Car Detection and Counting from Video Project using OpenCV Libary

Muhammed Kenan KAYIKCİ   
*Department of Information Systems and Faculty of Engineering*  
*Piri Reis University, Number of student 20200205007*Istanbul, Turkey  
kenankaykci@hotmail.com

*Abstract*— Car counting is a process to understand traffic flow on roads to assess the traffic. Like high spec cameras and algorithms, technology improved to help due to the high density of traffic. The main purpose of roads is traffic, because of that technology invented some simply clever assistant ideas like speed cameras, red light recognition and emergency lane detector. This project purpose is helping the traffic flow by counting the number of vehicles crossed to understand traffic intensity. This algorithm can apply to all the cameras included live support server. Multi car recognition can be implemented by help of OpenCV library that included in python coding language. The performance of this system is up to two main subjects: hardware and physical. We can minimalize this by precision hardware settings.

Keywords-Internet of Things (IoT), Morphological Operations, Car Counting, Python, Traffic Flow

# Introduction

A camera has a big role of this intelligent algorithm to work. Camera needs to be high resolution and positioned close to the road so our algorithm cannot take the unwanted results of the system. For the system we did used python coding language because of the library and easy to use format of the language. After the video added to our python folder with help of the OpenCV library we can code the algorithm easily and efficiently at the same time. With that OpenCV added to our library of python we can use the morphological filters and equations to filter out of the video that is no big trouble that our algorithm can understand the moving vehicles during the video time. For the counting we just put a line length about the road and within specific pixels that vehicles moved so the algorithm isn’t always counting the counted vehicles. After all of those coding done, we put an text top of the video that we can see how many cars passed the red line also when the car actually passing the line algorithm changes the colour of the line so the user can understand code is working. The main of this project is to understand the moving vehicle and counting it for the user.

# The Background of the algorithm

## Python Coding Languege

Firstly, we used python to use enormous amount of library and sources that we can get the help we needed from internet. Python’s codes are easy to use and when the coding all wrote right it could be used like an .exe file that we use to open the applications. After we decide that we will use the python we need to have a virtual environment to run these files whatever device we like to run at. With the Microsoft Visual Studio Code, we can create local virtual environment to specific folder. When we created our virtual environment, we can work on the project file.

## Python Libaries That We Used for the Program

When we installed the python there is a library called ‘pip’ that we can install other libraries from the command prompt without any of internet browsers. Help of the pip we will install the OpenCV library by typing ‘pip -m install OpenCV’. We used ‘-m’ because we are installing it for the virtual environment not for our local device. OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Which we used this library for algorithm for the vehicle counting. When we downloaded the OpenCV NumPy automatically downloads without our desire, but we will use NumPy for the calculating and using the formulas, arrays, Numeric and Numarray libraries.

# The algorithm

For the start we use the OpenCV libraries morphological filters that included. We are using morphological filters to help the algorithm understand the objects movement easily. For the morphological equations we used is dilation for closing and we used structuring element size of 5x5. We could use opening and then closing but it’s always counting the trees and other objectives that can see from the camera.

After all the filter we put into the algorithm we can see that detection of the cars can be happen by the main structure of the vehicles. But for the counting we need to set a rectangle and a centre point. We made the rectangle by OpenCV command ‘enumerate’ to get the cars shape roughly. Another factor to consider for our object detection is to give a centre point. We are doing it by taking the length of the rectangle and divide by half. In conclusion our program can detect the cars easily.

Last part of our algorithm is if we don’t put any of lines or point it will count the vehicles by how many frames are the cars inside of the video. To prevent this, we need to put line for the reference point. Furthermore, we give an offset for the line prevention to miss understanding the cars for birds or any other objects we don’t want.

All in all, we have the requirements to understand and input the value to user. For the summation for the data to user we put a text to video that counter value of passed cars thought the line.

# The Video Used For This Program

For the video there were lots of examples to go through taken by me and other video files from internet. The main important thing is that the camera angle needs to be looking to road it doesn’t require to centre the road from above. The videos are closer to road gives smooth recognition.

For this picture [1] the camera angle is ~75° so we can clearly see algorithm can understand incorrectly but if we came to other picture [2] the camera angle is ~50° so the algorithm work not flawlessly but the error percentage is lower than first one.

The ‘video3’ videos wrong inputs [1]

The ‘video’ videos wrong inputs [2]

##### Acknowledgment *(Heading 5)*

First at all I want to thank for everyone that helped through ‘Stackoverflow’ that made my questions clear. There were lots of examples on the internet but there were old codes, so it needed some recoding to do. After all of the videos and examples that I gone through this code just came up. There were lots of morphology coding algorithm needed but the last style was the smoothest one to run at. There were lots of combinations of dilation and erosion but the most efficiently way to use closing. For the structuring element we tried to use (3,3) but for the high resolution the (5,5) structuring element were a bit better.

##### References

Firstly, I used my teacher’s lessons [1] for helping me understand the why I need to use morphological equations.

The Stackoverflow [2] and Github [3] comes first for most of the reference coming from. And I cannot mention Python, OpenCV [4] and NumPy [5] libraries to help me a lot.

Lastly, I did used lots of videos for covering my unknown issues for morphological equations.

1. Prof.Dr. Yıldıray YALMAN Piri Reis University (head of information systems engineering department), Digital Image Processing lesson and Assist. Prof. Dr.Mehmet Zeki KONYAR, Computer Vision lesson
2. OpenCV TypeError: Expected cv::UMat for argument 'src' - What is this? From ‘autonocat’ user, Solution by ‘gameon67’ ttps://stackoverflow.com/questions/54249728/opencv-typeerror-expected-cvumat-for-argument-src-what-is-this
3. Expected cv::UMat for argument 'src'. From ‘K R E A M nickname: iseegr8tfuldeadppl’ https://github.com/opencv/opencv/issues/14077
4. OpenCV (Open Source Computer Vision Library), https://opencv.org
5. NumPY, <https://numpy.org>
6. Tracking & Count using Yolov5 DeepSort from ‘AI4LIFE’ https://www.youtube.com/watch?v=7gz3vUmy0C0&t=863sY.
7. counting vehicles tracking & counting using yolov5 from ‘FREEDOM TECH’ <https://www.youtube.com/watch?v=BuLfLOhobtU>
8. Python Project Tutorial- Vehicle Detection And Counting using OpenCV from ‘CodeWithKrian’ (Indian languege no english subtitles) https://www.youtube.com/watch?v=6kZftXunlTY&t=2967s
9. OpenCV Python Tutorial For Beginners from ‘ProgrammingKnowledge’ https://www.youtube.com/watch?v=xSzsD4kXhRw&t=678s