our approach with state-of-the-art methods [17]–[19], we have used the identical evaluation metrics which are explained below:

A. INTERSECTION OVER UNION (IoU)

Intersection over Union is a famous evaluation metric used to determine the performance of object detection algorithms. It defines as a measure of a predicted region overlapped with the actual ground truth region. We have used an IoU threshold of 0.5 for the detections. The formula for computing IoU is mentioned below:

$$\frac{\text{Area of Overlap region}}{\text{Area of Union region}} \tag{3}$$

B. PRECISION

Precision is defined as the ratio of correctly predicted region and the total predicted region. The formula for precision is explained below:

$$\frac{\text{Predicted area in ground truth}}{\text{Total area of predicted region}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

[1]ICDAR-2013 dataset is publicly available at: https://bit.ly/2RLgFYu
[2]TabStructDB is publicly available at: https://bit.ly/2XonOEx

C. RECALL

Recall is calculated as the ratio of correctly predicted region and the total ground truth region. The formula for recall is explained as follows:

$$\frac{\text{Predicted area in ground truth}}{\text{Total area of ground truth region}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

D. F1-MEASURE

Harmonic mean of precision and recall is known as the F1-measure or F1-measure. The formula for F1-measure is:

$$\frac{2\times \text{ Precision } \times \text{ Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

It is essential to understand that the precision, recall, and F1-measure are calculated independently for each document, followed by an average over the complete dataset. This evaluation criterion reduces the bias from a single document containing several rows and columns.

As described in Section IV, we have evaluated our proposed approach on the two publicly available datasets: ICDAR-2013 table structure recognition dataset and TabStructDB. Apart from evaluating the datasets on their respective test sets, we have appraised the generalization potential of our approach through the cross-dataset evaluation.

E. ICDAR-2013

Since we are using the modified version of the ICDAR-2013 dataset and we report results based on rows and columns, our approach cannot be directly compared with any of the participants of the ICDAR-2013 table competition [16] and other methods operating on cell-level information. Hence, we compare our approach with the other image-based models that have reported results on rows and columns. To enable the direct comparison with those approaches, we have used the same train/test split proposed by Schreiber *et al.* [17].

Table 4 summarizes the results of image-based table structure recognition methods on ICDAR-2013 dataset. Results depict that our proposed Mask R-CNN with optimized anchors (with and without the involved post-processing method) has outperformed the previous state-of-the-art techniques with an average F1-measure of almost 0.94 and 0.95 respectively. Although results on the column detection of our model are comparable with the DeepTabStR [18], our anchor optimization method has surpassed the performance of row detections resulting in noticeable improvement on the average results and a significant relative error reduction of 25% (without post-processing) and 35% (with post-processing).

For the cross-dataset evaluation, we have trained our models on the TabStructDB dataset and tested on the complete and test set of the ICDAR-2013 dataset. We have reported the results without including the improvement from the proposed post-processing method for all the three models (Mask R-CNN, Faster R-CNN, and deformable Faster R-CNN). This enables us to compare our anchor optimization-based

TABLE 3. Table structure recognition performance on cross-dataset evaluations. In this table, † represents the only approach that has not utilized the optimized anchors, and the results are taken from DeepTabStR [18] to have a direct comparison. The rest of the models operate on optimized anchors. We have achieved all of the results without incorporating the proposed post-processing method.

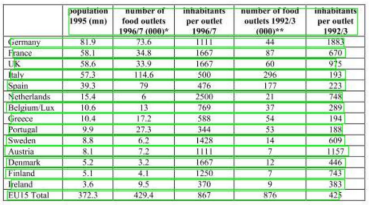
Training Dataset	Testing Dataset	Model	Row			Column			Average F1-measure
			Precision	Recall	F1-measure	Precision	Recall	F1-measure	
ICDAR-13 (Training set)	ICDAR-13 (Test Set)	Faster R-CNN	0.8974	0.9154	0.9063	0.9456	0.9488	0.9510	0.9286
		Deformable Faster R-CNN	0.9071	0.9221	0.9145	0.9511	0.9588	0.9549	0.9347
		Deformable Faster R-CNN†	0.8817	0.4097	0.4531	0.9520	0.9497	0.9497	0.7014
		Mask R-CNN	**0.9106**	**0.9326**	**0.9206**	**0.9605**	**0.9659**	**0.9632**	**0.9419**
	TabStructDB (Complete)	Faster R-CNN	0.6968	0.6632	0.6796	0.6845	0.6973	0.6908	0.6852
		Deformable Faster R-CNN	0.7034	**0.6972**	0.7003	**0.7883**	**0.7561**	**0.7719**	**0.7361**
		Deformable Faster R-CNN†	0.5545	0.2785	0.4531	0.7681	0.7489	0.7533	0.6032
		Mask R-CNN	**0.7189**	0.6850	**0.7034**	0.7037	0.7157	0.7097	0.7011
	TabStructDB (Test Set)	Faster R-CNN	0.6788	0.6634	0.6710	0.7041	0.7255	0.7146	0.6928
		Deformable Faster R-CNN	0.6925	0.6812	0.6868	0.7152	0.7377	0.7263	0.7066
		Deformable Faster R-CNN†	0.5492	0.2622	0.3009	**0.7687**	0.7462	**0.7501**	0.5255
		Mask R-CNN	**0.7142**	**0.6937**	**0.7039**	0.7376	**0.7525**	0.7484	**0.7237**
TabStructDB (Training set)	ICDAR-13 (Complete)	Faster R-CNN	0.7577	0.7322	0.7447	0.6954	0.7125	0.7038	0.7242
		Deformable Faster R-CNN	0.7821	0.7514	0.7664	0.7023	0.7344	0.7180	0.7422
		Deformable Faster R-CNN†	0.6048	0.5507	0.5660	**0.7308**	**0.7518**	**0.7422**	0.6541
		Mask R-CNN	**0.8263**	**0.7729**	**0.7987**	0.7143	0.7226	0.7184	**0.7677**
	ICDAR-13 (Test Set)	Faster R-CNN	0.7321	0.7144	0.7231	0.6543	0.6411	0.6476	0.6853
		Deformable Faster R-CNN	0.7932	0.6350	0.7053	0.6621	0.6721	0.6671	0.6862
		Deformable Faster R-CNN†	0.5279	0.4625	0.4818	**0.6701**	**0.6768**	**0.6705**	0.5761
		Mask R-CNN	**0.8296**	**0.7586**	**0.7925**	0.6453	0.6307	0.6379	**0.7354**
	TabStructDB (Test Set)	Faster R-CNN	0.9074	0.9254	0.9163	0.9556	0.9588	0.9572	0.9367
		Deformable Faster R-CNN	0.9111	0.9285	0.9197	**0.9605**	0.9659	**0.9632**	0.9414
		Deformable Faster R-CNN†	0.8921	0.9125	0.8945	0.9585	**0.9682**	0.9594	0.9269
		Mask R-CNN	**0.9314**	**0.9504**	**0.9408**	0.9523	0.9489	0.9506	**0.9417**

approach with the prior method. In Table 3, it is evident that our deformable Faster R-CNN with optimized anchors have outperformed the deformable Faster R-CNN with conventional anchors [18] both on the complete and test set of ICDAR-2013 with an average F1-measure of 0.74 and 0.68 respectively. In contrast, our original approach with Mask R-CNN yields the average F1-measure of almost 0.74 for the test set and almost 0.77 for the complete dataset. These results in Table 3 explain the diversity between the two datasets and indicate that there is still room in generalizing the system over various datasets.

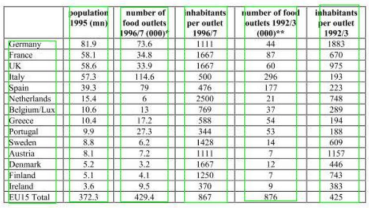
Fig. 8 portrays fragments of correctly recognized tabular structures whereas Fig. 9 depicts some of the cases where rows and columns are not properly detected by the system. In case of incorrect recognition, the model fails to detect few rows in Fig. 9(a) and 9(c) because of having several rows with small width in a document image. In another case in Fig. 9(b), the system was unable to recognize the row spanning in multiple lines. Although most of the columns are correctly detected by the model, there are few instances where the system either does not capture the whole column area or merges multiple small columns into a single column (Fig. 9(d-f)).

TABLE 4. Table structural recognition performance comparison on ICDAR-2013 dataset. Outstanding results are highlighted. Our proposed system of Mask R-CNN with optimized anchors has out-smarted the prior state-of-the-art approaches with and without including the proposed post-processing method.

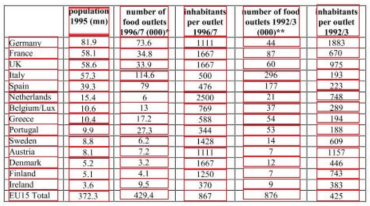
Model	Row			Column			Average		
	Precision	Recall	F1-measure	Precision	Recall	F1-measure	Precision	Recall	F1-measure
DeepDeSRT [17]	-	-	-	-	-	-	0.9593	0.8736	0.91444
TableNet [13]	-	-	-	-	-	-	0.9307	0.9001	0.9151
DeepTabStR [18]	0.8845	0.8945	0.8861	**0.9688**	0.9630	**0.9655**	0.9319	0.9308	0.9298
Siddiqui *et al.* [19]	0.9233	0.9203	0.9190	0.9281	0.9341	0.9288	0.9257	0.9272	0.9239
Proposed System (With post-processing)	**0.9468**	**0.9452**	**0.9460**	0.9605	**0.9659**	0.9632	**0.9537**	**0.9556**	**0.9546**
Proposed System (Without post-processing)	0.9106	0.9326	0.9206	0.9605	**0.9659**	0.9632	0.9355	0.9441	0.9419



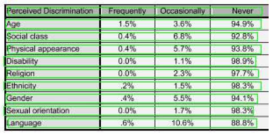
(a) Row Predictions



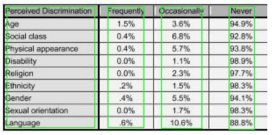
(b) Column Predictions



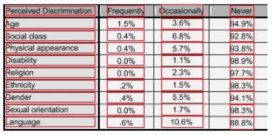
(c) Cell Predictions



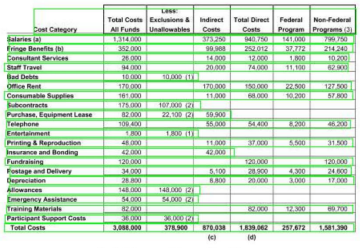
(d) Row Predictions



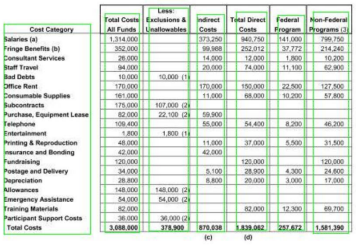
(e) Column Predictions



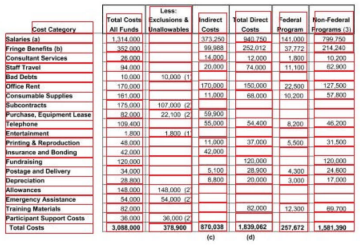
(f) Cell Predictions



(g) Row Predictions



(h) Column Predictions

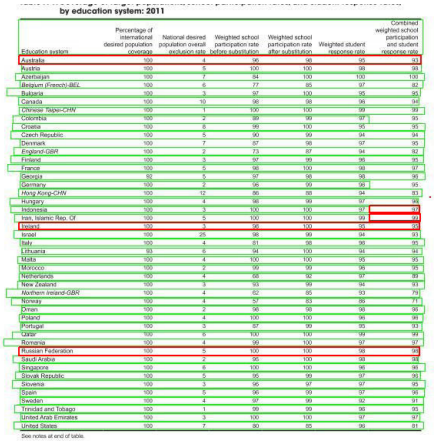


(i) Cell Predictions

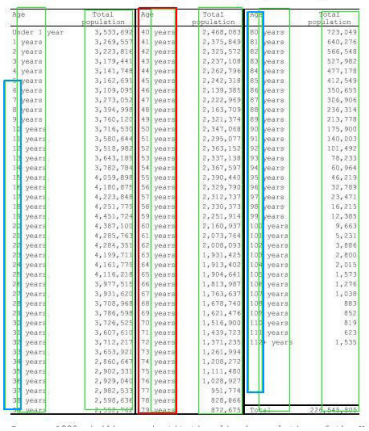
FIGURE 8. Correctly Recognized Table Structures.

F. TabStructDB

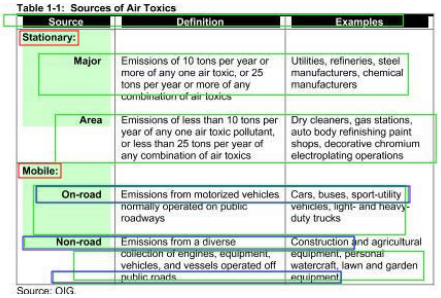
Along with the ICDAR-2013 dataset, we have compared our approach to the TabStructDB dataset. It is evident in the Table 5 that our proposed system has outperformed the baseline results established by the DeepTabStR [18] with an average F1-measure of 0.9417. It is important to mention



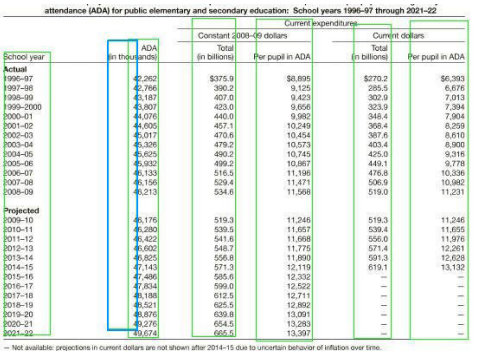
(a) Row Predictions



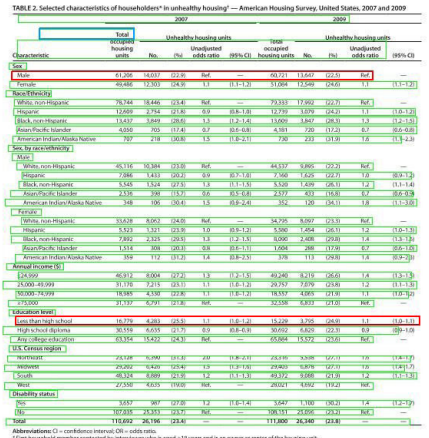
(d) Column Predictions



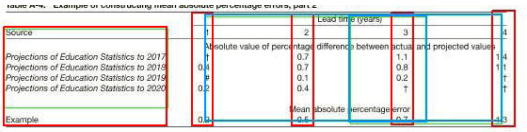
(b) Row Predictions



(e) Column Predictions



(c) Row Predictions



(f) Column Predictions

FIGURE 9. Examples representing incorrectly recognized row and column detection. The green color shows true positives, the blue color depicts false positives, and the red color portrays false negatives for both rows and columns.

that since we have trained our models for rows and columns separately, we have compared our results with theirs achieved on separate training methods with the same train/test split of the dataset.

For the cross-dataset evaluation, a noticeable fall in performance can be perceived in Table 3 when the system (trained on ICDAR-2013) is evaluated on complete and test set of TabStructDB. One of the main reasons for this decline is the disparity in the annotation scheme. The annotations of ICDAR-2013 are limited to textual regions only, while TabStructDB has been labeled with complete rows and columns without considering the textual regions. Since this is an unrealistic scenario, we have not applied the proposed post-processing method while evaluating the performance of our system on the TabStructDB dataset. However, in a direct comparison with deformable Faster R-CNN having

TABLE 5. Table structural recognition performance comparison on TabstructDB dataset. Outstanding results are highlighted. Our proposed system of Mask R-CNN with optimized anchors has outperformed prior baseline results.

Model	Row			Column			Average F1-measure
	Precision	Recall	F1-measure	Precision	Recall	F1-measure	
DeepTabStR [18]	0.9093	0.9404	0.9186	**0.9560**	**0.9628**	**0.9559**	0.9372
Proposed System	**0.9314**	**0.9504**	**0.9408**	0.9523	0.9489	0.9506	**0.9457**

conventional anchors [18], our deformable Faster R-CNN with optimized anchors have shown superior results with an average F1-measure of 0.74 and 0.71 on the complete and test set of TabStructDB.

VI. CONCLUSION AND FUTURE WORK

We have proposed a novel approach that employs object detection as a base and adds intelligent automatic estimation of anchor boxes suitable for table structure recognition. In this paper, we exhibit that current object detectors have already shown remarkable improvements in resolving the problem of table detection [8], [17], are also highly effective in improving the performance of table structure recognition systems. We have adopted the anchor optimization technique that predicts the viable anchors facilitates the object detection process with faster and better results. Furthermore, we have proposed an additional but optional component in our paper: a simple post-processing method showing impressive results in real-world scenarios. Without incorporating the post-processing method, our achieved results have outperformed the state-of-the-art image-based table structure recognition system on the publicly available ICDAR-2013 dataset with an average F1-measure of 94.19%, reducing the relative error to more than 25%. With post-processing, the average F1-measure further improves to 95.46%, resulting in a relative error reduction of more than 35%. Moreover, we surpassed the baseline results on the publicly available TabStructDB dataset with an average F1-measure of 94.57%. The obtained results recommend the idea of exploiting optimized anchors in object detectors for table structure recognition systems.

Although our proposed post-processing technique is nearly applicable to all kinds of document images, some exceptional cases exist. Hence, better post-processing methods should be developed. Our model had a hard time detecting rows spanning multiple lines in the table. An exciting direction could be to detect the cells directly instead of rows and columns. Instead of using the traditional convolutional neural networks, recently proposed CoordConv [57] could also be exploited in the object detection algorithms in order to provide the system with extra contextual information. Along with guided anchors, attention-based region proposal networks [58] could be an interesting future direction. This paper tackles table structure recognition in business-like scanned document images. It would be interesting to examine this approach for the datasets that contain historical document images such as ICDAR-2019 (cTDaR) [59].

REFERENCES

[1] M. Z. Afzal, M. Kramer, S. S. Bukhari, F. Shafait, and T. M. Breuel, "Improvements to uncalibrated feature-based stereo matching for document images by using text-line segmentation," in *Proc. 10th IAPR Int. Workshop Document Anal. Syst.*, Mar. 2012, pp. 394–398.

[2] M. Krämer, M. Z. Afzal, S. S. Bukhari, F. Shafait, and T. M. Breuel, "Robust stereo correspondence for documents by matching connected components of text-lines with dynamic programming," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, 2012, pp. 734–737.

[3] K. A. Hashmi, R. B. Ponnappa, S. S. Bukhari, M. Jenckel, and A. Dengel, "Feedback learning: Automating the process of correcting and completing the extracted information," in *Proc. Int. Conf. Document Anal. Recognit. Workshops (ICDARW)*, Sep. 2019, pp. 116–121.

[4] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 2, Sep. 2007, pp. 629–633.

[5] K. Mokhtar, S. S. Bukhari, and A. Dengel, "OCR error correction: State-of-the-art vs an NMT-based approach," in *Proc. 13th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2018, pp. 429–434.

[6] R. Zanibbi, D. Blostein, and J. Cordy, "A survey of table recognition," *Document Anal. Recognit.*, vol. 7, no. 1, pp. 1–16, Mar. 2004.

[7] M. F. Hurst, "The interpretation of tables in texts," Ph.D. dissertation, School Inform., Univ. Edinburgh, Edinburgh, U.K., 2000.

[8] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, "DeCNT: Deep deformable CNN for table detection," *IEEE Access*, vol. 6, pp. 74151–74161, 2018.

[9] A. C. e Silva, A. M. Jorge, and L. Torgo, "Design of an end-to-end method to extract information from tables," *Int. J. Document Anal. Recognit.*, vol. 8, nos. 2–3, pp. 144–171, Jun. 2006.

[10] D. W. Embley, M. Hurst, D. Lopresti, and G. Nagy, "Table-processing paradigms: A research survey," *Int. J. Document Anal. Recognit.*, vol. 8, nos. 2–3, pp. 66–86, Jun. 2006.

[11] R. Saha, A. Mondal, and C. V. Jawahar, "Graphical object detection in document images," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 51–58.

[12] Z. Chi, H. Huang, H.-D. Xu, H. Yu, W. Yin, and X.-L. Mao, "Complicated table structure recognition," 2019, *arXiv:1908.04729*. [Online]. Available: http://arxiv.org/abs/1908.04729

[13] S. S. Paliwal, V. D, R. Rahul, M. Sharma, and L. Vig, "TableNet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 128–133.

[14] Y. Li, L. Gao, Z. Tang, Q. Yan, and Y. Huang, "A GAN-based feature generator for table detection," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 763–768.

[15] Y. Huang, Q. Yan, Y. Li, Y. Chen, X. Wang, L. Gao, and Z. Tang, "A YOLO-based table detection method," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 813–818.

[16] M. Gobel, T. Hassan, E. Oro, and G. Orsi, "ICDAR 2013 table competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1449–1453.

[17] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "DeepDeSRT: Deep learning for detection and structure recognition of tables in document images," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1162–1167.

[18] S. A. Siddiqui, I. A. Fateh, S. T. R. Rizvi, A. Dengel, and S. Ahmed, "DeepTabStR: Deep learning based table structure recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1403–1409.

[19] S. A. Siddiqui, P. I. Khan, A. Dengel, and S. Ahmed, "Rethinking semantic segmentation for table structure recognition in documents," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1397–1402.

[20] T. Kieninger and A. Dengel, "Applying the T-RECS table recognition system to the business letter domain," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 518–522.

[21] S. Klampfl, K. Jack, and R. Kern, "A comparison of two unsupervised table recognition methods from digital scientific articles," *D-Lib Mag.*, vol. 20, no. 11, p. 7, Nov. 2014.

[22] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[24] J. Hu, R. S. Kashi, D. Lopresti, and G. T. Wilfong, "Evaluating the performance of table processing algorithms," *Int. J. Document Anal. Recognit.*, vol. 4, no. 3, pp. 140–153, Mar. 2002.

[25] B. Coüasnon and A. Lemaitre, *Handbook of Document Image Processing and Recognition, Chapter Recognition of Tables and Forms*, D. Doermann and K. Tombre, Eds. London, U.K.: Springer, 2014.

[26] S. Khusro, A. Latif, and I. Ullah, "On methods and tools of table detection, extraction and annotation in PDF documents," *J. Inf. Sci.*, vol. 41, no. 1, pp. 41–57, Feb. 2015.

[27] D. P. Lopresti and G. Nagy, "A tabular survey of automated table processing," in *Proc. Sel. Papers 3rd Int. Workshop Graph. Recognit. Recent Adv.*, 1999, pp. 93–120.

[28] D. Lopresti and G. Nagy, "Automated table processing," in *Proc. 3rd Int. Workshop. Graph. Recognit. Recent Adv. (GREC)*, Jaipur, India. Berlin, Germany: Springer, 2000, p. 93.

[29] E. R. Dougherty, *Electronic Imaging Technology*, vol. 60. Bellingham, WA, USA: SPIE, 1999.

[30] T. Kieninger and A. Dengel, "A paper-to-HTML table converting system," in *Proc. Document Anal. Sys. (DAS)*, vol. 98, 1998, pp. 356–365.

[31] T. G. Kieninger, "Table structure recognition based on robust block segmentation," in *Document Recognition V*, vol. 3305. Bellingham, WA, USA: International Society for Optics and Photonics, 1998, pp. 22–32.

[32] Y. Wangt, I. T. Phillipst, and R. Haralick, "Automatic table ground truth generation and a background-analysis-based table structure extraction method," in *Proc. 6th Int. Conf. Document Anal. Recognit.*, 2001, pp. 528–532.

[33] Y. Wang, I. T. Phillips, and R. M. Haralick, "Table structure understanding and its performance evaluation," *Pattern Recognit.*, vol. 37, no. 7, pp. 1479–1497, Jul. 2004.

[34] G. Nagy and S. C. Seth, "Hierarchical representation of optically scanned documents," in *Proc. 7th Int. Conf. Pattern Recognit.*, 1984, pp. 347–349.

[35] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "Learning to detect tables in scanned document images using line information," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1185–1189.

[36] A. Shigarov, A. Mikhailov, and A. Altaev, "Configurable table structure recognition in untagged PDF documents," in *Proc. ACM Symp. Document Eng.*, Sep. 2016, pp. 119–122.

[37] R. Rastan, H.-Y. Paik, and J. Shepherd, "TEXUS: A unified framework for extracting and understanding tables in PDF documents," *Inf. Process. Manage.*, vol. 56, no. 3, pp. 895–918, May 2019.

[38] S. R. Qasim, H. Mahmood, and F. Shafait, "Rethinking table recognition using graph neural networks," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 142–147.

[39] W. Xue, Q. Li, and D. Tao, "ReS2TIM: Reconstruct syntactic structures from table images," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 749–755.

[40] A. Cleeremans, D. Servan-Schreiber, and J. L. McClelland, "Finite state automata and simple recurrent networks," *Neural Comput.*, vol. 1, no. 3, pp. 372–381, 1989.

[41] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li, "TableBank: Table benchmark for image-based table detection and recognition," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 1918–1925.

[42] S. A. Khan, S. M. D. Khalid, M. A. Shahzad, and F. Shafait, "Table structure extraction with bi-directional gated recurrent unit networks," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1366–1371.

[43] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[44] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2015, *arXiv:1506.01497*. [Online]. Available: http://arxiv.org/abs/1506.01497

[45] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010.

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2014, pp. 740–755.

[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[50] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[51] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[52] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *Proc. 12th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, May 2017, pp. 650–657.

[53] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9308–9316.

[54] N. Ketkar, "Introduction to Keras," in *Deep Learning With Python*. New York, NY, USA: Springer, 2017, pp. 97–111.

[55] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and S. Ghemawat, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: http://arxiv.org/abs/1603.04467

[56] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "ICDAR2017 competition on page object detection," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1417–1422.

[57] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the CoordConv solution," 2018, *arXiv:1807.03247*. [Online]. Available: http://arxiv.org/abs/1807.03247

[58] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: http://arxiv.org/abs/1706.03762

[59] L. Gao, Y. Huang, H. Dejean, J.-L. Meunier, Q. Yan, Y. Fang, F. Kleber, and E. Lang, "ICDAR 2019 competition on table detection and recognition (cTDaR)," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1510–1515.

KHURRAM AZEEM HASHMI received the bachelor's degree in computer science from the National University of Computer and Emerging Sciences, Pakistan, in 2016, and the M.S. degree from the Technical University of Kaiserslautern. He is currently pursuing the Ph.D. degree with the German Research Center for Artificial Intelligence (DFKI GmbH) and the Technical University of Kaiserslautern, under the supervision of Prof. Dr. Didier Stricker. He has publications in the journals of Multidisciplinary Digital Publishing Institute (MDPI) and IEEE Access. His research interests include deep learning for computer vision, specifically in object detection and activity recognition. He is also interested in the area of pattern recognition and document analysis. Previously, he has worked in the field of document layout understanding and post-OCR error corrections. He is also a Reviewer for IEEE Access.

DIDIER STRICKER is currently a Professor with the University of Kaiserslautern and the Scientific Director with the German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, where he leads the Research Department Augmented Vision. From 2002 to June 2008, he lead the Department ''Virtual and Augmented Reality'' at Fraunhofer Institute for Computer Graphics (Fraunhofer IGD), Darmstadt, Germany. In this function, he initiated and participated too many national and international projects in the areas of computer vision and virtual and augmented reality. In 2006, he received the Innovation Prize of the German Society of Computer Science. He serves as a Reviewer for different European or National Research Organizations and a Regular Reviewer for the most important journals and conferences in the areas of VR/AR and computer vision.

MARCUS LIWICKI (Member, IEEE) received the M.S. degree in computer science from the Free University of Berlin, Germany, in 2004, the Ph.D. degree from the University of Bern, Switzerland, in 2007, and the habilitation degree from the Technical University of Kaiserslautern, Germany, in 2011. Currently, he is the Chaired Professor at Luleå University of Technology and a Senior Assistant with the University of Fribourg. His research interests include machine learning, pattern recognition, artificial intelligence, human–computer interaction, digital humanities, knowledge management, ubiquitous intuitive input devices, document analysis, and graph matching. He is a member of the IAPR, the Editor or a Regular Reviewer for international journals, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING, International Journal of Document Analysis and Recognition (editor), *Frontiers of Computer Science* (editor), *Frontiers in Digital Humanities* (editor), *Pattern Recognition*, and *Pattern Recognition Letters*. He is a member of the governing board of the International Graphonomics Society and a member of the International Association for Pattern Recognition, where he is the Vice president of the Technical Committee 6. He chaired several International Workshops on Automated Forensic Handwriting Analysis and the International Workshop on Document Analysis Systems 2014. Furthermore, he serves as Program Committee Member and a Reviewer for various international conferences and workshops in the area of computer vision, pattern recognition, and document analysis as well as machine learning and e-learning.

MUHAMMAD NOMAN AFZAL received the bachelor's degree in computer science from the Islamia University of Bahawalpur, Pakistan. He is currently involved in research and development in the area of artificial intelligence. He is a Deep Learning Enthusiast. He has over seven years of work experience with different types of deep learning techniques. However, he is mostly interested in object detection. His general interests are deep learning in challenging environments. He has also worked with deploying artificial intelligence at the edge. He has vast experience in mobile development. Furthermore, he is involved in academia where he delivers lectures on machine learning.

MUHAMMAD ZESHAN AFZAL received the master's degree in visual computing from the University of Saarland, Germany, in 2010, and the Ph.D. degree in artificial intelligence from the University of Technology, Kaiserslautern, Germany, in 2016. At an application level, his experience includes generic segmentation framework for natural, human activity recognition, document and medical image analysis, scene text detection and recognition, and online and offline gesture recognition. Moreover, a special interest in recurrent neural networks for sequence processing applied to images and videos. He also worked with numerics for tensor valued images. He worked both in the industry (Deep Learning and AI Lead Insiders Technologies GmbH) and academia (TU Kaiserslautern). His research interests include deep learning for vision and language understanding using deep learning. He is a member of IAPR. He received the Gold Medal for the Best Graduating Student in computer science from IUB Pakistan in 2002 and secured a DAAD (Germany) Fellowship in 2007.