

Graphical Object Detection in Document Images

Ranajit Saha, Ajoy Mondal and C V Jawahar

Centre for Visual Information Technology,

International Institute of Information Technology, Hyderabad, India,

Email: ranajit.saha@students.iiit.ac.in, ajoy.mondal@iiit.ac.in and jawahar@iiit.ac.in

Abstract—Graphical elements: particularly tables and figures contain a visual summary of the most valuable information contained in a document. Therefore, localization of such graphical objects in the document images is the initial step to understand the content of such graphical objects or document images. In this paper, we present a novel end-to-end trainable deep learning based framework to localize graphical objects in the document images called as Graphical Object Detection (GOD). Our framework is data-driven and does not require any heuristics or meta-data to locate graphical objects in the document images. The GOD explores the concept of transfer learning and domain adaptation to handle scarcity of labeled training images for graphical object detection task in the document images. Performance analysis carried out on the various public benchmark data sets: ICDAR-2013, ICDAR-POD 2017 and UNLV shows that our model yields promising results as compared to state-of-the-art techniques.

Keywords-Graphical object localization; deep neural network; transfer learning; data-driven.

I. INTRODUCTION

With the rapidly increasing number of digital documents, the manual extraction and retrieval of information from them has become an in-feasible option - Automated methods are coming to the rescue. There are many tools and methods available to convert the digital documents into process-able texts. The graphical page elements, such as tables, figures, equations play a very important role in understanding and extracting the information from the documents. Therefore, the detection of those objects from the documents has attracted a lot of attention in the research community. Because of no common dimensions and diversity of layouts of the tables and figures in the documents, detection of them is considered to be a challenging problem.

In this paper, our objective is to detect graphical objects like tables, figures, equations from varied types of digitally generated documents. This problem is conceptually similar to object detection in natural scene images. The diverse layouts and structure of equations, tables and figures make the detection difficult for the rule based systems [1]–[4]. Tables have different structures such as cell separated table, table with no separation lines between the rows and columns or alternating colours to separate the cells. All these make the table detection itself very challenging. Sometimes, tables and figures have a high degree of inter-class similarity between themselves and also among other graphical objects,

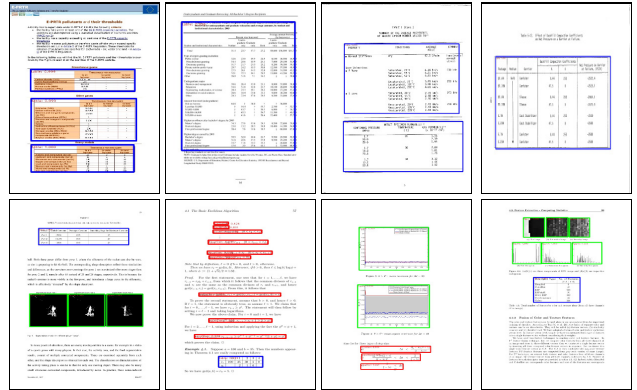


Figure 1: Visual illustration of various graphical object detection results of the public benchmark data sets by the GOD framework. **First Row:** detection of various forms of tables in various documents. **Second Row:** detection of multiple graphical objects: tables, equations and figures with various forms and multiple numbers in a single page. Blue, Green and Red colors represent the bounding boxes of table, figure and equation, respectively.

e.g. some charts and plots with several intersections of horizontal and vertical lines will resemble the structure of the tables. Traditional rule based methods have had difficulties to detect them with high precision [5]–[8].

In this paper, we present a novel end-to-end trainable deep learning based framework, called as Graphical Object Detection (GOD), for detecting graphical objects particularly tables, figures and equations in the document images. Our presented framework based on the recent object detection algorithms in computer vision [9]–[11] is data-driven and independent of any heuristic rules for detecting graphical objects in the document images. Usually, deep learning based approaches require a large amount of labelled training data which is not available in our task. To solve the scarcity of labelled training data, the GOD explores the concept of transfer learning and domain adaptation for graphical object detection task in the document images. Experiments on the various public benchmark data sets conclude that the GOD is superior than state-of-the-art techniques for localizing various graphical objects in the document images.

Figure 1 visually illustrates the success of GOD technique for localizing various graphical objects present in the document images. In particular, the contributions of this work are as follows:

- We present an end-to-end trainable deep learning based approach to localize graphical object in the document images inspired by the concept of recent object detection algorithms in computer vision [9], [11].
- We perform transfer learning to fine-tune a pre-trained model for our graphical object detection task in the document images.
- Our GOD framework obtains the superior results on public benchmark **ICDAR-POD2017** (please see Table III), **ICDAR-2013** (please see Table IV) and **UNLV** (please see Table V) than state-of-the-arts.

II. RELATED WORK

Automatic information extraction from the digital documents requires the detection and understanding of graphical objects such as tables, figures, equations, etc. A good number of researchers have contributed to localize various page objects and layout analysis on the different types of documents.

A. Rule based Methods

Before deep learning era, most of the developed approaches [1]–[4] for table detection were based on the assumption of table structures and exploiting a prior knowledge on object properties by analyzing tokens extracted from the documents. Tupaj *et al.* [2] developed an optical character recognition (OCR) based table recognition system which initially identifies the potential table region through the combination of white space and keyword analysis. Keyword analysis determines the header which is regarded as the starting line of the table. The vertical and horizontal passes through the document page identify the white spaces between the cells of the table region. This rule based method is highly dependent on the set of keywords used for analyzing the headers and the inherent assumption about the structure of the tables. Fang *et al.* [3] analyzed the page layout and detected the table from PDF documents based on separator mining which identifies the visual separators of the cells. This method assumes that the table cells are separated either by ruling lines or by white spaces. The disadvantage of such rule based methods is that they heavily rely upon the type and structure of tables under consideration, hence fail to detect the tables whose layout structures are different.

B. Deep Learning based Methods

Deep Convolutional Neural Networks (DCNNs) have proved to be useful in highly complex computer vision tasks to identify visually distinguished features of images. The effectiveness of DCNNs are also being explored in recent years in document object analysis by various research

groups [5]–[8]. Hao *et al.* [5] applied deep learning for detection of tables in PDF documents. They considered heuristic rules to propose regions with table-like structures in the document and then classified them into table or non-table regions using a CNN. However, the shortcomings of using rule to identify table-like structures are not yet overcome completely by this method. Schreiber *et al.* [6] proposed a deep learning based solution for table detection and structure identification, which does not require any assumption about the structure of the tables. They fine-tuned Faster R-CNN [9] with two different backbone architectures: ZFNet [12] and VGG-16 [13] for the table detection task. Gilani *et al.* [7] also used the same approach. However, they applied image transformation by stacking three different distance transformed layers before passing them through Faster R-CNN. This model can deal with only identifying table regions in the documents.

Several algorithms [14]–[18] have been developed to identify text and non-text or graphical regions from document images. As per [19], the classification techniques can be broadly classified into three approaches (i) region or block based, (ii) pixel based and (iii) connected component based classification methods. However, none of these methods classify the graphical regions in the document images, which is a complex task as the structures of the graphical regions do not follow any general rule. Kavasidis *et al.* [8] proposed a saliency based CNN architecture for document layout analysis which used semantic segmentation to classify each pixel into different classes. This model at first extracted saliency features corresponding to the text, table and figure regions. The extracted segmentation maps by the saliency detector are passed through binary classifiers to classify them. This model used a binary classifier for each of the classes, which are trained separately. This model is able to detect four different kinds of page objects: table, bar chart, pie chart and line chart. Xiaohan Yi *et al.* [20] proposed a dynamic programming based region proposal method for page object detection.

III. GRAPHICAL OBJECT DETECTION

This section presents the details about the proposed GOD algorithm inspired by the recent object detection algorithms in computer vision [9], [11]. Detection of graphical objects like tables, figures, equations, etc. is basically localization of these objects within a document image. The problem is conceptually similar to the detection of objects in natural scene images. Natural scene objects have visually identifiable characteristics, those features can be extracted by using CNN. Similarly graphical objects can also be identified by the specific characteristics of them, which CNN is capable of. The compelling performances of Faster R-CNN and Mask R-CNN on PASCAL VOC [21] and COCO [22] data sets makes us adapting the framework into our graphical object detection work. Training of deep networks requires a large amount

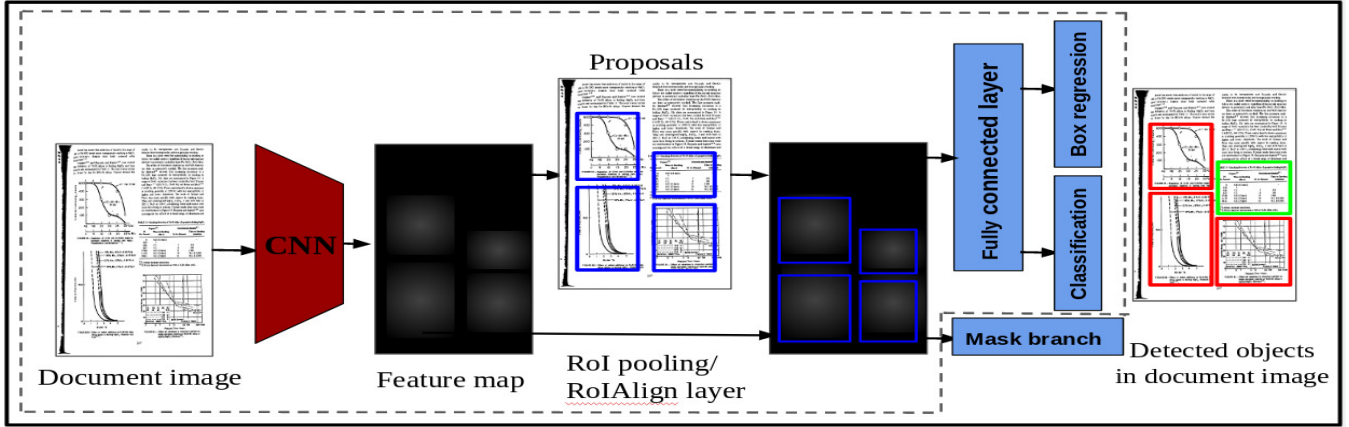


Figure 2: GOD framework. The model takes an image as input and generates the feature map. RPN proposes the region based on the feature map. Detection network uses the proposed regions and feature map to detect various graphical objects.

of data, which we lack in the document object detection domain. Hence, we consider domain adaptation and transfer learning in our work. We test the abilities of Faster R-CNN and Mask R-CNN, originally built for the natural scene images, to cope with detecting graphical objects in the document images. The performances of ImageNet [23] pre-trained models as the backbone of GOD model inspired by faster R-CNN is listed in Table I.

Figure 2 displays the overview of the GOD method. Dotted region in Figure 2 is the Faster R-CNN block. As described by [9], Faster-RCNN is comprised of two modules. The first module is Region Proposal Network (RPN), responsible for proposing regions which might contain objects. The second module, Fast-RCNN detector [24], detects the objects using the proposed regions. RPN attracts the attention of the Fast-RCNN as to look into the proposed regions. The whole system works a single, unified network for the object detection task. Similar to Faster R-CNN, Mask R-CNN [11] adopts the same two stage procedure (i) Region Proposal Network (RPN) for proposing regions and (ii) prediction of the class label and bounding box regression along with binary mask of each region of interests.

Region Proposal Network (RPN) proposes rectangular regions, each associated with objectness score [9] which tells the detector whether the region contains an object or the background. This module is capable of handling input images of any size. RPN module is also *translation-invariant*. Translation invariant means if an object in the image is translated, the proposal should also be translated and the same RPN module should predict the region proposal in either of the location.

The input image is first passed through a convolutional layer which generates a feature map. The Region of Interest (RoI) pooling layer in case of Faster R-CNN performs max pooling to convert the nonuniform size inputs to fixed-size feature maps. The output feature vector is then passed

through fully connected layers: box-regression (*reg*) and classification (*cls*) layer for predicting object bounding box and class label. While in case of Mask R-CNN, RoIAlign layer generates fixed sized feature by preserving the exact spatial locations. Finally, the fixed sized features pass to two different modules: multiple layer perceptron to predict object bounding box and class label and mask module to predict segmentation mask.

A. Implementation Details

The model is trained and tested on various data sets with fixed image sizes, but the dimensions of the images from a particular data set vary in a range. The images are then resized to a fixed size of 600×600 before passing through the Faster-RCNN. The model is implemented using PyTorch and trained and tested on Nvidia GeForce GTX 1080 Ti GPUs with batch size of 4. We used Caffe pre-trained models of VGG-16 [13] and ResNets [25], trained on ImageNet [23], as the backbone of the Faster R-CNN. For VGG-16, the last max pooling layer is not used in the model.

Since, the effective receptive field of the input image is large, we use 3×3 sliding windows in the RPN. To generate $k=30$ anchor boxes, we considered 6 different anchor scales in powers of 2 from 8 to 512 and anchor ratios of 1 through 5 so that the region proposals can cover almost every part of the image irrespective of the image size. We used stochastic gradient descent SGD as an optimizer with initial learning rate = 0.001 and the learning rate decays after every 5 epochs and it is equal to 0.1 times of the previous value. For further implementation and architecture details, please refer to the source code at: <https://github.com/rnjtsh/graphical-object-detector>.

IV. EXPERIMENTS

A. Evaluation Measures

We use mean average precision (mAP) [21], average precision, recall and $F1$ measures [6], [8], [26] to evaluate the performance of our algorithm for detecting graphical objects in the document images.

B. Public Data Sets

ICDAR-POD2017 [27]: This data set consists of 2417 English document images selected from 1500 scientific papers of CiteSeer. It includes large variety in both layout and object styles: single-column, double-column, multi-column and various kinds of equations, tables and figures. This data set is divided into training set consisting of 1600 and test set consisting of 817 images.

ICDAR-2013 [28]: This data set contains 67 PDFs with 150 tables - 40 PDFs excerpt from the US Government and 27 PDFs from EU. We use converted images corresponding to PDFs for our experiment. We use this data set only for testing purpose due to limited to number of images.

Marmot table recognition data set¹: It consists of 2000 document images with a great variety in page layout and table styles. We use this data set only for training our network.

UNLV data set [29]: It contains 2889 pages of scanned document images from variety of sources (Magazines, Newspapers, Business letters, Annual reports, etc). Among them, only 427 images contain table zone. We consider 427 images for evaluation purpose.

C. Ablation Study

We conduct a number of ablation experiments in the context of document object detection to quantify the importance of each of the components of our algorithm and to justify various design choices. Our GOD (Faster R-CNN) uses **ICDAR-POD2017** data set for this purpose.

Pre-trained Model: Researchers have already proven that deeper neural networks are beneficial for large scale image classification. To further analyze the GOD (Faster R-CNN), we conduct an experiment with different depths of pre-trained models. The details of detection scores of the GOD (Faster R-CNN) with different pre-trained models with same settings are listed in Table I. The table quantitatively shows that with the same setting, the deeper network obtains better detection accuracy.

IoU Threshold: We conduct a study on the detection performance of the GOD (Faster R-CNN) with varying IoU threshold. Table II shows the detection statistics of the GOD (Faster R-CNN) on varying IoU threshold. With the same setting, increasing threshold value can reduce the detection accuracy (i.e. values of mAP and Ave F1). This is because of reduction in the number of true positive when increasing IoU

| Models | Test Performance: AP | | | mAP |
|------------|----------------------|--------------|--------------|--------------|
| | Equation | Table | Figure | |
| VGG-16 | 0.807 | 0.934 | 0.857 | 0.866 |
| ResNet-50 | 0.817 | 0.950 | 0.819 | 0.862 |
| ResNet-101 | 0.894 | 0.959 | 0.857 | 0.899 |
| ResNet-152 | 0.903 | 0.962 | 0.851 | 0.905 |

Table I: Performances of different backbone models on **ICDAR-POD2017** with IoU threshold 0.6. Deeper pre-trained model obtains higher detection accuracy. **AP**: average precision. Bold value indicates the best result.

threshold. From the Table II, we observed that we obtained the best mAP and Ave F1 when the IoU threshold is set to 0.5.

| IoU | Test Performance: AP | | | mAP | Test Performance: F1 | | | Ave F1 |
|-----|----------------------|-------|--------|-------|----------------------|-------|--------|--------|
| | Eqn. | Table | Figure | | Eqn. | Table | Figure | |
| 0.5 | 0.922 | 0.980 | 0.881 | 0.928 | 0.922 | 0.977 | 0.851 | 0.917 |
| 0.6 | 0.903 | 0.962 | 0.851 | 0.905 | 0.889 | 0.959 | 0.878 | 0.909 |
| 0.7 | 0.810 | 0.937 | 0.815 | 0.854 | 0.832 | 0.952 | 0.833 | 0.872 |
| 0.8 | 0.742 | 0.916 | 0.786 | 0.814 | 0.772 | 0.927 | 0.833 | 0.844 |

Table II: Greater IoU threshold reduces the performance of the GOD (Faster R-CNN) model on **ICDAR-POD2017** due to reduction in number of true positives.

D. Graphical Object Detection in Document Images

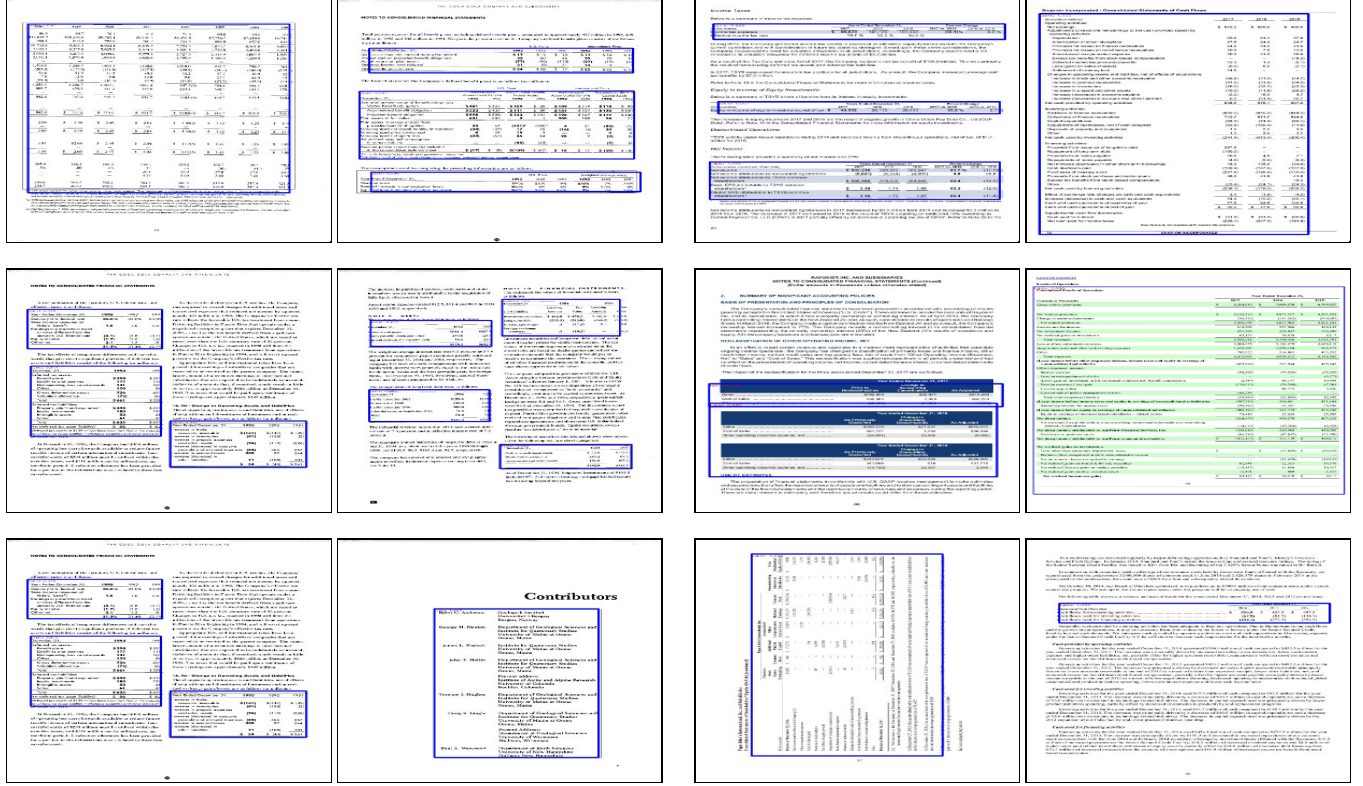
Comparison with the state-of-the-arts on ICDAR-POD2017: We compare the GOD with state-of-the-art techniques on **ICDAR-POD2017** data set. Methods like NLPR-PAL, HustVision, FastDetectors and VisInt submitted to ICDAR 2017 POD competition [27] are considered as state-of-the-art techniques. Table III highlights that our method GOD (Mask R-CNN) is better than the state-of-the-art approaches with respect to both these measures: mAP and Ave F1 when IoU threshold is set to 0.8 and 0.6. From the Table, it is observed that the performance of the existing techniques except NLPR-PAL are drastically reduced by increasing IoU threshold value. On the other hand, the performance of the GOD (Mask R-CNN) and GOD (Faster R-CNN) are reasonable stable while changing IoU threshold value from 0.6 to 0.8. This observation conclude that the GOD is robust with respect to IoU threshold value.

Figure 3(a) displays the results of localization of various graphical objects in the document images of **ICDAR-POD2017** data set using the GOD (Mask R-CNN). From Figure, it can be observed that the GOD (Mask R-CNN) is able to detect all graphical objects: table, equation and figure with large variability present in a single page. It is also observed that it is able to detect multiple tables, figures and equations in a single page.

E. Table Detection in Document Images

Comparison with the state-of-the-arts on ICDAR-2013: Due to limited number of images, we consider this

¹<http://www.icst.pku.edu.cn/cpdp/sjzy/index.htm>



(a)

(b)

Figure 4: (a) Results of the table localization using GOD (Mask R-CNN) in the document images of UNLV data set. (b) Results of table localization using the GOD (Mask R-CNN) in the document images of GO-IIT-5K data set. Blue color represents the predicted bounding box of the table.

accurately detects the various types of tables with respect to style, content and size. This observation concludes that the GOD is robust with respect to variability of tables present in the document images.

| Methods | Test Performance | | |
|-----------------------------|------------------|--------------|--------------|
| | Recall | Precision | F1 |
| Tran <i>et al.</i> [26] | 0.964 | 0.952 | 0.958 |
| DeepDeSRT [6] | 0.962 | 0.974 | 0.968 |
| Kavasidis <i>et al.</i> [8] | 0.981 | 0.975 | 0.978 |
| GOD(Faster R-CNN) | 0.974 | 0.987 | 0.981 |
| GOD(Mask R-CNN) | 1.0 | 0.982 | 0.991 |

Table IV: Comparison with state-of-the-arts based on recall, precision and F1 with IoU = 0.5 on ICDAR-2013 data set. Bold value indicates the best result.

Comparison with the state-of-the-arts on UNLV: We compare the performance of the GOD with the state-of-the-arts: Tesseract [30], Abbyy² and Gilani *et al.* [7] on UNLV data set. Table V displays the comparison results. Figure 4(a) displays the results of table localization in document images

on UNLV data set. Figure 4 highlights that the GOD (Mask R-CNN) is able to localize multiple tables with varying style in a single page.

| Methods | Test Performance | | |
|--------------------------|------------------|--------------|--------------|
| | Recall | Precision | F1 |
| Tesseract [30] | 0.643 | 0.932 | 0.761 |
| Abbyy | 0.643 | 0.950 | 0.767 |
| Gilani <i>et al.</i> [7] | 0.907 | 0.823 | 0.863 |
| GOD(Faster R-CNN) | 0.867 | 0.929 | 0.897 |
| GOD(Mask R-CNN) | 0.910 | 0.946 | 0.928 |

Table V: Comparison with state-of-the-arts on UNLV data set based on recall, precision and F1 with IoU = 0.5. Bold value indicates the best result.

F. Results on GO-IIT-5K data set

Though the GOD performs good on the public benchmark data sets. To establish the adaptability of the GOD trained model to localize tables in different types of documents, we show the results of table detection in documents of annual reports of various companies. For this purpose, we create a data set named as GO-IIT-5K by annotating 5K

²<https://www.abbyy.com/en-eu/ocr-sdk/>

| Methods | Test Performance | | | |
|-----------------------------------|------------------|---------------|---------------|---------------|
| | Recall | Precision | F1 | AP |
| *GOD (Faster R-CNN) | 0.8198 | 0.8966 | 0.8562 | 0.8461 |
| *GOD (Faster R-CNN) [†] | 0.8413 | 0.9035 | 0.8712 | 0.8637 |
| *GOD (Mask R-CNN) | 0.8995 | 0.8562 | 0.8778 | 0.8734 |
| *GOD (Mask R-CNN) [†] | 0.9659 | 0.9389 | 0.9524 | 0.9558 |
| **GOD (Faster R-CNN) | 0.7538 | 0.8425 | 0.7936 | 0.7632 |
| **GOD (Faster R-CNN) [†] | 0.7891 | 0.8610 | 0.8250 | 0.7926 |
| **GOD (Mask R-CNN) | 0.8841 | 0.8510 | 0.8676 | 0.8567 |
| **GOD (Mask R-CNN) [†] | 0.9218 | 0.9283 | 0.9250 | 0.9199 |

Table VI: Performance of the trained GOD model on **GO-IIT-5K** with IoU = 0.5. **GOD**: trained on public benchmark **ICDAR-2013**, **ICDAR-POD2017**, **UNLV** and **Marmot** data sets and tested on test images of **GO-IIT-5K** data set. **GOD[†]**: trained on the public benchmark **ICDAR-2013**, **ICDAR-POD2017**, **UNLV** and **Marmot** data sets, then fine-tuned on training images and tested on test images of **GO-IIT-5K** data set. ‘*’ indicates that data set is randomly divided into training and test sets. ‘**’ indicates that data set is divided into training and test sets based on company. Bold value indicates the best result.

pages of annual reports which are downloaded from *ICAEW website*³. This data set is divided into training set consisting of 3K and test set consisting of 2K images. This data set is different from **ICDAR-2013**, **ICDAR-POD2017** and **UNLV**. The trained GOD model on **ICDAR-2013**, **ICDAR-POD2017**, **UNLV** and **Marmot** data sets is tested on test images of **GO-IIT-5K** data set and obtain good accuracy. While we fine-tuned this trained model on training images of **GO-IIT-5K** data set and tested on test images of this data set, we obtain better table detection accuracy. It is also noted that GOD (Mask R-CNN) performs better than GOD (Faster R-CNN) for both the cases: with out fine-tuning and with fine-tuning by training images of **GO-IIT-5K** data set. Table VI highlights the performance of the GOD on **GO-IIT-5K** data set. Figure 4(b) shows the visual results obtained by GOD (Mask R-CNN). This experiment illustrates the adaptability of the trained GOD (Mask R-CNN) model to localize the graphical objects mostly table in the different type of document images. The GOD method detects a wide range of graphical objects present in the documents giving very promising results. While the other existing methods need to object wise separate classifiers, the same GOD model works for the various categories of the graphical object. Detecting tables in the scientific document such as **ICDAR** and **UNLV** data sets is an easy task as the tables or cells are well structured. The table detection task outperforms all the state-of-the-art methods in every aspect. Tables without well-defined boundaries of tables or cells are detected quite accurately. Detection of tables in the annual reports of various companies seem to be a difficult task since the format

³<https://www.icaew.com/library/company-research/company-reports-and-profiles/annual-reports>

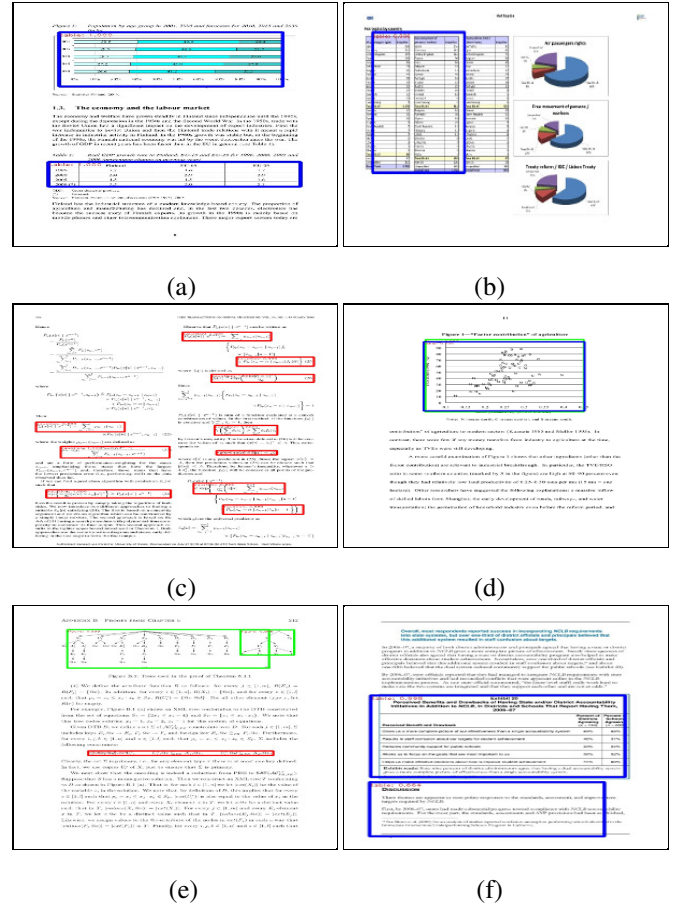


Figure 5: Examples where the GOD fails to accurately localize the graphical objects. (a) figure is detected as table. (b) table is partially detected (here, object of interest is table). (c) some equations are not detected. (d) figure is detected as both figure and table. (e) single figure is detected as multiple figures. (f) paragraph is detected as table. Blue, Green and Red colors represent the predicted bounding boxes of table, figure and equation, respectively.

of the table varies a lot for different companies and some tables are not exactly aligned as a structured one; some tables are containing texts in between two cells, for which human opinion about whether the whole table should be considered as a table is also not uniform. Figures are generally easy to identify but they can also sometimes be tricky as some figures may contain some regions which are resembled to be the tables. Another difficult object to detect is equations as they are continued to the next few lines which sometimes are unaligned. Sometimes, equations are embedded in the text regions. When, the same types/categories of objects are in very close proximity and the objects are small in size such as equations as compared to the whole document, the model fails to accurately detect the regions between two such object regions to fall into any one of the object

regions hence resulting in low mAP with high IoU score. But if the gap between two smaller sized objects is enough then objects of all sizes are localized pretty accurately. Our model outperforms the existing methods where objects were detected based on their fixed general structures of the objects.

Although, the GOD performs reasonably good to localize graphical objects: tables, figures and equations in the documents. However, it sometime fails to localize those objects having ambiguity in appearance. Figure 5 displays some cases where the GOD fails to properly localize graphical objects present in the document images. From Figure 5 (a), it is observed that one graph with table type structure is detected as table. Multiple tables close to each other in the document, GOD detects them partially as a single table (Figure 5 (b)). Among multiple equations in the document, only few of them are detected by the GOD (Figure 5 (c)). One graph plot with vertical lines is detected as both table and figure (Figure 5 (d)). One figure in the document is detected as two different graphs (Figure 5 (e)). A paragraph is detected as table (Figure 5 (f)).

V. CONCLUSION

This paper presents a novel end-to-end trainable deep learning based framework to localize graphical objects in the document images, called as Graphical Object Detection (GOD), inspired by recent object detection algorithms: Faster R-CNN [9] and Mask R-CNN [11]. To handle scarcity of labelled training samples, the GOD employs the concept to transfer learning to localize graphical object in the document images. Experiments on the various public benchmark data sets conclude that our model yields promising results as compared to state-of-art techniques. From the experiments, it is also concluded that it is able to localize multiple tables, equations and figures with large variability present in the documents. The trained model is easily adapted to localize tables in the different kinds of documents like annual reports of various companies.

REFERENCES

- [1] J. Chen and D. Lopresti, "Table detection in noisy off-line handwritten documents," in *ICDAR*, 2011.
- [2] S. Tupaj, Z. Shi, C. H. Chang, and H. Alam, "Extracting tabular information from text files," *EECS Department, Tufts University, Medford, USA*, 1996.
- [3] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage PDF documents via visual separators and tabular structures," in *ICDAR*, 2011.
- [4] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *IWDAR*, 2010.
- [5] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for PDF documents based on convolutional neural networks," in *IWDAR*, 2016.
- [6] 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.
- [7] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *ICDAR*, 2017.
- [8] I. Kavasidis, S. Palazzo, C. Spampinato, C. Pino, D. Gior-dano, D. Giuffrida, and P. Messina, "A saliency-based convolutional neural network for table and chart detection in digitized documents," *arXiv preprint arXiv:1804.06236*, 2018.
- [9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.
- [10] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.
- [11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *ICCV*, 2017.
- [12] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ACCV*, 2014.
- [13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [14] O. Okun, D. Doermann, and M. Pietikainen, "Page segmentation and zone classification: the state of the art," OULU Univ (Finland) Dept of Electrical Engineering, Tech. Rep., 1999.
- [15] M. A. Moll and H. S. Baird, "Segmentation-based retrieval of document images from diverse collections," in *DRR*, 2008.
- [16] N. Nayef and J.-M. Ogier, "Text zone classification using unsupervised feature learning," in *ICDAR*, 2015.
- [17] L. A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. on PAMI*, 1988.
- [18] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch, "Text/graphics separation revisited," in *IWDAS*, 2002.
- [19] V. P. Le, N. Nayef, M. Visani, J.-M. Ogier, and C. De Tran, "Text and non-text segmentation based on connected component features," in *ICDAR*, 2015.
- [20] X. Yi, L. Gao, Y. Liao, X. Zhang, R. Liu, and Z. Jiang, "CNN based page object detection in document images," in *ICDAR*, 2017.
- [21] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *IJCV*, 2010.
- [22] 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.
- [23] 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.
- [24] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [26] 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," *IJC*, 2015.
- [27] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "ICDAR2017 competition on page object detection," in *ICDAR*, 2017.
- [28] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "ICDAR 2013 table competition," in *ICDAR*, 2013.
- [29] A. Shahab, F. Shafait, T. Kieninger, and A. Dengel, "An open approach towards the benchmarking of table structure recognition systems," in *IWDAS*, 2010.
- [30] R. Smith, "An overview of the Tesseract OCR engine," in *ICDAR*, 2007.