INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 6)
Available online at: https://www.ijariit.com

DNN Design for Object Detection in Airport Runway Operations

Jithin P. Sajeevan
lithoeratech@gmail.com
LRDC, Bangalore, Karnataka

Prajval P.
prajvalprasannakumar@gmail.com
APS College of Engineering, Bengaluru, Karnataka

Shourya Mishra
shourya.17.becs@acharya.ac.in
Acharya Institute of Technology, Bengaluru, Karnataka

Shwetha G. P.
shwethapishe1999@gmail.com
Acharya Institute of Technology, Bengaluru, Karnataka

ABSTRACT

In this project, we attempted to solve the problem of object detection in airport runway environments by leveraging DNNs. The DNN model developed will be able to classify objects and also accurately localize objects of different classes. The major aspect here is the task of object detection is going to be considered as a regression problem to object bounding box masks. It would be a multi-scale inference procedure that can generate high-resolution object detection at a minimal cost. Here we were able to test our custom dataset on three object detection models. The models considered were (1) Faster-RCNN with VGG-16, (2) YOLOv3 with darknet53, and (3) SSD Inception V2(coco) from the TensorFlow object detection API. All the above algorithms were trained using the Tensorflow framework. The paper gives a brief comparison in the performance of the above-mentioned algorithms when trained and tested while keeping the main goal to be object detection in airport runways. We trained the YOLOv3 with darknet53 on our custom dataset and were able to obtain the classifier accuracy for the bounding boxes to be 74.00% which was the best of the three algorithms and has outperformed many previous works. The YOLOv3 could be considered a fast training algorithm when the model is trained on a powerful GPU and could play a major role in the field in terms of real-time object detection scenarios.

***Keywords**: Deep Neural Networks (DNN), Region-Based Convolutional Neural Network (RCNN), You Only Look Once(YOLOv3), Single Shot Detection (SSD), Faster-RCNN with VGG-16*

1. INTRODUCTION

The real-time object detection scenario is still filled with numerous challenges and a lot of room to improve. The major objective under consideration here is to improve object detection in airport runway environments. The three algorithms are chosen here while treating the classification problem as a regression still hold distinct differences in their architectures. The models will be trained to give the probabilities that, given an input image belongs to the corresponding class. The way we want to determine the position is by identifying the coordinates of the bounding box of the object or subject. The main goal here is to obtain the coordinates of the bounding boxes, typically by identifying the coordinates of the corner or the center along with the height and width of the bounding box. The key metric parameter observed here would be the classifier accuracy for the bounding boxes which measures the ability of the algorithm to classify the images and detect the part of the image with the subject to draw a bounding box around them.

2. AIRPORT RUNWAY OPERATIONS DATASET

A custom dataset was collected specifically to serve the purpose of the goal. Images were scouted online from various sources and collected 645 images of the airport runway environment. The collected images were classified into six targeted classes. The following classes were considered : (1) airplanes, (2) people, (3) baggage carrier, (4) tow tractor, (5) hangar, and (6) birds. The classes were labeled using the Label Image application and annotated in PascalVOC and the MS coco format.

The challenging part of the above classes was the detection and classification of the birds class, which would be considered quite tricky for any algorithm as there is a similarity between the birds class and airplane class concerning the size and shape of the two subjects.

In the case of cross-validation 5 images from each class were considered to find the performance of the trained algorithm.

The airport runway operations data set was used for the object detection task and is made available at an online repository[8] for training and test data used in this project and can be retrieved from the storage for further research and testing in future.

3. DATA PRE-PROCESSING

The data pre-processing part of the process makes it easy for the AI - Computer Vision algorithms to further process the images and serve their purpose. The resolution of the images is fixed to 300 x 300 pixels. The input shape of the images is, however (300, 300, 3), the last coordinate 3 dimensions representing the R, G, and B color channel format of the images. The resolution of the images in the case of YOLOv3 is set to 416 x 416. The horizontally flipped cases of the images were also included in the training to show different perspectives of the images. All the images are resized to a factor of 1.

4. ALGORITHMS

4.1 Faster RCNN With VGG 16

In the Faster RCNN model, the main insight for its design was to replace the slow selective search algorithm with a fast neural net, specifically, it introduced the Region Proposal Network (RPN)[1]. The Faster RCNN consists of three major components as discussed here.

4.2 Convolutional Layers

Convolution networks are generally composed of convolution layers, pooling layers, and the last component which is the fully connected or another extended network that will be used for an appropriate task like classification and detection. Convolution was computed by sliding filters all along the input image and the result would be a two-dimensional matrix called feature map. Pooling consists of decreasing the number of features in the feature maps by eliminating pixels with low values[6].

4.3 Region Proposed Networks (RPN)

RPN is a small neural network sliding on the last feature map of the convolutional layers and predicting whether there is an object or not and also predicts the bounding box for those objects[6].

4.4 Classes and Bounding Box Prediction

We use another fully connected neural network that takes the regions proposed by the RPN as the input and predicts the object class (classification) and the bounding box (regression). To train the architecture we make use of SGD to optimize convolutional layers filters, RPN weights, and the last fully connected layer weights[6].

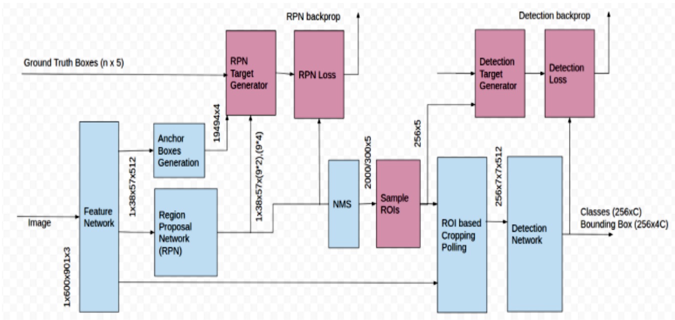

Fig. 1: Faster-RCNN Architecture

Hyper parameters, Parameters and Key Features involved in the training process is listed below :
base_model = ‘vgg16’,
anchor_box_scales in CNN = 128, 256, 512,
anchor_box_ratios = 1 : 1, 1 : 2√2, 2√2 : 1,
number_of_anchors = 9,
total_number_of_bbox = 2000,
number of region of interests = 4,
learning_rate = 0.01,
activation_function = ‘relu’ (Rectified Linear Unit), regularizer = l2,
epsilon = 1e - 4,
optimizers = ‘Adam’,
‘sgd’(stochastic gradient descent), number_of_epochs = 25 and loss_function = ‘mae’ (Mean Absolute Error)[4].The general architecture is represented by Fig-1.

A. YOLO V3 with Darknet-53

‘You Only Look Once’ is an Object Detection Algorithm. The most protruding feature of YOLO v3 is that it makes detections at three different scales. YOLO is fully CNN. This algorithm does not depend on multiple neural networks.The architecture of this algoritm is given in Fig-2.
YOLO v3 algorithm uses Darknet-53, a convolutional neural network that is 53 layers deep. The model is applied to images at multiple locations for training the model to detect the objects of different classes in the image, where bounding boxes are created around the object detected. The model is also tested for multiple images after training.

Hyperparameters, Parameters and Key Features involved in the training process is listed below :
base_model = ‘darknet53’,
anchor_box_scales in CNN = 128, 256, 512
anchor_box_ratios = 1 : 1, 1 : 2√2, 2√2 : 1,
number_of_anchors = 9,
total_number_of_bbox = 2000,
number of region of interests = 4,
regularizer = l2,
learning_rate = 0.001,
activation = 'linear', 'leaky', number_of_epochs = 25 and loss_function = ‘mae’ (Mean Absolute Error).

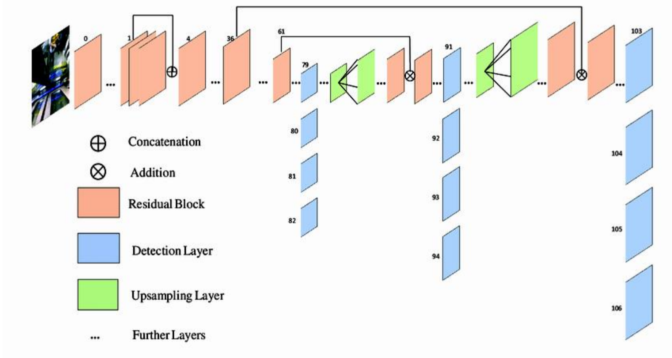

Fig. 2: YOLO v3 Architecturess

B. SSD with Inception V2

Our final model is SSD Inception V2 (Single-Shot Detector). It provides enormous speed gains over the Faster R-CNN in a different manner. The SSD performs two tasks in a “single-shot”, it simultaneously predicts the bounding box and the class as it starts to process the images.The general architecture of SSD algorithm is represented in Fig-3.

Training the SSD poses a few unique challenges. Here the filtering step is skipped and instead we classify and draw bounding boxes from every single position in the image, using multiple shapes, at different scales. Hence, generated a much greater number of bounding boxes than the other models, and nearly all of them are negative examples and to fix this imbalance, SSD does two things. Primarily, it uses non-max suppression to group together highly overlapping boxes into a single box. Later the model uses hard negative mining in to balance the classes during training.[5].

Ultimately SSD is not so different from the Faster R-CNN, it simply skips the “region proposal” step instead considering every single bounding box in every location simultaneously with its classification and since it performs everything in one shot it is the fastest of the three algorithms.

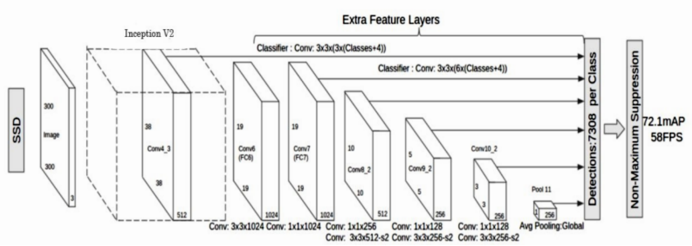

Fig. 3: SSD Architecture

Hyperparameters, Parameters and Key Features involved in the training process is listed below :
base_model = 'inception_v2',
anchor_box_scales in CNN = 128, 256, 512,
anchor_box_ratios = 1 : 1, 1 : 2√2, 2√2 : 1,
number_of_anchors = 9,
total_number_of_bbox = 2000,
activation_function = 'relu' (Rectified Linear Unit),
learning_rate = 0.01,
regularizer = l2,
epsilon = 1,
optimizers = 'Adam',
'sgd'(stochastic gradient descent), number_of_epochs = 25; and loss_function = 'mae' (Mean Absolute Error)[4].

5. COMPARISON

All the algorithms were trained using pre-trained weights available. The number of bounding boxes used in all the 3 algorithms are the same. The performance of the algorithms in terms of their classifier accuracy for bounding boxes are summarized in Table 1.

In all three cases, the process of Non-Maximum suppression is utilized to eliminate the negative resulting bounding boxes and retain the ground truth bounding box.

Table 1: Performance of algorithm

Algorithm	Number of Bounding Box	Accuracy
Faster-RCNN	2k	59.28%
YOLO v3	2k	74.00 %
SSD Inception V2	2k	52.67%

5. CONCLUSION

Training and testing of three sophisticated algorithms to serve the purpose of object detection for airport runway operations was successfully carried out. The Faster-RCNN while providing decent accuracy fails to achieve the height of the YOLOv3 accuracy of 74.00%. The YOLOv3 unifies the separate components of object detection into a single neural network. In comparison with its previous versions, the speed has been traded off for boosts inaccuracy, this has to do with the increase of complexity of the underlying architecture of Darket53.While SSD happens to be the fastest running algorithm, YOLOv3 still gives better accuracy in testing.

6. REFERENCES

[1] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. 2013:580-587.
[2] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 39(6):1137.
[3] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779 -788.
[4] Kaggle, a subsidiary of Google LLC, is an online community of data scientists and machine learning practitioners.
[5] Deep Learning for Object Detection: A Comprehensive Review, by Joyce Xu. In towardsdatascience.com, 2017.
[6] Faster R-CNN(object detection) implemented by Keras for custom data, by Yinghan Xu. In towardsdatascience.com, 2018.
[7] Object Detection Method Based on YOLOv3 using Deep Learning Networks(IJITEE) A. Vidyavani , K. Dheeraj, M. Rama Mohan Reddy, KH. Naveen Kumar Volume-9 Issue-1 Nov 2019.
[8] Datasets, Training and test archieve, Google drive link,retrieved from https://www.shorturl.at/efjKX.

