CMSC 170 MACHINE PROBLEM 2
Object Detection

Total: 100 points (20% of grade)
Deadline: May 29, 2018 (Tuesday)

INTRODUCTION

The second machine problem is about **object detection.** You are allowed to form your **own group with 4 members**. Each group will choose their own object detection topic, and have it approved by the instructor.

The group will create **Python** codes, using the **OpenCV** library, that will detect an object of interest within an image / video (series of images). The program (**detector.py**) will receive an image as an input (presumably containing the object of interest) and the program should output the bounding box coordinates of the detected object. You can visualize the results by drawing the bounding box over the input image.

DATASET

Create a dataset containing **20 images**. It should be made up of images that *contain* the object of interest, and at least 3 images that *doesn't contain* the object of interest. The images in the dataset must have **variety**: the object of interest could be in varying *angles*, *scale*, *locations*, *lighting conditions*, etc. For simplicity, include only images that have one instance of the object.

You will need to **manually label** the images. The correct answer in a detection problem is a bounding box, represented by two points: the top-left corner, and the bottom-right corner. Determine what these coordinates are for each image, and save these coordinates in a file (e.g. **labels.csv**).
Example format: filename, x1, y1, x2, y2 (where x1,y1 = top-left, x2,y2 = bottom-right)

EVALUATION

To evaluate the performance of your detector, use the **Intersection over Union** (IoU) metric. It computes the intersection area of the correct bounding box and the predicted bounding box. If this intersection area is big, then we predicted correctly most parts of the correct bounding box. But, we also need to consider the area of the union of the correct and predicted bounding boxes, and make sure that the union's size is not too far from the size of the intersection. Read this article for a more detailed explanation of the IoU metric: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

Create **evaluate.py** that will loop through all 20 images in your dataset and compute the Intersection over Union metric for each image based on the image's correct bounding box (from *labels.csv)* and the predicted bounding box (from *detector.py*). Display the filename and IoU metric of each image, and compute the average IoU for the whole dataset.

REPORT PAPER

Create a report paper for your project where you will 1) *introduce* your problem, 2) list some *related works*, 3) describe your *dataset,* 4) discuss your *detection algorithm* in detail, 5) describe the *evaluation metric* you used (intersection over union), and 6) present and discuss your *experiment results,* including IoU metric scores. The suggested format for the report paper is attached at the last page.

DEMO VIDEO

In lieu of a project defense, you will create a demo video to showcase that your object detector works. Create **demo.py** that will loop through all 20 images in your dataset and use your object detector on each image. It should display both the *correct bounding box* (green) and *predicted bounding box* (red), together with the intersection over union (IoU) score at the top-left corner. The video could just be a simple screen capture while you use *demo.py* on the test images.

The demo video must show **true positive** detections (correct box & predicted box has big overlap) and **true negative** detections (image with no object has no predicted box). It may also contain **false positives** (predicted box is in the wrong place), and **false negatives** (image has object but detector doesn't predict box).

SCORING

Experiment Results	Based on how well your detector works	**40**
Report Paper	Based on how well-written and comprehensive your report paper is	**30**
Codes + Dataset	Based on correctness of Python codes and completeness of dataset	**15**
Demo Video	Based on how well your demo video showcases your project	**15**
Total		**100**

SUBMISSION

1. Create a zip file containing the ff:
    - Python **codes** (*detector.py, evaluate.py, demo.py*)
    - **Dataset** (20 images) with *labels.csv*
    - PDF of your **report paper**
    - Demo **video**
2. Email your output to jrdaradal@up.edu.ph on or before **May 29, 2018 (Tuesday), 11:59PM,** with the subject format: **[CMSC 197 - MP2 - Lastname1/ Lastname2 / Lastname3 / Lastname4]**
3. Submissions on or before the deadline will receive +5 points for punctuality.
4. **Hard deadline**: Submissions after **May 31, 2018 (Thursday), 3PM** will no longer be accepted. Failure to submit will result in a grade of INC.
5. Rate yourself and your groupmates (1-10) and submit your ratings to your instructor.

REPORT PAPER FORMAT

Project Title

Member 1, Member 2
Member 3, Member 4

-

INTRODUCTION

(*Introduce the problem here - what are you detecting? why is it worth solving? who will benefit from this? Include also at least 3 related works and cite them in your references.)*

DATASET

(*Describe the dataset you created. How do the images look like? Provide a screenshot of the images to show variety in the dataset: different angles, scale, lighting conditions, etc. You can also display the correct bounding boxes that you manually labeled. Describe also your process of manually labeling the dataset.)*

ALGORITHM

(*How does your object detector work? Enumerate the steps and provide the details of your algorithm. Did you use existing object detection algorithms? Include any image processing techniques you performed on the image before the detection)*

EXPERIMENT RESULTS

(*Briefly explain the Intersection over Union metric. Present the results of your object detector. Attach screenshots of examples of successful and failed detections. Display the IoU metric values per image, together with the average IoU for the whole dataset, in a table. Discuss your results. Is your detector good enough? Are the predicted bounding boxes close to the correct bounding box? Were there any false positives and false negatives? What were some problems you encountered?)*

REFERENCES

(*Insert references here)*