UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

LOCALIZATION OF TABLES AND PLOTS IN DOCUMENTS USING DEEP NEURAL NETWORKS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

VISHNU PRIYATAMKUMAR KADIYALA
Norman, Oklahoma
2022

LOCALIZATION OF TABLES AND PLOTS IN DOCUMENTS USING DEEP NEURAL NETWORKS

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Samuel Cheng, Chair

Dr. Justin Metcalf

Dr. Bin Zheng

ACKNOWLEDGEMENTS

First and foremost, I am extremely grateful to my Chair, Dr. Samuel Cheng for his invaluable advice, continuous support, and patience during my master’s study. His immense knowledge and plentiful experience have encouraged me all the time in my academic research and daily life. I would also like to acknowledge his immense support during my hardships and encouraging words to keep me motivated during this time. I would also like to thank Dr. Bin Zheng and Dr. Justin Metcalf for their support of my thesis and for being a part of my committee. Finally, I would like to express my gratitude to my parents and friends. Without their tremendous understanding and encouragement over the past few years, it would be impossible for me to complete my study.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION 1

CHAPTER 2: BACKGROUND 5

2.1 Deep Learning 5

2.2 Object Detection Frameworks 7

2.2.1 Traditional Object Detection Frameworks 7

2.2.2 Deep Learning based Object detection Framework (two stage detectors) 10

2.2.3 Deep Learning based Object detection Framework (single stage detectors) 13

2.3 Related Work 15

CHAPTER 3: METHODOLOGY 17

3.1Data Generation 18

3.2 Annotation 19

3.3 Experimental Setup 21

CHAPTER 4: EXPERIMENTAL RESULTS 25

4.1 Metrics of Evaluation 25

4.1.1 Intersection over Union (IoU) 25

4.1.2 Mean Average Precision(mAP) 26

4.2 Single Shot multibox Detector 1 – MobileNet 27

4.3 Single Shot multibox Detector 2 – RetinaNet 30

4.4 CenterNet 32

CHAPTER 5: ANALYSIS and DISCUSSION 36

CHAPTER 6: CONCLUSION 39

6.1 Summary 39

6.2 Future Scope 40

Bibliography 41

LIST OF FIGURES

Figure 1: A Convolutional Network Example 6
Figure 2: Multi-layer Artificial Neural Network 7
Figure 3: Haar Like features used in initial object detection framework 8
Figure 4: Histograms of oriented gradients for object detection framework 9
Figure 5: Deformable part-based Model in object detection framework 10
Figure 6: Regions with CNN features for object detection 11
Figure 7: Fast RCNN for object detection 12
Figure 8: Faster RCNN for object detection 12
Figure 9: Illustration of a Feature Pyramid network used for object detection 13
Figure 10: You Only Look Once (YOLO) for Object detection 14
Figure 11: SSD: Single Shot Multibox Detector for Object detection 15
Figure 12: Sample Training Dataset Image 19
Figure 13: Sample Annotation using LabelImg Annotation Software on generated Dataset File 21
Figure 14: Learning window for a Retina Net Network at 9000 epochs 24
Figure 15: Intersection over Union Representation and Formula Evaluation 26
Figure 16: Intersection over Union Example for Poor, Good and Excellent Prediction 26
Figure 17: Accurate prediction of Plot and name block for SSD1 – mobilenet 28
Figure 18: Classification loss over 10,000 epochs for SSD1- Mobilenet 29
Figure 19: Localization loss over 10,000 epochs for SSD1 - Mobilenet 29
Figure 20: Accurate Prediction of Table and Nameblock on SSD2 – RetinaNet 30
Figure 21: RetinaNet Loss for Object Center over 10000 epochs 31
Figure 22: RetinaNet Loss for Box scale over 10000 epochs 31
Figure 23:Accurate Prediction of table and Name block by CenterNet 32
Figure 24: CenterNet Loss for box scale over 10000 epochs 33
Figure 25: CenterNet Loss for object center over 10000 epochs 33
Figure 26: Accurate Prediction on a PDF file downloaded from internet 35
Figure 27: Learning rate over 10000 epochs 43
Figure 28: time per step for each model 43
Figure 29: mAP precision for all the models 44
Figure 30: AR Recall for all the models 44

ABSTRACT:

There has been an immense increase in number of scientific publications being published every single day, it has been increasingly difficult to keep up with all the new results being published. In this research, we localized and detected all the plots and tables from documents using deep neural networks. We generated a custom document dataset and manually annotated it to train and evaluate object detection models and their customizability. We used two Single shot multi detector models with base model of MobileNet, RetinaNet and CenterNet model. We trained these models over 10000 epochs on the custom generated dataset. All three models were able to localize and detect the plots and tables with accurately predicted bounding boxes. The results were as follows with CenterNet having the highest mAP score of 92 and highest AR of 93.88 followed by RetinaNet with mAP score of 91.1 and AR of 93.76 lastly, MobileNet based SSD with mAP score of 89.04 and AR of 91.54.

KEYWORDS: Bounding box, Custom dataset, Document extraction, Localization, Object Detectio

CHAPTER 1: INTRODUCTION

In today’s world Machine learning and Deep learning play a particularly key role in many applications and use cases. It influences the way we develop our future systems and increases the possibility of automation not only in the field of Computer Science but also impacts other distinctly related fields such as biology, geology and many more (Alpaydin, 2021). What makes Machine learning such an interesting field is its ability to adopt to the problem and learn by supplying sufficient training data. The applications of machine learning and deep learning have been of interest for at least two decades now (Wason, 2018) (Mitchell, 2006). This also accompanied the development of neural networks due to their ability to adapt to non-linear datasets. Neural networks exploded the popularity of the whole field. The pace of development in machine learning and deep learning has introduced some amazing new subfields such as Computer Vision, Natural Language Processing, and many more, and quickly started solving new problems such as Object detection and Segmentation (Tautz) (Rosenfeld) (Haralick).

In recent years, Computer vision has evolved greatly as a field; applications such as autonomous driving and pose estimation have gained immense popularity. However, due to the introduction of large image datasets such as COCO, CIFAR10, MNIST and many others have shifted the research focus. The papers in this era have started to optimize the models to a particular dataset, making the overall turnaround time of less than a year to build new models and convert it into a publication. With such focus for years, and development for many different applications introduced much scientific literature such as Journal papers, Conference and Publications. The explosion of the number of papers being produced every single year is enormous and poses a problem for peer review of these papers (Ruder, 2018). According to ICML official publications, there were 1184 papers published in the ICML 2021 conference alone, using deep learning algorithms for localization will help the user compare the results and consolidates all the information in form of snippets (Z. -Q. Zhao, Nov. 2019). This opens a gap for assessing these models on a different and a new completely disjoint dataset to evaluate the performance of models in a real-world scenario. Segmenting all the information from a paper will help in extracting information from different sections of the paper such as tables and plots. This will help us analyze a paper in a shorter period.

Object detection is one of the most fundamental problems in computer vision, object detection seeks to identify and locate objects in an image or a series of images by training on a large number of natural images from particular categories (Wu, (2020).). Deep learning techniques have appeared as a powerful strategy for learning on the examples and have led to breakthroughs in the field of computer vision (Wu, (2020).). Object detection is used to either localize instances of a particular object or used to generalize the detection of an object category by creating bounding boxes around the train and test images, a rectangle tightly bound to the object being localized. Some publications such as the (Krizhevsky) have achieved huge traction in the scientific community for their Deep convolutional network called AlexNet which achieved a record-breaking image classification accuracy in Large Scale Visual Recognition Challenge (ILSVRC). The AlexNet localizes and classifies about two hundred different classes from the image dataset of LSVRC.

In any research, the results that have been previously published will be used to form a firm basis for the present and future work. To extract these results from a scientific journal is very tedious and laborious. As mentioned earlier, the explosion of the number of scientific pieces of literature in this field makes it close to impossible to go through all the literature being published. The ability to compare results from different authors and find interesting results that will provide the basis for future research makes it an important task to accomplish. Advancements in the field of computer vision and notable contributions in Segmentation and object detection make it an interesting approach to use these technologies to help localize and identify the results and consolidate them in a manner that makes it easier for a researcher to compare. The researcher, however, will have to go through the scientific literature once we establish the interest and basis for his research by detecting tables and plots from the scientific literature.

Overall, the research problems that we are trying to solve in this thesis are

1. Can the Scientific literature be localized and consolidated?
2. How can we best localize the results of all the literature?
3. What different methods can we use to perform the task? and evaluate them

Chapter 1 provided the necessary introduction to the problem at hand and why it is important to solve the problem while also supplying some information about the field of Computer Vision and how it relates to the problem. Chapter 2 will supply background information on object detection methods and data generation techniques; it will also provide information about the different metrics that are used in the object detection field for reference. Chapter 3 will supply the methods used in generating the data and how the object detection model is run to establish a validation for the future chapter. Chapter 4 will supply the result achieved by different methods and the bounding boxes around the different train and test examples. It will also supply the list of hyperparameters used to achieve the results. In Chapter 5 we will discuss the results using different metrics introduced in Chapter 2. We will establish tradeoffs by using different methods and models shown in Chapter 4. Chapter 6 provides the inference and conclusion of the entire thesis and discusses about the future scope and limitations of the thesis.

CHAPTER 2: BACKGROUND

2.1 Deep Learning

Object detection algorithms are based on deep learning and many larger models that use images as an input use convolutional neural networks to extract feature maps from the input image. Hence, it is necessary to provide some background about the way CNNs operate and their role in Deep learning used in this thesis.

Deep learning has revolutionized the machine learning field. It has played a significant role in most applications such as image classification, processing, speech recognition and natural language processing. Convolutional Neural Networks (CNN) have special convolutional layers that empower deep learning in accomplishing such complex tasks. The role of convolution is to simply perform mathematical convolution between the input image and the weights of the produced kernel. Where the kernel generated is either defined by the user or randomly generated by the modules. This convolution can occur in a single dimension also known as 1D convolution or occur over multiple dimensions, usually two dimensions known as 2D convolution. For convolutions over 1D, they occur between multiple layers and usually carry an activation function to introduce non-linearity such as sigma (LeCun). Equation 1 depicts a convolution between a layer and the weights of the image, where x is a layer *l-1* in the training example and w is a filter or weights of the kernel that is being used on layer *x.* Equation 2 depicts a convolution over multiple dimensions. The convolution is in between the input feature map N^{l-1} and the filter weights of the corresponding kernel. Sigma is an elementwise non-linear function that is used as an activation function typically a rectified linear unit (RELU). It is used to maximize the value of x by eliminating any value that goes below 0. If the 2D convolution returns a negative value, it uses a zero for the feature map.

$$x^{l-1} * w^{l} \tag{1}$$

$$x_j^l = \sigma(\sum_{i=1}^{N^{l-1}} x_i^{l-1} * w_{i,j}^l + b_j^l) \tag{2}$$

Figure1 depicts a Convolutional Neural Network; as an input image is offered to the network, a series of convolutions occurs on the image creating *N* feature maps. The RELU activation function is then applied to the feature maps to eliminate negative values. These are then max pooled to acquire the best features while eliminating redundant or unnecessary values such as noise in feature maps. It also helps in providing downsampling/upsampling of feature maps.

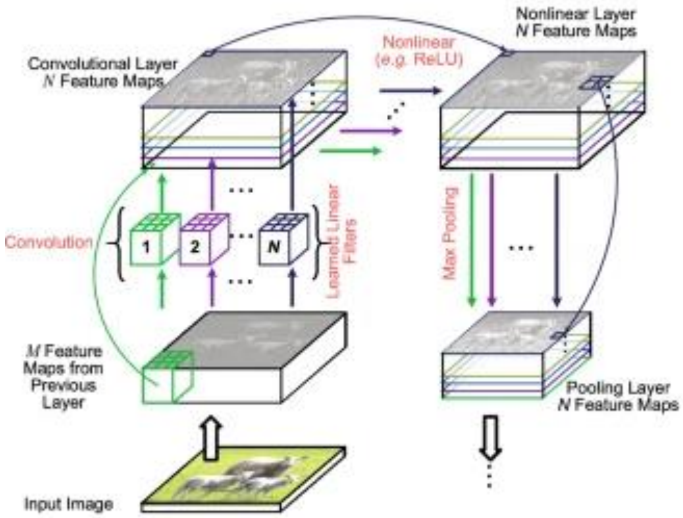


Figure 1: A Convolutional Network Example

Neural networks follow the feature maps generated by the convolutional layers. Figure 2 represents a multi-layer perceptron (M.W Gardner). A multi-layer perceptron takes feature maps from the convolutional layers, and updates weights and biases for the network by backpropagation algorithm usually referred to as learning (Oludare Isaac Abiodun, 2018) (Simpson).

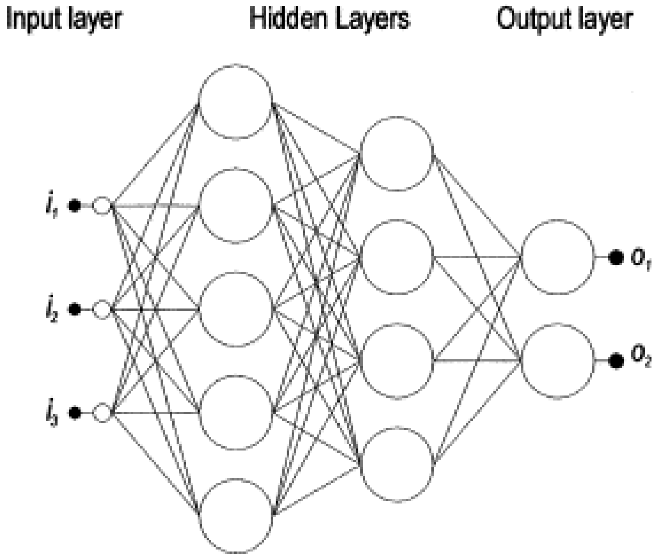


Figure 2: Multi-layer Artificial Neural Network

2.2 Object Detection Frameworks

2.2.1 Traditional Object Detection Frameworks

There has been a steady increase in object detection, its feature representations, and classifiers for recognition, it started from providing the Neural network with hand crafted features to letting machine pick the best learning features. We will refer to it as traditional object detection and for more advanced framework we will refer as deep learning-based object detection. One of the major contributions to this field is by the paper (Jones, 2001), the framework detects human faces in real-time. It uses sliding windows to go through all possible locations and scales in an image and checks for human face at any point in the process. The sliding window tries to find haar like features also known as haar wavelets. In the initial version, they used handpicked features and quickly shifted to a more automated AdaBoost algorithm.

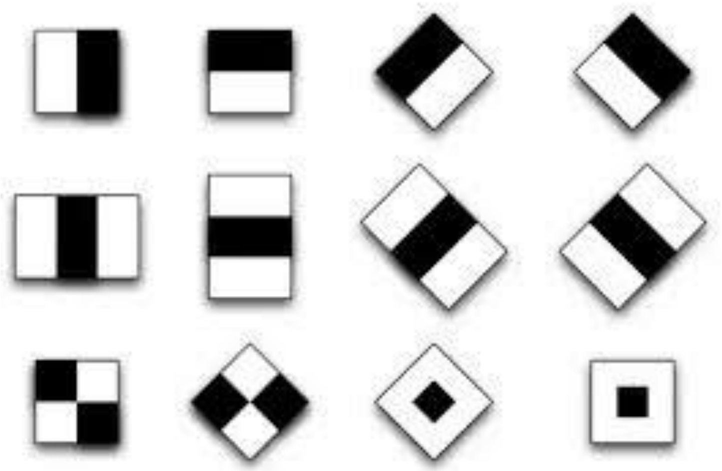


Figure 3: Haar Like features used in initial object detection framework

Histograms of oriented gradients for human detection also known as the HOG detector (Triggs, 2005). This framework provided an improvement over the sliding windows approach. It improved on scale invariant features and included some shape contexts. HOG works on a technique called blocks, which is a dense pixel grid in which the gradients are constituted from magnitude and direction of change of intensity in pixels within the block. HOGs were known for their work in pedestrian detection. To account for the scale invariance, HOG detector resized the image into different form factors and used the same size window for detection.

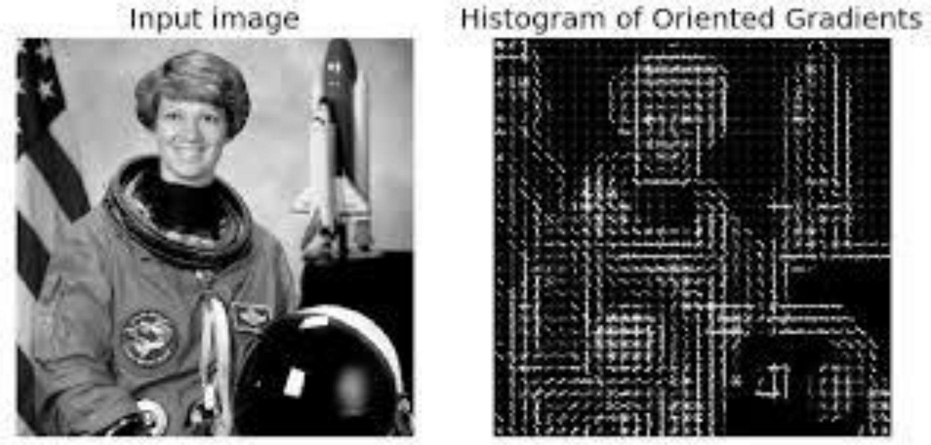


Figure 4: Histograms of oriented gradients for object detection framework

Deformable part-based model (DPM) originally proposed by (Pedro F. Felzenszwalb, 2008) is basically an extension to the HOG detector, it was initially used for pedestrian detection later modified to detect a car. It used a divide and conquer approach by detecting windows, body, and wheels of a car. The image is decomposed into several parts and finally would ensemble detections of all the objects mentioned earlier. There were other important techniques that were introduced in this research such as hard negative mining, bound box regression and context priming. Later the author improved detection by implementing a cascade architecture. The cascade architecture improved the speed of by over ten times with the same accuracy (Ross B. Girshick Forrest, 2014).

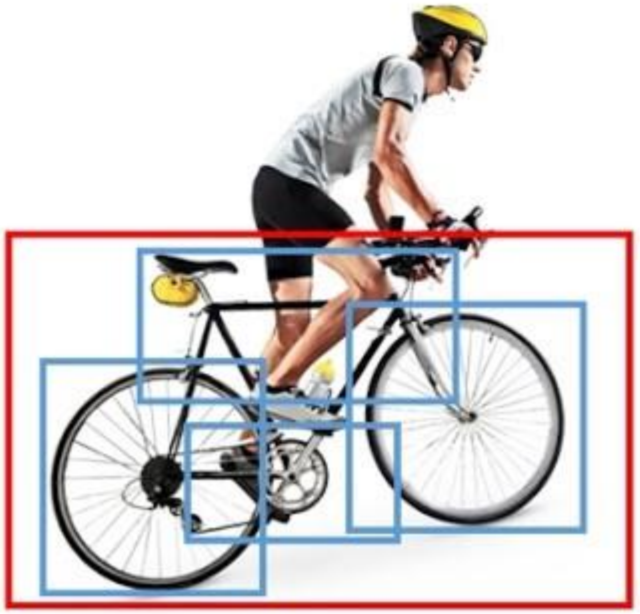


Figure 5: Deformable part-based Model in object detection framework

2.2.2 Deep Learning based Object detection Framework (two stage detectors)

Around the year 2010 object detection saturated at the traditional level. It advanced from hand-picking features to introduction of deep convolutional; networks that are best at learning robust and high-level features. This change was introduced by regions with CNN features (RCNN). The deep learning era introduced us to two types of models two stage detection and a one stage detection. The RCNN extracts a set of object proposals by selective search. Each image is resized to a fixed image size for the network and a pre-trained CNN to extract image features. These features are then fed into a classifier like Support Vector Machine (SVM) to predict presence of an object and its category (Girshick R. a., 2013).

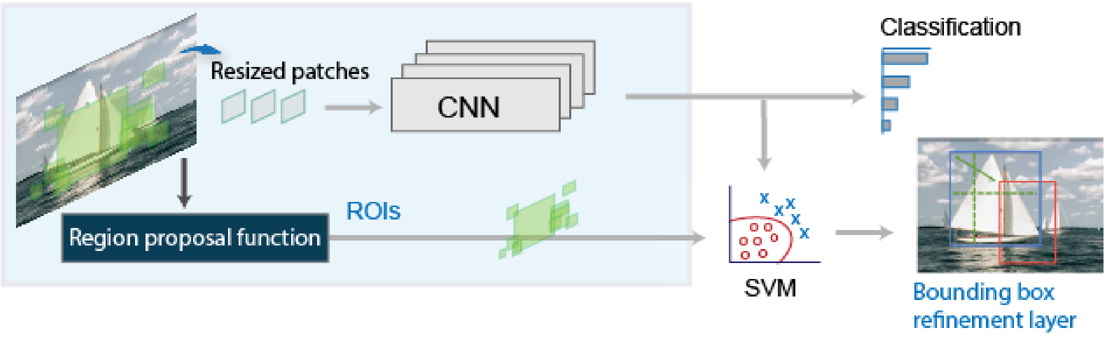


Figure 6: Regions with CNN features for object detection

With an initial approach with deep learning the RCNN performed better than the traditional methods. Although, it took an enormous amount of time to train as there were a lot of redundant feature computations and over two thousand bounding boxes from one image. This introduced the Spatial Pyramid Pooling Networks (SPPNet). The author of SPPNet (Kaiming He, 2014) suggests that adding multiple pooling layers with different scales at the transition of convolutional layer to fully connected layer. An SPP net enables CNN to generate representations of fixed lengths without having to scale the image.

Fast RCNN uses complete images unlike the RCNN where only the features from proposed region are used. Fast RCNN achieves this by generating region of interest from the convolutional network and uses these features to feed into a fully connected layer, it uses a softmax regression technique during the bounding box generation to suppress multiple object predictions (Girshick R. , 2015).

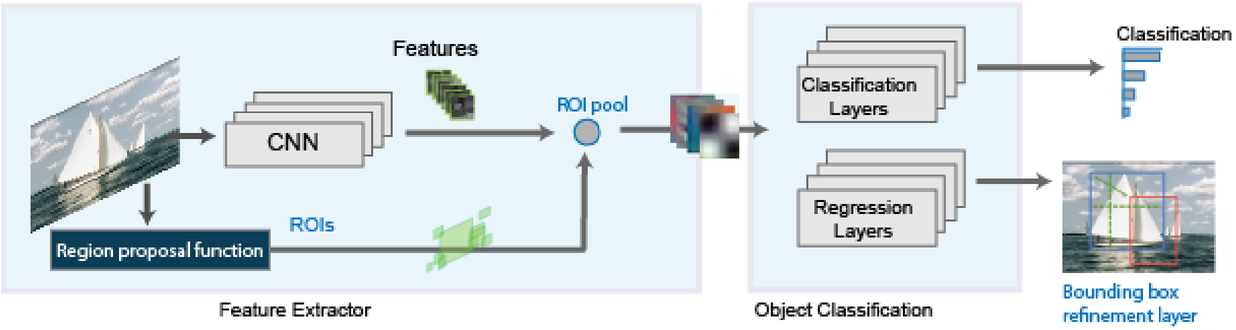


Figure 7: Fast RCNN for object detection

The Fast RCNN was short lived with the introduction of faster RCNN by (Ren, 2015). It was one of the first real-time deep learning object detector. The models discussed prior used selective search algorithms to find the ROIs. Faster RCNN eliminated the selective search and introduced a region proposal network where the network would learn by itself without intervention from external algorithms. The feature map could recognize the region proposals and these region proposals were reshaped using pooling layers and predict the values for bounding boxes.

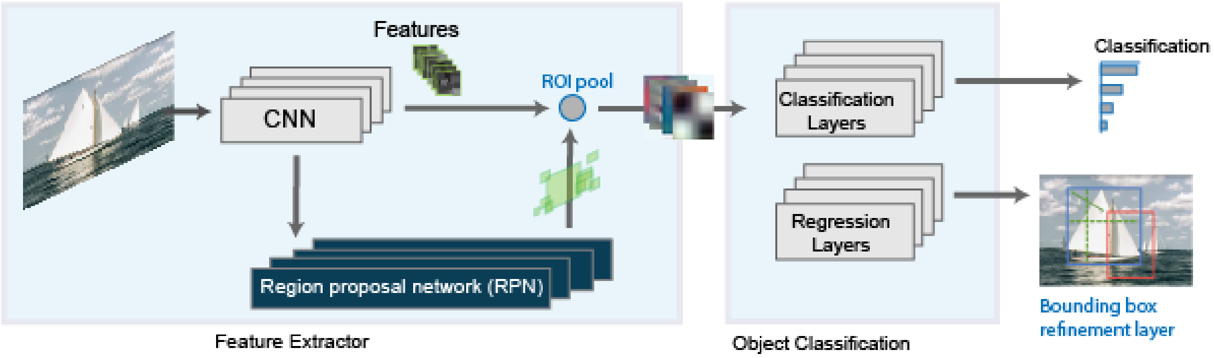


Figure 8: Faster RCNN for object detection

Introduction of Feature Pyramid Networks is recognized as one of the breakthroughs in object detectors. The RCNN networks were not able to capture smaller objects in the image. The FPN network used a simple image pyramid to scale image into different shapes and sends it to the network. Once the detections were detected on each scale predictions were combined using different methods. FPN also introduced multiple CNN layers making the networks deeper in nature and used a top-down architecture while holding onto lateral connections. As CNN forms feature pyramid through forward propagation, FPN shows advances for detecting objects of varying scale (Lin T.-Y. a., 2016).

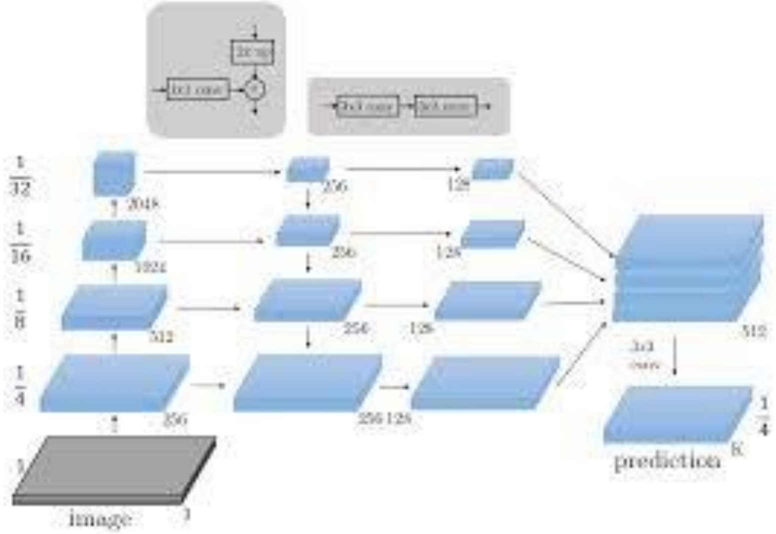


Figure 9: Illustration of a Feature Pyramid network used for object detection

2.2.3 Deep Learning based Object detection Framework (single stage detectors)

A novel approach to object detection was introduced with You only look once (YOLO) by (Redmon, 2015). All the previous models used Region networks or ROIs to localize and predict objects in the region. YOLO revolutionized by training on complete images, it optimizes its performance by checking on the probability of object appearing in a particular set of pixels. A single CNN simultaneously predicts multiple bounding boxes across all the classes. Its divides an image into nxn grid and checks for the probability of object in each grid, if the object is seen to have a high probability the following grid is responsible to localize the object. Each grid cell predicts a bounding box with a confidence scores. Another feature of YOLO is that it creates anchor boxes based on the training sets. It is a set of defined box sizes based on the object training set. Once a prediction is made it uses the anchor boxes to predict the confidence. It eliminates the ones with less confidence and retains the higher confidence boxes. It uses a non-maxima suppression (NMS) where an object has multiple bounding boxes overlap with each other and detect the same object.

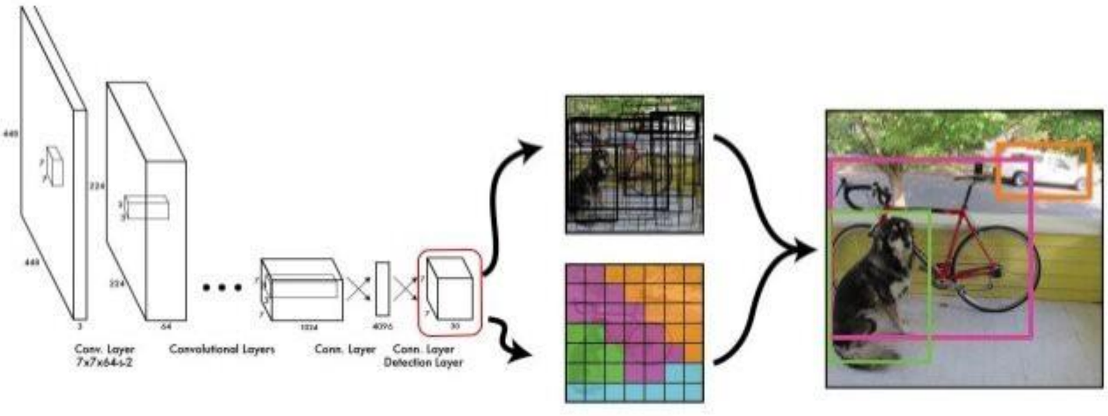


Figure 10: You Only Look Once (YOLO) for Object detection

Although YOLO was one of the fastest detection networks, it was not as precise as the RCNN networks. To improve the efficiency the author (Berg, 2016) introduced SSD: Single Shot Multibox Detector. The SSD introduced multiple reference multi resolution detection. It significantly increased the accuracy of object detection models. The notable increase in speed with mean average precision of 74.3% at 59fps for a 300x300 image and mean average precision of 76.9% for a 512x512 image surpassing the Faster RCNN results for the first time. SSD also introduced modality to object detection models with an interchangeable base model. The SSD has a base model to extract feature maps. In the paper author used VGG16 network. Other methods like VGG19 and Resnet can also be used for feature map extraction. Once the feature maps are extracted a series of convolution filters are added. These convolutional layers decrease the size of the image progressively allowing predictions at different scales. In the end a NMS layer is used to suppress and eliminate overlapping boxes.

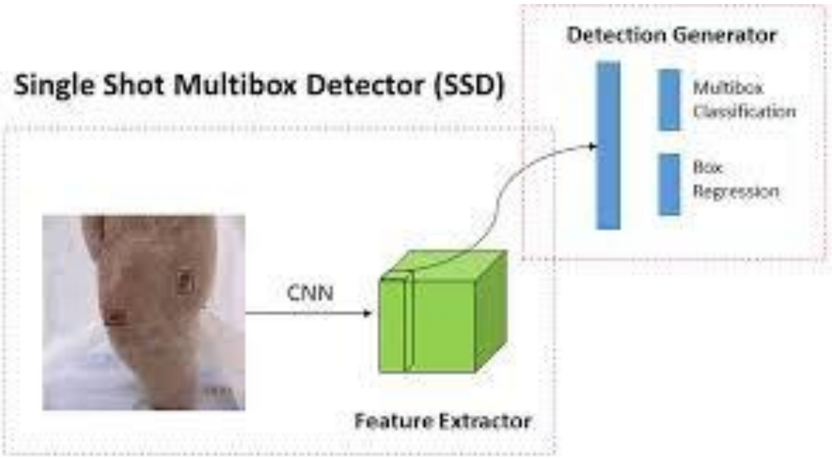


Figure 11: SSD: Single Shot Multibox Detector for Object detection

The problem of extreme class imbalance has been addressed by RetinaNet. This network introduces a new loss known as Focal Loss. The idea of Focal loss is that the easy negative samples are allowed to have lower loss to have the network detect and predict harder misclassification examples during training. Focal loss enables one-stage detectors to achieve comparable accuracies with the slower two stage detectors. Retina net uses FPN and Resnet as the backbone for feature extraction and includes class specific subnetworks for classification and bounding boxes. Retina net is considered as the state-of-the-art and outperforms two stage detectors like Faster RCNN (Lin T.-Y. a., 2017) (Borah, 2020).

2.3 Related Work

Previous methods and attempts to extract results included extracting results from a latex file. In this paper by (Mayank Singh, 2019) results were extracted from the source file in form of a tuple of the task, dataset, metric name, and metric value and stored in form of a leaderboard that can be accessed with the keywords in question. They try to extract information from the tables and values only and are not able to extract information from plots and images as they use a source file.

Our approach on the other hand is to detect all the results from paper that are presented in form of tables and plots.

Another interesting approach is presented by (Yufang Hou, 2019) as they extract textual information from the papers. They add a metric name, task, and dataset. They also use text interceptors to extract tabular information. They treat this as a Natural Language Processing (NLP) problem and use NLI and BERT-based approaches to extract the metrics information. In our approach, we treat it as a segmentation problem and an object detection problem to understand and extract information from the PDF files that are provided to the machine learning model.

Closer to our approach is the paper by (Kardas, (2020).), In this paper, they design and develop a machine learning pipeline to extract results and develop a leaderboard. The contrast is they use text from the paper to extract information and tables to extract information that is results. It also focuses on extracting the metadata from the information. Our approach is more straightforward as we treat it as an image and extract all the tables and plots individually for the user to evaluate. The complexity of this application is such that a user must understand and evaluate the context of the information presented in the paper.

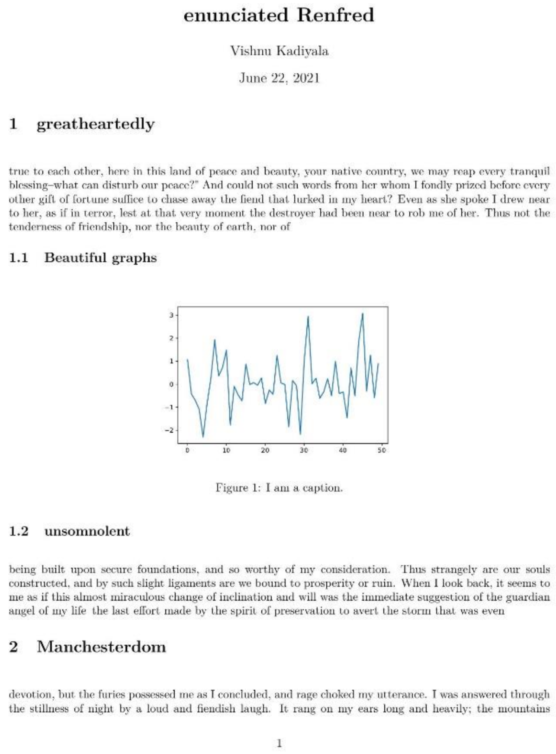
CHAPTER 3: METHODOLOGY

Data is an essential part of the Machine learning Field. In any Machine Learning model, the data is usually divided into training, validation, and test sets. The training set helps machines learn from the existing data and know the ground truth; the validation data is part of the training dataset but is not encountered by the learning algorithm. It helps in understanding the progress and direction of the machine learning model. The machine learning model never discovers the test dataset. It is used to determine the accuracy of the machine learning model on a completely different set of unseen data. The machine learning models that we build heavily rely on the data it sees in the training set. The weights and biases are optimized for the training set that it encounters. Hence, data generation plays a vital role in machine learning-based models, especially research. The way data has been generated will affect the way machine learns. The dataset generated for this research consists of PDF files generated using python. Each PDF file has been generated carefully considering the variability of the data on the page itself and the machine processing capability. The main objective from a data perspective is to teach machines to localize the tables and plots from PDFs. There has been a huge increase in the datasets available for problems like object detection and other machine learning models. The most widely used dataset, COCO consists of 10,000 images of 1792 different classes and some multiclass objects. For the scope of this research, the data does not need to include as many classes instead the data only needs to consist of tables and plots as the two different classes, and to add added complexity; we will use a name block as the third class.

3.1Data Generation

The main goal of dataset generation is to produce high-quality data. This includes the dataset being diverse and representing real-life scenarios. The paper submissions to the journals and conferences are strictly governed, and the quality of the papers is predictable as they follow a set of rules and flow. But at the same time, each individual paper has distinct characteristics in terms of the number of figures, tables, and plots present in the paper. Also, the location of the presented information can vary widely. This also shows that the variability that can be introduced is limited.

The data for this research, as mentioned above, is generated using a python library, PyLatex to generate tex files, and the tex files are then converted into PDFs. Generating the data was a massive task with python as we designed an Automator to create the tex files on a need basis, the Automator consisted of several components mainly, the font and size of the document and if it must include tables and/or plots in the PDF and for additional complexity the dimensions of the table and numbers of graphs in the plot. The data also includes text to emulate the real scenario papers. The text is produced by putting together all the words in English and choosing two words from them as the title and a verse from sekhsphere's novel to produce the text presented beside the table/plot. Figure 1 highlights a sample dataset that includes a name block and a plot. A sample image consists of multiple sections to produce more realistic papers. To reduce the bias the dataset has been produced in different partitions. Initially, a set of 1000 images were produced to include only the plot and the center. These images were annotated and fed into the network to check compatibility with the model. Once, the compatibility was established. A further 1000 images were produced with tables of similar shape and size. In the later productions, the variability of the dataset increased by varying the position and size of the table and plot. By including random omissions of plots and tables from the data. The PDF files are then converted into pages to individual jpeg files for annotation.



enunciated Renfred

Vishnu Kadiyala

June 22, 2021

1 greatheartedly

true to each other, here in this land of peace and beauty, your native country, we may reap every tranquil blessing–what can disturb our peace?" And could not such words from her whom I fondly prized before every other gift of fortune suffice to chase away the fiend that lurked in my heart? Even as she spoke I drew near to her, as if in terror, lest at that very moment the destroyer had been near to rob me of her. Thus not the tenderness of friendship, nor the beauty of earth, nor of

1.1 Beautiful graphs

Figure 1: I am a caption.

1.2 unsomnolent

being built upon secure foundations, and so worthy of my consideration. Thus strangely are our souls constructed, and by such slight ligaments are we bound to prosperity or ruin. When I look back, it seems to me as if this almost miraculous change of inclination and will was the immediate suggestion of the guardian angel of my life the last effort made by the spirit of preservation to avert the storm that was even

2 Manchesterdom

devotion, but the furies possessed me as I concluded, and rage choked my utterance. I was answered through the stillness of night by a loud and fiendish laugh. It rang on my ears long and heavily; the mountains

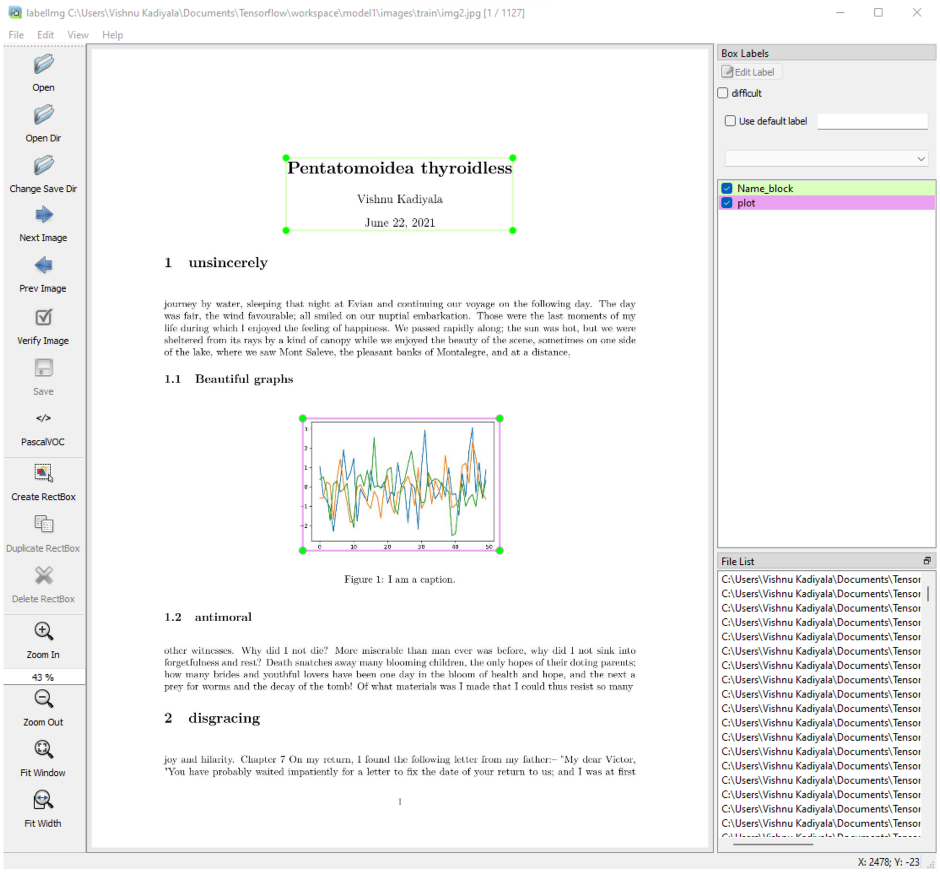
1

Figure 12: Sample Training Dataset Image

3.2 Annotation

Data annotation is a process of labeling data to show the outcome to a machine learning model to predict. It sets up the ground truth for predictions and deep learning models are evaluated based on how far they plot from the actual prediction. The loss function as discussed in the earlier chapter is solely based on the difference between annotated image and the predicted image. Data annotation reveals the features that the model uses to predict non-annotated data.

In this dataset, all the 10,000 images were manually annotated using LabelImg (Tzutalin, 2015). Labelimg is a graphical image annotation tool that uses python and Qt console for the interface. The annotated images generate an XML file with the coordinates for a bounding box and the corresponding label of the identified class. The process of annotation is to draw a bounding box on the object of interest and select a class it belongs to as shown in the figure. The following XML file will keep important metadata form the image and annotation. The annotation coordinates are described in terms of xmin, ymin, xmax and ymax representing the four corners of a rectangle surrounding the area of interest.


File
Edit
View
Help
Open
Open Dir
Change Save Dir
Next Image
Prev Image
Verify Image
Save
</>
PascalVOC
Create RectBox
Duplicate RectBox
Delete RectBox
Zoom In
43 %
Zoom Out
Fit Window
Fit Width
Box Labels
Edit Label
File List
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
C:\Users\Vishnu Kadiyala\Documents\Tensor
3
2
1
0
-1
-2
0
10
20
30
40
50

labelImg C:\Users\Vishnu Kadiyala\Documents\Tensorflow\workspace\model1\images\train\img2.jpg [1 / 1127]

□ difficult

□ Use default label

☑ Name_block
☑ plot

X: 2478; Y: -23

Pentatomoidea thyroidless

Vishnu Kadiyala

June 22, 2021

1 unsincerely

journey by water, sleeping that night at Evian and continuing our voyage on the following day. The day was fair, the wind favourable; all smiled on our nuptial embarkation. Those were the last moments of my life during which I enjoyed the feeling of happiness. We passed rapidly along; the sun was hot, but we were sheltered from its rays by a kind of canopy while we enjoyed the beauty of the scene, sometimes on one side of the lake, where we saw Mont Saleve, the pleasant banks of Montalegre, and at a distance,

1.1 Beautiful graphs

Figure 1: I am a caption.

1.2 antimoral

other witnesses. Why did I not die? More miserable than man ever was before, why did I not sink into forgetfulness and rest? Death snatches away many blooming children, the only hopes of their doting parents; how many brides and youthful lovers have been one day in the bloom of health and hope, and the next a prey for worms and the decay of the tomb! Of what materials was I made that I could thus resist so many

2 disgracing

joy and hilarity. Chapter 7 On my return, I found the following letter from my father:– "My dear Victor, "You have probably waited impatiently for a letter to fix the date of your return to us; and I was at first

1

Figure 13: Sample Annotation using LabelImg Annotation Software on generated Dataset File

3.3 Experimental Setup

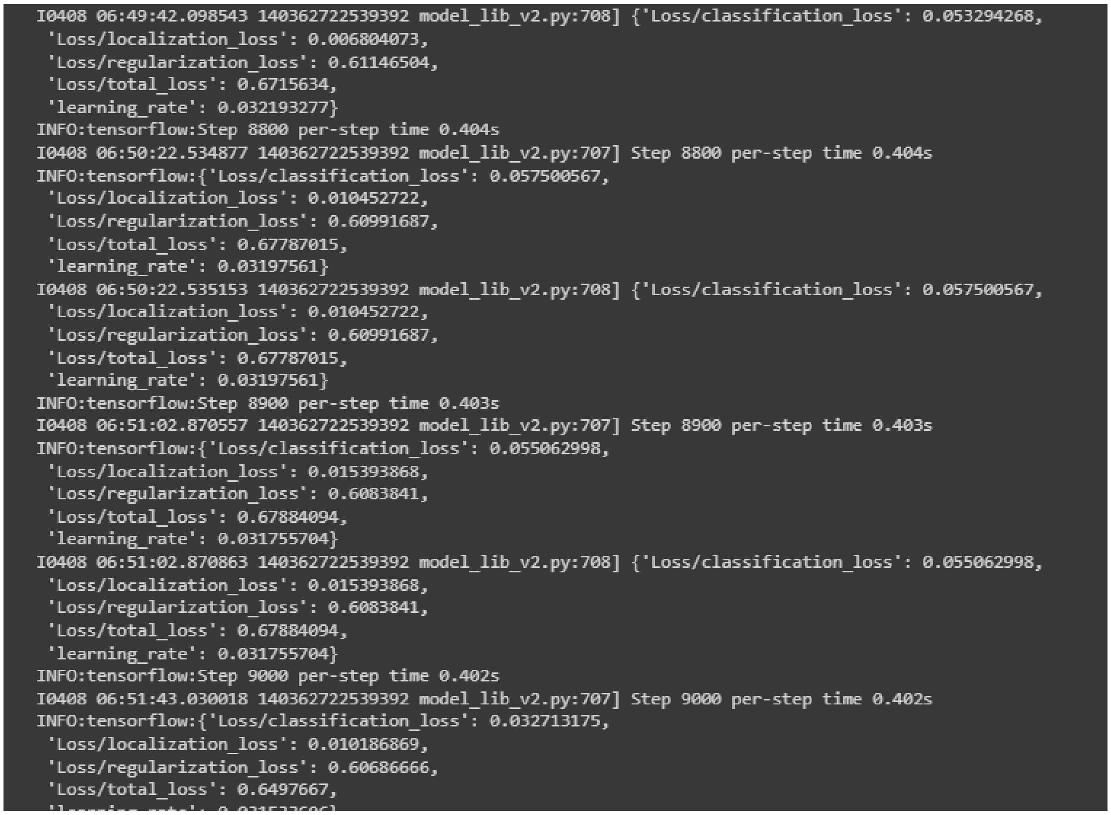
To use annotated data with the model we need to convert and parse the XML files into readable CSV files. This CSV file is a consolidation of all the training and test set examples with their bounding box information present with it, now the data is prepared and ready for training with the models. In this research, we used pre-trained models from TensorFlow Object Detection API (Martín Abadi, 2015). We specifically chose the models that have been pre-trained on some of the famous image datasets like CIFAR10 and COCO to evaluate them on our custom dataset. We used Mobile Net, RetinaNet and CenterNet for comparison.

The models were set up in Google Colab’s Jupyter notebook style coding. The models were trained with Colab GPUs having approximately 16GB VRAM and storage from google drive. The whole dataset consumed about 15GB of storage space while the model and its checkpoints consumed about 100GB of storage on the google drive. This is mainly due to the multiple checkpoints that the models generate as they train to refer to if the model starts overfitting. The pipeline configuration files are generated model specific and include key details about the model except the architectural details. The Architectural details are included in the pre-trained model itself. The pipeline configuration includes information about the batch and file size, it’s location, resolution of the image and also allows us to customize the hyper parameters of the model.

Configurations/Models	**MobileNet**	**RetinaNet**	**CenterNet**
Batch Size	32/epoch	64/epoch	64/epoch
Image resizing to	640x640	512x512	512 x 512
optimizer	adam	adam	adam
Initial learning rate	0.001	0.0001	0.001
Max number of prediction boxes	100	100	100
Focal Loss	False	True	False

Table1*: Model configuration comparison for pre-trained model evaluation.*

Models in Table 1 were trained for 10,000 epochs with the mentioned configuration. The learning rate of the model is starting at an extremely low range because the model already has a set of weights and biases. The objective of training is only to fine-tune the models to achieve desired results. The models were trained using TensorFlow and generate checkpoints every 100 epochs providing the ability to choose finest changes. All the models were tried to train with similar parameters to avoid biases due to setup. Such as training batch size and same optimizer adam. We chose a different learning rate for the resnet architecture as per the suggestion that it would converge with a higher learning rate. All the learning rates are adjusted during the training by the models. The differences visible in the table are mostly based on the architectural changes. The SSD models do include an input for focal loss Although, we turned it off during training to micmic literature review version of SSD. It will essentially be considered as a RetinaNet if the Focal loss were to be included. The image size is different for the initial SSD as it used mobilenet for its feature extraction and it is a feature of mobilenet providing the ability to generate features for a larger image size without impact on the time consumed in feature map production.



```
I0408 06:49:42.098543 140362722539392 model_lib_v2.py:708] {'Loss/classification_loss': 0.053294268,
 'Loss/localization_loss': 0.006804073,
 'Loss/regularization_loss': 0.61146504,
 'Loss/total_loss': 0.6715634,
 'learning_rate': 0.032193277}
INFO:tensorflow:Step 8800 per-step time 0.404s
I0408 06:50:22.534877 140362722539392 model_lib_v2.py:707] Step 8800 per-step time 0.404s
INFO:tensorflow:{'Loss/classification_loss': 0.057500567,
 'Loss/localization_loss': 0.010452722,
 'Loss/regularization_loss': 0.60991687,
 'Loss/total_loss': 0.67787015,
 'learning_rate': 0.03197561}
I0408 06:50:22.535153 140362722539392 model_lib_v2.py:708] {'Loss/classification_loss': 0.057500567,
 'Loss/localization_loss': 0.010452722,
 'Loss/regularization_loss': 0.60991687,
 'Loss/total_loss': 0.67787015,
 'learning_rate': 0.03197561}
INFO:tensorflow:Step 8900 per-step time 0.403s
I0408 06:51:02.870557 140362722539392 model_lib_v2.py:707] Step 8900 per-step time 0.403s
INFO:tensorflow:{'Loss/classification_loss': 0.055062998,
 'Loss/localization_loss': 0.015393868,
 'Loss/regularization_loss': 0.6083841,
 'Loss/total_loss': 0.67884094,
 'learning_rate': 0.031755704}
I0408 06:51:02.870863 140362722539392 model_lib_v2.py:708] {'Loss/classification_loss': 0.055062998,
 'Loss/localization_loss': 0.015393868,
 'Loss/regularization_loss': 0.6083841,
 'Loss/total_loss': 0.67884094,
 'learning_rate': 0.031755704}
INFO:tensorflow:Step 9000 per-step time 0.402s
I0408 06:51:43.030018 140362722539392 model_lib_v2.py:707] Step 9000 per-step time 0.402s
INFO:tensorflow:{'Loss/classification_loss': 0.032713175,
 'Loss/localization_loss': 0.010186869,
 'Loss/regularization_loss': 0.60686666,
 'Loss/total_loss': 0.6497667,
 'learning_rate': 0.031533606}
```

Figure 14: Learning window for a Retina Net Network at 9000 epochs

CHAPTER 4: EXPERIMENTAL RESULTS

4.1 Metrics of Evaluation

Object detection considers accuracies as one of the model metric, but the accuracies of these models are exceedingly high and thus, we should also consider other metrics for evaluating these models. Precision and Recall play a vital role in evaluating the model. Precision gives the proportion of positive example identifications are correct meaning, how accurate our predictions are and Recall gives the proportion of actual positives identified correctly meaning, how good we are able to find the positive examples.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

Where,

- TP = True positive
- FN = False Negative
- FP = False Positive

4.1.1 Intersection over Union (IoU)

Intersection over union is an evaluation metric for object detection models. We need the ground truth bounding boxes from the testing data and predicted bounding box from our model. The IoU metric is simply a ratio between area of overlap over the area of union.

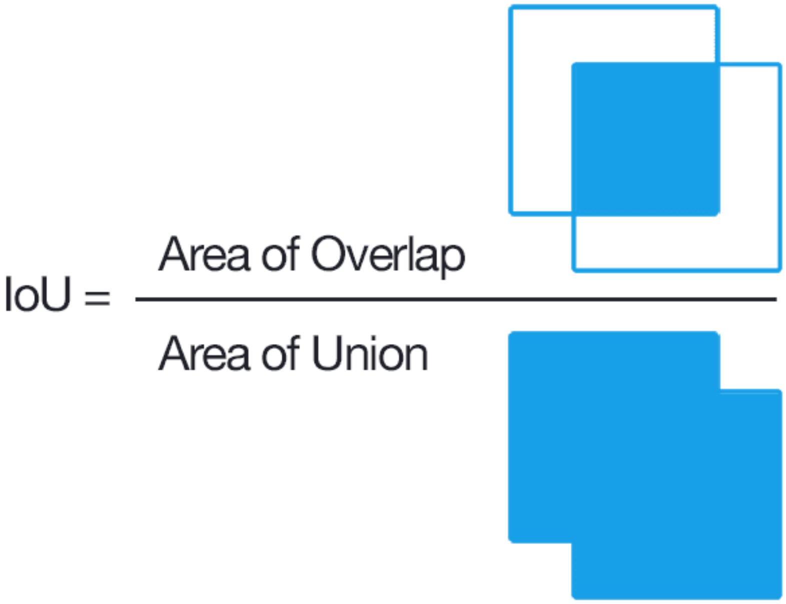


Figure 15: Intersection over Union Representation and Formula Evaluation

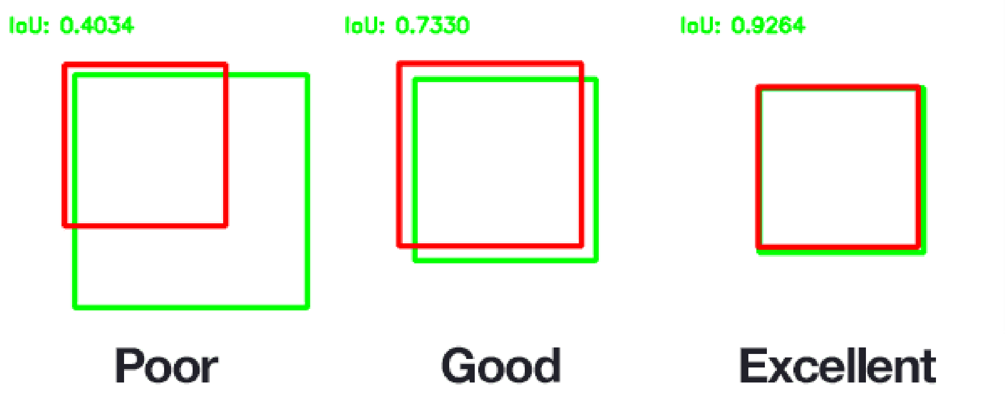


Figure 16: Intersection over Union Example for Poor, Good and Excellent Prediction

4.1.2 Mean Average Precision(mAP)

Mean Average Precision compares the ground truth bounding box to detected bounding box and calculates a score. The average precision is a summary of precision and recall. It is a weighted sum of precisions at each threshold where the weight is increase in recall.

$$AP = \sum_{k=0}^{k=t-1} [recalls(k) - recalls(k+1)] * Precisions(\,k) \qquad (5)$$

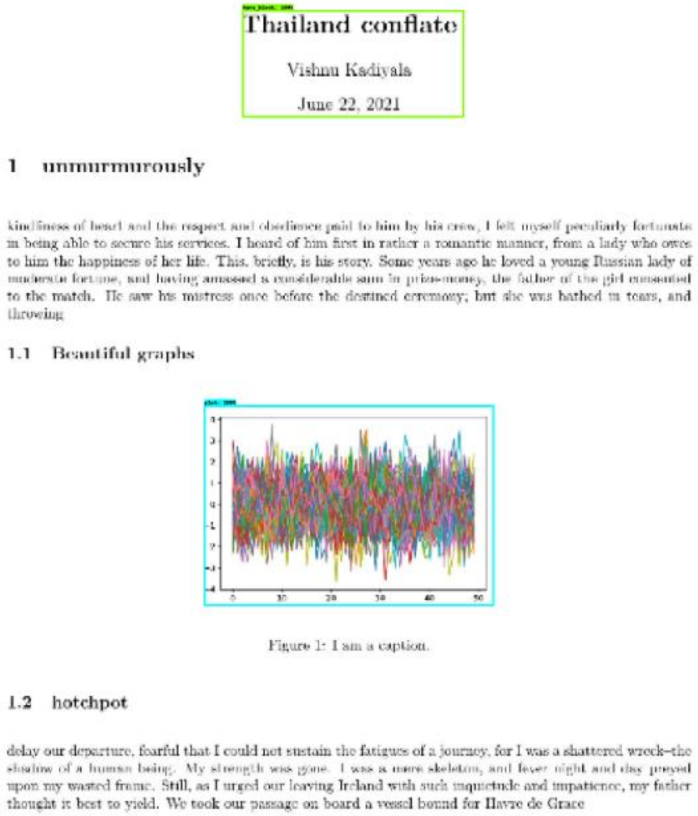
$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_{k^1} \qquad (6)$$

Where,

- AP_k = the Average Precision of Class k
- n = Number of classes
- t = number of thresholds
- Recalls(n) = 0
- Precisions(n) = 1

4.2 Single Shot multibox Detector 1 – MobileNet

Following the training process of the algorithm, the prediction performed by SSD-1 model achieved reliable results in predicting the bounding box for test examples. It was able to detect the name block with a confidence of 85% and plot with a confidence of 100%. The SSD-1 Network was the fastest to train over 10000 epochs. It took over 1h 30mins to perform all the epochs over a batch size of 32/epoch. Figure 17 shows that the model was able to extract features from the network and predict the bounding boxes accurately.



Thailand conflate

Vishnu Kadiyala

June 22, 2021

1 unmurmurously

kindliness of heart and the respect and obedience paid to him by his crew, I felt myself peculiarly fortunate in being able to secure his services. I heard of him first in rather a romantic manner, from a lady who owes to him the happiness of her life. This, briefly, is his story. Some years ago he loved a young Russian lady of moderate fortune, and having amassed a considerable sum in prize-money, the father of the girl consented to the match. He saw his mistress once before the destined ceremony; but she was bathed in tears, and throwing

1.1 Beautiful graphs

Figure 1: I am a caption.

1.2 hotchpot

delay our departure, fearful that I could not sustain the fatigues of a journey, for I was a shattered wreck–the shadow of a human being. My strength was gone. I was a mere skeleton, and fever night and day preyed upon my wasted frame. Still, as I urged our leaving Ireland with such inquietude and impatience, my father thought it best to yield. We took our passage on board a vessel bound for Havre de Grace

Figure 17: Accurate prediction of Plot and name block for SSD1 – mobilenet

The following Figure 18 highlights the classification loss of the SSD-1 over the 10000 epochs. The loss started at about 0.1 and quickly dived down to about 0.46 over 3000 epochs. The classification loss bifurcates between different classes present in the training set. In this case it had the ability to differentiate between plots, tables and Name blocks accurately. Figure 19 shows the localization loss. The localization loss is how accurately the model is able to find the bounding box of the image and the loss dived down to about 0.01 at approximately 6000 epochs.

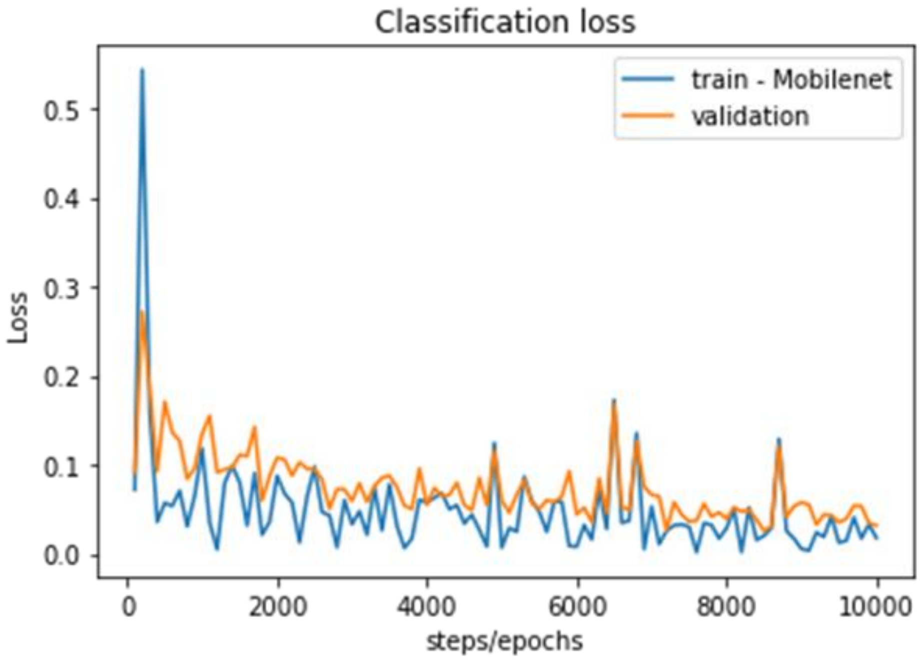


Figure 18: Classification loss over 10,000 epochs for SSD1- Mobilenet

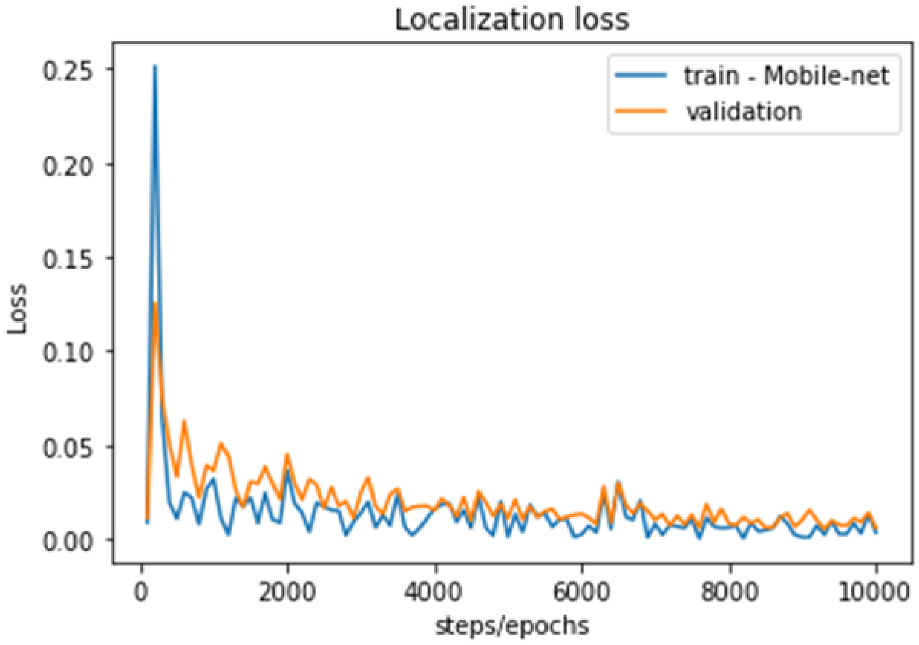
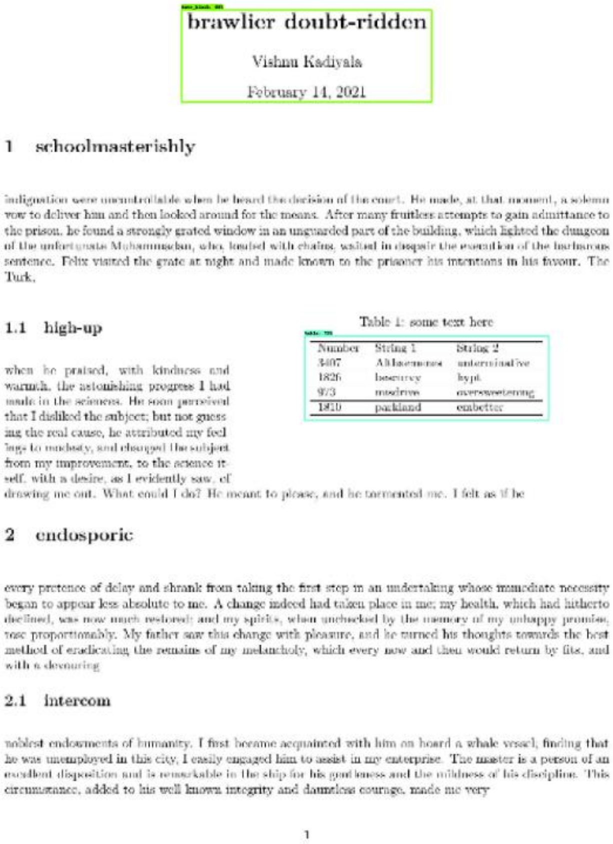


Figure 19: Localization loss over 10,000 epochs for SSD1 - Mobilenet

4.3 Single Shot multibox Detector 2 – RetinaNet

Following the training process, RetinaNet was able to localize the name block, plot and the table on a test example. It was able to detect the name block with a high confidence of about 90% and table with a confidence of 60%. The SSD-2 Network trained for about 2hr 30mins through 10000 epochs over a batch size of 64/epoch. Figure 20 showcases the ability of RetinaNet to localize and classify all the classes effectively.



brawlier doubt-ridden

Vishnu Kadiyala

February 14, 2021

1 schoolmasterishly

indignation were uncontrollable when he heard the decision of the court. He made, at that moment, a solemn vow to deliver him and then looked around for the means. After many fruitless attempts to gain admittance to the prison, he found a strongly grated window in an unguarded part of the building, which lighted the dungeon of the unfortunate Muhammadan, who, loaded with chains, waited in despair the execution of the barbarous sentence. Felix visited the grate at night and made known to the prisoner his intentions in his favour. The Turk,

1.1 high-up

when he praised, with kindness and warmth, the astonishing progress I had made in the sciences. He soon perceived that I disliked the subject; but not guessing the real cause, he attributed my feelings to modesty, and changed the subject from my improvement, to the science itself, with a desire, as I evidently saw, of drawing me out. What could I do? He meant to please, and he tormented me. I felt as if he

Table 1: some text here

Number	String 1	String 2
3407	Althaemenes	unterminative
1826	beseurey	hypt
973	mudrive	oversweetening
1810	parkland	embetter

2 endosporic

every pretence of delay and shrank from taking the first step in an undertaking whose immediate necessity began to appear less absolute to me. A change indeed had taken place in me; my health, which had hitherto declined, was now much restored; and my spirits, when unchecked by the memory of my unhappy promise, rose proportionably. My father saw this change with pleasure, and he turned his thoughts towards the best method of eradicating the remains of my melancholy, which every now and then would return by fits, and with a devouring

2.1 intercom

noblest endowments of humanity. I first became acquainted with him on board a whale vessel; finding that he was unemployed in this city, I easily engaged him to assist in my enterprise. The master is a person of an excellent disposition and is remarkable in the ship for his gentleness and the mildness of his discipline. This circumstance, added to his well known integrity and dauntless courage, made me very

1

Figure 20: Accurate Prediction of Table and Nameblock on SSD2 – RetinaNet

The following Figure 21 highlights the Object center loss of the SSD-2 over the 10000 epochs. The loss started at about 0.2 and quickly dived down to about 0.06 over 10000 epochs. The Object center loss is the ability of the model to find the center of the bounding box in the training set. In this case it had the ability to find the center of the plots, tables and Name blocks accurately. Figure 22 shows the box scale loss. The box scale loss is model's ability to predict size of the object using some predefined bounding boxes and loss dived down to about 0.20 at approximately 3800 epochs.

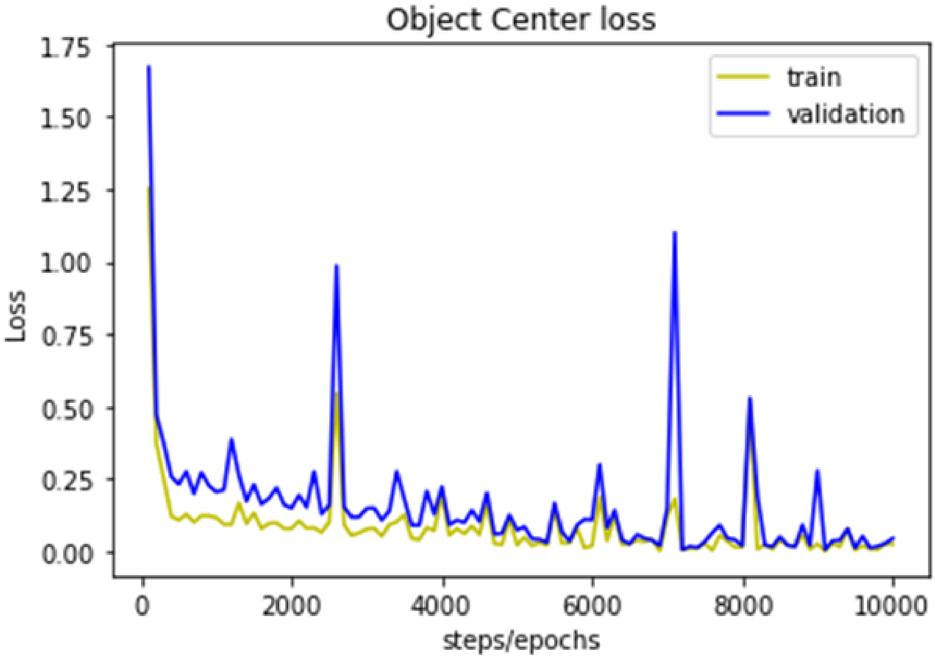


Figure 21: RetinaNet Loss for Object Center over 10000 epochs

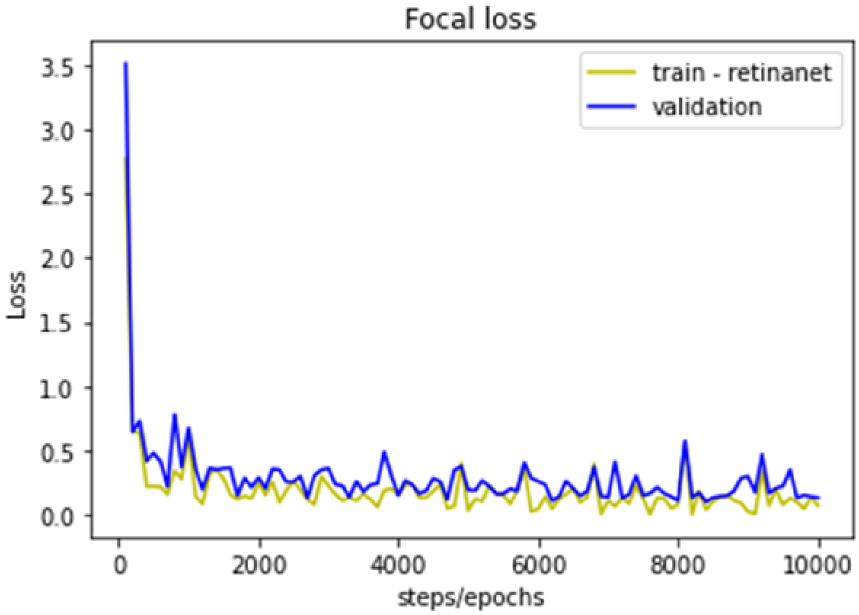
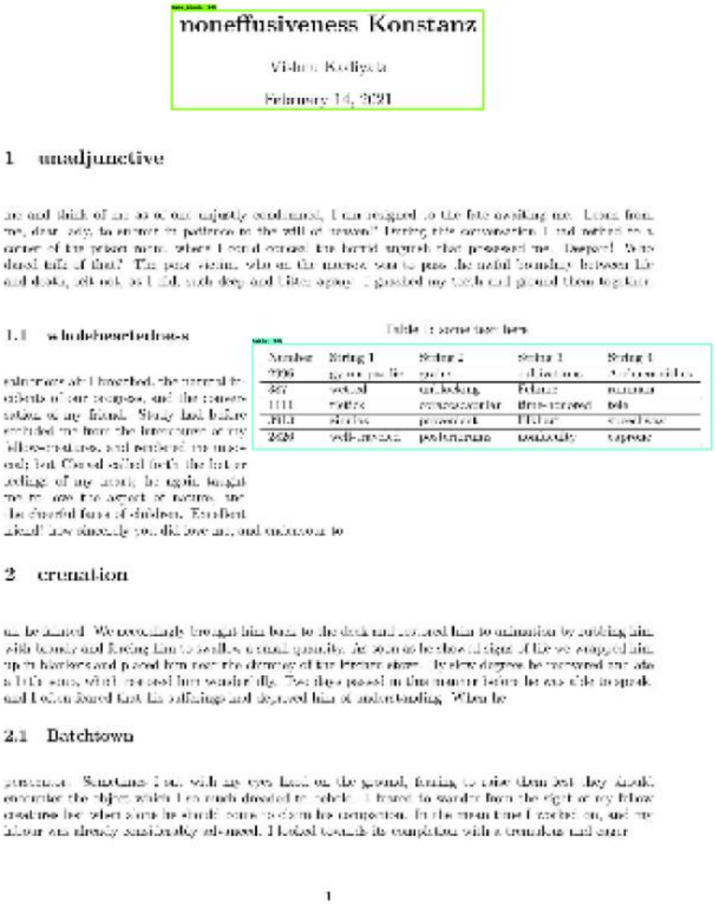


Figure 22: RetinaNet Loss for Box scale over 10000 epochs

4.4 CenterNet

CenterNet is the final model we used to Localize the information from PDF files. It had the best result in the whole bunch with a great confidence reaching to an almost 100% each time. In Figure 23, The model is able to recognize different objects and it is able to classify and localize tables, plots and name blocks.



noneffusiveness Konstanz

Vishnu Kodiyala

February 14, 2021

1 unadjunctive

me and think of me as of one unjustly condemned, I am resigned to the fate awaiting me. Learn from me, dear lady, to submit in patience to the will of heaven!" During this conversation I had retired to a corner of the prison room, where I could conceal the horrid anguish that possessed me. Despair! Who dared talk of that? The poor victim, who on the morrow was to pass the awful boundary between life and death, felt not, as I did, such deep and bitter agony. I gnashed my teeth and ground them together

1.1 wholeheartedness

salubrious air I breathed, the natural incidents of our progress, and the conversation of my friend. Study had before secluded me from the intercourse of my fellow-creatures, and rendered me unsocial; but Clerval called forth the better feelings of my heart; he again taught me to love the aspect of nature, and the cheerful faces of children. Excellent friend! how sincerely you did love me, and endeavour to

Table 1: some text here

Number	String 1	String 2	String 3	String 4
7706	[illegible]	[illegible]	[illegible]	[illegible]
387	wetted	unlocking	Felucca	[illegible]
1111	[illegible]	[illegible]	three-storeyed	bola
3013	[illegible]	[illegible]	[illegible]	[illegible]
2420	well-[illegible]	[illegible]	[illegible]	[illegible]

2 crenation

as he fainted. We accordingly brought him back to the deck and restored him to animation by rubbing him with brandy and forcing him to swallow a small quantity. As soon as he showed signs of life we wrapped him up in blankets and placed him near the chimney of the kitchen stove. By slow degrees he recovered and ate a little soup, which restored him wonderfully. Two days passed in this manner before he was able to speak, and I often feared that his sufferings had deprived him of understanding. When he

2.1 Datchtown

persecutor. Sometimes I sat with my eyes fixed on the ground, fearing to raise them lest they should encounter the object which I so much dreaded to behold. I feared to wander from the sight of my fellow creatures lest when alone he should come to claim his companion. In the mean time I worked on, and my labour was already considerably advanced. I looked towards its completion with a tremulous and eager

1

Figure 23:Accurate Prediction of table and Name block by CenterNet

The following Figure 24 highlights the Object center loss of the CenterNet over the 10000 epochs. The loss started at about 0.5 and quickly dived down to about 0.15 over 5000 epochs. The Object center loss is the ability of the model to find the center of the bounding box in the training set. In this case it had the ability to find the center of the plots, tables and Name blocks accurately. Figure

25 shows the object center loss. The object center loss is model's ability to find the center of object while predicting bounding boxes and loss started less than 0.1 and stayed pretty stable with some high notes around the end of learning.

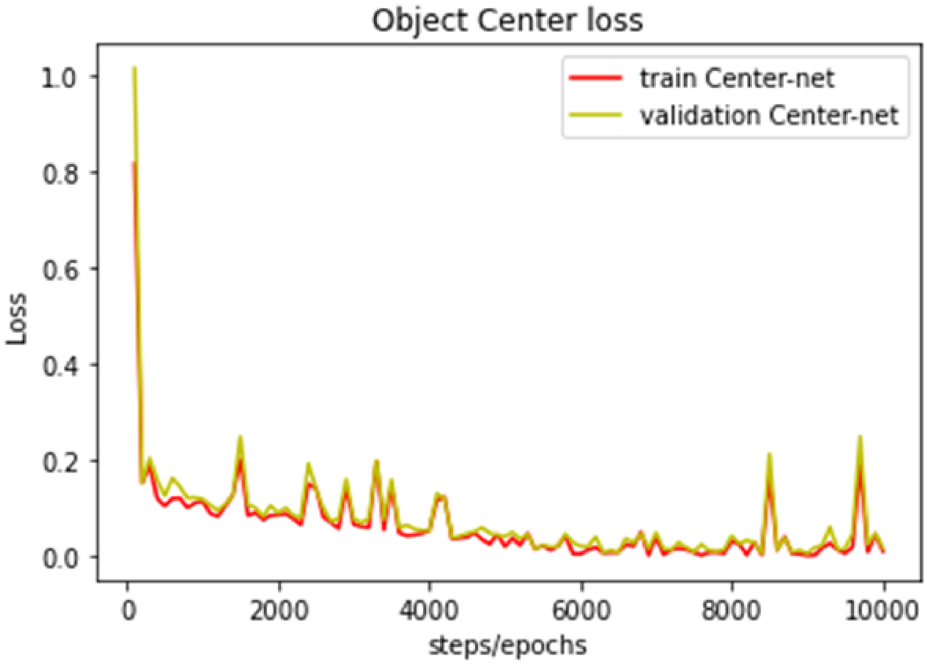


Figure 24: CenterNet Loss for box scale over 10000 epochs

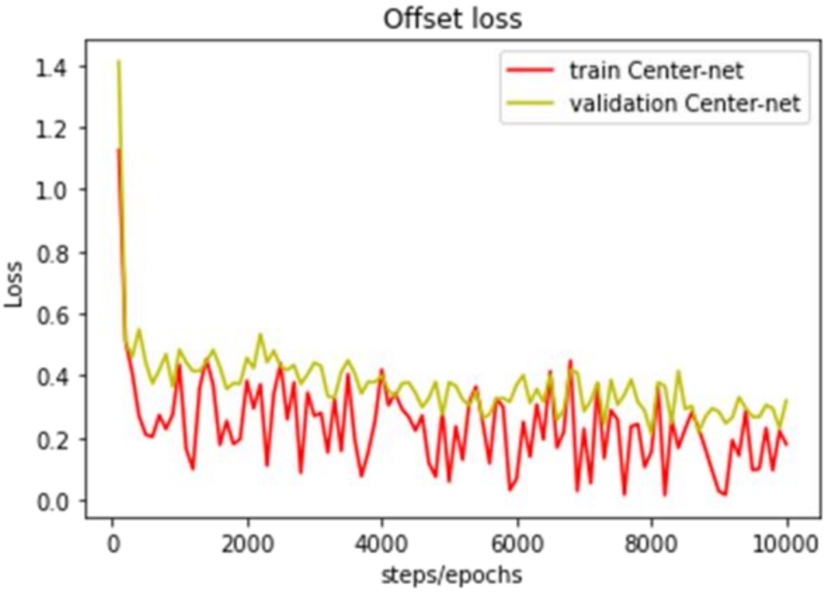


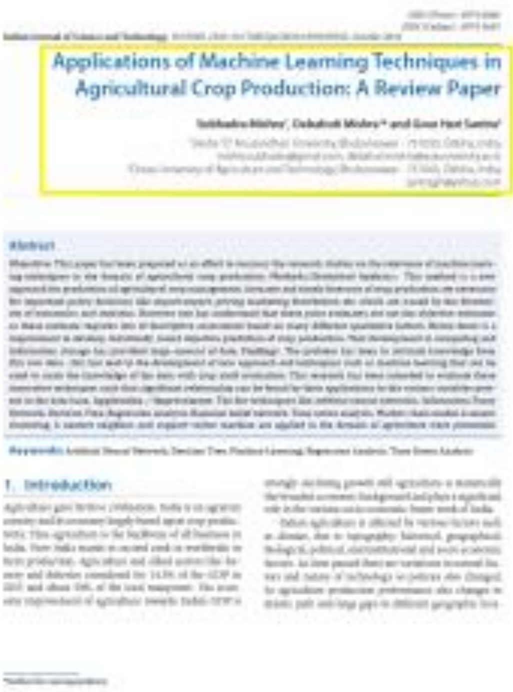
Figure 25: CenterNet Loss for object center over 10000 epochs

The plots shown in figure 26 represent total loss of all 3 models during the learning phase and how they compare with each other. The total loss for SSD-1 reduced at a slower pace as compared to SSD-2. The Retina Net had the best learning overall with the total loss going to as low as 0.5 over the 10000 epochs.

Table 2 presents evaluation of the model over test data. The mAP represents Mean Average precision and SSD-1 Mobilenet achieved about 89.04 while SSD-2 achieved mAP of 91.1 and CenterNet about 92. The mAP over 50 IoU was very accurate for all the models and mAP over 75 IoU varied a little with CenterNet scoring the highest and SSD-1 Mobilenet scoring the lowest. AR represents average recall over the testing dataset. The model checks for Average recall over 1 epoch, 10 epochs and 100 epochs to get an accurate understanding at each possible point. The AR for all the models remained about the same and not vary with increase in the number of epochs.

Metrics\Models	**MobileNet**	**RetinaNet**	**CenterNet**
mAP	89.04	91.1	92
mAP @ 50 IoU	1	1	1
mAP @ 75 IoU	99.17	99.66	99.67
AR @ 1	91.51	93.76	93.86
AR @ 10	91.54	93.76	93.88
AR @ 100	91.54	93.8	93.88

Table 2: *Mean Average Precision and Average Recall for all the models*



Applications of Machine Learning Techniques in Agricultural Crop Production: A Review Paper

Subhadra Mishra[1], Debahuti Mishra[1]* and Gour Hari Santra[2]

Abstract

[illegible]

Keywords: [illegible]

1. Introduction

[illegible]

Figure 26: Accurate Prediction on a PDF file downloaded from internet

P-test performed on the output results i.e, mAP and AR show the following results. We divided the data into 2 parts. Each part has 100 values, and each value is mAP score of CenterNet over 20 test images. We build a hypothesis that there is no change in the mean over 20 images and the mAP over each sub-sample is not significant. performing t-test on the data reveals that the mean of group A is 91.3812 and the mean of group B is 91.68. standard deviation of group A is 0.31245 and group B is 0.40697. Variance of group A is 0.097625 and group B is 0.165625. Calculating the t-value using the above details is 2.026. going through the table we can see that for the degrees of freedom over 120 samples is significant and falls between p(0.05) and p(0.025). Hence, we reject the null hypothesis and suggests that there is significance in the results.

CHAPTER 5: ANALYSIS and DISCUSSION

The models that we are evaluating to localize information from PDF files are performing accurately. However, the three models differ in the way they approach to the solution and predict the boxes. The plots presented in the results section highlight different trends. Figures Include total training loss plots for all the models. It is interesting to note that the MobileNet is converging slower as compared to resnet RetinaNet and CenterNet networks. The CenterNet model converges fastest compared to the SSD based models, this is mainly due to the Center Net’s Hourglass approach. This shows that the models are very adaptive and can be customized to new datasets within five to six thousand epochs.

Table 2 reveals an especially important insight that at 50% IoU all the models are at a 100% accurate. The mAP scores for all the models lie around 90, with the least being mobilenet mainly due to its convergence, it carries a higher loss. The RetinaNet has an mAP of 91.1 and CenterNet has the highest mAP of 91.1 at a 75% IoU all the models have an excess of 99% accuracy. With the reduction in batch size for mobilenet due to its power-hungry approach it seems that mobilenet is not the best model suitable for the application. With change in mAP scores across different IoUs AR changes almost negligibly. It reveals that all the 3 models have a good fit on training data and rules out the possibility of overfitting on the training data.

Another interesting observation is that the confidence level of predictions on the table and name blocks are low sometimes, revealing that that data generated has low bias and more variance, It is not an ideal dataset to learn on although, it is good one to draw different observations and analysis from it. If the data was trivial the mAP and AR would always be 1.

Loss plots for individual models reveal how quickly the model is able to predict on the new dataset. The classification loss for mobilenet in Figure 18 has an increase in classification loss around step 6000. It reveals that the mobilenet model was not able to differentiate between different classes. This also suggests that the mobilenet with its high accuracy is a great model to detect multiple objects in an image but it falls short of other models in classifying different object types. The Localization for SSD-1 mobilenet also increase around the same time step as classification loss as the model couldn't predict the center of the object, It can also be due to introduction of new type of data such as a change in data from tables to plots coupled with mobilenet SSD's slower convergence rate.

Loss plots for SSD-2 Resnet and RetinaNet are directly comparable in terms of type of loss itself. Comparing Figure 21 with Figure 18, Resnet's advantage over mobilenet is seen in the box size prediction and at the 8000th step there is a slight increase or a small buldge representing the same change in data observed in mobilenet. The Localization loss varies a lot in retinanet although, the overall loss itself is too low, this is essentially helpful in the overcoming a situation when the model is stuck in a local minima.

Loss plots for Centernet reveal that it is performing better than the SSD based models including the convergence time, mAP and AR. Although, CenterNet takes longer to perform predictions. To summarize it, The mobile net and resnet SSD models perform a good enough job in a time efficient manner to localize. The Retina Net does a better job at prediction but is slower as compared to the SSD based models.

The Object detection models were able to localize material from scientific literature. Compared to (Kardas, (2020).) our model can be trained to localize many formats of data. Our method falls short in ranking method and requires some manual intervention to analyze the results exported from the scientific literature.

CHAPTER 6: CONCLUSION

6.1 Summary

In this research we found that in the era of exponentially evolving and constantly innovating world, the number of publications has skyrocketed. The object detection models that we evaluated can be used to localize and extract information from PDFs and be presented to a user. We evaluated three different object detection models built in TensorFlow on a custom generated and annotated dataset of PDF files.

In Chapter 2, We provided a background for Deep learning and evolution of Object detection. We also provided related work notably by (Kardas, (2020).) and (LeCun) and the ranking system cell recognition based approach to detect results from a paper.

In Chapter 3, We described the dataset generation process using an Automator and pyLatex script for python to develop tex files. We also discussed the annotation and setup that is used to perform this research.

In Chapter 4, We showed all the plots that represent different loss functions generated by object detection models SSD-1 MobileNet, SSD-2 RetinaNet and CenterNet to understand the learning process of Deep neural networks on our custom generated dataset.

In Chapter 5, We discussed that the three model results, showed various levels of convergence to the dataset. The CenterNet converging the quickest followed by SSD-2 RetinaNet and finally the SSD-1 MobileNet. also assuring the quality of data generation with low bias and high variance.

We also discussed Mean Average Precision (mAP) and Average Recall (AR) that were used as metrics to evaluate the three models. With SSD-1 MobileNet scoring an mAP score of 89.04 and AR at 100 epochs of 91.54, SSD-2 RetinaNet with mAP score of 91.1 and AR at 100 epochs of 93.8 finally, the CenterNet with the highest mAP score of 92 and highest AR at 100 epochs of 93.88. Finally concluding that CenterNet is the best model to localize and extract information from the scientific literature as we are not time constrained and can be achieved eventually.

6.2 Future Scope

We would like to cover two possible extensions to this work. Firstly, we would like to automate the review process of the results by including a form of leaderboard for scientific literature based on relevancy and impact of the research. Secondly, we would like to add more variability to the data generation process by using General Adversarial Networks (GANs) and automate the annotation process to save a lot of time.

Bibliography

Alpaydin, E. (2021). *Machine learning.* MIT Press.

Berg, W. L.-Y. (2016). {SSD}: Single Shot {MultiBox} Detector. *Springer International Publishing,* doi: 10.1007/978-3-319-46448-0_2.

Borah, C. (2020, Nov 1). *Evolution of Object Detection*. Retrieved from Analytics Vidhya: https://medium.com/analytics-vidhya/evolution-of-object-detection-582259d2aa9b

Girshick, R. (2015). Fast R-CNN. *arXiv*, 10.48550/ARXIV.1504.08083 https://doi.org/10.48550/arxiv.1504.08083.

Girshick, R. a. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv*, https://doi.org/10.48550/arxiv.1311.2524, https://arxiv.org/abs/1311.2524 doi:10.48550/ARXIV.1311.2524.

Haralick, R. M. (n.d.). Image segmentation techniques. . *Computer vision, graphics, and image processing, 29(1), 100-132.*

Jones, P. V. (2001). Rapid object detection using a boosted cascade of simple features.

Kaiming He, X. Z. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *CoRR*, abs/1406.4729 http://arxiv.org/abs/1406.4729.

Kardas, M. C. ((2020).). Axcell: Automatic extraction of results from machine learning papers. *arXiv preprint arXiv:2004.14356.*

Krizhevsky, A. S. (n.d.). ImageNet classification with deep convolutional neural networks. *NIPS (pp. 1097–1105).*

LeCun, Y. B. (n.d.). Gradient based learning applied to document recognition. . *Proceedings of the IEEE, 86(11), 2278–2324.*

Lin, T.-Y. a. (2016). Feature Pyramid Networks for Object Detection}. *arXiv*, https://doi.org/10.48550/arxiv.1612.03144.

Lin, T.-Y. a. (2017). Focal Loss for Dense Object Detection. *arXiv*, https://doi.org/10.48550/arxiv.1708.02002.

M.W Gardner, S. D. (n.d.). Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,. *Atmospheric Environment,*, 2627-2636.

Mayank Singh, R. S. (2019). Automated early leaderboard generation from comparative tables. InAdvances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019. *Springer*, 244-257.

Mitchell, T. (2006). The discipline of machine learning. *Machine Learning Department technical report CMU-ML-06-108, Carnegie Mellon*.

Oludare Isaac Abiodun, A. J. (2018). State-of-the-art in artificial neural network applications. *Heliyon,*, Volume 4, Issue 11,.

Pedro F. Felzenszwalb, R. B. (2008). Object Detection with Discriminatively Trained.

Prasad, D. K. (2013). Object Detection in Real Images. *arXiv: Computer Vision and Pattern Recognition*. Retrieved 4 10, 2022, from https://arxiv.org/abs/1302.5189

Redmon, J. a. (2015). You Only Look Once: Unified, Real-Time Object Detection. *arXiv*, https://doi.org/10.48550/arxiv.1506.02640.

Ren, S. a. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv*, https://doi.org/10.48550/arxiv.1506.01497.

Rosenfeld, A. (n.d.). Computer vision: basic principles. *Proceedings of the IEEE, 76(8), 863-868.*

Ross B. Girshick Forrest, N. I. (2014). Deformable Part Models are Convolutional Neural Networks}. *CoRR*, abs/1409.5403.

Ruder, S. (2018). Tracking the Progress in Natural Language Processing.

Simpson, P. K. (n.d.). Foundations of neural networks.

Tautz, D. (. (n.d.). Segmentation. Developmental cell, 7(3), 301-312.

Triggs, N. D. (2005). Histograms of oriented gradients for human detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005*, , pp. 886-893 vol. 1, doi: 10.1109/CVPR.2005.177.

Wason, R. (2018). Deep learning: Evolution and expansion. 701-708.

Wu, X. S. ((2020).). Recent advances in deep learning for object detection. . *Neurocomputing, 396, 39-64.*

Yufang Hou, C. J. (2019). Identification of Tasks, Datasets, Evaluation Metrics, and Numeric Scores for Scientific Leaderboards Construction. *ACL 2019*, (pp. 5203–5213).

Z. -Q. Zhao, P. Z.-T. (Nov. 2019). Object Detection With Deep Learning: A Review," in IEEE Transactions on Neural Networks and Learning Systems,. vol. 30, no. 11, pp. 3212-3232, doi: 10.1109/TNNLS.2018.2876865.

Appendix A

Additional plots and graphs

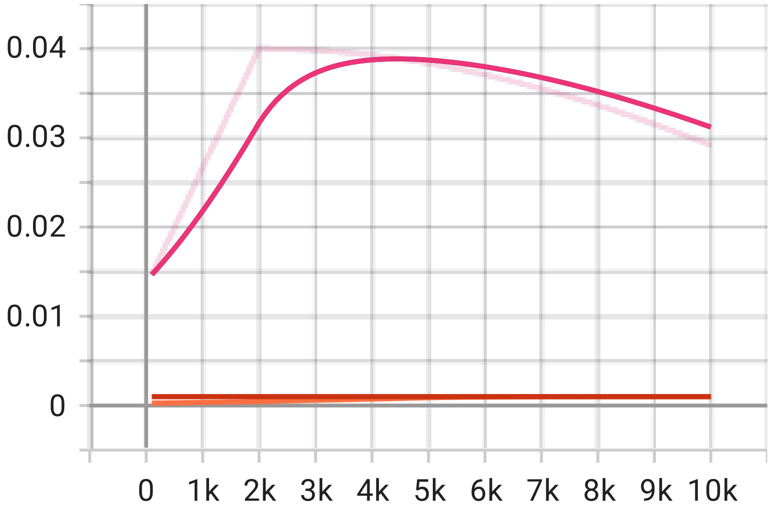


Figure 27: Learning rate over 10000 epochs

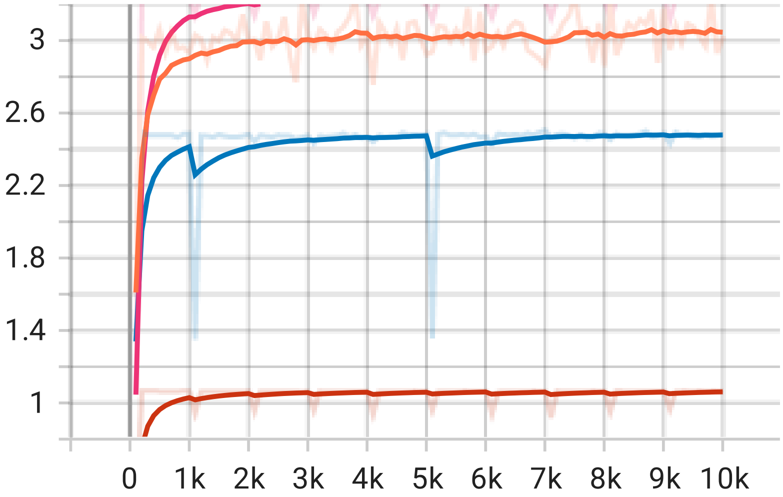


Figure 28: time per step for each model

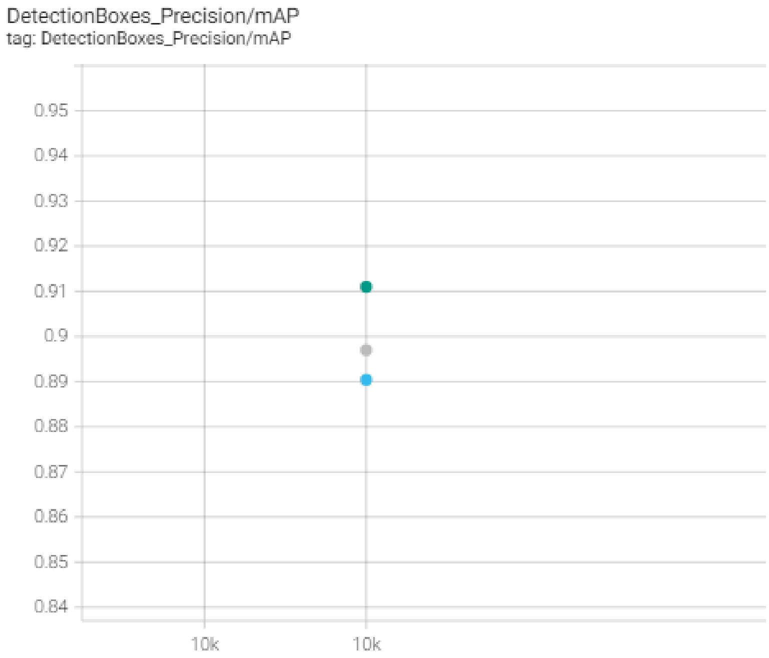


Figure 29: mAP precision for all the models

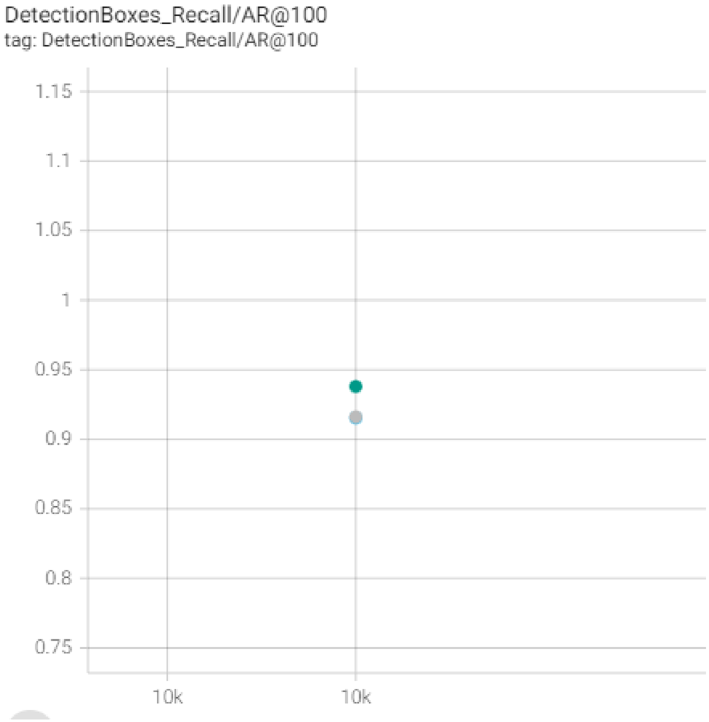


Figure 30: AR Recall for all the models