Computer Vision - Project 2
Visual traffic monitoring at a road intersection

Ionescu Andrei (407)
andreei.ionescu@gmail.com

1 Introduction

The objective of this project is to create an automatic procedure that:

1. Task 1 - Classifies whether a lane out of 9 is occupied based on given queries. (solved)
2. Task 2 - Track a car using an axis-aligned bounding box. (solved)
3. Task 3 - Count the trajectories of vehicles in a given video (unsolved).

2 Task 1

In the first task we are given a static image that has plenty of cars scattered around, some on the lanes while others in the intersections or on lanes that we are not interested in. Because the scene is static and we are given context videos that take place ahead of the respective images, we can build a background image where no car is present. While this is generally subjected to imperfections (especially in environments that change), in this case the test images are static so this technique is suitable. In this manner, I sampled 3-5 images from each context video and using GIMP and image masks I removed cars that were present in the image so that I can obtain a background model.

The next step is to transform both the input image and the background image into grayscale to flatten the color channels. After this I subtract the two and take the absolute value. In order to smoothen out some noise I apply a Gaussian blur operation followed by an adaptive thresholding operation by using Otsu's Algorithm. The noise that still is present in the image is further diminished by applying multiple erosion operations with rectangular kernels followed by many dilation morphological operations to strengthen the presence of the cars which have high presence. Now we can further improve this *change map* by applying a contour algorithm and find some rectangles which can cover these smaller changes. By taking the union of all found rectangles we can build a boolean mask that indicates areas of potential changes.

The core of finding the cars is through a **FasterRCNN** network which predicts the bounding boxes for the cars in the RGB image. Some false positive bboxes are removed if their centroids are not present in the boolean mask obtained earlier. The final part of the method consists in localizing in what zone/lane the bounding boxes are present.

To solve this I used GIMP to build 9 boolean masks indicating each lane. Moreover, I also took the union of these to define three zones A (1-3), B (3-6), C (6-9). Firstly, I see if the centroid is near one of the zones. If affirmative, then I assign the car in that zone, otherwise the bounding box is eliminated. Secondly, I use the corners of the bounding boxes to assign the cars into each lane. For zone A, the 3/4 point of the lower edge of the bbox is used. For zone B, the left corner is leveraged. For zone C, the same 3/4 point of the lower edge is used. This trick has high accuracy and works in many edge cases even when cars overlap or are very crowded.

3 Task 2

The second task has videos as input and so this is a tougher challenge that requires a more adaptive approach. Therefore, I use two pretrained neural networks on the COCOv1 dataset: YOLOv8 and FasterRCNN. At each frame I try to apply one model or the other in order to obtain detections. Then, I use the OCSORT algorithm to track the generated bounding boxes. Sometimes, the tracker loses the target and has many id switches, or fails terribly when cars are behind traffic poles. To overcome this problem I track all seen bounding boxes and their respective ids. If the tracker loses its focus but then regains it, a smaller id has bigger priority and that bbox is used. If no known id is found, I try to find the closest one according to two metrics: the IOU and the distance between centroids along with applying Lowe's ratio test to account for crowded cases. If nothing works, meaning that I have no high confidence on choosing a new bbox for which an id switch might have occurred, than I just pick the last seen bbox. However, this only stands for a couple of frames, because this case is also the case when a car goes out of the camera view. The method still has some of the mentioned problems but it seems more robust.

4 Task 3

It is left unsolved but some ideas might be to consider the traffic rules to predict where a car will go, or to build a grid over the intersection and track each's car orientation.

5 Conclusion

The algorithms and approaches tackled in the course of this project try to provide a robust solution but they still have some error cases.