CS 585, Spring 2024
PA3: Tracking Vehicles on the Commonwealth Avenue

Out date: 2/14/2024

Due date: 2/27/2024

Code and data:
https://drive.google.com/drive/folders/16Eml-MCgg-_giw53DARjS9ury5HuOpU-?usp=sharing

Summary of the Assignment

This assignment has two parts:

1. Single Object Tracking with a Bayesian Recursive Filter: Given detected 2D locations of a vehicle over multiple frames in the video, you need to implement an alpha-beta filter or a Kalman filter to generate a smooth 2D track of the vehicle based on these 2D observations.

2. Multi-Object Tracking and Data Association: Given the bounding boxes of multiple objects detected in the video, you need to track them by assigning a unique ID to each object over the video as long as they are detected.

THE MOST IMPORTANT RULES:

1. No deep learning for detecting and tracking!
2. No video forgery!

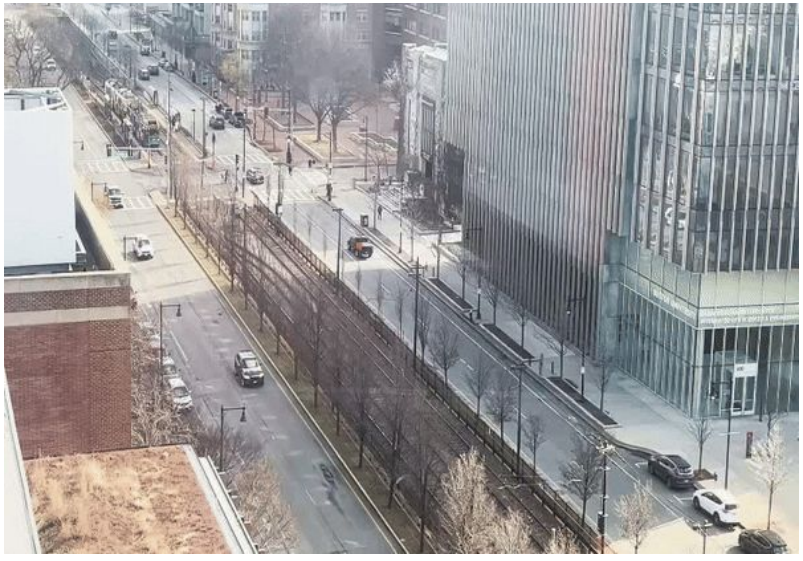
Once we find evidences indicating the submission violates the above rules, **your final score of PA2 is 0**, no matter how much you implemented that do not violate them.

Part 1:

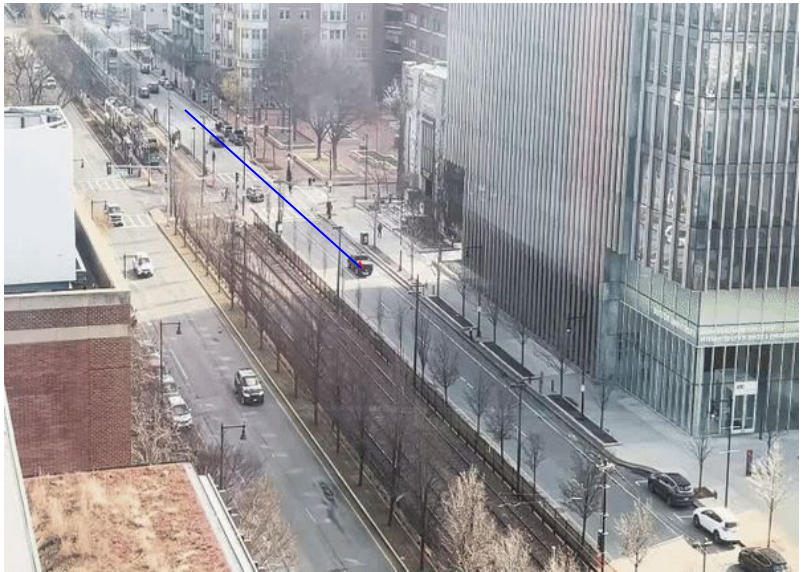
You are given a list of centers for a vehicle over frames in a video, formatted as [[x_1,y_1],...,[x_n,y_n]. The object location might be missing in certain frames, annotated as [-1,-1].

Your task is to build a tracker that estimates the 2D trajectory of the vehicle and smoothes the 2D track with an alpha-beta filter or Kalman filter.


```
{"obj": [[-1, -1], [312,
228],...,[166, 100], [166, 101]
}
```

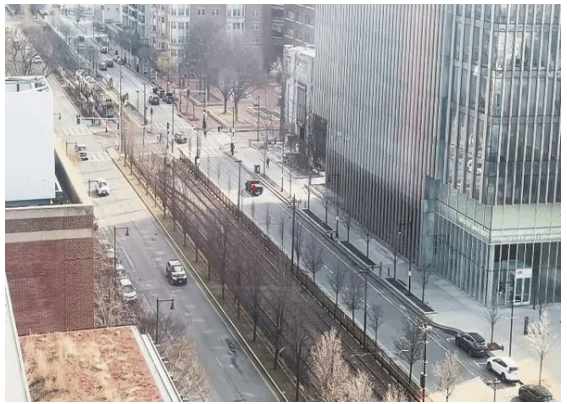



Red: track based on observed centers
Blue: ideal track smoothed (e.g. by filter)



Grading Criteria

1. Code (10 points + 10 extra credits)
   a. We are going to check whether your code includes deep learning code and if it can generate proper tracks in a video. (Easy 10 points if you are honest in what you've done.)
   b. Optional: Implement a Kalman filter (instead of the Alpha-Beta filter) to estimate the vehicle track. (10 extra pts)
2. Tracks (20 points)
   a. Write your smoothed tracks in the format that is the same as the given one. However, your track should not include any [-1,-1]. We will compare the track with the one based on the observed centers. If they are similar enough, you will get full credit. If not, points will be deducted proportionally according to how dissimilar they are.
   b. Check **part_1_object_tracking.json** as an submission example.
3. Visualization (20 points)
   a. A .gif or .mp4 that draws the track of the object.
   b. Your track should be following the moving vehicle (10 pts), and it should be sufficiently smooth (10 pts).



Part 2:

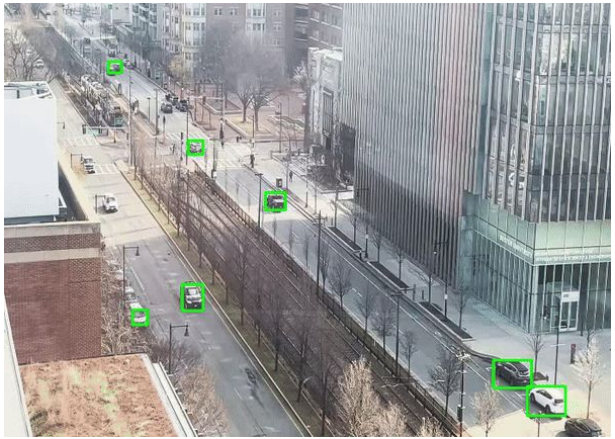
You are given a list of object bounding boxes in each frame of a video.

Your task is to build an object assignment algorithm that assigns a unique ID to each object.

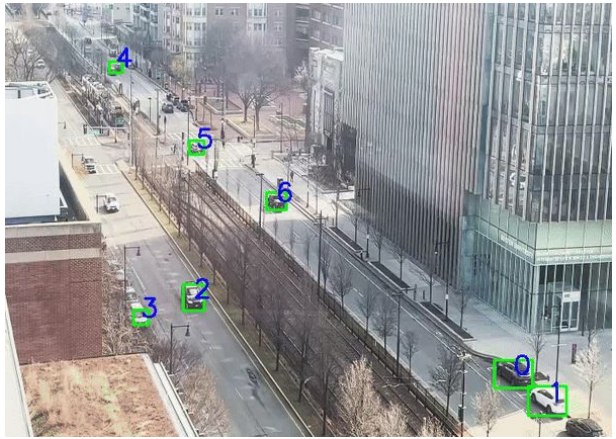
Ideally, one object should only have one unique ID (**a bad example is given on the right**)


```
{"0": [{"x_min": 564, "y_min": 410, "width": 45,
"height": 32}, …, {"x_min": 302, "y_min": 219,
"width": 23, "height": 19}, {"x_min": 212, "y_min":
160, "width": 17, "height": 16},...]
```


a list of object bounding boxes in each frame of a video.

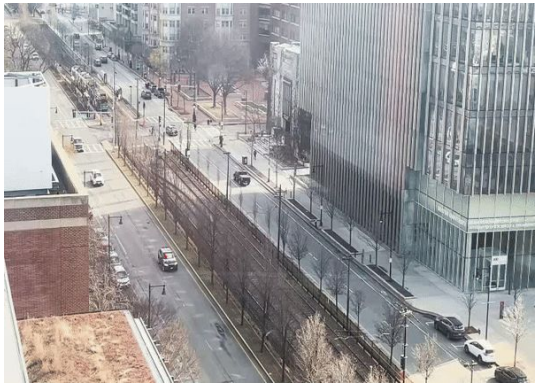


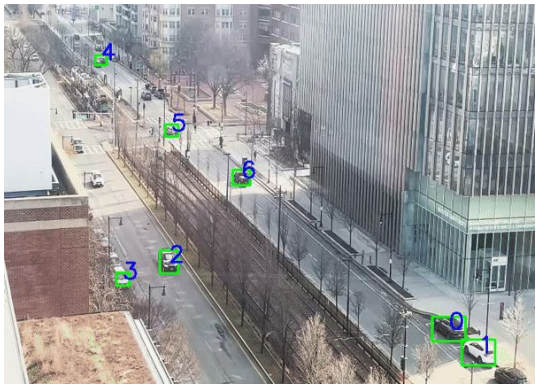
Bad example: (A naive assignment based on closest object distance between last and current frame)



Grading Criteria

1. Code (10 points + 10 extra credits)
   a. We are going to check whether your code includes deep learning code and if it can generate proper IDs in a video. (Easy 10 points if you are honest in what you've done.)
   b. Optional: Implement the Hungarian algorithm for matching. (10 extra pts)
2. Object assignment (20 points)
   a. Modify the given bounding boxes data file, add an additional "id" key-value pair to each item to indicate the unique ID for an detected object in the video. Other than that, do not change anything in the file. Submit the modified data file.
   b. We have tracker for the detected objects in the video. We will run our code to look for the ID of these objects to see if your IDs of these objects are consistent over different frames. (20 pts)
   c. Check **part_2_frame_dict.json** as an submission example.
3. Visualization (20 points)
   a. A .gif or .mp4 that draws the IDs of the detected objects. (10 pts)
   b. Each detected object should have a unique ID, and the IDs should be consistent over different frames (10 pts)





Things to help you with finishing the first stage:

1. Drawing with OpenCV:
    a. cv.puttext
    b. cv.rectangle
    c. cv.circle
    d. cv.polyline
2. Working with video frames (check lab 3)
3. Alpha-beta filter:
    a. https://en.wikipedia.org/wiki/Alpha_beta_filter
4. Hungarian algorithm:
    a. https://en.wikipedia.org/wiki/Hungarian_algorithm
5. pandas and json to read and output data:
    a. Use ChatGPT for code examples

We will provide demo code to read and visualize the data, and the rest is left to you to finish.