International Journal For Research in Applied Science and Engineering Technology

iJRASET

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VII Month of publication: July 2021

DOI: https://doi.org/10.22214/ijraset.2021.36501

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

Automatic Vehicle Overspeed Detection and Recognition

Md Khaja Moinuddin[1], Jatin Gulrajani[2], Karthik Naik[3], Dr. Ameet chavan[4]
[1, 2, 3]UG.Student, ECE, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad

Abstract: ***Vehicle speed detection is an important part of traffic management system. It plays an important role in tracking systems, automatic video surveillance and also to avoid collision. This paper proposes a method to estimate the speed of vehicles on the highways and city areas. The proposed method can be effectively implemented to control the over speed vehicles and to found guilty in leading to traffic accidents. Each vehicle in the video recorded by the camera is identified. A bounding box is created on the identified vehicle and its centroid coordinates are marked. The analysis of speed is done using mathematical formulae which are embedded in the software. The existing research in this field has certain limitations. The first limitation is consumption of a lot of memory to store videos in the hard drive. The second limitation is inaccuracy of the system in unpleasant weather conditions such fog, haze, rain, and heavy winds, etc. Some systems failed to crate proper bounding box as it is necessary for accurate analysis of the motion of the vehicle and its speed. Another disadvantage is that shadow produced by vehicles on different lanes of the road causes a commotion, and the system detects the shadow as a different object and creates a bounding box over it.There are other hardware based methods such as radar gun also.Cosine errors occurred when the direction of the vehicle and the radar gun doesn't match. The objective of the proposed work is to develop a system which can provide the alternative to the radar based systems which can detect multiple vehicles at the same time. We evaluated the proposed method on various traffic videos and discovered that it accurately detects the speed of a vehicle and performs many state-of-the-art approaches.***

Keywords: Speed estimation, Magnetic Sensor, Background Subtraction, and Multiple Vehicle Detection are some of the features available.

I. INTRODUCTION

In recent years, much research has been performed for developing real time traffic monitoring systems for managing the traffic flow of roadways, prevention of accidents, providing secure transportation, etc. Within these works, one aim is to realize different applications such as estimation of vehicle speeds on the roadways, determination of traffic intensity and if necessary, to direct the vehicles to less dense roads, manage the lighting times of traffic lights automatically, etc. . But according to our literature survey, side view images hadn't been used before for speed estimation in these applications except in . The methods given in use roadside line scan cameras for speed estimation and use different methods from ours. In this paper, we present some of the first results of our ongoing research project on the problem of real time estimation of moving vehicles by using side view video images. To find vehicle speed, any digital video camera which acquires images in visible light spectrum may be used. Frame sampling rate, geometric and radiometric resolutions, and distortion amounts of the optical system of the camera affect the precision of the estimated speeds.

Solutions and the models to be used for speed estimation problem vary according to the applications and their final purposes. When applications related to vehicle speed estimation problems are investigated, two main fields are distinguished: traffic surveillance and driver assistance systems or intelligent vehicle systems . Traffic surveillance systems generally involve those applications which require global information on the general traffic situation of the roadways rather than individual vehicles travelling on the roads. For example, estimation of speed of traffic flow of a roadway at different times and dates , belongs to this group, as well as determination of the traffic density, timing of the traffic lights, signalisation works, etc. On the other hand, there are different applications which require speed information of each individual vehicle on traffic scenes. Furthermore, driver assistance systems and intelligent vehicle systems necessitate individual vehicle speeds.

The starting point of many works for traffic surveillance applications are based on the segmentation of the moving objects, and for this purpose background subtraction methods are mostly used . For this purpose, each pixel of the successive frame images are subtracted such that $I(x,y,t) - I(x,y,t + \Delta t)$. the absolute value of this subtraction operation is used. In order to eliminate the object shadows, some other operations are often performed on the segmented images.

In this paper, we examine the problem of real time speed estimation of one moving vehicle from side view video images. The proposed solution to this problem may be used directly for traffic law enforcement to prevent the drivers from exceeding the speed limits. Furthermore, the proposed methods may also be used within a sensor network for active driver assistance and security systems. We are currently developing an intelligent sensor network to be used for both driver assistance and for automatic mobile vehicles . Side view images and the proposed methods will be an important part of this network.

In order to solve the speed estimation problem of an individual vehicle using video frame images, many points which are identified on the image of the vehicle should be selected. Then, the displacement amounts of each selected point between two successive image frames and per unit time, should be found. Those displacement amounts per unit time are essentially equal to the instantaneous speeds of each point. These briefly explained tasks must be performed automatically and also within a very short time period of less than one second. Since the nature of the problem is ill posed, many technical problems relating to the above tasks must be solved. Even if we ignore the physical structure of the problem for a moment, the matters related to the selection process of the points to be tracked and tracking those selected points on the successive image frames involve difficult problems to be solved too. For example, because of the motion of the moving vehicle, if a selected point cannot be seen on the next frame or it falls into the out of vision range of the camera, what should be done Some other problems are how will the time passed be measured? If displacement vectors of the points have been obtained in the image coordinate system, what will be their corresponding absolute values in the object space? It is possible to find solutions to those problems by using different approaches according to the underlying problem. In this paper, all of those problems mentioned above will be handled and we will give the first results of our ongoing studies on the proposed solutions to those problems. In conjunction with this, we will explain the approaches that we used to estimate the speed of a vehicle as well as the image processing procedure that we used to select the tracking points and computation of displacement vectors, *etc.*

II. LITERATURE REVIEW

From the rigorous review of related work and published literature it is observed that many researchers have designed different techniques.

H. Chung-Lin, et al and L. Wen-Chieh, et al presents a new approach to identifying one of the significant applications of video-based supervision systems is the traffic surveillance. So, for many years the researches have investigated in the Vision-Based Intelligent Transportation System (ITS), transportation planning and traffic engineering applications to extract ueful and precise traffic information for traffic image analysis and traffic flow control like vehicle count,vehicle trajectory, vehicle tracking, vehicle flow, vehicle classification, traffic density, vehicle velocity, traffic lane changes, license plate recognition, etc. In the past, the vehicle detection, segmentation and tracking systems used to determine the charge for various of vehicles for automation toll levy system.

N. K. Kanhere et al. and S. T. Birchfield et al. introduced "Real-Time Incremental Segmentation and Tracking of Vehicles at Low amera ngles Using Stable Features," Intelligent Transportation Systems. with automatic camera calibration, also related work on "Vehicle type classification from visual-based dimension estimation," in Intelligent Transportation Systems. The Intelligent Transportation System (ITS) provides services related to different modes of transport and traffic management systems with an integration of traffic control centers. Video-Based investigation for traffic surveillance has been a vital part of ITS. The traffic surveillance in urban environment have become more challenging compared to the highways due to various factors like camera placement, cluttered background, pose variation, object occlusion and illumination changes. This paper provides review on video-based vehicle surveillance for detection, tracking and behavior analysis with systematic description. In this survey we classify the dynamic attributes of vehicle with respect to vehicle motion and appearance characteristics, including velocity, direction of movement, vehicle trajectories on a single camera.[2,3-4].

W. Wei,et al, and K. H. Lim,et al Provides vehicle recognition system, it is used to detect (the vehicles) or detect the traffic lanes [4-6] or classify the type of vehicle class on highway roads like cars, motorbikes, vans, heavy goods vehicles (HGVs), buses and etc,[5,6]. However, the traditional vehicle systems may be declines and not recognized well due to the vehicles are occluded by other vehicles or by background obstacles such as road signals, trees, weather conditions, and etc., and the performance of these systems depend on a good traffic image analysis approaches to detect, track and classify the vehicles. Recently, B. Han, et al proposed an enhanced version of Motion-segmentation-based change detection. The detection of moving object's regions of change in the same image sequence which captured at different intervals is one of interested fields in computer vision. The change detection is used in a wide range of applications, including video surveillance, medical diagnosis and treatment, remote sensing, underwater sensing, and civil infrastructure [16]. The traffic image analysis, which included the moving/motion vehicle, is one of the video surveillance branches. ApproacheS for Detection and Segmentation Even though numerous research papers have been published for moving vehicle detection (background subtraction, frame differencing [17-22], and motion based methods), it is still a difficult task to detect and segment vehicles in dynamic scenes.

It consists of three main approaches to detect and segment the vehicle, as mentioned below:

1) Background Subtraction Methods.
2) Feature Based Methods.
3) Frame Differencing and Motion Based methods.

A. *Previous Works*

Using image/video processing and object detection methods for vehicle detection and traffic flow estimation purposes has attracted a huge attention for several years. Vehicle detection/tracking processes have been performed using one of these methodologies[8]:

1) Matching
2) Threshold and segmentation
3) Point detection
4) Edge detection
5) Frame differentiatio
6) Optical flow methods

It can be said that one of the most important researches in object detection fields, which has resulted in the autoscope video detection systems is introduced in [15]. In some works such as [21], forward and backward image differencing method used to extract moving vehicles in a roadway view. Some studies like [17] and [4] proved that the use of feature vectors from image region can be extremely efficient for vehicle detections goals. Some others represented the accurate vehicle dimension estimation using a set of coordinate mapping functions as it can be seen in [16]. Furthermore, some studies have developed a variety of boosting algorithms for object detection using machine learning methods which can detect and classify moving objects by both type and color such as [18] and [19]. Named approaches have both their advantages and disadvantages.

B. *Background Information*

1) *Video Processing:* Video processing is a subcategory of Digital Signal Processing techniques where the input and output signals are video streams. In computers, one of the best ways to reach video analysis goals is using image processing methods in each video frame. In this case, motions are simply realized by comparing sequential frames[7]. Video processing includes pre-filters, which can cause contrast changes and noise elimination along with video frames pixel size conversions[6]. Highlighting particular areas of videos, deleting unsuitable lighting effects, eliminating camera motions and removing edge-artifacts are performable using video processing methods[29]. OpenCv library of python is equipped with functions that allow us to manipulate videos and images. OpenCVPython makes use of Numpy, which is a library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.[34]

2) *RGB to Grayscale Conversion:* In video analysis, converting RGB color image to grayscale mode is done by image processing methods. The main goal of this conversion is that processing the grayscale images can provide more acceptable results in comparison to the original RGB images[11]. In video processing techniques the sequence of captured video frames should be transformed from RGB color mode to a 0 to 255 gray level. When converting an RGB image to a grayscale mode, the RGB values for each pixel should be taken, and a single value reflecting the brightness percentage of that pixel should be prepared as an output[2].
3) *Power-Law Transformation:* Enhancing an image provides better contrast and a more detailed image as compared to a non-enhanced one. There are several image enhancement techniques such as powerlaw transformation, linear method and Logarithmic method. Image enhancement can be done through one of these grayscale transformations. Among them, power-law transformation method is an appropriate technique which has the basic form below.

V = A v γ

(1) International Journal of Electrical Electronics & Computer Science Engineering Special Issue - ICSCAAIT-2018 | E-ISSN : 2348-2273 | P-ISSN : 2454-1222 Available Online at www.ijeecse.com 142 Where V and v are output and input gray levels,

γ is Gamma value and A is a positive constants (in the common case of A=1). The python code that implements power law transformation ispower_law_transformation=cv2.pow(gray,0.6) The second argument is the gamma value. Consequently, choosing the proper value of γ can play an important role in image enhancement process and preparing suitable details identifiable in image.

4) *Canny Edge Detection:* Object detection can be performed using image matching functions and edge detection. Edges are points in digital images at which image brightness or gray levels changes suddenly in amount.[33] The main task of edge detection is locating all pixels of the image that correspond to the edges of the objects seen in the image. Among different edge detection methodologies, Canny algorithm is a simple and powerful edge detection method. Since edge detection is susceptible to noise in the image, first step is to remove the noise in the image with a 5x5 Gaussian filter. Smoothened image is then filtered with a Sobel kernel in both horizontal and vertical direction to get first derivative in horizontal direction (Gx) and vertical direction (Gy)[9]. From these two images, we can find edge gradient and direction for each pixel as follows: Edge_Gradient(G)=√G2x+G2y (2) Angle(θ)=tan−1(Gy/Gx) (3) Gradient direction is always perpendicular to edges. It is rounded to one of four angles representing vertical, horizontal and two diagonal directions. After getting gradient magnitude and direction, a full scan of image is done to remove any unwanted pixels which may not constitute the edge. For this, at every pixel, pixel is checked if it is a local maximum in its neighborhood in the direction of gradient. OpenCV puts all the above in single function, cv2.Canny() [12].
5) *The Kalman Filter:* Images typically have a lot of speckles caused by noise which should be removed by the means of filtration. The Kalman filter is a powerful and useful tool to estimate a special process using some kind of feedback information[14].

The Kalman filter is used to provide an improved estimate based on a series of noisy estimates.

This filter specifies that the fundamental process must be modeled by a linear dynamical structure:

xk = Fk-1xk-1 + wk-1

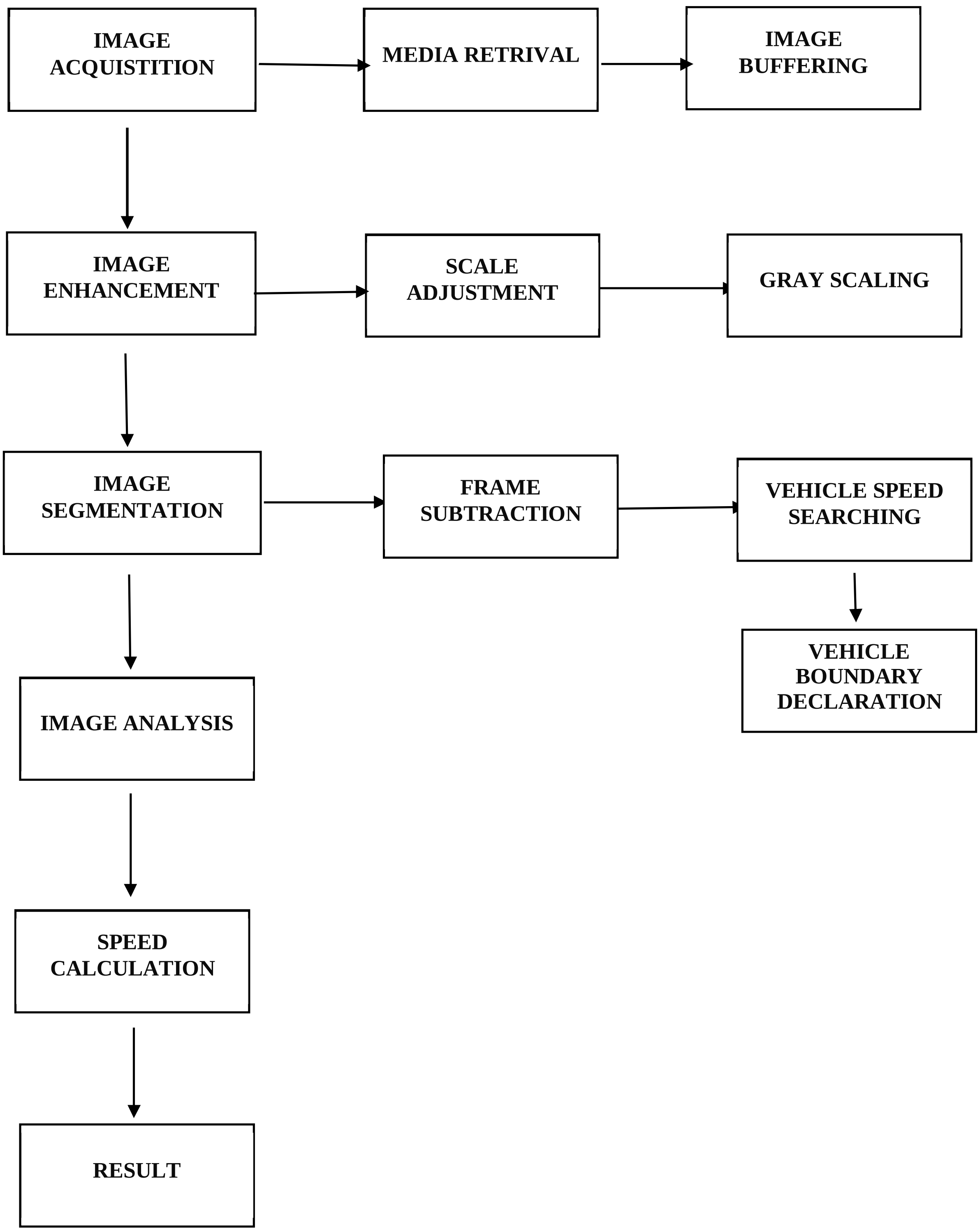
(4) yk = Hkxk + vk

(5) Where xk and yk are the state and measurement vectors, wk and vk are the process and measurement noise, Fk and Hk are the transition and measurement values and k is desired time step[28].

The Kalman filter also specifies that the measurements and the error terms express a Gaussian distribution, which means in vehicle detection each vehicle can only be tracked by one Kalman filter [22],[31]. Therefore the number of Kalman filters applied to each video frame depends on the number of detected vehicles.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

BLOCK DIAGRAM



III. MOTION VEHICLE DETECTION AND SEGMENTATION APPROACHES

A. Video Subsystem Design

Vehicles detection must be implemented at different environment where the light and the traffic status changing. In our proposed system, we accept the traffic flow video from a camera and convert video into frames extract reference backgrounds and performs detection of moving objects. The system we propose consists of three stages.

1) *System Initialisation:* System gets initialised and set up in this stage. Camera records continuous stream of data and sends to the system for analysis
2) *Background Subtraction:* In this stage, a set of frames are taken into focus and on successive analysis and operations background subtraction takes place.
3) *Vehicle Detection:* In this stage, using the subtracted background image all the moving vehicles/objects can be tracked and counted Our system works in two modes, pre-recorded video mode and real-time camera mode. We can provide prerecorded traffic flow video for detection and counting of vehicles. Real time camera mode application accepts the video from the camera and tracks the vehicles. A classification system like the one proposed here can provide important data for a particular design scenario. Our system uses a single camera mounted on a pole or other tall structure, looking down on the traffic scene. It can be used for detecting and classifying vehicles in multiple lanes and for any direction of traffic flow. The system requires only the camera calibration parameters and direction of traffic for initialization.

B. Moving Object Detection In Opencv

OpenCV stands for Open Source Computer Vision Library and is designed in C & C++ specifically for increased computational efficiency, supported by most Operating Systems. OpenCV for providing effective solutions for complex image processing and vision algorithm for real time application for UG and PG students projects. Computer Vision (CV) applications require extensive knowledge of digital signal processing, mathematics, statistics and perception . Example applications of the OpenCV library include Human-Computer Interaction, Object Identification, Segmentation and Recognition, Face Recognition, Gesture Recognition, Camera and Motion Tracking, Ego Motion, Motion Understanding, Stereo and Multi-Camera Calibration and Depth Computation and Mobile Robotics. OpenCV library contains over 500 functions which can be used in above application areas. OpenCV has many powerful image processing functions .

C. System Design

1) *Resolution Setting:* It is expected to check the current screen resolution before capturing the video stream from the web camera. If the resolution is less than 1024 x 768, this application may not produce the desired results. For optimal performance, increase the resolution to 1024 x 768 or higher. The majority of applications require such classification and counting on previously stored videos. For this purpose, the option for counting vehicles from store video is given.
2) *Object Detection:* This section was written in Microsoft Visual C++ with the OpenCV library. The system is intended to begin receiving images from the web camera. Every frame in the video will be processed to find a moving object. The proposed system's activity diagram

a) *Python Programming:* Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes codreadability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a garbage collection system using reference counting. Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible and much Python 2 code does not run unmodified on Python 3. Python 2 was discontinued with version 2.7.18 in 2020.

b) Open CV-Python: Open-cv is a huge open-source library for computer vision, machine learning, and image processing. Open-cv supports a wide variety of programming languages like python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the hand-writing of a human. OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability. Compared to languages like C/C++, Python is slower. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation. OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays.

D. How OpenCV-Python Bindings are Generated

In OpenCV, all algorithms are implemented in C++. But these algorithms can be used from different languages like Python, Java etc. This is made possible by the bindings generators. These generators create a bridge between C++ and Python which enables users to call C++ functions from Python. To get a complete picture of what is happening in background, a good knowledge of Python/C API is required. A simple example on extending C++ functions to Python can be found in official Python documentation[1]. So extending all functions in OpenCV to Python by writing their wrapper functions manually is a time-consuming task. So OpenCV does it in a more intelligent way. OpenCV generates these wrapper functions automatically from the C++ headers using some Python scripts which are located in modules/python/src2. We will look into what they do.

First, modules/python/CMakeFiles.txt is a CMake script which checks the modules to be extended to Python. It will automatically check all the modules to be extended and grab their header files. These header files contain list of all classes, functions, constants etc. for that particular modules.

Second, these header files are passed to a Python script, modules/python/src2/gen2.py. This is the Python bindings generator script. It calls another Python script modules/python/src2/hdr_parser.py. This is the header parser script. This header parser splits the complete header file into small Python lists. So these lists contain all details about a particular function, class etc. For example, a function will be parsed to get a list containing function name, return type, input arguments, argument types etc. Final list contains details of all the functions, enums, structs, classes etc. in that header file.

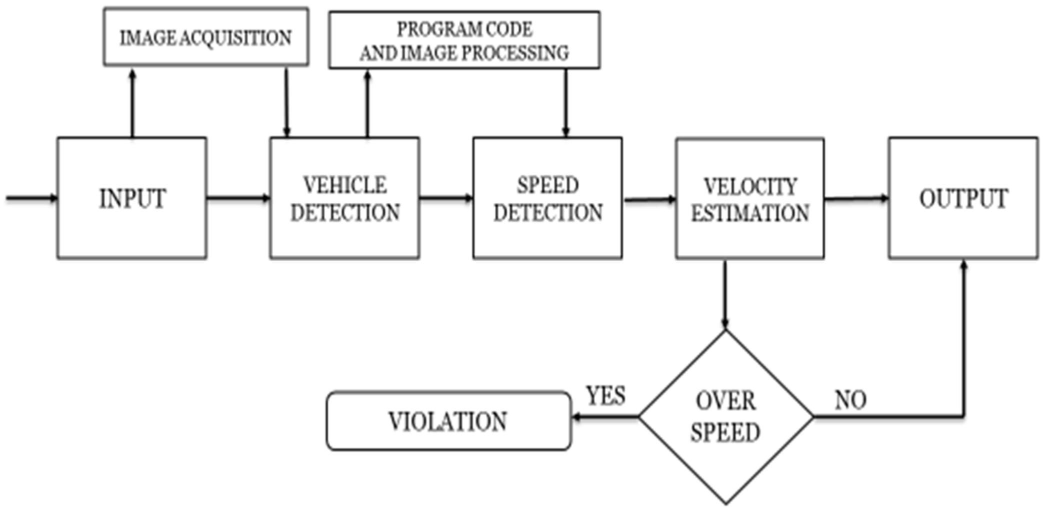
But header parser doesn't parse all the functions/classes in the header file. The developer has to specify which functions should be exported to Python. For that, there are certain macros added to the beginning of these declarations which enables the header parser to identify functions to be parsed. These macros are added by the developer who programs the particular function. In short, the developer decides which functions should be extended to Python and which are not. Details of those macros will be given in next session.

So header parser returns a final big list of parsed functions. Our generator script (gen2.py) will create wrapper functions for all the functions/classes/enums/structs parsed by header parser (You can find these header files during compilation in the build/modules/python/ folder as pyopencv_generated_*.h files). But there may be some basic OpenCV datatypes like Mat, Vec4i, Size. They need to be extended manually. For example, a Mat type should be extended to Numpy array, Size should be extended to a tuple of two integers etc. Similarly, there may be some complex structs/classes/functions etc. which need to be extended manually. All such manual wrapper functions are placed in modules/python/src2/cv2.cpp.

So now only thing left is the compilation of these wrapper files which gives us cv2 module. So when you call a function, say res = equalizeHist(img1,img2) in Python, you pass two numpy arrays and you expect another numpy array as the output. So these numpy arrays are converted to cv::Mat and then calls the equalizeHist() function in C++. Final result, res will be converted back into a Numpy array. So in short, almost all operations are done in C++ which gives us almost same speed as that of C++.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com

BLOCK DIAGRAM



IV. ADVANTAGES AND DISADVANTAGES

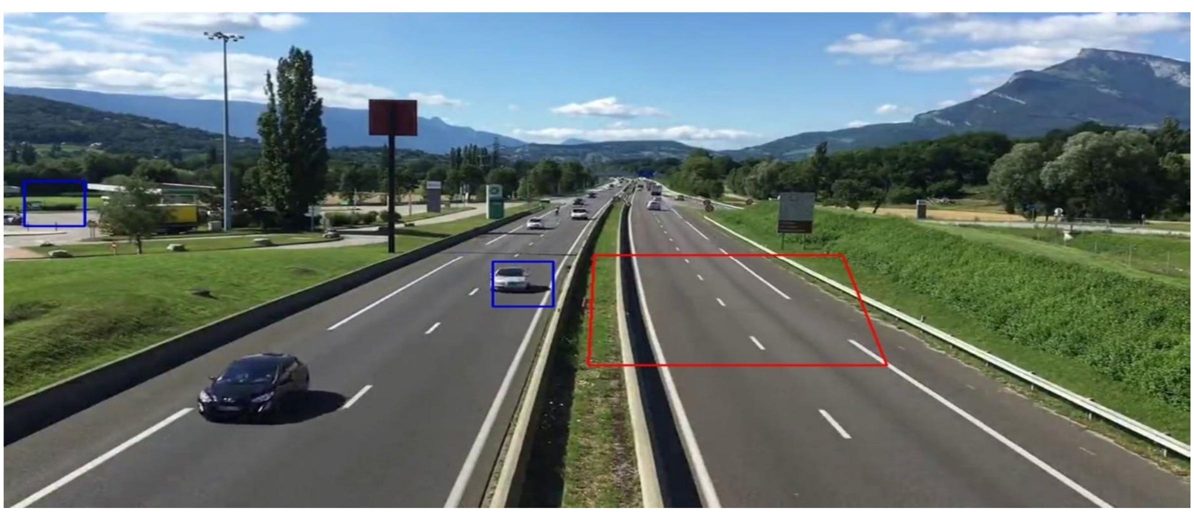
A. Advantages

1) OpenCV is completely free to use.
2) Because the OpenCV library is written in C/C++, it is extremely fast. It can now be used with Python.
3) It requires less RAM to operate, possibly as little as 60-70 MB
4) Computer Vision, like OpenCV, is portable and can run on any device that supports C.

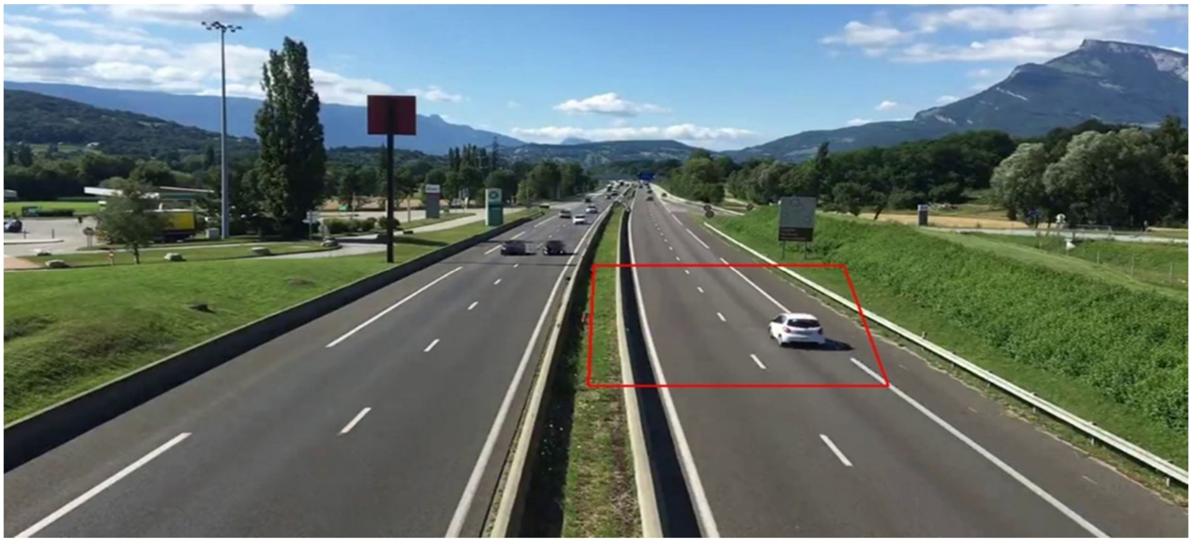
B. Disadvantages

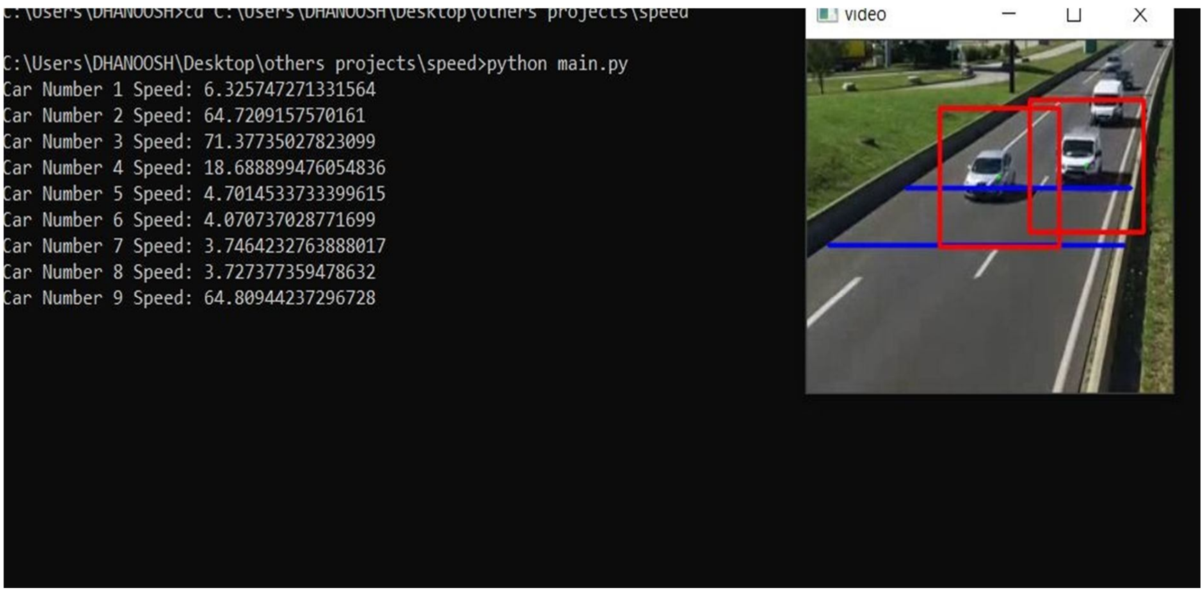
1) When compared to MATLAB, OpenCV does not provide the same ease of use.
2) OpenCV has its Flann library. This causes problems when attempting to use the OpenCV library alongside the PCL library.

V. RESULT



International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VII July 2021- Available at www.ijraset.com





VI. CONCLUSION

Due to increase in expressway, highways and traffic congestion, there is a huge amount of potential applications of vehicle detection and tracking on expressway and highways. In this paper we have demonstrated vision based system for effective detection and counting of vehicles running on roads. The main aim of our system is to detect the moments of vehicles by analyzing camera pictures with the help of computer vision. Vehicle counting process accepts the video from single camera & detects the moving vehicles and counts them. Vehicle detection and counting system on highway is developed using OpenCV image development kits.

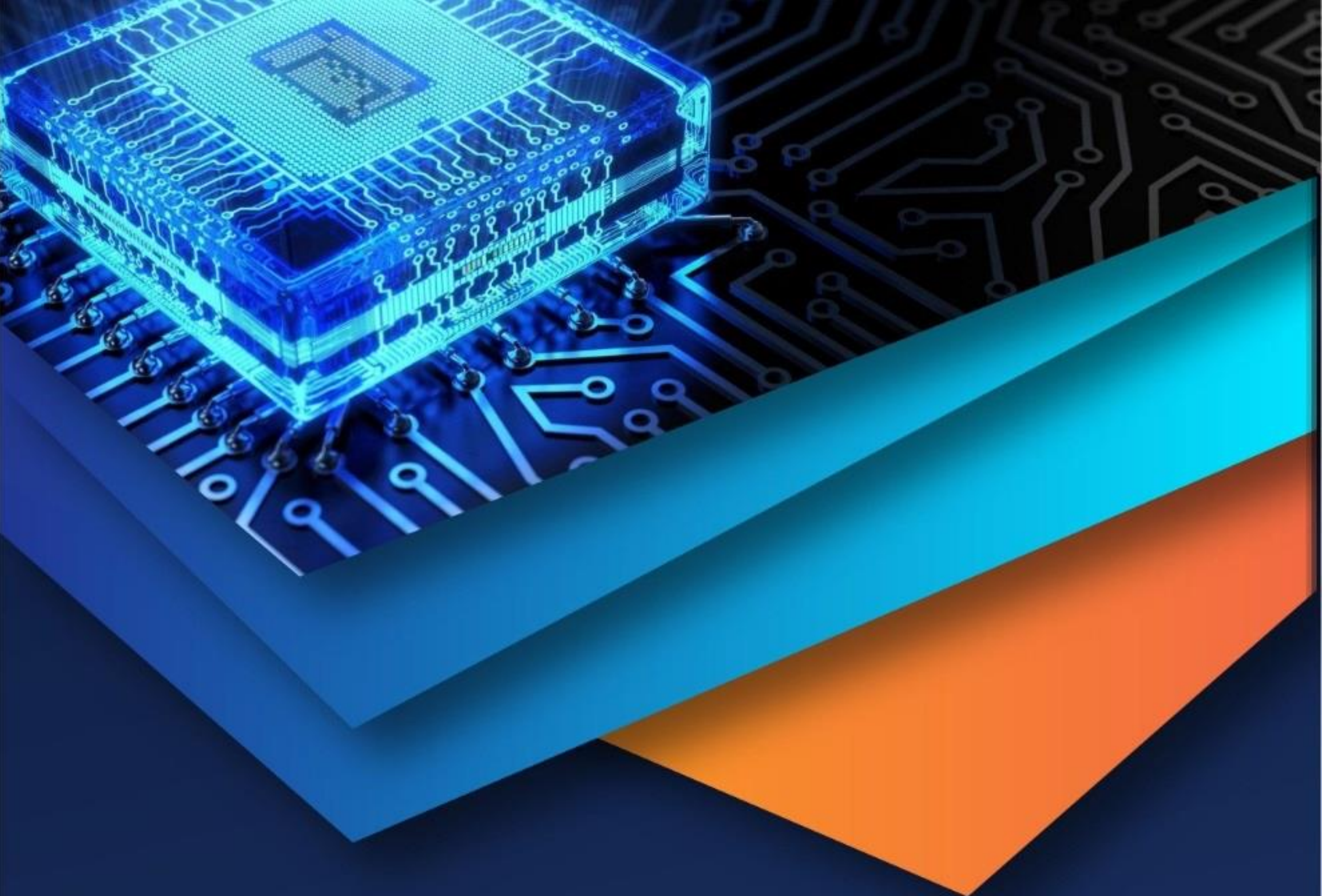
we propose that the Kalman filter algorithm is capable of estimating the accurate speed of the moving vehicle. Gaussian mix model was collaborated along with this algorithm for making accurate depiction of the moving objects. The combination of optical stream and the Kalman channel helps in predicting the results even when there is a low picture quality. In our future research, we aim to improve the DBSCAN division in order to recognize each article in gathering of the vehicles and also use flexible heaps of pixels for perceiving the speed from vertical advancements

VII. FUTURE WORK

Creating a multiple object tracking system that would perform vehicle classification and thus separate vehicles by type, compute origin-destination matrices by identifying each vehicle and tracking it through a monitored road traffic network, and so on.

REFERENCES

[1] H. Chung-Lin and L. Wen-Chieh, "A vision-based vehicle identification system," in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, 2004, pp. 364- 367 Vol.4.
[2] Z. Wei, et al., "Multilevel Framework to Detect and Handle Vehicle Occlusion," Intelligent Transportation Systems, IEEE Transactions on, vol. 9, pp. 161-174, 2008.
[3] N. K. Kanhere and S. T. Birchfield, "Real-Time Incremental Segmentation and Tracking of Vehicles at Low Camera Angles Using Stable Features," Intelligent Transportation Systems, IEEE Transactions on, vol. 9, pp. 148-160, 2008.
[4] N. K. Kanhere, "Vision-based detection, tracking and classification of vehicles using stable features with automatic camera calibration," ed, 2008, p. 105.
[5] H. S. Lai, et al., "Vehicle type classification from visualbased dimension estimation," in Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE, 2001, pp. 201-206.
[6] Z. Zhigang, et al., "A real-time vision system for automatic traffic monitoring based on 2D spatio- temporal images," in Applications of Computer Vision, 1996. WACV '96., Proceedings 3rd IEEE Workshop on, 1996, pp. 162-167.
[7] W. Wei, et al., "A method of vehicle classification using models and neural networks," in Vehicular Technology Conference, 2001. VTC 2001 Spring. IEEE VTS 53rd, 2001, pp. 3022-3026 vol.4.
[8] Cathey F.W., Dailey D.J. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. Proceedings of the IEEE Intelligent Vehicles Symposium; Las Vegas, NV, USA. 6–8 June 2005; pp. 777–782. [Google Scholar]
[9] Grammatikopoulos L., Karras G., Petsa E. Automatic estimation of vehicle speed from uncalibrated video sequences. Proceedings of the Int. Symposium on Modern Technologies, Education and Professional Practice in Geodesy and Related Fields; Sofia, Bulgaria. 3–4 November 2005; pp. 332–338. [Google Scholar]
[10] Melo J., Naftel A., Bernardino A., Victor J.S. Viewpoint independent detection of vehicle trajectories and lane geometry from uncalibrated traffic surveillance cameras. Proceedings of the International Conference on Image Analysis and Recognition; Porto, Portugal. 29 September–1 October 2004; pp. 454–462. [Google Scholar]
[11] Morris B., Trivedi M. Improved vehicle classification in long traffic video by cooperating tracker and classifier modules. Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance; Sydney, Australia. 22–24 November 2006. [Google Scholar]
[12] Douxchamps D., Macq B., Chihara K. High accuracy traffic monitoring using road-side line scan cameras. Proceedings of the IEEE Intelligent Transportation Systems Conference (ITLS 2006); Toronto, Canada. September 2006; pp. 875–878. [Google Scholar]
[13] Gupte S., Masaud O., Martin F.K.R., Papanikolopoulos N.P. Detection and classification of vehicles. IEEE Trans. Intell. Transp. Syst. 2002;3:37–47. [Google Scholar]
[14] Guo M., Ammar M.H., Zegura E.W. V3: A vehicle to vehicle live viedo streaming architecture. Pervasive Mob. Comput. 2005;1:404–424. [Google Scholar]
[15] Dailey D.J., Cathey F.W., Pumrin S. An algorithm to estimate mean traffic speed using uncalibrated cameras. IEEE Trans. Intell. Transp. Syst. 2000;1:98–107. [Google Scholar]



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY