IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101115

# OBJECT RECOGNITION FOR CLIMATIC DATA

**Pooja Anbuselvan**

Bangalore Institute of Technology, Bengaluru, Karnataka

**Abstract:** Object recognition is the task of recognizing the object and labelling the object in an image or video scene. The proposed method seeks to implement an object recognition model for climatic data. Accurate characterization of objects in climate simulations and observational data archives is critical for understanding the trends and potential impacts of events in the climate. An object in an image can be recognized by extracting the features like colour, texture, or shape. Based on these features, objects of large-scale weather patterns are classified into various classes and each class is assigned a name. This paper presents an overview of object recognition methods by including two classes of object detectors. Two stage detectors such as Faster R-CNN focus more on accuracy, whereas the primary concern of one stage detectors such as YOLOv3 is speed. Faster Region-based Convolutional Neural Network method (Faster R-CNN) and You Only Look Once (YOLO) are the algorithms used for object recognition in this project. The results obtained from the two prominent approaches Faster R-CNN and YOLOv3 are compared

**Keywords -** Object recognition, climatic data, Convolutional neural network, Yolov3, Faster RCNN.

## 1. INTRODUCTION

Approximately two-thirds of our globe is covered by clouds at any given time. It's not surprising that clouds play a significant role in the Earth's climate! Clouds are necessary for the earth's atmosphere system to function. Clouds assist in the regulation of the Earth's energy balance by reflecting and dispersing solar radiation as well as absorbing infrared energy. Clouds are necessary for precipitation and are hence an important part of the hydrologic cycle. Clouds reveal what kinds of atmospheric processes are taking place. Cumulus clouds, for example, signify surface heating and atmospheric turbulence. Clouds aid in the redistribution of excess heat from the equator to the poles. Changes in the climate have an impact on clouds, just as clouds have an impact on the climate. Cloud-climate feedback is the name given to this interaction. It's one of the most difficult topics in climate science research

Object recognition is essentially object detection and image classification. Object detection is inseparably related to other similar techniques of computer vision, such as image recognition and image segmentation, allowing us to recognize and interpret the image or video scenes. Image recognition only outputs a class mark for an object detected, and image segmentation provides an interpretation of the elements of a scene at the pixel level. Its special ability to identify objects within an image or video is what distinguishes object detection from these other tasks. This then enables us to count certain items and then track them. We aim to use object recognition models to be able to recognize clouds in a given image.

This chapter gives an overview of the motivation behind the development of this model, the scope, and objectives of the solution. The proposed model of the project is also detailed here.

The proposed method presented in this paper is for cloud detection and classification. With deep learning, particularly CNN, the model is able to train and learn to identify clouds. Firstly, the real- time You Only Look Once (YOLO) Algorithm is used which as of now is one of the most effective algorithms for object detection. It differs from the rest of the neural network models because it uses one convolutional network that predicts the bounding boxes and the corresponding probabilities. The model makes its detection based on the final weights obtained after training.

The arrangement of this paper is as follows. Below in section 2, Related works in the field of object recognition methods are covered. It includes both two stage and one stage detectors with their methodologies and drawbacks. Section 3 includes the Dataset preparation and pre-processing of the data. Section 4 elaborates on the approach used in this project. Section 5 includes the flow of the model. Section 6 describes implementation results and comparison of object detection methods based on speed, loss and accuracy. Finally, section 7 summarizes the conclusion

## 2. RELATED WORK

The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. To unify the RPN with fast RCNN they have proposed a training scheme in this paper [1] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks that alternates fine-tuning between region proposal task and object detection task. It is translation invariant, both in terms of the anchors and the functions that compute proposals relative to the anchors. This method is cost-efficient and faster. The limitation of this method is that it is inefficient when it comes to dealing with extreme shapes or scales.

  

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101115

The next paper [2] presents an exclusive survey of 3D CNN-based object recognition techniques. 3D-based CNN is divided into three main parts.

- Volumetric, where the depth of the object is calculated by various methods
- View-based: reconstruction from overlapping 2D
- Motion-based, where the time interval between frames, is used as a 3rd dimension.

The conclusion of this paper is that multi-view Based Methods are better. Limitations are: Memory usage, computations are high and large size of data is required for training.

A 3D point cloud segment, which may include background clutter, is addressed in this paper [3]. Voxnet uses volumetric occupancy grid representation with 3D CNN. Voxnet has the occupancy grid which estimates the spatial occupancy and the 3D CNN which predicts the class label. The limitations are: Architecture is complex and depending on the number of points, raytracing may also be a bottleneck.

This review paper [4] Object Detection with Deep Learning a review, provides a detailed description of region proposal-based methods (R-CNN, SPP-net, Fast R-CNN, Faster R-CNN) and classification/regression-based methods (YOLO, SSD). This paper also provides a detailed explanation of each method and its limitations. Conclusion: RFCN and Faster RCNN perform the best. The limitations are RCNN: requires fixed image size, expensive, time-consuming. SPP-net: storage is expensive, fine-tuning cannot be updated. Faster RCNN: time-consuming, does not support scalability FPN: memory consumption is high, inconsistencies.

This paper [5] implements YOLOv2 which works on the framework called Darknet which is employed to train neural networks, inspired by GoogleNet architecture which is written in C/CUDA. It first makes a pass through the entire image, making grids (number of grids is based on the complexity of the image) followed by classification and localization of the image is performed in each grid cell. Next, thresholding is performed and based on the value grid cells with the highest probabilities are picked. The threshold value here is set to 0.35(can be changed). Anchor Boxes are used which detect the objects in a single grid cell and give the location of the object. Finally, Non-max suppression uses Intersection over Union for final detection. YOLOv2 uses batch normalization in all convolutional layers which improves mAP localization errors.

YOLO is an object detection approach based on deep learning. It presents a single convolutional neural network for location and classification. YOLO is used as a reference to create a better object detection method called optimized YOLO (OYOLO). The purpose of this paper [6] is to classify and locate objects of images captured in a traffic scene. There are 2 channels in this system. One is OYOLO responding to high-speed requirements and the other is OYOLO + RFCN (Fully Convolutional network) in which the images are fed for the improvement of accuracy. For challenging images during the night, pre-processing procedure is done using the histogram equalization approach. The final mean average precision of the system is about 86.4% on the testing set.

This paper [7] is based on the You Only Look Once (YOLO) algorithm that explains frames detection as a regression problem and how a simple pipeline is used for the entire image as a whole by using single convolution networks. It is trained on the loss function. Divides the image into S*S grids, for each grid cell B bounding boxes are possible, confidence is calculated for all and the Intersection Over Union between the grids and the ground truth box. C conditional class probability is also required to calculate. Network architecture is inspired by the GoogLeNet model for image classification. The limitations of this paper are: Incorrect localizations, small objects cannot be detected, it struggles to generalize to objects in new or unusual aspect ratios or configurations.

## 3. PREPROCESSING DETAILS:

### Dataset preparation:

The dataset used in this project has been taken from the Cirrus Cumulus Stratus Nimbus (CCSN) Database from the Harvard Dataverse, version 2 [8]. The dataset contains images of clouds with a fixed resolution of 256 X 256 pixels in the jpeg format. The set of photos are incredibly diverse and include complex scenes with several objects besides the selected classes. The dataset has been split into 80% training data and 20% testing data. The images used for training have been annotated using the LabelImg tool for YOLO and Makesense.ai tool for Faster R-CNN and have been labelled as ‘cloud’.

  

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101115

Figure 1: Image samples from the CCSN dataset

**Data Pre-processing:**

"Garbage In Garbage Out" is a phrase commonly used in the Deep learning community to emphasize that the model's quality is determined by the quality of the training data. The same can be said for data labeling annotations. As a Deep learning model learns by looking at samples, similarly the model's outcome is influenced by the labels we provide it during the training phase. Plotting the bounding boxes for images is the first and foremost step before implementing and comparing the two algorithms. A bounding box is an imaginary rectangle that serves as a point of reference for object detection and creates a collision box for that object [11]. The bounding boxes help outline the object of interest within each image by defining the coordinates as well as the height and width. Bounding boxes help train an algorithm in what a cloud looks like. The efficiency of the model can be increased by using a diverse dataset and accurately drawing the bounding boxes.

A labeled dataset is required to train the Yolov3 algorithm hence, a labeling tool called LabelImg is used for annotating the images for YOLO, a .txt file with the same name is created which includes the object class, object coordinates, height, and width in the YOLO labelling format. Here, the object class value will be 0 as there is only a single class labelled cloud

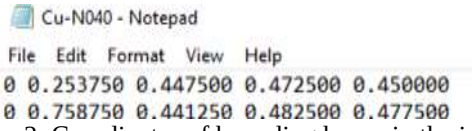


Figure 2: Coordinates of bounding boxes in the image

In Faster RCNN, the images are annotated using a tool named MAKESENSE.AI. Each image has an xml file with the same name created in PASCAL VOC format. Each xml file explicitly mentions the coordinates of each of the bounding boxes with its object name. The file also includes the depth, height, width of the image. After reading this file the coordinates xmin, ymin, xmax, ymax are extracted and saved separately in a new annotation file which is used for training and testing

## 4. APPROACH:

### 4.1 FASTER RCNN:

Faster RCNN is an improvised version of Fast RCNN. Fast RCNN uses the selective search for generating regions of interest whereas Faster RCNN uses a separate layer known as the Region proposal network. The following provides a brief overview of the algorithm.

1. An image is passed as an input to the ConvNet which returns the feature map for that image.
2. These feature maps act as input to the Region proposal network. The RPN Layer returns the object proposals along with their objectness score.
3. An RoI pooling layer is applied to the output of the RPN Layer proposals to reduce them to the same size.
4. Finally, the proposals are passed to a fully connected layer which has a softmax layer to classify and a linear regression layer to provide the bounding boxes.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101115

#### 4.1.2. BASE NETWORK

VGG-16 network is used as the base network. It has already been trained on the ImageNet dataset. Abstractions are created at each layer based on previous information. For example, the first layer usually learns about the edges and the second layer learns about the patterns. The spatial dimensions of the convolutional feature map are much smaller than the original image, but it has greater depth. Pooling applied between the convolutional layers decreases the height and width of the feature map. The depth increases depending on the number of filters the convolution layer has.

#### 4.1.3. RPN LAYER

The convolution feature map generated by the backbone layer is passed through the RPN network and anchors are generated by sliding window convolution applied on the input feature map. Differently shaped and sized k anchor boxes are generated. This layer is used to find up to a predefined number of regions, which may contain objects. For each anchor, RPN predicts two things:

- The probability that an anchor is an object ignoring the class it belongs to
- The bounding box regressor for adjusting the anchors to better fit the object.

The network generates the maximum number of k- anchor boxes. For each of the different sliding positions in the image, by the default the value of k=9 (3 scales of (128*128, 256*256 and 512*512) and 3 aspect ratio of (1:1, 1:2 and 2:1)) [12]. N = W* H* k anchor boxes are generated. The output is further passed onto an intermediary convolution layer and the output generated from this is passed onto 2 layers: the classification layer and the regression layer.

The classification branch gives an output of size (H, W, 18). This output is used to give probabilities of whether or not each point in the backbone feature map (size: H x W) contains an object within all 9 of the anchors at that point. The regression branch gives an output of size (H, W, 36). This output is used to give the 4 regression coefficients of each of the 9 anchors for every point in the backbone feature map (size: H x W). These regression coefficients are used to improve the coordinates of the anchors that contain objects.[13]

#### 4.1.4. ROI POOLING

From the RPN layer bounding boxes of different sizes are passed onto the ROI Pooling Layer. There may be bounding boxes with no classes assigned to them therefore in this case the ROI Pooling Layer takes each proposal, crops it, and then passes it through the pre-trained base network. The extracted features can be used as input to the classifier. Then these feature maps are passed to a fully connected layer which has a softmax for classification and a linear regression layer for predicting the bounding boxes.

#### 4.1.5. TRAINING AND LOSS FUNCTION

Training and Loss Function (RPN): We remove all the cross-boundary anchors so that they do not increase the loss function. The IoU threshold and non-max suppression threshold are set as 0.7. This
decreases the number of anchors considerably. Backpropagation and stochastic gradient descent are used to provide end-to-end training to the RPN Layer. It generates each mini-batch from the anchors of a single image. The loss function is trained on 256 random anchors with positive and negative samples in the ratio of 1:1 rather than on each anchor. If an image contains <128 positives then it uses more negative samples. Positive label is assigned only if both IoU and NMS values are greater than the threshold values. The loss function obtained is a sum of both regression as well as classification loss [14]. The training loss for the RPN is also a multi- task loss, given by:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*).$$

Figure 3: Loss function for Faster RCNN

Where,
pi = predicted probability of anchors contains an object or not. pi* = ground truth value of anchors contains an object or not. ti = coordinates of predicted anchors.
ti* = ground truth coordinate associated with bounding boxes. Lcls = Classifier Loss (binary log loss over two classes). Lreg = Regression Loss (Here, Lreg = R(ti-ti*) where R is smooth L1 loss) Ncls = Normalization parameter of mini-batch size (~256).
Nreg = Normalization parameter of regression

### 4.2. YOLOv3

YOLO is a state-of-the-art real-time object detection algorithm. In this project, the Yolo algorithm is applied to detect clouds in the images. The idea behind this algorithm is that a single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. The YOLOv3 uses a variant of Darknet which is a framework to train neural networks. Darknet originally has 53 layers and for the detection task, another 53 layers are stacked onto

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101115

it, accumulating to a total of a 106-layer fully convolutional architecture [10]. Darknet is also used as a feature extractor and it includes 3 x 3 and 1 x 1 filters with skip connections like the residual network.

A brief working of YOLOv3 algorithm is listed below:

- The input is a batch of images of shape (416, 416, 3)
- YOLOv3 passes this image to a single convolutional neural network (CNN)
- The grid cells can predict B Boundary boxes and each box has a box confidence score.
- Each bounding box is described by a vector, the vector includes x, y, w, h, c.
- The output will be a list of bounding boxes alongside the list of recognized classes.
- Finally, the IoU (Intersection over union) and NMS (Non-Max suppression) is performed to avoid selecting overlapping boxes.

Each bounding box contains 5 elements which are x, y, w, h and a box confidence score.The confidence score shows how likely the box contains an object which is called objectness and how accurate the boundary box is.The bounding box width w and height h is normalized by the image width and height.The x and y values are offsets to the corresponding cell. Hence, x, y, w,and h are in the range of 0 and 1.

In our implementation, the input image is resized to 416 x 416 x 3 and divided to 13 x 13 cells. Each cell contains 9 bounding boxes. The bounding box generates a score which represents the probability that the box contains an object. As the detection takes place in the 82nd, 94th and 106th layer of the Yolo architecture, the number of filters in the preceding layers were optimized according to the number of classes in the data. The initial weights required were uploaded to train the model. As the dataset contains one class, the maximum batch was set to 2000 and 18 filters in the three convolutional layers before the YOLO layers.

The YOLOv3 model was built on Google Colaboratory with the support of its built-in GPU using an open-source neural network framework called Darknet from AlexeyAB's repository [9]. Using the YOLOv3 initial weights from the repository, this project was able to immediately run training on the custom dataset. To customize our object detector to identify clouds, the model was trained on the customized dataset to increase the efficacy of the neural network.

| **Parameters** | **Value** |
|---|---|
| Input image size | 416 x 416 x 3 |
| Number of cells per image | 13 x 13 |
| Number of bounding boxes per cell | 9 |
| Classes | [cloud] |
| NMS threshold | 0.6 |

Table.1: Yolov3 implementation parameters

## 5. METHODOLOGY

Several steps were taken to ensure the system was able to obtain the desired output. The figure below shows the flow of general operation of the proposed system and its functionalities**.**

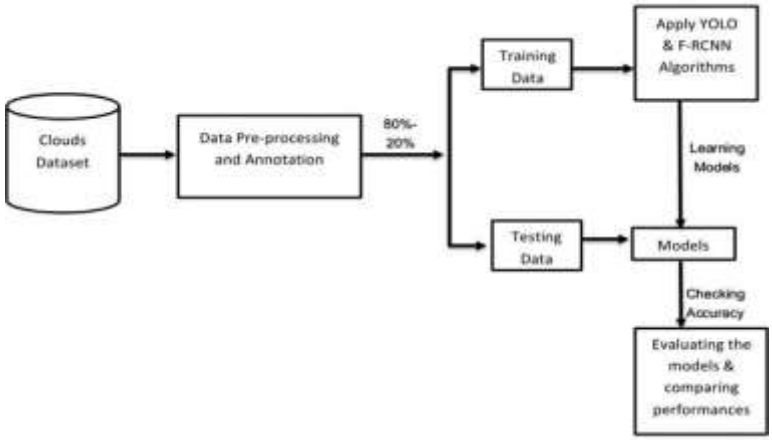


Figure.4: Flow diagram of the proposed system

  

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101115

## 6. IMPLEMENTATION RESULTS AND ANALYSIS

The model was tested on test images of the CCSN dataset. Based on the final weights which were obtained post-training, the model uses Intersection-Over-Union (IOU) to find the overlap ratio of the detected box and the ground truth box. However, any detection that has an IOU above 0.4 is considerably good and detections below 0.4 were filtered out completely.

Non-Maximal Suppression (NMS), a process that makes use of IOU to seek the best bounding box is applied. NMS essentially removes any of the predicted bounding boxes that have a detected probability that is lower than a set threshold of NMS and selects the bounding box with the highest probability. By doing so, only one bounding box for an object is returned.

The threshold for classification was set to 0.5 and non-maximum suppression for the detection boxes overlapping was set to 0.6. The two approaches have proven to be successful from the testing done. In terms of processing frame speed, the YOLO algorithm surpassed the FRCNN method. However, it needed more accuracy. The transfer learning-based FRCNN architecture was highly successful. The trained model achieved an overall accuracy of 60 %. Fig. 5 and 6 show the samples of the detection of Faster R-CNN and Yolo models respectively

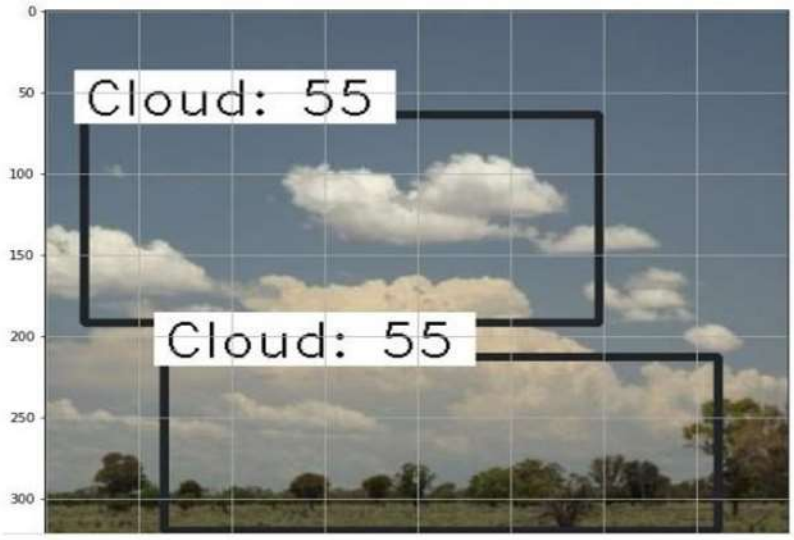


Figure.5: Cloud detected on test image for Faster RCNN

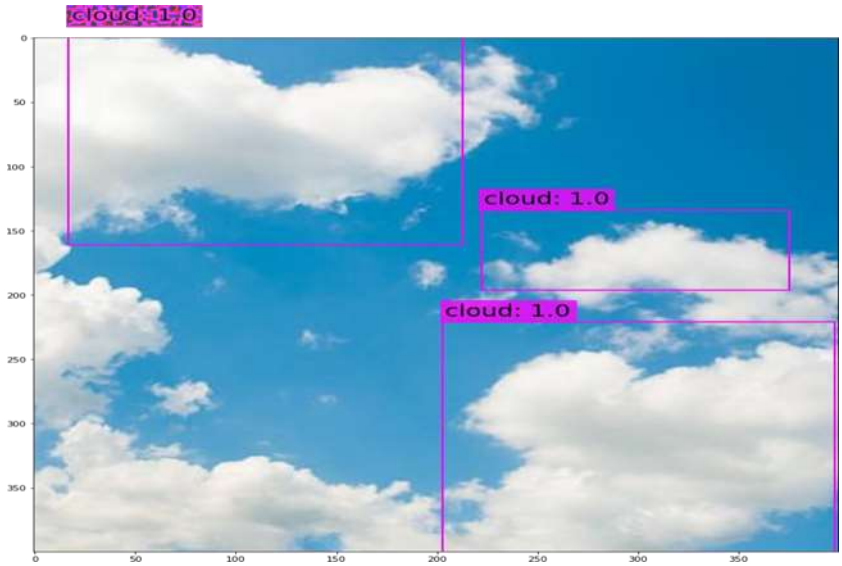


Figure 6: Cloud Detected on test image for Yolov3 model

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 11, November 2021

DOI: 10.17148/IJARCCE.2021.101115

A myriad of parameters was considered to pick the model which works best in our case. The slight difference in the accuracy was the rationale for biasing our choice towards Faster RCNN. Faster RCNN proved to be a better model framework in terms of prediction performance, flexibility, explaining ability, and accuracy. The loss calculated for Faster RCNN was ≅ 2.77

| Model | Accuracy | Average Loss | Time taken( 1 image) |
|---|---|---|---|
| **Faster RCNN** | 0.60 | 2.77 | ≅ 11 seconds |
| **YOLO** | 0.50 | 0.294 | ≅ 1.5 seconds |

Table 2: Observed results

In Table 2, three different metrics (Accuracy, Loss and Time) have been used to present a complete and reliable analysis.

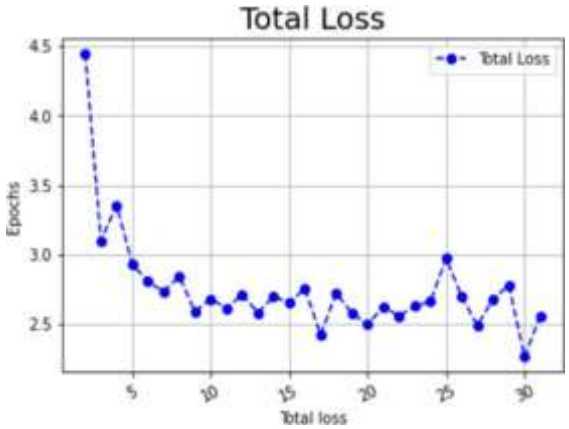


Figure 7: Loss plot for Faster RCNN

Figure 7. shows the total loss for each epoch for the Faster RCNN algorithm. The loss includes the sum of classifier and regression loss. It can be observed that there is a steep decrease as the number of epochs decreases for the first 20 epochs

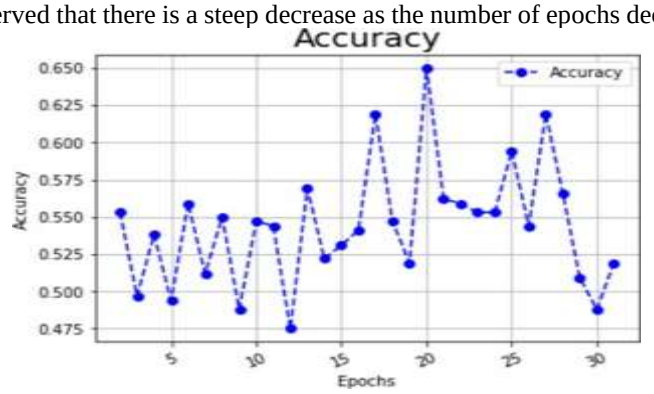


Figure 8: Accuracy plot for Faster RCNN

Figure 8. shows the accuracy plot for Faster RCNN. The highest accuracy reached was approximately 65.0

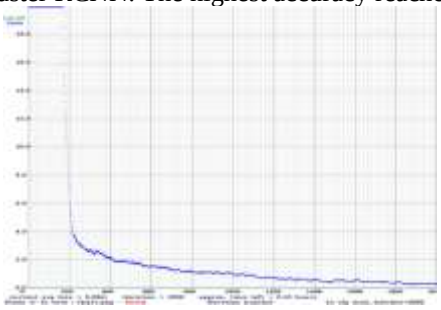

Figure 9: Loss plot for YOLOv3

  

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 10, Issue 11, November 2021
DOI: 10.17148/IJARCCE.2021.101115

The above graph depicts the loss plot of the yolov3 algorithm. The graph above shows the performance of the neural network during the training. It shows the average loss versus the iterations that occurred during training. The average loss obtained at the end of training 2000 max_batches is 0.294. It is found that the average losses during the training exponentially decreased as the iterations increased.

## 7. CONCLUSION

Deep learning is a burgeoning field of study, and currently, there are a number of potential alternatives to CNN models. Both the algorithms have been previously proven to be highly efficient considering a wide range of inputs. YOLO is known for its simpler architecture and speed. Faster RCNN overcomes the drawback of YOLO and can detect objects at a small scale in images. Faster RCNN has a more complex architecture involving the RPN layer whereas YOLO provides an elegant design of code. However, the accuracy obtained is not at par with the accuracy obtained on well-known datasets such as COCO and IMAGENET.

From the values obtained we can infer that the YOLO model has a slightly lower loss when compared to Faster RCNN. Loss can signify how well the model performs on a simple input. If the prediction is perfect the loss obtained is 0. As mentioned above we can also conclude that the accuracy of the Faster RCNN model is better when compared to the YOLO model. In accordance with the inference drawn from the literature survey, the YOLO model takes less time for object detection. If time and loss values are considered as the prime deciding factors, YOLO will be an obvious gain as compared to Faster RCNN.

Since the annotation was done on an online tool the exported values for annotation boxes will depend on the human accuracy of drawing the bounding boxes. The accuracy of the bounding boxes could be affected by naive judgement. Owing to the originality of the dataset the accuracies obtained could be termed as fair. The difficulties of Deep learning algorithms in recognizing the object of interest efficiently were highlighted, and comparison studies of the two Deep learning algorithms were conducted

## 8. REFERENCES

[1] "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. doi:10.1109/tpami.2016.2577031

[2] "3D convolutional neural network for object recognition: a review." Singh, R. D., Mittal, A., & Bhatia, R. K. (2018).Multimedia Tools and Applications. Springer Science+Business Media, LLC, part of Springer Nature 2018. doi: 10.1007/s11042-018-6912-6

[3] "VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition". Daniel Maturana and Sebastian Scherer. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). doi:10.1109/iros.2015.7353481

[4] "Object Detection with Deep Learning: A Review" Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu,Xindong Wu. IEEE Transactions on Neural Networks and Learning Systems (Volume: 30, Issue :11, Nov. 2019)

[5] "YOLOv2 based Real Time Object Detection" Sakshi Gupta, Dr. T. Uma Devi. International Journal of Computer Science Trends and Technology (IJCST) – Volume 8 Issue 3, May-Jun 2020

[6] "An object detection system based on YOLO in traffic scene," J. Tao, H. Wang, X. Zhang, X. Li and H. Yang, 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 2017, pp. 315-319, doi: 10.1109/ICCSNT.2017.8343709.

[7] "You only look once: Unified, real-time object detection." Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.

[8] "Cirrus Cumulus Stratus Nimbus (CCSN) Database" Liu, Pu, Harvard Dataverse, V2, 2019, https://doi.org/10.7910/DVN/CADDPD

[9] "Alexey. AlexeyAB/Darknet. 2016. 2020. GitHub", https://github.com/AlexeyAB/darknet Web. 2 July. 2020.

[10] https://medium.com/analytics-vidhya/yolov3-real-time-object-detection-54e69037b6d0

[11] https://www.google.com/url?q=https://keymakr.com/blog/what-are-bounding-boxes/&sa=D&source=editors&ust=1628775620806262&usg=AOvVaw2KCDPYu7tp4AJ0KCzL 4K3f

[12] https://www.google.com/url?q=https://www.geeksforgeeks.org/faster-r-cnn-ml/&sa=D&source=editors&ust=1628775620807057&usg=AOvVaw0aPNJlmSCq4JbKbxVJi02E

[13] https://www.google.com/url?q=https://towardsdatascience.com/faster-r-cnn-for-object- detection-a-technical-summary-474c5b857b46&sa=D&source=editors&ust=1628775620806741&usg=AOvVaw0RALUahjS5Rq Q2w6v8nUc-

[14] https://www.google.com/url?q=https://www.geeksforgeeks.org/faster-r-cnn-ml/&sa=D&source=editors&ust=1628775620807355&usg=AOvVaw0lIUBP7o-RFi5w11MK0j9V

  