Proceedings of the Second International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2020)
IEEE Xplore Part Number: CFP20K58-ART; ISBN: 978-1-7281-4167-1

Real Time Vehicle Detection, Tracking and Counting Using Raspberry-Pi

Apeksha P Kulkarni
School of Computer Science and Engineering
KLE Technological University, Vidyanagar
Hubballi-580031, Karnataka, India
apekshapk17@gmail.com

Vishwanath P Baligar
School of Computer Science and Engineering
KLE Technological University, Vidyanagar
Hubballi-580031, Karnataka, India
vpbaligar@kletech.ac.in

***Abstract*—Population explosion leads to an unprecedented increase in the number of physical objects or vehicles on road. As a result, the number of road accidents increases due to a very heavy traffic flow. In this paper, traffic flow is monitored by using computer vision paradigm, where images or sequence of images provides a betterment on the road view. In order to detect vehicles, monitor and estimate traffic flow using low cost electronic devices, this research work utilizes camera module of raspberry pi along with Raspberry Pi 3. It also aims to develop a remote access using raspberry-pi to detect, track and count vehicles only when some variations occur in the monitored area. The proposed system captures video stream like vehicles in the monitored area to compute the information and transfer the compressed video stream for providing video based solution that is mainly implemented in Open CV by Python Programming. The proposed method is considered as an economical solution for industries in which cost-effective solutions are developed for traffic management.**

Keywords—ComputerVision; Traffic flow; OpenCV;Vehicle Detection; Vehicle Track; Vehicle Count.

I. INTRODUCTION

It is observed that, the major hindrance on road is due to heavy traffic flow during peak hours especially when people commute to work. The total number of vehicles or objects exceeds its capacity by causing a blockage for emergency vehicles such as fire fighter and rescue vehicles, furthermore wastage of fuels adds more to the environmental pollution which is not adoptable for a country's economic growth. In order to develop an efficient, reliable, cleaner and safer mode of transportation, it is necessary to make the road transportation system automated as much as possible.

The primary research focus is dedicated towards the detection and tracking of objects, where it finally keeps the count of vehicles in the particular monitored area. The need of traffic surveillance system is to provide construction engineers and other associates to plan in an economical way and proper decisions were taken based on density of the vehicles and the statistics obtained by the low cost electronic devices [1]. Moreover, it also provides solution to major problems such as vehicle accidents, vehicle theft detection, managing parking areas, and other security threats.

The main cause for the interest in traffic management activity is to utilize the computer vision techniques in real-time conditions. [8]

The major challenge that caused hindrance to our work is vehicle segmentation in various atmospheric conditions such as night, snowy or dusty weather conditions. As a solution to it we have used a different [2] pre-processing unit based on Histogram Equalization to improve the resolution of video and morphological processing to add or remove pixels in the boundaries of objects, where video depends on shape and size of the structuring elements before processing towards the next stage.

It is also observed that the vehicles moving towards same point either in lighter or darker region or vice-versa might have the same colour as background for detecting vehicle remains more challenging and this leads to fault in count of vehicles [7]. So, as a solution to it we have employed background subtraction technique to register the vehicle ID if it crosses the given threshold.

II. PROPOSED SYSTEM

This paper uses a well-known platform called Linux server for surveillance and recording video using Raspberry-Pi. The approach uses Raspberry pi to record a video when something moves inside the monitored area.

The proposed system efficiently manages to distinguish the vehicles from the surrounding environmental variability and improves the low resolution videos through Histogram equalization technique in order to maintain uniformity of videos in terms of resolution and also in removal of noise from videos. Then the implementation of background

subtraction algorithm helps to detect objects and track them based on the particle filter algorithm. Tracked vehicles are counted based on the threshold given to different sized vehicles based on the area.

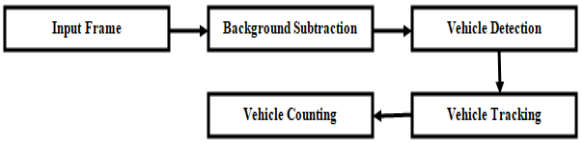

Figure 1: Proposed system design

III. SYSTEM DESIGN

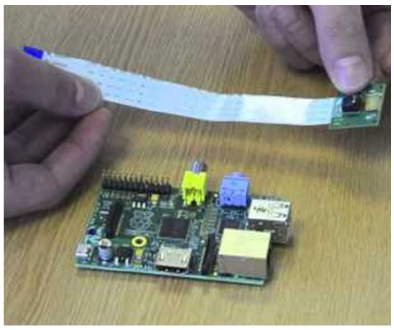

Figure 2: Raspberry-Pi

This paper makes use of as shown in figure 2

1. **Raspberry-Pi 3**
   Raspberry-Pi 3 is 512 RAM with 1.2GHz.
2. **Raspberry-Pi camera module**
   Raspberry-Pi has a connector to plug in a camera module to capture the video.
3. **Power supply**
   Raspberry-Pi uses 400mA of current that plugs the micro USB.
4. **Micro SD Card**
   Raspberry-Pi uses SD Card to store and install libraries and run operating system for this device. Minimum 64GB or higher is required
5. **Wi-Fi USB adaptor**
   It establishes a connection between camera and network [2].

Remote access of Raspberry-Pi

To operate under Raspberry-Pi, it is necessary to install all the recent features and drivers for updated operating system with a correct access to an internet connection.

Commands to be followed in Raspberry-Pi

$ sudo apt-get update
This command updates the operating system with recent features and drivers.

$ sudo raspi-config
raspi-config opens the configuration tool that is written and maintained by raspbian operating system.

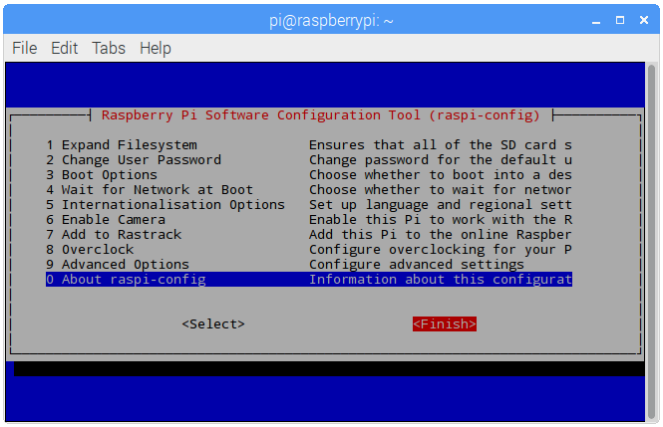

Figure 3: Raspberry-Pi configuration

Raspi camera module can be enabled by using Up and Down keys as shown in figure 3

To capture image from camera module use command
$ raspistill -o veh_img.jpg
Raspistill captures the still images where –o indicates the output to be saved in veh_img in jpg format.

To capture video from camera use command.
$ raspivid -o video.h264 -t 1000
Raspivid captures the videos for 1000sec in encoded in h264 format.

Open RaspiCam Remote app provide login details such as IP address, username and password as shown in figure 4.

Figure 4: Remote login for surveillance

IV. SYSTEM IMPLEMENTATION

Proposed system mainly makes use of three important modules namely

A. Vehicle detection
B. Vehicle tracking
C. Vehicle counting

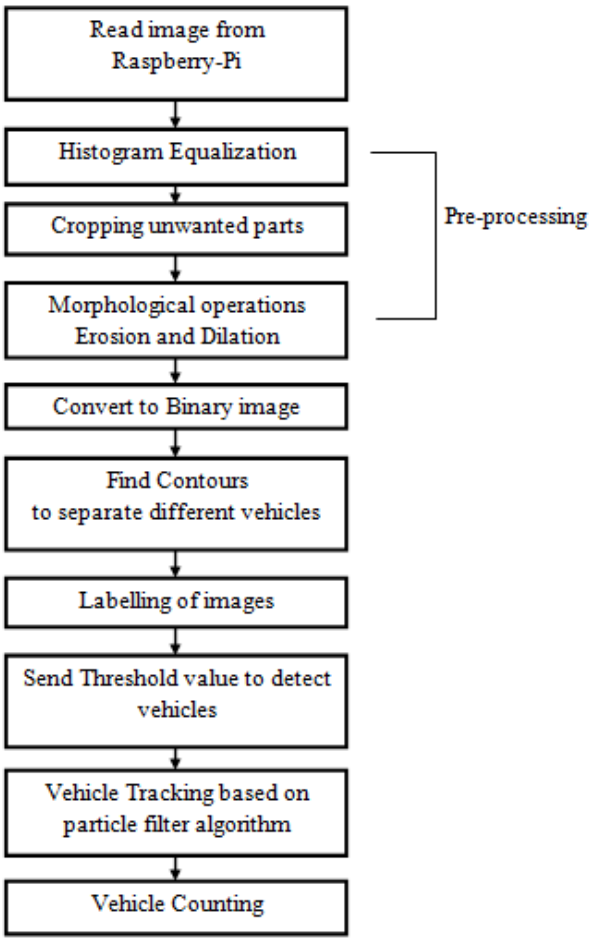

Figure 5: Flowchart for Proposed System

A. Vehicle Detection

Computer vision paradigm provides vision for identifying objects that belong to classes that might be a vehicle or person in a video. The use of object detection in computer vision paradigm [4] is to solve real world challenges in areas like image search and video surveillance to detect a person or vehicle.

To detect vehicles, we subtract image of the road without vehicles from another image having vehicles on road. The background pixels would cancel each other out and the vehicles or objects present in the foreground will appear.

Background subtraction is easy to implement and suitable threshold is given for detecting vehicles is an area that remains less than 20000. Figure 6 shows objects within the given threshold are detected. It is observed that the vehicles are not the only objects that moves on/across the road. There are pedestrians, animals, people with carts etc. Also, objects change in size and shape as they move across the view [5].

Figure 7 shows the masked image of the detected objects. Masking is done to highlight the desired objects in a video frame. The objects with an area smaller than the threshold value are ignored. Selecting Region of Interest (ROI) helps in detecting objects, for vehicle detection and further implementation is done by tracking the detected vehicles.

A. Background Subtraction

Background subtraction is process of extracting the target image from original image. ID

Origin: It is the original image is coloured or grayscaled image of 8-bit or 32-bit floating point.
Target: The target image is either 32-bit or 64-bit floating point.
Alpha: Weight of input image. Speed of updation is decided by alpha, set a lower value for this variable in existing frames.

$$Target(x,y) = (1-alpha).target(x,y) + alpha.origin(x,y) \quad (1)$$

B. Vehicle Tracking

Path followed by an object with the purpose to determine the observed direction of target on a near real- time surveillance and security for traffic control without affecting human computer intervention.
The major goal of tracking is to determine the target objects in sequential frames of video. Object change in shape and size over time in such scenarios [3] so motion model for recovering trajectories and models with high accuracy for a small number of vehicles.

Bounded boxes around a detected object are seen. Centroid of bounding boxes determines the object detected. For For tracking current objects we match the input centroids to existing object centroids and compute the distance between each pair of object such as Euclidean distance [8]. The object needs to be registered if the number of consecutive frames of the objects is disappeared. In order to register a new centroid as a trackable object it must satisfy condition that input centroids should be greater than the number of existing centroids.

C. Vehicle Counting

Vehicles tracked are counted when they leave the frame or cross a line at an exit point of the frame [4]. To count vehicles moving in two different directions we make use of counting lines that is down count as red line and up count as blue line. Counted vehicles are classified based on the perimeter, if the perimeter of bounding box is less than 300 it is counted as bike, if the perimeter of bounding box is less than 500 it is counted as car and if the perimeter of bounding box is greater than 500 it is counted truck/bus. Figure 8 shows the vehicles counted and classified based the threshold given to them.

V. RESULT ANALYSIS

From experimental analysis it is evident that vehicles detected at an accuracy rate of about 97.39% and vehicle tracked at a rate of about 98.26% as shown below.

Figure 6 shows objects within the given threshold are detected. It is observed that vehicles are not the only objects that move on/across the road.

Figure 7 shows the masked image of objects detected. Masking is done to highlight the desired objects in a video frame.

Figure 8 shows the vehicles counted and classified based the threshold given to them.

Finally information about the vehicles crossed with height, width and vehicle ID is maintained only to track the vehicles for counting either moving upwards or downwards as shown in figure 9.

Figure 6: Image threshold for video frame

Figure 7: Masked image for video frame

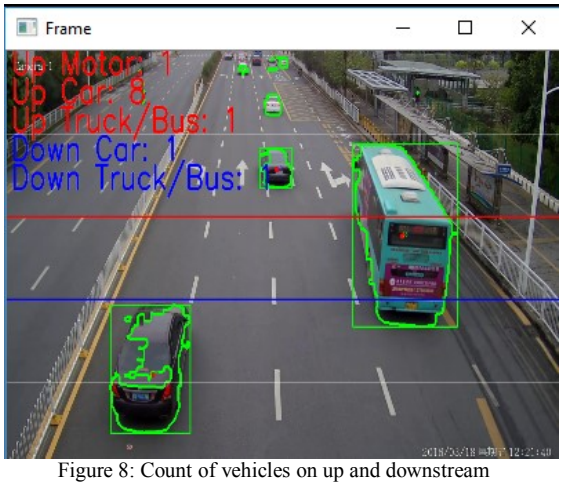

Figure 8: Count of vehicles on up and downstream

```
ID: 147 crossed going up at Mon Aug 26 19:14:22 2019
Height::: 66
Width::: 40
ID: 142 crossed going up at Mon Aug 26 19:14:23 2019
Height::: 105
Width::: 92
ID: 140 crossed going up at Mon Aug 26 19:14:24 2019
EOF
UP: 45
DOWN: 13
```

Figure 9: Information of vehicles crossed

TABLE I
Proposed system count with accuracy rate

A	B	C	D	E	F	G	H
60	1	3	4	4	4	100%	100%
300	2	7	9	9	9	100%	100%
450	11	6	17	18	18	94.4%	100%
1401	17	21	38	40	42	95%	95.23%
1812	34	48	82	85	81	96.47%	95.29%
2512	67	45	112	115	113	97.39%	98.26%

A - Number of input frames.
B - Up count
C - Down count
D - Total number of vehicles in truth
E - Vehicle detected
F - Vehicle tracked
G - Accuracy of vehicle detection
H - Accuracy of vehicle tracking

VI. CONCLUSION AND FUTURRE SCOPE

In this paper, we have presented the unitized techniques to achieve improvement and outperformance in the vehicle detection and counting process. Major technique used to improve detection of vehicles is the use of background subtraction algorithm. The proposed method eliminates the unnecessary portion and differentiates the vehicles in a more accurate manner. Further, we track vehicles in each frame based on the information obtained from previous frame. Experimental results, implemented with Open CV, indicates that the accuracy rate reaches to 97.1% for object detection, and 98.4% for object tracking. From the table, we observe that the proposed method is effective to detect, track, and count moving vehicles in a more accurate and

successful manner irrespective of weather conditions such as night, snowy or dusty weather conditions [2].

As a future work, major improvements can be included by upgrading to higher versions of raspberry pi. This greatly reduces the processing time. Traffic on highways cause occlusions, here two vehicles are combined to treat two vehicles as a single entity. Due to heavy wind, camera might get affected due to vibrations. This leads to partial detection of vehicles. Alarm system might be incorporated as an additional future enhancement.

References

[1] Li, D., Liang, B., & Zhang, W. (2014). Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV. 2014 4th IEEE International Conference on Information Science and Technology.

[2] Sorwar, T., Azad, S. B., Hussain, S. R., & Mahmood, A. I. (2017). Real-time Vehicle monitoring for traffic surveillance and adaptive change detection using Raspberry Pi camera module. 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC).

[3] Chauhan, N. S., Rahman, F., Sarker, R., & Pious, M. M. H. (2018). Vehicle detection, tracking and counting using linear quadratic estimation technique. 2018 2nd International Conference on Inventive Systems and Control (ICISC).

[4] Yang, H., & Qu, S. (2017). Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition . IET Intelligent Transport Systems.

[5] Soleh, M., Jati, G., Sasongko, A. T., Jatmiko, W., & Hilman, M. H. (2017). A real time vehicle counting based on adaptive tracking approach for highway videos.

[6] Chhadikar, N., Bhamare, P., Patil, K., & Kumari, S. (2019). Image processing based Tracking and Counting Vehicles. 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA).

[7] Robert, K. (2009). Video-based traffic monitoring at day and night vehicle features detection tracking. 2009 12th International IEEE Conference on Intelligent Transportation Systems.

[8] Aqel, S., Hmimid, A., Sabri, M. A., & Aarab, A. (2017). Road traffic: Vehicle detection and classification. 2017 Intelligent Systems and Computer Vision (ISCV).

[9] Foresti, G. L., & Snidaro, L. (2005). Vehicle Detection and Tracking for Traffic Monitoring. Lecture Notes in Computer Science.

[10] https://www.researchgate.net/publication/281039935_Real-time_moving_vehicle_detection_tracking_and_counting_system_implemented_with_OpenCV.

[11] Arya, K. V., Tiwari, S., & Behwalc, S. (2016). Real-time vehicle detection and tracking. 2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).

[12] Chen, Z., Ellis, T., & Velastin, S. A. (2012). Vehicle detection, tracking and classification in urban traffic. 2012 15th International IEEE Conference on Intelligent Transportation Systems.