Report

Cleaning the dataset:

- Removing images that are all black
- Removing images that I considerer to big to process with my computer (with or height greater than 3000 pixels)

Training

- Given the time limitations and the no GPU constraints I used a pre-trained ResNet50 for object detection and only fine tuned the last 2 layers. I also limited the epochs to 7 so it can train the model in under two days (it should train in approximately 30 hours in CPU).
- I tested mask segmentation of the region of interest but didn't work as well as just drawing a circle inside the bounding box. Another advantage of the circle is that the radius can be chosen to improve the scoring. Given that I use the shapely buffer function to create compact polygons I decreased the radius of the circle to 90% of the original boxes predicted by the neural network.
- I created a function to dynamically choose the confidence for selecting the bounding boxes because different objects have different confidence, for example chimneys are predicted with very low confidence while other objects are predicted with very high confidence and usually those different objects aren't in the same image as a chimney so the confidence threshold will depend on the image.

Detailed Code
For more detail I've added two jupyter notebooks in the code section describing each section of the code.

Docker Readme
There is a README.txt file for instructions on how I tested the docker container

Processing requirements
- The docker image is python=3.8
- I use pythorch for CPU given the no GPU constraint
- It should run perfectly in Linux AWS instance (m4.2xlarge)
- Other python libraries can be seen in the Dockerfile

Language of code: Python

Map projection necessary to run code
Transforms are obtained from the tiff images using rasterio library

Source(s) of unclassified remote sensing imagery
I only used the provided images