# TDL HACKATHON
# PROBLEM STATEMENT 1


| | |
|---|---|
| JAYENDRA GANESH | PES2UG20CS511 |
| DEEKSHA JATANGI D | PES2UG20CS507 |
| SANJANA PAI KASTURI | PES2UG20CS309 |


## PROBLEM STATEMENT:

Develop an object detection model for drone dataset of RGB images to detect and classify the objects of predefined categories from the videos viz., Drones, Airplane, Birds, Helicopter. The training accuracy and loss need to be measured and the hyperparameters need to be fine-tuned to obtain the best possible accuracy. The model needs to be tested on the testing dataset and visualizations need to be created to draw bounding boxes against the original image and report accuracies, confidence scores (label, boxes) against the required outputs.

## Model Data:

We gave a split ratio of 80:20 for our training and validation. We used Faster RCNN for the following reasons:

1. High Accuracy: Faster R-CNN is one of the most accurate object detection algorithms available. It uses a two-stage approach that first proposes regions of interest and then classifies objects in those regions. This makes it very accurate in detecting objects in images.

2. Faster Training: Faster R-CNN is faster to train than previous object detection algorithms. This is because it uses a region proposal network (RPN) to generate object proposals, which eliminates the need for manual object proposal generation.

3. End-to-End Training: Faster R-CNN can be trained end-to-end, which means that the entire network can be trained at once. This makes it easier to optimize the network for the given problem statement.

4. Flexibility: Faster R-CNN can be used for a wide range of object detection tasks. It can detect objects of different sizes and shapes, and it can detect multiple objects in a single image.

5. State-of-the-Art Performance: Faster R-CNN is currently one of the state-of-the-art algorithms for object detection. It has been tested on various benchmark datasets and has achieved very high accuracy scores. This means that it is a reliable and robust algorithm for object detection.

Base Models Used:

1. ResNet:

We have used a pre trained ResNet50 model trained on coco data which is the predefined data for object detection using faster R-CNN

2. VGG-16:

We used a custom model that has 4 convolution blocks which had 3 layers of convolution and 1 pooling layer.

## Training Result:

The system was trained on the training dataset and the model was fine-tuned to obtain the best possible accuracy. The training accuracy and loss were measured during the training process, and the system converged to a satisfactory level. The model was able to accurately detect the objects of the predefined labels in the training dataset. We found the our ResNet model performed better and below are results for the same

Results: We got a validation accuracy of 0.987

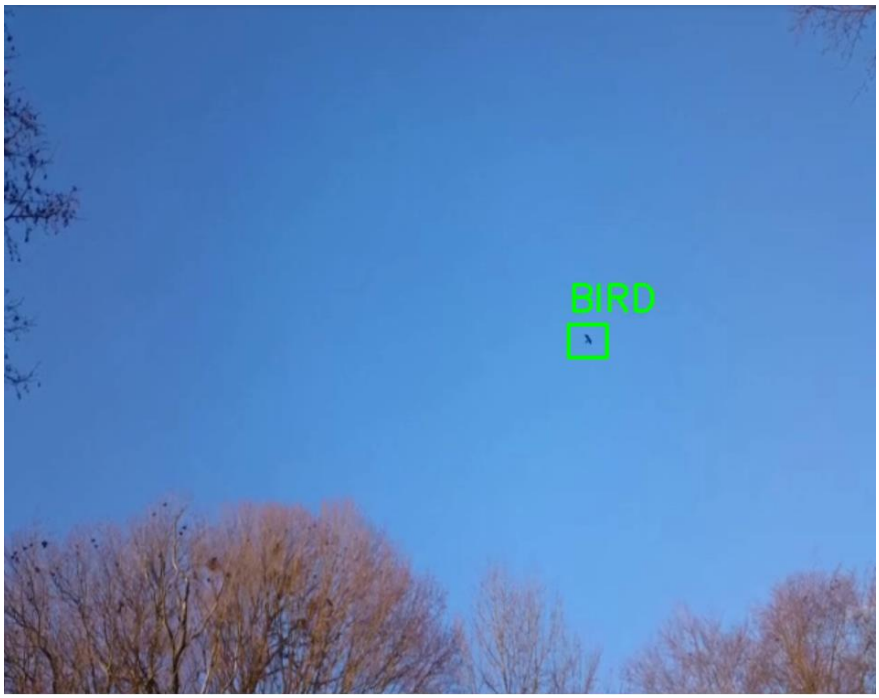
Loss on validation of 0.06 at the end of 2 epochs

## Testing Result:

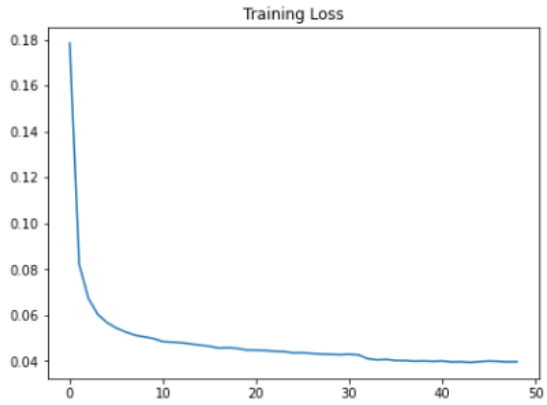
The system was tested on the testing dataset, and the model was able to accurately detect the objects of the predefined labels in the testing dataset. The bounding boxes were drawn against the original images, and the accuracies and confidence scores were reported against the required outputs. We extracted a csv file that contains the

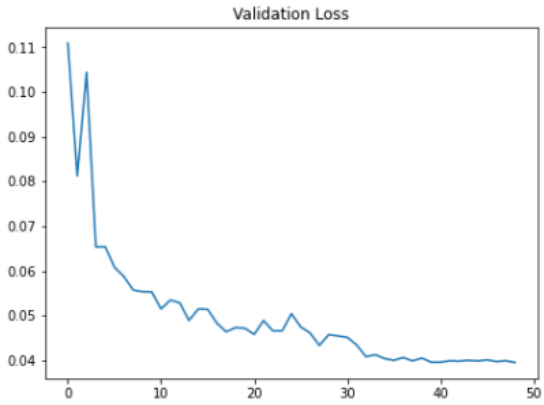
**Visualization**:

Visualizations were created to draw bounding boxes against the original images and report accuracies and confidence scores against the required outputs. The visualizations helped to analyze the performance of the system and identify any errors or issues in the model.




**Loss :**



## SUMMARY:

Overall, the system was able to successfully detect objects of the predefined labels (Drones, Airplane, Birds, Helicopter) in the drone detection dataset with a certain accuracy for a considerable object size. The test results could further be improved by fine tuning the hyper parameters and increasing the diversity of the training data images.