

Abstract

This document contains the write up for HW06 which is a software based approach to count cars and vehicles on express ways using OpenCV and C++.

Introduction

Presently, the traffic department in various cities doesn't have any record of traffic passing through various freeways or they use hardware based approach to do that which turns out to be quite costly and requires maintenance. Having a good record of traffic can be used to improve and implement various developmental projects in a proper way. Thus, we will be building a solution to count the vehicles passing under a bridge in a video. This will be done in C++ using OpenCV library.

This idea can further be extended to detect and record the speed of vehicle eliminating the use of expensive hand-held speed detecting devices used by traffic police. A software solution to this will be a cheaper alternative. We can even extend it further and use it to recognize a License plate. This could be informative to track a suspect and find the freeway/route on which that vehicle was seen last. This could be done only if we use high resolution CCTV cameras and not normal CCTV cameras. But for now, we will be detecting count of vehicles in a video.

Approach

We used recorded videos as an input to the program and read the video frame by frame, applying various operations on each frame. For counting cars, we will draw a line on the frame and when the car has crossed that line, we will increment the counter by 1.

We first draw a horizontal line at around middle of the frame. We didn't draw it too high or too low at the bottom as if we draw it too near , say at 10th or 20th row the contours merge into each other as the vehicle moves away from us. If we draw it too far say, on 70th or 80th row we get too many contours in a car itself and hence it is counted more than once.

We then change the frame into gray image, smooth it using Gaussian blur,find the difference between consecutive frames,threshold it and then apply erosion and dilation on it so that only the cars/vehicles is detected and nothing else. We then find the contours in the frame and draw convex hull around them. We find that the contours that have area greater than 400 are vehicles on the road, and hence we consider only that contours that have area greater than 400.

We then store all the possible blobs in a frame to a vector and match the blobs of next frame to current existing frames.This is done by calculating the next position(centroid) of the blob. We increase the count of car when contours crosses the horizontal line. We do this by comparing the y-coordinates of the horizontal line and the contour. If the y-coordinate of the contour in previous frame was below the line and in current frame its less than the y-coordinate of horizontal line, it has crossed the line and hence, we increment the counter. This is repeated for every frame.

Result

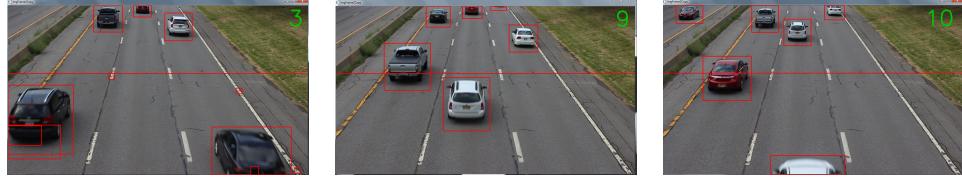


Figure 1: Snapshots from the output.

Analysis

The code initially was not detecting large vehicles and was detecting some vehicles many times, hence giving the wrong count. To fix this, we did the dilation and erosion from a large structuring element, placed the horizontal line at optimum position so that it does not detect false cases. This case was fixed.

Also, the code works for videos that are stable and does not work efficiently for videos that are not stable. We need to use video stabilization for that before applying this code.

For the input video, this code gave near 100 percent accuracy and can be optimized further to deal with all kind of videos and scenarios.

References

- [1] Prof. Thomas Kinsman, for recording the videos, specifically to see us succeed
- [2] Mr. James and Mr. Sam (System Admin CS department) and team for setting up the required resources
- [3] Code reference - <https://github.com/MicrocontrollersAndMore/>