Scientific Journal of Impact Factor (SJIF): 5.71

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 5, Issue 03, March -2018

Vehicle Speed Detection and Monitoring Using Feature Extraction

[1]Mrs.M.Maheshwari, *Asst.Professor(Grade 1)*
[2]V.Ramani, [3]A.Ramya, [4]R.Nandhini

Department of Computer Science Engineering
Panimalar Engineering College Chennai-600123, Tamil Nadu , India

Abstract:- *This paper presents about expanding number of mixture and electric vehicles, a developing number of auto passerby mishap turn into an issue on the grounds that these vehicles are for all intents and purposes quiet at low speeds where tire commotion isn't huge. To adapt to this issue, a continuous moving toward vehicle discovery and the speed estimation is viewed as that the vehicle speed is checked and furthermore the vehicle is caught, pictures put away to a predetermined information way and the video stream can be screen through the high ways, private spots and so forth. The speed of the vehicle additionally changes every once in a while at some specific point the vehicle tends to cross the each segment of the activity flags so at place of movement path this procedure can be utilized to dodge such speeding and rash driving by hinting the speed of the vehicle through website page through neighbourhood.*

Keywords---*Speed detection;Image processing;Haar cascade;Motion detection.*

I. INTRODUCTION

Activity observing and administration framework is as yet developing with developing urbanization.This paper goes for speed recognition or estimation of vehicles from picture stream.Nowadays the most widely recognized approach to quantify speed is by utilizing the radar gear, thusly it is vital to propose some other ideas like estimating vehicle speed from picture stream.Instead of equipment reliance that is issue with radar framework we can utilize picture processing,which is basically in light of programming usage.

There are a few papers where numerous vehicles speed recognition are used.Most essential piece of speed estimation framework is question discovery and following are proposed.The change that is accomplished in PC vision and machine learning, we can discover use of these strategies in numerous different regions. One of them is activity checking and administration framework, where the significance is as yet developing with developing urbanization. This paper goes for speed discovery or estimation of vehicles from video stream. In exhibit, the most widely recognized approach to gauge speed is by utilizing the radar gear, along these lines it is critical to propose some other ideas like estimating vehicle speed from video stream. Rather than equipment reliance that is issue with radar frameworks we can utilize picture preparing, which is mostly in view of programming execution.

The speed location is physically utilized by the people by utilizing the speed sensor and furthermore the estimation of the sensor which is fitted to the steady peruser and the information is observed through a human association so the speed can be checked and the vehicle is effectively get out by the experts in view of this framework the static speed peruser without the network of the information to the webpage.The picture can't be taken for the up and coming preparing and furthermore the framework can get one speed discovery of the vehicle at one casing for each time.

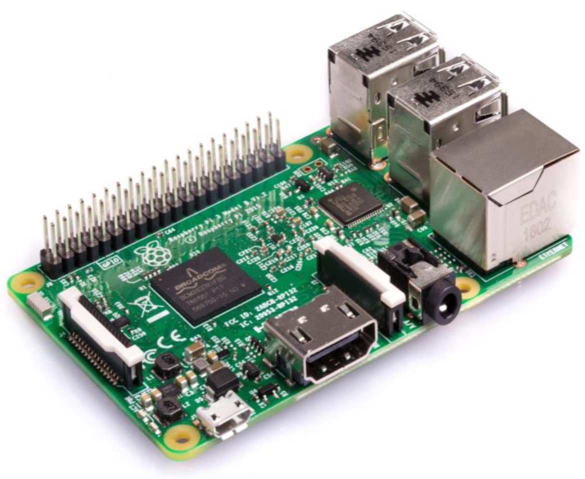
II. PROPOSED SYSTEM

Vehicle speed recognition framework out and about is very mind boggling and needs a high particular of the PC. Consequently the inserted innovation is quickly creating; it is conceivable to utilize Raspberry Pi as the preparing board to gauge and gauge the vehicle speed. Despite the fact that the picture preparing required a considerable measure of assets, the Open CV library is another other option to deal with complex picture handling. For the future work, the threading innovation, which ready to deal with I/O (Input yield) substantial assignment can be utilized. The information which is caught the pictures is spared in the specific chose way and this way can be called and the pictures can be shown through by nearby website page and furthermore through picture records in the envelope.

1.Raspberry pi

Raspberry Pi is a progression of little single-board PCs created in the United Kingdom by the Raspberry Pi Foundation to advance the educating of essential software engineering in schools and in creating nations. The first model ended up much more prominent than expected, offering outside its objective market for utilizations, for example, mechanical autonomy. It does exclude peripherals(such as keyboards,mice and cases).However,some extras have been incorporated into a few official and informal groups.

As indicated by the Raspberry Pi Foundation, more than 5 million Raspberry P is were sold by February 2015, making it the top of the line British PC. By November 2016 they had sold 11 million units, and 12.5m by March 2017, making it the third smash hit "broadly useful computer".[11] In July 2017, deals came to almost 15 million.



1.1 Hardware

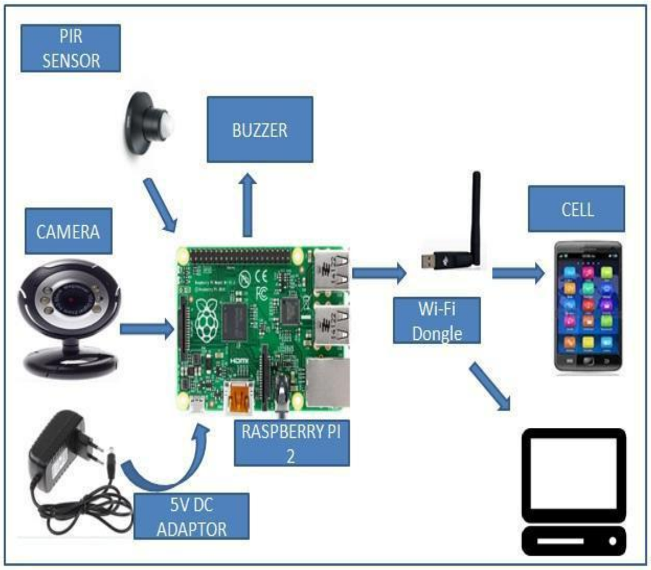
The Raspberry Pi equipment has advanced through a few forms that element varieties in memory limit and fringe gadget support.This Models A, B, A+, and B+. Show An, A+ and the Pi Zero do not have the Ethernet and USB center point segments. The Ethernet connector is inside associated with an extra USB port. In Model An, A+, and the Pi Zero, the USB port is associated straightforwardly to the framework on a chip (SoC). On the Pi 1 Model B+ and later models the USB/Ethernet chip contains a five-point USB center, of which four ports are accessible, while the Pi 1 Model B just gives two. On the Pi Zero, the USB port is additionally associated straightforwardly to the SoC, however it utilizes a miniaturized scale USB (OTG) port.

1.2 RAM

On the more seasoned beta Model B loads up, 128 MB was assigned as a matter of course to the GPU, leaving 128 MB for the CPU. On the initial 256 MB discharge Model B (and Model A), three unique parts were conceivable. The default split was 192 MB (RAM for CPU), which ought to be adequate for independent 1080p video unraveling, or for straightforward 3D, however most likely not for both together. 224 MB was for Linux just, with just a 1080p edge cradle, and was probably going to fizzle for any video or 3D. 128 MB was for substantial 3D, perhaps at the same time with video interpreting (e.g. XBMC). Nearly the Nokia 701 utilizations 128 MB for the Broadcom Video Core IV.

For the later Model B with 512 MB RAM at first there were new standard memory split documents discharged(arm256_start.elf, arm384_start.elf, arm496_start.elf) for 256 MB, 384 MB and 496 MB CPU RAM (and 256 MB, 128 MB and 16 MB video RAM). In any case, a week or so later the RPF discharged another rendition of start.elf that could read another passage in config.txt (gpu mem=xx) and could progressively dole out a measure of RAM (from 16 to 256 MB in 8 MB ventures) to the GPU, so the more established strategy for memory parts wound up old, and a solitary start.elf worked the same for 256 and 512 MB Raspberry Pi.

1.3 METHODOLOGY



The methodology of this project design can be divided into two sections, hardware and software implementation.It is advantageous as it offers reliability and privacy on both sides. It is authenticated and encrypted on the receiver side, hence it offers only the person concerned to view the details .Necessary action can be taken in short span of time in the case of emergency conditions such as elderly person falling sick, military areas, smart homes, offices, industries etc. ,Future work is to locate the number of persons located exactly on that area and their position so that accurate information can be obtained on the receiver side.

2.Working with Raspberry Pi Camera Board

This illustration demonstrates to you best practices to catch and process pictures from Raspberry Pi® Camera Board module utilizing the MATLAB® Support Package for Raspberry Pi Hardware.

2.1 Introduction

The Raspberry Pi Camera Board is a hand crafted add-on module for Raspberry Pi equipment. It joins to Raspberry Pi equipment through a custom CSI interface. The sensor has 5 megapixel local determination in still catch mode. In video mode it bolsters catch resolutions up to 1080p at 30 outlines for every second. The camera module is light weight and little settling on it a perfect decision for portable projects.In this case you will figure out how to make a camera board question associate with the Raspberry Pi Camera Board, catch pictures from the camera and process them in MATLAB.

2.2 Prerequisites

We recommend completing Getting Started with MATLAB Support Package for Raspberry Pi Hardware example.

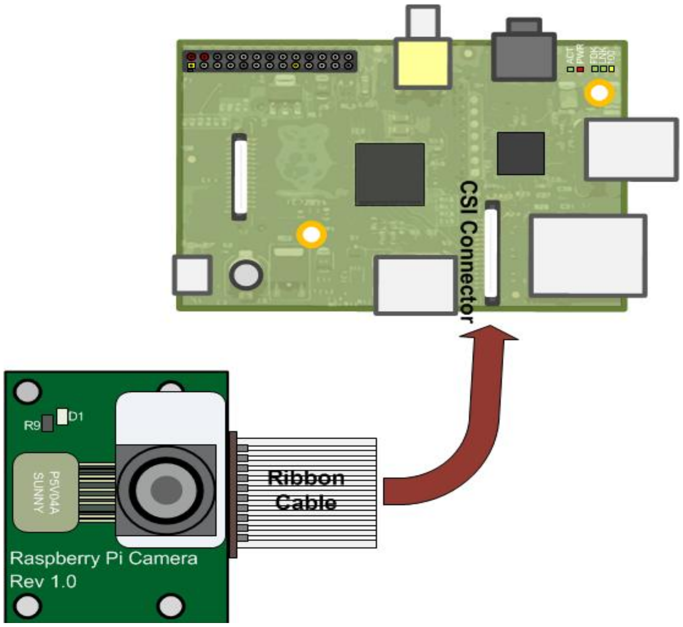
2.3 Required Hardware

To run this example you will need the following hardware:

- Raspberry Pi hardware
- A power supply with at least 1A output
- A Camera Board

2.4 Connect Camera Board

The camera board attaches to the Raspberry Pi via a ribbon cable. One end of the ribbon cable goes to the camera PCB and the other end attached to Raspberry Pi hardware itself. You need to get the ribbon cable connections the right way, or the camera will not work. On the camera PCB, the blue backing on the cable should be facing away from the PCB, and on the Raspberry Pi hardware it should be facing towards the Ethernet connection.



3. Create a Camera Board object

Create a camera board object by executing the following command on the MATLAB prompt.

```
clear rpi
rpi = raspi();
cam = cameraboard(rpi,'Resolution','640x480');
```

The cam is a handle to a cameraboard object. Let's display the images captured from Raspberry Pi Camera Board in MATLAB.

```
for i = 1:100
   img = snapshot(cam);
   image(img);
   drawnow;
end
```

4. Inspect object properties

The MATLAB command line interface for Camera Board has a number of properties that expose the features of the Camera. To view the properties of the cameraboard class type the following on the MATLAB prompt.

Cam

Using the properties of the cameraboard object, you can flip the images horizontally or vertically, change image quality parameters such as brightness, contrast, saturation and sharpness and access advanced camera features such as image stabilization and image effects. The Resolution and FrameRate properties cannot be changedafter instantiation. If you want to change these properties, clear the cameraboard object from MATLAB workspace and create a new object by specifying the new Resolution and FrameRate parameters.

clear cam

```
 cam=cameraboard(rpi,'Resolution','320x240','FrameRate',30);
```

Other properties of the cameraboard object can be changed at any time. But it takes 5 frames for the new setting to take effect. Let's try flipping the image horizontally.

```
figure(1);
for i = 1:5
img = snapshot(cam);
end
image(img);
cam.HorizontalFlip = true;
for i = 1:5
   img = snapshot(cam);
end
figure(2);
image(img);
```

Image effects

Let's try a simple image inversion algorithm on the images captured from Raspberry Pi Camera Board.

```
figure(1);
for i = 1:100
   img = snapshot(cam);
   img = 255 - img;
   image(img);
   drawnow;
end
```

The image inversion creates a color negative effect. The Raspberry Pi Camera Board itself can invert the images by setting ImageEffect property to 'negative'.

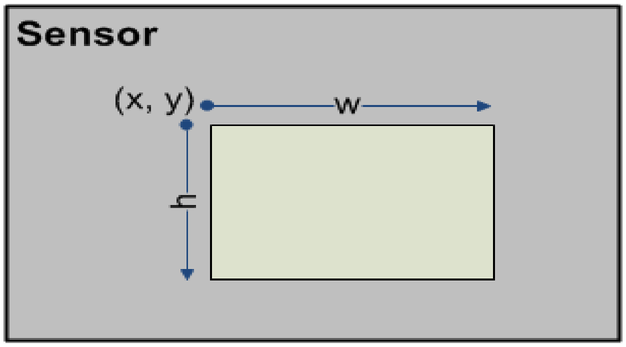
```
figure(1);
cam.ImageEffect = 'negative';
for i = 1:100
   img = snapshot(cam);
 image(img);
   drawnow;
end
```

Here is a more interesting image effect.

```
figure(1);
cam.ImageEffect = 'sketch';
for i = 1:100
   img = snapshot(cam);
   image(img);
   drawnow;
end
```

Digital zoom

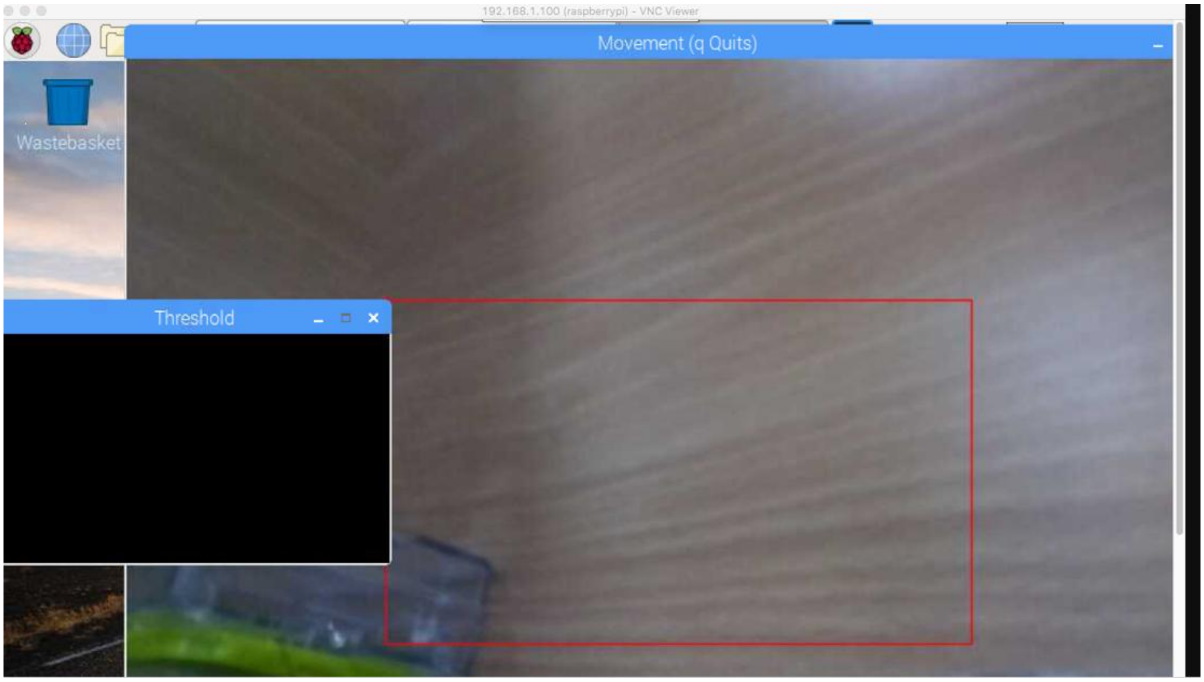
Raspberry Pi Camera Board allows a region of the sensor to be used as the image capture area. This region, called region of interest (ROI), is specified as a normalized vector [x y w h] where x, y defines the top left corner and w and h specifies the width and height.



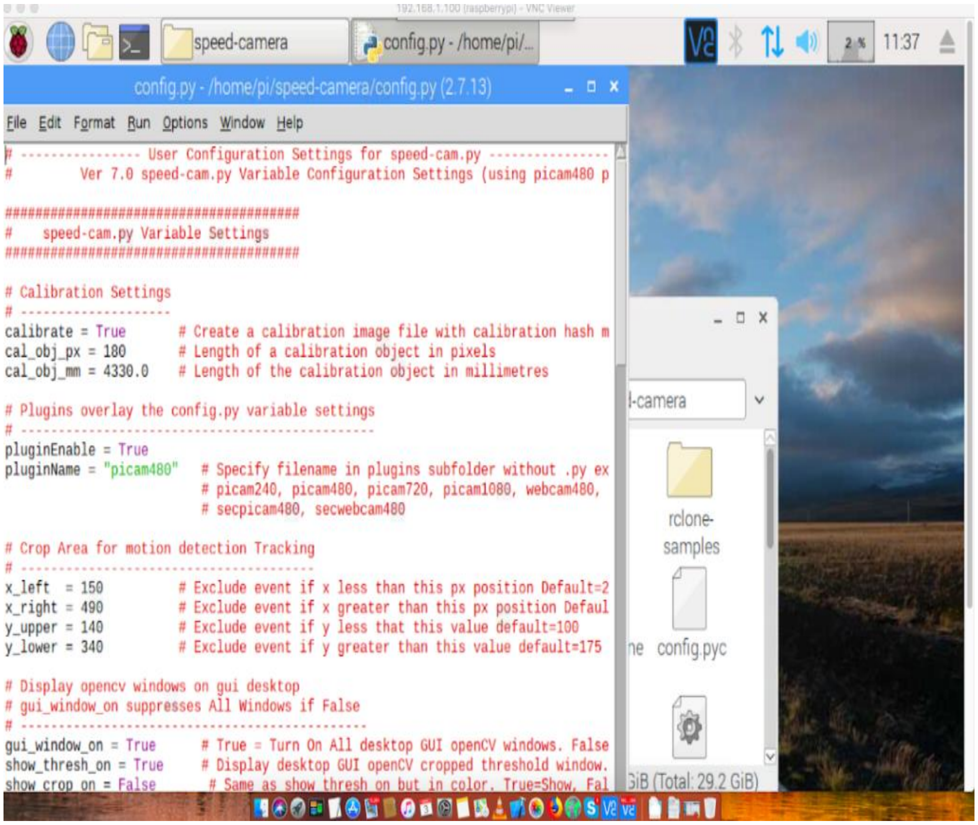
Reducing ROI while holding the output image size constant results in a digital zooming effect. The following MATLAB code varies the x and y parameters of the ROI to zoom into the lower right part of the sensor. The approximate area of the sensor being captured is indicated by a red rectangle.

```
figure(1);
roi = [0 0 1 1];
cam.ROI = [0 0 1 1];
for i = 1:10
   img = snapshot(cam);
end
subplot(211);
image(img);
drawnow;
rect = rectangle('Position',[1 1 320 240]);
rect.EdgeColor = 'red';
for i = 1:200
  img = snapshot(cam);
  if i > 20
    fc = (i - 5)*0.0025;
    roi(1:2) = [fc, fc];
    roi(3:end) = [1-fc, 1-fc];
    cam.ROI = roi;
    subplot(211);
    rect.Position = roi.*[320 240 320 240];
    drawnow;
    subplot(212);
    image(img);
    drawnow;
  end
end
```

III.Snap Shots

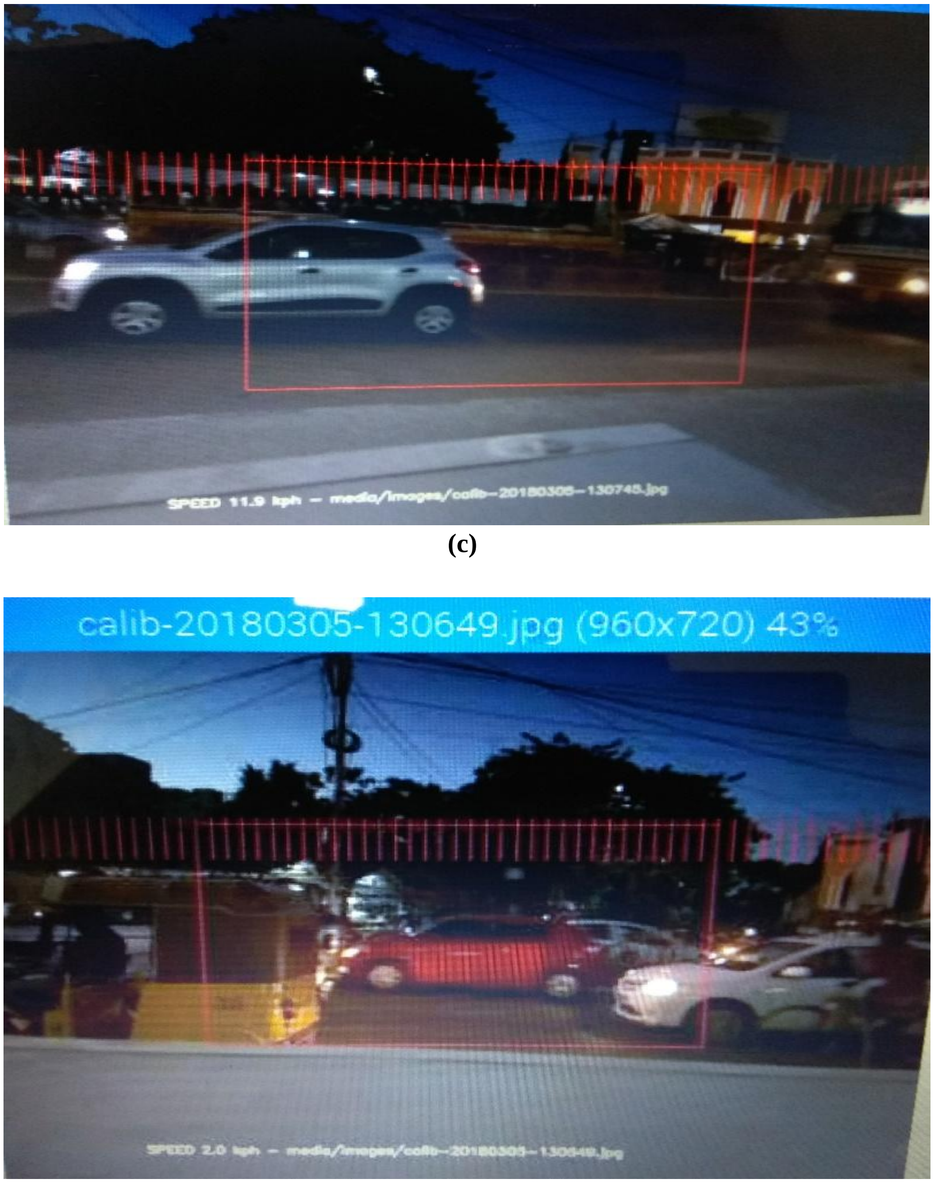


(a)


192.168.1.100 (raspberrypi) - VNC Viewer
speed-camera
config.py - /home/pi/...
11:37
config.py - /home/pi/speed-camera/config.py (2.7.13)
File Edit Format Run Options Window Help
rclone-samples
config.pyc
GiB (Total: 29.2 GiB)

```
# ---------------- User Configuration Settings for speed-cam.py ----------------
#         Ver 7.0 speed-cam.py Variable Configuration Settings (using picam480 p

#######################################
#    speed-cam.py Variable Settings
#######################################

# Calibration Settings
# -------------------
calibrate = True        # Create a calibration image file with calibration hash m
cal_obj_px = 180        # Length of a calibration object in pixels
cal_obj_mm = 4330.0     # Length of the calibration object in millimetres

# Plugins overlay the config.py variable settings
# -----------------------------------------------
pluginEnable = True
pluginName = "picam480"    # Specify filename in plugins subfolder without .py ex
                           # picam240, picam480, picam720, picam1080, webcam480,
                           # secpicam480, secwebcam480

# Crop Area for motion detection Tracking
# -------------------------------------
x_left  = 150          # Exclude event if x less than this px position Default=2
x_right = 490          # Exclude event if x greater than this px position Defaul
y_upper = 140          # Exclude event if y less that this value default=100
y_lower = 340          # Exclude event if y greater than this value default=175

# Display opencv windows on gui desktop
# gui_window_on suppresses All Windows if False
# -----------------------------------------
gui_window_on = True       # True = Turn On All desktop GUI openCV windows. False
show_thresh_on = True      # Display desktop GUI openCV cropped threshold window.
show_crop_on = False        # Same as show thresh on but in color. True=Show, Fal
```

(b)


SPEED 11.9 kph – media/images/calib-20180305-130745.jpg
calib-20180305-130649.jpg (960x720) 43%
SPEED 2.0 kph – media/images/calib-20180305-130649.jpg

(c)

(d)

Figure 1: Speeding event detection under difference scenarios.

(a) Primal stage (b) coding (c) High speed vehicle with 11kmph (d) Images of the captured by the system.

IV. Conclusion

This paper introduces a technique for vehicle identification and speed computation utilizing picture processing.Increasing consideration has been fixated on building up these advances on vehicles to use them for enhancing driving conditions, especially on well being and activity effici propose a straightforward and minimal effort raspberry pi for vehicle detection,speed estimation and the separation between the camera and the moving objects.The created stage is versatile and taken a toll effective.We examine the difficulties related with running such system,and propose arrangements in view of machine learning ways to deal with play out the estimation and the classification.Experimental comes about demonstrates a 99 percent precision in the vehicle discovery utilizing haar cascade algorithm.

V. References

1. T.M. Hussain, A.M. Baig, T.N. Saadawi, S.A. Ahmed, “Infrared pyroelectric sensor for detection of vehicular traffic using digital signal processing techniques”, IEEE Transactions on Vehicular Technology, 44(3),pp. 683-689, 1995.
2. J. Gajda, R. Sroka, M. Stencel, A. Wajda, “A vehicle classification based on inductive loop detectors”, Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE (Volume:1).pp. 460-464, 2001.
3. K.T. Song, C.H. Chen, C.H. C. Huang, “Design and experimental study of an ultrasonic sensor system for lateral collision avoidance at low speeds”, Intelligent Vehicles Symposium, pp. 647-652, 2004
4. Fazenda, BM, Atmoko, H, Gu, F, Guan, L and Ball, A 2009, Acoustic based safety emergency vehicle detection for intelligent transport systems , in: ICROS-SICE International Joint Conference 2009, 18th - 21st August 2009, Fukuoka, Japan.
5. S. P. Lohmeier, R. Rajaraman, V. C. Ramasami. Development of an ultra-wideband radar system for vehicle detection at railway crossings. IEEE Conference on Ultra Wideband Systems and Technologies.207 - 211.2002
6. J.F. Alves, G.R. Cacnio, D.R. Stevens, “Video image processor and method for detecting vehicles”, US5535314 A, 1996.
7. Z.W. He, J.L. Liu, P.H. Li, “New method of background update for video-based vehicle detection”, The 7th International IEEE Conference on Intelligent Transportation Systems, pp. 580-584, 2004
8. E. Mosabbeb, M. Sadeghi and M. Fathy, "A new approach for vehicle Detection in Congested Traffic Scenes Based on Strong Shadow Segmentation", Springer, part II, LNCS 4842, pp. 427-436, 2007
9. J. Hsieh, L. Chen and D. Chen, "Symmetrical Surf and its Applications to Vehicle Detection and Vehicle Make and Model Recognition", IEEE Trans, Intell. Transp. Syst., 15(1), pp. 6-20, 2014
10. Y.B. Rao, “Automatic vehicle recognition in multiple cameras for video surveillance”, The Visual Computer, 31(3): pp. 271-280,2015