Assignment 3

Siyuan Wu

Student ID:301313026

Kaggle Name: LargeHead Goose

Late day: 1 late day used

Part 1

List of configs

```
# Set the configs for the detection part in here.
# TODO: approx 15 lines
'''
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.merge_from_file(model_zoo.get_config_file("COCO-Detection/faster_rcnn_X_101_32x8d_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")  # Let training initialize from model zoo
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-Detection/faster_rcnn_X_101_32x8d_FPN_3x.yaml")  # Let training initialize from model zoo
cfg.SOLVER.IMS_PER_BATCH = 2
cfg.SOLVER.BASE_LR = 0.00025
cfg.SOLVER.MAX_ITER = 7000
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 512
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1
```

List of modifications and Improving factors

- Increase maximum iterations
    - The original iteration was set to 500, but the loss is still decreasing rapidly. Therefore, higher iterations were tested until total_loss become stable
- Switch the original pretrained model to “faster_rcnn_X_101_32x8d_FPN_3x.yaml”
    - On the Model_ZOO website, it introduces the model has a slightly higher box AP of 43% compared to the original “faster_rcnn_R_101FPN_3x.yaml” model’s 42%
- Employing data augmentations and custom trainer
    - The list of transformations for data augmentation includes random horizontal/vertical flip, random lighting, random contrast, random saturation, and random brightness.
    - To use custom trainer, all image is resized to a fixed size of 512 x 512 so that there would be enough memory to finish the transformation.
    - The data augmentation will be performed 3 times which augmentation happens every 1000 iterations. Applying transformation multiple times can be thought of as increasing training data in some way.

Training loss on the last 400 iterations: loss become stable around 0.44

```
[11/02 10:18:24 d2.utils.events]:  eta: 0:03:55  iter: 7119  total_loss: 0.533  loss_cls: 0.091  loss_box_reg: 0.205  loss_rpn_cls: 0.018  loss_rpn_loc: 0.119  time: 0.8996  data_time: 0.4987  lr: 0.000500  max_mem: 7174M
[11/02 10:18:39 d2.utils.events]:  eta: 0:03:27  iter: 7139  total_loss: 0.288  loss_cls: 0.047  loss_box_reg: 0.132  loss_rpn_cls: 0.006  loss_rpn_loc: 0.084  time: 0.8216  data_time: 0.2618  lr: 0.000500  max_mem: 7174M
[11/02 10:18:56 d2.utils.events]:  eta: 0:03:18  iter: 7159  total_loss: 0.457  loss_cls: 0.088  loss_box_reg: 0.228  loss_rpn_cls: 0.010  loss_rpn_loc: 0.137  time: 0.8243  data_time: 0.3164  lr: 0.000500  max_mem: 7452M
[11/02 10:19:12 d2.utils.events]:  eta: 0:03:10  iter: 7179  total_loss: 0.356  loss_cls: 0.076  loss_box_reg: 0.188  loss_rpn_cls: 0.012  loss_rpn_loc: 0.106  time: 0.8121  data_time: 0.2582  lr: 0.000500  max_mem: 10477M
[11/02 10:19:29 d2.utils.events]:  eta: 0:02:59  iter: 7199  total_loss: 0.402  loss_cls: 0.077  loss_box_reg: 0.212  loss_rpn_cls: 0.010  loss_rpn_loc: 0.110  time: 0.8289  data_time: 0.3520  lr: 0.000500  max_mem: 10477M
[11/02 10:19:47 d2.utils.events]:  eta: 0:02:47  iter: 7219  total_loss: 0.488  loss_cls: 0.088  loss_box_reg: 0.230  loss_rpn_cls: 0.008  loss_rpn_loc: 0.122  time: 0.8415  data_time: 0.3840  lr: 0.000500  max_mem: 10477M
[11/02 10:20:03 d2.utils.events]:  eta: 0:02:35  iter: 7239  total_loss: 0.359  loss_cls: 0.065  loss_box_reg: 0.169  loss_rpn_cls: 0.015  loss_rpn_loc: 0.096  time: 0.8325  data_time: 0.2794  lr: 0.000500  max_mem: 10477M
[11/02 10:20:22 d2.utils.events]:  eta: 0:02:22  iter: 7259  total_loss: 0.351  loss_cls: 0.062  loss_box_reg: 0.171  loss_rpn_cls: 0.017  loss_rpn_loc: 0.110  time: 0.8479  data_time: 0.4430  lr: 0.000500  max_mem: 10477M
[11/02 10:20:37 d2.utils.events]:  eta: 0:02:11  iter: 7279  total_loss: 0.373  loss_cls: 0.065  loss_box_reg: 0.188  loss_rpn_cls: 0.011  loss_rpn_loc: 0.096  time: 0.8334  data_time: 0.2259  lr: 0.000500  max_mem: 10477M
[11/02 10:20:51 d2.utils.events]:  eta: 0:01:58  iter: 7299  total_loss: 0.307  loss_cls: 0.059  loss_box_reg: 0.171  loss_rpn_cls: 0.003  loss_rpn_loc: 0.085  time: 0.8205  data_time: 0.1728  lr: 0.000500  max_mem: 10477M
[11/02 10:21:07 d2.utils.events]:  eta: 0:01:47  iter: 7319  total_loss: 0.441  loss_cls: 0.081  loss_box_reg: 0.170  loss_rpn_cls: 0.023  loss_rpn_loc: 0.123  time: 0.8194  data_time: 0.3180  lr: 0.000500  max_mem: 10477M
[11/02 10:21:24 d2.utils.events]:  eta: 0:01:35  iter: 7339  total_loss: 0.349  loss_cls: 0.078  loss_box_reg: 0.184  loss_rpn_cls: 0.010  loss_rpn_loc: 0.102  time: 0.8217  data_time: 0.3342  lr: 0.000500  max_mem: 10477M
[11/02 10:21:40 d2.utils.events]:  eta: 0:01:23  iter: 7359  total_loss: 0.357  loss_cls: 0.076  loss_box_reg: 0.186  loss_rpn_cls: 0.012  loss_rpn_loc: 0.093  time: 0.8202  data_time: 0.2745  lr: 0.000500  max_mem: 10477M
[11/02 10:21:57 d2.utils.events]:  eta: 0:01:11  iter: 7379  total_loss: 0.382  loss_cls: 0.077  loss_box_reg: 0.189  loss_rpn_cls: 0.010  loss_rpn_loc: 0.108  time: 0.8229  data_time: 0.3538  lr: 0.000500  max_mem: 10839M
[11/02 10:22:15 d2.utils.events]:  eta: 0:00:59  iter: 7399  total_loss: 0.337  loss_cls: 0.069  loss_box_reg: 0.188  loss_rpn_cls: 0.013  loss_rpn_loc: 0.122  time: 0.8286  data_time: 0.3838  lr: 0.000500  max_mem: 10839M
[11/02 10:22:32 d2.utils.events]:  eta: 0:00:47  iter: 7419  total_loss: 0.423  loss_cls: 0.075  loss_box_reg: 0.221  loss_rpn_cls: 0.013  loss_rpn_loc: 0.104  time: 0.8284  data_time: 0.3000  lr: 0.000500  max_mem: 10839M
[11/02 10:22:45 d2.utils.events]:  eta: 0:00:35  iter: 7439  total_loss: 0.386  loss_cls: 0.069  loss_box_reg: 0.195  loss_rpn_cls: 0.008  loss_rpn_loc: 0.095  time: 0.8174  data_time: 0.1448  lr: 0.000500  max_mem: 10839M
[11/02 10:23:04 d2.utils.events]:  eta: 0:00:24  iter: 7459  total_loss: 0.416  loss_cls: 0.078  loss_box_reg: 0.200  loss_rpn_cls: 0.010  loss_rpn_loc: 0.106  time: 0.8261  data_time: 0.4484  lr: 0.000500  max_mem: 10839M
[11/02 10:23:21 d2.utils.events]:  eta: 0:00:12  iter: 7479  total_loss: 0.442  loss_cls: 0.094  loss_box_reg: 0.216  loss_rpn_cls: 0.013  loss_rpn_loc: 0.104  time: 0.8273  data_time: 0.3286  lr: 0.000500  max_mem: 10839M
```

Final accuracy for the AP50: 69.3

AP	AP50	AP75	APs	APm	APl
47.833	69.303	55.719	38.138	54.952	80.068

Visualization of test samples

Test sample 1

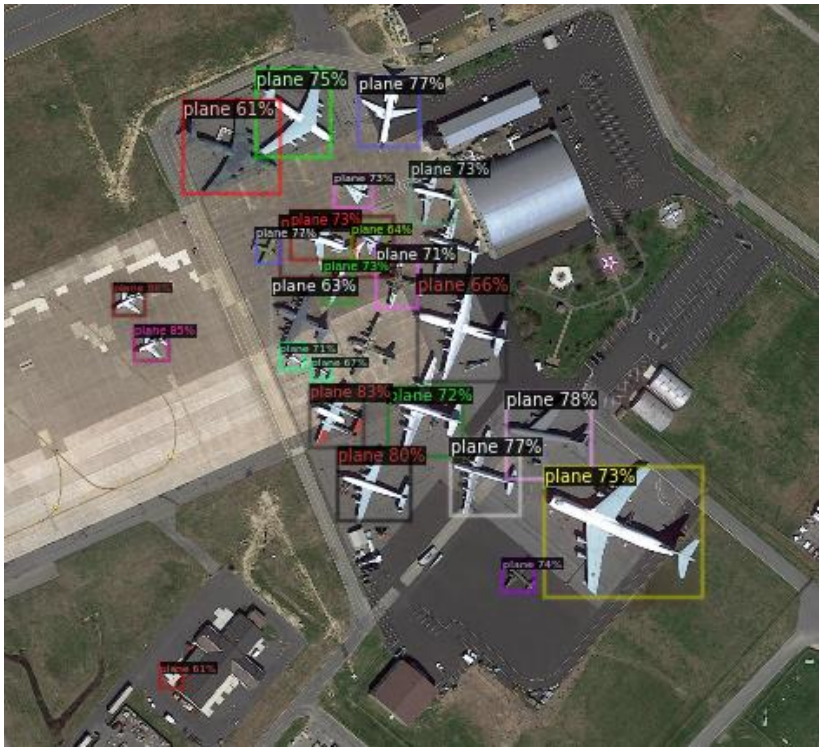
Test sample 2

Test sample 3

Ablation study

- Effect of increasing maximum iterations from 500 to 1500
    - AP50 increase from 0.33 to 0.56

- Visualization on 500 max iterations

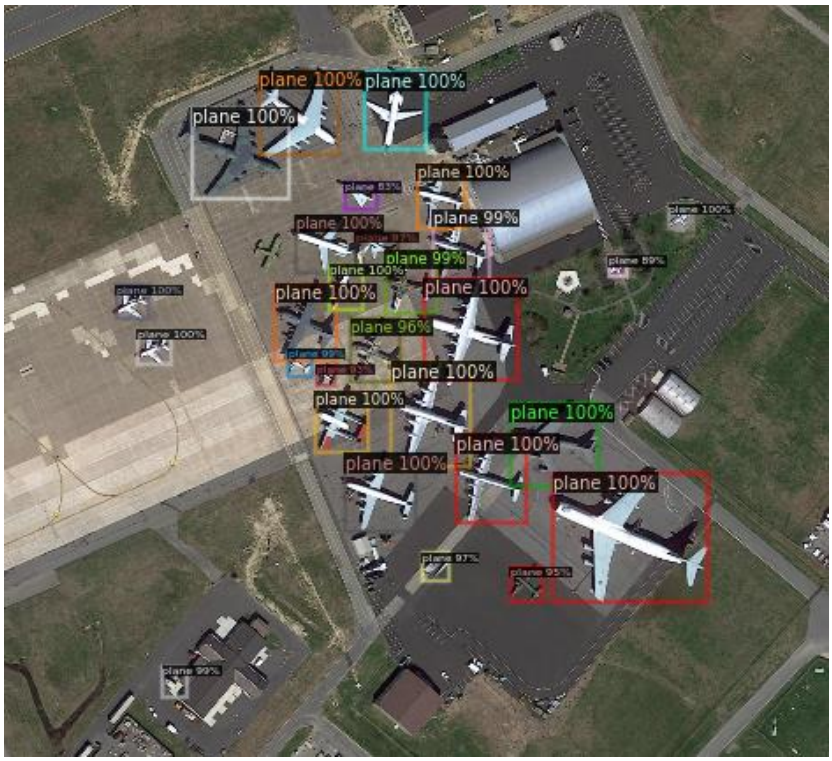


- Visualization on 1500 max iterations

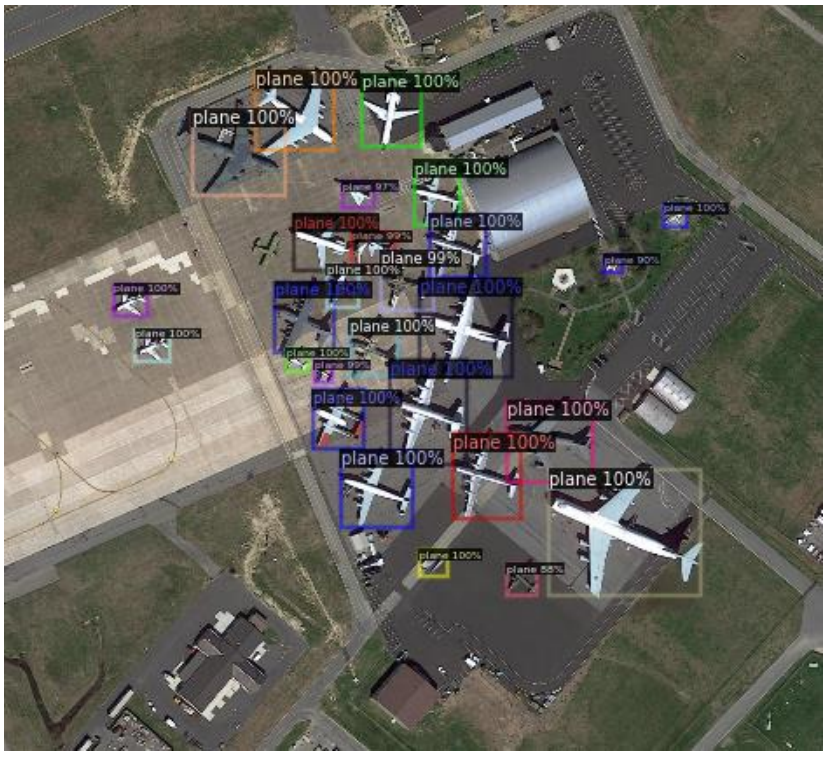
- Effect of using different pretrain model with 1500 max iterations
    - AP50 increase from original model's 0.56 to newer model's 0.59
    - Visualization on using original pretrain model

- Visualization on using different pretrain model

- Effect of employing data augmentations on 7000 max iterations
    - AP50 increase from 0.68 to 0.69
    - Visualization on no data augmentations training with 7000 max iterations



- Visualization on training with augmentations with 7000 max iterations



Part 2

Hyperparameter setting

- Batch size:4
- Learning rate: 0.001
- Number of epochs: 100
- Optimizer: Adam optimizer

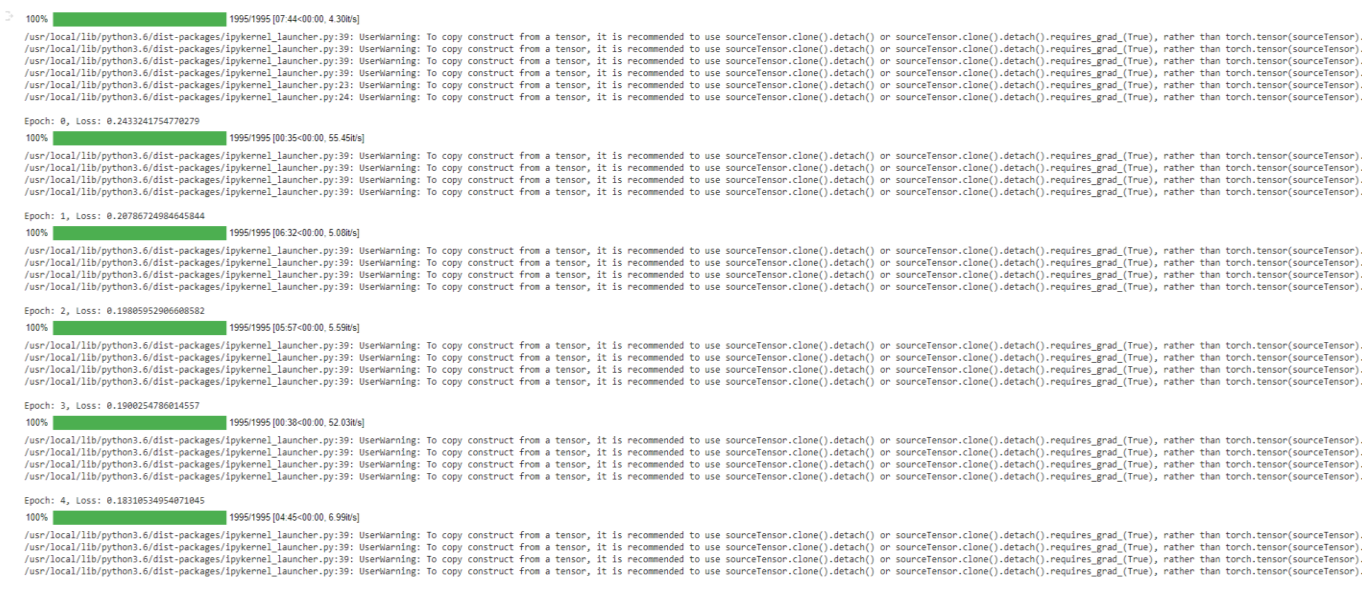
Net architecture

For net architecture, I expand the original architecture by adding 3 more down and upsampling layers. For each downsampling layers, the output channel will have a doubled size of the input channel, also, I added one more conv layer in the downsampling layer. The general idea is to increase feature channels so that detailed features can be detected. I used Adam optimizer instead of the original SGD optimizer because many pieces of research state that it converges faster than SGD. The number of epochs is optimal at 100 as the loss becomes stable at around 90 epochs.

Loss function and training loss

- The loss function is the default loss function, nn.BCEWithLogitsLoss()
- The loss decreases from 0.243 at epoch 1 to 0.099 at epoch 250. Below are the two images displaying the training loss of the first 5 and last 5 epochs.

Training loss of first 5 epochs



```
100% 1995/1995 [07:44<00:00, 4.30it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:23: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:24: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 0, Loss: 0.2433241754770279
100% 1995/1995 [00:35<00:00, 55.45it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 1, Loss: 0.20786724984645844
100% 1995/1995 [06:32<00:00, 5.08it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 2, Loss: 0.19805952906608582
100% 1995/1995 [05:57<00:00, 5.59it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 3, Loss: 0.1900254786014557
100% 1995/1995 [00:38<00:00, 52.03it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 4, Loss: 0.18310534954071045
100% 1995/1995 [04:45<00:00, 6.99it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 5, Loss: 0.17610149085521698
```

Training loss of last 5 epochs

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 245, Loss: 0.09835249185562134
100% 1995/1995 [00:37<00:00, 53.27it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 246, Loss: 0.09795171767473221
100% 1995/1995 [00:37<00:00, 53.07it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 247, Loss: 0.0980483889579773
100% 1995/1995 [00:37<00:00, 53.22it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 248, Loss: 0.0970105454325676
100% 1995/1995 [00:37<00:00, 52.72it/s]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:39: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.clone().detach() or sourceTensor.clone().detach().requires_grad_(True), rather than torch.tensor(sourceTensor).

Epoch: 249, Loss: 0.09932919591665268
```

Final mean IoU of Model: 0.866

```
IoU: 0.8657255636101202
```

Test plane image and predicted masks

1.

2.

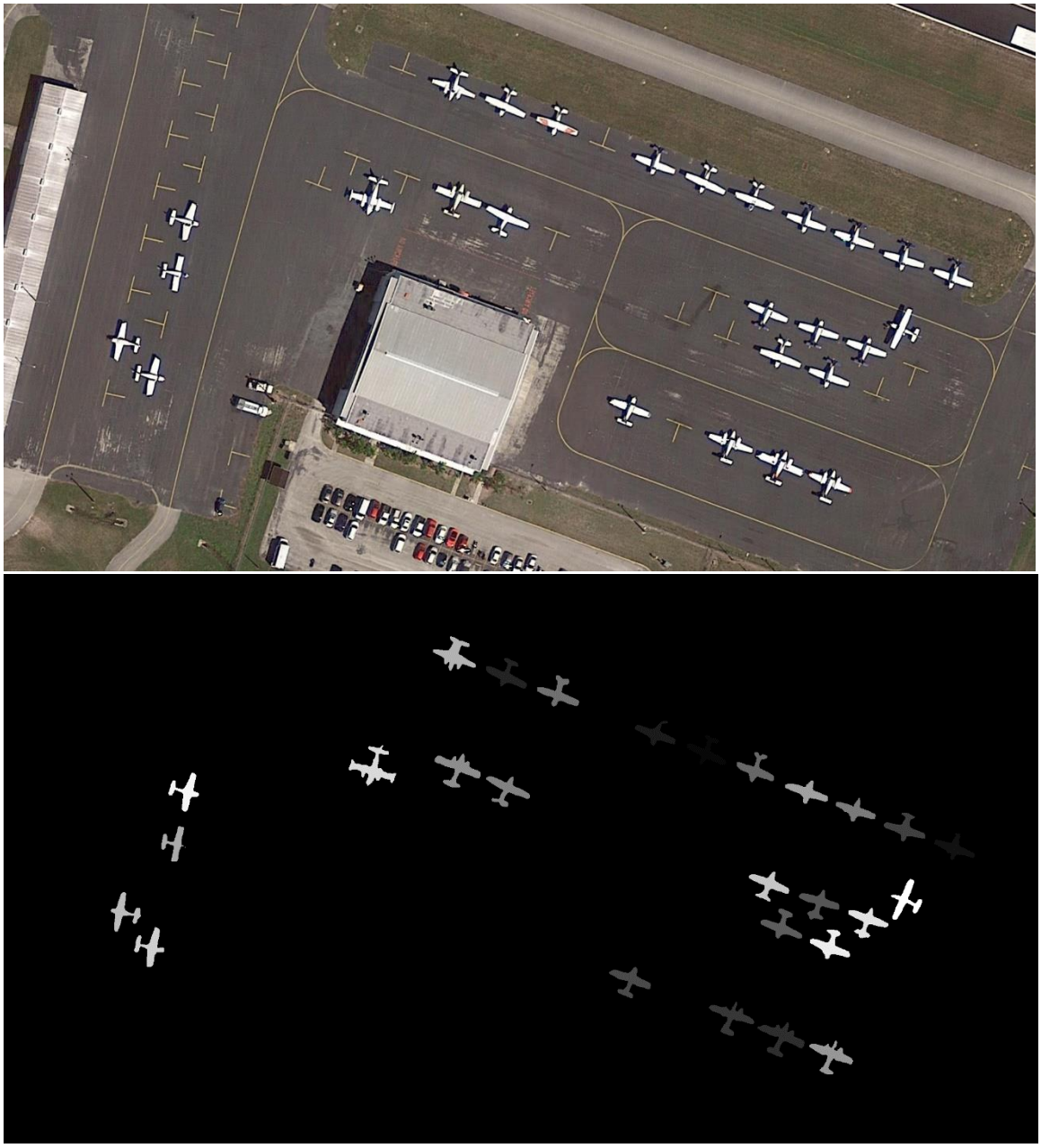
3.

Part 3

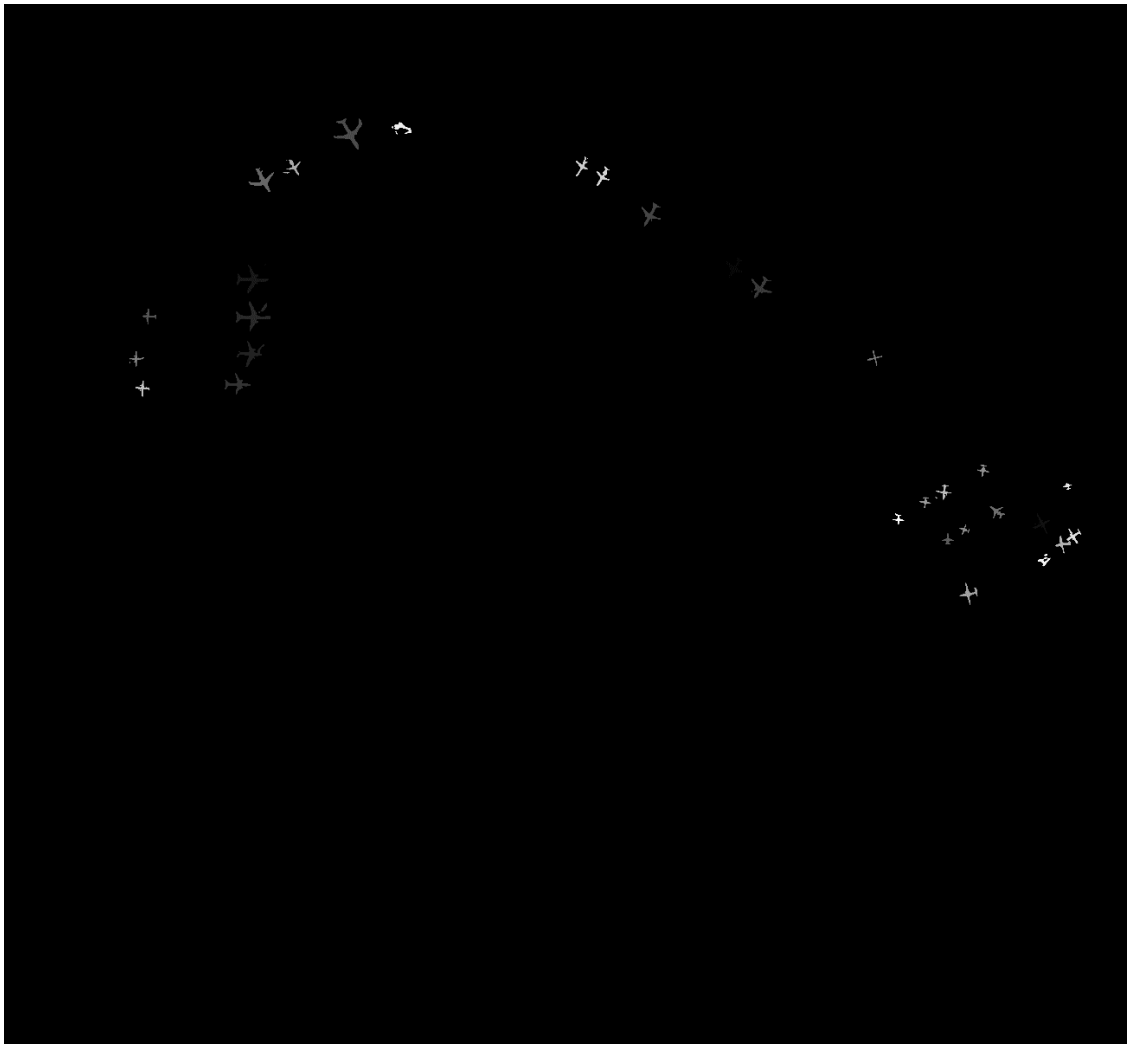
Kaggle ID: LargeHead Goose

Best score: 0.78351

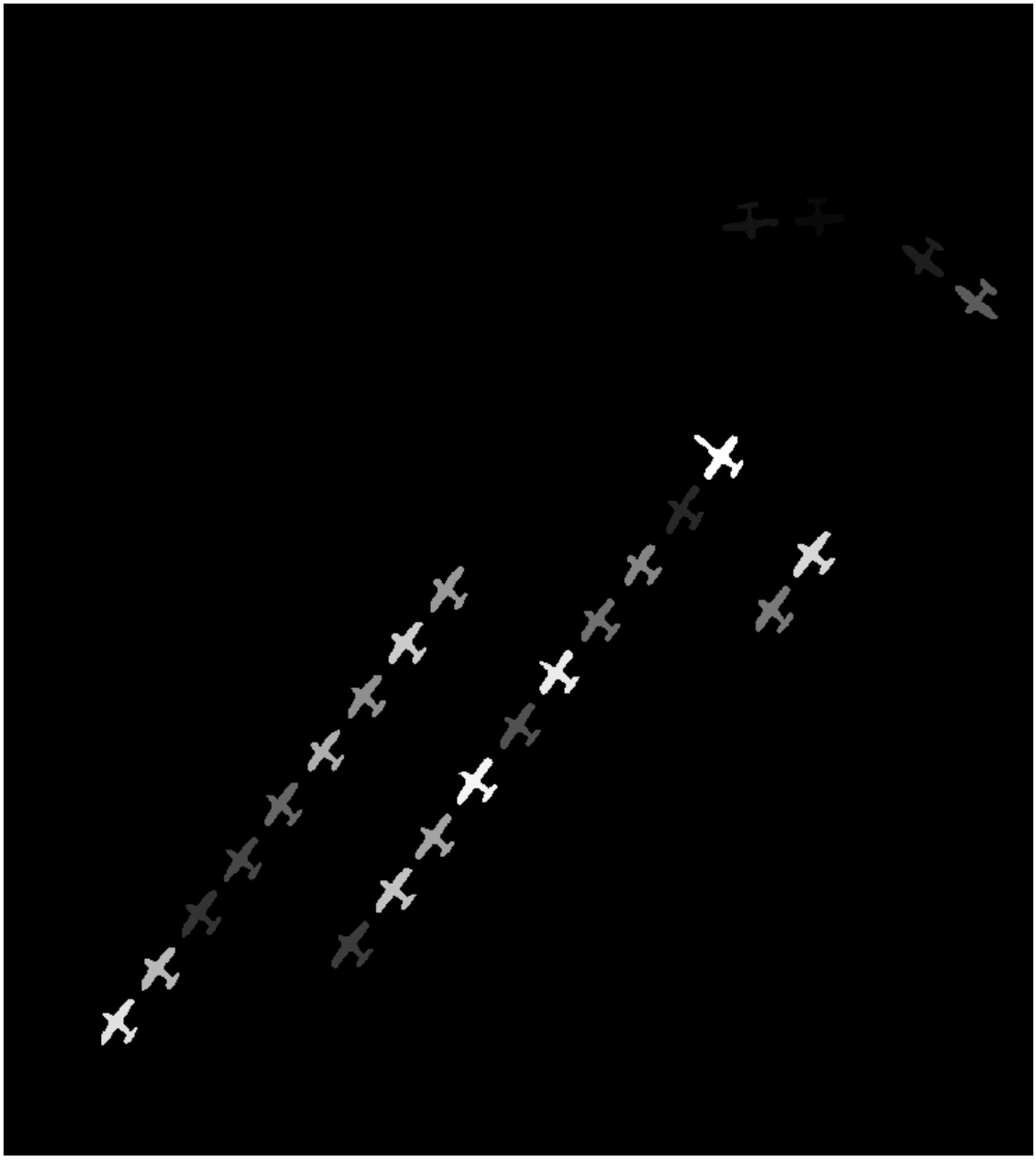
1.



2.



3.



Part 4

Evaluation:

```
[11/02 22:50:31 d2.evaluation.coco_evaluation]: Evaluation results for bbox:
|   AP   |  AP50  |  AP75  |  APs   |  APm   |  APl   |
|:------:|:------:|:------:|:------:|:------:|:------:|
| 32.413 | 54.795 | 35.281 | 22.247 | 41.597 | 61.102 |
Loading and preparing results...
DONE (t=0.15s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *segm*
COCOeval_opt.evaluate() finished in 1.15 seconds.
Accumulating evaluation results...
COCOeval_opt.accumulate() finished in 0.02 seconds.
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.099
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.333
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.022
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.053
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.114
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.351
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.008
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.060
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.131
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.069
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.159
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.407
[11/02 22:50:32 d2.evaluation.coco_evaluation]: Evaluation results for segm:
|  AP   |  AP50  |  AP75  |  APs  |  APm   |  APl   |
|:-----:|:------:|:------:|:-----:|:------:|:------:|
| 9.918 | 33.327 | 2.176  | 5.337 | 11.391 | 35.120 |
OrderedDict([('bbox', {'AP': 32.41296940155823, 'AP50': 54.794504093292616, 'AP75':
```

Visualization

Difference from part 3

From the evaluation output, the AP50 of this model for the bounding box is 54.8, which has roughly the same accuracy for the bounding box as part 1 using the same configuration. However, in part 3, the mean IoU for semantic segmentation is 86, but this model only has AP50 at 33, which indicates that the result obtained is far less accurate than the result from part 3. Nevertheless, the advantage of using this model is that it saves time to train the net, and it also doesn't need images to be cached in memory which saves computing resources.