*Article*

# HybridTabNet: Towards Better Table Detection in Scanned Document Images

**Danish Nazir[1,2,†], Khurram Azeem Hashmi[1,2,3,†], Alain Pagani[3], Marcus Liwicki[4], Didier Stricker [1,3] and Muhammad Zeshan Afzal [1,2,3*]**

[1] Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany; dn8034@gmail.com (D.N.); khurram_azeem.hashmi@dfki.de (K.A.H.); muhammad_zeshan.afzal@dfki.de (M.Z.A.); alain.pagani@dfki.de (A.P.); didier.stricker@dfki.de (D.S.);
[2] Mindgarage, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany
[3] German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany,
[4] Department of Computer Science, Luleå University of Technology, 971 87 Luleå, Sweden; marcus.liwicki@ltu.se (M.L.);
* Correspondence: muhammad_zeshan.afzal@dfki.de
† These authors contributed equally to this work.

**Abstract:** Tables in the document image are one of the most important entities since they contain crucial information. Therefore, accurate table detection can significantly improve information extraction from tables. In this work, we present a novel end-to-end trainable pipeline, HybridTabNet, for table detection in scanned document images. Our two-stage table detector uses the ResNeXt-101 backbone for feature extraction and Hybrid Task Cascade (HTC) to localize the tables in scanned document images. Moreover, we replace conventional convolutions with deformable convolutions in the backbone network. This enables our network to detect tables of arbitrary layouts precisely. We evaluate our approach comprehensively on ICDAR-13, ICDAR-17 POD, ICDAR-19, TableBank, Marmot, and UNLV. Apart from the ICDAR-17 POD dataset, our proposed HybridTabNet outperforms earlier state-of-the-art results without depending on pre and post-processing steps. Furthermore, to investigate how the proposed method generalizes unseen data, we conduct an exhaustive leave-one-out-evaluation. In comparison to prior state-of-the-art results, our method reduces the relative error by 27.57% on ICDAR-2019-TrackA-Modern, 42.64% on TableBank (Latex), 41.33% on TableBank (Word), 55.73% on TableBank (Latex + Word), 10% on Marmot, and 9.67% on UNLV dataset. The achieved results reflect the superior performance of the proposed method.

**Keywords:** Table detection, table localization, deep learning, Hybrid Task Cascade, Object detection, deformable convolution, deep neural networks, computer vision, scanned document images, document image analysis.

check for updates

**Citation:** Nazir, D; Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. Title. *Preprints* **2021**, *1*, 0. https://doi.org/

Received:
Accepted:
Published:

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## 1. Introduction

Rapid growth in the digitization of documents has alleviated the demand for methods that can process information accurately and efficiently. Due to the size of the corpus, it has become impractical to employ humans to extract the information. Along with the text, the digital documents contain various graphical page objects like tables, figures, and formulas [1]. While state-of-the-art OCR (Optical Character Recognition) [2–4] systems can process the raw text in document images, they are vulnerable to extract information from the graphical page objects [5]. Hence, it is important to first localize these page objects in document images such that information can be retrieved accurately. Tables are one of the most important page objects in documents because they summarize a major piece of information compactly and precisely. In this paper, we have taken a step forward towards improving the table detection methods in document images.

It has already been established in the community of table understanding [6–12] that table detection in document images hold two major challenges: 1) low inter-class variance (between different classes such as tables, figures and charts). 2) High intra-class variance (within the single class such as tables with and without ruling lines). Due to these challenges, it is highly complex to come up with custom heuristics that can assist in developing robust and generic table detection system [13].

So far, we have seen a similar trend in the advancement of object detection algorithms in computer vision [14–17] with the progress in table detection systems [6–8,11,12]. Although recent object detection frameworks have noticeably improved the performance of table detection approaches [7,18], there is a room in further reducing the close false positives. These case of close false positives can be resolved by leveraging the instance segmentation networks where an additional segmentation loss is added along with the bounding box and classification loss [12,17,19].

In this paper, we have advanced the research for the problem of table detection in scanned document images by introducing the idea of implementing novel state-of-the-art hybrid task cascade networks [20] equipped with deformable convolutions [21]. Unlike prior methods, the proposed technique neither relies on prepossessing methods to transform the raw document images nor requires any rule-based post-processing method to refine the predictions. Moreover, the introduced method is not only applicable for scanned document images but also for PDF documents. Furthermore, the added deformable convolutions in our employed ResNeXt-101 backbone network solve the problem of detecting tables with arbitrary layouts.

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

- We propose HybridTabNet, a novel table detection system by incorporating deformable convolutions in the backbone network of an instance segmentation-based Hybrid Task Cascade (HTC) network.
- During our exhaustive evaluation, we accomplish state-of-the-art performance on five well-recognized publicly available datasets for table detection in scanned document images.
- We present the superiority of the proposed method by reporting results with a leave-one-out scheme on several table detection datasets. The employed strategy sets a new direction, indicating the generalization capabilities of the proposed method.

The remaining paper is organized as follows: Section 2 briefly discusses the earlier literature available on the task of table detection. Section 2.1 talks about the rule-based methods, whereas Section 2.2 highlights learning-based approaches. Section 3 explains the proposed table detection framework by discussing the employed deformable convolutions, backbone network, and object detection algorithm. Section 4 describes the essential details of the datasets that are utilized in the experiments. Section 5 explains the evaluation criteria, whereas Section 6 provides the experiment details and presents both quantitative and qualitative analysis of the proposed method. Section 7 concludes the paper and outlines possible future directions.

## 2. Related Work

Table understanding is an integral step towards document image analysis. Over the past few decades, several researchers have presented solutions for the task of detecting tables having arbitrary layouts in documents. Earlier, most of the proposed methods either rely on custom heuristics or leverage the external meta-data information to tackle the problem of table detection [22–26]. Later, researchers exploited statistical learning [27] followed by deep learning-based approaches to alleviate the generalization capabilities of table detection systems [6–8,10–12,28–32]. This section highlights the brief overview about some of these approaches.

### *2.1. Rule-based Approaches*

Based on our knowledge, the first work on detecting tables in document images was introduced by Itonori et al. [22] in 1993. The approach defines the table as a block of text that follows fixed constraints. In that same year, Chandran and Kasturi [24] came up with a table detection method that relies on vertical and horizontal lines. Pyreddy and Croft [33] presented the system that leverages the custom heuristics to retrieve structural elements from text and separates tabular areas from the extracted elements.

Pivk et al. [34] published the system that is capable of transforming tables embedded in HTML documents into logical structures. This work defines the set of relevant table layout which are exploited to extract tables. Along with tabular layouts, grammar was defined to recognize tables in documents [26]. Hu et al. [35] proposed a table detection method relying on the correlation of white spaces and vertical connected component analysis. For the comprehensive summarization of these rule-based approaches, readers may refer to [13,36–39]. Although these rule-based methods work well on documents with similar tabular layouts, they are laborious in terms of finding optimal heuristics. Furthermore, these conventional approaches are vulnerable to producing generic solutions. Therefore, approaches with better generalization capabilities are required to solve the problem of table detection in document images.

### 2.2. Learning-based Approaches

Kieninger and Dengel [41] introduced T-Recs which is a clustering approach to detect tables in documents. later, in a follow up work, PDF-TREX [42] is proposed. This method applied T-Recs to extract tables from PDF documents. Along with unsupervised learning [41,43], supervised learning was exploited to detect tables in documents [44]. The proposed system, Tabfinder transforms a document into an MXY tree representation. Subsequently, the method proposes the possible tables by looking for the blocks that are enclosed in vertical and horizontal ruling lines. Hidden Markov Models (HMMs) [45,46] and the combination of SVM classifier and custom heuristics [47] has also been exploited to produce table detection methods that depend on visible ruling lines in tables. Although machine learning-based methods have improved the performance of table detection systems, they either rely on the additional meta-data information or tables having specific layouts such as the presence of ruling lines and so on.

With the recent surge of deep learning-based algorithms in computer vision, a similar trend can be seen in the table understanding community. To begin with, Hao et al. [48] implemented a deep Convolutional Neural Network (CNN) to extract spatial features which were later combined with custom heuristics and meta-information from PDF to classify tabular regions in documents. Later object detection algorithms [10,14–17] are heavily explored to develop robust and data-driven image-based table detection systems [6,8–12,28,30].

Gilani et al. [11] employed Faster R-CNN [15] to detect tables in document images. In this work, the raw document images are first transformed by modifying their pixel values using the distance transform mechanism. These transformed images are fed to the object detection network to aid the process of recognizing tabular structures. An end-to-end image based table detection method has been published by Schreiber et al. [6]. The proposed method exploited Faster R-CNN [15] with a pretrained backbones (ZFNet [49] and VGG-16 [50]).

The system GOD (Graphical Object Detection) [12] is an object detection framework that detects graphical page objects in document images. In the proposed work, the author empirically claimed that Mask R-CNN [16] works better as compared to Faster R-CNN [15] in recognizing graphical page objects in scanned document images. A similar conclusion has been presented by Zhong et al. [51] in which their novel proposed dataset PubLayNet is evaluated on both Faster and Mask R-CNN.

Instead of conventional convolutions, Siddiqui et al. [7] employed deformable convolutions to detect tables in document images. The authors empirically established that the dynamic receptive field of deformable convolutions adapts better in detecting tabular boundaries having arbitrary layouts. In another work, Faster R-CNN [29] is employed to tackle the problem of table detection in document images. The final tabular area is retrieved by refining the coroners of predicted tabular boundaries. Vo et al. [30] presented

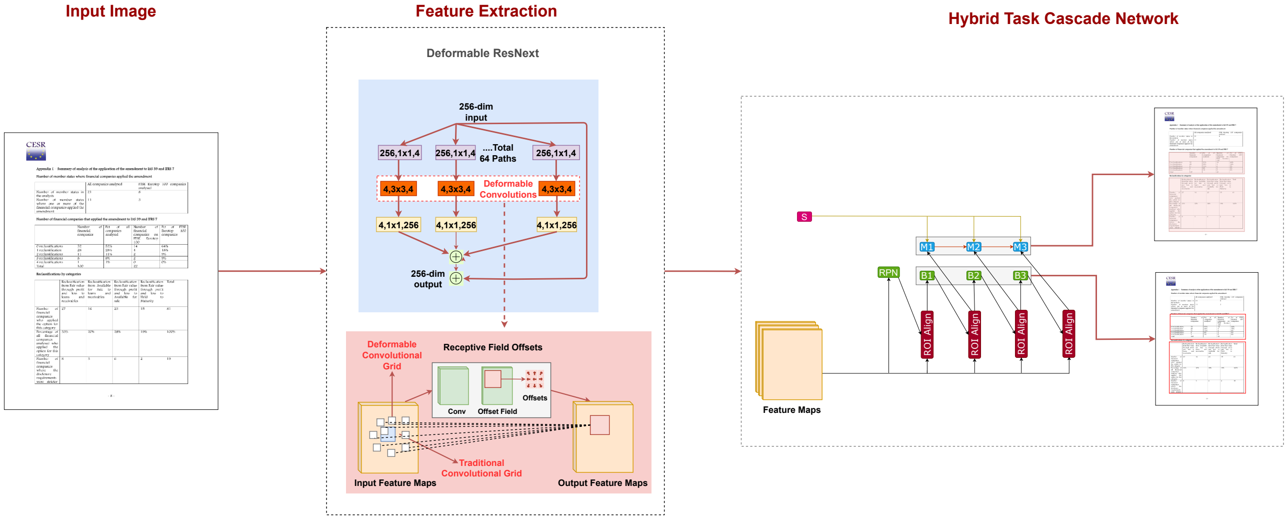


**Figure 1.** HybridTabNet for Table detection and segmentation. In the first step, it performs feature extraction using ResNeXt-101 [40] with deformable convolution layers. The second step utilizes Hybrid Task Cascade network to regress bounding box and semantic mask coordinates of the table in the image.

an ensembling technique in which Fast R-CNN [14] and Faster R-CNN [15] are combined to detect graphical page objects in document images.

Since tabular images are limited in number, Several of the above-mentioned approaches leverage fine-tuning techniques [6,7,11]. In one of the recent works [28], it has been proposed that close-domain fine-tuning performs better as compared to open-domain fine-tuning for detecting tables in document images. In order to establish this conclusion, the authors exploit Mask R-CNN [16], RetinaNet [52], SSD [53], and YOLO [54] to perform the task of table detection.

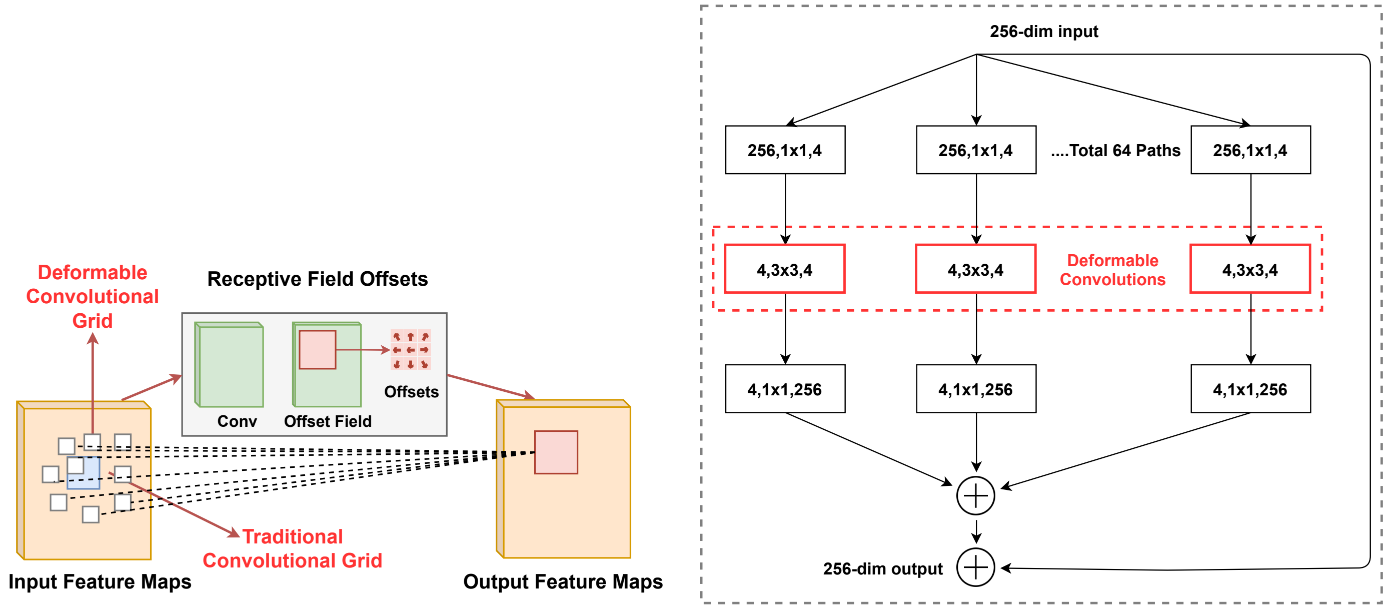
CascadeTabNet [8] is an end-to-end table detection system that operates on Cascade Mask R-CNN with is an extension of Cascade R-CNN [17]. Along with the novel object detection network, the proposed approach relies upon transfer learning, image transformation, and data augmentation techniques to produce state-of-the-art results for table detection in document images.

## 3. Method

Figure 1 illustrates the pipeline of the proposed HybridTabNet. It comprises a ResNeXt-101 [40] with deformable convolution layers and a Hybrid Task Cascade network (HTC). ResNeXt-101 extracts features maps from the dataset, and HTC uses the extracted feature maps to propose regions through Region Proposal Network (RPN). It performs Region of Interest (ROI) align or pooling on the proposed regions, and the bounding box and semantic heads use pooled feature maps to compute bounding boxes and semantic regions. The whole pipeline is trained in an end to end manner. The following sections will describe the essential components of our proposed approach.

### *3.1. Deformable Convolution*

Convolutional Networks [55] have been very successful over the past years on applications like object detection and segmentation [56] [57][58]. However, they cannot model complex geometric transformations due to their fixed kernel size. Deformable convolutional layers [21] was introduced to overcome this limitation. The intuition behind deformable convolutional layers is to add 2D offsets at regular grid sampling positions in the standard convolution operation, which deforms the constant receptive field of the preceding activation unit. The added offsets are learnable from the preceding feature maps. The receptive fields of the deformable layers are adaptive, which changes according to



**(a)** Visualization of 3 × 3 deformable convolution.

**(b)** Architecture of modified ResNeXt-101 block.

**Figure 2.** The components of our feature extraction pipeline. Part **a)** shows the structure of deformable convolutions where the traditional convolutional grid (in blue) is transformed into deformable grid(in white) by adding 2D offsets. Part **b)** shows that the conventional convolutions are replaced with deformable convolutions in ResNeXt-101 to extract tables at multiple scales.

the scale of the object, and It allows to capture objects at different scales [21]. Another advantage of deformable layers is that it adds fewer learnable parameters to the existing model [21].

The deformable convolution operation can be defined as follows.

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta \mathbf{p}_n) \tag{1}$$

Where $\mathcal{R}$ is any kernel of size $n \times n$, $\mathbf{w}$ is the weight of the kernel, $\mathbf{x}$ is the input feature map, $\mathbf{y}$ is the output of convolution operation, $\mathbf{p_0}$ is the starting position of each kernel, $\mathbf{p_n}$ is enumerating along with all the positions in $\mathcal{R}$ and $\Delta\mathbf{p}_n$ denotes the offsets added to the normal convolution operation.

Figure 2a depicts that 2D offsets for the deformable layer are obtained by applying a convolutional layer over the input feature maps. The spatial resolution and dilation of the convolution kernel are the same as in the current convolutional layer. The output channels are of dimension 2N, where N corresponds to the number of 2D offsets.

### 3.2. ResNeXt-101

ResNeXt [40] is a variant of ResNet [59] and it exposes a new dimension called Cardinality along with width and depth. Cardinality defines the size of the transform set, which greatly contributes to the performance of ResNeXt [40]. The experiments have shown that cardinality has shown better performance than going wide and deep [40]. We use ResNeXt-101 as a backbone for feature extraction with Cardinality = 64 and bottleneck width = 4d. Figure 2b illustrates that the convolutional layers in blocks c3-c5 are replaced by deformable layers.

### 3.3. Hybrid Task Cascade

Cascade architecture has been very successful and effective in tasks such as object detection [17]. However, to successfully apply the idea of cascade architecture to instance segmentation problems was still an open-ended research question until HTC [20] was introduced. The main idea behind HTC is to leverage the relationship between object de-

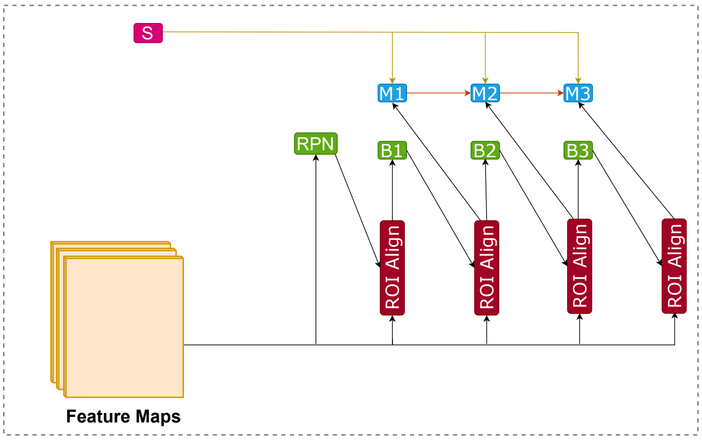


**Figure 3.** Explained architecture of Hybrid Task Cascade Network. It utilizes three box and mask heads in a cascading architecture to produce accurate predictions.

tection and segmentation tasks. Instead of treating detection and segmentation as different problems, it performs joint multi-stage processing. The joint multi-stage processing refers to the combination of object detection and segmentation at each stage. Due to the joint multi-stage processing, the improvement in one task, e.g. detection, improves the mask prediction and segmentation task [20]. It also utilizes spatial context to distinguish the background from the foreground. The semantic branch (S) provides spatial cues which complement the bounding box and mask features.

Figure 3 exhibits the architecture of HTC. It has multiple heads for both bounding box and semantic segmentation to process input at different scales. It consists of segmentation branch (S) with mask (M1,M2,M3) and bounding box (B1,B2,B3) heads. The RPN head predicts preliminary object proposals for these feature maps, whereas the semantic segmentation branch predicts per-pixel semantic segmentation for the whole image through a fully convolutional architecture trained jointly with other branches. In the first stage of architecture, it applies RoI pooling to the output features maps of the backbone model. B1 takes the output of RoI pooling as an input to make RoI-wise predictions. Each head makes two predictions: bounding box classification scores and box regression points. In the second stage, M1 generates pixel-wise segmentation masks for positive RoI's. The rest of the stages follows the same flow. At the inference time, object detection made by Bbox heads are complemented with segmentation masks made by mask head for all detected objects.

Equation 2 explains the flow of Figure 3.

$$\begin{aligned} \mathbf{x}_t^{box} &= \mathcal{P}(\mathbf{x}, \mathbf{r}_{t-1}) + \mathcal{P}(S(\mathbf{x}), \mathbf{r}_{t-1}) \\ \mathbf{r}_t &= B_t\left(\mathbf{x}_t^{box}\right) \\ \mathbf{x}_t^{\text{mask}} &= \mathcal{P}(\mathbf{x}, \mathbf{r}_t) + \mathcal{P}(S(\mathbf{x}), \mathbf{r}_t) \\ \mathbf{m}_t &= M_t\left(\mathcal{F}\left(\mathbf{x}_t^{\text{mask}}, \mathbf{m}_{t-1}^{-}\right)\right) \end{aligned} \tag{2}$$

where S indicates the semantic segmentation head, $\mathbf{x}$ indicates the CNN features of backbone network, $\mathbf{x}_t^{box}$ and $\mathbf{x}_t^{mask}$ indicates box and mask features derived from $\mathbf{x}$ and the input RoI. $\mathcal{P}(.)$ is a pooling operator which could be RoI Align or RoI pooling, $B_t$ and $M_t$ denote the box and mask head at the $t$-th stage, $\mathbf{r}_t$ and $\mathbf{m}_t$ represent the corresponding box predictions and mask predictions. Equation 2 indicates that box and mask heads of each stage takes RoI features extracted by the backbone network and semantic features given by semantic segmentation head. It is essential for HTC because it can differentiate between tables in a cluttered background by exploiting the semantic features.

Since modules given in Equation 2 are differentiable [20]. HTC can be trained in an

doi:10.20944/preprints202108.0360.v1

end-to-end manner. The overall loss function can be formulated in the form of multi-tasking [20] learning.

$$\begin{aligned}&\mathcal{L} = \textstyle\sum_{t=1}^{T} \alpha_t(\mathcal{L}_{bbox}^{t} + \mathcal{L}_{mask}^{t}) + \beta\mathcal{L}_{seg},\\ &\mathcal{L}_{bbox}^{t}(c_i, \mathbf{r}_t, \hat{c}_t, \hat{\mathbf{r}}_t) = \mathcal{L}_{cls}(c_t, \hat{c}_t) + \mathcal{L}_{reg}(\mathbf{r}_t, \hat{\mathbf{r}}_t),\\ &\mathcal{L}_{mask}^{t}(\mathbf{m}_t, \hat{\mathbf{m}}_t) = BCE(\mathbf{m}_t, \hat{\mathbf{m}}_t)\\ &\mathcal{L}_{seg} = CE(\mathbf{s}, \hat{\mathbf{s}}).\end{aligned} \tag{3}$$

Where $\mathcal{L}_{bbox}^{t}$ is the loss of the bounding box predictions at stage $t$ and it combines two terms $\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$, respectively for classification and bounding box regression. $\mathcal{L}_{\mathrm{mask}}^{t}$ is the loss of mask prediction at stage $t$, which adopts the binary cross entropy form as in Mask R-CNN [16]. $\mathcal{L}_{seg}$ is the semantic segmentation loss in the form of cross entropy. The coefficients $\alpha_t$ and $\beta$ are used to balance the contributions of different stages and tasks.

## 4. Datasets

### *4.1. ICDAR-13*

ICDAR-2013 [60] is one of the widely used datasets not only for the problem of table detection but table structure recognition as well. The dataset consists of PDF files that are converted into images to perform an image-based table detection method. There are a total of 238 images that are utilized in evaluating our approach. In order to obtain a direct comparison with prior state-of-the-art-approaches [6,7]. we have used an IoU threshold of 0.5 to calculate the f1-score.

### *4.2. ICDAR-17 POD*

ICDAR-2017-POD (Page Object Detection) [1] is another dataset that is released at ICDAR in 2017. Along with tables, the dataset also has information for the boundaries of figures and formulas. This is fairly a bigger dataset than ICDAR-13 [60]. The dataset consists of 2417 images in total where 1600 images are used for the training purpose and 817 images are employed in testing. Since the prior works [7] have been evaluated with an IoU threshold of 0.6 and 0.8, we have also evaluated our approach in the same manner.

### *4.3. ICDAR-19*

ICDAR-2019 [61] is the outcome of the recently organized competition for table recognition at ICDAR 2019. This novel dataset contains two types of document images (modern and historical). Modern document images are retrieved from scientific papers and commercial documents whereas the archival part of the dataset contains hand-written document images. As suggested in the competition, for the modern part of the dataset, 600 images are allocated for training whereas 240 images are for the testing purpose. Similarly, for the historical part, 600 images are assigned for training and 199 images are adopted for the testing.

### *4.4. Marmot*

Before the advent of TableBank [62], marmot was one of the largest publicly available datasets for the task of table detection. Institute of Computer Science and Technology (PekingUniversity) proposed this dataset which was later elaborated by Fang et al. [63]. The dataset consists of 2000 images where a ratio of almost 1:1 is present between the positive to negative samples. Since the original version of the dataset has few incorrect annotations, we have employed the corrected version of the dataset from [6]. Hence, instead of 2000, 1967 images are utilized in our evaluation.

### *4.5. UNLV*

UNLV dataset is one of the most recognized datasets in the document analysis community. In general, the dataset is comprised of almost 10000 document images. However, only 427 of them contain tables. In our experiments, we have only utilized the document images that contain tabular information.

doi:10.20944/preprints202108.0360.v1

### 4.6. TableBank

Li *et al.* [62] introduced TableBank as one of the most prominent datasets in the table community. Since this dataset has 417,000 document images, we use this dataset to train our network. It is essential to highlight that instead of the whole dataset, we utilize 1500 images each from Word and Latex split and 3000 images from the Word+Latex split to compare our results with prior state-of-the-art approach [8].

## 5. Evaluation Metrics

In this section, we will discuss the evaluation criteria used to evaluate our approach to table detection. We have used similar evaluation metrics to current state-of-the-art approaches [8] [7][12] for comparison of our results.

Precision, recall and, f1-scores are calculated on IoU [64] [65] threshold 0.5,0.6, 0.7, 0.8 and 0.9, respectively.

### 5.1. Intersection of Union

Intersection over Union (IoU) [64][65] is one of the most famous evaluation metrics used in object detection benchmarks. It measures the overlap between predicted and ground truth data. A higher value of IoU means that there is more overlap in predicted and ground truth regions. We use IoU thresholds from 0.5, 0.6, 0.7, 0.8, 0.9 to evaluate our table predictions. The formula for IoU is summarized in Equation 4.

$$\mathbf{IOU} = \frac{Area\ of\ Intersection}{Area\ of\ Union} \tag{4}$$

### 5.2. Precision

Precision is the ratio of correctly predicted observations to the total predicted observations. Equation 5 depicts the formula of precision.

$$\mathbf{Precision} = \frac{True\ Positives}{TruePositives\ +\ False\ Positives} \tag{5}$$

### 5.3. Recall

Recall is the ratio of correctly predicted observations to the total observations in ground truth. Equation 6 depicts the formula of recall.

$$\mathbf{Recall} = \frac{True\ Positives}{TruePositives\ +\ False\ Negatives} \tag{6}$$

### 5.4. F1-score

F1-score is the harmonic mean of precision and recall. Equation 7 exhibits the formula of the f1-score.

$$\mathbf{F1\text{-}score} = 2 \times \frac{Precision\ \times\ Recall}{Precision\ +\ Recall} \tag{7}$$

### 5.5. Weighted-Average

We evaluate the f1-score at IoU thresholds 0.5, 0.6, 0.7, 0.8, 0.9 and report the weighted-average (W.Avg) on the datasets. It allows us to give more importance to f1-scores, precision and recall with higher IoU thresholds. Equation 8 depicts the formula of weighted-average.

$$\mathbf{Weighted\text{-}Average} = \frac{\sum\limits_{n=0.5}^{0.9} n \times S_n}{\sum\limits_{n=0.5}^{0.9} S_n} \tag{8}$$

Where n represents the IoU threshold and $S_n$ highlights the score achieved on a specific IoU threshold, ranging from 0.5 - 0.9. All of the weighted- average precision, recall, and f1-score are calculated in the similar fashion.

## 6. Experiments and Results

To perform all of our experiments, we use the MMDetection [66] framework, an open-source framework for object detection based on Pytorch [67]. We experiment HTC with backbone models ResNet-50 [59] and ResNeXt-101 [40] with deformable convolutions on different datasets to extract the best possible results. We use the configuration files *resnet50_fpn* and *rexnext101_64x4d_fpn_c3-c5 _deconv* (cardinality = 64 and Bottleneck width = 4 with deformable convolutions in resnet stage 3 to 5) from MMDetection [66] to implement our backbones models. Both of the models are pretrained on COCO-2017[68] dataset and use Feature Pyramid Network (FPN) [69] neck. FPN extract features at multiple spatial scales to obtain both low and high-level structures in the image. The ResNet-50 [59] uses learning schedule of 1x whereas ResNeXt-101 [40] uses 20x learning schedules. We evaluate our results on IoU thresholds 0.5, 0.6, 0.7, 0.8, 0.9 which allows us to perform direct comparison with state-of-the-art approaches [8] [7][12] [6] in tables detection and segmentation..

In the subsequent sections, we will discuss our results on different datasets and compare them with state-of-the-art methods.

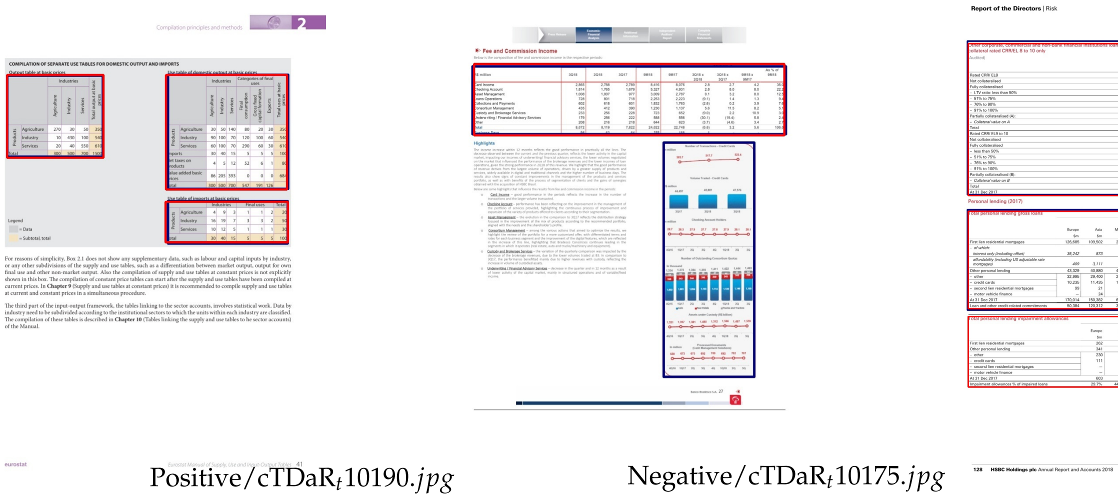
### *6.1. ICDAR-19*

We finetune HybridTabNet [20] with backbone models i.e. ResNet-50 [59] and ResNeXt-101 [40] on ICDAR 2019 Track-A Modern dataset [61]. For training and evaluation of our approach, We use the official train-test split given by ICDAR 2019 [61].

**Table 1.** HybridTabNet results on ICDAR-19 dataset with deformable ResNeXt-101 and ResNet-50 backbones. W.Avg denotes weighted-average of the respective measure on IoU threshold.

| Backbone | IOU Threshold | Precision | Recall | F1-score |
|---|---|---|---|---|
| ResNeXt-101 | 0.5 | 0.9538 | 0.9526 | 0.9532 |
| | 0.6 | 0.9372 | 0.9487 | 0.9424 |
| | 0.7 | 0.9274 | 0.9486 | 0.9336 |
| | 0.8 | 0.9208 | 0.9331 | 0.9278 |
| | 0.9 | 0.8957 | 0.9057 | 0.9017 |
| | **W.Avg** | **0.9231** | **0.9346** | **0.9283** |
| ResNet-50 | 0.5 | 0.9280 | 0.9204 | 0.9242 |
| | 0.6 | 0.9059 | 0.9222 | 0.9139 |
| | 0.7 | 0.8949 | 0.9109 | 0.9028 |
| | 0.8 | 0.8796 | 0.8953 | 0.8874 |
| | 0.9 | 0.8315 | 0.8463 | 0.8388 |
| | **W.Avg** | **0.8817** | **0.8940** | **0.8877** |

Table 1 summarizes the quantitative results of our approach. Our approach achieves the highest precision of 0.9538 and a recall of 0.9526 on a lower IoU threshold of 0.5 with the ResNeXt-101 backbone. However, from the IoU threshold of 0.5 to 0.8, there is only a slight decline in f1-scores. It shows that the performance of our approach is not only limited to the lower IoU threshold. ResNet-50 backbone achieves the highest precision of 0.9280 and a recall of 0.9204 at 0.5 IoU threshold, which is lower than the ResNext-101 backbone. Moreover, compared to ResNeXt-101 W.Avg of 0.9283, the W.Avg of f1-score for



Positive/cTDaR$_t$10064.*jpg*

**(a)** True Positive.

Positive/cTDaR$_t$10190.*jpg*

**(b)** False Positive.

Negative/cTDaR$_t$10175.*jpg*

**(c)** False Negative.

**Figure 4.** HybridTabNet results on ICDAR-2019 Track-A Modern dataset. **a)**, **b)** and **c)** shows True positive, False positive and False negative results respectively. Blue outline shows the predicted table and red outline highlights ground truth table.

ResNet-50 is only 0.8877. Therefore in the succeeding datasets, HybridTabNet uses only ResNeXt-101 with deformable convolutions as backbone in our experiments.

Figure 4 presents the qualitative results of the HybridTabNet on the ICDAR-2019 dataset. In Figure 4b, it confuses a group of bar charts with the table, whereas in Figure 4c, it fails to detect a table without a boundary.

### *6.2. ICDAR-17 POD*

We finetune HybridTabNet with ResNeXt-101 backbone on ICDAR-17-POD [1] dataset. Table 2 quantifies the results of HybridTabNet on the ICDAR-17-POD dataset. It achieves the highest precision of 0.8820 and recall of 0.9972 on IoU thresholds of 0.5 to 0.8. On IoU threshold 0.9, it achieves the recall of 0.9836 and precision of 0.8698. Overall, the recall value is high and close to 1, whereas the precision value is low. This result means that it rarely fails to detect table region in the image but also incorrectly labels regions as tables. Figure 5 depicts the qualitative results of HybridTabNet. Figure 5b shows the False Positive predicted by our model where it confuses an image containing a graph as a table. Figure 5c shows False Negative of our approach where it fails to detect a clear table.

### *6.3. TableBank*

TableBank [62] is a unique dataset that comprises three types of documents, i.e., Latex, Word and a mixture of Latex and Word documents. It has a separate dataset for each document type. We use a smaller train-test split for training which is defined by the current state-of-the-art approach [8]. It allows us to perform direct comparison of our results with their results. We perform finetuning of HybridTabNet on each of the three datasets in the TableBank dataset.

**Table 2.** HybridTabNet results ICDAR-17 POD results with a deformable ResNeXt-101 backbone. W.Avg denotes weighted-average of the respective measure on IoU threshold.

| Model | IOU Threshold | Precision | Recall | F1-score |
|---|---|---|---|---|
| HybridTabNet | 0.5 | 0.8820 | 0.9972 | 0.9360 |
| | 0.6 | 0.8820 | 0.9972 | 0.9360 |
| | 0.7 | 0.8820 | 0.9972 | 0.9360 |
| | 0.8 | 0.8795 | 0.9945 | 0.9335 |
| | 0.9 | 0.8698 | 0.9836 | 0.9232 |
| | **W.Avg** | **0.8782** | **0.9930** | **0.9321** |

**Table 3.** HybridTabNet results on each document type of TableBank dataset. W.Avg denotes weighted-average of the respective measure on IoU threshold.

| Dataset | IOU Threshold | F1-score |
|---|---|---|
| Latex | 0.5 | 0.9805 |
| | 0.6 | 0.9798 |
| | 0.7 | 0.9782 |
| | 0.8 | 0.9718 |
| | 0.9 | 0.9346 |
| | **W.Avg** | **0.9661** |
| Word | 0.5 | 0.9702 |
| | 0.6 | 0.9678 |
| | 0.7 | 0.9655 |
| | 0.8 | 0.9640 |
| | 0.9 | 0.9624 |
| | **W.Avg** | **0.9654** |
| Both | 0.5 | 0.9749 |
| | 0.6 | 0.9726 |
| | 0.7 | 0.9706 |
| | 0.8 | 0.9671 |
| | 0.9 | 0.9495 |
| | **W.Avg** | **0.9653** |

Table 3 summarizes the results of HybridTabNet on TableBank dataset. It achieves the highest f1-score of 0.9805 on the 0.5 IoU threshold in Latex documents. There is only a slight drop in the f1-score of HybridTabNet from IoU thresholds of 0.6 to 0.8, which shows that its performance is not limited to lower IoU thresholds. However, at the IoU threshold of 0.9, there is a significant drop in the performance of our approach. Similarly, in Word and a mixture of Latex and Word documents, the f1-score almost remains constant from 0.5 to 0.8. For the IoU threshold of 0.9, the f1-score of Word is similar to lower thresholds, i.e. from 0.5 to 0.8. Conversely, for the mixture of Latex and Word, the f1-score at 0.9 is low.

Figures 6, 7 and 8 illustrates the results on Latex, Word and the mixture of Latex and Word, respectively. Figures 6a , 7a. and 8a show the correct predictions of HybridTabNet. Figure 6b shows the over-segmentation of tables, whereas Figure 6c displays the table which our approach failed to detect. Figures 7b and 8b show the case where a figure which is similar to table is predicted as table. Analysis of Figures 7c and 8c, shows that our approach faces difficulty when the table has no border.

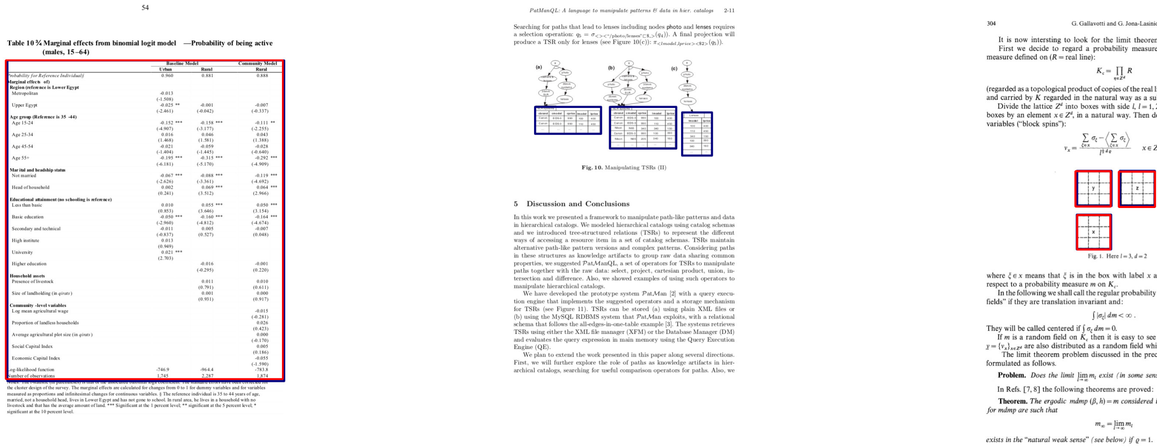
### 6.4. Marmot

Marmot [63] dataset comprises of English and Chinese documents. We perform a mixture of both types to create a new dataset. We use HybridTabNet, which is finetuned on the ICDAR-17 dataset, to extract results on the mixed Marmot dataset. Table 4 shows the results of HybridTabNet on Marmot dataset. Our approach achieves the highest precision of 0.9624 and recall of 0.9612 at the IoU threshold of 0.5. There is a slight decrease in precision and recall for IoU thresholds of 0.6 and 0.7. However, IoU thresholds 0.8 and 0.9 achieve lower precision and recall than other IoU thresholds due to the mixed language of documents.

**Table 4.** HybridTabNet results on mixed (Chinese + English) Marmot dataset. W.Avg denotes weighted-average of the respective measure on IoU threshold.

| IOU Threshold | Precision | Recall | F1-score |
|---|---|---|---|
| 0.5 | 0.9624 | 0.9612 | 0.9568 |
| 0.6 | 0.9521 | 0.9556 | 0.9539 |
| 0.7 | 0.9471 | 0.9505 | 0.9488 |
| 0.8 | 0.9347 | 0.9381 | 0.9364 |
| 0.9 | 0.9000 | 0.9031 | 0.9016 |
| **W.Avg** | **0.9351** | **0.9378** | **0.9358** |

Figure 9 shows the qualitative results of HybridTabNet on mixed Marmot dataset. Figure 9b shows the under-segmentation of the tables, whereas in Figure 9c our approach fails to detect two tables without proper boundaries.



Positive/POD$_0$816.*jpg*

**(a)** True Positive.

Positive/POD$_0$188.*jpg*

**(b)** False Positive.

Negative/POD$_0$806.*jpg*

**(c)** False Negative.

**Figure 5.** HybridTabNet results on ICDAR-2017 dataset. **a)**, **b)** and **c)** shows True positive, False positive and False negative results respectively. Blue outline shows the predicted table and red outline highlights ground truth table.

Positive/1401.0022$_1$6.$jpg$ Positive/1401.0308$_5$7.$jpg$ Negative/1401.1392$_8$.$jpg$

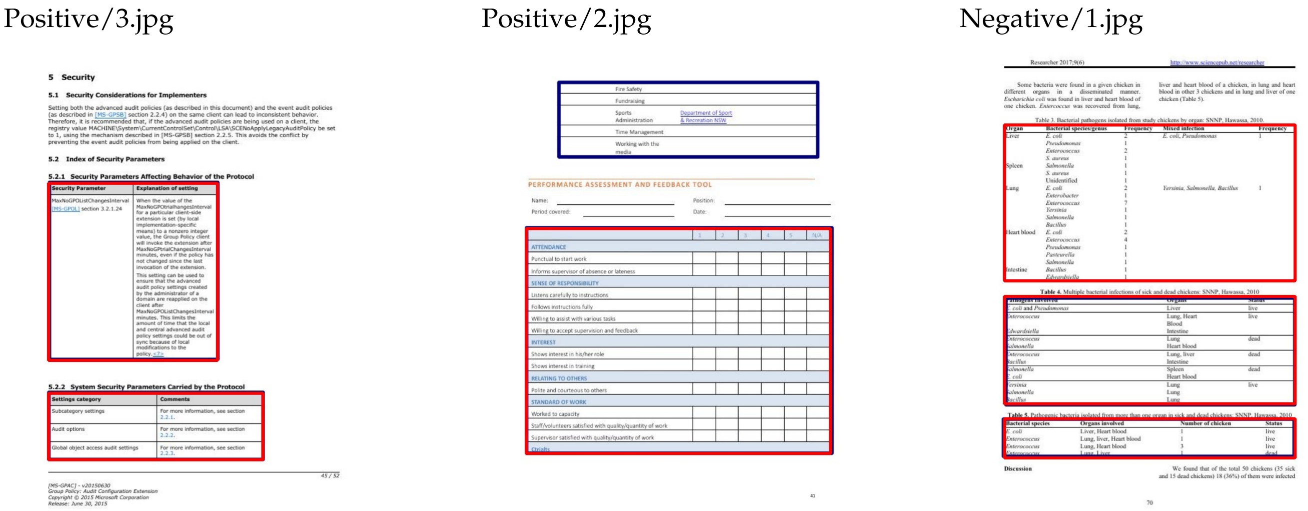
**(a)** True Positive. **(b)** False Positive. **(c)** False Negative.

**Figure 6.** HybridTabNet results on TableBank Latex dataset. **a)**, **b)** and **c)** shows True positive, False positive and False negative results respectively. Blue outline shows the predicted table and red outline highlights ground truth table.

### 6.5. UNLV

We finetune HybridTabNet on UNLV [70] dataset. Table 5 illustrates the results of HybridTabNet on UNLV dataset. Our approach achieves the highest precision of 0.9268 and recall of 0.9620 on the IoU threshold of 0.5. There is a slight decrease in precision and recall for IoU thresholds of 0.6, 0.7 and 0.8. However, at 0.9 IoU, our approach achieves the precision of 0.7921 and recall of 0.8228, which is worse than lower thresholds. We manually inspected the failure cases and found that the boundaries of tables and figures are ambiguous in the dataset. Such cases are shown in Figure 10.

Figure 10 depicts the results of HybridTabNet on UNLV dataset. Figure 10b shows the False Positive predicted by our model in which our approach performs over-segmentation of the table. Figure 10c illustrates the False Negative result, which shows that our approach fails to detect boundary-less tables.



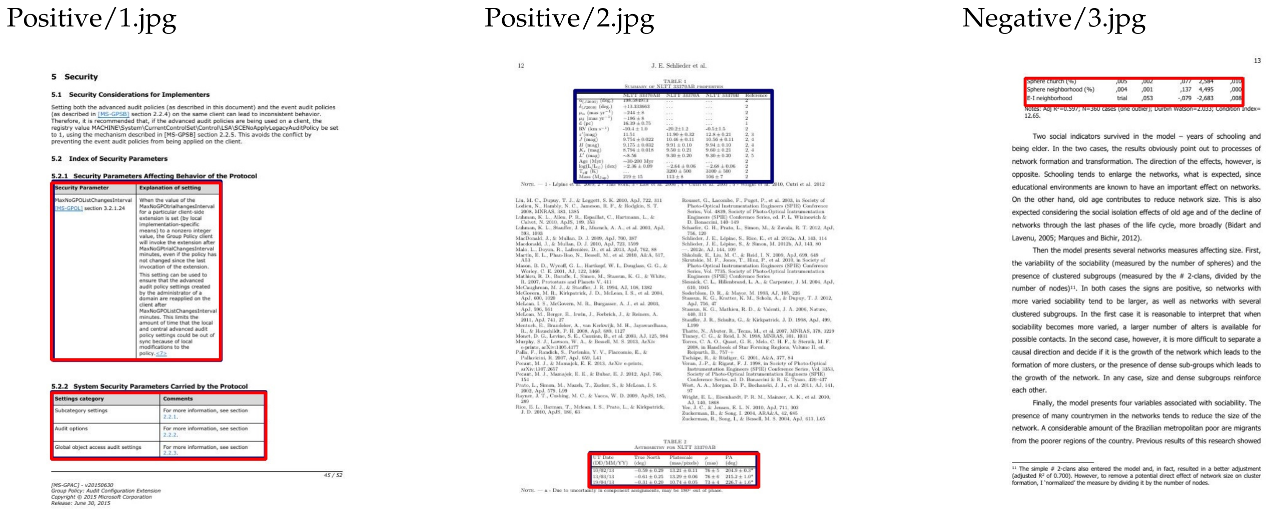
Positive/3.jpg Positive/2.jpg Negative/1.jpg

**(a)** True Positive. **(b)** False Positive. **(c)** False Negative.

**Figure 7.** HybridTabNet results on TableBank Word dataset. **a)**, **b)** and **c)** shows True positive, False positive and False negative results respectively. Blue outline shows the predicted table and red outline highlights ground truth table.

**Table 5.** HybridTabNet results on UNLV dataset. W.Avg denotes weighted-average of the respective measure on IoU threshold.

| IOU Threshold | Precision | Recall | F1-score |
|---|---|---|---|
| 0.5 | 0.9268 | 0.9620 | 0.9440 |
| 0.6 | 0.9146 | 0.9493 | 0.9316 |
| 0.7 | 0.9146 | 0.9493 | 0.9316 |
| 0.8 | 0.9024 | 0.9367 | 0.9192 |
| 0.9 | 0.7921 | 0.8228 | 0.8074 |
| **W.Avg** | **0.8820** | **0.9157** | **0.8986** |



**(a)** True Positive.

**(b)** False Positive.

**(c)** False Negative.

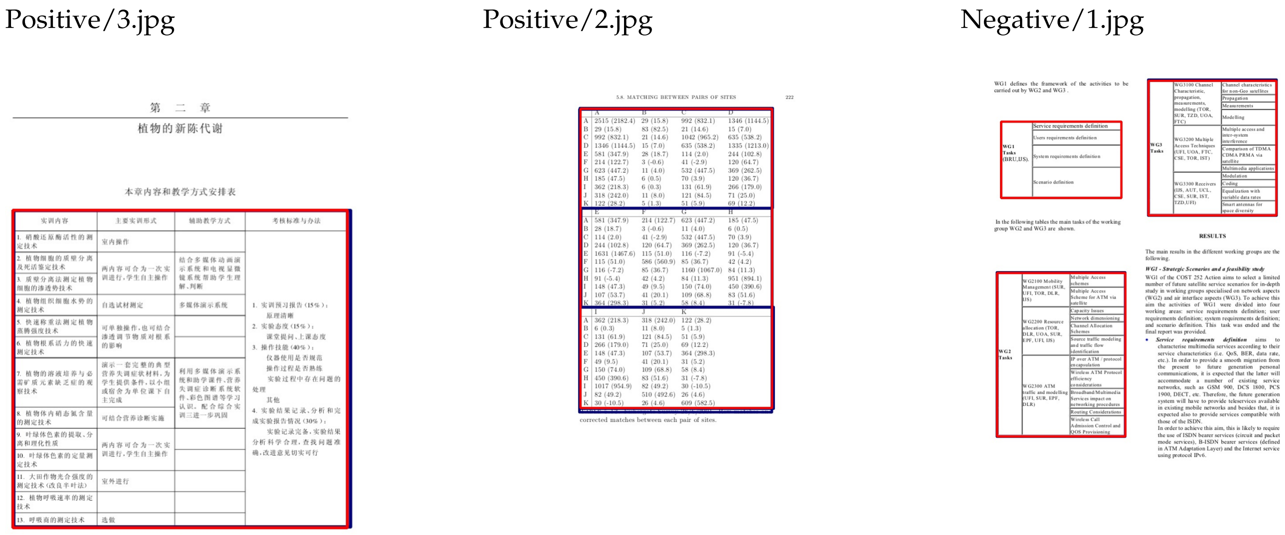
**Figure 8.** HybridTabNet results on TableBank Both (Latex + Word) dataset. **a)**, **b)** and **c)** shows True positive, False positive and False negative results respectively. Blue outline shows the predicted table and red outline highlights ground truth table.

### 6.6. *Comparison with state-of-the-art approaches*

#### 6.6.1. ICDAR-19

Table 6 summarizes the comparison of HybridTabNet and current state-of-the-art approaches [8] [71] on ICDAR-2019-TrackA-Modern dataset. Our approach achieves the W.Avg of 0.9283, which is close to the state-of-the-art performance. It does not beat the state-of-the-art performance of table region detection because they utilize image processing techniques like Dilation and Smudging for effective learning. The advantage of our approach is that we do not use any image preprocessing or post-processing techniques on the data. Our model directly takes the raw training data without any image preprocessing technique and learns effective representations. It outputs the accurate table region masks and bounding boxes directly at inference time due to its architectural design and the techniques implanted during its training.

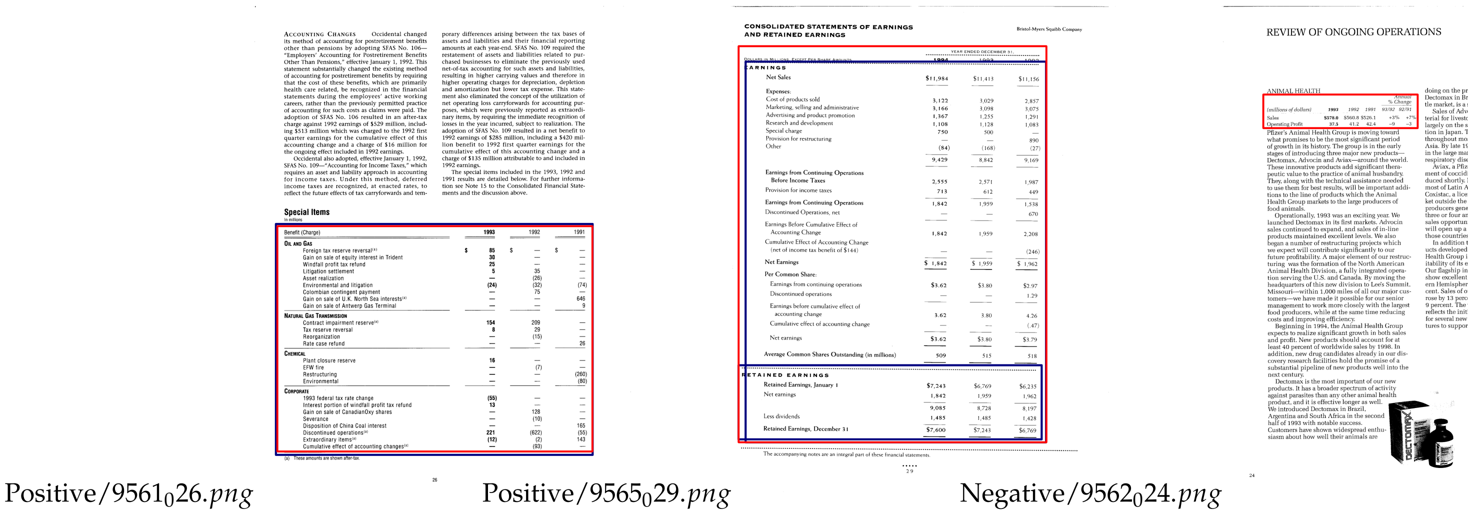
The aforementioned points make our technique much better than the earlier state-of-the-art methods. Even without utilizing any image preprocessing or post-processing techniques, our model achieves better performance than current state-of-the-art approaches.



Positive/3.jpg Positive/2.jpg Negative/1.jpg

**(a)** True Positive. **(b)** False Positive. **(c)** False Negative.

**Figure 9.** HybridTabNet results on Marmot dataset. **a)**, **b)** and **c)** shows True positive, False positive and False negative results respectively. Blue outline shows the predicted table and red outline highlights ground truth table.



Positive/9561$_0$26.*png* Positive/9565$_0$29.*png* Negative/9562$_0$24.*png*

**(a)** True Positive. **(b)** False Positive. **(c)** False Negative.

**Figure 10.** HNet results on UNLV dataset. **a)**, **b)** and **c)** shows True positive, False positive and False negative results respectively. Blue outline shows the predicted table and red outline highlights ground truth table.

### 6.6.2. ICDAR-17

From Table 6, we can observe that the current state-of-the-art approaches [7] [12] [71] for ICDAR-17 POD dataset are evaluated only on 0.6 and 0.8 IoU thresholds. We achieve the f1-scores of 0.9360 and 0.9335 on IoU thresholds 0.6 and 0.8. If we directly compare our approach results on the mentioned IoU thresholds with current state-of-the-art methods, it is apparent that we do not achieve state-of-the-art performance. It is because our method produces a comparatively lower precision score on the IoU thresholds mentioned above. However, the recall of our approach is near to 1 for IoU thresholds 0.5 to 0.7. We also provide results at 0.5,0.8 and 0.9 IoU thresholds for the sake of completeness and future bench-marking.

**Table 6.** Comparison of HybridTabNet on f1-scores with previous state-of-the-art methods. Our proposed method achieves state-of-the-art performance on every dataset except ICDAR-2017-POD. W.Avg denotes weighted-average of the respective measure on IoU threshold.

| **Dataset** | **Method** | **IOU** | | | | | **W.Avg** |
|---|---|---|---|---|---|---|---|
| | | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** | |
| ICDAR-2019-TrackA-Modern | TableRadar [71] | – | 0.969 | 0.957 | 0.951 | 0.897 | 0.940 |
| | NLPR-PAL [71] | – | 0.979 | 0.966 | 0.939 | 0.850 | 0.927 |
| | Cascade-TabNet [8] | – | 0.943 | 0.934 | 0.925 | 0.901 | 0.901 |
| | Ours | **0.9532** | **0.9424** | **0.9336** | **0.9278** | **0.9017** | **0.9283** |
| ICDAR-2017-POD | Fast Detectors [72] | – | 0.921 | – | 0.896 | – | – |
| | PAL [72] | – | 0.960 | – | 0.951 | – | – |
| | GOD [12] | – | 0.971 | – | 0.968 | – | – |
| | DeCNT [7] | – | 0.968 | – | 0.952 | – | – |
| | Ours | **0.9360** | 0.9360 | 0.9360 | 0.9335 | 0.9232 | 0.9321 |
| ICDAR-2013 | Cascade-TabNet [8] | 1.0 | – | – | – | – | – |
| | Ours | **1.0** | – | – | – | – | – |
| TableBank(Latex) | Cascade-TabNet [8] | 0.9660 | – | – | – | – | – |
| | Ours | **0.9805** | **0.9798** | **0.9782** | **0.9718** | **0.9346** | **0.9661** |
| TableBank(Word) | Cascade-TabNet [8] | 0.9492 | – | – | – | – | – |
| | Ours | **0.9702** | **0.9678** | **0.9655** | **0.9640** | **0.9624** | **0.9654** |
| TableBank(Both) | Cascade-TabNet [8] | 0.9433 | – | – | – | – | – |
| | Ours | **0.9749** | **0.9726** | **0.9706** | **0.9671** | **0.9495** | **0.9653** |
| Marmot | DeCNT [7] | 0.895 | – | – | – | – | – |
| | CDeC-Net [9] | 0.952 | – | – | 0.840 | 0.769 | – |
| | Ours | **0.9568** | **0.9539** | **0.9488** | **0.9364** | **0.9016** | **0.9358** |
| UNLV | GOD [12] | 0.928 | – | – | – | – | – |
| | CDeC-Net [9] | 0.938 | 0.883 | – | – | – | – |
| | Ours | **0.9440** | **0.9316** | **0.9316** | **0.9192** | **0.8074** | **0.8986** |

#### 6.6.3. ICDAR-13

Table 6 shows the results of HybridTabNet on ICDAR-2013 [60] dataset. The current state-of-the-art approach [8] has already achieved the perfect f1-score of 1.0 at 0.5 IoU threshold. Our approach achieves an f1-score of 1.0 at 0.5 IoU threshold, which is state-of-the-art performance.

#### 6.6.4. TableBank

TableBank [62] dataset consists of three subset datasets which are Latex, Word and a418mixture of Latex and Word documents. Table 6 shows the comparison of HybridTabNet and current state-of-the-art approach Cascade-TabNet [8]. Cascade-TabNet evaluates Latex, Word and a mixture of Latex and Word documents only at the IoU threshold of 0.5. It achieves f1-scores of 0.9660, 0.9492 and 0.9433 on Latex, Word and their mixture.

We also evaluate our approach on 0.5 and achieve f1-scores of 0.9805, 0.9702 and 0.9749 on Latex, Word and their mixture. If we directly compare the results, we achieve state-of-the-art performance on each subset of TableBank. Moreover, CascadeTabNet [8] apply line correction on the test data as an image postprocessing technique to improve their results. However, we do not use any such image preprocessing or postprocessing techniques. It makes our technique and approach far more superior than CascadeTabNet [8]. Furthermore, we also report results on 0.6, 0.7, 0.8, and 0.9 IoU thresholds, which can be used for future benchmarking on the dataset.

**Table 7.** Results of our leave one out dataset strategy. HybridTabNet achieve state-of-the-art performance on Marmot dataset. W.Avg denotes weighted-average of the respective measure on IoU threshold.

| **Train Dataset** | **Test Dataset** | **IOU** | | | | | **W.Avg** |
|---|---|---|---|---|---|---|---|
| | | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** | |
| UNLV + ICDAR-2013<br>+ TableBank (Both) + Marmot<br>+ ICDAR-2017-POD | ICDAR-2019-Modern | 0.8232 | 0.8080 | 0.7878 | 0.7676 | 0.7060 | 0.7706 |
| ICDAR-2019-TrackA-Modern + UNLV<br>+ TableBank (Both) + Marmot<br>+ ICDAR-2013 | ICDAR-2017-POD | 0.8950 | 0.8904 | 0.8848 | 0.8462 | 0.8139 | 0.8601 |
| ICDAR-2019-TrackA-Modern + UNLV<br>+ TableBank (Both) + Marmot<br>+ ICDAR-2017-POD | ICDAR-2013 | 0.8253 | 0.8253 | 0.7889 | 0.7671 | 0.6190 | 0.7516 |
| ICDAR-2019-TrackA-Modern + UNLV<br>+ TableBank (Word) + Marmot<br>+ ICDAR-2017-POD + ICDAR-2013 | TableBank(Latex) | 0.9514 | 0.9477 | 0.9444 | 0.9337 | 0.8667 | 0.9235 |
| ICDAR-2019-TrackA-Modern + UNLV<br>+ TableBank (Latex) + Marmot<br>+ ICDAR-2017-POD + ICDAR-2013 | TableBank(Word) | 0.9325 | 0.9263 | 0.9222 | 0.9171 | 0.9066 | 0.9191 |
| ICDAR-2019-TrackA-Modern + UNLV<br>+ Marmot + ICDAR-2017-POD<br>+ ICDAR-2013 | TableBank(Both) | 0.9490 | 0.9436 | 0.9402 | 0.9320 | 0.8861 | 0.9262 |
| ICDAR-2019-TrackA-Modern + UNLV<br>+ TableBank (Both) + UNLV<br>+ ICDAR-2017-POD | Marmot | 0.9623 | 0.9608 | 0.9565 | 0.9493 | 0.9290 | 0.9493 |
| ICDAR-2017-POD + Marmot<br>+ TableBank (Both) + ICDAR-2013<br>+ ICDAR-2019-TrackA-Modern | UNLV | 0.8089 | 0.7876 | 0.7664 | 0.7112 | 0.5031 | 0.8986 |

#### 6.6.5. Marmot

Table 6 illustrates the comparison of HybridTabNet and current state-of-the-art approaches DeCNT [7] and CDeC-Net [9]. DeCNT achieves the f1-score of 0.895 on 0.5 IoU threshold and CDeC-Net [9] achieves the f1-scores of 0.952, 0.840 and 0.769 on 0.5, 0.8 and 0.9 IoU thresholds. Similarly, Our approach achieves the f1-scores of 0.9568, 0.9364 and 0.9016 on 0.5, 0.8 and 0.9 IoU thresholds. The direct comparison of our results with CDeC-Net and DeCNT proves that we achieve state-of-the-art results on the Marmot dataset. We also evaluate our approach on 0.6, 0.7 IoU thresholds for future benchmarking on the dataset.

#### 6.6.6. UNLV

The current state-of-the-art approaches GOD [12] and CDeC-Net [9] on UNLV [70] are evaluated on IoU thresholds of 0.5 and 0.6. Table 6 presents the comparison of HybridTabNet and state-of-the-art approaches on the UNLV dataset. GOD achieves the f1-score of 0.928 on the 0.5 IoU threshold, whereas CDeC-Net achieves the f1-score of 0.938 and 0.883 on IoU thresholds of 0.5 and 0.6. For a direct comparison, we evaluate our approach from 0.5 to 0.9 IoU thresholds. We obtain f1-scores of 0.9440 and 0.9316 on 0.5 and 0.6 IoU thresholds, which is better than current state-of-the-art methods, thus achieving state-of-the-art performance on UNLV dataset .

### 6.7. Leave-one-out Evaluation

This section explores the employed evaluation strategy to measure the generalization and cross datasets performance of HybridTabNet. To the best of our knowledge, this is the first comprehensive cross datasets evaluation study which consists of several datasets. The idea is as follows: We combine all available datasets except one into a single dataset. This new dataset becomes our training dataset, whereas the dataset which is left out becomes our test dataset. In the case of ICDAR-2019-TrackA-Modern, other datasets such as ICDAR-2017-POD, ICDAR-2013, Marmot, UNLV, and TableBank are combined and become single training dataset whereas ICDAR-2019-TrackA-Modern becomes our test dataset. We repeat this process for all of the datasets, and performance evaluation is done on 0.5 to 0.9 IoU thresholds.

Table 7 presents the results of leave-one-out evaluation of HybridTabNet. The results are not promising for datasets including ICDAR-2013, ICDAR-2017-POD and ICDAR-2019-TrackA-Modern because the combined training datasets do not resemble the test dataset. Conversely, we achieve f1-scores of 0.9623, 0.9608, 0.9565, 0.9493, 0.9290, 0.9493 on 0.5 to 0.9 IoU thresholds for Marmot dataset. These results are better than the ones presented for Marmot in Table 6, and therefore it achieves state-of-the-art performance. Similarly, the results on TableBank and UNLV are also comparable to state-of-the-art results.

## 7. Conclusion and Future Work

This paper presents a novel approach, HybridTabNet, for table detection from scanned document images. The approach uses the ResNeXt-101 backbone for feature extraction and also replaces regular convolutions with deformable convolutions. The proposed approach is the Hybrid Task Cascade network for table detection that uses cascade architecture, for instance, segmentation. Our method surpasses existing state-of-the-art table detection methods in all the datasets except for ICDAR-2017-POD. The relative improvement of error in terms of weighted average amounts to 27.57% for ICDAR-2019-TrackA-Modern, 42.64% for TableBank (Latex), 41.33% for TableBank (Word), 55.73% for TableBank (Latex + Word), 10% for Marmot, and 9.67% for UNLV. For ICDAR-2013, the proposed approach achieves a perfect score for precision and recall, which is on par with the previous state-of-the-art. However, for ICDAR-2017-POD, the proposed approach does not outperform the state-of-the-art methods. It is because ICDAR-2017-POD contains a lot of other graphical page components that are similar to tables. Other methods rely on pre-and/or post-processing to transform the data for favourable results. However, our approach works on raw images. Moreover, we incorporate the Leave-one-out evaluation for all the datasets that shows the algorithm's generalization capabilities—a direction for evaluating table detection algorithms to follow in the future.

An important future direction is the development of generalized table detection methods that can work with various types of tables instead of being tuned for a specific dataset. We plan to extend this work to create a unified representation that eliminates the pre-and post-processing steps. Moreover, another interesting direction can be to explore table structure recognition with the proposed approach.

**Author Contributions:** writing—original draft preparation, D.N., K.A.H., M.Z.A.; writing—review and editing, K.A.H., M.Z.A., M.L.; supervision and project administration, A.P., D.S. All authors have read and agreed to the submitted version of the manuscript.

**Funding:** The work leading to this publication has been partially funded by the European project INFINITY under Grant Agreement ID 883293.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gao, L.; Yi, X.; Jiang, Z.; Hao, L.; Tang, Z. ICDAR2017 competition on page object detection. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 1417–1422.
2. Zhao, Z.; Jiang, M.; Guo, S.; Wang, Z.; Chao, F.; Tan, K.C. Improving deep learning based optical character recognition via neural architecture search. 2020 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2020, pp. 1–7.
3. Hashmi, K.A.; Ponnappa, R.B.; Bukhari, S.S.; Jenckel, M.; Dengel, A. Feedback Learning: Automating the Process of Correcting and Completing the Extracted Information. 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). IEEE, 2019, Vol. 5, pp. 116–121.
4. van Strien, D.; Beelen, K.; Ardanuy, M.C.; Hosseini, K.; McGillivray, B.; Colavizza, G. Assessing the Impact of OCR Quality on Downstream NLP Tasks. ICAART (1), 2020, pp. 484–496.
5. Bhatt, J.; Hashmi, K.A.; Afzal, M.Z.; Stricker, D. A Survey of Graphical Page Object Detection with Deep Neural Networks. *Applied Sciences* **2021**, *11*, 5344.
6. Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; Ahmed, S. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. 2017 14th IAPR international conference on document analysis and recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 1162–1167.
7. Siddiqui, S.A.; Malik, M.I.; Agne, S.; Dengel, A.; Ahmed, S. Decnt: Deep deformable cnn for table detection. *IEEE Access* **2018**, *6*, 74151–74161.
8. 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. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 572–573.
9. Agarwal, M.; Mondal, A.; Jawahar, C. CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images. *arXiv preprint arXiv:2008.10831* **2020**.
10. Huang, Y.; Yan, Q.; Li, Y.; Chen, Y.; Wang, X.; Gao, L.; Tang, Z. A YOLO-based table detection method. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 813–818.
11. Gilani, A.; Qasim, S.R.; Malik, I.; Shafait, F. Table detection using deep learning. 2017 14th IAPR international conference on document analysis and recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 771–776.
12. Saha, R.; Mondal, A.; Jawahar, C. Graphical object detection in document images. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 51–58.
13. Coüasnon, B.; Lemaitre, A. Recognition of tables and forms, 2014.
14. Girshick, R. Fast r-cnn. Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497* **2015**.
16. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
17. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.
18. Paliwal, S.S.; Vishwanath, D.; Rahul, R.; Sharma, M.; Vig, L. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 128–133.
19. Hashmi, K.A.; Stricker, D.; Liwicki, M.; Afzal, M.N.; Afzal, M.Z. Guided Table Structure Recognition through Anchor Optimization. *arXiv preprint arXiv:2104.10538* **2021**.
20. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; others. Hybrid task cascade for instance segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4974–4983.
21. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks, 2017, [arXiv:cs.CV/1703.06211].
22. Itonori, K. Table structure recognition based on textblock arrangement and ruled line position. Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93). IEEE, 1993, pp. 765–768.
23. Tupaj, S.; Shi, Z.; Chang, C.H.; Alam, H. Extracting tabular information from text files. *EECS Department, Tufts University, Medford, USA* **1996**.
24. Chandran, S.; Kasturi, R. Structural recognition of tabulated data. Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93). IEEE, 1993, pp. 516–519.
25. Hirayama, Y. A method for table structure analysis using DP matching. Proceedings of 3rd International Conference on Document Analysis and Recognition. IEEE, 1995, Vol. 2, pp. 583–586.
26. Green, E.; Krishnamoorthy, M. Recognition of tables using table grammars. Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval, 1995, pp. 261–278.
27. Kieninger, T.G. Table structure recognition based on robust block segmentation. Document Recognition V. International Society for Optics and Photonics, 1998, Vol. 3305, pp. 22–32.
28. Casado-García, Á.; Domínguez, C.; Heras, J.; Mata, E.; Pascual, V. The benefits of close-domain fine-tuning for table detection in document images. International Workshop on Document Analysis Systems. Springer, 2020, pp. 199–215.

doi:10.20944/preprints202108.0360.v1

29. Sun, N.; Zhu, Y.; Hu, X. Faster R-CNN based table detection combining corner locating. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 1314–1319.
30. Vo, N.D.; Nguyen, K.; Nguyen, T.V.; Nguyen, K. Ensemble of deep object detectors for page object detection. Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, 2018, pp. 1–6.
31. Mondal, A.; Lipps, P.; Jawahar, C. IIIT-AR-13K: a new dataset for graphical object detection in documents. International Workshop on Document Analysis Systems. Springer, 2020, pp. 216–230.
32. Hashmi, K.A.; Liwicki, M.; Stricker, D.; Afzal, M.A.; Afzal, M.A.; Afzal, M.Z. Current Status and Performance Analysis of Table Recognition in Document Images with Deep Neural Networks. *IEEE Access* **2021**.
33. Pyreddy, P.; Croft, W.B. Tintin: A system for retrieval in text tables. Proceedings of the second ACM international conference on Digital libraries, 1997, pp. 193–200.
34. Pivk, A.; Cimiano, P.; Sure, Y.; Gams, M.; Rajkovič, V.; Studer, R. Transforming arbitrary tables into logical form with TARTAR. *Data & Knowledge Engineering* **2007**, *60*, 567–595.
35. Hu, J.; Kashi, R.S.; Lopresti, D.P.; Wilfong, G. Medium-independent table detection. Document Recognition and Retrieval VII. International Society for Optics and Photonics, 1999, Vol. 3967, pp. 291–302.
36. Zanibbi, R.; Blostein, D.; Cordy, J.R. A survey of table recognition. *Document Analysis and Recognition* **2004**, *7*, 1–16.
37. e Silva, A.C.; Jorge, A.M.; Torgo, L. Design of an end-to-end method to extract information from tables. *International Journal of Document Analysis and Recognition (IJDAR)* **2006**, *8*, 144–171.
38. Khusro, S.; Latif, A.; Ullah, I. On methods and tools of table detection, extraction and annotation in PDF documents. *Journal of Information Science* **2015**, *41*, 41–57.
39. Embley, D.W.; Hurst, M.; Lopresti, D.; Nagy, G. Table-processing paradigms: a research survey. *International Journal of Document Analysis and Recognition (IJDAR)* **2006**, *8*, 66–86.
40. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv preprint arXiv:1611.05431* **2016**.
41. Kieninger, T.; Dengel, A. The t-recs table recognition and analysis system. International Workshop on Document Analysis Systems. Springer, 1998, pp. 255–270.
42. Oro, E.; Ruffolo, M. TREX: An approach for recognizing and extracting tables from PDF documents. 2009 10th International Conference on Document Analysis and Recognition. IEEE, 2009, pp. 906–910.
43. Fan, M.; Kim, D.S. Detecting table region in PDF documents using distant supervision. *arXiv preprint arXiv:1506.08891* **2015**.
44. Cesarini, F.; Marinai, S.; Sarti, L.; Soda, G. Trainable table location in document images. Object recognition supported by user interaction for service robots. IEEE, 2002, Vol. 3, pp. 236–240.
45. e Silva, A.C. Learning rich hidden markov models in document analysis: Table location. 2009 10th International Conference on Document Analysis and Recognition. IEEE, 2009, pp. 843–847.
46. Silva, A. Parts that add up to a whole: a framework for the analysis of tables. *Edinburgh University, UK* **2010**.
47. Kasar, T.; Barlas, P.; Adam, S.; Chatelain, C.; Paquet, T. Learning to detect tables in scanned document images using line information. 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013, pp. 1185–1189.
48. Hao, L.; Gao, L.; Yi, X.; Tang, Z. A table detection method for pdf documents based on convolutional neural networks. 2016 12th IAPR Workshop on Document Analysis Systems (DAS). IEEE, 2016, pp. 287–292.
49. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. European conference on computer vision. Springer, 2014, pp. 818–833.
50. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
51. Zhong, X.; Tang, J.; Yepes, A.J. Publaynet: largest dataset ever for document layout analysis. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 1015–1022.
52. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
53. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. European conference on computer vision. Springer, 2016, pp. 21–37.
54. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.
55. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **2012**, *25*, 1097–1105.
56. Zhao, Z.Q.; Zheng, P.; tao Xu, S.; Wu, X. Object Detection with Deep Learning: A Review, 2019, [arXiv:cs.CV/1807.05511].
57. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. doi:10.1109/access.2019.2939201.
58. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image Segmentation Using Deep Learning: A Survey, 2020, [arXiv:cs.CV/2001.05566].
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition, 2015, [arXiv:cs.CV/1512.03385].
60. Göbel, M.; Hassan, T.; Oro, E.; Orsi, G. ICDAR 2013 table competition. 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013, pp. 1449–1453.

61. Gao, L.; Huang, Y.; Déjean, H.; Meunier, J.L.; Yan, Q.; Fang, Y.; Kleber, F.; Lang, E. ICDAR 2019 competition on table detection and recognition (cTDaR). 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 1510–1515.
62. Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M.; Li, Z. Tablebank: Table benchmark for image-based table detection and recognition. Proceedings of The 12th Language Resources and Evaluation Conference, 2020, pp. 1918–1925.
63. Fang, J.; Tao, X.; Tang, Z.; Qiu, R.; Liu, Y. Dataset, ground-truth and performance metrics for table detection evaluation. 2012 10th IAPR International Workshop on Document Analysis Systems. IEEE, 2012, pp. 445–449.
64. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. Proceedings of the 24th ACM international conference on Multimedia, 2016, pp. 516–520.
65. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 658–666.
66. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C.C.; Lin, D. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155* **2019**.
67. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H.; Larochelle, H.; Beygelzimer, A.; dAlché-Buc, F.; Fox, E.; Garnett, R., Eds.; Curran Associates, Inc., 2019; pp. 8024–8035.
68. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context, 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.
69. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection, 2017, [arXiv:cs.CV/1612.03144].
70. Shahab, A.; Shafait, F.; Kieninger, T.; Dengel, A. An open approach towards the benchmarking of table structure recognition systems. Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, 2010, pp. 113–120.
71. Gao, L.; Huang, Y.; Déjean, H.; Meunier, J.L.; Yan, Q.; Fang, Y.; Kleber, F.; Lang, E. ICDAR 2019 competition on table detection and recognition (cTDaR). 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 1510–1515.
72. Gao, L.; Yi, X.; Jiang, Z.; Hao, L.; Tang, Z. ICDAR2017 competition on page object detection. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 1417–1422.