Pattern Recognition 142 (2023) 109698

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Dataset agnostic document object detection

Ajoy Mondal*, Madhav Agarwal, C.V. Jawahar

CVIT, International Institute of Information Technology, Hyderabad, India

ARTICLE INFO

Article history:
Received 21 April 2022
Revised 5 May 2023
Accepted 18 May 2023
Available online 20 May 2023

Keywords:
Document object detection
Table detection
Figure detection
Equation detection
Cascade Mask R-CNN
Deformable convolution

ABSTRACT

Localizing document objects such as tables, figures, and equations is a primary step for extracting information from document images. We propose a novel end-to-end trainable deep network, termed *Document Object Localization Network* (DOLNet), for detecting various objects present in the document images. The proposed network is a multi-stage extension of Mask R-CNN with a dual backbone having deformable convolution for detecting document objects with high detection accuracy at a higher IoU threshold. We also empirically evaluate the proposed DOLNet on the publicly available benchmark datasets. The proposed DOLNet achieves state-of-the-art performance for most of the bench-mark datasets under various existing experimental environments.

Our solution has three important properties: (i) a single trained model DOLNet‡ that performs well across all the popular benchmark datasets, (ii) reports excellent performances across multiple, including with higher IoU thresholds, and (iii) consistently demonstrate the superior quantitative performance by following the same protocol of the recent works for each of the benchmarks.

1. Introduction

Rapid growth in information technology has led to an exponential increase in the production and storage of digital document images over the last few decades. Extracting information from such a large corpus is impractical for humans. Hence, needful information could be lost or unutilized over time. Digital documents have many objects (such as tables, figures, equations, logos, and signatures) other than text. All such objects are collectively termed as *document object*. These objects also show wide variations in their appearance. Therefore, any attempt to detect document objects must be generic and applicable across a wide variety of documents and use cases. Localizing document objects becomes challenging due to high intra-class variability and inter-class similarity.

In the past, the document objects, mainly tables, are localized using metadata and semantic information present in the objects [1,2]. These methods failed to localize objects in scanned documents due to metadata unavailability. Recently, several deep neural networks-based solutions have been available for localizing different document objects such as tables [3–13], figure & formula [14] and various objects [15,16] in documents. Benchmark datasets — ICDAR-2013 [17], ICDAR-POD-2017 [15], CTDAR [18], UNLV [19], DeepFigures [20], PubLayNet [16], Marmot [21], TableBank [7], DocBank [22], and IIIT-AR-13K [23] are publicly available for document object detection tasks.

We observe from the literature that the recent existing methods [5,12] provide state-of-the-art performance on many benchmark datasets. However, these methods provide different trained models corresponding to different datasets to achieve state-of-the-art performance. These methods need to provide a single model that can achieve state-of-the-art performance on all existing benchmark datasets. Most methods [5,12,14,16] use a single threshold value, commonly 0.5, for document object detection training. It could lead to a noisy document object detection at a higher threshold during evaluation. Liu *et al.* [24] discuss that a Convolution Neural Network (CNN) based object detectors generally use a backbone network to extract features for detecting objects. These backbones are usually designed for image classification and pre-trained on either ImageNet or MS-COCO. Hence, the direct use of these backbones may lead to sub-optimal performance on document object detection [3–8,11,14,16,25].

To address issues ((i) different trained models corresponding to different datasets to achieve state-of-the-art performance, (ii) backbone designed for image classification and pre-trained on ImageNet use to extract features for detecting document objects, and (iii) exiting document object detectors commonly use a threshold of 0.5, which leads to noisy detection and frequently degrades the performance for higher thresholds) mentioned above, we propose a *Document Object Localization Network*, called as DOLNet, to detect objects more accurately present in document images. The

* Corresponding author.
E-mail addresses: ajoy.mondal@iiit.ac.in (A. Mondal), madhav.agarwal@research.iiit.ac.in (M. Agarwal), jawahar@iiit.ac.in (C.V. Jawahar).

https://doi.org/10.1016/j.patcog.2023.109698
0031-3203/

proposed DOLNet 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 a higher threshold. Inspired by [24], we use a composite backbone consisting of multiple identical backbones with composite connections between neighbor backbones to our DOLNet to improve detection accuracy. To model geometric transformations, we also incorporate deformable convolution [27] in the backbones. We extensively evaluate DOLNet 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 DOLNet achieves state-of-the-art performance on IIIT-AR-13K for document object detection. In the case of only table detection, DOLNet achieves state-of-the-art performance on almost all existing benchmark datasets. We also achieve high accuracy and tight bounding box detection at a higher IoU threshold than the previous benchmark results.

We summarise our main contributions as follows:

- Present an end-to-end trainable deep architecture, DOLNet which consists of cascade Mask R-CNN containing composite backbones with deformable convolution to detect document objects more accurately.
- 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 for table detection.
- Achieve state-of-the-art results on IIIT-AR-13K for detecting various document objects.
- Achieve state-of-the-art results on almost all publicly available benchmark datasets for table detection.

This work extends several aspects of our published conference paper [13]. First, instead of localizing only tables, we localize several document objects such as tables, figures, mathematical equations, signatures, and logos in the documents. Secondly, we conduct extensive experiments to validate the effectiveness of our model. Thirdly, we provide a more insightful discussion regarding document object detection tasks. Finally, we draw several important conclusions and highlight several promising future directions.

The rest of the article is organized as follows. Section 2 briefly discusses the related work. The proposed DOLNet: Document Object Localization Network is presented in Section 3. Detail on experiments is presented in Section 4. We analyze the obtained results in Section 5, Section 6 and Section 7. Finally, we make conclusive remarks in Section 8.

2. Related work

Like text, other document objects such as tables, figures, equations, signatures, and logos are also essential components in the document. Detecting such document objects is a fundamental step for document understanding. Over time, various algorithms have been proposed in the literature to solve the problems (particularly table detection). Initially, methods are developed depending on heuristics or metadata information. Such methods provide mainly precise solutions depending on the characteristics of documents. Later machine learning and deep learning are applied to obtain more generic solutions independent of the nature of the document. Existing work on this problem can be broadly divided into two categories — (i) Rule-based Techniques and (ii) Learning-based Techniques.

2.1. Rule-based techniques

Document object, mainly table detection, was started by Itonori [1] in 1993 using heuristic rules leading to text block arrangement and text line position for locating tables in documents. Chandran *et al.* [28] localized tables by extracting all vertical and horizontal lines. The authors detect tables using white stream recognition in vertical and horizontal directions in the missing row and column separator lines.

Following these works, several table detection approaches [29] have been developed using improved heuristic rules. Shafait and Smith [30] detect tables in the document using a layout analysis of Tesseract [31] with tab-stops indicating where a text block starts and ends. Mandal *et al.* [32] identify table regions by analyzing distinct columns in which the gaps between fields are more significant than the gaps between words in the text line. In the same direction, Zhang *et al.* [33] extract tables from Chinese ink documents using row-column heuristics. Table detection based on multi-clue heuristic is proposed in [34]. Though these methods perform well on documents with limited layouts, they need more manual effort to find better heuristic rules. Moreover, rule-based approaches need to obtain generic solutions. Therefore, it is necessary to develop machine-learning strategies to solve the table detection problem.

2.2. Learning-based techniques

Machine learning alleviates the rule-based approaches for document object detection issues mentioned above. Kieninger and Dengel [35] used bottom-up clustering of given word segments to recognize table regions in documents. Later, table detection problems are solved by different machine learning algorithms. Silva [36] formulates table detection as a sequence labeling problem and solves it using Hidden Markov Models. Kasar *et al.* [37] detect table utilizing the intersection of lines and various hand-crafted features with a Support Vector Machine (SVM). Fan and Kim [38] ensemble of a decision of multiple classifiers Naive Bayes, logistic regression and SVM to detect a table region. In [39], page objects are extracted using the AdaBoost cascade of weak classifiers and Haar-like features. Learning methods improve table detection accuracy significantly.

The great success of deep CNNs in computer vision motivates to development of several algorithms for detecting document objects. Augusto *et al.* [40] develop an algorithm based on Fast R-CNN [41] for extracting the layout of a document. Successively, several approaches using Faster R-CNN [42] are proposed for table detection [3–9], table detection and data extraction [43], figure and formula detection [14], document object detection [15] and document layout analysis [16]. Mask R-CNN [44] is used for figure and formula detection [14], table detection [10,11], document object detection [16]. Similarly, YOLO [45] for table detection [11]; cascade Mask R-CNN in CASCADETABNET [12], CDEC-NET [13], and Graph Neural Network (GNN) [46] for table detection in document images exist in the literature. All these methods are data-driven without requiring any heuristics or metadata, robust to document types, layouts, and reduce the efforts of hand-crafted feature engineering in CNNs. Fully Convolutional Network (FCN) for table detection and recognition [47].

Gilani *et al.* [3] used the Faster R-CNN model to detect tables in document images. Instead of the original document image, the distance-transformed image is input to fine-tune the pre-trained model to work on document datasets effortlessly. In the same direction, the transformed document image is taken as input to Faster R-CNN model for detecting tables in [6]; and figures and mathematical equations in [14] 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 docu-

ments. Similarly, Zhong *et al.* [16] also experimentally established that Mask R-CNN outperforms Faster R-CNN for extracting semantic regions from documents.

It is observed that each detection method is sensitive to a specific type of object. In [11], the authors discuss the benefit of fine-tuning from a close domain on four different object detection models — Mask R-CNN [44], RetinaNet [48], SSD [49] and YOLO [45]. The experiments highlight that the accuracy improvement and the close domain fine-tune approach avoid over-fitting and solve a small training set. In the same direction, Li *et al.* [50] investigate cross-domain document object detection where a detector is trained to detect objects in the target domain using only labeled data from the source domain.

Performance of Faster R-CNN is reduced when the document contains large-scale variate tables. Siddiqui *et al.* [5] incorporate deformable CNN in Faster R-CNN to adapt to different scales and transformations, which allow the model to detect table accurately. Sun *et al.* [8] combined corner information with the detected table region by Faster R-CNN to refine the detected table's boundary, reducing false positives. While Prasad *et al.* [12] propose a CascadeTab-Net to solve two sub-problems — table detection and cell detection in a single framework which improves the detection accuracy. Riba *et al.* [46] propose a table detection technique using GNN to learn the inherent structure present in the table and obtain improved accuracy. In the recent work, Ma et al. [51] use CornerNet as a new region proposal network to generate higher-quality table proposals for Faster R-CNN for detecting tables in heterogeneous document images. Due to the use of a higher-quality region proposal, the proposed approach has significantly improved the localization accuracy of Faster R-CNN for table detection.

In most detection algorithms [42,44], the Intersection over Union (IoU) threshold is frequently used to define positives/negatives. Most of the existing document object detection methods used this threshold to determine output quality. While most of them used the threshold of 0.5, which leads to an inaccurate bounding box of the document object, and detection performance frequently degrades for more significant thresholds (more than 0.5). Information extraction from the document objects requires an accurate bounding box for further analysis.

3. DOLNet: Document object localization network

The success of deep Convolution Neural Networks (CNNs) for solving various computer vision problems inspire researchers to explore and design models for detecting objects (mainly tables) in document images [3–8,10,11,14,16]. All these deep models provide high detection accuracy. However, all these models suffer from the following shortcomings — (i) all existing document object detection networks use a backbone to extract features for detecting document objects that are 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 document object detection may result in sub-optimal performance. A more powerful backbone is needed to extract more representational features and improve detection accuracy.

However, it is very expensive to train a deeper and more powerful backbone on ImageNet and get better performance. (ii) CNNs have limitations modeling large transformations due to fixed geometric module structures — a convolution filter samples the input feature map corresponding to a fixed location, a pooling layer reduces the spatial resolution at a fixed ratio and an RoI into a fixed spatial bin. It leads to the lack of handling of the geometric transformations. (iii) All these document object detectors use the Intersection over Union (IoU) threshold to define positives, negatives, and detection quality. They commonly use a threshold of 0.5, which leads to noisy (low-quality) detection and frequently degrades the performance for higher thresholds. The major hindrance in training a network at a higher IoU threshold is reducing positive training samples with an increasing IoU threshold. These issues are also a bottleneck of CNNs based object detection techniques [41,42,44] in natural scene images.

Over time, various solutions [24,26,27] are proposed to handle the above-stated problems for object detection in natural images. Lie *et al.* [24] proposed a CBNet that stacks 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.* [27] 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. The output of one detector is fed as an input to the subsequent detector, maintaining the number of positive samples at higher thresholds.

Inspired by the solutions provided by [24,26,27] for issues discussed earlier in natural scene images, we propose a novel architecture DOLNet for detecting document objects accurately in the documents. It comprises a cascade Mask R-CNN with a composite backbone having deformable convolution filters instead of conventional convolution filters. Fig. 1 displays an overview of our proposed architecture for document object localization in document images. We discuss each component of DOLNet in detail:

3.1. Cascade mask R-CNN

Cai and Vasconcelos [26] proposed Cascade R-CNN a multi-stage extension of Faster R-CNN [42]. 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. DOLNet comprises a sequence of three detectors trained with increasing IoU thresholds of 0.5, 0.6, and 0.7, respectively. The proposals generated by the RPN network are passed through the 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. Rather than the initial distribution, each regressor is optimized for the bounding box distribution generated by the previous regressor. The bounding box regressor trained for a certain IoU threshold tends to produce higher IoU threshold bounding boxes. It helps in re-sampling an example distribution of a 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, enabling the network to train on higher IoU threshold values.

3.2. Composite backbone

We use a dual backbone-based architecture [24] which creates a composite connection between the parallel stages of two adjacent ResNext-101 backbones (one called the assistant backbone and the other called lead backbone). The high-level output features of the assistant backbone are fed as input to the corresponding stage of the lead backbone. In a conventional network, the output (denoted by $\mathbf{x}^l$) of previous $l-1$ stages is fed as input to the l^{th} stage, given by:

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

where $F^l(.)$ is a non-linear transformation operation of l^{th} stage. However, our network takes input from previous stages and the

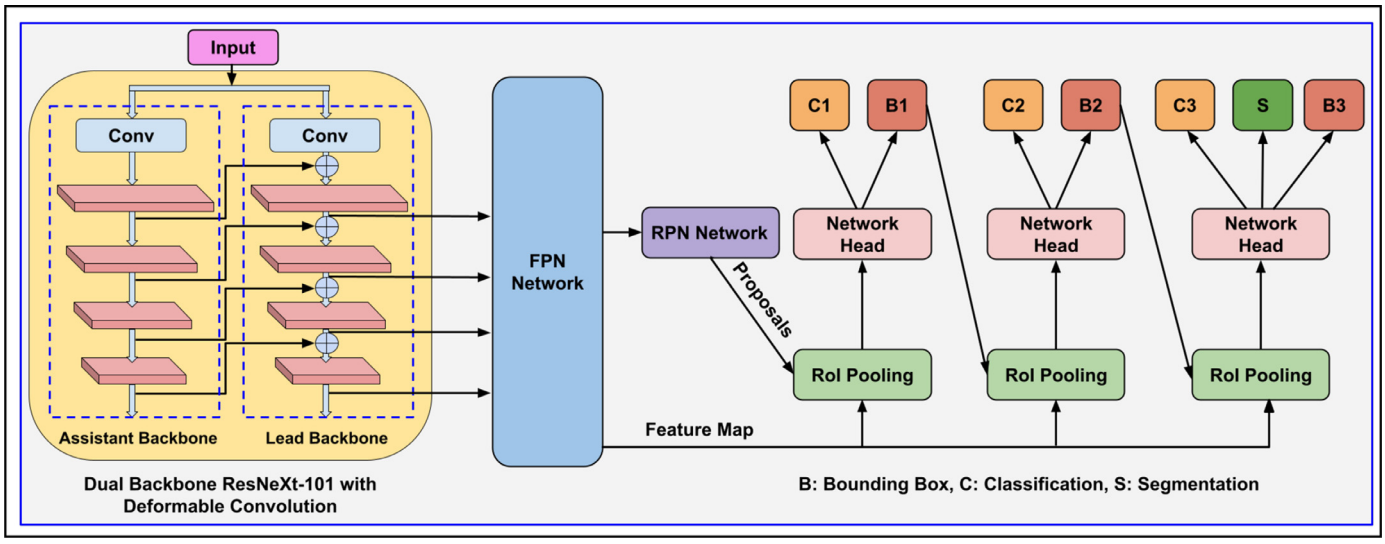

Fig. 1. Illustration of the proposed DOLNet composed of Cascade Mask R-CNN with composite backbone having deformable convolution instead of conventional convolution.

parallel stage of the assistant backbone. For a given stage l of the lead backbone (lb), input is a combination of the output of previous $l-1$ stages of the lead backbone and parallel l^{th} stage of the assistant backbone (b), given by:

$$\mathbf{x}_{lb}^{l} = F_{lb}^{l}(\mathbf{x}_{k}^{l-1} + q(\mathbf{x}_{b}^{l})), \quad l \geq 2, \tag{2}$$

where $q(.)$ 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.

3.3. Deformable convolution

The commonly used backbone, ResNext architectures, have 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 field. It performs well for layers at the lower hierarchy. Still, when the objects appear at arbitrary scales and transformations, generally at a higher level, the convolution operation does not perform well in capturing the features. We replace the fixed receptive field CNN with deformable CNN [27] in each of our dual backbone architectures. The grid is deformable as each grid point can be moved by a learnable offset. In a conventional convolution, we sample over the input feature map $\mathbf{x}$ using a regular grid R, given by

$$\mathbf{z}(p_0) = \sum_{p_n \in R} w(p_n)\ \mathbf{x}(p_0 + p_n). \tag{3}$$

Whereas in a deformable convolution, for each location p_0 on the output feature map $\mathbf{z}$, we augment the regular grid using the offset Δp_n such that $\{\Delta p_n | n = 1, ..., N\}$, where $N = |R|$, given by

$$\mathbf{z}(p_0) = \sum_{p_n \in R} w(p_n)\ \mathbf{x}(p_0 + p_n + \Delta p_n). \tag{4}$$

Deformable convolution is operated on R, with each point augmented by a learnable offset Δp_n. The offset value, Δp_n, is a trainable parameter. It helps to enable 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 Fig. 2.

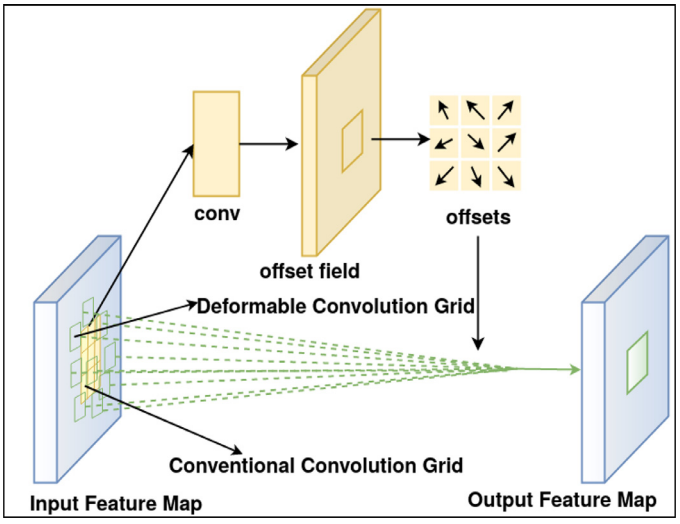

Fig. 2. Illustration of the deformable convolution. The offsets are obtained by applying a convolutional layer over the input feature map.

3.4. Loss function

At each stage t, the proposed DOLNet head includes a classifier h_t (C in Fig. 1) and a bounding box regressor f_t (B in Fig. 1). The classifier and regressor are optimized for the corresponding IoU threshold u^t, where $u^t > u^{t-1}$. The parameters of the classifier and regressor are learned with the loss function defined as

$$L(\mathbf{x}^t, g) = L_{cls}(h_t(\mathbf{x}^t), y^t) + \lambda[y^t \geq 1] L_{loc}(f_t(\mathbf{x}^t, \mathbf{b}^t), \mathbf{g}), \tag{5}$$

where $\mathbf{b}^t = f_{t-1}(\mathbf{x}^{t-1}, \mathbf{b}^{t-1})$, $\mathbf{g}$ is the ground truth document object for the document image patch $\mathbf{x}^t$, $\lambda = 1$ is the trade-off coefficient, y^t is the label of $\mathbf{x}^t$ under the IoU threshold u^t, and according to [27], $[\cdot]$ is the indicator function. L_{cls} and L_{loc} are classification and bounding box regression losses. The use of $[\cdot]$ implies that the IoU threshold u of bounding box regression is identical to that used for classification. This cascade learning has three important consequences for detector training. First, the potential for over-fitting at large IoU thresholds u is reduced since positive examples become plentiful at all stages. Second, detectors of deeper stages are optimal for higher IoU thresholds. Third, because some outliers are removed as the IoU threshold increases, the learning effectiveness

of bounding box regression increases in the later stages. It simultaneous improvement of hypotheses and detector quality enables the DOLNet to beat the paradox of high-quality document object detection. At inference, the same cascade is applied. The quality of the hypotheses is improved sequentially, and higher-quality detectors are only required to operate on higher-quality hypotheses, for which they are optimal.

Classification:

The classifier is a function $h(\mathbf{x})$ that assigns a document image patch $\mathbf{x}$ to one of $M+1$ document object classes, where class 0 contains background and the remaining classes of the document objects to detect. $h(\mathbf{x})$ is a $M+1$-dimensional estimate of the posterior distribution over classes, i.e., $h_k(\mathbf{x}) = p(y = k|\mathbf{x})$, where y is the class label. Given a training set $(\mathbf{x}_i, y_i)$, it is learned by minimizing the classification risk

$$R_{cls}[h] = \sum_i L_{cls}(h(\mathbf{x}_i, y_i)), \tag{6}$$

where

$$L_{cls}(h(\mathbf{x}), y) = -log[h_y(\mathbf{x})] \tag{7}$$

is the cross-entropy loss.

Bounding Box Regression:

A bounding box $\mathbf{b} = (b_x, b_y, b_w, b_h)$ contains the four coordinates of a document image patch $\mathbf{x}$. Bounding box regression aims to regress a candidate bounding box $\mathbf{b}$ into a target bounding box $\mathbf{g}$ using a regressor $f(\mathbf{x}, \mathbf{b})$. This is learned from a training set $(\mathbf{g}_i, \mathbf{b}_i)$, by minimizing the risk

$$R_{loc}[f] = \sum_i L_{loc}(f(\mathbf{x}_i, \mathbf{b}_i), \mathbf{g}_i). \tag{8}$$

According to the work [41],

$$L_{loc}(\mathbf{a}, \mathbf{b}) = \sum_{i \in \{x,y,w,h\}} S(a_i - b_i), \tag{9}$$

where

$$S(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise,} \end{cases} \tag{10}$$

is the smooth function.

3.5. Implementation details

We implement DOLNet in Pytorch using MMdetection toolbox [52]. 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 with FPN as the network head. We train DOLNet 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 the warm-up schedule for the first 500 iterations. DOLNet 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 epochs and 6 epochs. In the case of fine-tuning, we use 12 epochs in total. Our model uses three IoU threshold values — 0.5, 0.6, and 0.7. We use 0.5, 1.0, and 2.0 as anchor ratios with a single anchor scale of 8. We set the batch size to 1 during training and 512 RoIs per document image. The source code is available at[1].

[1] https://github.com/mdv3101/CDeCNet

Table 1
Shows the statistics of used data sets for the experiments. **T:** table, **F:** figure, **E:** equation, **NI:** natural image, **L:** logo, **S:** signature, **TL:** title, **TT:** text, **LT:** list, **AT:** abstract, **ACK:** acknowledgment, **AN:** affiliation, **AR:** author, **B:** bibliography information, **BY:** body, **CT:** conflict statement, **CR:** copyright, **CE:** correspondence, **D:** dates, **ER:** editor, **GY:** glossary, **KS:** keywords, **PN:** page number, **R:** references, **TA:** title author, **TE:** type, **U:** unknown, **C:** caption, **p:** paragraph, **SN:** section and **FR:** footer.

Dataset	Category Label	Training Set	Validation Set	Test Set
ICDAR-2013 [17]	1: T	-	-	238
ICDAR-POD-2017 [15]	3: T, F and E	1600		817
UNLV [19]	1: T	-	-	424
Marmot [21]	1: T	2K	-	-
ICDAR-2019 [18]	1: T	1200		439
IIIT-AR-13K [23]	5: T, F, NI, L and S	9K	2K	2K
DeepFigures[a] [20]	2: T and F	5.5M	-	-
TableBank-word[a] [7]	1: T	163K	1K	1K
TableBank-Latex[a] [7]	1: T	253K	1K	1K
TableBank-both[a] [7]	1: T	417K	2K	2K
PubLayNet[a] [16]	5: T, F, TL, TT and LT	340K	11K	11K
DocBank[a] [22]	12: AT, AR, C, E, F, FR, LT, P, R, SN, T and TL	400K	50K	50K

[a] Ground truth bounding boxes are annotated automatically.

4. Experimental setting

4.1. Dataset

We use publicly available benchmark datasets — ICDAR 2013 table competition (i.e., ICDAR-2013) dataset [17], ICDAR 2017 competition on page object detection (i.e., ICDAR-POD-2017) dataset [15], ICDAR 2019 competition on table detection and recognition (i.e., CTDaR) dataset [18], UNLV [19], DeepFigures [20], PubLayNet [16], Marmot table recognition dataset [21], TableBank [7], GroTOAP2 [53], DocBank [22] and IIIT-AR-13K [23] for document object detection task.

Table 1 shows statistics of the used dataset. We observe from the table that datasets (except ICDAR-POD-2017, DeepFigures, PubLayNet, GroTOAP2 and DocFigure) containing only one object category (e.g., table), are subsets of IIIT-AR-13K dataset (containing five object categories). While DeepFigures is the largest dataset, the ground truths are automatically annotated. On the other hand, IIIT-AR-13K is the manually annotated largest dataset. Fig. 3 shows sample page images of the existing datasets. In [23], Mondal *et al.* establish the effectiveness of IIIT-AR-13K dataset over larger datasets (e.g., TableBank and PubLayNet for localization of tables in document images.

4.2. Evaluation measures

We use various popular existing measures — Precision, Recall, F-measure, and mean Average Precision (mAP) to evaluate the performance of the proposed techniques for localizing document objects. We define True Positive (TP), False Positive (FP), and False Negative (FN) to calculate precision, recall, and f-measure. We use the concept of Intersection over Union (IoU), which computes intersection over the union of the two bounding boxes — the bounding box for the ground truth and the predicted bounding box of an object category.

- **TP:** if IoU $\geq T_h$, we classify the detected object as True Positive (TP).
- **FP:** if IoU $< T_h$, we classify the detected object as False Positive (FP).
- **FN:** when ground truth is present in the image and model fails to detect the object, we classify it as False Negative (FN).

Here $T_h \in [0.5, 0.95]$ is a pre-defined IoU threshold. Precision, Recall, F-measure and Average Precision are defined as

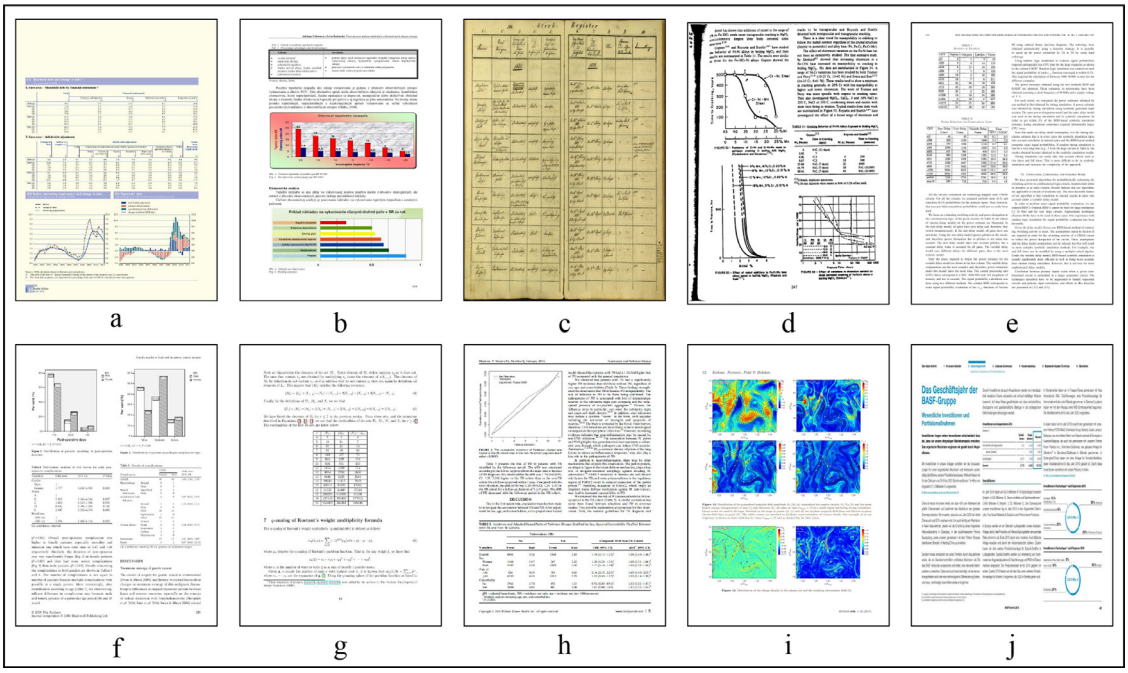

Fig. 3. Shows sample page images of publicly available benchmark datasets — (a) ICDAR-2013, (b) ICDAR-POD-2017, (c) ICDAR-2019, (d) UNLV, (e) Marmot, (f) PubLayNet, (g) TableBank, (h) GROTOAP2, (i) DocBank, and (j) IIIT-AR-13K.

- $Precision = \frac{TP}{TP+FP}$.
- $Recall = \frac{TP}{TP+FN}$.
- $F-measure = \frac{2.Precision.Recall}{Precision+Recall}$.
- **Average Precision (AP):** We average precision at a set of 11 spaced recall points (0, 0.1, 0.2,., 1) where we interpolate the corresponding precision for a certain recall value r by taking the maximum precision whose recall value $\tilde{r} > r$.

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, 0.2, .., 1\}} p_{interp}(r), \tag{11}$$

where $p_{interp}(r) = \max_{\tilde{r} > r} p(\tilde{r})$.

- **mean Average Precision (mAP):** We average APs corresponding to all object categories present in a document.

We use two strategies to calculate all these measures.

Strategy-A: We use a single value of Intersection over Union (IoU) threshold of 0.5 to calculate all these values.

Strategy-B: We use multiple values of (IoU) thresholds of 0.50:0.05:0.95 to calculate all these values. We average all these measures over multiple values of (IoU) thresholds. Precision, Recall, F-measure, and AP are averaged over multiple IOU values.

For a fair comparison, we evaluate the proposed DOLNet on the same IOU threshold values as mentioned in the respective existing works. We perform multi-scale testing at 7 different scales (with 3 smaller scales, original scale, and 3 larger scales). We select detection output as a final result if it presents at least 4 test cases out of 7 scales. It helps to eliminate false positives and provides consistent results.

5. Object detection and fine tuning

5.1. Document object detection

We explore DOLNet for detecting various objects (e.g., table, figure, natural image, logo, signature, mathematical equation, etc.) present in documents. We use existing benchmark datasets PubLayNet and IIIT-AR-13K to evaluate the performance of DOLNet for detecting various document objects.

Comparison with State-of-the-Arts on PubLayNet:

Table 2 shows performance comparison between DOLNet with the state-of-the-art methods F-RCNN [16] and M-RCNN [16] for detecting various document objects. We observe that among existing techniques, M-RCNN performs better than F-RCNN. The models perform worst on titles as the title appears less in the document than in other categories. DOLNet performs better than the existing techniques on table, figure, and list categories. DOLNet obtains 1.8%, 2.6%, and 3.6% better than M-RCNN on a table, figure, and list, respectively. Those object categories have more regular shapes and more distinctive differences from other object categories.

Comparison with State-of-the-Arts on IIIT-AR-13K:

We provide performance comparison between DOLNet and the state-of-the-art techniques Mask R-CNN [23] in Table 3. DOLNet obtains better detection accuracy for all categories of objects than Mask R-CNN. Among all categories, DOLNet obtains the highest (3.82%) improved detection result on the logo category than other categories. Since DOLNet uses a dual backbone with a deformable convolutional filter, it can handle document objects of various scales.

5.2. Table detection

In this section, we discuss the obtained results of table localization in documents using the existing techniques and the proposed DOLNet. We train the models using training images with a single category to localize tables in the test documents. We follow a similar training procedure used in the existing techniques to localize tables in the respective datasets. We compare the performance of DOLNet on table detection task with state-of-the-art methods for individual datasets.

Comparison with State-of-the-Arts on ICDAR-2013:

We compare the results using DOLNet with state-of-the-art techniques on ICDAR-2013. We follow the same training procedures used in each of the existing training methods and evaluate our DOLNet on ICDAR-2013. Table 4 shows the comparison with state-of-the-art techniques. The table shows that each method uses a different training procedure to detect tables in documents. Under

Table 2
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on PubLayNet dataset. **TL:** title, **TT:** text, **LT:** list, **T:** table, **F:** figure, **Ave.:** average, **#:** no. of images. Values in bold indicate the best results.

	Training		Validation		IoU	**Score (mAP)**↑					
Method	Dataset	#	Dataset	#		TT	TL	LT	T	F	Ave.
F-RCNN [16]	PubLayNet	340K	PubLayNet	11K	[0.50:0.95]	0.910	0.826	0.883	0.954	0.937	0.902
M-RCNN [16]	PubLayNet	340K	PubLayNet	11K	[0.50:0.95]	**0.916**	**0.840**	0.886	0.960	0.949	**0.910**
DOLNet	PubLayNet	340K	PubLayNet	11K	[0.50:0.95]	0.844	0.663	**0.929**	**0.978**	**0.977**	0.878

Table 3
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on IIIT-AR-13K dataset. **T:** table, **F:** figure, **NI:** natural image, **L:** logo, **S:** signature, **Ave.:** average and **#:** no. of images. Values in bold indicate the best results.

	Training		Test		IoU	**Score (mAP)**↑					
Method	Dataset	#	Dataset	#		T	F	NI	L	S	Ave.
Mask R-CNN [23]	IIIT-AR-13K	9K	IIIT-AR-13K	2K	0.5	0.965	0.869	0.895	0.469	0.912	0.822
DOLNet	IIIT-AR-13K	9K	IIIT-AR-13K	2K	0.5	**0.982**	**0.872**	**0.929**	**0.851**	**0.989**	**0.925**

Table 4
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on ICDAR-2013 dataset. **A:** anchor optimization, **PG:** post-processing technique, **SF:** semantic features, **D1:** Marmot+UNLV+ICDAR-POD-2017, *: the authors reported 0.996 in table however in discussion they mentioned 0.994. **#:** no. of images. Values in bold indicate the best results.

	Training		Fine-tuning		Test		IoU	Score			
Method	Dataset	#	Dataset	#	Dataset	#		**R**↑	**P**↑	**F1**↑	**mAP**↑
DeCNT [5]	D1	4808	-	-	ICDAR-2013	238	0.5	0.996*	0.996*	0.996*	-
DOLNet	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**	-
DOLNet	Marmot	2K	-	-	ICDAR-2013	238	0.5	**1.000**	0.981	**0.991**	**0.995**
F-RCNN [16]	PubLayNet	340K	ICDAR-2013	170	ICDAR-2013	238	0.5	0.964	0.972	0.968	
M-RCNN [16]	PubLayNet	340K	ICDAR-2013	170	ICDAR-2013	238	0.5	0.955	0.940	0.947	-
DOLNet	PubLayNet	340K	ICDAR-2013	170	ICDAR-2013	238	0.5	**0.968**	**0.987**	**0.977**	**0.959**
YOLOv3+A+PG [54]	ICDAR-2017	1.6K	-	-	ICDAR-2013	238	0.5	0.949	1.000	0.973	-
DOLNet	ICDAR-2017	1.6K	-	-	ICDAR-2013	238	0.5	**1.000**	**1.000**	**1.000**	**1.000**
Khan *et al.* [55]	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 [4]	Marmot	2K	ICDAR-2013	204	ICDAR-2013	34	0.5	0.962	0.974	0.968	-
DOLNet	Marmot	2K	ICDAR-2013	204	ICDAR-2013	34	0.5	**1.000**	**1.000**	**1.000**	**1.000**
M-RCNN [11]	Pascal VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.770	0.140	0.230	-
RetinaNet [11]	Pascal VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.580	0.560	0.570	-
SSD [11]	Pascal VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.680	0.540	0.600	-
YOLO [11]	Pascal VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.580	0.920	0.750	-
DOLNet	Pascal VOC	16K	ICDAR-2013	178	ICDAR-2013	60	0.6	**0.844**	**1.000**	**0.922**	**0.844**
M-RCNN [11]	TableBank	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	**0.970**	0.700	0.810	-
RetinaNet [11]	TableBank	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.770	0.830	0.800	-
SSD [11]	TableBank	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.680	0.620	0.650	-
YOLO [11]	TableBank	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.650	**1.000**	0.780	-
DOLNet	TableBank	199K	ICDAR-2013	178	ICDAR-2013	60	0.6	0.933	**1.000**	**0.967**	**0.933**

each training procedure, our DOLNet obtains the best results compared to the existing techniques. Our proposed model obtains (F1 score 1.0) better results than state-of-the-art technique — DeCNT (with F1 score 0.996) for detecting table in ICDAR-2013. We have many trained (DOLNet) models under the various training procedures corresponding to the existing works.

Comparison with State-of-the-Arts on ICDAR-POD-2017:

Table 5 shows the comparison results on ICDAR-POD-2017 with the existing techniques. We observe from the table that YOLOv3‡+A+P [54] obtains (F1 scores 0.975 and 0.971) the state-of-the-art performance on ICDAR-POD-2017 for both threshold values 0.6 and 0.8, respectively. This particular technique uses YOLO with anchor optimization and post-processing to detect tables, which obtains the best results for this dataset. The proposed CDeCNet, without any post-processing, obtains comparable outputs (F1 scores 0.954 and 0.947) for thresholds 0.6 and 0.8, respectively.

Comparison with State-of-the-Arts on ICDAR-2019:

The performance of the proposed DOLNet is compared with state-of-the-art techniques on ICDAR-2019 and reported in Table 6. State-of-the-art method TableRadar [18] obtain the best F1 score 0.945 on IoU threshold 0.8. Our DOLNet obtain the best F1 score 0.913 on IoU threshold 0.9. The result shows that DOLNet is more robust than the state-of-the-art method TableRadar [18]. In the setting of the model trained with Pascal VOC and TableBank-LATEX and fine-tuned with archive images, DOLNet obtains the best results (F1 scores 0.971 and 0.954, respectively which are better (0.041 and 0.004) than the state-of-the-art method YOLO [11]) on archive images.

Comparison with State-of-the-Arts on UNLV:

Table 7 presents the comparison results between DOLNet and state-of-the-art methods under the various experimental environments. In a practical environment like - training with Marmot, fine-tuning with UNLV and test on UNLV, our DOLNet obtains the best result (F1 score 0.938), which is 0.010 more than GOD [10] result (F1 score 0.928). While the model trained with UNLV and tested on UNLV, our DOLNet also obtains the best result (F1 score 0.910) as compared to state-of-the-art technique Gilani *et al.* [3] (F1 score 0.863).

Table 5
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on ICDAR-POD-2017. **A:** anchor optimization. **P:** post-processing. **D2:** ICDAR-2013+ICDAR-POD-2017+UNLV+Marmot. ‡: model trained with multiple object categories. **#:** no. of images. Values in bold indicates the best results.

	Training		Fine-tuning		Test		IoU	Score			
Method	Dataset	#	Dataset	#	Dataset	#		**R↑**	**P↑**	**F1↑**	**mAP↑**
FastDetectors‡ [15]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.6	0.940	0.903	0.921	0.925
PAL‡ [15]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.6	0.953	0.968	0.960	0.933
GOD‡ [10]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.6	-	-	0.971	**0.989**
DSP-SC‡ [36]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.6	0.962	0.974	0.968	0.946
YOLOV3‡+A+P [54]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.6	**0.972**	**0.978**	**0.975**	-
DOLNet‡	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.6	0.931	0.977	0.954	0.920
FastDetectors‡ [15]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.8	0.915	0.879	0.896	0.884
PAL‡ [15]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.8	0.943	0.958	0.951	0.911
GOD‡ [10]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.8	-	-	0.968	**0.974**
DSP-SC‡ [36]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.8	0.953	0.965	0.959	0.923
YOLOV3‡+A+P [54]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.8	**0.968**	**0.975**	**0.971**	-
DOLNet‡	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.8	0.924	0.970	0.947	0.912
DECNT [5]	D2	4229	-	-	ICDAR-2017	817	0.6	**0.971**	0.965	**0.968**	-
DOLNet	D2	4229	-	-	ICDAR-2017	817	0.6	0.943	**0.977**	0.960	**0.938**
DECNT [5]	D2	4229	-	-	ICDAR-2017	817	0.8	**0.937**	**0.967**	**0.952**	-
DOLNet	D2	4229	-	-	ICDAR-2017	817	0.8	0.918	0.951	0.935	**0.895**
Faster R-CNN+CL [8]	ICDAR-2017	549	-	-	ICDAR-2017	243	0.6	**0.956**	0.943	0.949	-
F+M-RCNN [56]	ICDAR-2017	549	-	-	ICDAR-2017	243	0.6	0.944	0.944	0.944	-
DOLNet	ICDAR-2017	549	-	-	ICDAR-2017	243	0.6	0.943	**0.974**	**0.959**	**0.931**
F+M-RCNN [56]	ICDAR-2017	549	-	-	ICDAR-2017	243	0.8	0.903	0.903	0.903	-
DOLNet	ICDAR-2017	549	-	-	ICDAR-2017	243	0.8	**0.928**	**0.958**	**0.943**	**0.902**
M-RCNN [11]	Pascal VOC	16K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.850	0.320	0.460	-
RetinaNet [11]	Pascal VOC	16K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.860	0.650	0.740	-
SSD [11]	Pascal VOC	16K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.710	0.490	0.580	-
YOLO [11]	Pascal VOC	16K	ICDAR-2017	1200	ICDAR-2017	400	0.6	**0.940**	0.900	0.920	-
DOLNet	Pascal VOC	16K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.932	**0.981**	**0.956**	**0.925**
M-RCNN [11]	TableBank	199K	ICDAR-2017	1200	ICDAR-2017	400	0.6	**0.950**	0.720	0.820	-
RetinaNet [11]	TableBank	199K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.870	0.920	0.890	-
SSD [11]	TableBank	199K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.710	0.550	0.620	-
YOLO [11]	TableBank	199K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.940	0.940	0.940	-
DOLNet	TableBank	199K	ICDAR-2017	1200	ICDAR-2017	400	0.6	0.914	**0.980**	**0.947**	**0.905**

Table 6
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on ICDAR-2019. **#:** no. of images. Values in bold indicates the best results.

	Training		Fine-tuning		Test		IoU	Score			
Method	Dataset	#	Dataset	#	Dataset	#		**R↑**	**P↑**	**F1↑**	**mAP↑**
TableRadar [18]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.8	**0.940**	0.950	**0.945**	-
NLPR-PAL [18]	ICDAR-2019	1200			ICDAR-2019	439	0.8	0.930	0.930	0.930	-
Lenovo Ocean [18]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.8	0.860	0.880	0.870	-
DOLNet	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.8	0.934	**0.953**	0.944	**0.922**
TableRadar [18]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	0.890	0.900	0.895	-
NLPR-PAL [18]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	0.860	0.860	0.860	-
Lenovo Ocean [18]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	0.810	0.820	0.815	-
DOLNet	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	**0.904**	**0.922**	**0.913**	**0.843**
M-RCNN [11]	Pascal VOC	16K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.640	0.600	0.620	-
RetinaNet [11]	Pascal VOC	16K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.660	0.860	0.740	-
SSD [11]	Pascal VOC	16K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.350	0.310	0.330	-
YOLO [11]	Pascal VOC	16K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.910	0.950	0.930	-
DOLNet	Pascal VOC	16K	ICDAR-2019	599	ICDAR-2019	198	0.6	**0.962**	**0.981**	**0.971**	**0.949**
M-RCNN [11]	TableBank	199K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.850	0.760	0.810	-
RetinaNet [11]	TableBank	199K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.740	0.910	0.820	-
SSD [11]	TableBank	199K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.350	0.350	0.350	-
YOLO [11]	TableBank	199K	ICDAR-2019	599	ICDAR-2019	198	0.6	**0.950**	0.950	0.950	-
DOLNet	TableBank	199K	ICDAR-2019	599	ICDAR-2019	198	0.6	0.924	**0.984**	**0.954**	**0.909**

In the case of training with private data set and testing on complete data set, Arif and Shafait [6] obtain better F1 score (0.896) than DOLNet (F1 score 0.829). While training with D4 i.e., ICDAR-2013+ICDAR-2017+Marmot data set, our DOLNet obtains the best F1 score 0.794 as compared to DECNT [5] with F1 score 0.767. Existing environments like training with Pascal VOC and TableBank and fine-tuning with UNLV and testing on UNLV, our DOLNet obtains better results than the state-of-the-art technique YOLO [11]. Our single model DOLNet† trained with IIIT-AR-13K and fine-tuned with respective datasets obtain better results than DOLNet on existing experiment environments.

Comparison with State-of-the-Arts on Marmot:

Table 8 shows the comparison between the performances of DOLNet and the existing techniques under various experimental

Table 7
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on UNLV. **D4:** ICDAR-2013+ICDAR-2017+Marmot. **#:** no. of images. Values in bolt indicates the best results.

	Training		Fine-tuning		Test		IoU	Score			
Method	Dataset	#	Dataset	#	Dataset	#		**R↑**	**P↑**	**F1↑**	**mAP↑**
GOD [10]	Marmot	2K	UNLV	340	UNLV	84	0.5	0.910	0.946	0.928	-
DOLNet	Marmot	2K	UNLV	340	UNLV	84	0.5	**0.925**	**0.952**	**0.938**	**0.912**
Gilani *et al.* [3]	UNLV	340	-	-	UNLV	84	0.5	**0.907**	0.823	0.863	-
DOLNet	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**	-
DOLNet	private	1019	-	-	UNLV	427	0.5	0.745	**0.912**	0.829	**0.711**
DeCNT [5]	D4	4622	-	-	UNLV	424	0.5	**0.749**	0.786	0.767	-
DOLNet	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**	-
DOLNet	Pascel VOC	16K	UNLV	302	UNLV	101	0.6	0.805	**0.961**	0.883	**0.788**
M-RCNN [11]	TableBank	199K	UNLV	302	UNLV	101	0.6	0.830	0.660	0.740	-
RetinaNet [11]	TableBank	199K	UNLV	302	UNLV	101	0.6	0.830	0.810	0.820	-
SSD [11]	TableBank	199K	UNLV	302	UNLV	101	0.6	0.660	0.720	0.690	-
YOLO [11]	TableBank	199K	UNLV	302	UNLV	101	0.6	**0.950**	0.930	0.940	-
DOLNet	TableBank	199K	UNLV	302	UNLV	101	0.6	0.894	**0.991**	**0.943**	**0.889**

Table 8
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on Marmot. D3: ICDAR-2013+ICDAR-2017+UNLV. **E:** English and **C:** Chinese. **#:** no. of images. Values in bold indicates the best results.

	Training		Fine-tuning		Test		IoU	Score			
Method	Dataset	#	Dataset	#	Dataset	#		**R↑**	**P↑**	**F1↑**	**mAP↑**
DeCNT [5]	D3	3079	-	-	Marmot	1967	0.5	**0.946**	0.849	0.895	-
DOLNet	D3	3079			Marmot	1967	0.5	0.930	**0.975**	**0.952**	**0.911**
MFCN+contour +CRF [57]	Various Doc	130	-	-	Marmot	2000	0.8	0.731	0.762	0.747	-
DOLNet	Various Doc	130			Marmot	2000	0.8	**0.836**	**0.845**	**0.840**	**0.716**
MFCN+contour +CRF [57]	Various Doc	130	-	-	Marmot	2000	0.9	0.471	0.481	0.476	-
DOLNet	Various Doc	130			Marmot	2000	0.9	**0.765**	**0.774**	**0.769**	**0.600**
M-RCNN [11]	Pascal VOC	16K	Marmot(E)	744	Marmot(E)	249	0.6	0.750	0.370	0.490	-
RetinaNet [11]	Pascal VOC	16K	Marmot(E)	744	Marmot(E)	249	0.6	0.860	0.750	0.800	-
SSD [11]	Pascal VOC	16K	Marmot(E)	744	Marmot(E)	249	0.6	0.760	0.670	0.710	-
YOLO [11]	Pascal VOC	16K	Marmot(E)	744	Marmot(E)	249	0.6	**0.960**	0.900	0.930	-
DOLNet	Pascel VOC	16K	Marmot(E)	744	Marmot(E)	249	0.6	0.946	**0.993**	**0.969**	**0.942**
M-RCNN [11]	TableBank	199K	Marmot(E)	744	Marmot(E)	249	0.6	0.930	0.720	0.810	-
RetinaNet [11]	TableBank	199K	Marmot(E)	744	Marmot(E)	249	0.6	0.860	0.930	0.900	-
SSD [11]	TableBank	199K	Marmot(E)	744	Marmot(E)	249	0.6	0.750	0.710	0.730	-
YOLO [11]	TableBank	199K	Marmot(E)	744	Marmot(E)	249	0.6	**0.970**	0.950	**0.960**	-
DOLNet	TableBank	199K	Marmot(E)	744	Marmot(E)	249	0.6	0.925	**0.993**	0.959	**0.924**
M-RCNN [11]	Pascal VOC	16K	Marmot(C)	754	Marmot(C)	252	0.6	0.830	0.520	0.640	-
RetinaNet [11]	Pascal VOC	16K	Marmot(C)	754	Marmot(C)	252	0.6	0.850	0.780	0.810	-
SSD [11]	Pascal VOC	16K	Marmot(C)	754	Marmot(C)	252	0.6	0.700	0.570	0.630	-
YOLO [11]	Pascal VOC	16K	Marmot(C)	754	Marmot(C)	252	0.6	0.960	0.950	0.960	-
DOLNet	Pascal VOC	16K	Marmot(C)	754	Marmot(C)	252	0.6	**0.966**	**0.988**	**0.977**	**0.959**
M-RCNN [11]	TableBank	199K	Marmot(C)	754	Marmot(C)	252	0.6	0.980	0.820	0.890	-
RetinaNet [11]	TableBank	199K	Marmot(C)	754	Marmot(C)	252	0.6	0.870	0.870	0.870	-
SSD [11]	TableBank	199K	Marmot(C)	754	Marmot(C)	252	0.6	0.670	0.610	0.640	-
YOLO [11]	TableBank	199K	Marmot(C)	754	Marmot(C)	252	0.6	0.930	0.970	0.950	-
DOLNet	TableBank	199K	Marmot(C)	754	Marmot(C)	252	0.6	**0.966**	**0.994**	**0.980**	**0.962**

conditions. We observe from the table that DOLNet obtains better results than the state-of-the-art techniques — DeCNT [5] and MFCN+contour+CRF [57] when a complete set of images is used for evaluation purposes. Under the training with Pascal VOC and TableBank-LaTeX, DOLNet obtains 3.9%, 1.7%, and 3.0% better F1 score that the state-of-of-art method YOLO [11] for English and Chinese documents.

Comparison with State-of-the-Arts on TableBank:

The comparison between the performances of DOLNet and the existing techniques under various experimental conditions is presented in Table 9. DOLNet obtains better F1 scores than the state-of-the-art techniques Li *et al.* [7] on TableBank excepting Tabelbank-word images.

5.3. Comparison with state-of-the-Art models parameters

Table 10 presents the comparison between parameters of DOLNet and the existing table detection models. Table highlights that compared to all existing table detection models, DOLNet has 144M trainable parameters, two times the existing model's parameters. Due to the dual/composite backbone, DOLNet has

Table 9
Illustrates comparison between the proposed DOLNet and state-of-the-art techniques on TableBank. **#:** no. of images, **L:** Latex and **W:** Word. Values in bold indicates the best results.

	Training		Fine-tuning		Test		IoU	Score			
Method	Dataset	#	Dataset	#	Dataset	#		**R↑**	**P↑**	**F1↑**	**mAP↑**
Li *et al.* [7]	TableBank-L	253K	-	-	TableBank-W	1K	0.5	**0.956**	0.826	**0.886**	-
					TableBank-L	1K	0.5	0.975	0.987	0.981	-
					TableBank-both	2K	0.5	**0.962**	0.872	0.915	-
DOLNet	TableBank-L	253K	-	-	TableBank-W	1K	0.5	0.868	**0.873**	0.871	**0.762**
					TableBank-L	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-L	199K	-	-	TableBank-L	1K	0.6	0.980	0.960	0.940	-
RetinaNet [11]	TableBank-L	199K	-	-	TableBank-L	1K	0.6	0.860	0.980	0.920	-
SSD [11]	TableBank-L	199K	-	-	TableBank-L	1K	0.6	0.970	0.960	0.965	-
YOLO [11]	TableBank-L	199K	-	-	TableBank-L	1K	0.6	**0.990**	0.980	0.985	-
DOLNet	TableBank-L	199K	-	-	TableBank-L	1K	0.6	0.978	**0.995**	**0.986**	**0.974**

Table 10
Illustrates the comparison of the proposed DOLNet's parameters with the state-of-the-art model's parameters.

Model	Parameters
YOLOv3+A+PG [54]	65.2M
TableRadar [18]	52.3M
Li et al. [7]	52.3M
YOLO [11]	65.2M
M-RCNN [11]	64.0M
M-RCNN [16]	64.0M
Mask-RCNN [23]	64.0M
RetinaNet [11]	71.13M
DOLNet	144.0M

more parameters than the existing models and obtains the best detection results.

5.4. Ablation study

We perform a series of experiments to check the effectiveness of the proposed method. We train five models on the Marmot dataset and evaluate on the ICDAR-2013. Our two baseline models — cascade R-CNN obtains an F1 score of 0.960, and cascade Mask R-CNN achieves an F1 score of 0.981 at IoU threshold 0.5. The F1 score highlights that cascade Mask R-CNN is better than cascade R-CNN for table detection. We incorporate the deformable convolution in the baseline model cascade Mask R-CNN and obtain an F1 score of 0.990. Adding the deformable convolution in the base model improves table detection accuracy by 0.9%. We incorporate the dual backbone on the baseline model cascade Mask R-CNN and obtain an F1 score of 0.984. Incorporating the dual backbone also improves the performance over baseline performance. Again we include deformable convolution instead of convolution in the dual backbone and call it DOLNet, which attains the best F1 score 1. 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 selected DOLNet as our final model for the table detection task.

5.5. Discussion on visual results

We present sample correctly detected visual results in Fig. 4 and sample false/wrongly detected visual results in Fig. 5 using DOLNet. In the case of ICDAR-2013 and ICDAR-2017, a single page contains multiple tables, and tables are very close to each other. DOLNet detects all tables accurately (see Fig. 4). While ICDAR-2019 contains historical handwritten tables, DOLNet also accurately detects handwritten tables. PubLayNet, TableBank, UNLV, and Marmot contain multiple tables with diversity in style and size. In such cases, DOLNet accurately detects all tables on a single page. textcolorblueFrom Fig. 5, we observe that DOLNet detects larger boundaries than actual boundary for a few tables in ICDAR-2013. Table headings and captions are detected as part of the table. It is because of only consideration of visual information, not textual information in DOLNet. In the case of ICDAR-2017, a few documents contain several figures which visually closely look like a table. Since DOLNet considers only visual features and predicts a few false positives corresponding to those figures. ICDAR-2019 contains both archival and modern documents. In the case of modern documents, multiple tables are located very close to each other. In such documents, DOLNet detects multiple tables within a single bounding box. The archive documents contain double-column formatted documents containing handwritten tables. The DOLNet detects handwritten text as a part of the table since tables are handwritten. At the same time, PubLayNet, Marmot, TableBank, UNLV datasets contain a few long tables with multiple sub-tables. Instead of detecting a complete table, the DOLNet detects only some parts of longer tables or/and all sub-tables separately. It is because of using only visual cues in DOLNet.

6. Performance across multiple IoU thresholds

Effect of IoU Threshold on Document Object Detection:
We also perform another set of experiments to check the robustness of DOLNet for detecting various document objects while varying IoU thresholds. Table 12 shows the quantitative results of DOLNet for document object detection on PubLayNet under various IoU thresholds. For all categories of document objects excepting Title, DOLNet obtains consistent results (F1 score and mAP) while varying the IoU threshold from 0.5 to 0.9. Since Title is much smaller than other categories, DOLNet obtains the higher performance (0.969 F1 score and 0.949 mAP) at IoU threshold 0.5 and reduces around 50% performance (0.255 F1 score and 0.067 mAP) at IoU threshold 0.9. This set of experiments also highlights the robustness of DOLNet for detecting document objects on varying IoU thresholds.

Effect of IoU Threshold on Table Detection:
We check the robustness of DOLNet under various IoU thresholds for detecting tables on various benchmark datasets. Tables 13 and 14 show the performance of DOLNet for detecting tables on ICDAR-2013, ICDAR-POD-2017, ICDAR-2019, UNLV PubLayNet and TableBank datasets, respectively, under multiple IoU thresholds. CDeC-Net is robust while varying IoU thresholds from 0.5 to 0.8 for detecting tables. Only ICDAR-2013, UNLV and Marmot datasets, the performance (F1 score) of DOLNet reduces less than 30.0% while varying IoU threshold from 0.8 to 0.9. All other datasets, DOLNet obtains consistent results while varying IoU threshold. These ex-

Table 11
Illustrates the performances of various models. All models are tested on ICDAR-2013 data set with 0.5 as IoU threshold. **BB:** indicates backbone, **DBBB:** indicates dual or composite backbone, and **DC:** indicates deformable convolution. Cascade Mask R-CNN with composite ResNeXt-101 having deformable convolution as backbone i.e., DOLNet obtains best results as compared to other models. We select DOLNet as our final model.

Models	Score			
	R↑	**P↑**	**F1↑**	**mAP↑**
Cascade R-CNN + BB(ResNeXt-101)	0.963	0.958	0.960	0.953
Cascade Mask R-CNN + BB (ResNeXt-101)	0.987	0.975	0.981	0.975
Cascade Mask R-CNN + BB (ResNeXt-101 + DC)	0.991	0.990	0.990	0.986
Cascade Mask R-CNN + DBB (ResNeXt-101)	0.987	0.981	0.984	0.973
Cascade Mask R-CNN + DBB (ResNeXt-101 + DC) (i.e., DOLNet)	**1.000**	**1.000**	**1.000**	**0.995**

Table 12
Illustrates the performance of DOLNet for detecting document objects under varying IoU thresholds on PubLayNet.

PubLayNet		IoU Threshold				
		0.5	0.6	0.7	0.8	0.9
Text	**R↑**	0.887	0.887	0.885	0.872	0.813
	P↑	0.988	0.988	0.986	0.971	0.906
	F1↑	0.938	0.937	0.935	0.921	0.860
	mAP↑	0.887	0.886	0.884	0.868	0.797
Title	**R↑**	0.951	0.946	0.929	0.764	0.250
	P↑	0.986	0.982	0.964	0.793	0.259
	F1↑	0.969	0.964	0.947	0.779	0.255
	mAP↑	0.949	0.944	0.923	0.673	0.067
List	**R↑**	0.962	0.958	0.955	0.950	0.922
	P↑	0.963	0.959	0.957	0.951	0.924
	F1↑	0.963	0.958	0.956	0.951	0.923
	mAP↑	0.959	0.955	0.952	0.946	0.913
Table	**R↑**	0.989	0.989	0.987	0.985	0.975
	P↑	0.992	0.992	0.991	0.988	0.978
	F1↑	0.991	0.990	0.989	0.987	0.977
	mAP↑	0.989	0.988	0.986	0.983	0.970
Figure	**R↑**	0.996	0.995	0.992	0.985	0.966
	P↑	0.995	0.993	0.990	0.983	0.965
	F1↑	0.995	0.994	0.991	0.984	0.965
	mAP↑	0.996	0.995	0.990	0.982	0.956

Table 13
Illustrates the performance of DOLNet under varying IoU thresholds for detecting tables on ICDAR-2013, ICDAR-POD-2017, ICDAR-2019 and UNLV datasets. **Th:** Threshold.

IoU	Performance on Various Benchmark Datasets															
	ICDAR-2013				ICDAR-POD-2017				ICDAR-2019				UNLV			
Th	**R↑**	**P↑**	**F1↑**	**mAP↑**	**R↑**	**P↑**	**F1↑**	**mAP↑**	**R↑**	**P↑**	**F1↑**	**mAP↑**	**R↑**	**P↑**	**F1↑**	**mAP↑**
0.5	1.000	1.000	1.000	1.000	0.934	0.990	0.962	0.931	0.946	0.987	0.966	0.939	0.770	0.960	0.865	0.742
0.6	1.000	1.000	1.000	1.000	0.931	0.987	0.959	0.927	0.939	0.980	0.959	0.929	0.758	0.944	0.851	0.717
0.7	0.987	0.987	0.987	0.981	0.931	0.987	0.959	0.927	0.936	0.977	0.956	0.926	0.734	0.915	0.825	0.674
0.8	0.942	0.942	0.942	0.899	0.928	0.983	0.955	0.924	0.930	0.971	0.950	0.913	0.663	0.826	0.744	0.551
0.9	0.660	0.660	0.660	0.459	0.902	0.957	0.929	0.883	0.895	0.934	0.915	0.841	0.496	0.618	0.557	0.314

Table 14
Illustrates the performance of DOLNet under varying IOU thresholds for detecting tables on Marmot, PubLayNet, TableBank, and IIIT-AR-13K datasets.

IoU	Performance on Various Benchmark Datasets															
	Marmot				PubLayNet				TableBank				IIIT-AR-13K			
Th	**R↑**	**P↑**	**F1↑**	**mAP↑**	**R↑**	**P↑**	**F1↑**	**mAP↑**	**R↑**	**P↑**	**F1↑**	**mAP↑**	**R↑**	**P↑**	**F1↑**	**mAP↑**
0.5	0.916	0.991	0.953	0.909	0.977	0.996	0.986	0.977	0.979	0.995	0.987	0.976	0.982	0.976	0.979	0.981
0.6	0.911	0.985	0.948	0.899	0.977	0.995	0.986	0.976	0.978	0.995	0.986	0.974	0.979	0.973	0.976	0.977
0.7	0.905	0.979	0.942	0.891	0.976	0.994	0.985	0.974	0.978	0.995	0.986	0.9744	0.973	0.967	0.970	0.970
0.8	0.887	0.960	0.924	0.859	0.974	0.992	0.983	0.972	0.977	0.993	0.985	0.970	0.959	0.954	0.956	0.955
0.9	0.823	0.891	0.857	0.742	0.965	0.983	0.974	0.959	0.966	0.982	0.974	0.950	0.892	0.887	0.890	0.871

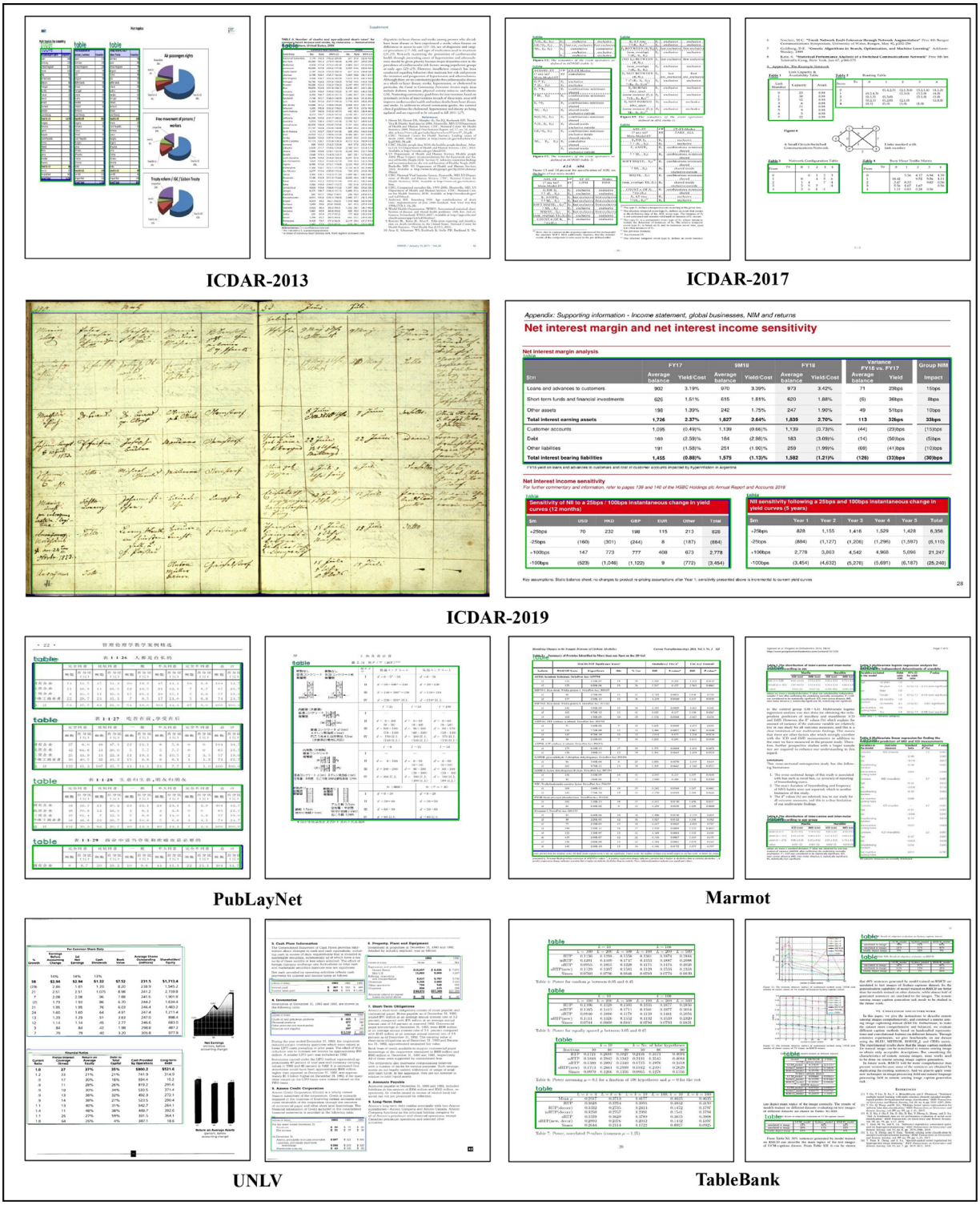

Fig. 4. Shows sample visual detection results of publicly available benchmark datasets — ICDAR-2013, ICDAR-POD-2017, ICDAR-2019, UNLV, Marmot, PubLayNet and TableBank. Blue colored rectangles indicate the ground truth bounding boxes. Green colored rectangles indicate the predicted bounding boxes using DOLNet.

periments highlight that the performance of DOLNet is robust against the IoU threshold.

7. Single trained model working across datasets

Table 15 presents the comparative results between the state-of-the-art techniques and the created unique model DOLNet† on various benchmark datasets. The unique model DOLNet† is created by training with the IIIT-AR-13K dataset. Table 15 highlights that our proposed model DOLNet obtains F1 score 1.0 while trained with D1 (Marmot+UNLV+ICDAR-POD-2017) dataset. On the other hand, our single model DOLNet† trained on IIIT-AR-13K with single object category: Table attains very close results (F1 score 0.968) to the state-of-the-art model DOLNet on ICDAR-2013.

In case of ICDAR-2017, the table also presents that YOLOV3+A+PG obtains state-of-the-art results: F1 scores 0.975 and 0.971 for

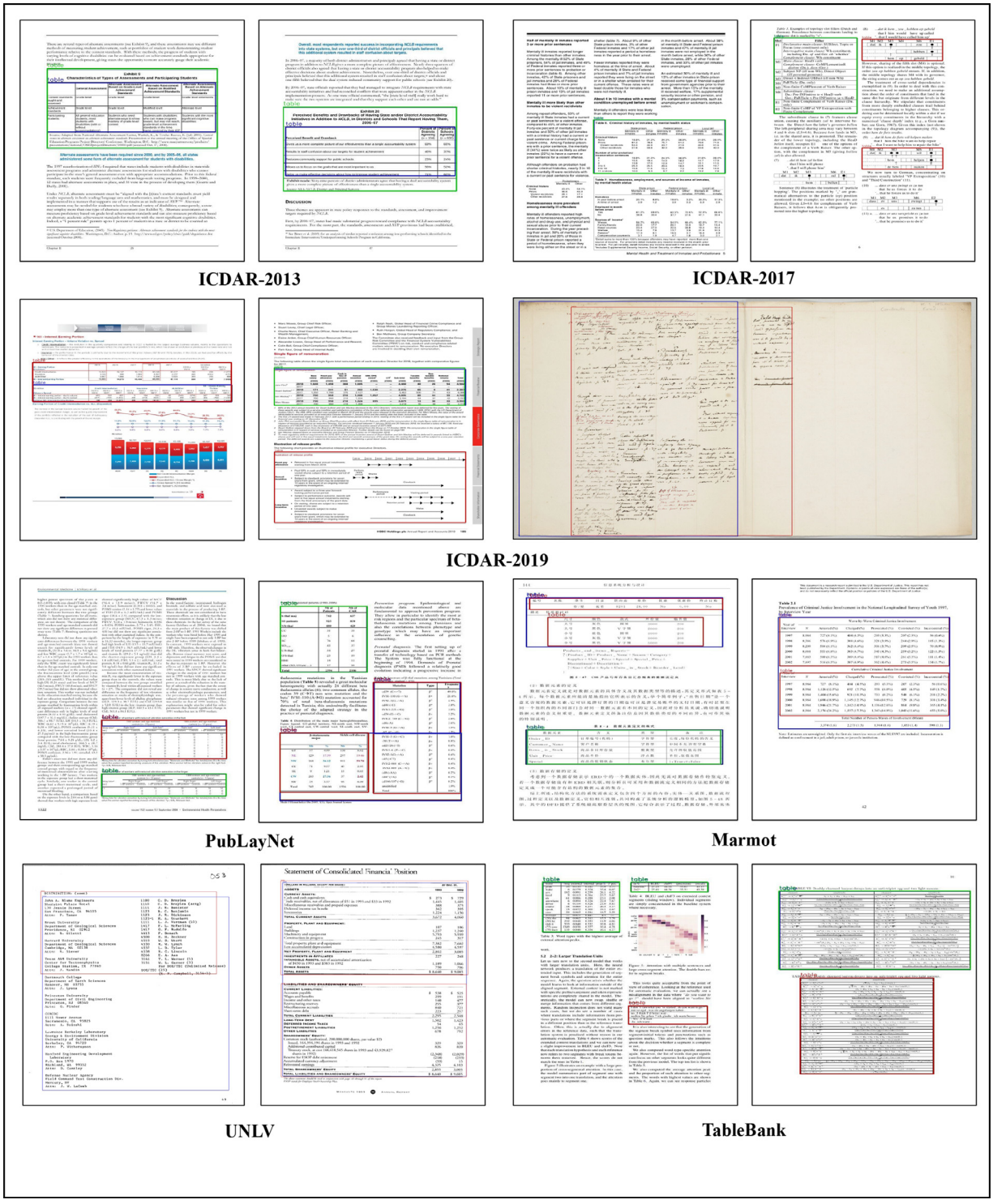

Fig. 5. Shows sample visual erroneous detection results of publicly available benchmark datasets — ICDAR-2013, ICDAR-POD-2017, ICDAR-2019, UNLV, Marmot, PUBLAYNET and TABLEBANK. Blue colored rectangles indicate the ground truth bounding boxes. Green colored rectangles indicate the predicted bounding boxes using DOLNET. Red colored rectangles indicate the false detection results obtained using DOLNET.

IoU thresholds 0.6 and 0.8, respectively. While our unique model DOLNET† trained on IIIT-AR-13K with multiple object categories, fine tuned on ICDAR-POD-2017 and evaluated on ICDAR-POD-2017 attains very close results (F1 scores 0.959 and 0.955) to the best model YOLOV3+A+PG (F1 scores 0.975 and 0.971) for IoU thresholds 0.6 and 0.8, respectively.

We also observe from the table that TABLERADAR obtains state-of-the-art performance (F1 score 0.945) for IoU threshold 0.8 and the proposed model DOLNET obtains state-of-the-art performance (F1 score 0.913) for IoU threshold 0.9 on a complete test set of ICDAR-2019. While our unique model DOLNET† trained on IIIT-AR-13K with single object category: Table, fine-tuned on ICDAR-2019 and evaluated on ICDAR-2019 obtains better results (F1 scores 0.950 and 0.915) than the best models (F1 scores 0.945 and 0.913) for IoU thresholds 0.8 and 0.9, respectively.

Table 15
Illustrates comparison between the state-of-the-art techniques and the created unique model DOLNet† on various benchmark datasets. The unique model DOLNet† is created by training with IIIT-AR-13K of single object category (table). **#:** no. of images, **A:** anchor optimization, **PG:** post-processing technique, **D1:** Marmot+UNLV+ICDAR-2017, **D4:** ICDAR-2013+ICDAR-2017+Marmot. Blue color indicates state-of-the-art results. Values in bold indicate the best results.

	Training		Fine-tuning		Test		IoU	Score			
Method	Dataset	#	Dataset	#	Dataset	#		R↑	P↑	F1↑	mAP↑
DOLNet	D1	4808	-	-	ICDAR-2013	238	0.5	**1.000**	**1.000**	**1.000**	**1.000**
DOLNet†	IIIT-AR-13K	9K	-	-	ICDAR-2013	238	0.5	0.942	0.993	0.968	0.942
YOLOv3+A+PG [54]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.6	**0.972**	**0.978**	**0.975**	-
YOLOv3+A+PG [54]	ICDAR-2017	1600	-	-	ICDAR-2017	817	0.8	**0.968**	**0.975**	**0.971**	-
DOLNet†	IIIT-AR-13K	9K	ICDAR-2017	1600	ICDAR-2017	817	0.6	0.931	0.987	0.959	0.927
DOLNet†	IIIT-AR-13K	9K	ICDAR-2017	1600	ICDAR-2017	817	0.8	0.928	0.983	0.955	0.924
TableRadar [18]	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.8	**0.940**	0.950	0.945	-
DOLNet	ICDAR-2019	1200	-	-	ICDAR-2019	439	0.9	**0.904**	0.922	0.913	**0.843**
DOLNet†	IIIT-AR-13K	9K	ICDAR-2019	1200	ICDAR-2019	439	0.8	0.930	**0.971**	**0.950**	0.913
DOLNet†	IIIT-AR-13K	9K	ICDAR-2019	1200	ICDAR-2019	439	0.9	0.895	**0.934**	**0.915**	0.841
DOLNet	D4	4622	-	-	UNLV	424	0.5	0.736	0.852	0.794	0.657
DOLNet†	IIIT-AR-13K	9K	D4	4622	UNLV	424	0.5	0.729	**0.894**	**0.812**	**0.674**
DOLNet	Various Doc	130			Marmot	2000	0.8	**0.836**	**0.845**	**0.840**	**0.716**
DOLNet	Various Doc	130			Marmot	2000	0.9	0.765	0.774	0.769	0.600
DOLNet†	IIIT-AR-13K	9K	Various Doc	130	Marmot	2000	0.8	0.833	0.837	0.835	0.710
DOLNet†	IIIT-AR-13K	9K	Various Doc	130	Marmot	2000	0.9	**0.772**	**0.775**	**0.773**	**0.603**
DOLNet	TableBank-L	199K	-	-	TableBank-L	1K	0.6	**0.978**	**0.995**	**0.986**	**0.974**
DOLNet†	IIIT-AR-13K	9K	TableBank-L	199K	TableBank-L	1K	0.6	0.970	0.990	0.980	0.965

The table also presents that the state-of-the-art model (DOLNet) obtains an F1 score 0.794 on complete UNLV for IoU threshold 0.5. Our unique model: DOLNet† trained on IIIT-AR-13K with single object category: Table, obtains better results (F1 score 0.812) than the state-of-the-art model (F1 score 0.794). Table highlights that the proposed model DOLNet obtains the best results (F1 scores 0.840 and 0.769) on the complete Marmot dataset for IoU thresholds 0.8 and 0.9, respectively. While our unique model DOLNet† trained on IIIT-AR-13K, obtains close result (F1 score 0.835) to the best model (F1 scores 0.840) at IoU threshold 0.8 and better result (F1 score 0.773) than the best model (F1 score 0.769) at IoU threshold 0.9.

The table also presents that the proposed model DOLNet obtains the best results (F1 score 0.986). While our unique model DOLNet† trained on IIIT-AR-13K with single object category: Table, attains close results (F1 score 0.980) to the best model (F1 score 0.986). With these experiments, we conclude that instead of various models, only our single model DOLNet† trained on IIIT-AR-13K, achieves very close (sometimes even better) performance to the best models for respective datasets. We also conclude that our unique model DOLNet† works across all datasets. It is because IIIT-AR-13K contains tables with diverse layouts, contents, and structures.

8. Conclusion

We introduce a DOLNet, consisting 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 DOLNet 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 DOLNet‡ 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 document objects — tables, figures, logos, and mathematical expressions. Though DOLNet achieves high performance, it fails to properly detect tables in documents having very closely located multiple tables, larger tables with several sub-tables, and figures that look like tables. In the future, we plan to employ visual and textual deep feature fusion (similar to [58]) for better model performance. Existing document object (e.g., table) detection methods rely heavily on many annotated documents and require a long training time. In computer vision, few-shot object detection [59,60] is recently very popular for handling such real challenges. In the future, few-shot leraning can be explored to detect document objects to address real challenges.

Declaration of Competing Interest

We have declared that we don't have any financial and personal relationships with other people or organisations that could inappropriately influence (bias) our work.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work is supported by MeitY, Government of India.

References

[1] K. Itonori, Table Structure Recognition Based on Textblock Arrangement and Ruled Line Position, in: ICDAR, 1993, pp. 765–768.
[2] S. Tupaj, Z. Shi, C.H. Chang, H. Alam, Extracting Tabular Information from Text Files, EECS Department, Tufts University, Medford, USA 1 (1996).
[3] A. Gilani, S.R. Qasim, I. Malik, F. Shafait, Table detection using deep learning, in: ICDAR, volume 1, 2017, pp. 771–776.
[4] S. Schreiber, S. Agne, I. Wolf, A. Dengel, S. Ahmed, DeepDeSRT: deep learning for detection and structure recognition of tables in document images, in: ICDAR, volume 1, 2017, pp. 1162–1167.
[5] S.A. Siddiqui, M.I. Malik, S. Agne, A. Dengel, S. Ahmed, DeCNT: deep deformable CNN for table detection, IEEE Access 6 (2018) 74151–74161.
[6] S. Arif, F. Shafait, Table detection in document images using foreground and background features, in: DICTA, 2018, pp. 1–8.
[7] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Z. Li, TableBank: table benchmark for image-based table detection and recognition, in: LREC, 2020, pp. 1918–1925.
[8] N. Sun, Y. Zhu, X. Hu, Faster R-CNN based table detection combining corner locating, in: ICDAR, 2019, pp. 1314–1319.
[9] Y. Liu, Y. Jin, C. Huang, W. Bao, Table detection method based on feature pyramid network with faster R-CNN, in: ICDIP, volume 11519, 2020, pp. 73–80.
[10] R. Saha, A. Mondal, C.V. Jawahar, Graphical Object Detection in Document Images, in: ICDAR, 2019, pp. 51–58.
[11] A. Casado-García, C. Domínguez, J. Heras, E. Mata, V. Pascual, The benefits of close-domain fine-tuning for table detection in document images, in: DAS, 2020, pp. 199–215.
[12] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, K. Sultanpure, CascadeTabNet: an approach for end to end table detection and structure recognition from image-based documents, in: CVPRW, 2020, pp. 572–573.

[13] M. Agarwal, A. Mondal, C.V. Jawahar, CDeC-Net: composite deformable cascade network for table detection in document images, in: ICPR, 2020, pp. 9491–9498.
[14] J. Younas, S.T.R. Rizvi, M.I. Malik, F. Shafait, P. Lukowicz, S. Ahmed, FFD: figure and formula detection from document images, in: DICTA, 2019, pp. 1–7.
[15] L. Gao, X. Yi, Z. Jiang, L. Hao, Z. Tang, ICDAR 2017 competition on page object detection, in: ICDAR, volume 1, 2017, pp. 1417–1422.
[16] X. Zhong, J. Tang, A.J. Yepes, PubLayNet: largest dataset ever for document layout analysis, in: ICDAR, 2019, pp. 1015–1022.
[17] M. Göbel, T. Hassan, E. Oro, G. Orsi, ICDAR 2013 table competition, in: ICDAR, 2013, pp. 1449–1453.
[18] L. Gao, Y. Huang, H. Déjean, J.-L. Meunier, Q. Yan, Y. Fang, F. Kleber, E. Lang, ICDAR 2019 competition on table detection and recognition (cTDaR), in: ICDAR, 2019, pp. 1510–1515.
[19] A. Shahab, F. Shafait, T. Kieninger, A. Dengel, An open approach towards the benchmarking of table structure recognition systems, in: DAS, 2010, pp. 113–120.
[20] N. Siegel, N. Lourie, R. Power, W. Ammar, Extracting scientific figures with distantly supervised neural networks, in: ACM/IEEE on joint conference on digital libraries, 2018, pp. 223–232.
[21] J. Fang, X. Tao, Z. Tang, R. Qiu, Y. Liu, Dataset, ground-truth and performance metrics for table detection evaluation, in: DAS, 2012, pp. 445–449.
[22] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, M. Zhou, Docbank: a benchmark dataset for document layout analysis, arXiv (2020).
[23] A. Mondal, P. Lipps, C.V. Jawahar, IIIT-AR-13K: a new dataset for graphical object detection in documents, in: DAS, 2020, pp. 216–230.
[24] Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, H. Ling, CBNNet: a novel composite backbone network architecture for object detection, in: AAAI, volume 34, 2020, pp. 11653–11660.
[25] N.D. Vo, K. Nguyen, T.V. Nguyen, K. Nguyen, Ensemble of deep object detectors for page object detection, in: UIMC, 2018, pp. 1–6.
[26] Z. Cai, N. Vasconcelos, Cascade R-CNN: high quality object detection and instance segmentation, IEEE Trans. PAMI 43 (2019) 1483–1498.
[27] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: ICCV, 2017, pp. 764–773.
[28] S. Chandran, S. Balasubramanian, T. Gandhi, A. Prasad, R. Kasturi, A. Chhabra, Structure recognition and information extraction from tabular documents, Int. J. Imaging Syst. Technol. 7 (1996) 289–303.
[29] B. Gatos, D. Danatsas, I. Pratikakis, S.J. Perantonis, Automatic table detection in document images, in: ICAPR, 2005, pp. 609–618.
[30] F. Shafait, R. Smith, Table detection in heterogeneous documents, in: DAS, 2010, pp. 65–72.
[31] R. Smith, An overview of the tesseract ocr engine, in: ICDAR, volume 2, 2007, pp. 629–633.
[32] S. Mandal, S. Chowdhury, A.K. Das, B. Chanda, A simple and effective table detection system from document images, IJDAR 8 (2006) 172–182.
[33] X.w. Zhang, M.R. Lyu, G.z. Dai, Extraction and segmentation of tables from chinese ink documents based on a matrix model, Pattern Recognit. 40 (2007) 1855–1867.
[34] G.V.S.S.K.R. Naganjaneyulu, N.V. Sathwik, A. Narasimhadhan, A multi clue heuristic based algorithm for table detection, in: TENCON, 2016, pp. 1246–1249.
[35] T. Kieninger, A. Dengel, The T-RECS table recognition and analysis system, in: DAS, 1998, pp. 255–270.
[36] A.C.e. Silva, Learning rich hidden markov models in document analysis: table location, in: ICDAR, 2009, pp. 843–847.
[37] T. Kasar, P. Barlas, S. Adam, C. Chatelain, T. Paquet, Learning to detect tables in scanned document images using line information, in: ICDAR, 2013, pp. 1185–1189.
[38] M. Fan, D.S. Kim, Table region detection on large-scale PDF files without labeled data, CoRR (2015).
[39] P. Forczmański, A. Markiewicz, Two-stage approach to extracting visual objects from paper documents, Mach. Vis. Appl. 27 (2016) 1243–1257.
[40] D.A.B. Oliveira, M.P. Viana, Fast CNN-based document layout analysis, in: ICCVW, 2017, pp. 1173–1180.
[41] R. Girshick, Fast R-CNN, in: ICCV, 2015, pp. 1440–1448.
[42] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: NIPS, volume 28, 2015, pp. 1–9.
[43] S. Luo, M. Wu, Y. Gong, W. Zhou, J. Poon, Deep structured feature networks for table detection and tabular data extraction from scanned financial document images, arXiv (2021).
[44] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: ICCV, 2017, pp. 2961–2969.
[45] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: CVPR, 2016, pp. 779–788.
[46] P. Riba, L. Goldmann, O.R. Terrades, D. Rusticus, A. Fornés, J. Lladós, Table detection in business document images by message passing networks, Pattern Recognit. 127 (2022) 108641–108653.
[47] S.S. Paliwal, D. Vishwanath, R. Rahul, M. Sharma, L. Vig, TableNet: deep learning model for end-to-end table detection and tabular data extraction from scanned document images, in: ICDAR, 2019, pp. 128–133.
[48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: ICCV, 2017, pp. 2980–2988.
[49] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: ECCV, 2016, pp. 21–37.
[50] K. Li, C. Wigington, C. Tensmeyer, H. Zhao, N. Barmpalios, V.I. Morariu, V. Manjunatha, T. Sun, Y. Fu, Cross-domain document object detection: benchmark suite and method, in: CVPR, 2020, pp. 12915–12924.
[51] C. Ma, W. Lin, L. Sun, Q. Huo, Robust table detection and structure recognition from heterogeneous document images, Pattern Recognit. 133 (2023) 109006–109023.
[52] 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, D. Lin, MMDEtection: open MMLab detection toolbox and benchmark, arXiv (2019).
[53] D. Tkaczyk, P. Szostek, L. Bolikowski, GROTOAP2 - The methodology of creating a large ground truth dataset of scientific articles, D Lib. Mag. 20 (2014).
[54] Y. Huang, Q. Yan, Y. Li, Y. Chen, X. Wang, L. Gao, Z. Tang, A YOLO-based table detection method, in: ICDAR, 2019, pp. 813–818.
[55] S.A. Khan, S.M.D. Khalid, M.A. Shahzad, F. Shafait, Table structure extraction with bi-directional gated recurrent unit networks, in: ICDAR, 2019, pp. 1366–1371.
[56] Y. Li, L. Gao, Z. Tang, Q. Yan, Y. Huang, A GAN-based feature generator for table detection, in: ICDAR, 2019, pp. 763–768.
[57] D. He, S. Cohen, B. Price, D. Kifer, C.L. Giles, Multi-scale Multi-task FCN for semantic page segmentation and table detection, in: ICDAR, volume 1, 2017, pp. 254–261.
[58] S. Bakkali, Z. Ming, M. Coustaty, M. Rusiñol, Visual and textual deep feature fusion for document image classification, in: CVPRW, 2020, pp. 562–563.
[59] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, T. Darrell, Few-shot object detection via feature reweighting, in: CVPR, 2019, pp. 8420–8429.
[60] Q. Fan, W. Zhuo, C.-K. Tang, Y.W. Tai, Few-shot object detection with attention-rpn and multi-relation detector, in: CVPR, 2020, pp. 4013–4022.

Ajoy Mondal received the Ph.D. degree in computer science from Jadavpur University, Kolkata, India in 2018. Currently he is a Postdoctoral Fellow at CVIT, International Institute of Information Technology, Hyderabad, India. His research interests include document image processing, machine learning, computer vision, pattern recognition and image processing.

Madhav Agarwal is currently a MS student at CVIT in International Institute of Information Technology, Hyderabad, India. His research interests include document image processing, computer vision, and machine learning.

C. V. Jawahar is a Professor at International Institute of Information Technology, Hyderabad, India. He published around 400 articles in reputed journals and conferences. He is an associate editor of IEEE Tran. PAMI. He has acted as area chair and program chair, reviewer, PC member of many conferences. His areas of research include computer vision, machine learning and document image analysis.