2019 4th International Electrical Engineering Conference (IEEC 2019)
Jan, 2019 at IEP Centre, Karachi, Pakistan

Efficient traffic control system

[1]Syed Umaid Ahmed, [2]Talha Ashraf and [3]Farah Kareem

[1] Department of Electrical Engineering, NED University of Engineering and Technology,
Karachi, 75290, Pakistan (syedumaidahmed96@gmail.com)

[2]Department of Electrical Engineering, NED University of Engineering and Technology,
Karachi, 75290, Pakistan (talhaashraf987@gmail.com)

[3]Department of Electronic Engineering, NED University of Engineering and Technology,
Karachi, 75290, Pakistan (farikareem@gmail.com)

Abstract: This research paper presents an innovative project for the betterment of our existing traffic regulation and surveillance system by the endorsement of an effective law system through a systematic procedure. This project will count the number of vehicles, capture their pictures, calculate their speed along with the time and date when it crosses our system. Thus, the timing of the traffic signals can be controlled so as to facilitate traffic jams. The most alluring aspect of this project is its capability of detecting the movement of wrong-way vehicles and dispatching a notification to the administrator alarming him of the situation. This project aims to replace an obsolete traffic management system with a cost friendly and an efficient system using a Raspberry Pi model 3B as a microcomputer along with the OpenCV library used to add webcam to the system. The motion of vehicles will be detected, activating our system, thus, recording the aforementioned specification to create a database. Algorithms were implemented in the microprocessor to carry out the procedures.

Keywords: Traffic surveillance and regulation, vehicle detection, law endorsement, tracking, Raspberry Pi model 3B.

I. INTRODUCTION

Our traffic management system has numerous problems that arise mainly due to increase in the number of vehicles on the road. Traffic congestion and an irregularity of traffic laws are the major problems of our system that needs to be addressed. Different proposals have been coming through time for betterment of this system, mostly less efficient and expensive to implement. Nowadays, many of the previous systems have become outdated, and a user friendly and an economical system has become a necessity. Traffic congestion is a result of mismanagement of the traffic system [1].

Radio Frequency Identification (RFID), [2] which is a wireless technology, has played a huge role in the field of traffic management. It provides a user-friendly and a cost-efficient solution, significantly reducing infrastructure costs. Vehicle identification, security, tracking, and parking management can be addressed with RFID. Urban congestion issues are addressed by the Traffic Management System (TMS) [3] which makes use of smaller system placed in various areas, thus providing time to time remedial solutions for the flaws in the system as a whole. The consideration of traffic lights for controlling traffic congestion, can play an important role in the traffic management system as it can decrease vehicle accidents by regulating high volume traffic roads with low volume traffic [4]. Observing the current systems, our project which is Raspberry Pi based, shall revitalize the existing system with some innovation [5]. It calculates the number of vehicles on a two-way road which helps us to calculate the density of the vehicles on that particular road, which in turn possibly gets us an idea to manage the traffic management system [6][7].

II. DESIGN CONCEPT AND PROCESS

The prime focus of this project is to introduce a system capable of detecting motion of the vehicles along with their speeds on a two-way road using Raspberry Pi model 3B. Our model is programmed with the help of Python programming language and it uses OpenCV library that provides webcam to the microprocessor for image processing [8]. It also uses video streaming that detects fast moving vehicles and tracks their motional activity. Whenever a motion is detected in the form of passing of a vehicle; an image of the vehicle is processed, along with time and date which is further recorded in the database as a CSV file [9]. With this database, we can identify a fast moving vehicle through an interpretation of the data. This how a fast moving vehicle could be taxed.

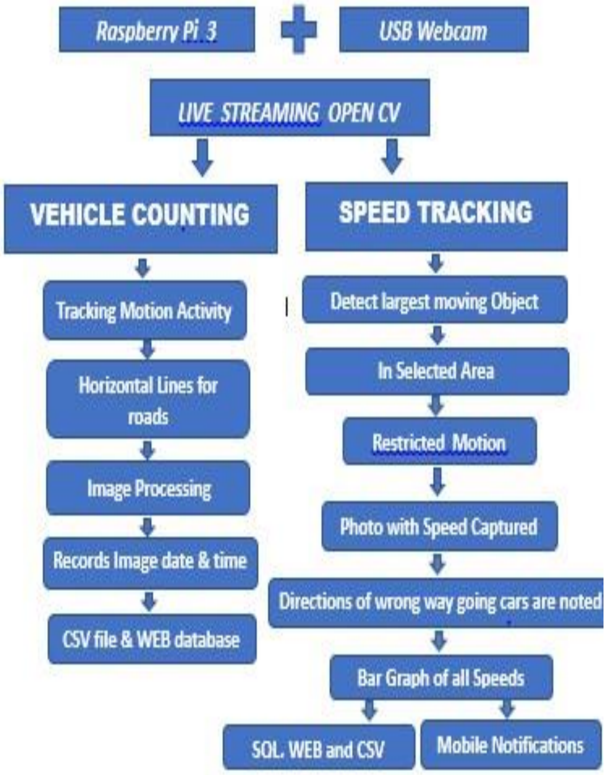


Fig 1. Procedure of the traffic controlling system using Raspberry Pi

Then the restricted mode is used in which selected areas are examined to capture images of vehicles along with their speeds. Wrong-way vehicles and stolen cars can also be identified with a dispatch of a notification to the administrator [10] [11].Bar graphs will be generated of the speed of vehicles along with their time of entries. Fig 1. explains this procedure.

III. METHODOLOGY

Our system makes use of the Raspberry Pi model 3B using OpenCV library. System is designed for getting images from the webcam where moving objects in the video are framed and processed. The activity diagram of the system proposed is shown in Fig 2. Then process of identification of the vehicles is done. A vehicle is considered as a rectangular shape. To identify each vehicle, a rectangular shape of known dimension is drawn on the street so as to align the vehicle on that particular rectangle on the road. Whenever a vehicle passes, An OpenCV background segmentation algorithm (Mixture of Gaussians Segmenter, BackgroundSubtractorMOG2) is run [12]. The initial step of the segmentation process detects the regions that contain unknown objects. In the segmentation process, regions which contain unknown objects are to be detected initially. The extraction of an object occurs in the next step. It is done after extracting certain features from the regions previously segmented. Each object denotes a vehicle that is segmented in a rectangular shape, that particular rectangle's center and area show the features of that vehicle. Each vehicle is counted and tracked using those features. [13].

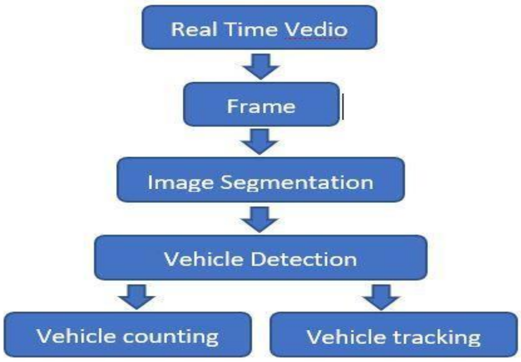


Fig.2: The flow chart of vehicle detection and counting system

1) *Vehicle detection:*

The movement of vehicles in the segmented areas can be easily detected using the process of new edge detection. The vehicles are identified whenever they pass over the assumed rectangular area, which was initially assigned on the road. The video source in Fig. 3 shows the location of a vehicle [14].

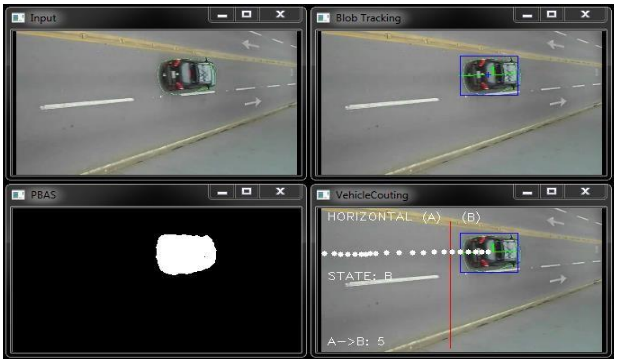


Fig.3. Vehicle Detection

2) *Vehicle counting:*

A vehicle is symbolized by a rectangle. The number of vehicles on the road can be found by counting all the rectangles in each frame, done in the vehicle detection process. The frame in our consideration is split into two portions giving the area of the road that needs to be accessed during the vehicle counting as shown in the Fig. 4. A variable is defined as variable I whose initial value is 0. The variable acts like a counter as its value increases whenever an object passes [15] [16].

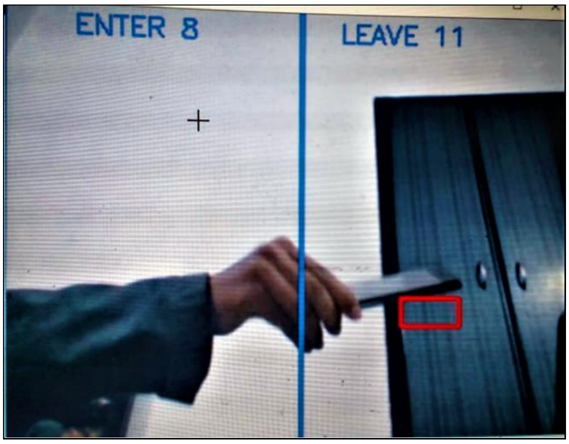


Fig. 4: Vehicle counting

3) *Speed calculation and vehicle tracking:*

Blob tracking method is used to track vehicles. The position of the blob in frames can be found by identifying and tracing the movements of vehicles in the respective frames [17]. This will determine the position of the blob at every instance. The analysis of the movement of the vehicles in the frames also provide us with the average speed of vehicles.

IV. IMPLEMENTATION AND RESULTS

All basic steps covered in the project development are discussed in this section.

A. Hardware Implementation

The project uses a low cost Raspberry Pi Microcomputer and webcam for image processing applications. A Linux based operating system (Raspbian Stretch) burned in a memory card with python scripts helps in accessing the GPIO pins for controlling external hardware. A USB webcam in this project is use for video streaming.

1. Raspberry Pi Board

Fig 5: Raspberry Pi 3B

The Raspberry Pi 3B has an embedded Broadcom system, with a BCM2837 SoC chipset. It has a ARM Cortex-A53 instruction set. It has a quad-core 64-bit processor with a processing speed of 1.2 GHz. For loading Linux operating system, it makes use of a SD RAM of 1 GB already present in it. For data storage, it uses an external micro SD card as it does not include a built-in hard drive. It also includes a 10/100 Mbps Ethernet socket for the connection of the internet to Raspberry Pi. Our model possesses GPU Videocore-IV 400 MHz multimedia co-processor, Bluetooth 4.1, 802.11 b/g/n Wireless LAN, 40 GPIO pins, 4 USB ports and CSI camera port, connecting our camera with Raspberry Pi. We choose Raspberry Pi over any high level controller such as Arduino because of the processing capabilities of Raspberry Pi as it has a 1.2 GHz processor. Another reason for using this model is that it is the most powerful image processing tool incorporating heavy tasks which cannot be carried out using any other high level micro controller. Its Wi-Fi module, USB and image processing cameras, gets itself an edge against other controllers which lack these features.

2. UVC Driver Camera

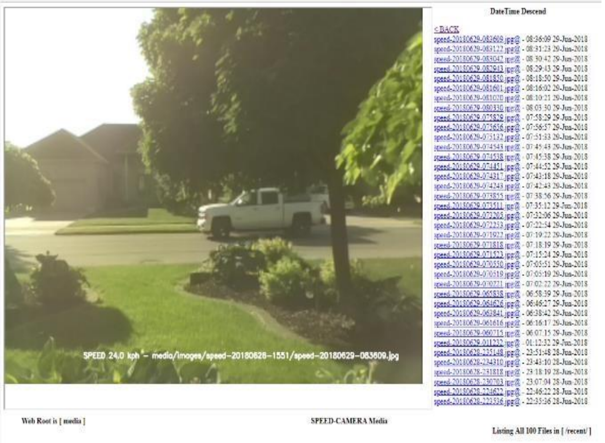
A UVC (Universal Video Class) provides driver support for USB video class compatibility devices. It implements externally connected cameras (webcam) to any other device (here in our case, it is Raspberry Pi). It provides access to Raspberry Pi to assess and control videos made by webcam.

B. Software implementation

This is a Raspberry Pi OpenCV vehicle speed camera program. It is written in python programming language and uses OpenCV to detect and track object motion. The results are recorded on speed photos and in a CSV log file that can be imported into another program for additional processing. The program will detect and track motion in the field of view and use OpenCV to calculate the largest counter and return its x, y coordinate. Motion detection is restricted between y-upper and y-lower variables (road area). If a track is longer than track_len_trigger variable, average speed will be calculated (based on IMAGE_VIEW_FT variable) and a speed photo will be taken which is saved in the image folder. If log-data-to-file=True then a speed_track.log file will be created/updated with event data, stored in CSV (Comma Separated Values) format.

C. Website & Database

The complete database of the known and unknown objects is prepared on website. Camera streaming is viewed on the local website using the dependencies such as SQL, PHP and Apache. The administrator can access or change any of the required information. The live steam is also included to monitor outdoor activities. It sends a text message to the administrator whenever a vehicle moves on the wrong side of a road. In case of an issue in the internet connectivity, a log file is also generated on the backend for storing the database. A copy of the database is also sent to the mobile and email address of the administrator after a certain period allotted in the python scripts [18].


SPEED 24.0 kph ~ media/images/speed-20180628-1551/speed-20180629-083609.jpg

DateTime Descend

< BACK
speed-20180629-083609.jpg@ - 08:36:09 29-Jun-2018
speed-20180629-083122.jpg@ - 08:31:23 29-Jun-2018
speed-20180629-083042.jpg@ - 08:30:42 29-Jun-2018
speed-20180629-082943.jpg@ - 08:29:43 29-Jun-2018
speed-20180629-081850.jpg@ - 08:18:50 29-Jun-2018
speed-20180629-081601.jpg@ - 08:16:02 29-Jun-2018
speed-20180629-081020.jpg@ - 08:10:21 29-Jun-2018
speed-20180629-080330.jpg@ - 08:03:30 29-Jun-2018
speed-20180629-075829.jpg@ - 07:58:29 29-Jun-2018
speed-20180629-075656.jpg@ - 07:56:57 29-Jun-2018
speed-20180629-075132.jpg@ - 07:51:33 29-Jun-2018
speed-20180629-074543.jpg@ - 07:45:43 29-Jun-2018
speed-20180629-074538.jpg@ - 07:45:38 29-Jun-2018
speed-20180629-074451.jpg@ - 07:44:52 29-Jun-2018
speed-20180629-074317.jpg@ - 07:43:18 29-Jun-2018
speed-20180629-074243.jpg@ - 07:42:43 29-Jun-2018
speed-20180629-073855.jpg@ - 07:38:56 29-Jun-2018
speed-20180629-073511.jpg@ - 07:35:12 29-Jun-2018
speed-20180629-073205.jpg@ - 07:32:06 29-Jun-2018
speed-20180629-072253.jpg@ - 07:22:54 29-Jun-2018
speed-20180629-071922.jpg@ - 07:19:22 29-Jun-2018
speed-20180629-071818.jpg@ - 07:18:19 29-Jun-2018
speed-20180629-071523.jpg@ - 07:15:24 29-Jun-2018
speed-20180629-070550.jpg@ - 07:05:51 29-Jun-2018
speed-20180629-070519.jpg@ - 07:05:19 29-Jun-2018
speed-20180629-070221.jpg@ - 07:02:22 29-Jun-2018
speed-20180629-065838.jpg@ - 06:58:39 29-Jun-2018
speed-20180629-064626.jpg@ - 06:46:27 29-Jun-2018
speed-20180629-063841.jpg@ - 06:38:42 29-Jun-2018
speed-20180629-061616.jpg@ - 06:16:17 29-Jun-2018
speed-20180629-060715.jpg@ - 06:07:15 29-Jun-2018
speed-20180629-011232.jpg@ - 01:12:32 29-Jun-2018
speed-20180628-235148.jpg@ - 23:51:48 28-Jun-2018
speed-20180628-234310.jpg@ - 23:43:10 28-Jun-2018
speed-20180628-231818.jpg@ - 23:18:19 28-Jun-2018
speed-20180628-230703.jpg@ - 23:07:04 28-Jun-2018
speed-20180628-224622.jpg@ - 22:46:22 28-Jun-2018
speed-20180628-223536.jpg@ - 22:35:36 28-Jun-2018

Web Root is [media]

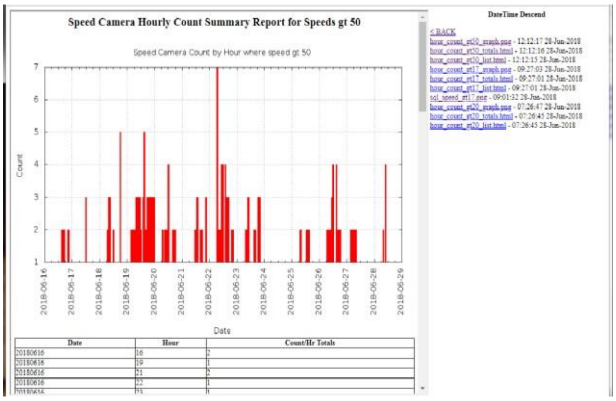
SPEED-CAMERA Media

Listing All 100 Files in [/recent/]

Fig. 6: View speed-cam data and trends from web browser

D. Results:

This bar graph shows the data collected after implementation of our project, showing the number of vehicles passing hour by hour in a day. It shows the testing of the algorithm on different images processed for tracking and counting vehicle extracted from the video made by our webcam place on the road. This will incorporate us the calculation of the speed of vehicles passing along the road, with their time and date of passing.



Date	Hour	Count/Hr Totals
20180616	16	2
20180616	19	1
20180616	21	2
20180616	22	1
20180616	23	1

DateTime Descend

< BACK
hour_count_gt50_graph.png - 12:12:17 28-Jun-2018
hour_count_gt50_totals.html - 12:12:16 28-Jun-2018
hour_count_gt50_list.html - 12:12:15 28-Jun-2018
hour_count_gt17_graph.png - 09:27:03 28-Jun-2018
hour_count_gt17_totals.html - 09:27:01 28-Jun-2018
hour_count_gt17_list.html - 09:27:01 28-Jun-2018
sql_speed_gt17.png - 09:01:32 28-Jun-2018
hour_count_gt20_graph.png - 07:26:47 28-Jun-2018
hour_count_gt20_totals.html - 07:26:45 28-Jun-2018
hour_count_gt20_list.html - 07:26:45 28-Jun-2018

Fig. 7 Bar graph of speed testing

V. CONCLUSION

We presented an innovative idea for the implementation of traffic laws and its effective regulation. Never before was this idea of identifying wrong-way vehicles and implication of taxes on it introduced, so this paper intends to revolutionize all obsolete projects that were costly and ineffective. This project provides a comprehensive approach to handle traffic nemesis. Each algorithm in our module has been successfully designed and tested, and results are also formulated in tabular form.

REFERENCES

[1] "Real-time Traffic Light Control and Congestion Avoidance System", by Sabya Sanchi Kanojia International Journal of Engineering Research and Applications, 2012.

[2] Veltech Dr.RR & Dr.SR University. "Smart traffic system using raspberry pi by applying dynamic color changer algorithm," IEEE International Conference on Smart Technologies and Management (ICSTM), India. 2 - 4 August 2017. pp.146-150.

[3] Omid Avatefipour, Froogh Sadry, Student Member, "Traffic Management System Using IoT Technology - A Comparative Review," IEEE Electrical and Computer Engineering Department, University of Michigan – Dearborn.

[4] Kanungo, A., Sharma, A., & Singla, C. (2014, March). Smart traffic lights switching and traffic density calculation using video processing. In Engineering and computational sciences (RAECS), 2014 recent advances in (pp. 1-6). IEEE.

[5] Chong, H. F., & Ng, D. W. K. (2016, December). Development of IoT device for traffic management system. In Research and Development (SCOReD), 2016 IEEE Student Conference on (pp. 16). IEEE.

[6] Salama, A.S, Saleh, B.K, Eassa, M.M, "Intelligent cross road traffic management system (ICRTMS)," Computer Technology and Development (ICCTD)". IEEE International Conference 2010.

[7] SkRiyazhussain, C.R.S. Lokesh, P.Vamsikrishna, GoliRohan, et.al., "Raspberry Pi Controlled Traffic Density Monitoring System", IEEE WiSPNET 2016 conference. [8] N. J. Uke and R. C. Thool, "Moving Vehicle Detection for Measuring Traffic Count Using OpenCV," Journal of Automation and Control Engineering, vol. 1, no. 4, pp. 349–352, 2013.

[9] Stefan Badura, Anton Lieskovsky "Intelligent Traffic System: Cooperation of MANET and Image Processing," Integrated Intelligent Computing (ICIIC), 2010 International Conference.

[10] ARanganath, T SreeValli "Intellgent Mangement System For Density Based Control, Stolen Vehicle And Auto Clerance", Aug 2015.

[11] Implementing "Intelligent Traffic Control System Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection" by Rajeshwari Sundar, Santhoshs Hebbar, and Varaprasad Golla, IEEE SENSORS JOURNAL, VOL. 15, NO. 2, FEBRUARY 2015.

[12] M. Piccardi, "Background subtraction techniques: a review," IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 30993104,2004.

[13] C.Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," Proc. IEEE CVPR 1999, pp. 246-252, June 1999.

[14] Karric Kwong, Robert Kavaler, Ram Rajagopal, and Pravin Varaiya "Real-Time Measurement of Link Vehicle Count and Travel Time in a Road Network" IEEE transactions on intelligent transportation system, December 2010.

[15] L. Li, W. Huang, T. Y. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," Proc. of the Eleventh ACM international conference on Multimedia, Berkeley, CA, USA, pp. 2-10, November 2003.

[16] Daigavane, P. M., and P. R. Bajaj, "Real Time Vehicle Detection and Counting Method for Unsupervised Traffic Video on Highways," International Joumal of Computer Science and Network Security IJCSNS, vol.l 0, no.8, pp. 112-117, August 2010.

[17] G. Salvi, "An Automated Vehicle Counting System Based on Blob Analysis for Traffic Surveillance,".

[18] Xi Yu, Fuquan Sun, Xu Cheng, et.al.," Intelligent Urba n Traffic Management System Based on Cloud Computing and Internet of Things", Dalian High-tech Zone Innovation of Science and Technology Plan: 20113006 .