CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images

by

Madhava Krishna, Ajoy Mondal, C V Jawahar

in

ICPR

Report No: IIIT/TR/2020/-1

Centre for Robotics
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2020

CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images

Madhav Agarwal
CVIT, IIIT, Hyderabad, India
madhav14130@gmail.com

Ajoy Mondal
CVIT, IIIT, Hyderabad, India
ajoy.mondal@iiit.ac.in

C. V. Jawahar
CVIT, IIIT, Hyderabad, India
jawahar@iiit.ac.in

***Abstract*—Localizing page elements/objects such as tables, figures, equations, etc. is the primary step in extracting information from document images. We propose a novel end-to-end trainable deep network, (CDeC-Net) for detecting tables present in the documents. The proposed network consists of a multistage extension of Mask R-CNN with a dual backbone having deformable convolution for detecting tables varying in scale with high detection accuracy at higher IoU threshold. We empirically evaluate CDeC-Net on the publicly available benchmark datasets with extensive experiments.**

Our solution has three important properties: (i) a single trained model CDeC-Net‡ that performs well across all the popular benchmark datasets; (ii) we report excellent performances across multiple, including higher, thresholds of IoU; (iii) by following the same protocol of the recent papers for each of the benchmarks, we consistently demonstrate the superior quantitative performance. Our code and models are publicly available at https://github.com/mdv3101/CDeCNet for enabling reproducibility of the results.

***Keywords*—** Page object, table detection, Cascade Mask R-CNN, deformable convolution, single model.

I. Introduction

Rapid growth in information technology has led to an exponential increase in production and storage of digital documents over the last few decades. Extracting information from such a large corpus is impractical for humans. Hence, useful information could be lost or not utilized over time. Digital documents have many other page objects (such as tables and figures) beyond the text. These page objects also show wide variations in their appearance. Therefore any attempt to detect page objects such as tables need to be generic and applicable across a wide variety of documents and use cases. In this paper, we are interested in the detection of tables. It is well known [1]–[11] that the localisation of tables and other page element is challenging due to the high degree of intra-class variability (due to different layouts of the table, inconsistent use of ruling lines). The presence of inter-class similarity (graphs, flowcharts, figures having large number of horizontal and vertical lines which resembles to table) adds further challenges.

Table detection is still a challenging problem in the research community. This is an active area of research [3]–[11]. However, we observe that most of these attempts develop different table detection solutions for different datasets. We argue that this may be the time to consider the possibility of a single solution (say a trained model) that works across wide variety of documents. We provide a single model CDeC-Net‡ trained with IIIT-AR-13K dataset [12] and evaluate on popular benchmark datasets. Table I shows the comparison with the state-of-the-art techniques for respective datasets. We observe from the table that our single model CDeC-Net‡ performs better than state-of-the-art techniques for ICDAR-2019 (cTDaR) [13], UNLV [14], and PubLayNet [9] datasets. In case of ICDAR-2013 [15], ICDAR-POD-2017 [16], Marmot [17], and TableBank [7], single model CDeC-Net‡ obtains comparable results to the state-of-the-art techniques. By following the same protocol of the state-of-the-art papers, we also report superior performance consistently across all the datasets, as presented in Table III and will discuss later in this paper.

Dataset	Method	Score			
		R↑	P↑	F1↑	mAP↑
ICDAR-2013	DeCNT [3]	**0.996***	**0.996***	**0.996***	-
	CDeC-Net‡ (our)	0.942	0.993	0.968	**0.942**
ICADR-2017	YOLOv3 [18]	**0.968**	**0.975**	**0.971**	-
	CDeC-Net‡ (our)	0.899	0.969	0.934	0.880
ICADR-2019	TableRadar [13]	**0.940**	0.950	0.945	-
	CDeC-Net‡ (our)	0.930	**0.971**	**0.950**	**0.913**
UNLV	GOD [10]	0.910	0.946	0.928	-
	CDeC-Net‡ (our)	**0.915**	**0.970**	**0.943**	**0.912**
Marmot	DeCNT [3]	**0.946**	0.849	**0.895**	-
	CDeC-Net‡ (our)	0.779	**0.943**	0.861	**0.756**
TableBank	Li et al. [7]	**0.975**	0.987	**0.981**	-
	CDeC-Net‡ (our)	0.970	**0.990**	0.980	**0.965**
PubLayNet	M-RCNN [9]	-	-	-	0.960
	CDeC-Net‡ (our)	**0.975**	**0.993**	**0.984**	**0.978**

TABLE I: Illustrates comparison between our single model CDeC-Net‡ and state-of-the-art techniques on existing benchmark datasets. We create the single model CDeC-Net‡ by training CDeC-Net with IIIT-AR-13K and fine-tuning with training set of respective datasets. ***:** indicates the authors reported 0.996 in table however in discussion they mentioned 0.994.

Early attempts in localizing tables are based on meta-data extraction and exploitation of the semantic information present in the tables [19]–[21]. However, the absence of meta-data in the case of scanned documents makes these methods futile. In recent years, researchers employ deep neural networks [1]–[11] in an attempt to provide a generic solution for localizing page objects, specifically tables from document images. Siddiqui *et al.* [3] provide state-of-the-art performance on many benchmark datasets by incorporating deformable convolutions [22] in their network. However, even their work is limited by the ability to provide a single model that achieves state-of-the-art performance on all the existing benchmark datasets. In general, the existing deep learning models are trained on a single IoU threshold, commonly 0.5, following the practice followed in computer vision literature. It leads to a noisy table detection at a higher threshold value during evaluation. It is a drawback of the existing table detection techniques. Liu *et al.* discuss in [23] that generally, a CNN based object detector uses a backbone network to extract features for detecting objects. These backbones are usually designed for the image classification task and are pre-trained on either ImageNet [24] or MS-COCO [25] datasets. Hence, directly employing them to extract features for table detection [1]–[11] may result in sub-optimal performance. Training a more powerful backbone is also

expensive. It is a major bottleneck of these existing table detection techniques.

To address the issues mentioned above, we propose a composite deformable cascade network, called as CDeC-Net, to detect tables more accurately present in document images. The proposed CDeC-Net consists of a multi-stage object detection architecture, cascade Mask R-CNN [26]. The cascade Mask R-CNN network is composed of a sequence of detectors trained with increasing IoU thresholds to address the problem of noisy detection at higher threshold. Inspired by [23] we use composite backbone, which consists of multiple identical backbones having composite connections between neighbor backbones, in CDeC-Net to improve detection accuracy. We also incorporate deformable convolution [22] in the backbones to model geometric transformations. We extensively evaluate CDeC-Net on publicly available benchmark datasets — ICDAR-2013, ICDAR-POD-2017, UNLV, Marmot, ICDAR-2019 (cTDaR), TableBank, and PubLayNet under various existing experimental environments. The extensive experiments show that CDeC-Net achieves state-of-the-art performance on all existing benchmark datasets except ICDAR-2017. We also achieve high accuracy and more tight bounding box detection at higher IoU threshold than the previous benchmark results.

We summarise our main contributions as follows:

- We present an end-to-end trainable deep architecture, CDeC-Net which consists of Cascade Mask R-CNN containing composite backbones with deformable convolution to detect tables more accurately in document images.
- We provide a single model trained on IIIT-AR-13K and achieve very close competitive results to the state-of-the-art techniques on all existing benchmark datasets (Refer Table I).
- We achieve state-of-the-art results on publicly available benchmark datasets except ICDAR-2017 (Refer Table III).

II. Related Work

Table detection is an essential step towards document analysis. Over the times, many researchers have contributed to the detection of tables in documents of varying layouts. Initially, the researchers have proposed several approaches based on heuristics or meta-data information to solve this particular problem [19], [21], [27]–[32]. Later, the researchers explore machine learning, more specifically deep learning, to make the solution generic [1]–[11].

A. Rule Based Approaches

The research on table detection in document images was started in 1993. In the beginning, Itonori [19] proposed a rule-based approach that led to the text-block arrangement and ruled line position to localize the table in the documents. At the same time, Chandran and Kasturi [27] developed a table detection approach based on vertical and horizontal lines. Following these, several research works [21], [28]–[32] have been done for table detection using improved heuristic rules. Though these methods perform well on the documents having limited layouts, they need more manual efforts to find a better heuristic rule. Moreover, rule-based approaches fail to obtain generic solutions. Therefore, it is necessary to employ machine learning approaches to solve the table detection problem.

B. Learning Based Approaches

Statistical learning approaches have been proposed to alleviate the problems mentioned earlier in table detection. Kieninger and Dengel [33] applied an unsupervised learning approach for the table detection task. This method significantly differs from the previous rule-based approaches [21], [28]–[32] as it uses a clustering of given word segments. Cesarini *et al.* [34] used a supervised learning approach using a hierarchical representation based on the MXY tree. This particular method detects the table with different features by maximizing the performance on a particular training set. Later, the solution of the table detection problem is formulated using various machine learning problems such as (i) sequence labeling [35], (ii) SVM with various hand-crafted features [36], and (iii) ensemble of various models [37]. Learning methods improve table detection accuracy significantly.

Dataset	Category Label	Training Set	Validation Set	Test Set
ICDAR-2013	1: T	170		238
ICDAR-POD-2017	3: T, F, and E	1600		817
UNLV	1: T			424
Marmot	1: T	2K		
ICDAR-2019 (cTDaR)	1: T	1200		439
TableBank-word[1]	1: T	163K	1K	1k
TableBank-LaTeX[1]	1: T	253K	1K	1k
TableBank-both[1]	1: T	417K	2K	2k
PubLayNet[1]	5: T, F, TL, TT, and LT	340K	11K	11K
IIIT-AR-13K	5: T, F, NI, L, and S	9K	2K	2k

TABLE II: Statistics of datasets. **T:** indicates table. **F:** indicates figure. **E:** indicates equation. **NI:** indicates natural image. **L:** indicates logo. **S:** indicates signature. **TL:** indicates title. **TT:** indicates text. **LT:** indicates list.

The success of deep convolutional neural network (CNN) in the field of computer vision, motivates researchers to explore CNN for localizing tables in the documents. It is a data-driven method and has advantages — (i) it is robust to document types and layouts, and (ii) it reduces the efforts of hand-crafted feature engineering in CNN. Initially, Hao *et al.* [38] used CNN to classify tables like structure regions extracted from PDFs using heuristic rule into two categories - table and non-table. The major drawbacks of this method are (i) use of the heuristic rule to extract table like region, and (ii) work on only non-raster PDF documents. The researchers explore various natural scene object detectors — Fast R-CNN [39] in [5], Faster R-CNN [40] in [1]–[9], Mask R-CNN [41] in [8]–[11], YOLO [42] in [11] to localize page objects more specifically tables in the document images. All these methods are data-driven and do not require any heuristics or meta-data to extract table like region similar to [38].

Gilani *et al.* [1] used Faster R-CNN to detect tables in the document images. Instead of the original document image, distance transformed image is taken as input to easily fine-tune the pre-trained model to work on various types of document images. In the same direction, the transformed document image is taken as input to Faster R-CNN model for detecting tables [6]; and figures and mathematical equations [8] present in document images. Saha *et al.* [10] experimentally established that Mask R-CNN performs better than Faster R-CNN for detecting graphical objects in the document images. Zhong *et al.* [9] also experimentally established that Mask R-CNN performs better than Faster R-CNN for extracting semantic regions from the documents. The performance of Faster R-CNN is reduced when documents contain large scale variate tables. Siddiqui *et al.* [3] incorporated deformable CNN in Faster R-CNN to adapt the different scales and transformations which allows the model to detect scale variate tables accurately. Sun *et al.* [4] combined the corner information with the detected table region by Faster R-CNN to refine the boundaries of the detected tables to reduce false positives. It is observed that every detection method is sensitive to a certain type of object. Vo *et al.* [5] combine outputs of two object detectors — Fast R-CNN and Faster R-CNN in order to exploit the advantages of the two models for page object detection. Due to the limited number of images in the existing training set, it is challenging to train such a detection model for table detection. Fine-tune is one solution to such a problem. In [11], the authors discuss the benefit of fine-tuning from a close domain on four different object detection models — Mask R-CNN [41], RetinaNet [43], SSD [44] and YOLO [42]. The experiments

[1]Ground truth bounding boxes are annotated automatically.

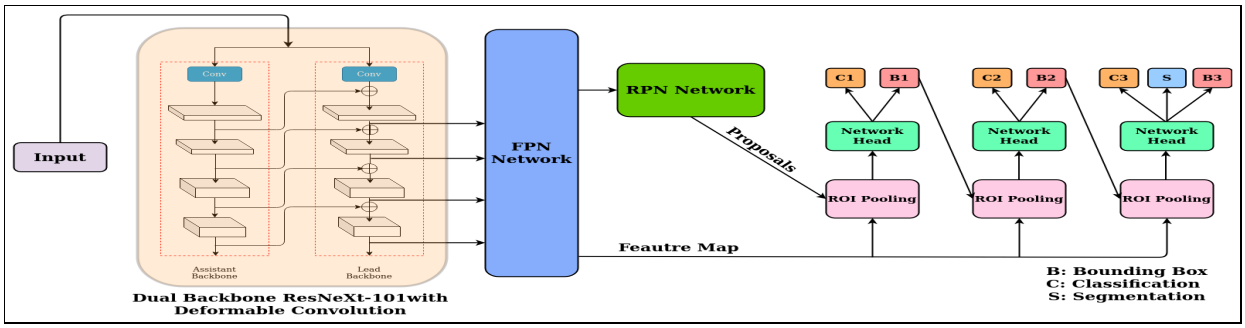

Fig. 1: Illustration of the proposed CDeC-Net which is compose of cascade Mask R-CNN with composite backbone having deformable convolution instead of conventional convolution.

highlight that the close domain fine-tuning approach avoids over-fitting, solves the problem of having a small training set and improves detection accuracy.

C. Related Datasets

Various benchmark datasets — ICDAR-2013 [15], ICDAR-POD-2017 [16], UNLV [14], Marmot [17], ICDAR-2019 (cTDaR) [13], TableBank [7], PubLayNet [9], and IIIT-AR-13K [12] are publicly available for table detection tasks. Table II shows the statistics of these datasets. Among them, ICDAR-2013, UNLV, Marmot, ICDAR-2019, TableBank are popularly used for table detection while ICDAR-POD-2017, PubLayNet, and IIIT-AR-13K datasets for various page object (including table) detection task. We use all datasets for our experiments.

III. CDeC-Net: Composite Deformable Cascade Network

The success of deep convolution neural networks (CNN)s for solving various computer vision problems inspire researchers to explore and design models for detecting tables in document images [1]–[11]. All these deep models provide high table detection accuracy. However, the previous table detection models suffer from the following shortcomings — (i) all existing table detection networks use a backbone to extract features for detecting tables, which is usually designed for image classification tasks and pre-trained on ImageNet dataset. Since almost all of the existing backbone networks are originally designed for the image classification task, directly applying them to extract features for table detection may result in sub-optimal performance. A more powerful backbone is needed to extract more representational features and improve the detection accuracy. However, it is very expensive to train a deeper and powerful backbone on ImageNet and get better performance. (ii) CNNs have limitations to model large transformation due to the fixed geometric structures of CNN modules — a convolution filter samples the input feature map correspond to a fixed location, a pooling layer reduces the spatial resolution at a fixed ration and a RoI into a fixed spatial bin, etc. This leads to a lack of handling geometric transformations. (iii) All these table detectors use the intersection over union (IoU) threshold to define positives, negatives, and finally, detection quality. They commonly use a threshold of 0.5, which leads to noisy (low-quality) detection and frequently degrades higher thresholds' performance. The major hindrance in training a network at a higher IoU threshold is the reduction of positive training samples with increasing IoU threshold. All these issues are also a bottleneck of CNNs based object detection techniques [39]–[41] in natural scene images.

Over the time, various solutions [22], [23], [26] are proposed to handle the above stated problems for object detection in natural images. Lie *et al.* [23] proposed CBNet which comprises of stacking multiple identical backbones by creating composite connections between them. It helps in creating a more powerful backbone for feature extraction without much additional computational cost. Dai *et al.* [22] introduced deformable convolution in the object detection network to make it more scale-invariant. It captures the features using a variable receptive field and makes detection independent of the fixed geometric transforms. Cai and Vasconcelos [26] proposed a multi-stage object detection architecture in which subsequent detectors are trained with increasing IoU thresholds to solve the last problem. One detector's output is fed as an input to the subsequent detector, maintaining the number of positive samples at higher thresholds.

Inspired by the solutions provided by [22], [23], [26] for issues discussed earlier in natural scene images, we propose a novel architecture CDeC-Net for detecting tables accurately in the document images. It is composed of Cascade Mask R-CNN with a composite backbone having deformable convolution filters instead of conventional convolution filters. Figure 1 displays an overview of our proposed architecture for table localization in document images. We discuss each component of CDeC-Net in detail:

A. Cascade Mask R-CNN

Cai and Vasconcelos [26] proposed Cascade R-CNN which is a multi-stage extension of Faster R-CNN [40]. Cascade Mask R-CNN has a similar architecture as Cascade R-CNN, but along with an additional segmentation branch, denoted by 'S', for creating masks of the detected objects. CDeC-Net comprises of a sequence of three detectors trained with increasing IoU thresholds of 0.5, 0.6, and 0.7, respectively. The proposals generated by RPN network are passed through ROI pooling layer. The network head takes ROI features as input and makes two predictions — classification score (C) and bounding box regression (B). The output of one detector is used as a training set for the next detector. The deeper detector stages are more selective against close false positives. Each regressor is optimized for the bounding box distribution generated by the previous regressor, rather than the initial distribution. The bounding box regressor trained for a certain IoU threshold, and it tends to produce bounding boxes of higher IoU threshold. It helps in re-sampling an example distribution of higher IoU threshold and uses it to train the next stage. Hence, it results in a uniform distribution of training samples for each stage of detectors and enabling the network to train on higher IoU threshold values.

B. Composite Backbone

We use a dual backbone based architecture [23] which creates a composite connection between the parallel stages of two adjacent ResNeXt-101 backbones (one is called assistant backbone and other is called lead backbone). The assistant backbone's high-level output features are fed as an input to the corresponding lead backbone's

stage. In a conventional network, the output (denoted by x^l) of previous l-1 stages is fed as input to the l-th stage, given by:

$$x^l = F^l(x^l - 1), l \geq 2. \quad (1)$$

where $F^l(.)$ is a non-linear transformation operation of l-th stage. However, our network takes input from previous stages as well as parallel stage of assistant backbone. For a given stage l of lead backbone(bl), input is a combination of the output of previous l-1 stages of lead backbone and parallel l-th stage of assistant backbone(ba), given by:

$$x^l_{bl} = F^l_{bl}(x^{l-1}_k + g(x^l_{ba})), l \geq 2, \quad (2)$$

where g(.) represents composite connection. It helps the lead backbone to take advantage of the features learned by the assistant backbone. Finally, the output of the lead backbone is used for further processing in the subsequent network.

C. Deformable Convolution

The commonly used backbone, ResNeXt architectures, has conventional convolution operation, in which the effective receptive field of all the neurons in a given layer is the same. The grid points are generally confined to a fixed 3×3 or 5×5 square receptive fields. It performs well for layers at the lower hierarchy. Still, when the objects appear at the arbitrary scales and transformations, generally at the higher-level, the convolution operation does not capture the features well. We replace the fixed receptive field CNN with deformable CNN [22] in each of our dual backbone architectures. The grid is deformable as a learnable offset can move each grid point. In a conventional convolution, we sample over the input feature map x using a regular grid R, given by

$$y(p_0) = \sum_{p_n \in R} w(p_n)x(p_0 + p_n). \quad (3)$$

Whereas in a deformable convolution, for each location p_o on the output feature map y, we augment the regular grid using the offset Δp_n such that $\{\Delta p_n | n = 1, ..., N\}$, where $N = |R|$, given by

$$y(p_0) = \sum_{p_n \in R} w(p_n)x(p_0 + p_n + \Delta p). \quad (4)$$

Deformable convolution is operated on R but with each point augmented by a learnable offset Δp. The offset value, Δp, is itself a trainable parameter. It enables each neuron to alter its receptive field based on the preceding feature map by creating an explicit offset. It makes the convolution operation agnostic for varying scales and transformations. The deformable convolution is shown in Figure 2.

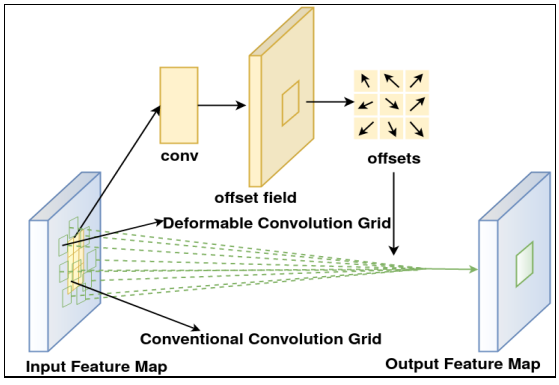

Fig. 2: Illustration of the deformable convolution.

D. Implementation Details

We implement CDeC-Net in Pytorch using MMdetection toolbox [45]. We use NVIDIA GeForce RTX 2080 Ti GPU with 12 GB memory for our experiments. We use pre-trained ResNeXt-101 (with blocks 3, 4, 23 and 3) on MS-COCO [25] with FPN as the network head. We train CDeC-Net with document images scaled to 1200 × 800, while maintaining the original aspect ratio, as the input. We use 0.00125 as an initial learning rate with a learning rate decay at 25 epoch and 40 epoch. We use 0.0033 as warmup schedule for first 500 iterations. CDeC-Net is trained for 50 epochs. However, for larger datasets such as PubLayNet and Tablebank, the model is trained for 8 epochs in total with learning rate decay at 4 epoch and 6 epoch. In case of fine-tuning, we use 12 epochs in total. We use three IoU threshold values — 0.5, 0.6, and 0.7 in our model. We use 0.5, 1.0 and 2.0 as anchor ratio with a single anchor scale of 8. The batch size of 1 is used for training our models.

IV. Experiments

A. Evaluation Measures

Similar to the existing table localization tasks [1]–[11] in document images, we also use recall, precision, F1, and mean average precision (mAP) to evaluate the performance of CDeC-Net. For fair comparison, we evaluate the proposed CDeC-Net on same IoU threshold values as mentioned in the respective existing papers. We perform multi-scale testing at 7 different scales (with 3 smaller scales, original scale, and 3 larger scales). We select detection output as final result if it presents in at least 4 test cases out of 7 scales. It helps in eliminating the false positives and provide consistent results.

B. Comparison with State-of-the-Arts on Benchmark Datasets

Dataset	Method	Score			
		R↑	P↑	F1↑	mAP↑
ICDAR-2013	DeCNT [3]	0.996	0.996	0.996	-
	CDeC-Net (our)	**1.000**	**1.000**	**1.000**	**1.000**
ICADR-2017	YOLOv3 [18]	**0.968**	**0.975**	**0.971**	-
	CDeC-Net (our)	0.924	0.970	0.947	**0.912**
ICADR-2019	TableRadar [13]	**0.940**	0.950	**0.945**	-
	CDeC-Net (our)	0.934	**0.953**	0.944	**0.922**
UNLV	GOD [10]	0.910	0.946	0.928	-
	CDeC-Net (our)	**0.925**	**0.952**	**0.938**	**0.912**
Marmot	DeCNT [3]	**0.946**	0.849	0.895	-
	CDeC-Net (our)	0.930	**0.975**	**0.952**	**0.911**
TableBank	Li et al. [7]	0.975	0.987	0.981	-
	CDeC-Net (our)	**0.979**	**0.995**	**0.987**	**0.976**
PubLayNet	M-RCNN [9]	-	-	-	0.960
	CDeC-Net (our)	**0.970**	**0.988**	**0.978**	**0.967**

TABLE III: Illustrates comparison between CDeC-Net and state-of-the-art techniques on the existing benchmark datasets.

Comparison with current state-of-the-art techniques on various benchmark datasets is shown in Table III. We observe from the table that CDeC-Net outperforms state-of-the-art techniques on ICADR-2013, UNLV, Marmot, TableBank, and PubLayNet datasets. For ICDAR-2019, CDeC-Net obtains very close performance to the state-of-the-art techniques. In case of ICDAR-2017 dataset, the performance of CDeC-Net is 2.4% lower than the state-of-the-art method.

Tables IV-VII presents the comparative results between the proposed CDeC-Net and the existing techniques on various benchmark datasets under the existing experimental environments. In most of the cases, CDeC-Net performs better than the existing techniques. The cascade Mask R-CNN in CDeC-Net leads to significant reduction in number of false positives, which is evident from the high precision

Method	Training		Fine-tuning		Test		IoU	Score			
	Dataset	#Image	Dataset	#Image	Dataset	#Image		R↑	P↑	F1↑	mAP↑
DeCNT [3]	D1	4808	-	-	ICDAR-2013	238	0.5	0.996*	0.996*	0.996*	-
CDeC-Net (our)	D1	4808	-	-	ICDAR-2013	238	0.5	**1.000**	**1.000**	**1.000**	**1.000**
GOD [10]	Marmot	2K	-	-	ICDAR-2013	238	0.5	**1.000**	**0.982**	**0.991**	-
CDeC-Net (our)	Marmot	2K	-	-	ICDAR-2013	238	0.5	**1.000**	0.981	**0.991**	**0.995**
F-RCNN [9]	PubLayNet	340K	ICADR-2013	170	ICADR-2013	238	0.5	0.964	0.972	0.968	
M-RCNN [9]	PubLayNet	340K	ICADR-2013	170	ICADR-2013	238	0.5	0.955	0.940	0.947	-
CDeC-Net (our)	PubLayNet	340K	ICADR-2013	170	ICADR-2013	238	0.5	**0.968**	**0.987**	**0.977**	**0.959**
YOLOv3+A+PG [18]	ICDAR-2017	1.6K	-	-	ICADR-2013	238	0.5	0.949	1.000	0.973	-
CDeC-Net (our)	ICDAR-2017	1.6K	-	-	ICADR-2013	238	0.5	**1.000**	**1.000**	**1.000**	**1.000**
Khan et al. [46]	Marmot	2K	ICDAR-2013	204	ICDAR-2013	34	0.5	0.901	0.969	0.934	-
TableNet+SF [47]	Marmot	2K	ICDAR-2013	204	ICDAR-2013	34	0.5	0.963	0.970	0.966	-
DeepDeSRT [2]	Marmot	2K	ICDAR-2013	204	ICDAR-2013	34	0.5	0.962	0.974	0.968	-
CDeC-Net (our)	Marmot	2K	ICDAR-2013	204	ICDAR-2013	34	0.5	**1.000**	**1.000**	**1.000**	**1.000**
M-RCNN [11]	Pascel VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.770	0.140	0.230	-
RetinaNet [11]	Pascel VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.580	0.560	0.570	-
SSD [11]	Pascel VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.680	0.540	0.600	-
YOLO [11]	Pascel VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.580	0.920	0.750	-
CDeC-Net (our)	Pascel VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	**0.844**	**1.000**	**0.922**	**0.844**
M-RCNN [11]	TableBank-LaTeX	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	**0.970**	0.700	0.810	-
RetinaNet [11]	TableBank-LaTeX	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.770	0.830	0.800	-
SSD [11]	TableBank-LaTeX	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.680	0.620	0.650	-
YOLO [11]	TableBank-LaTeX	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.650	**1.000**	0.780	-
CDeC-Net (our)	TableBank-LaTeX	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.933	**1.000**	**0.967**	**0.933**
Kavasidis et al. [48]	Custom dataset	45K	-	-	ICDAR-2013	238	0.5	0.981	0.975	0.978	-
PFTD [49]	-	-	-	-	ICADR-2013	238	0.5	0.915	0.939	0.926	-
Tran et al. [50]	-	-	-	-	ICDAR-2013	238	0.5	0.964	0.952	0.958	-
CDeC-Net‡ (our)	IIIT-AR-13K	9K	-	-	ICDAR-2013	238	0.5	0.942	0.993	0.968	0.942

TABLE IV: Illustrates comparison between the proposed CDeC-Net and state-of-the-art techniques on ICDAR-2013 dataset. **A:** indicates anchor optimization, **PG:** indicates post-processing technique, **SF:** indicates semantic features, **D1:** indicates Marmot+UNLV+ICDAR-2017, ***:** indicates the authors reported 0.996 in table however in discussion they mentioned 0.994. CDeC-Net‡**:** indicates a single model which is trained with IIIT-AR-13K dataset.

Method	Training		Fine-tuning		Test		IoU	Score			
	Dataset	#Image	Dataset	#Image	Dataset	#Image		R↑	P↑	F1↑	mAP↑
TableRadar [13]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.8	**0.940**	0.950	**0.945**	-
NLPR-PAL [13]	ICDAR-2019	1200			ICDAR-2019	439	0.8	0.930	0.930	0.930	-
Lenovo Ocean [13]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.8	0.860	0.880	0.870	-
CDeC-Net (our)	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.8	0.934	**0.953**	0.944	**0.922**
TableRadar [13]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	0.890	0.900	0.895	-
NLPR-PAL [13]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	0.860	0.860	0.860	-
Lenovo Ocean [13]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	0.810	0.820	0.815	-
CDeC-Net (our)	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	**0.904**	**0.922**	**0.913**	**0.843**
M-RCNN [11]	Pascel VOC	16K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.640	0.600	0.620	-
RetinaNet [11]	Pascel VOC	16K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.660	0.860	0.740	-
SSD [11]	Pascel VOC	16K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.350	0.310	0.330	-
YOLO [11]	Pascel VOC	16K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.910	0.950	0.930	-
CDeC-Net (our)	Pascel VOC	16K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	**0.962**	**0.981**	**0.971**	**0.949**
M-RCNN [11]	TableBank-LaTeX	199K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.850	0.760	0.810	-
RetinaNet [11]	TableBank-LaTeX	199K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.740	0.910	0.820	-
SSD [11]	TableBank-LaTeX	199K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.350	0.350	0.350	-
YOLO [11]	TableBank-LaTeX	199K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	0.950	0.950	0.950	-
CDeC-Net (our)	TableBank-LaTeX	199K	ICDAR-2019 (archive)	599	ICDAR-2019 (archive)	198	0.6	**0.924**	**0.984**	**0.954**	**0.909**
CDeC-Net‡ (our)	IIIT-AR-13K	9K	-	-	ICDAR-2019	439	0.8	0.625	0.871	0.748	0.551
CDeC-Net‡ (our)	IIIT-AR-13K	9K	ICDAR-2019	1200	ICDAR-2019	439	0.8	0.930	0.971	0.950	0.913

TABLE V: Illustrates comparison between the proposed CDeC-Net and state-of-the-art techniques on ICDAR-2019 dataset. CDeC-Net‡**:** indicates a single model which is trained with IIIT-AR-13K dataset.

Method	Training		Fine-tuning		Test		IoU	Score			
	Dataset	#Image	Dataset	#Image	Dataset	#Image		R↑	P↑	F1↑	mAP↑
GOD [10]	Marmot	2K	UNLV	340	UNLV	84	0.5	0.910	0.946	0.928	-
CDeC-Net (our)	Marmot	2K	UNLV	340	UNLV	84	0.5	**0.925**	**0.952**	**0.938**	**0.912**
Gilani et al. [1]	UNLV	340	-	-	UNLV	84	0.5	**0.907**	0.823	0.863	-
CDeC-Net (our)	UNLV	340	-	-	UNLV	84	0.5	0.906	**0.914**	**0.910**	**0.861**
Arif and Shafait [6]	private	1019	-	-	UNLV	427	0.5	**0.932**	0.863	**0.896**	-
CDeC-Net (our)	private	1019	-	-	UNLV	427	0.5	0.745	**0.912**	0.829	**0.711**
DeCNT [3]	D4	4622	-	-	UNLV	424	0.5	**0.749**	0.786	0.767	-
CDeC-Net (our)	D4	4622	-	-	UNLV	424	0.5	0.736	**0.852**	**0.794**	**0.657**
M-RCNN [11]	Pascel VOC	16K	UNLV	302	UNLV	101	0.6	0.580	0.290	0.390	-
RetinaNet [11]	Pascel VOC	16K	UNLV	302	UNLV	101	0.6	0.830	0.810	0.820	-
SSD [11]	Pascel VOC	16K	UNLV	302	UNLV	101	0.6	0.640	0.660	0.650	-
YOLO [11]	Pascel VOC	16K	UNLV	302	UNLV	101	0.6	**0.950**	0.910	**0.930**	-
CDeC-Net (our)	Pascel VOC	16K	UNLV	302	UNLV	101	0.6	0.805	**0.961**	0.883	**0.788**
M-RCNN [11]	TableBank-LaTeX	199K	UNLV	302	UNLV	101	0.6	0.830	0.660	0.740	-
RetinaNet [11]	TableBank-LaTeX	199K	UNLV	302	UNLV	101	0.6	0.830	0.810	0.820	-
SSD [11]	TableBank-LaTeX	199K	UNLV	302	UNLV	101	0.6	0.660	0.720	0.690	-
YOLO [11]	TableBank-LaTeX	199K	UNLV	302	UNLV	101	0.6	**0.950**	0.930	0.940	-
CDeC-Net (our)	TableBank-LaTeX	199K	UNLV	302	UNLV	101	0.6	0.894	**0.991**	**0.943**	**0.889**
CDeC-Net‡ (our)	IIIT-AR-13K	9K			UNLV	424	0.5	0.770	0.96	0.865	0.742
CDeC-Net‡ (our)	IIIT-AR-13K	9K	private	1019	UNLV	427	0.5	0.776	0.958	0.866	0.750

TABLE VI: Illustrates comparison between the proposed CDeC-Net and state-of-the-art techniques on UNLV dataset. **D4:** indicates ICDAR-2013+ICDAR-2017+Marmot. CDeC-Net‡**:** indicates a single model which is trained with IIIT-AR-13K dataset.

Method	Training		Fine-tuning		Test		IoU	Score			
	Dataset	#Image	Dataset	#Image	Dataset	#Image		R↑	P↑	F1↑	mAP↑
Li et al. [7]	TableBank-LaTeX	253K	-	-	TableBank-Word	1K	0.5	**0.956**	0.826	**0.886**	-
					TableBank-LaTeX	1K	0.5	0.975	0.987	0.981	-
					TableBank-both	2K	0.5	**0.962**	0.872	0.915	-
CDeC-Net (our)	TableBank-LaTeX	253K	-	-	TableBank-Word	1K	0.5	0.868	**0.873**	0.871	**0.762**
					TableBank-LaTeX	1K	0.5	**0.979**	**0.995**	**0.987**	**0.976**
					TableBank-both	2K	0.5	0.924	**0.934**	**0.929**	**0.898**
M-RCNN [11]	TableBank-LaTeX	199K	-	-	TableBank-LaTeX	1K	0.6	0.980	0.960	0.940	-
RetinaNet [11]	TableBank-LaTeX	199K	-	-	TableBank-LaTeX	1K	0.6	0.860	0.980	0.920	-
SSD [11]	TableBank-LaTeX	199K	-	-	TableBank-LaTeX	1K	0.6	0.970	0.960	0.965	-
YOLO [11]	TableBank-LaTeX	199K	-	-	TableBank-LaTeX	1K	0.6	**0.990**	0.980	0.985	-
CDeC-Net (our)	TableBank-LaTeX	199K	-	-	TableBank-LaTeX	1K	0.6	0.978	**0.995**	**0.986**	**0.974**
CDeC-Net‡ (our)	IIIT-AR-13K	9K			TableBank-LaTeX	1K	0.6	0.779	0.961	0.870	0.759
CDeC-Net‡ (our)	IIIT-AR-13K	9K	TableBank-LaTeX	199K	TableBank-LaTeX	1K	0.6	0.970	0.990	0.980	0.965

TABLE VII: Illustrates comparison between the proposed CDeC-Net (our) and state-of-the-art techniques on TableBank dataset. CDeC-Net‡**:** indicates a single model which is trained with IIIT-AR-13K dataset.

values. Table IV presents the obtained results under various experimental settings for ICDAR-2013. We observe that for all experimental settings, CDeC-Net obtains the best results. In case of ICDAR-2019, CDeC-Net performs only 0.1% F1 score lower than state-of-the-art technique - TableRadar [13] at IoU threshold 0.8. At higher threshold value 0.9, CDeC-Net performs significantly (1.8% greater F1 score) better than the state-of-the-art technique - TableRadar [13]. For all other experimental settings, CDeC-Net also obtain the best results. For UNLV dataset, CDeC-Net performs (2.7% F1 score) better than the state-of-the-art method - DeCNT [3]. For TableBank dataset, CDeC-Net performs significantly better that state-of-the-art technique - Li *et al.* [7].

C. Effect of IoU Threshold on Table Detection

We evaluate the trained CDeC-Net on the existing benchmark datasets under varying IoU thresholds to test robustness of the proposed network. Our experiments on various benchmark datasets shows that CDeC-Net gives consistent results over varying IoU thresholds. Table VIII highlights that in case of ICDAR-2019 datasets, the CDeC-Net consistently obtains high detection accuracy under varying thresholds (in range 0.5-0.9). Our model also obtains consistent results (in range of 0.5-0.8) on ICDAR-2013 and UNLV datasets. Only at threshold 0.9, there is a performance drop on ICDAR-2013 and UNLV datasets.

IoU Threshold	Performance on Various Benchmark Datasets								
	ICDAR-2013			ICDAR-2019			UNLV		
	R↑	P↑	F1↑	R↑	P↑	F1↑	R↑	P↑	F1↑
0.5	1.000	1.000	1.000	0.946	0.987	0.966	0.770	0.960	0.865
0.6	1.000	1.000	1.000	0.939	0.980	0.959	0.758	0.944	0.851
0.7	0.987	0.987	0.987	0.936	0.977	0.956	0.734	0.915	0.825
0.8	0.942	0.942	0.942	0.930	0.971	0.950	0.663	0.826	0.744
0.9	0.660	0.660	0.660	0.895	0.934	0.915	0.496	0.618	0.557

TABLE VIII: Illustrates the performance of CDeC-Net under varying IoU thresholds.

D. Qualitative Results

A visualization of detection results[2] on ICDAR-2013, ICDAR-POD-2017, UNLV (first row, left to right), ICDAR-2019 (cTDaR), PubLayNet and TableBank (second row, left to right) obtained by CDeC-Net is shown in Figure 3. The figure highlights that the CDeC-Net properly detects complex table with high confidence score.

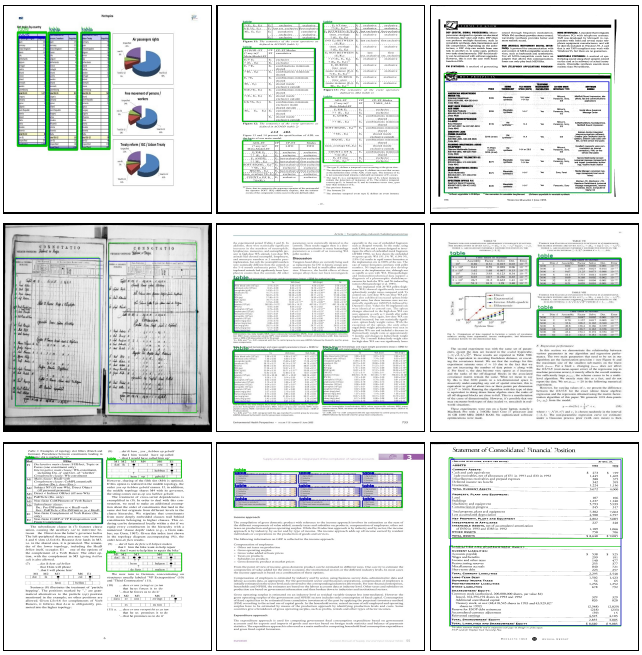

Fig. 3: Illustration of complex table detection results. Blue and Green colored rectangles correspond to ground truth and predicted bounding boxes using CDeC-Net. **First and Second Rows:** show examples where CDeC-Net accurately detects the tables. **Third Row:** shows examples where CDeC-Net fails to accurately detect the tables.

Third row of Figure 3 shows some examples where CDeC-Net model fails to properly detect the tables. In the first image, it detects two false positives that are visually similar to tables. The second and third images contain multiple closely spaced tables where CDeC-Net detects them as a single table.

E. Results of Single Model

Tables IV-VII presents the comparative results between the proposed CDeC-Net and the existing techniques on various benchmark datasets under the existing experimental environments. The last row of each table presents obtained results using our single model CDeC-Net‡ trained with IIIT-AR-13K dataset, fine-tuned with training images and evaluated on test images of the respective datasets. Table IV highlights that our single model CDeC-Net‡ attains very close results to our best model CDeC-Net on ICDAR-2013 dataset. In case of ICDAR-2019, our single model CDeC-Net‡ obtains the best performance at IoU threshold 0.8. In case of UNLV and TableBank datasets, the performance of single model CDeC-Net‡ are very close to our best performing model CDeC-Net.

Figure 4 presents the visual results obtained using our single model CDeC-Net‡ and our best model CDeC-Net. We select the best model under various existing experimental environments. First row of Figure 4 shows examples where single model CDeC-Net‡

[2]More detailed results are given in supplementary material

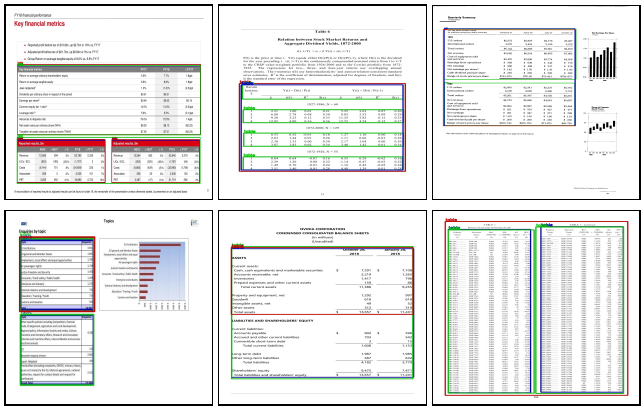

Fig. 4: Illustration visual results of the state-of-the-art CDeC-Net model and single CDeC-Net‡ model. Blue, Green, and Red colored rectangles correspond to ground truth and predicted bounding boxes using state-of-the-art CDeC-Net and single CDeC-Net‡ model respectively. **First Row:** shows examples where CDeC-Net‡ detects table accurately and CDeC-Net fails to detect table accurately. **Second Row:** shows examples where CDeC-Net detects table accurately and CDeC-Net‡ fails to detect table accurately.

performs better than the best model CDeC-Net. In those examples, our best model CDeC-Net predicts single bounding box for multiple tables. While single model CDeC-Net‡ accurately predicts bounding box corresponding to each table present in the document. The second row of Figure 4 presents examples where our best model CDeC-Net accurately detects all tables present in the documents. While our single model CDeC-Net‡ fails to predict bounding boxes corresponding to tables present in the documents.

F. Ablation Study

Models	**Score**			
	R↑	**P↑**	**F1↑**	**mAP↑**
Cascade Mask R-CNN with ResNeXt-101 as backbone	0.987	0.975	0.981	0.975
Cascade Mask R-CNN with composite ResNeXt-101 as backbone	0.987	0.981	0.984	0.973
Cascade Mask R-CNN with composite ResNeXt-101 having deformable convolution as backbone (i.e., CDeC-Net)	**1.000**	**1.000**	**1.000**	**0.995**

TABLE IX: Illustrates the performances of various models. All models are tested on ICDAR-2013 dataset with 0.5 as IoU threshold. Cascade Mask R-CNN with composite ResNeXt-101 having deformable convolution as backbone i.e., CDeC-Net obtains best results as compared to other models. We select CDeC-Net as our final model.

We perform a series of experiments to check the effectiveness of the proposed method. We train three models on Marmot dataset and evaluate on ICDAR-2013. Our baseline model — cascade Mask R-CNN achieves F1 score of 0.981 at IoU threshold 0.5. We incorporate the dual backbone in the baseline model and obtain an F1 score of 0.984. Again we incorporate deformable convolution instead of convolution in the dual backbone and call it as CDeC-Net, which attains the best F1 score 1.000. This particular experiment highlights

the utility of incorporating the key components — dual backbone and deformable convolution into the baseline model Cascade Mask R-CNN. We finally select CDeC-Net as our final model for table detection task.

V. Conclusion

We introduce a CDeC-Net, which consists of a cascade Mask R-CNN with a dual backbone having deformable convolution to detect tables present in documents with high accuracy at higher IoU threshold. The proposed CDeC-Net achieves state-of-the-art performance for most of the benchmark datasets under various existing experimental environments and significantly reduces the false positive detection even at the higher IoU threshold. We also provide a single model CDeC-Net‡ for all benchmark datasets, which obtains very close performance to the state-of-the-art techniques. We expect that our single model sets a standard benchmark and improves table detection accuracy and other page objects– figures, logos, mathematical expressions, etc. We extend the current framework for future work to a more challenging table structure recognition task.

Acknowledgement

This work is partly supported by MEITY, Government of India.

References

[1] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *ICDAR*, 2017.

[2] 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 *ICDAR*, 2017.

[3] S. A. Siddiqui, M. I. Malik, S. Agne, A. Dengel, and S. Ahmed, "DeCNT: Deep deformable CNN for table detection," *IEEE Access*, 2018.

[4] N. Sun, Y. Zhu, and X. Hu, "Faster R-CNN based table detection combining corner locating," in *ICDAR*, 2019.

[5] N. D. Vo, K. Nguyen, T. V. Nguyen, and K. Nguyen, "Ensemble of deep object detectors for page object detection," in *UIMC*, 2018.

[6] S. Arif and F. Shafait, "Table detection in document images using foreground and background features," in *DICTA*, 2018.

[7] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li, "TableBank: Table benchmark for image-based table detection and recognition," *arXiv*, 2019.

[8] J. Younas, S. T. R. Rizvi, M. I. Malik, F. Shafait, P. Lukowicz, and S. Ahmed, "FFD: Figure and formula detection from document images," in *DICTA*, 2019.

[9] X. Zhong, J. Tang, and A. J. Yepes, "PubLayNet: largest dataset ever for document layout analysis," in *ICDAR*, 2019.

[10] R. Saha, A. Mondal, and C. V. Jawahar, "Graphical object detection in document images," in *ICDAR*, 2019.

[11] Á. Casado-García, C. Domínguez, J. Heras, E. Mata, and V. Pascual, "The benefits of close-domain fine-tuning for table detection in document images," *arXiv*, 2019.

[12] A. Mondal, P. Lipps, and C. V. Jawahar, "IIIT-AR-13K: a new dataset for graphical object detection in documents," in *DAS*, 2020.

[13] L. Gao, Y. Huang, H. Déjean, J.-L. Meunier, Q. Yan, Y. Fang, F. Kleber, and E. Lang, "ICDAR 2019 competition on table detection and recognition (cTDaR)," in *ICDAR*, 2019.

[14] A. Shahab, F. Shafait, T. Kieninger, and A. Dengel, "An open approach towards the benchmarking of table structure recognition systems," in *DAS*, 2010.

[15] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "ICDAR 2013 table competition," in *ICDAR*, 2013.

[16] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "ICDAR 2017 competition on page object detection," in *ICDAR*, 2017.

[17] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, ground-truth and performance metrics for table detection evaluation," in *DAS*, 2012.

[18] Y. Huang, Q. Yan, Y. Li, Y. Chen, X. Wang, L. Gao, and Z. Tang, "A YOLO-based table detection method," in *ICDAR*, 2019.

[19] K. Itonori, "Table structure recognition based on textblock arrangement and ruled line position," in *ICDAR*, 1993.

[20] T. Kieninger, "Table structure recognition based on robust block segmentation," in *Electronic Imaging*, 1998.

[21] S. Tupaj, Z. Shi, C. H. Chang, and D. C. H. Chang, "Extracting tabular information from text files," in *EECS Department, Tufts University*, 1996.

[22] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *ICCV*, 2017.

[23] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, and H. Ling, "Cbnet: A novel composite backbone network architecture for object detection," *arXiv*, 2019.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.

[25] 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 *ECCV*, 2014.

[26] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. on PAMI*, 2019.

[27] S. Chandran and R. Kasturi, "Structural recognition of tabulated data," in *ICDAR*, 1993.

[28] Y. Hirayama, "A method for table structure analysis using DP matching," in *ICDAR*, 1995.

[29] E. Green and M. Krishnamoorthy, "Recognition of tables using table grammars," in *DAIR*, 1995.

[30] J. Hu, R. S. Kashi, D. P. Lopresti, and G. Wilfong, "Medium-independent table detection," in *Document Recognition and Retrieval VII*, 1999.

[31] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis, "Automatic table detection in document images," in *IPRIA*, 2005.

[32] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *DAS*, 2010.

[33] T. Kieninger and A. Dengel, "The T-RECS table recognition and analysis system," in *DAS*, 1998.

[34] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," in *Object recognition supported by user interaction for service robots*, 2002.

[35] A. C. e Silva, "Learning rich hidden Markov models in document analysis: Table location," in *ICDAR*, 2009.

[36] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "Learning to detect tables in scanned document images using line information," in *ICDAR*, 2013.

[37] M. Fan and D. S. Kim, "Table region detection on large-scale PDF files without labeled data," *CoRR*, 2015.

[38] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for PDF documents based on convolutional neural networks," in *DAS*, 2016.

[39] R. Girshick, "Fast R-CNN," in *CVPR*, 2015.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *CVPR*, 2017.

[42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017.

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *ECCV*, 2016.

[45] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open MMLab detection toolbox and benchmark," *arXiv*, 2019.

[46] S. A. Khan, S. M. D. Khalid, M. A. Shahzad, and F. Shafait, "Table structure extraction with bi-directional gated recurrent unit networks," in *ICDAR*, 2019.

[47] S. S. Paliwal, D. Vishwanath, 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 *ICDAR*, 2019.

[48] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Giordano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," *arXiv*, 2018.

[49] L. Melinda and C. Bhagvati, "Parameter-free table detection method," in *ICDAR*, 2019.

[50] D. N. Tran, T. A. Tran, A. Oh, S. H. Kim, and I. S. Na, "Table detection from document image using vertical arrangement of text blocks," *International Journal of Contents*, 2015.