

**A Mini- Project Report**  
**on**  
**“Table Detection in Images using Mmdetection Pytorch”**

Submitted to the  
Pune Institute of Computer Technology, Pune  
In partial fulfillment for the award of the Degree of  
Bachelor of Engineering  
in  
Information Technology  
by

<b>DEVASHISH PRASAD</b>	<b>33319</b>	<b>71924017E</b>
<b>KSHITIJ KAPADNI</b>	<b>33336</b>	<b>71924022M</b>
<b>SAMEER JOSHI</b>	<b>33334</b>	<b>71828813M</b>
<b>AJAY GOSAVI</b>	<b>33326</b>	<b>71828738L</b>

Under the guidance of

**Dr/Prof. R.B.MURUMKAR**



Department Of Information Technology  
Pune Institute of Computer Technology College of Engineering  
Sr. No 27, Pune-Satara Road, Dhankawadi, Pune - 411 043.

**2019-2020**

# CERTIFICATE

This is to certify that the project report entitled

## **Table Detection in Images using Mmdetection Pytorch**

**Submitted By :**

Devashish Prasad	33319
Kshitij Kapadni	33336
Sameer Joshi	33334
Ajay Gosavi	33326

is a bonafide work carried out by them under the supervision of Prof. R.B.MURUMKAR and it is approved for the partial fulfillment of the requirement of Software Laboratory Course-2015 for the award of the Degree of Bachelor of Engineering (Information Technology)

**Mr. R. B. Murumkar**

Internal Guide

Department of Information Technology

**Dr. A. M. Bagade**

Head of Department

Department of Information Technology

**Mr. R. B. Murumkar**

Internal Guide

Date :

Place:

Date:

## ACKNOWLEDGEMENT

The success and final outcome of this report required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I owe my deep gratitude to my project guide Dr. A.M. Bagade who took keen interest on my project work and guided me all along, till the completion of my project work by providing all the necessary information for developing a good report.

I would not forget to remember my external reviewer Mr. S . Deshmukh for her recommendations and more over for her timely support and guidance till the completion of my project work.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of IT Department which helped me in successfully completing my project report. Also, I would like to extend my sincere esteems to all staff in laboratory, especially Mr. B.S.Jadhav for his timely support.

( Student Name & Signature)

Devashish Prasad

Kshitij Kapadni

Sameer Joshi

Ajay Gosav

## CONTENTS

	CERTIFICATE	I
	ACKNOWLEDGEMENT	II
	LIST OF FIGURES	III
	LIST OF TABLES	IV
	NOMENCLATURE	V
CHAPTER	TITLE	PAGE NO.
	ABSTRACT	5
1.	INTRODUCTION	
1.1	Motivation	6
1.2	Application	6
2.	LITERATURE SURVEY	7
3.	DESIGN AND IMPLEMENTATION	
3.1	Dataset Description	8
3.2	Data Preprocessing	9
3.3	Data model selection	11
3.4	Model Building	12
4.	VISUALIZATION	13
5.	RESULT	16
6.	CONCLUSIONS AND FUTURE WORK	17
7.	REFERENCES	18

## II

### **Abstract**

The world is changing and going digital. The rapid growth in the number of handy devices with cameras has resulted in taking images of any document we encounter. The use of digitized documents instead of physical paper-based documents is growing rapidly.

This has made the information extraction from these image-based digital documents an essential task. Manual processing of these image based-documents requires more time and cost considering the number of documents generated each day.

These documents contain a variety of tabular information with variations in appearance and layouts. Other textual information in these documents can be extracted using various OCR (Optical Character Recognition) engines but extraction of tables from these images is a tedious task due to variations in the visually separable elements of the table.

All the previously proposed approaches used to consider table detection and table structure recognition as two separate sub-problems and solve them independently. Table detection includes detecting the region of the image that contains the table whereas table structure recognition includes segmentation of the rows and columns for extracting individual table cells from the table that is detected by the table detection model.

### III INTRODUCTION

#### A) **Motivation**

The motivation behind this project is to detect table and tabular data that is presents in different kinds of document following data science steps and approach to process large amount of data in form of images.

This has made the information extraction from these image-based digital documents an essential task. Manual processing of these image based-documents requires more time and cost considering the number of documents generated each day.

Using these we will be able to extract the tables and tabular data without having to manually go through the large documents. This makes it simpler and more time efficient that can be used to analyze this data.

#### B) **Application**

Extraction the tables and tabular data without having to manually go through the large documents can be useful in various application such as:

- a. Bills detection
- b. Research papers tabulation
- c. Word document parsing
- d. CA Audit documents

## IV LITERATURE SURVEY

In 1997, P. Pyreddy and, W. B. Croft [12] was the first to propose an approach of detecting tables using heuristics like a Character Alignment, holes and gaps. To improve accuracy, Wonkyo Seo et al. [15] used the Junctions (inter- section of the horizontal and vertical line) detection with some post-processing. T. Kasar et al. [10] also used the junction detection, but instead of heuristics, they passed the junction information to SVM.

With the ascent of Deep Learning and object detection, Azka Gilani et al. [9] was the first to propose a Deep learning- based approach for Table Detection by using Faster R-CNN based Model. They also attempted to improve the accuracy of models by introducing distance-based augmentation to detect tables. Some approaches tried to utilize the semantic information, Such as S. Arif and F. Shafait [1] attempted to improve the accuracy of Faster R-CNN by using semantic color-coding of text and Dafang He et al. [11], used FCN for semantic page segmentation with an end verification network is to determine whether the segmented part is the table or not.

In 1998, Kieninger and Dengel [15], proposed the initial approach for Table Structure Recognition by clubbing the text into chunks and dividing those chunks into cells based on the column border. Tables have many basic objects such as lines and characters. Waleed Waleed Farrukh et al. [7], used a bottom-up heuristic-based approach on these basic objects to construct the cells. Zewen, Chi et al. [5] proposed a graph-based approach for table structure recognition in which they used the SciTSR dataset constructed by them-selves for training the GraphTSR model.

Sebastian Schreiber et al. [14] were the first to perform table detection and structure recognition together with a 2 fold system which Faster RCNN for table detection and, Subsequently, deep learning-based semantic segmentation for table structure recognition. To make the model more generalize, Mohammad Mohsin et al. [13] used a combination of GAN based architecture for table detection and SegNet based encoder-decoder architecture for table structure seg-mentation.

Recently, Shubham Paliwal et al. [11], was first to propose a deep learning-based end-to-end approach to perform table detection and column detection using encoder-decoder with the VGG-19 as a base semantic segmentation method, where the encoder is the same and decoder is different for both tasks. After detection results for the table are obtained from the model, the rows are extracted from the table region using a semantic rule-based method. This approach uses a Tesseract OCR engine for text location.

## Design Details & Implementation

### 1.Dataset Description

#### 1.1 Extract

We use the public datasets of images having tables and its respective annotations in them. We use following datasets:

##### 1.Marmot Dataset

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

##### 2.ICDAR'19 Dataset and Evaluation Tool

- [https://github.com/cndplab-founder/ICDAR2019\\_cTDaR](https://github.com/cndplab-founder/ICDAR2019_cTDaR)
- [https://github.com/cndplab-founder/ctdar\\_measurement\\_tool](https://github.com/cndplab-founder/ctdar_measurement_tool)

##### 3.Github Dataset

- <https://github.com/sgrpanchal31/table-detection-dataset>

##### 4.Tablebank Dataset

- <https://github.com/doc-analysis/TableBank>

#### 1.2 Transform

Annotation of all datasets were in different formats. We converted all of these annotations into one format:

1. ICDAR
2. Marmot
3. Github

# Pascal\_VOC

\* COCO (Common Objects in Context)



## 1.3 Load

We save all the data in the drive and load it using following code:

```
[ ] from google.colab import drive
    drive.mount('/content/drive')
```

## 2.Data Preprocessing- Data Cleaning

### 2.1 Data Augmentation

Providing a large amount of training data can easily produce deep-learning-based models that can attain very high accuracy results. Deep-learning-based models have an immense hunger for data. The more we feed them with training data, the better these models perform at the predictions. For this interest, we try to implement an image-augmentation technique on the original training images to increase the size of training data. These techniques are widely used to artificially generate more training data, but not much of these techniques can be directly used for augmenting document images. Adding more training data also prevents models from over-fitting to the training data. Documents have text or content regions and blank spaces in them. As the text elements in documents are very small and the proposed model was used for detecting real-world objects in images, we try to make the contents better under standable to the object segmentation model by thickening the text regions and reducing the regions of the blank space. We use image transformation techniques that help the model to learn more efficiently. The transformed images are added in the original dataset, which also increases the amount of relevant training data for the model. We use two types of image transformation techniques as Dilation transform and Smudge transform.

- **Dilation transform**

In the dilation transform, we transform the original image to thicken the black pixel regions.

We convert the original images into binary images before applying the dilation transform.

```
[ ] import glob
import cv2
import numpy as np
img_files = glob.glob("/content/drive/My Drive/Mmdetection/VOC2007/Test/*.jpg")
for i in img_files:
    image_name = i.split("/")[-1]
    print(image_name)
    img = cv2.imread(i,0)
    _, mask = cv2.threshold(img,220,255,cv2.THRESH_BINARY_INV)
    kernal = np.ones((2,2),np.uint8)
    dst = cv2.dilate(mask,kernal,iterations = 3)
    cv2.imwrite("/content/drive/My Drive/Mmdetection/VOC2007/Augmented_Test/"+str(image_name),cv2.bitwise_not(dst))
```

## • Smudge transform

In the smudge transform, we transform the original image to spread the black pixel regions, a kind of smeary blurred black pixel region. Like the dilation transform, we convert the original images into binary images before applying the smudge transform.

```
[ ] img = cv2.imread(i)

x,y,_ = img.shape
b,g,r = cv2.split(img)

# Apply Basic Transformation
b = basicTransform(b)
r = basicTransform(r)
g = basicTransform(g)

# Perform the distance transform algorithm
b = cv2.distanceTransform(b, cv2.DIST_L2, 5) # ELCUDIAN
g = cv2.distanceTransform(g, cv2.DIST_L1, 5) # LINEAR
r = cv2.distanceTransform(r, cv2.DIST_C, 5) # MAX

# Normalize
r = cv2.normalize(r, r, 0, 1.0, cv2.NORM_MINMAX)
g = cv2.normalize(g, g, 0, 1.0, cv2.NORM_MINMAX)
b = cv2.normalize(b, b, 0, 1.0, cv2.NORM_MINMAX)

dist = cv2.merge((b,g,r))
dist = cv2.normalize(dist,dist, 0, 4.0, cv2.NORM_MINMAX)
dist = cv2.cvtColor(dist, cv2.COLOR_BGR2GRAY)

cv2.imshow("asab",dist)

data = dist.astype(np.float64) / 4.0
data = 1500 * data # Now scale by 255
dist = data.astype(np.uint16)

#normalizedImg = np.zeros((x, y))
#normalizedImg = cv2.normalize(dist, dist, 0, 25, cv2.NORM_MINMAX)

cv2.imwrite(PATH_TO_DEST+"/Smudge_"+image_name,dist)
cv2.imshow("aaa",dist)
cv2.waitKey(0)
```

### 3.Data Model Selection

For selecting the model which provides with better accuracy and predictions several models can be tested such as:

1. **RetinaNet** : Resnext-101 based RetinaNet model with cardinality = 32 and bottleneck width = 4d along with Feature Pyramid Network (FPN) neck.
2. **FasterRcnnHRNet** : Faster R-CNN with hrnetv2p w40 backbone (40 indicates the width of the high-resolution convolution) having Feature Pyramid Network (FPN) neck.
3. **CascadeRcnnResneXt** : Three staged Cascade R-CNN with Resnext- 101 backbone having cardinality = 64 and bottleneck width = 4d having Feature Pyramid Network (FPN) neck.
4. **CascadeRcnnHRNet** : Three staged Cascade R-CNN with hr- netv2p w32 backbone having Feature Pyramid Net- work (FPN) neck.
5. **CascadeMaskRcnnDeConv** : Three staged Cascade R-CNN with Resnet-50 backbone with c3-c5 (adding deformable convolutions in resnet stage 3 to 5) having Feature Pyra- mid Network (FPN) neck.
6. **CascadeMaskRcnnResneXt** : Three staged Cascade mask R-CNN with Resnext-101 backbone having cardinality = 64 and bot- tleneck width = 4d having Feature Pyramid Network (FPN) neck.
7. **CascadeMaskRcnnHRNet** : Three staged Cascade mask R-CNN with hrnetv2p w32 backbone having Feature Pyramid Net- work (FPN)

Model	IoU				WAvg.
	0.6	0.7	0.8	0.9	
Retina	0.818	0.785	0.762	0.664	0.749
FRcnnHr	0.889	0.877	0.862	0.781	0.847
CRccnHr	0.927	0.910	0.901	0.833	0.888
CRcnnX	<b>0.929</b>	<b>0.913</b>	<b>0.903</b>	<b>0.852</b>	<b>0.895</b>
CMRcnnDC	0.912	0.897	0.880	0.834	0.877
CMRcnnX	0.931	0.925	0.909	0.868	0.905
CMRcnnHr	<b>0.941</b>	<b>0.932</b>	<b>0.923</b>	<b>0.886</b>	<b>0.918</b>

Figure 2:Different models and accuracy throughput

## 4. Model Building

### 4.1 Model Architecture

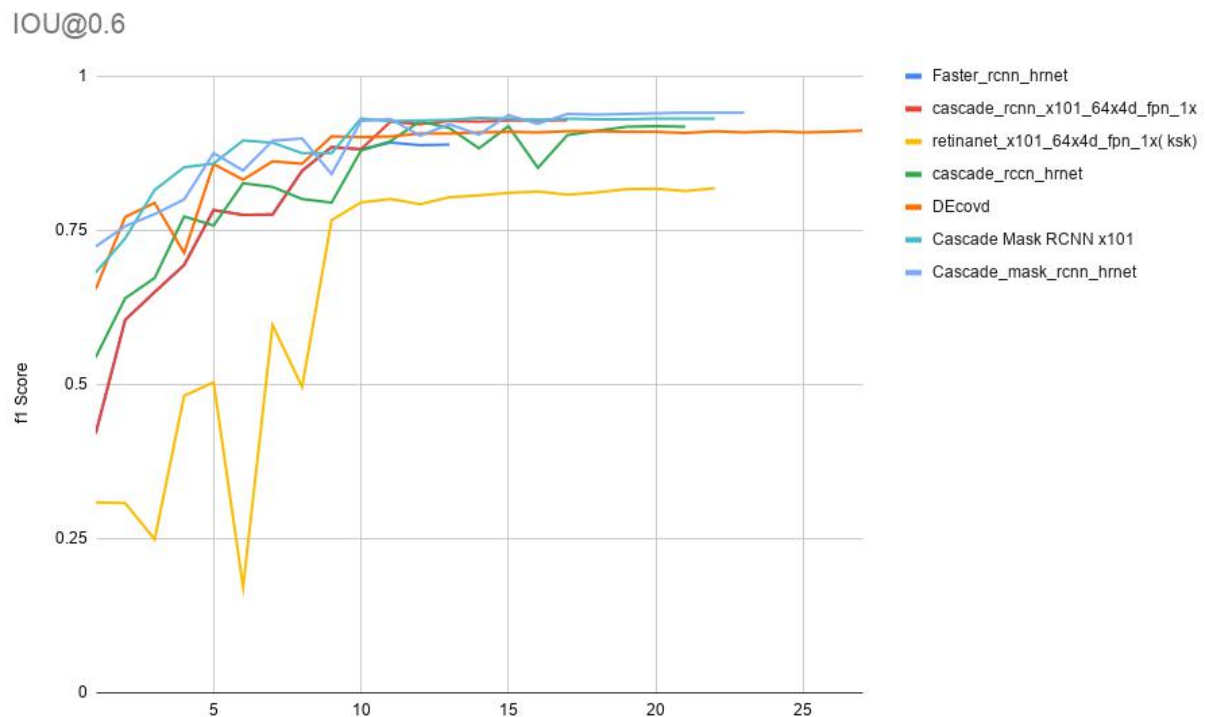
To attain very high accuracy results we use a model that was made by the combination of two approaches. Cascade RCNN was originally proposed by Cai and Vasconcelos to solve the paradox of high-quality detection in CNNs by introducing a multi-stage model. And a modified HRNet was proposed by Jing dong Wang et al. to attain reliable high resolution representations and multi-level representations for semantic segmentation as well as for object detection. Our experiments and analysis show that the cascaded multistaged model with the HRNet backbone network yields the best results due to the ability of both the approaches to strive for high accuracy object segmentation. CascadeTabNet is a three-staged Cascade mask R-CNN HRNet model. A backbone is a part of the model that transforms an image to feature maps, such as a ResNet50 without the last fully connected layer. CascadeTabNet uses HRNetV2p-W32 (32 indicates the width of the high-resolution convolution) as the backbone for the model. A neck is a part that connects the backbone and heads. It performs some refinements or reconfigurations on the raw feature maps produced by the backbone. A Feature Pyramid Network (FPN) based neck was implemented for the model as originally used in HRNetV2p. The multi-resolution representations proposed in (HRNetV1) were upsampled and concatenated for semantic segmentation to form HRNetV2. Then a feature pyramid was formed over HRNetV2 for object detection to form HRNetV2p. In HRNetV2p, all four-resolution representations are outputted from the network while the gray box indicates the way of obtaining the output representation from the input four-resolution representations. The benefit of this modification is that the capacity of the multi-resolution convolution is fully explored. This modification only adds a small parameter and computation overhead. Cascade mask R-CNN is similar to the architecture of Cascade R-CNN architecture. In Cascade R-CNN, the first stage is a proposal sub-network, in which the entire image is processed by a backbone network, such as HRNet, and a proposal head (“H0”) is applied to produce preliminary detection hypotheses, known as object proposals. In the second stage, these hypotheses are processed by a region-of-interest detection sub-network (“H1”), denoted as a detection head. A final classification score (“C”) and a bounding box (“B”) are assigned per hypothesis. The entire detector is learned end-to-end, using a multi-task loss with bounding box regression and classification components. The Cascade R-CNN architecture is extended to the instance segmentation task, by adding a segmentation branch similar to that of the Mask R-CNN. For image segmentation using the Cascade R-CNN, proposes multiple strategies.

## 4.2 Model Training

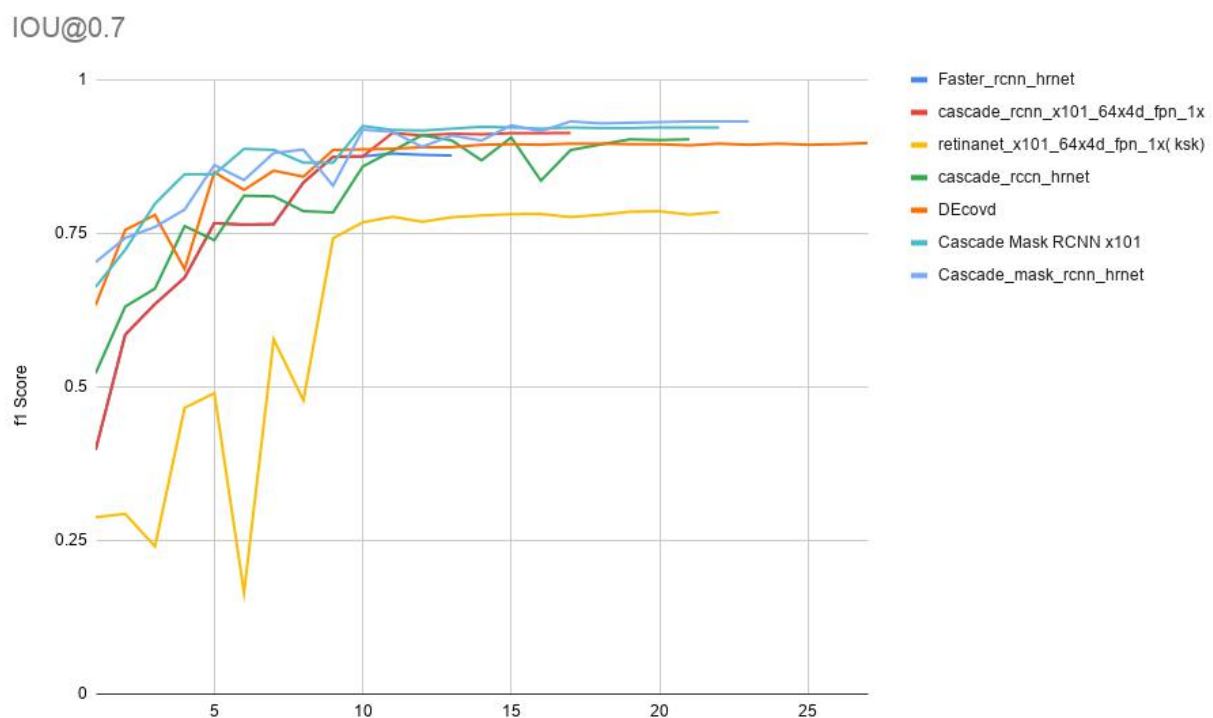
```
!python "/content/drive/My Drive/Mmdetection/mmdetection/tools/train.py" "/content/drive/My Drive/chunk cascade_mask_rcnn_hrnetv2p_w32_20e.py"
```

## 5. Visualization

### ● IOU 0.6

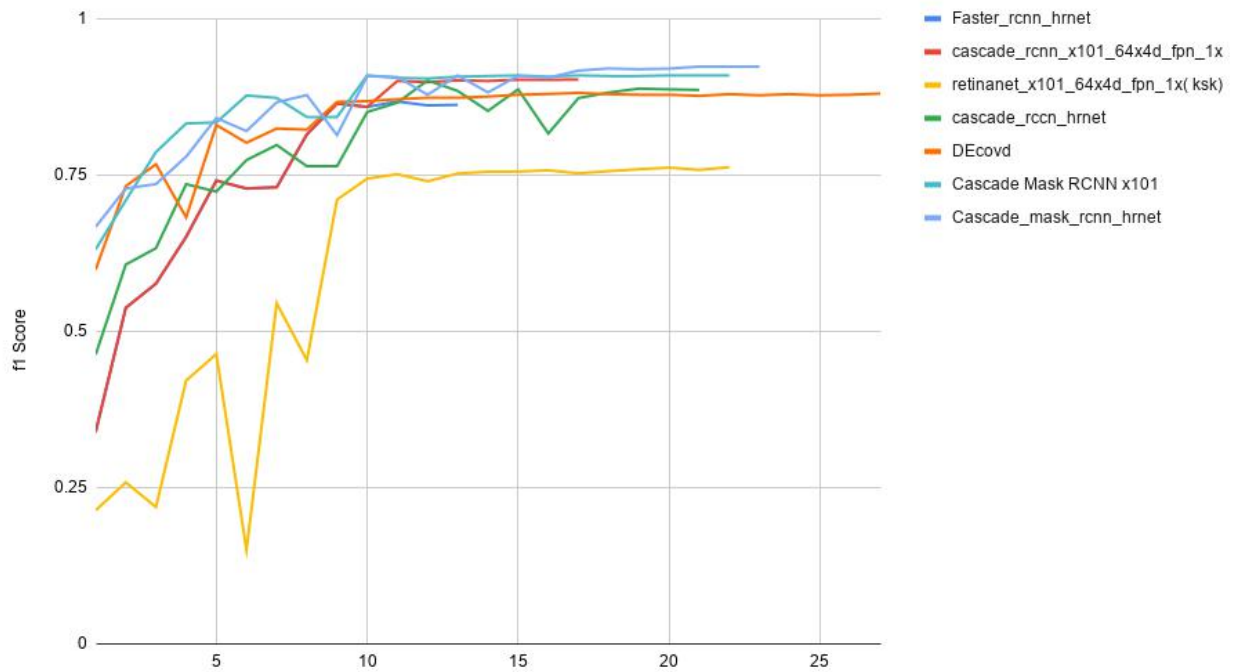


### ● IOU 0.7



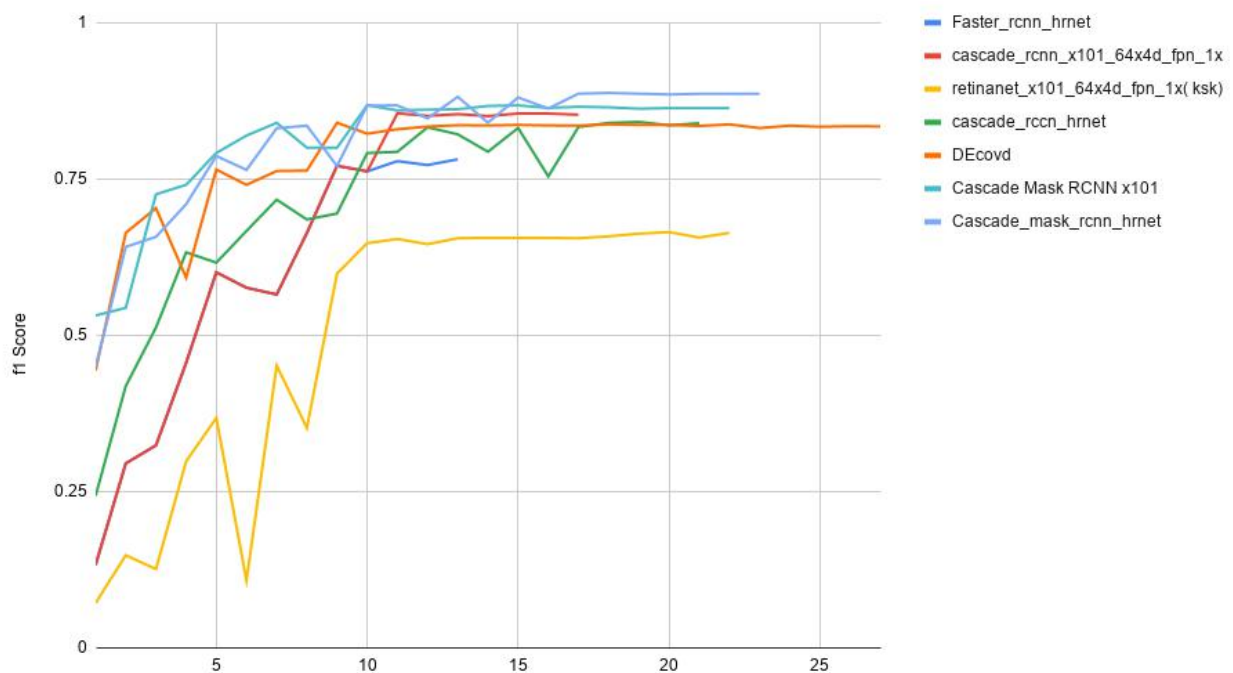
## ● IOU 0.8

IOU@0.8



## ● IOU 0.9

IOU@0.9






# RESULTS

On testing the data using the trained model i.e CascadeMaskRcnnHRNet  
On the sample data we get accurate results.

For example :

南京新创物业管理股份有限公司2016年度第三期融资募集说明书

2019年半年度报告



南京新创物业管理股份有限公司成立于2016年4月29日，注册资本为人民币300万元，由发行人的全资子公司南京创特物业管理有限公司持股50%。该公司主营业务为物业管理及配套设施；房屋、机电设备维修；保洁服务；园林绿化工程；电子技术开发，空调设备安装及维修；室内装饰；房产中介服务；停车场管理服务；餐饮服务（须取得许可或审批后方可经营）；会务服务；酒店管理。

截至2017年12月31日，南京新创物业管理有限公司资产合计为1,172.08万元，负债合计为508.85万元，所有者权益合计为663.23万元；2017年度，该公司实现营业收入1,349.61万元，净利润267.12万元。

## （二）发行人主要参股子公司情况

图表 5-8：截至 2016 年 9 月末发行人主要参股公司情况

单位：万元

序号	被投资单位名称	业务性质	注册资本	持股比例		核算方式
				直接	间接	
1	江苏五维电子科技有限公司	电子科技	3,000.00	20.00%		权益法
2	南京新城云和数字平台运营有限公司	电子科技	500.00		30.00%	权益法
3	南京新城物业管理有限公司	物业管理	200.00	40.00%		权益法

## 发行人主要参股子公司情况介绍：

### （1）江苏五维电子科技有限公司

江苏五维电子科技有限公司成立于2011年11月1日，注册资本5,000万

### 3.2.2 利息净收入

报告期内，本集团实现利息净收入512.97亿元，同比减少5.06亿元，下降0.98%。

下表列示报告期内本集团资产端项目利息收入、平均收益和成本情况

单位：人民币百万元

	本集团			上一年度		
	平均余额	利息收入	平均收益率(%)	平均余额	利息收入	平均收益率(%)
存放同业	3,253,488	36,110	1.11	3,101,400	21,307	0.69
拆出	1,113,481	80,791	7.26	1,017,390	10,349	1.01
存放中央银行款项	804,536	3,274	0.41	588,391	3,759	0.64
存放同业及拆入	221,890	0,170	0.08	270,628	0,000	0.00
其他金融资产	10,000	1,000	1.00	84,231	1,100	1.31
合计	5,393,395	121,275	2.25	5,082,040	36,415	0.72

单位：人民币百万元

	本集团			上一年度		
	平均余额	利息支出	平均成本率(%)	平均余额	利息支出	平均成本率(%)
同业存款	3,183,901	28,836	0.91	3,019,391	25,537	0.85
同业及存放中央银行款项	1,001,103	10,002	1.00	1,001,103	28,138	2.81
已发行债务证券	670,000	10,000	1.50	691,270	12,000	1.74
向中央银行借款	101,228	5,000	4.94	101,228	5,000	4.94
合计	5,056,232	53,838	1.06	5,012,992	70,675	1.41

### 3.2.2.1 利息收入

报告期内，本集团实现利息收入1,326.23亿元，同比增长140.30亿元，增长11.85%；本集团加大了零售业务的发展力度，零售贷款利息收入410.96亿元，同比增加81.05亿元，增幅20.40%，平均收益率6.43%，同比上升0.37个百分点。

### 贷款及净利息收入

单位：人民币百万元

	本集团			上一年度		
	平均余额	利息收入	平均收益率(%)	平均余额	利息收入	平均收益率(%)
公司贷款	1,432,419	42,812	3.00	1,291,781	40,398	3.13
零售贷款	1,285,719	41,000	3.19	1,098,401	31,000	2.83
其他贷款	120,403	1,463	1.21	104,017	470	0.45

注：其中，一般性贷款平均收益率为3.08%，中长期贷款平均收益率为3.07%

44

8

## CONCLUSION

This Report presents an end-to-end system for table detection task. It is shown that existing instance segmentation based CNN architectures which were originally trained for objects in natural scene images are also very effective for detecting tables. And, iterative transfer learning and image augmentation techniques can be used to learn efficiently from a small amount of data. The performance of our system is evaluated on the publicly available ICDAR 2019 table competition dataset and on TableBank for the table detection task. We achieve 3rd rank on cTDaR competition post-competition results for the table detection task. We achieve the highest accuracy for the table detection task on the TableBank dataset.



## REFERENCES

List all the material used from various sources for making this project proposals

- [1] S. Arif and F. Shafait. Table detection in document images using foreground and background features. In 2018 DigitalImage Computing: Techniques and Applications (DICTA), pages 1–8, 2018. 2
- [2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. 2, 3
- [3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation, 2019. 3
- [4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Mmdetection: Openmmlab detection toolbox and benchmark, 2019. 3, 6
- [5] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanx-uan Yin, and Xian-Ling Mao. Complicated table structure recognition, 2019. 2
- [6] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu. Dataset, ground-truth and performance metrics for table detection evaluation. In 2012 10th IAPR International Workshop on Document Analysis Systems, pages 445–449, 2012. 5
- [7] W. Farrukh, A. Foncubierta-Rodriguez, A. Ciubotaru, G. Jaume, C. Bejas, O. Goksel, and M. Gabrani. Interpreting data from scanned tables. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 02, pages 5–6, 2017. 2
- [8] L. Gao, Y. Huang, H. Déjean, J. Meunier, Q. Yan, Y. Fang, F. Kleber, and E. Lang. Icdar 2019 competition on table detection and recognition (ctdar). In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 1510–1515, 2019. 5, 7, 8
- [9] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait. Table detection using deep learning. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 771–776, 2017. 2, 5, 7
- [10] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet. Learning to detect tables in scanned document images using line information. In 2013 12th International Conference on Document Analysis and Recognition, pages 1185–1189, 2013.

- [11] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: Table benchmark for image-based table detection and recognition. arXiv preprint arXiv:1903.01949, 2019.
- [12] Shubham Paliwal, Vishwanath D, Rohit Rahul, Monika Sharma, and Lovekesh Vig. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images, 2020.
- [13] P. Pyreddy and W. B. Croft. Tinti: A system for retrieval in text tables title2:. Technical report, USA, 1997. 2 M. M. Reza, S. S. Bukhari, M. Jenckel, and A. Dengel. Table localization and segmentation using gan and cnn. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), volume 5, pages 152–157, 2019.
- [14] 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 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 01, pages 1162–1167, 2017.
- [15] Wonkyo Seo, Hyung Koo, and Nam Cho. Junction-based table detection in camera-captured document images. International Journal on Document Analysis and Recognition (IJDA), 18, 03 2014.