Signal processing: exercise 1

Task 1

Background subtraction

Background subtraction consists of building a model of the background of the video, which is defined as everything that doesn't move over the duration of the video. This model is a frame that contains everything you see in the video except the objects that move. To build it we select 50 random frames from the video and average them. The result is a frame with only the background, without the objects.
Once we have the background model(frame) the last step is to subtract it from every frame in the original video(note that we transform all frames including the background model to grayscale before the subtraction), this will result in grayscale frames that only contain the moving objects and the background is removed.

Frame differencing

Consists of subtracting pixel values from a group of consecutive grayscale frames, the result is that static objects are removed after the subtraction and we only endup with pixels with values above 0 for regions where there was movement. Then we can binarize the result using a threshold to end up with a frame of 0 or 1 pixels where 1 represents movement and 0 the contrary.

Approach followed on my application

In order to detect cars in the street my application uses background subtraction technique to detect big objects that move at least once during the duration of the video, ie: cars on the street. I combine a sequence of frames together for analysis mainly to reduce computation but I intentionally did not use frame differencing for motion detection as this technique misses cars while they are stopped at the traffic lights or while they stop for a small number of frames(ie: pixels don't change over the frames compared). Using background subtraction alone works better for the video provided.

Task 2

	Total number of cars	Cars per minute
Traffic_Laramie_1.mp4	7	2.36
Traffic_Laramie_2.mp4	4	2.27

Note: On Traffic_Laramie_1.mp4 the total number of cars is actually 6 but my system also detects the bike with 2 people going in the direction of city center as a car.

Application description

My jupyter notebook loads the video with filename provided in the the third cell and detects cars on main street going in the direction of city center, it tracks each car and at the end outputs the total car count and the number of cars per minute which is the total count divided by the duration of the video in minutes.
In order to detect and track cars I do the following:

1) Build model of background
2) For each frame remove background using model from previous step
3) For each frame apply a mask to remove everything that is not on main street in the side that goes towards city center
4) Find contours for all objects above certain size(discards pedestrians) which will be cars(or bike) going to city center
5) For each car store its coordinates in a set and increase counter every time an object does not have coordinates close to it(small euclidean distance) in the set. In every iteration will update this set with newest coordinates of objects and remove coordinates from objects that don't show up in frames anymore.