Visual Learning & Recognition- Homework 1
Parth Malpathak (pmalpath)

TASK 0:
Q0.1: Index 2020 contains the **train class(Class number 19 of 20)**

Q0.2:

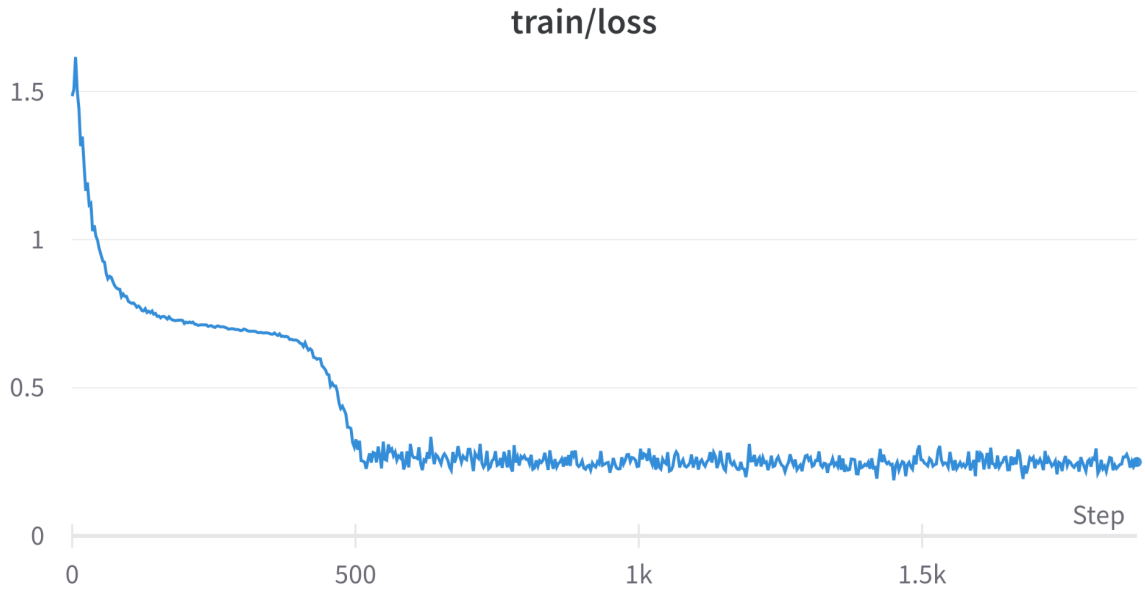
Q0.3:

TASK 1:

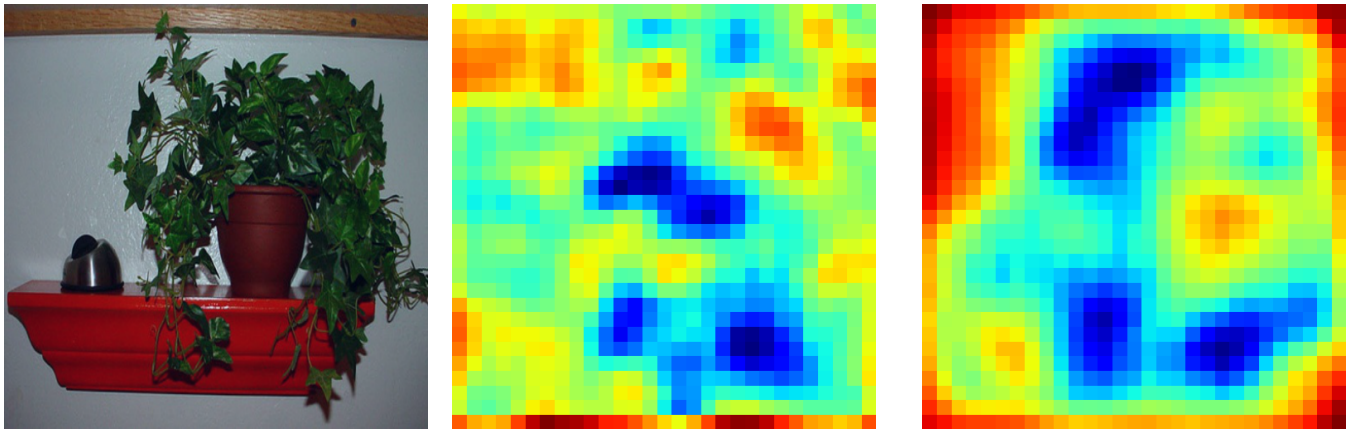
Q1.1: The train function is implemented which deals with traversing through the forward pass and computing the gradients and eventually updating the weights for the model to learn from the data. Similarly, valide function has been implemented which deals with testing of the model on unknown data.

Q1.2: The shape of the output from model for batch size of 32 is (32x20x29x29)

Q1.3: Loss~ 0.23



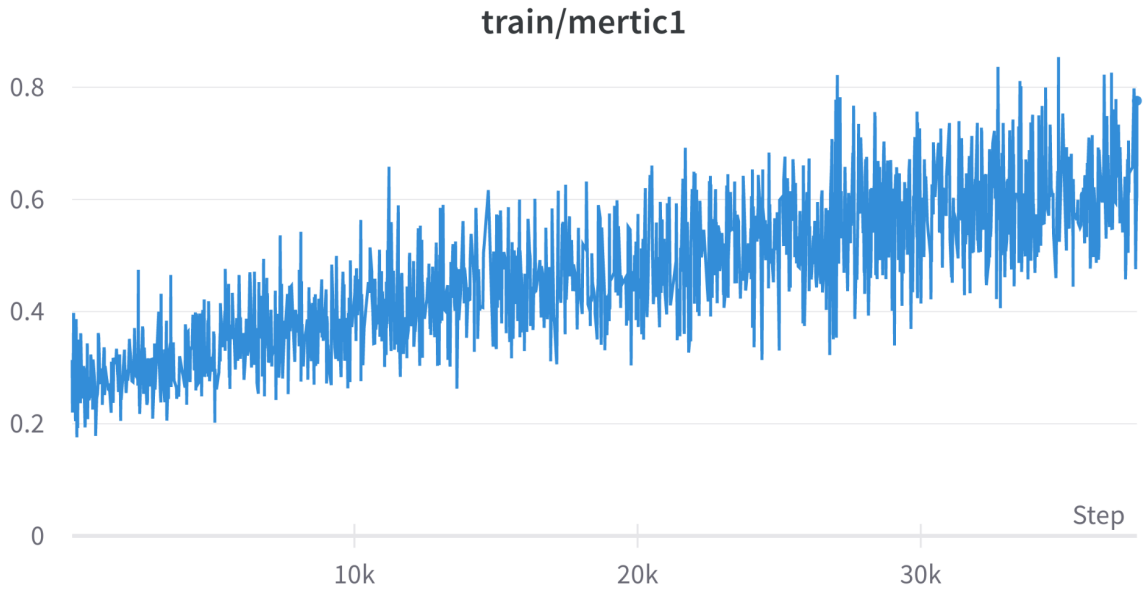
Heatmap visualization:



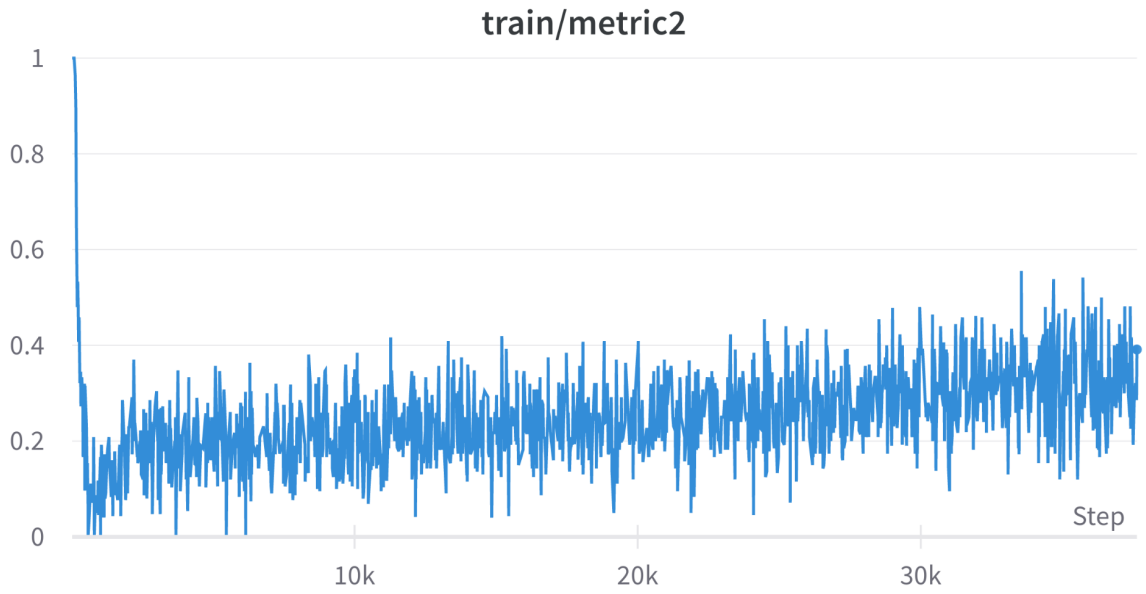
Q1.4: The Labels, as hinted in the question are 0 and 1. After checking the presence of labels in every image, we get to know that every image does not have all the classes present. At the start of training, when the weights have not been updated, we see a higher starting point for the loss but this value of loss reduces quickly as the model learns and generalizes itself with the presence of labels in every image.
Q1.5: The required functions have been written and the required values have been logged. We have assumed that image has labels considering there can be images without labels

Q1.6: Trained for Batch Size 16 for better visualization after getting approval from TAs.
Training Loss:

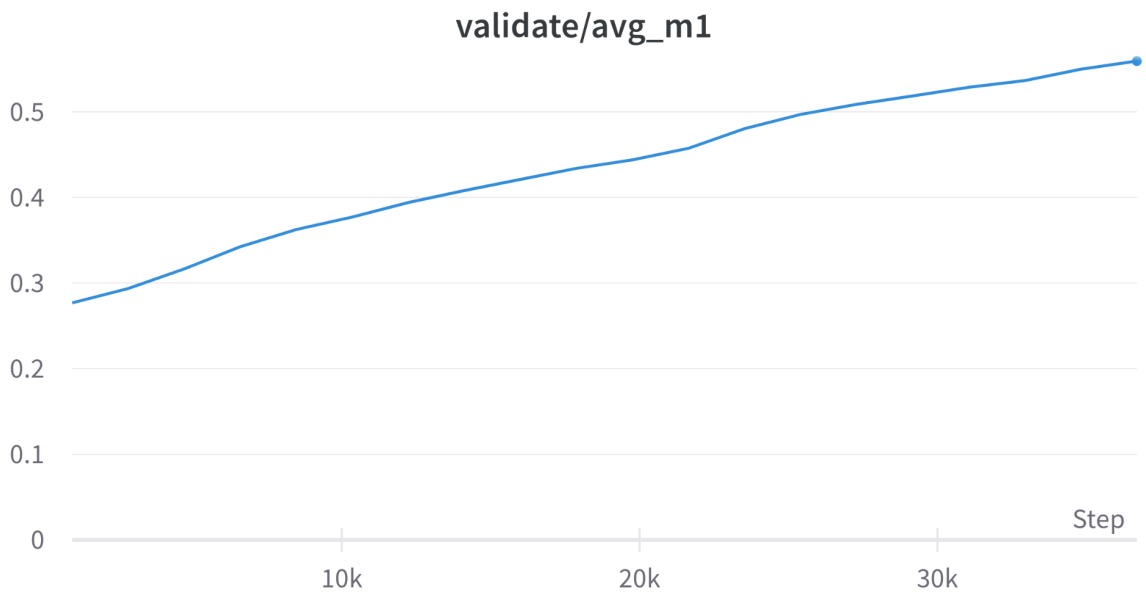
Training Metric 1:



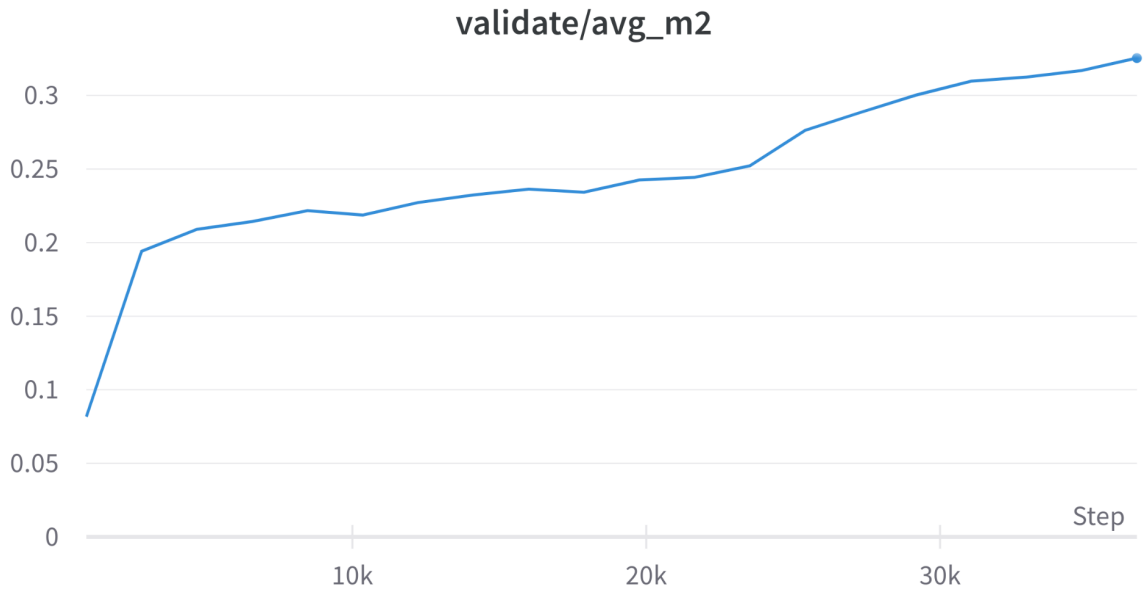
Training- Metric 2:



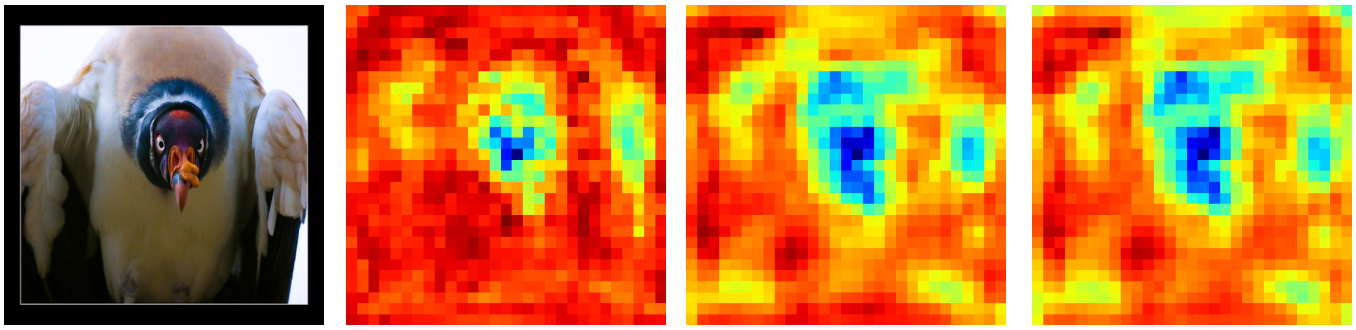
Validation- Average Metric 1:

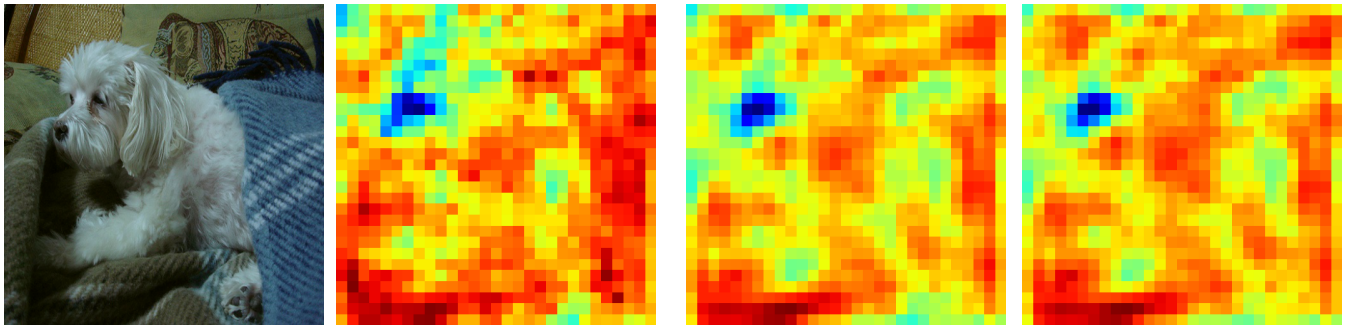


Validation- Average Metric 2:

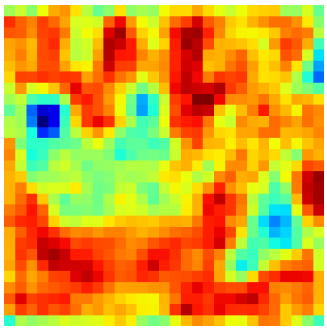


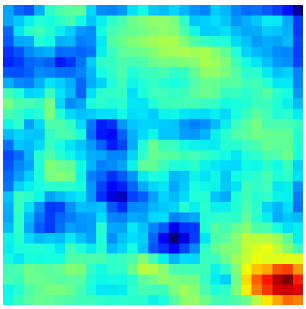
Training Heatmaps (Visualised for Epoch 1, 15, 30):

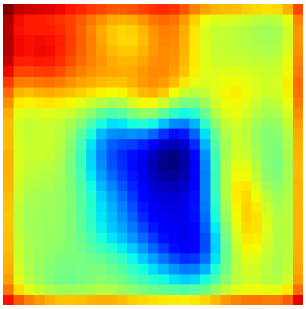




Validation Heatmaps:

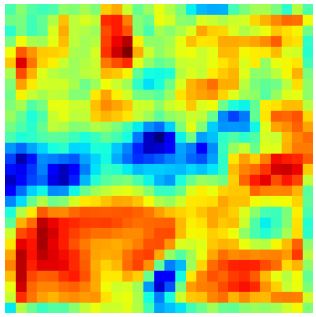


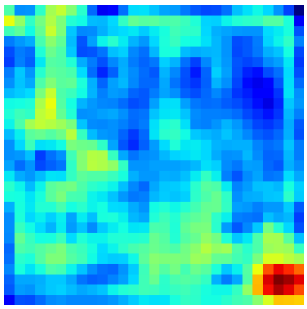



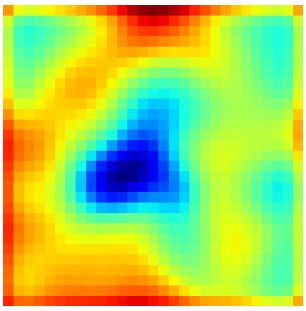



Training Loss at the end of training: 0.1473
Training Metric 1: 0.710
Training Metric 2: 0.429
Validation Metric 1: 0.565
Validation Metric 2: 0.329

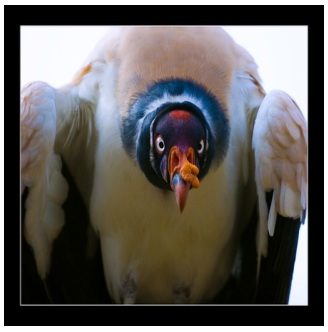
Q1.7:
Validation Heatmaps:

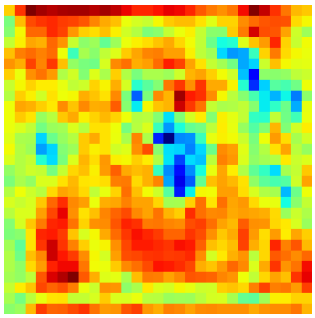



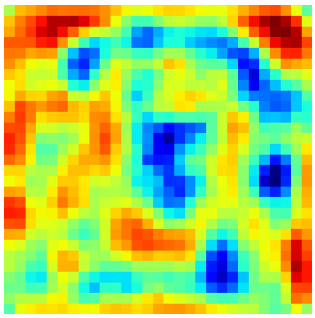


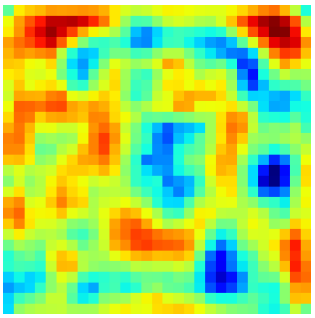


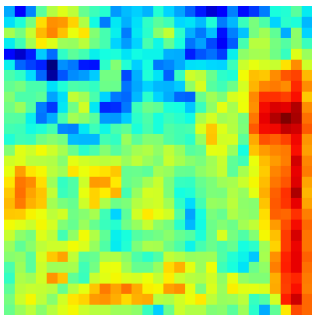
Training Heatmaps:

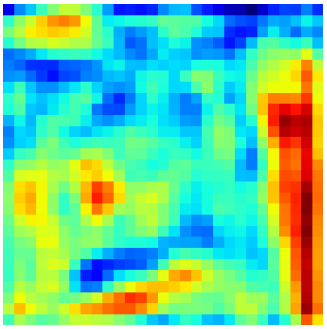


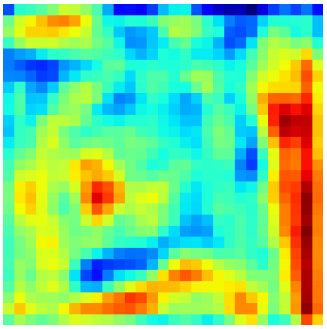












Training Loss: 0.1531
Training Metric 1: 0.573
Training Metric 2: 0.417
Validation Metric 1: 0.563
Validation Metric 2: 0.410

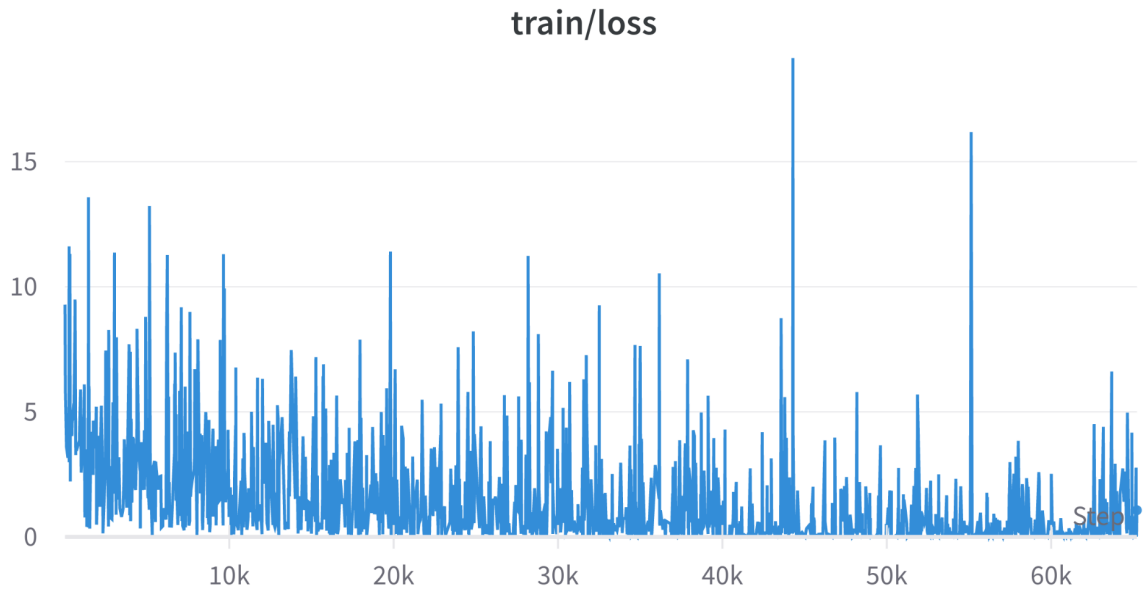
TASK 2:
Q2.3:
Attaching a code snippet

```
# Calculation for AP
for i in range (len(scores)):
    maximum_iou = 0
    selected_rois = boxes[i]

    for j, gt_box in enumerate(gt_boxes):
        if gt_class_list[j] == class_num: #Check for presence of gt_box in image
            recall_den += 1
            gt_roi = gt_box
            intersection_of_union = iou(selected_rois, gt_roi) #Compute IOU
            if intersection_of_union > maximum_iou:
                maximum_iou = intersection_of_union #Substitute iou as maximum if computed value is greater
                detected_gt = j
    if maximum_iou > 0.3: #Condition for True Positive and popping the GTs already taken care of.
        tp += 1
        gt_boxes.pop(detected_gt)
    else:
        fp += 1

precision = tp / (tp + fp) #Formula for Precision
recall = tp / recall_den #Formula for recall
average_precision = precision * recall
average_precision_list[class_num] = average_precision #Find Average Precision and store
```

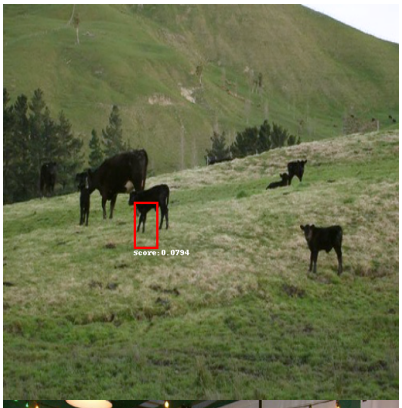
Q2.4:

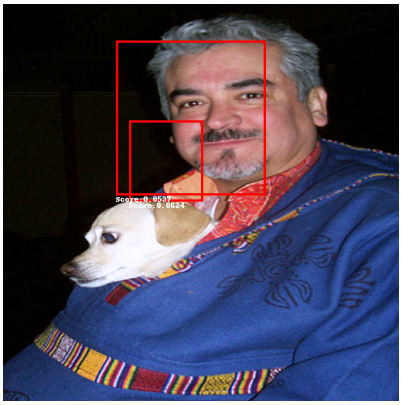


Q2.4: (Tanmay Chinchanikar helped me in understanding the nms plotting as my implementation using wandb and matplotlib was not working. He gave be a detailed explanation of how to plot using ImageDraw.)
First Epoch:

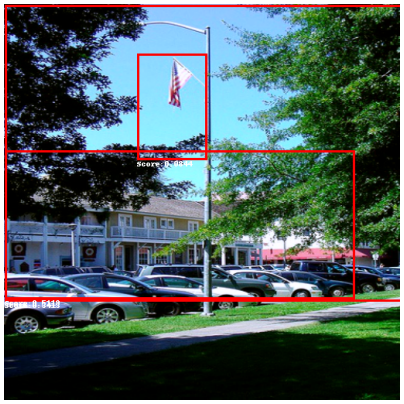




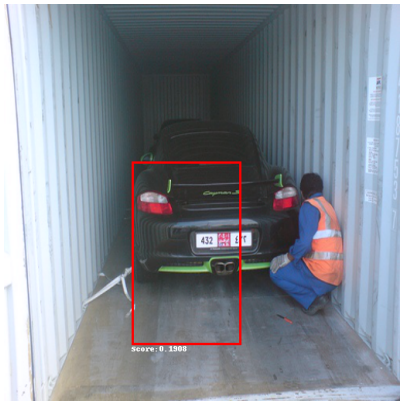



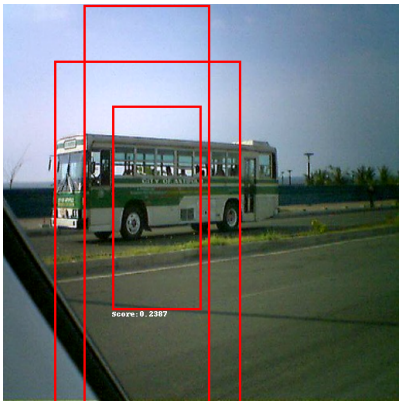


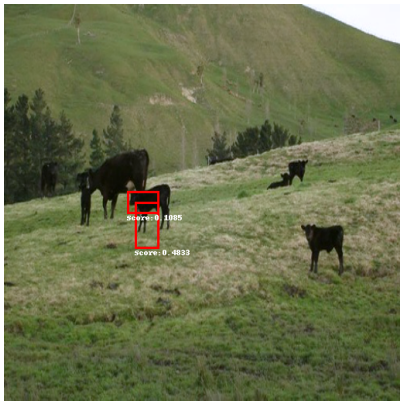

7th Epoch:




Score: 0.2404

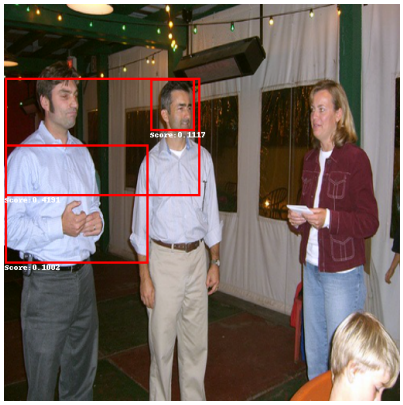

Score: 0.1908


Score: 0.2387


Score: 0.1885
Score: 0.4833


Score: 0.4938


Score: 0.5891


Score: 0.1117
Score: 0.1002


Score: 0.4504
Score: 0.0603