

Plane Detection

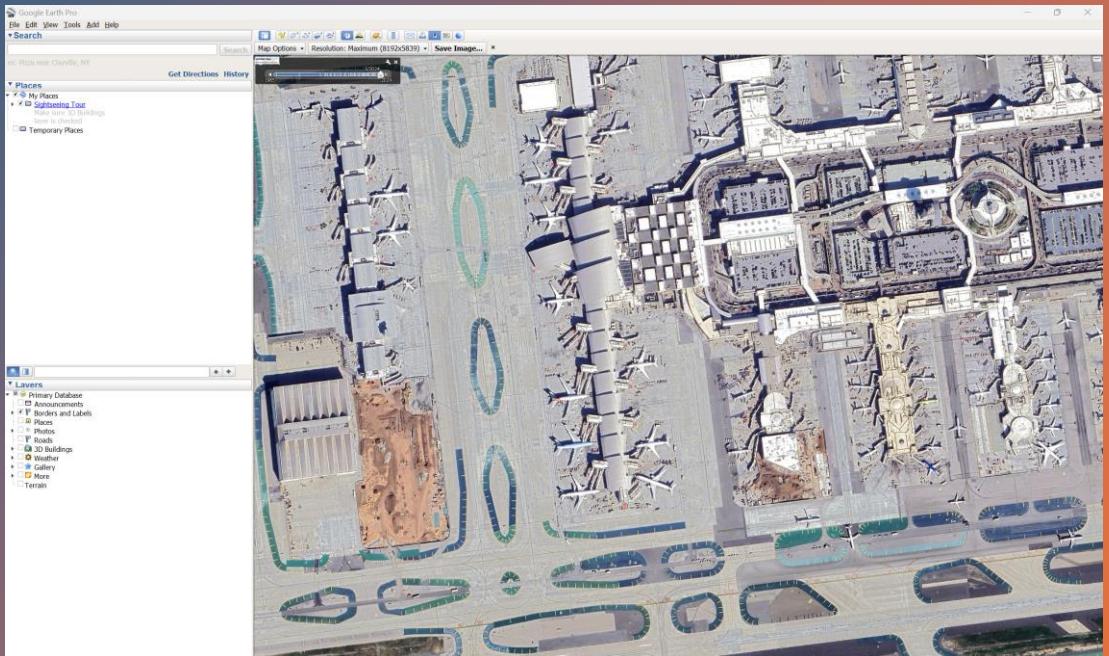


Goal & Why:

- Detect and count the amount of planes in a given satellite image of an airport.
- I just like planes.

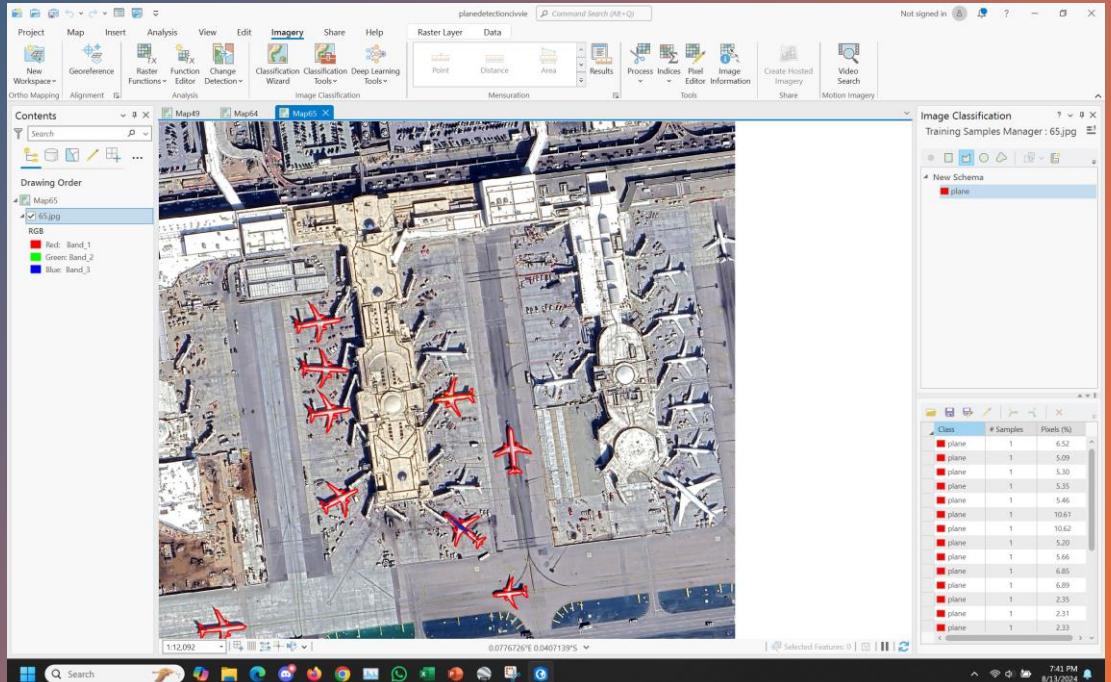
Data Acquisition

- Go to Google Earth.
- Go to location of choice.
- Choose time of image (if necessary because image quality and situation varies by time).
- Save image.
- Repeat steps 2-4 until desired amount of images acquired.



Annotation

- The machine needs references for what it's looking for, so we annotate the data.
- Annotation done on ArcGIS pro.
- Extremely tedious, yet somewhat therapeutic.



Preparing the Data

```
import os
import arcpy
# from arcpy import env
arcpy.CheckOutExtension("ImageAnalyst")
from arcpy.ia import *

arcpy.env.workspace = r"C:\Users\nicks\OneDrive\Documents\ArcGIS\Projects\MyProject19\MyProject19.gdb"
arcpy.env.overwriteOutput

aprxF = arcpy.mp.ArcGISProject(r"C:\Users\nicks\OneDrive\Documents\ArcGIS\Projects\MyProject19\MyProject19.aprx")

for num in range(1,37):
    print(num)
    arcpy.ia.ExportTrainingDataForDeepLearning(
        in_raster=r"C:\Users\nicks\OneDrive\Desktop\VSProjects\Betterplanedetection\images2" + '\\' + str(num) + ".jpg",
        out_folder=r"C:\Users\nicks\OneDrive\Desktop\VSProjects\Betterplanedetection\chips\chip" + str(num),
        in_class_data=r"C:\Users\nicks\OneDrive\Documents\ArcGIS\Projects\planedetectioncivvie\label" + str(num) + ".shp",
        image_chip_format="JPEG",
        tile_size_x=1024,
        tile_size_y=1024,
        stride_x=512,
        stride_y=512,
        output_nofeature_tiles="ALL_TILES",
        metadata_format="RCNN_Masks",
        start_index=0,
        class_value_field=None,
        buffer_radius=0,
        in_mask_polygons=None,
        rotation_angle=0,
        reference_system="MAP_SPACE",
        processing_mode="PROCESS_AS_MOSAICKED_IMAGE",
        blacken_around_feature="NO_BLACKEN",
        crop_mode="FIXED_SIZE",
        in_raster2=None,
        in_instance_data=None,
        instance_class_value_field=None,
        min_polygon_overlap_ratio=0
    )
```

- Images and Annotations need to be formatted for deep learning
- Automate process because there's a lot of images and processing takes a fair bit of time for each



Iteration 1:

- Run deep learning model with resnet 100.
- Results... alright, but not great

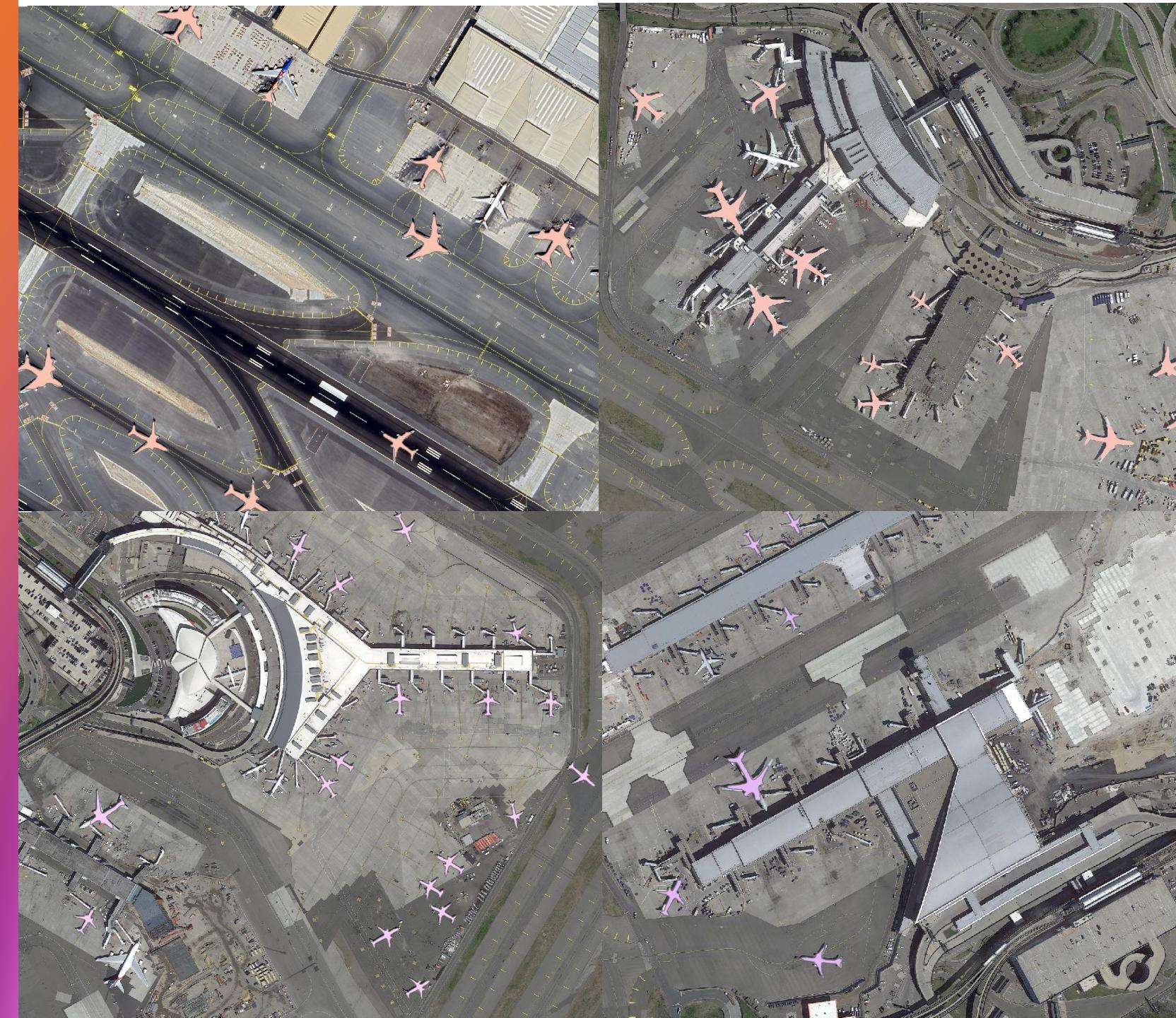
Iteration 2:

- Dumped some more images into the training samples and trained it from the 1st model.
- Results slightly better, but not very significant.



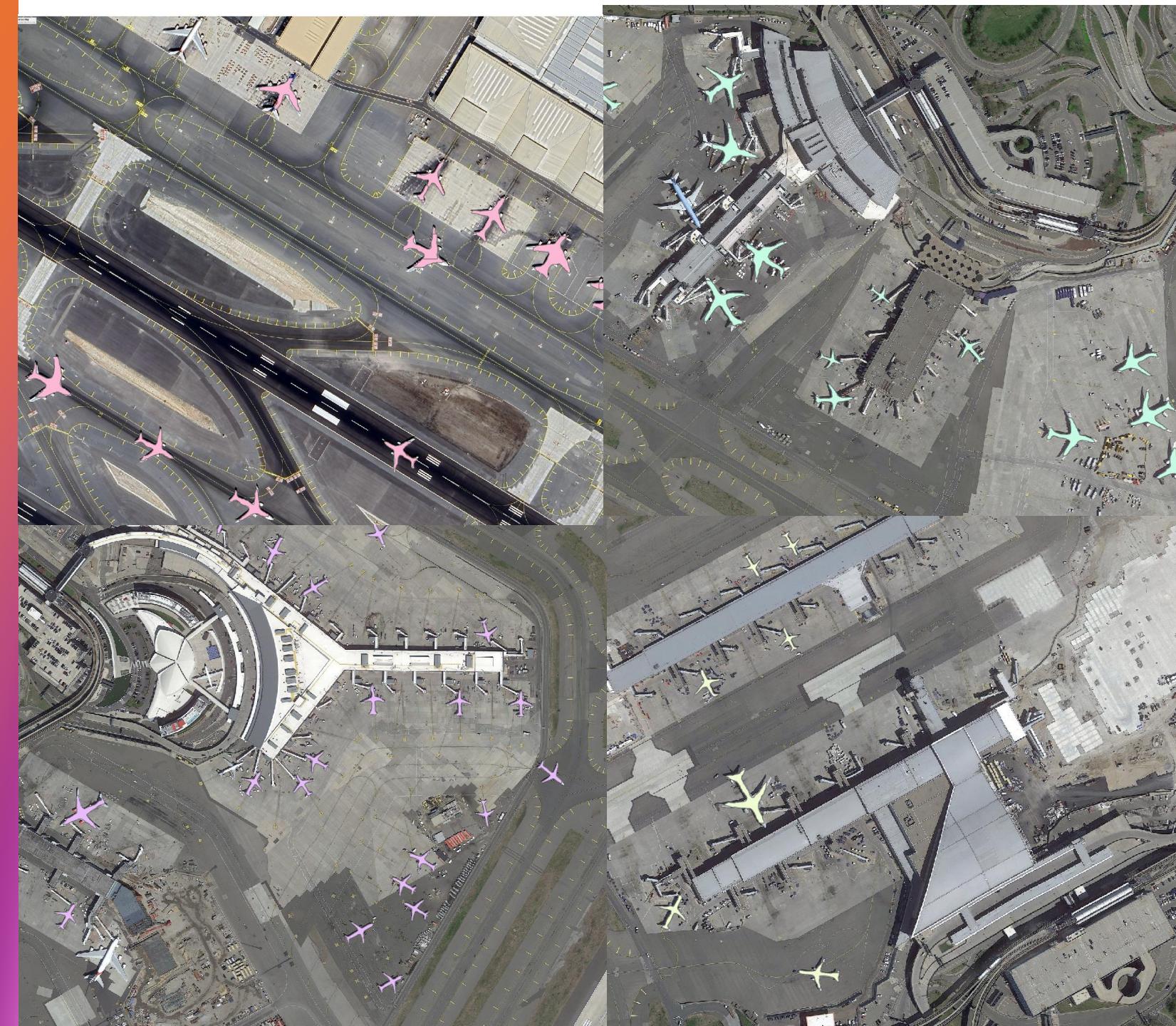
Iteration 3:

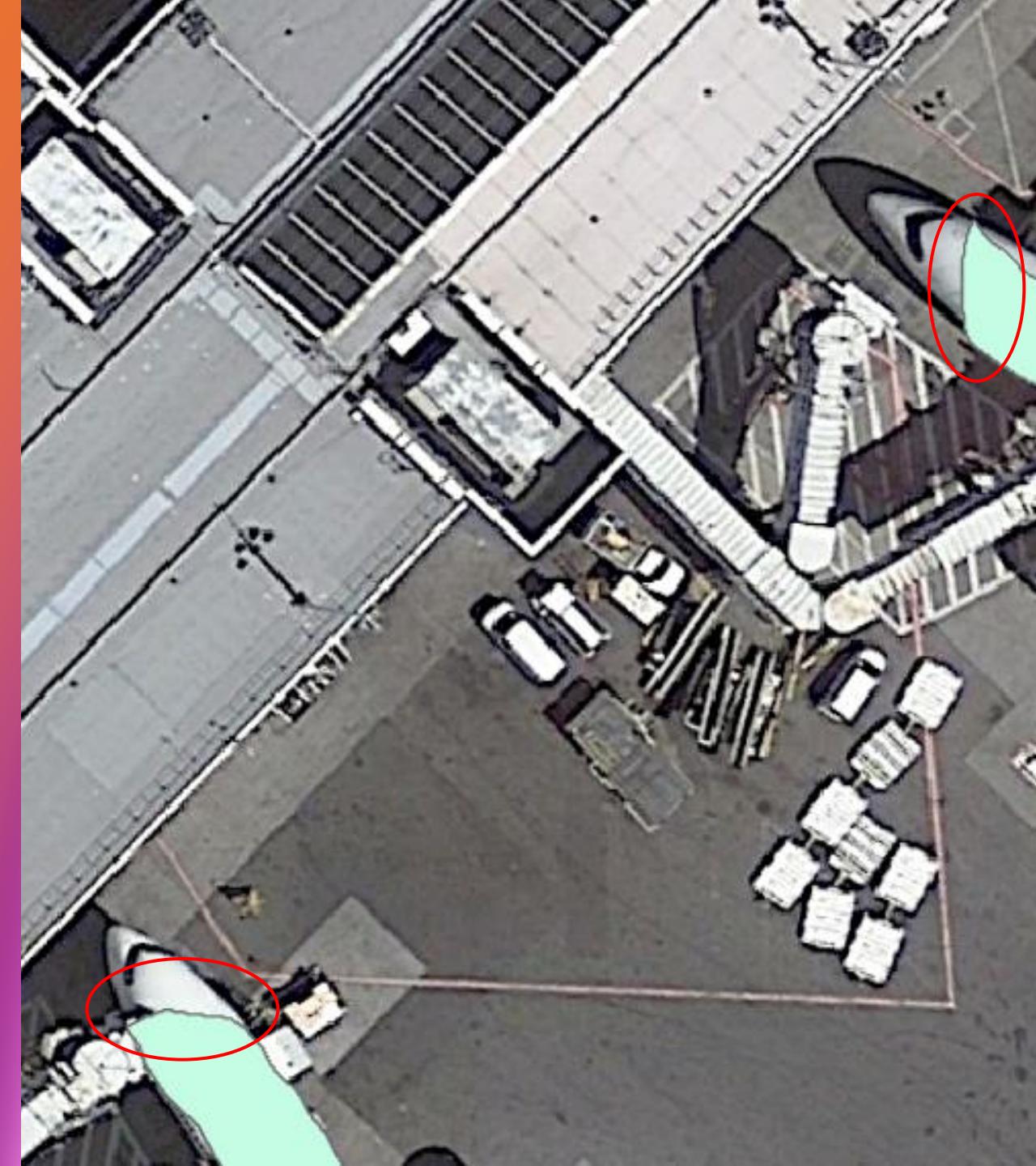
- Train resnet 150 model with the additional samples
- Still not much significant improvement



Iteration 4:

- Set new labelling standard of not including engine and end tip of plane in annotation, in other words simplify them. Hopefully it'll give less things the model needs to learn and improve it
- Also end up fixing some images with blatant annotation misses
- End up using less images than previous iteration because process is tedious.
- Still not much improvement, but according to training results the results are the best so far. Seems to be most promising method. Just need some more images.





Further thoughts:

- Results are quite decent, however there is a lot more room for improvement
 - I notice there's a lot of tapering near perfect or horizontal lines, which I suspect the tiling is screwing up the accuracy, so I'll have to work on that.
 - Some planes are blatantly not detected, but I imagine it's because I set the detection confidence to 90%, so a little lower might have them detected.