

PROJECT REPORT

Project Title : Drone Detection using Deep Learning Methods

Team Number : DL08

Team Members :

L Shiva Rudra	2020A8PS1246H
Juloori Amogh	2020AAPS0422H
Rakshith Dasari	2020AAPS0329H
Kaushik M	2020A4PS0661H

Course Title: Deep Learning (CS F425)

Course Instructor: Dr. Paresh Saxena



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

ABSTRACT

Unmanned Aerial Vehicles (UAVs), commonly known as drones have witnessed a massive increase in the past few years. Drones are not only being used for recreational purposes but also in a vast number of applications in engineering, disaster management, logistics, securing airports, etc. Extensive usage of drones has immediately raised security concerns due to the potential of their use in malicious activities. To address this problem, this project is aimed at analyzing the available drone detection solutions which can identify drones from the day and night camera feeds in real-time. This study also aims to identify the potential challenges involved in drone detection and study the various accuracy and loss metrics in drone detection problems. Finally, the drone detection solutions will be compared and a suitable COTS drone detection solution will be suggested such that it gives the best trade-off between false alarm and miss.

This project is focused on studying the nature of the YOLOv3 algorithm in object detection and understanding, implementing the methods used for improving the working of the algorithm in terms of giving better outputs for the given test data.

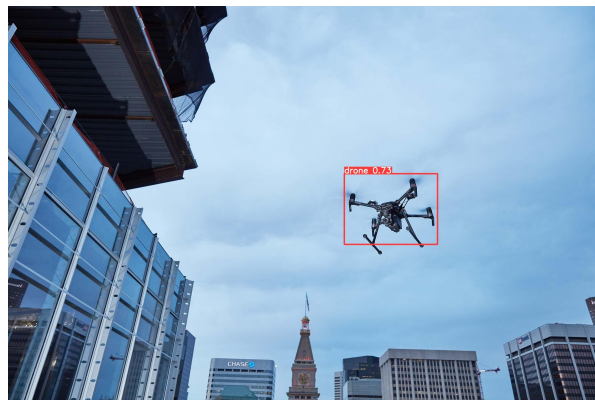
LITERATURE REVIEW

For the implementation of drone detection, we have utilized the state of the art YOLOv3 algorithm.

Bounding Boxes:

In the object detection branch of image processing, a key parameter under study is the bounding box. A bounding box is significant in both the training and testing phases of object detection. In terms of training the model, bounding boxes are used to enclose the object to be detected in the training dataset. The information about the geometry of these bounding boxes in each image is stored in the labels file of the respective image. This information helps the model to train and understand how the object under study looks like.

In the testing phase, images which haven't been used to train the model are given as the input and the bounding box drawn on the image by the detector is compared with the ground truth bounding block of the image and the evaluation metric, IoU is calculated, which helps in evaluating the accuracy of the detector.



The key geometric quantities used to describe a bounding box are box width, box height, coordinates of the center of the bounding box.

Metrics:

There are four important terms which help in understanding an object detection model:

- True Positive (TP) - A correct detection is made by the model.
- True Negative (TN) - A correct prediction is made by the model by not detecting the object, which is true.
- False Positive (FP) - An incorrect detection is made by the model.

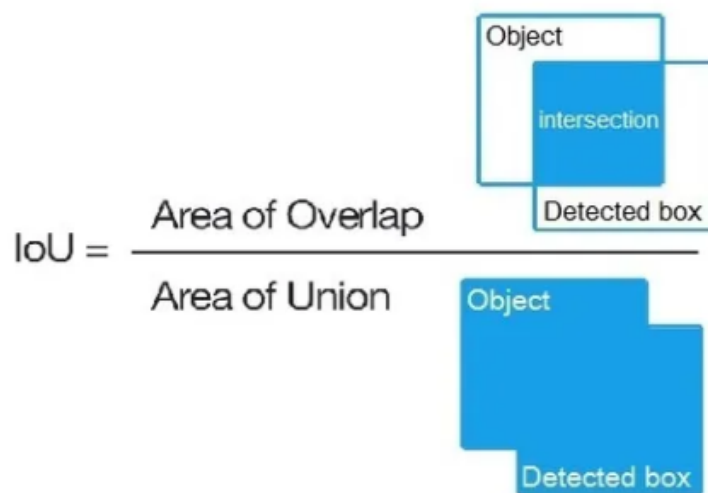
- False Negative (FN) - The object is missed/ not detected by the model.
In object detection, TN is not exactly useful since it is not used in labeling the image.

When we use a detector for the purpose of object detection, the performance of the model is signified by a few metrics, namely: IoU, Precision, Recall, AP, mAP.

1. IoU

IoU refers to Intersection over Union, which is the metric used to describe the probability of the detection in terms of the areas of the ground truth bounding block and the detected bounding block. When a detector detects an object in an image by drawing a bounding block enclosing the same, it can have some area overlapping with the ground truth bounding block of the image. In order to predict the accuracy of the detector, for that particular detection, the evaluative metric IoU is used.

$$IoU = \text{Area of Intersection} / \text{Area of Union}$$

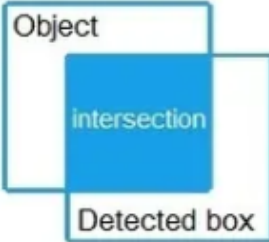


By default, an IoU > 0.6 is considered to be acceptable.

2. Precision

Precision is the measure of how accurate the model is in terms of detecting the relevant objects. It is measured as the ratio of the number of True Positives over the total number of detections made by the model.

$$Precision = TP / (TP + FP)$$



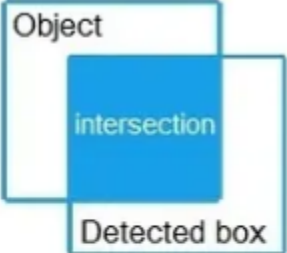
The diagram shows two overlapping rectangles. The top rectangle is labeled 'Object' and the bottom rectangle is labeled 'Detected box'. The intersection of the two rectangles is shaded blue and labeled 'intersection'.

$$\text{Precision} = \frac{\text{intersection}}{\text{Detected box}}$$

3. Recall

Recall is the measure of the performance of the model in terms of detecting the ground truths, by calculating the ratio of the number of true positives generated by the model to the total number of ground truths existing in the dataset.

$$\text{Recall} = TP / (TP + FN)$$



The diagram shows two overlapping rectangles. The top rectangle is labeled 'Object' and the bottom rectangle is labeled 'Detected box'. The intersection of the two rectangles is shaded blue and labeled 'intersection'.

$$\text{Recall} = \frac{\text{intersection}}{\text{Object}}$$

4. AP - Average Precision

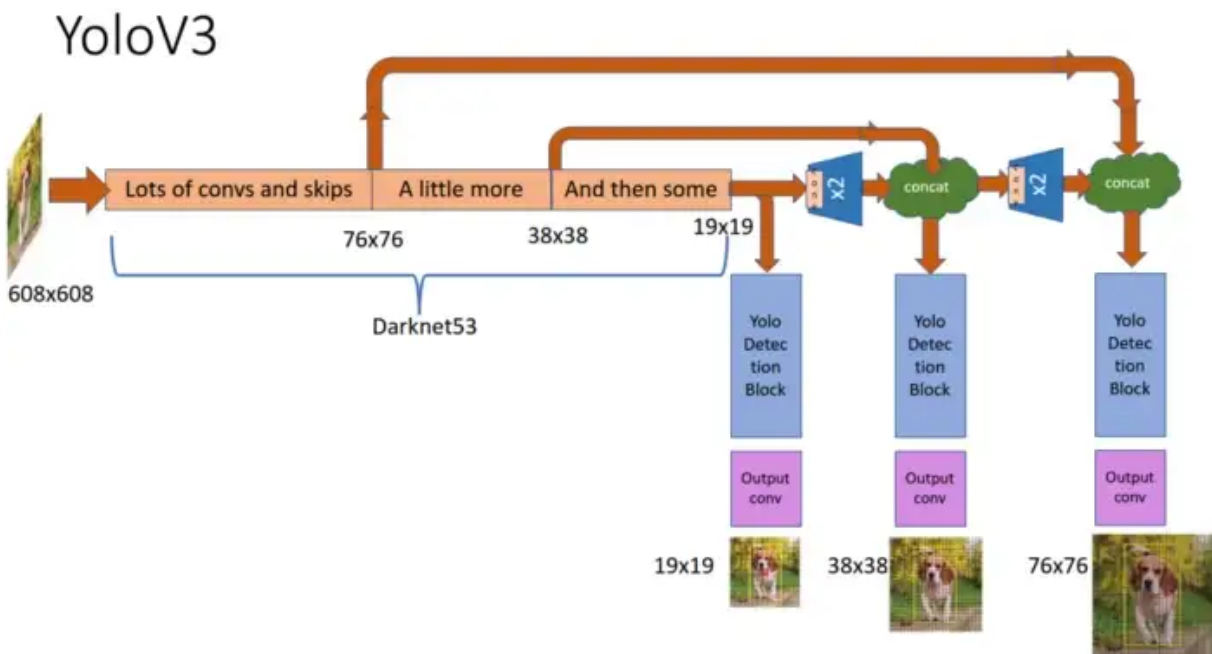
It is the area of the Precision-Recall curve over a given confidence interval for a single object class.

5. mAP - Mean Average Precision

It is the average of the AP for all the object classes.

Architecture

YOLO stands for “You Only Look Once”. It is a single shot detection technique that does not produce region proposals like the RCNN family. It is generally faster. The feature extraction and object localization were unified into a single monolithic block. Inspired by ResNet and FPN (Feature-Pyramid Network) architectures, YOLO-V3 feature extractor, called Darknet-53 (it has 52 convolutions) contains skip connections (like ResNet) and 3 prediction heads (like FPN) — each processing the image at a different spatial compression.



METHODOLOGY

Dataset Preparation:

- Preparation of the input dataset for YOLO requires two folders: Images and their corresponding Labels(consists of '.txt' files). The name of the files in the Images folder should match the corresponding label file names in the Labels folder.
- First, we obtained a dataset with around 1800 images and their labels of drones from 'Kaggle'. We divided the dataset into 'train' and 'val' folders that stand for 'training' and 'validation'.
- The training set is used to train the model and to form the corresponding weights and biases that will be applied whenever we test our model with any new image. The validation set is used to validate/verify the results obtained after training the data.
- But, there are limited drone images available open-source. Also, to improve the performance, it is required to add 'dummy data' like images of flying birds, planes and blank images of sky. Along with these, we even included images with a combination of drones and birds. This allows the model to properly distinguish between both of them.
- It is also advisable to use variations of images like blurred, greyscale, images with noise, random cropping, flipping, rotating of images, color jittering etc. This is called 'Data Augmentation'. It helps in improving accuracy and preventing overfitting.

Data Augmentation

- In image classification, it is usually recommended to randomly distort the spatial characteristics, for eg. randomly flip, rotate and crop images in order to improve accuracy and avoid overfitting.
- In object detection, one has to perform augmentation with more caution as object detection algorithms can be quite sensitive to such changes.
- Random geometry transformation includes random cropping, random horizontal flip, random expansion and random resize.
- Random color jittering includes changing brightness, hue, saturation, and contrast.

- For single-stage detectors, such as YOLO, random geometry transformation is encouraged.

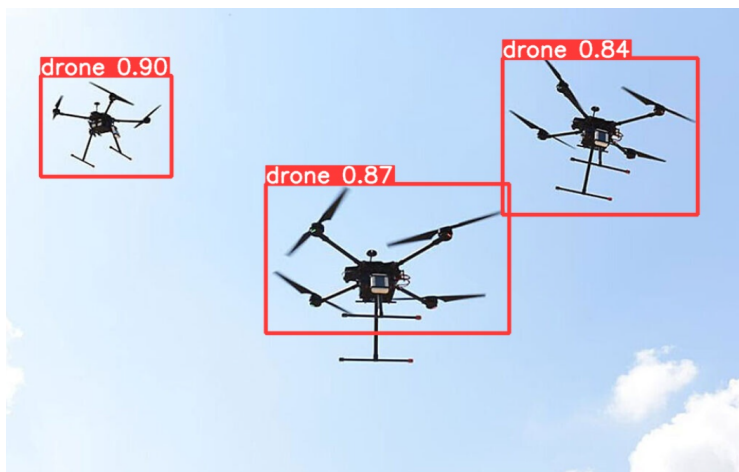
RESULTS

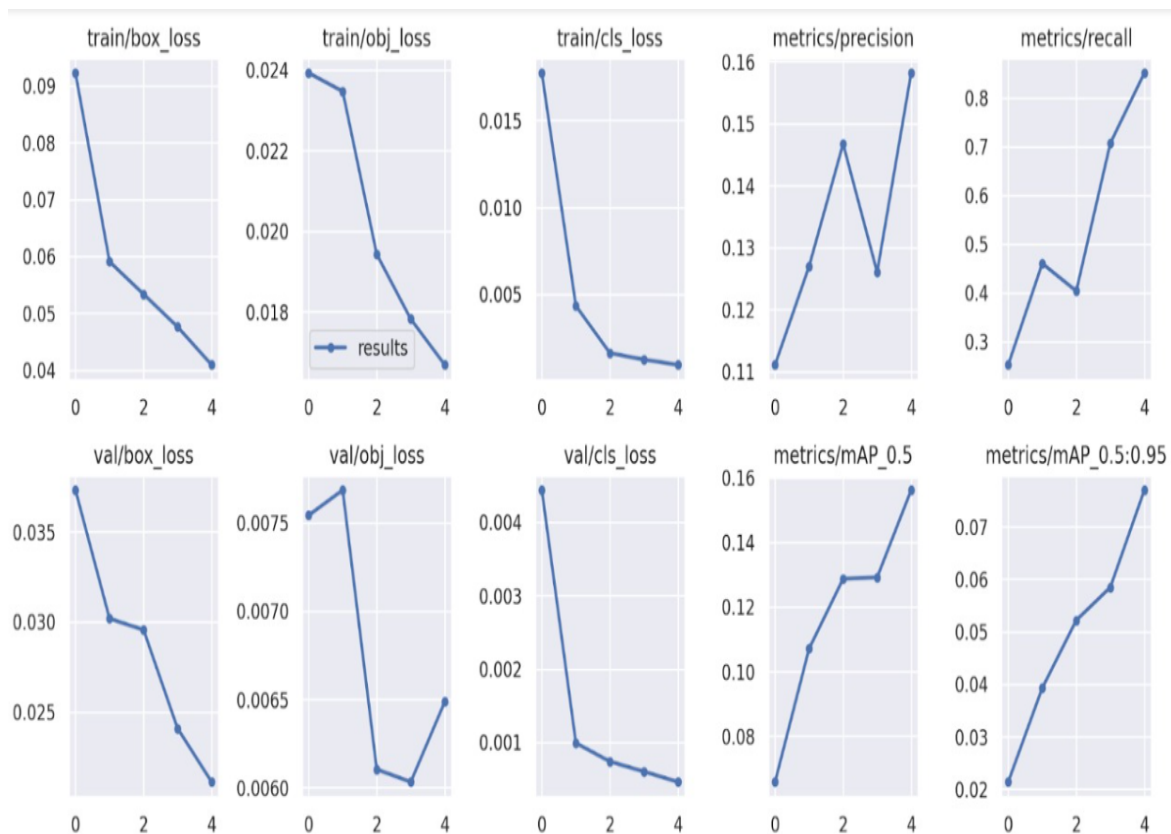
1) YoloV3 on Dataset without Bird Dataset or Augmentation(Implementation of the Paper):

Epoch count: 30

Batch Size: 16

Results:





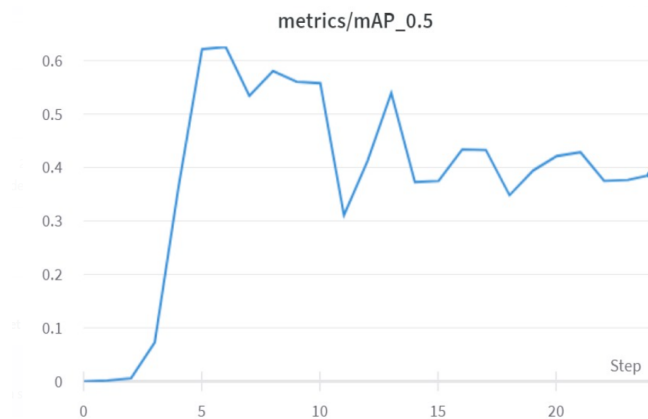
The model however incorrectly detected birds as drones as shown below. This could be a potential hassle during surveillance. We tried to resolve this by including birds in our dataset and adding it as a new label.



The paper had used a dataset of over 10000 images and ran over 150 epochs. Our systems were unable to sustain the load and it exceeded the limits of the free version of google colab.

IMPROVEMENT

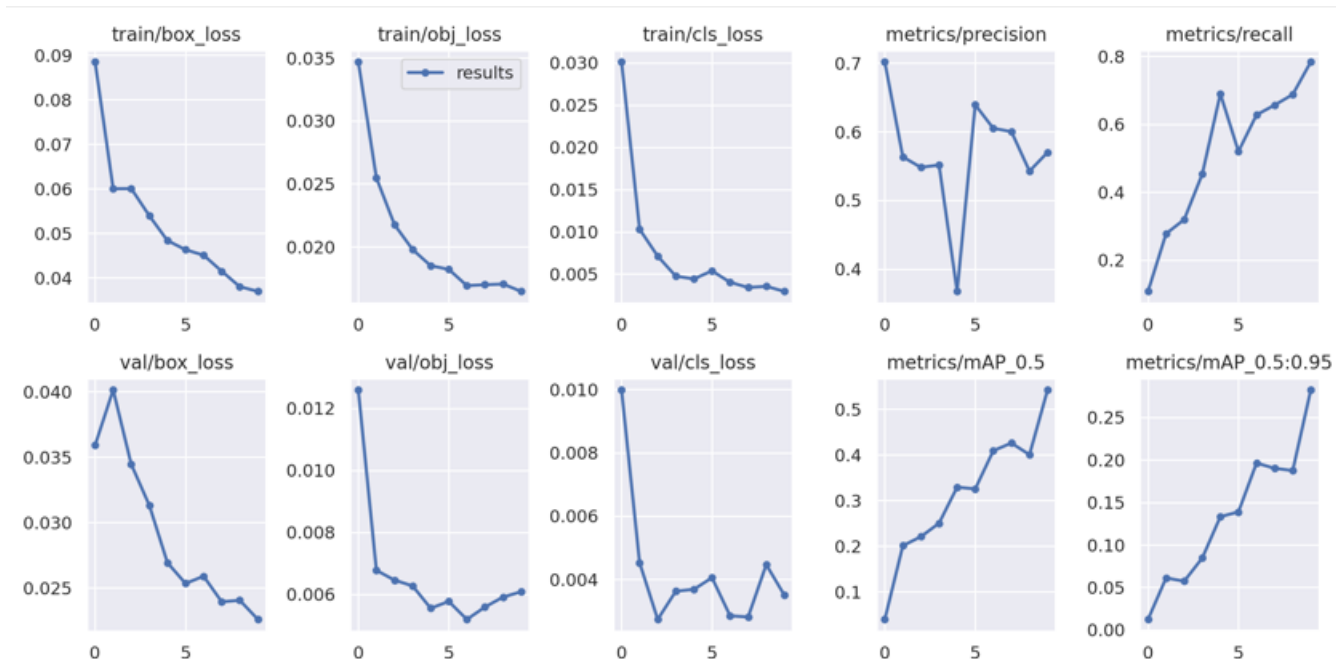
Training YoloV3 with Data Augmentation:



The mAP has significantly improved compared to the earlier case.

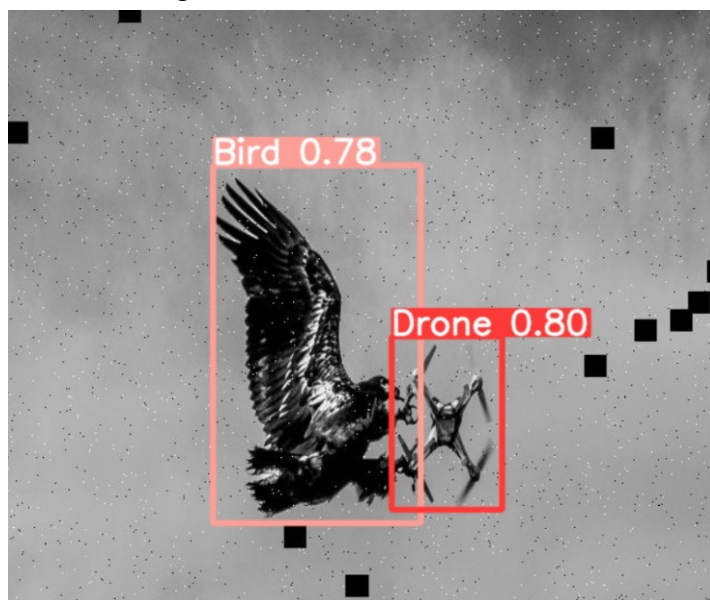
A lot of augmentation was done on the bird dataset because the number of training examples in it was significantly lesser than the drones.

Results after 10 epochs:



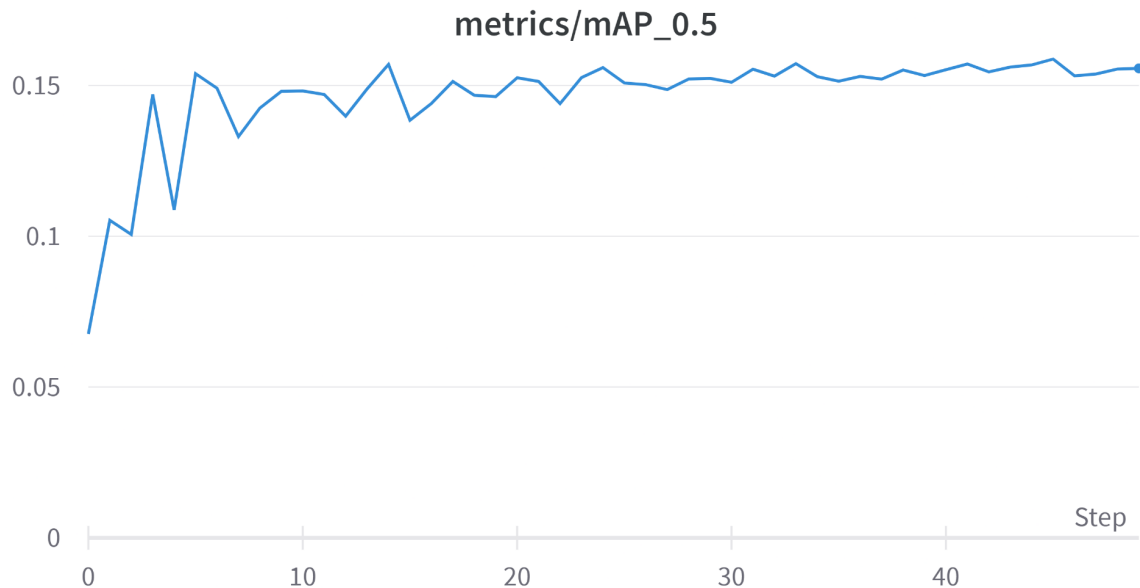
Results are pretty good with an improved mAP.

The following is the detection of a bird and a drone with the improvements applied:



EXPERIMENTS

1. Training on original drone dataset using yolov3 for 50 epochs

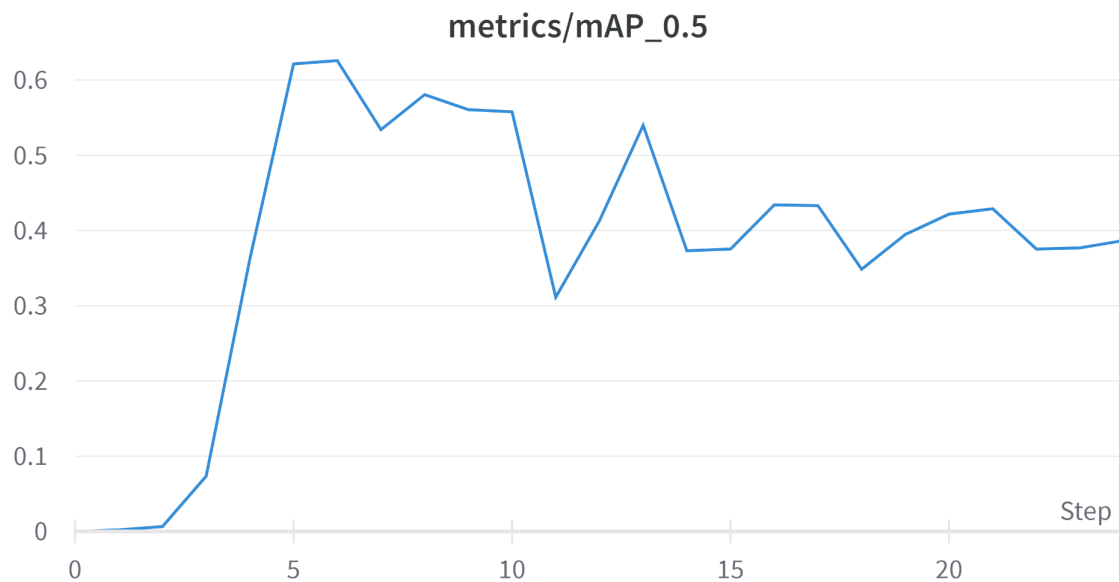


Different Augmentation Techniques like

- Random Cropping
- Brightness Change
- Cutout
- Mosaic

have been done

Ran on 25 epochs without the bird dataset:



It can be seen that the augmented dataset performs better than the original dataset but it tends to overfit after about 5 epochs. While the original dataset had an mAP@0.5 of 0.1592, the augmented dataset had an mAP@0.5 of 0.632 which is a significant improvement.

The tendency of the augmented dataset to overfit can be combated by performing Early Stopping. Hence, we eventually ran it for 10 epochs.

2. Training YoloV5 by Freezing Layers

Yolov5 uses CSPDarknet53 as a backbone. This backbone integrates gradient change into the feature map, resolves the repeating gradient information in wide backbones, which improves accuracy, minimizes the inference speed, and decreases the model size by reducing the parameters. Experimental results reveal that YOLOv3 outperforms YOLOv5 in terms of speed. However, YOLOv5 had the best recognition accuracy. The following are the architecture differences.

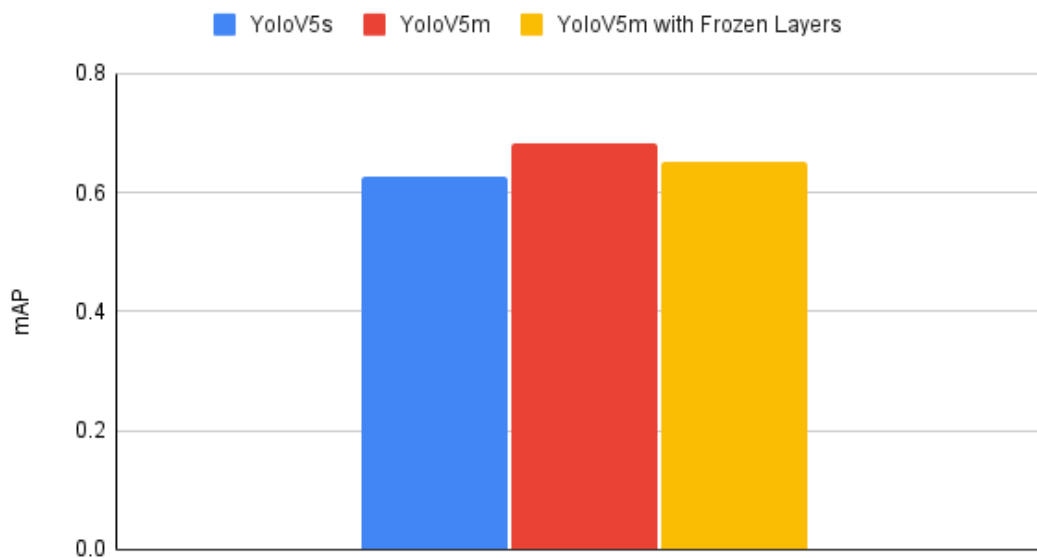
- In object detection algorithms, we generally use models which have been trained on the MS COCO dataset and fine tune it for our own dataset and we train all layers of this model.
- Pretrained models are powerful and hence we need not train the entire model. We can skip training a few layers of the model by using the '--freeze' argument while executing the **train.py** script.
- When we freeze a few layers, those layers don't get trained and hence backpropagation does not happen through those layers which reduces the training time.

- Even though we don't train a few layers, though the mAP of the frozen layer model is lower than the normal model, the difference is nominal and can hence be ignored.

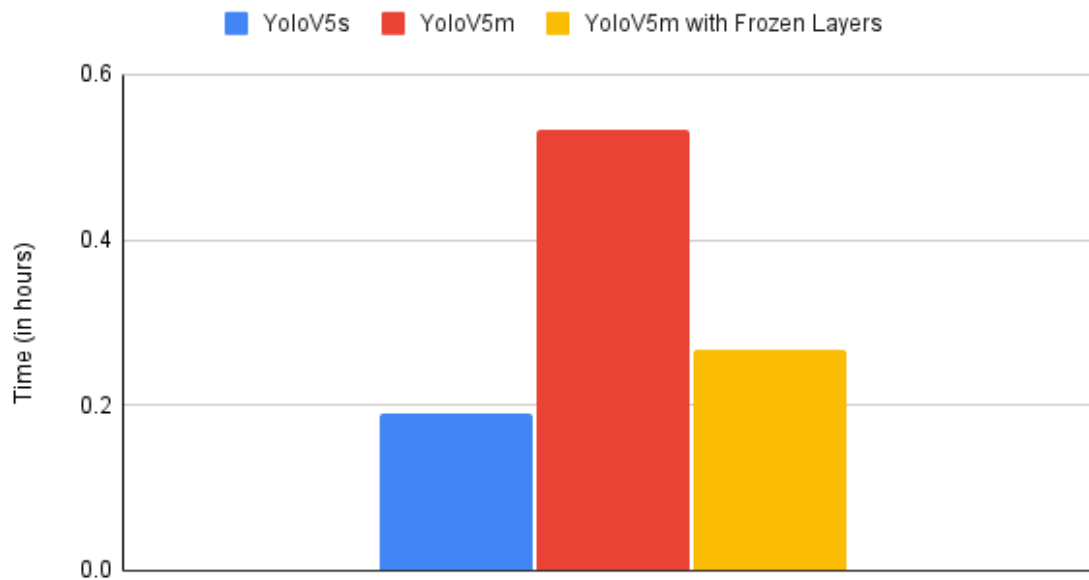
Results for Layer Freezing:

We train the model using three different pre-trained weights: Yolo 5s(small), YoloV5m(medium) and YoloV5m with frozen layers. The S version has 7.2 million parameters while the 5 version has 21.2 m parameters.

mAP Comparison



Training Time Comparison



INDIVIDUAL CONTRIBUTION:

Name	ID Number	Contributions
L Shiva Rudra	2020A8PS1246H	<ul style="list-style-type: none">• Finalized project idea.• Suggested the idea of including images of Birds and Planes into the dataset to improve the performance.• Created a dataset with birds and performed augmentation.• Implementation of Yolov3 with Bird-Drone dataset with Augmentation.

		<ul style="list-style-type: none"> • Implementation of Yolov5 with Drone dataset.
Juloori Amogh	2020AAPS0422H	<ul style="list-style-type: none"> • Training YOLOv3 and YOLOv5 for original and augmented datasets for different number of epochs. • Tested the model on different types of images and helping in assessing the performance and comparing the models.
Kaushik.M	2020A4PS0661H	<ul style="list-style-type: none"> • Proposal of idea for improvement using birds. • Proposed the idea to reduce training time using pre-trained weights in yolov5 by freezing backbone layers. • Did data augmentation of bird images.
Rakshith Dasari	2020AAPS0329H	<ul style="list-style-type: none"> • Yolov3,v5 implementation • Data Augmentation • Implementation of Layer Freezing • Research on improving results

FINDINGS AND ACCOMPLISHMENTS

The project has given a good learning experience in the field of deep learning and image processing. Training the model for different numbers of epochs and for different sizes possessing different qualities in terms of the images used, we have gained a decent experience in the field of image processing on how these parameters affect the working of the designed model. We explored different methods of improvement like: Image Augmentation that includes birds and planes. Layer Freezing allowed us to reduce the run-time.

In the end, YoloV5 with data augmentation performed the best.

BIBLIOGRAPHY

- <https://ieeexplore.ieee.org/abstract/document/9121150> - Paper followed
- <https://github.com/ultralytics/yolov3>
- <https://github.com/ultralytics/yolov5>
- <https://blog.paperspace.com/improving-yolo/>
- <https://sh-tsang.medium.com/review-bag-of-freebies-for-training-object-detection-neural-networks-d256c8fe8a9>
- <https://learnopencv.com/custom-object-detection-training-using-yolov5/#Custom-Object-Detection-Training-using-YOLOv5>