Project 3

Name: Darryl Basri

Student id: 301388030

Part 1

List of configs used:

```
cfg = get_cfg()
cfg.OUTPUT_DIR = "{}/output/".format(BASE_DIR)
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("plane_train",)
cfg.DATASETS.TEST = ("plane_test",)
cfg.DATALOADER.NUM_WORKERS = 4
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml")#
cfg.SOLVER.IMS_PER_BATCH = 4  # This is the real "batch size" commonly known to deep learning pec
cfg.SOLVER.BASE_LR = 0.0005  # pick a good LR
cfg.SOLVER.MAX_ITER = 1000    # 300 iterations seems good enough for this toy dataset; you will r
cfg.SOLVER.STEPS = []        # do not decay learning rate
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 512   # The "RoIHead batch size". 128 is faster, and g
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1  # only has one class (ballon). (see https://detectron2.readt
```

Configs modifications explained:

- Increase maximum iterations:
    - From the original of 500 iterations, I observed that the loss is still decreasing. So I increased first to 750, then to 1000, where it becomes stable.
- Batch size:
    - Larger batch size will allow us to cover more of the data faster since we have a limited resource.
- Learning rate:
    - Increase learning rate by 0.00025 to converge faster, and also without having decay.

Visualization:

Ablation study:

This model with learning rate 0.00025 and batch size 2, gets a training loss equal to 0.98 at 1000 iterations.

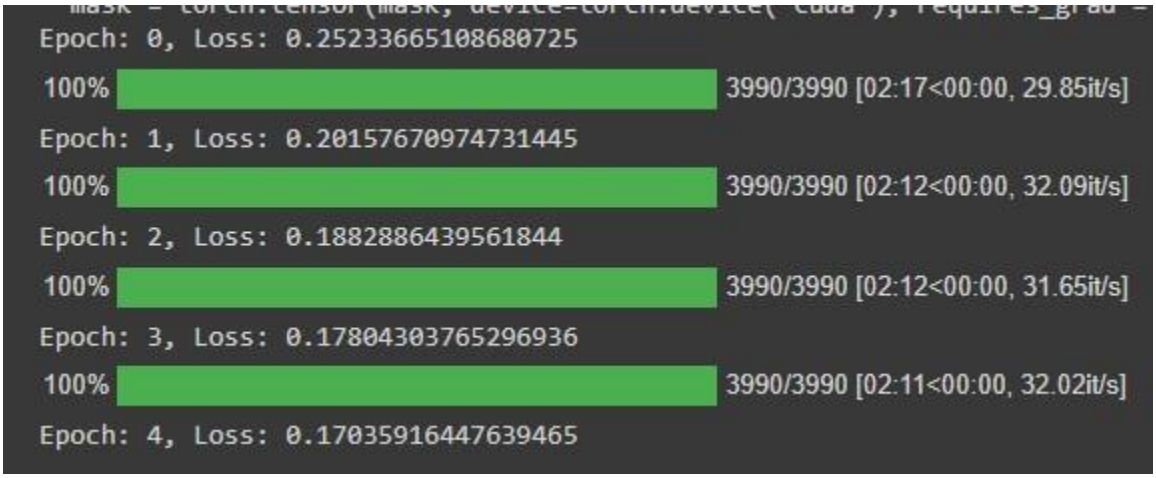
```
cfg = get_cfg()
cfg.OUTPUT_DIR = "{}/output/".format(BASE_DIR)
cfg.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml"))
cfg.DATASETS.TRAIN = ("plane_train",)
cfg.DATASETS.TEST = ("plane_test",)
cfg.DATALOADER.NUM_WORKERS = 4
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_R_101_FPN_3x.yaml") #
cfg.SOLVER.IMS_PER_BATCH = 2  # This is the real "batch size" commonly known to deep learning peop
cfg.SOLVER.BASE_LR = 0.00025  # pick a good LR
cfg.SOLVER.MAX_ITER = 1000    # 300 iterations seems good enough for this toy dataset; you will ne
cfg.SOLVER.STEPS = []        # do not decay learning rate
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 512   # The "RoIHead batch size". 128 is faster, and go
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1  # only has one class (ballon). (see https://detectron2.readth
# NOTE: this config means the number of classes, but a few popular unofficial tutorials incorrect
```

Part 2

Hyperparameter settings used:

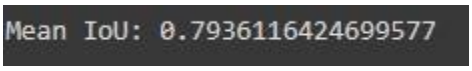
- Learning rate= 0.01
- Epochs= 5
- Batch size = 2
- Weight decay = 0.00001

The model architecture is very similar to the default one, except I used skip layer from resnet which improved the accuracy of the model. I also have 6 downsampling layers and 6 upsampling layers. The

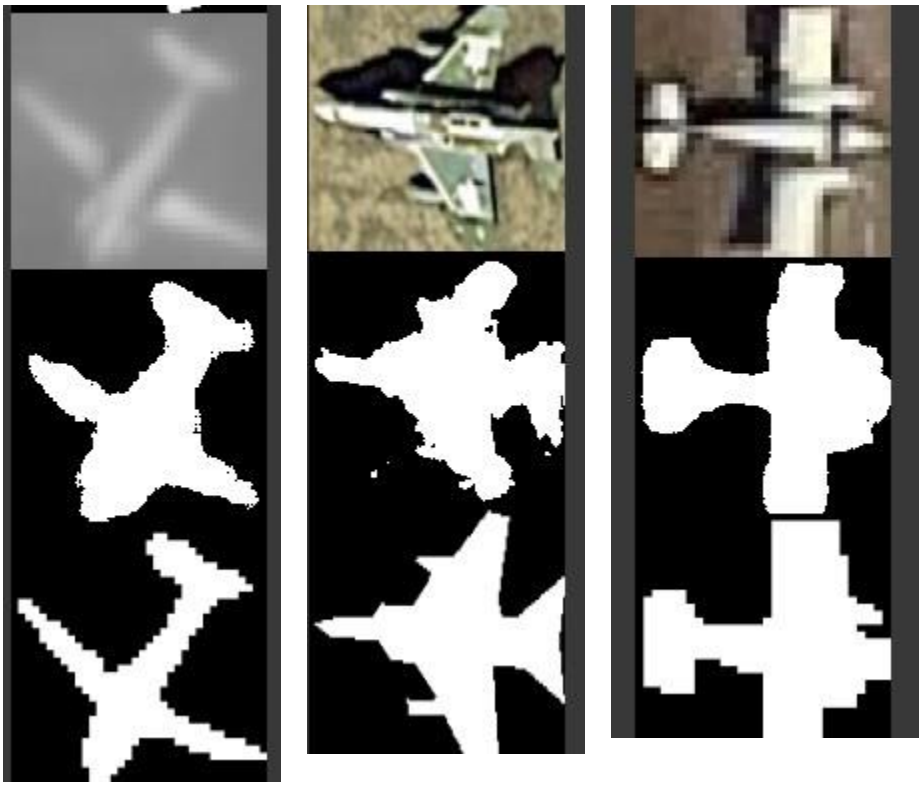


idea is to increase feature channels so detailed features are able to be detected. Below are the training loss :

Final mean IoU of Model:



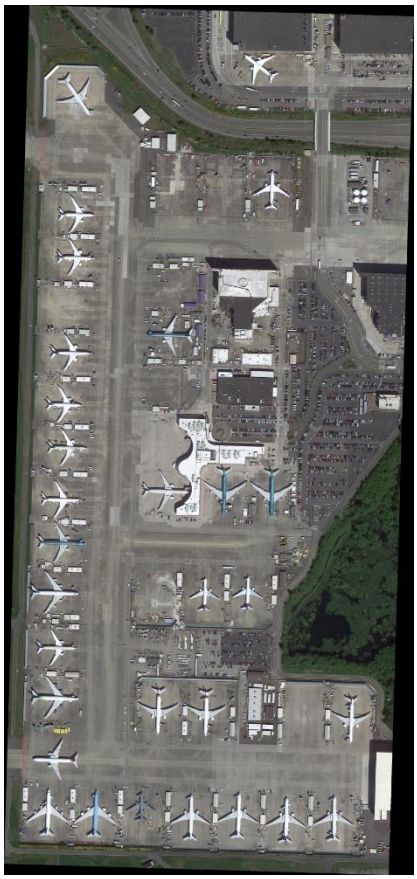
Visualization:

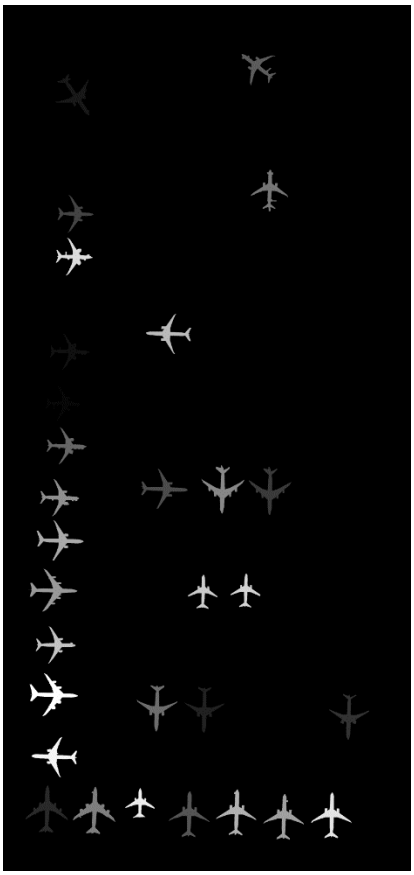


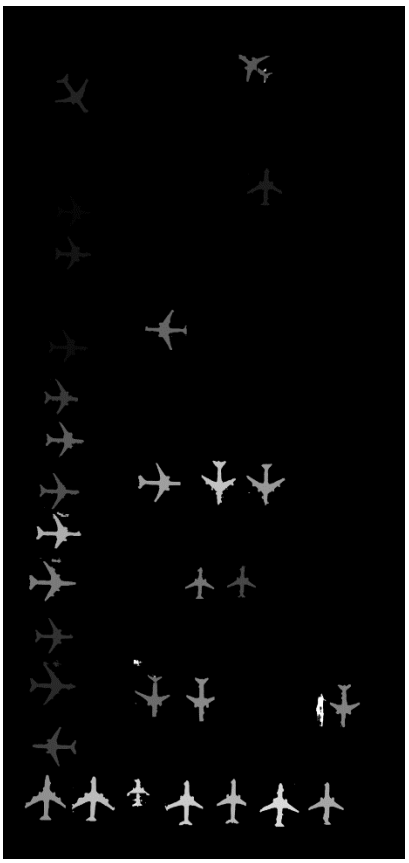
Part 3

Kaggle ID: Darryl Basri

Best score: 0.30024

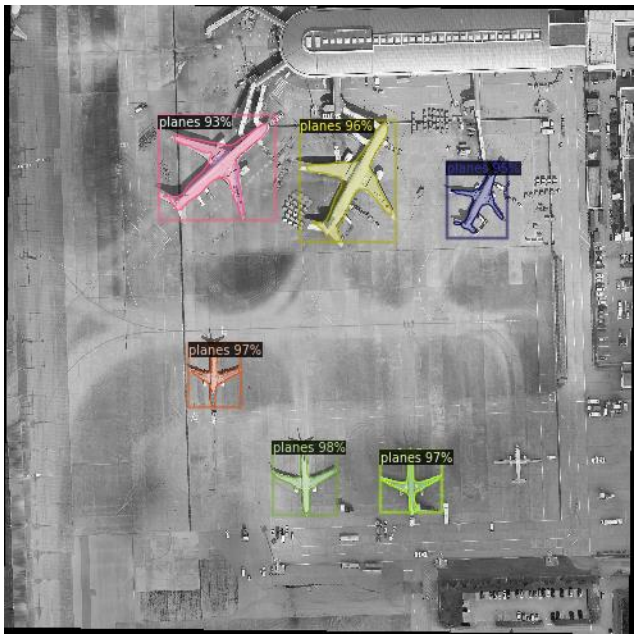


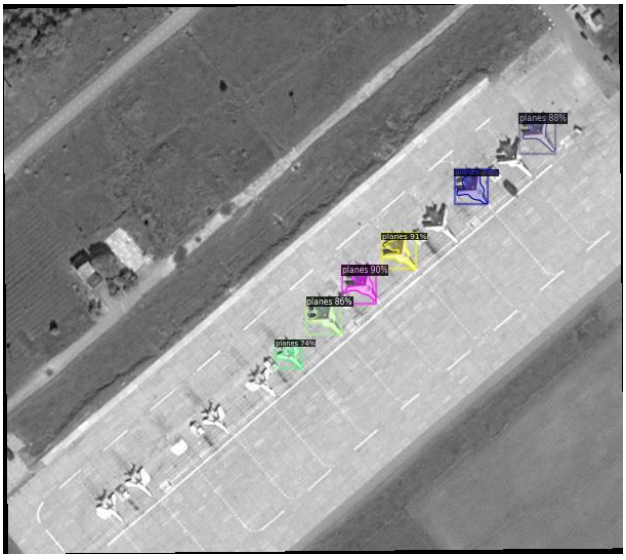




Part 4

Visualization :





The difference between this and part 3 is that it is obvious to see that the result is far less accurate than part 3. However, this model is very light in memory to train for which is useful if resources are limited.