Assignment 5

In this assignment you will implement two different object detection systems.

The goals of this assignment are:

- Learn about the object detection pipeline
- Understand how to build an anchor-based single-stage object detectors
- Understand how to build a two-stage object detector that combines a region proposal network with a recognition network

This assignment is due on ~~Monday, November 16th~~ **Wednesday, November 18th** at 11:59pm EDT.

Q1: Single-Stage Detector (54 points)

The notebook **single_stage_detector_yolo.ipynb** will walk you through the implementation of a fully-convolutional single-stage object detector similar to YOLO (Redmon et al, CVPR 2016). You will train and evaluate your detector on the PASCAL VOC 2007 object detection dataset.

Q2: Two-Stage Detector (46 points)

The notebook **two_stage_detector_faster_rcnn.ipynb** will walk you through the implementation of a two-stage object detector similar to Faster R-CNN (Ren et al, NeurIPS 2015). This will combine a fully-convolutional Region Proposal Network (RPN) and a second-stage recognition network.

Steps

1. Download the zipped assignment file

- Click here to download the starter code

2. Unzip all and open the Colab file from the Drive

Once you unzip the downloaded content, please upload the folder to your Google Drive. Then, open each `*.ipynb` notebook file with Google Colab by right-clicking the *.ipynb file. We recommend editing your `*.py` file on Google Colab, set the ipython notebook and the code side by side. For more information on using Colab, please see our Colab tutorial.

3. Work on the assignment

Work through the notebook, executing cells and writing code in `*.py,` as indicated. You can save your work, both `*.ipynb` and `*.py,` in Google Drive (click “File” -> “Save”) and resume later if you don’t want to complete it all at once.

While working on the assignment, keep the following in mind:

- The notebook and the python file have clearly marked blocks where you are expected to write code. **Do not write or modify any code outside of these blocks**.
- **Do not add or delete cells from the notebook**. You may add new cells to perform scratch computations, but you should delete them before submitting your work.
- **Run all cells, and do not clear out the outputs, before submitting.** You will only get credit for code that has been run.

4. Evaluate your implementation on Autograder

Once you want to evaluate your implementation, please submit the `*.py`, `*.ipynb` and other required files to Autograder for grading your implementations in the middle or after implementing everything. You can partially grade some of the files in the middle, but please make sure that this also reduces the daily submission quota. Please check our Autograder tutorial for details.

5. Download .zip file

Once you have completed a notebook, download the completed `uniqueid_umid_A5.zip` file, which is generated from your last cell of the `two_stage_detector_faster_rcnn.ipynb` file. Before executing the last cell in `two_stage_detector_faster_rcnn.ipynb`, please manually **run all the cells of notebook and save your results** so that the zip file includes all updates.

Make sure your downloaded zip file includes your most up-to-date edits; the zip file should include:

- *single_stage_detector.py*

- *two_stage_detector.py*
- *single_stage_detector_yolo.ipynb*
- *two_stage_detector_faster_rcnn.ipynb*
- *frcnn_detector.pt*
- *yolo_detector.pt*

6. Submit your python and ipython notebook files to Autograder

When you are done, please upload your work to Autograder (UMich enrolled students only). Your `*.ipynb` files *SHOULD include* all the outputs. Please check your outputs up to date before submitting yours to Autograder.
Note: Autograder for A5 will start working on November 5th.