CMPSC 497 - Deep Learning for Computer Vision: Homework 6

Professor: Huijuan Xu

Due: April 12, 2023 @ 11:59 EST

Two-Stage Detector

The notebook **two_stage_detector.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

Once you unzip the downloaded content, please upload the folder to your Google Drive. For more information on using Colab, please see our Colab tutorial.

2. Work on the assignment

Work through the notebook, executing cells and writing code in `two_stage_detector.ipynb.`

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

- The notebook has 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.
- **IMPORTANT: Run all cells, and do not clear out the outputs, before submitting.** You will only get credit for code that has been run.
- If you face an issue while downloading the dataset with the given code, then **download the dataset using the links directly as you did in previous assignments**. You can change the paths and code accordingly wherever required in the notebook.

Tips for the assignment: There are many cells and functions which require code changes. Start with changing one function at a time. For debugging purposes, checking output of each function directly and understanding if it is what was expected, will save a lot of time.

3. Expected output

Submit only the ipynb file. For grading, we will look at the output of the last 2 cells of the assignment, the loss function graph and mAP score for each class. We do not expect the best performing model and the aim of the assignment is for you to understand each part of training an object detector.

We will look at the loss function, and the expected output is a decreasing trend in the loss function. This will simply tell whether the training is happening properly or not. A specific loss value is not needed for this.

We will also look at the mAP score for each class, and the expectation is to have a mAP of at least 20 for each class.

4. Rubric

Section	**Percentage**
Backbone with Feature Pyramid Networks (FPN)	15%
Faster RCNN - stage one (Region Proposal Network, RPN)	30%
Faster RCNN - stage two (RoI Align)	20%
Helper code for Non Maximum Suppression (NMS)	10%
Training	25%