Exercise 1 Report

Task 2 Table

	Total number of cars	Cars per minute
Traffic_Laramie_1.mp4	6	2
Traffic_Laramie_2.mp4	4	2

Description of the frame differencing and background subtraction

The frame differencing technique is for finding interesting parts of a video image. You store the previous frame, subtract it from the new frame to calculate the absolute difference between them.

The difference is converted into a binary image. If a particular pixel value is greater than a certain threshold, it will be assigned the value for White (255) else Black(0). This tells that something in the video has moved or changed its position.

The background subtraction technique uses the same concept, but instead of the previous frame, it subtracts the background frame from the new frame and gets its absolute difference. It is used for detecting moving objects (cars, humans, etc) of a video image. The background frame is found by getting the median frame within randomly selected 50 frames, which is executed in the initial process. We also specify the number of consecutive frames per process in the initial setting. Perform frame differencing on each frame, sum these consecutive frame differences, and then find contours from the resulted frame.

Analysis of the application

Analyze the application in the following key points of Task 1 and Task 2.

Task 1

Contour area - To avoid detecting pedestrians, I identified 3000 as a threshold to discriminate between car and human.

Task 2

This task required tuning the following two settings to count cars correctly.

The number of consecutive frames - Analyze the first video and the second video have different car speeds, I needed to adjust the number of consecutive_frame per process for each video to avoid misdetection. For example, car speed is slower in the first video so if set consecutive_frame=4, it would result in duplicate detection. On the other hand, car speed is faster in the second video so if set consecutive_frame=5, it would result in no detection. Based on the experiment, I identified 5 for video 1 and 4 for video 2.

Detection area - Observe that cars go to the city centre pass by a certain area, the application only needs to watch that point to detect the targeted cars. I drew a rectangle in the detection area and counted only if the contour area of the car passes by the area. With this approach, the area size affected the detection performance. For example, it detects nothing if the area is too small and car speed is fast while it detects multiple times with too large area.

In conclusion, to use the application in practice, especially task 2 of counting cars, we should analyze the road condition in advances such as the average car speed and the number of cars so we can tune the two settings I stated above for the specific road. However, considering that these techniques don't use deep learning techniques and don't require any pre-training, the detection performance is quite impressive. This would be useful in many real-world fields that require live detection such as in grocery stores, highway or live music concerts, etc.