Assignment 4

(Using 4 late days)

Part 1

Ablation study:

Before doing any sort of data augmentation, this is the accuracy percentage with 750 iterations.

AP	AP50	AP75	APs	APm	APl
12.342	30.001	6.025	8.814	16.380	16.674

I then created a custom trainer class called CustomTrainer() instead of using the DefaultTrainer. The followiing data augmentation are defined in the custom_data() function:

```
        T.RandomFlip(prob=0.5, horizontal=False, vertical=True),
        T.RandomFlip(prob=0.5, horizontal=True, vertical=False),
        T.Resize((512,512)),
        T.RandomBrightness(0.3,2.0),
        T.RandomContrast(0.5, 1.5),
```

After adding the data augmentation, the AP50 score rises to 46.266

AP	AP50	AP75	APs	APm	APl
27.863	46.266	31.529	17.200	36.285	60.799

Next, I increased the number of iterations to 1000 with expectation to reduce the loss. The AP50 score increases to 49.22

AP	AP50	AP75	APs	APm	APl
30.514	49.220	35.001	19.611	39.038	62.223

I then take the “bbox" coordinates directly from each image data instead of calculating the bounding box based on the minimum and maximum values of the polygons coordinates, as well

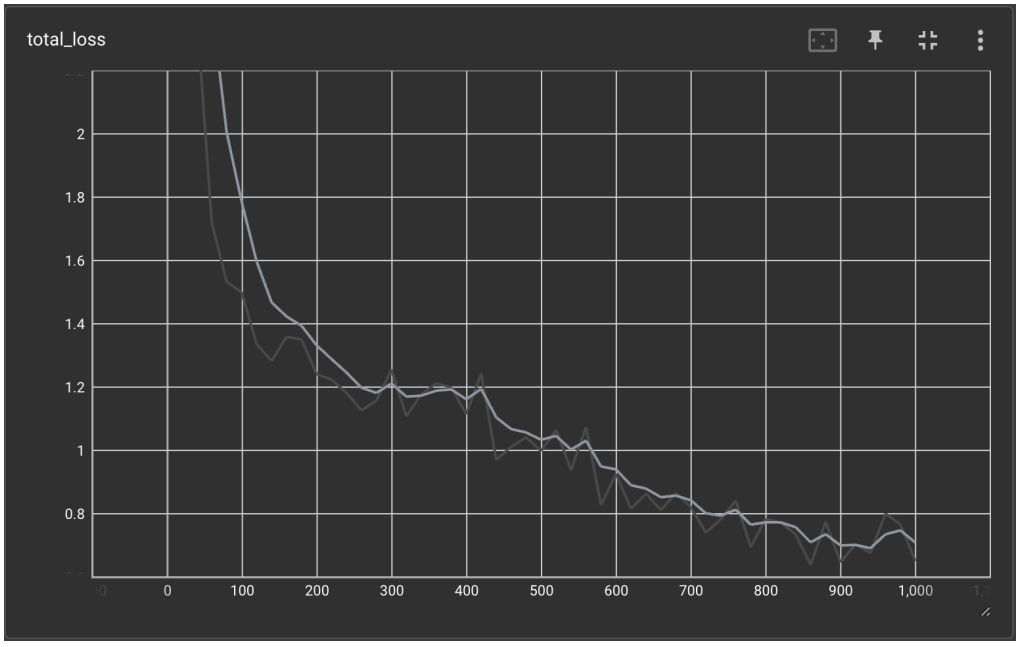
as the trasformation of polygon coordinates (like in the Detectrone2 Tutorial Provided in class)

```
obj = {
        "bbox": v["bbox"],
        "bbox_mode": BoxMode.XYWH_ABS,
        "segmentation": v["segmentation"],
        "category_id": 0,
        "category_name" : v["category_name"],
}
```

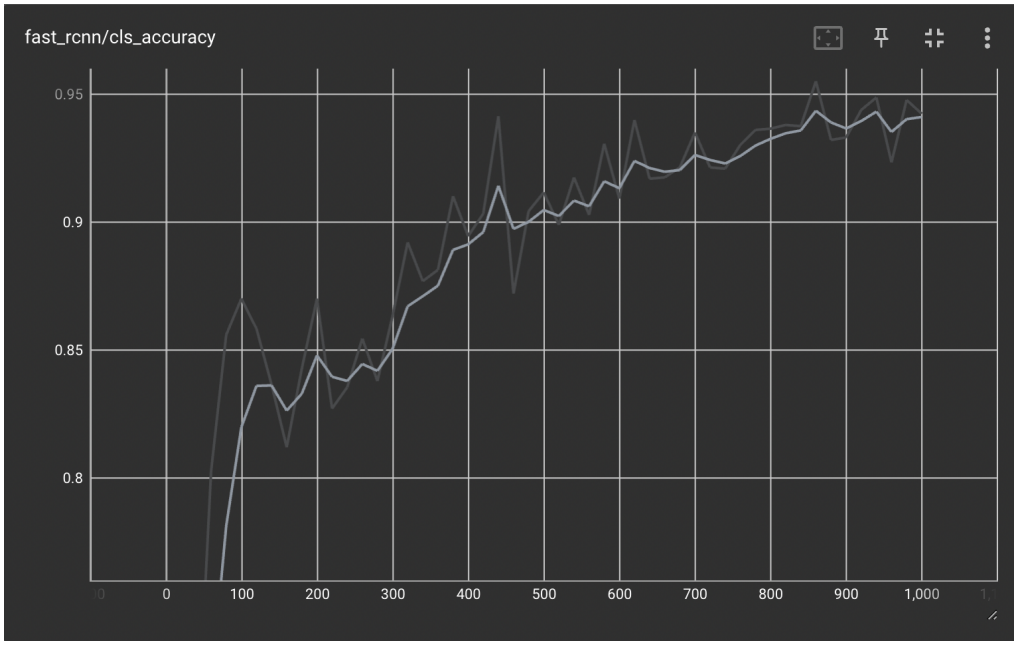
Using these coordinates improved the AP50 score to 51.534

AP	AP50	AP75	APs	APm	APl
30.529	51.534	33.652	20.574	39.179	59.388

Total Loss plot:



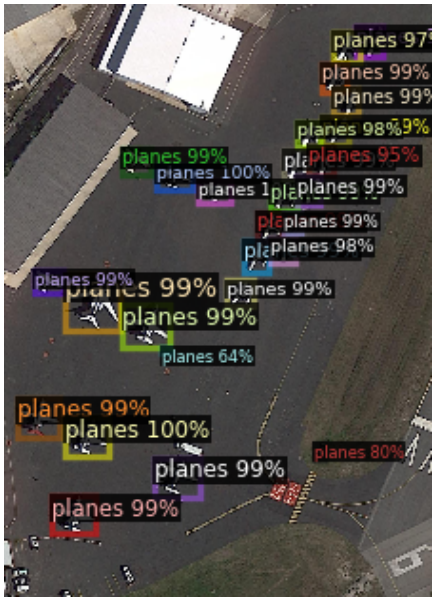
Accuracy Plot



Sample Visualization

Sample 1

Sample 2



Sample 3

Task 2:

Modifications on the hyperparameters:

Batch_size: 25
Learning_rate: 0.01
Num_epochs: 50

Model Architecture:
Based on the based model that was given, I added more downsampling layers and upsampling layers.
Downsampling layers: the self.input_conv accepts a 3-channel input image and processes it with an initial convolutional layer, increasing the number of channels to 8.
For each of the following downsampling, each block reduces the spatial dimensions and doubles the number of channels. The channel progression is as follows: 8, 16, 32, 64, 128, and 256. Increasing the number of channels allows the network to capture more diverse and complex features at each stage of down-sampling.
Upsampling layers: Corresponding up-sampling blocks to recover spatial information. Each block halves the number of channels and doubles the spatial dimensions. The channel progression is: 512, 256, 128, 64, 32, and 16.

```
Epoch: 45, Loss: 0.10224588215351105
100% 320/320 [00:42<00:00, 9.57it/s]
Epoch: 46, Loss: 0.1020212173461914
100% 320/320 [00:43<00:00, 9.24it/s]
Epoch: 47, Loss: 0.10003487765789032
100% 320/320 [00:43<00:00, 10.18it/s]
Epoch: 48, Loss: 0.09960930049419403
100% 320/320 [00:42<00:00, 9.56it/s]
Epoch: 49, Loss: 0.10003183782100677
```

Mean IoU: `#images: 7980, Mean IoU: 0.8230058938003423`

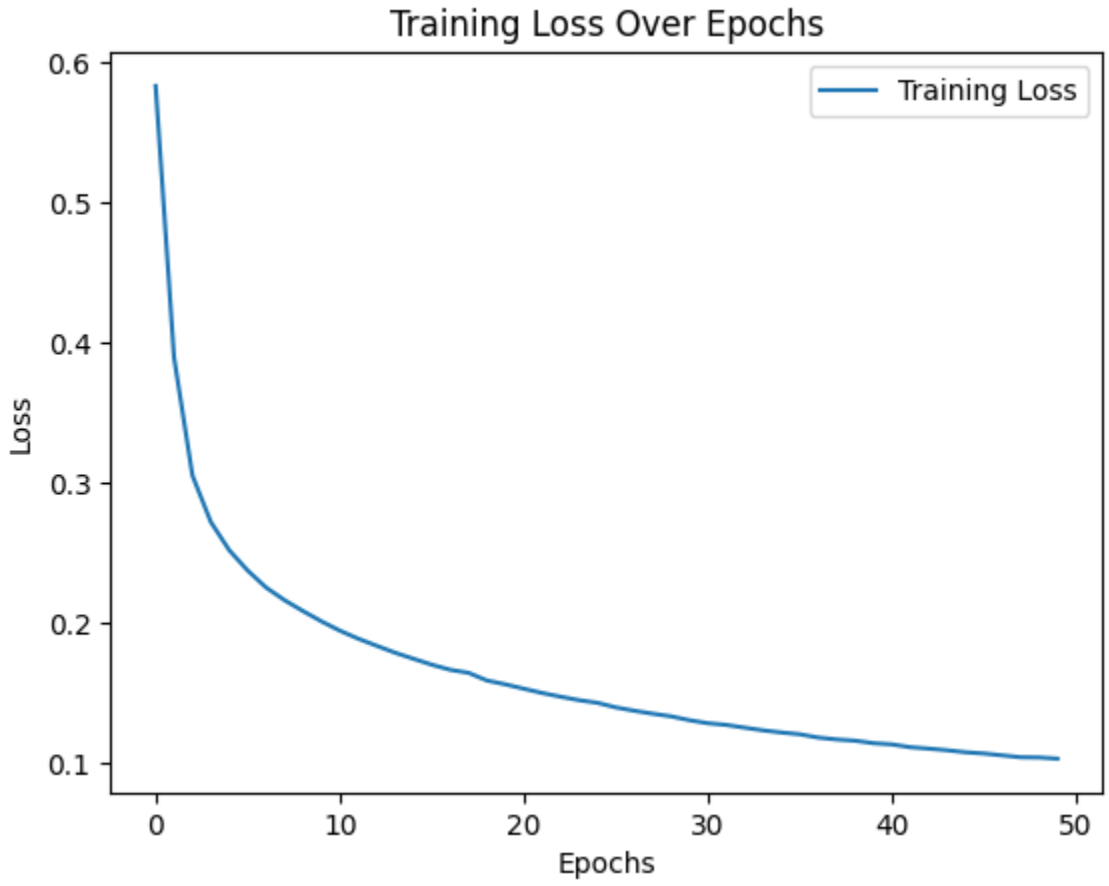
Initially, the training time was extensive. At first run, it took about 50 minutes in to finish one epoch. This is because the model had to read input images from Google Drive for every images. I was able to shorten this to approximately 40 seconds per epoch by declaring an **img_dict** dictionary variable in the **PlaneDataset** class. I then cache all the image into that dictionary. The training only had to read input from that dict instead of reading from Google Drive

Visualization:
Sample 1:

Sample 2:

Sample 3:

Loss Over Epochs Plot



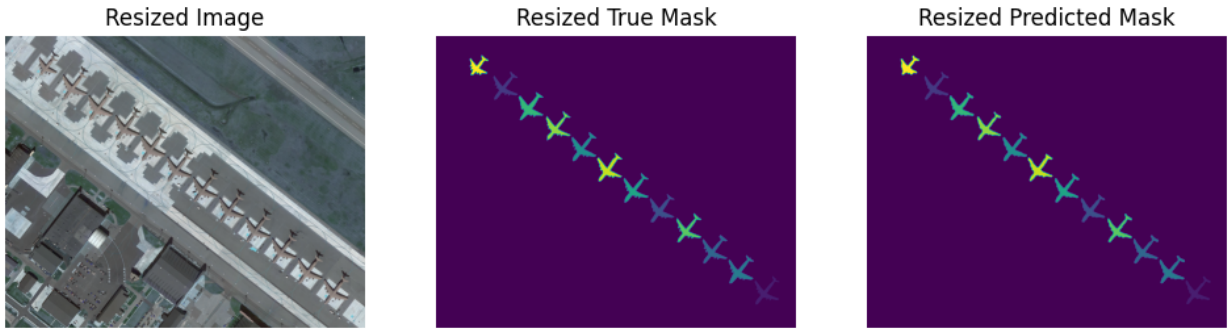
Part 3:

Kaggle Group: John A & Canh Nhat M & Paniz N

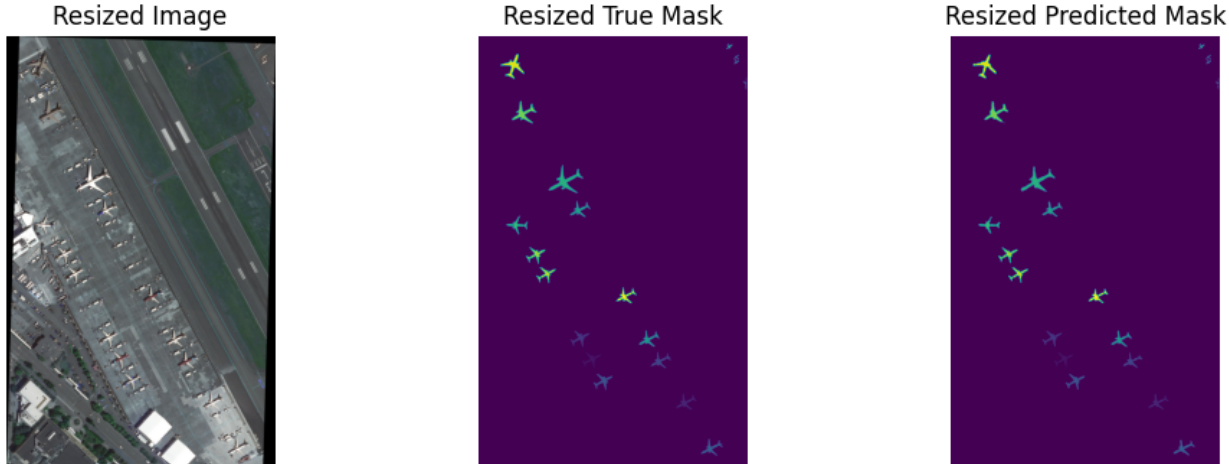
Accuracy P: 0.54532

Sample 1:

Sample 2:



Sample 3:

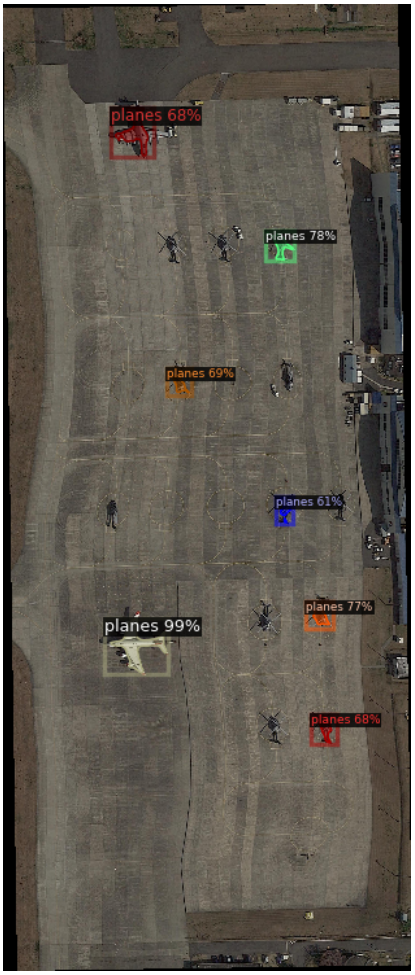


Part 4:

Part 4:

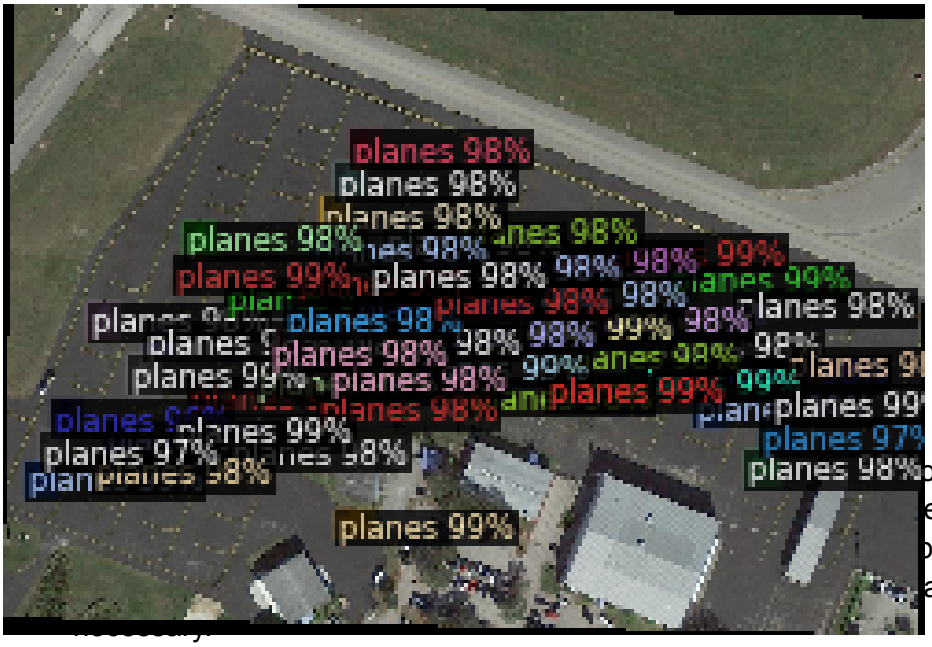
Sample Visualization

Sample 1



Sample 2

Sample 3:



of part 1.
ext of this project and the
previous one because data
ata to speed up training is not

```
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.305
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.515
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.337
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.206
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.392
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.594
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.016
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.127
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.341
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.205
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.432
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.729
[11/13 00:31:01 d2.evaluation.coco_evaluation]: Evaluation results for bbox:
|   AP   |  AP50  |  AP75  |  APs   |  APm   |  APl   |
|:------:|:------:|:------:|:------:|:------:|:------:|
| 30.529 | 51.534 | 33.652 | 20.574 | 39.179 | 59.388 |
```