A

PROJECT REPORT

on

Intelligent Traffic Light Control System

Submitted in partial fulfilment for the Award of Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

Thalla Pavan (160117737045)

&

V Pradyumna Reddy (160117737046)

Under the guidance of

S Rakesh ,
Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)
(Affiliated to Osmania University; Accredited by NBA (AICTE) and NAAC (UGC), ISO Certified 9001:2015), Kokapet (V), GANDIPET(M), HYDERABAD – 500 075

Website: www.cbit.ac.in

2020-2021

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (A)
Kokapet (Village), Gandipet, Hyderabad, Telangana-500075. www.cbit.ac.in
ISO Certified 9001:2015

COMMITTED TO RESEARCH, INNOVATION AND EDUCATION
42 years

CERTIFICATE

This is to certify that the project seminar report entitled "**Intelligent Traffic Light Control System**" submitted to **Chaitanya Bharathi Institute of Technology**, in partial fulfilment of the requirements for the award of degree of **B.E (Information Technology)** during the academic year 2020-21 is a record of original work carried out by **Thalla Pavan(160117737045)** and **V Pradyumna Reddy(160117737046)** during the period of study in the Dept. of IT, CBIT, Hyderabad.

Project Guide	Head of the Department
Mr S Rakesh	**Dr. K. Radhika.**
Assistant Professor, IT Dept.	Professor, IT Dept
CBIT, Hyderabad.	CBIT, Hyderabad.

DECLARATION

We declare that the project report entitled "**INTELLIGENT TRAFFIC LIGHT CONTROL SYSTEM**" is being submitted by us in the Department of Information Technology, Chaitanya Bharathi Institute of Technology (A), Osmania University. This is a record of bonafide work carried out by us under the guidance and supervision of Mr**. S.Rakesh**, Assistant Professor, Dept. of IT, C.B.I.T. No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The reports are based on the project work done entirely by us and not copied from any other source.

Thalla Pavan(160117737045) V Pradyumna Reddy(160117737046)

ACKNOWLEDGEMENTS

It is our privilege to acknowledge with a deep sense of gratitude and devotion for keen personal interest and invaluable guidance rendered by our Project Guide **S.Rakesh**, Assistant Professor, Department of Information, Chaitanya Bharathi Institute ofTechnology.

We take the opportunity to express my thanks to **Dr. K. Radhika**, Professor & Head of IT Department, CBIT for her valuable suggestions and moral support.

We are grateful to our Principal Dr. G.Partha Saradhi Varma, Chaitanya Bharathi Institute of Technology, for his cooperation and encouragement.

Finally, we also thank all the staff members, faculty of Dept. of IT, CBIT, and my friends, who with their valuable suggestions and support, directly or indirectly helped me in completing this project work.

ABSTRACT

In today's life we are facing many problems one of which is traffic congestion becoming more serious day after day. The major reason leading to traffic jam is the high number of vehicle which was caused by the population and the development of economy. Currently, traffic lights are automatically getting ON and OFF depending on the timer value changes. In this, time is being wasted by a green light on an empty road which can be utilized on some other lane. These frequent traffic problems like traffic jams have led to the rise of the need for an efficient traffic management method. Possible Solution is dynamic allocation of time to different lanes on signals.

Our project intelligent traffic control system based on image processing in Python. In this, traffic density of lanes is calculated using image processing which is done of images of lanes that are captured using camera. After image acquisition, image undergoes various image pre-processing, image enhancement and calculating vehicle density on the roads. According to this matching percentage, decision is to be taken for the dynamic allocation of time to different lanes.

LIST OF FIGURES

TABLE OF CONTENTS

1.INTRODUCTION

1.1 Overview

Fast transportation systems and rapid transit systems are nerves of economic developments for any nation. Mismanagement and traffic congestion results in long waiting times, loss of fuel and money. It is therefore utmost necessary to have a fast, economical and efficient traffic control system for national development. The monitoring and control of city traffic is becoming a major problem in many countries. With the ever-increasing number of vehicles on the road, the Traffic Monitoring Authority has to find new methods of overcoming such a problem. One way to improve traffic flow and Safety of the current transportation system is to apply automation and Intelligent control methods. Traffic congestion may result due to heavy traffic at a junction. To avoid congestion there are so many traffic management techniques available. But no technique is perfect by itself as the real time situations are generally continuously changing and the system has to adapt itself to change in the continuously changing circumstances. We have made an attempt to provide some traffic management strategy which is self-changing in nature, so as to fit into continuously changing real time traffic scenarios.

1.2 Applications

The project we developed can be used in the place of current traffic light system being used. This Reduce traffic congestion, makes the road safer by reducing the chance of accidents It decrease pollution by reducing congestion and it saves time of the vehicles waiting on signal. It reduces the workload of traffic police and RTO officers. Images are stored and will be available for other purposes like crime detection.

1.3 Problem Description

The problems of typical conventional traffic light Controller are mentioned below:

Heavy Traffic Jams:

With increasing number of vehicles on road, heavy traffic congestion has substantially increased in major cities. This happened usually at the main junctions commonly in the morning, before office hour and in the evening, after office hours. The main effect of this matter is increased time wasting of the people on the road. The solution for this problem is by developing the program which different setting delays for different junctions. The delay for junctions that have high volume of traffic should be setting longer than the delay for the junction that has low of traffic.

No traffic, but still need to wait:

At certain junctions, sometimes even if there is no traffic, people must wait. Because the traffic light remains red for the preset time period, the road users should wait until the light turn to green. If they run the red light, they have to pay fine. The solution of this problem is by developing a system which detects traffic flow on each road and set timings of signals accordingly. Moreover, synchronization of traffic signals in adjacent junctions is also necessary .

1.4 Aim of the project

Objective of proposed system is to improve efficiency of existing automatic traffic signalling system. The system will be image processing based adaptive signal controlling. The timing will be calculated, each time change automatically depending upon the traffic density on road. Proposed system will be functioning based on the technique known as image processing where the image of lane will be processed to estimate traffic density which further will be used to calculate the required time duration for controlling of signal lights based on comparison with experimental results. Maximum and minimum time limit will be maintained to prevent over waiting of vehicle in queue of other lanes which would be found out experimentally.

2. LITERATURE SURVEY

2.1 Intelligent Traffic Light Controller Using IR Sensors for Vehicle Detection, Year 2018, IEEE

Author's : Mr. Yogesh Shinde , Miss. Hemlata Powar

Abstract : Mr. Yogesh Shinde , Miss. Hemlata Powar propose a system that estimates the size of traffic in highways by using IR Sensors. In this system IR sensors are present on either sides of the road detect the presence of the vehicles and sends the information to the microcontroller. On the basis of those information, micro-controller will decide the glowing time of green light and red light. It means that the timing of the traffic lights is set according to the density of the vehicles

Methodologies : An IR transmitter and receiver are placed beside the road which detect the vehicles when they come in between.

Advantages : Implementation is cheaper.

Disadvantages : There should be a direct line of sight path between sensors and vehicles . Multiple detections were difficult.

2.2 Intelligent traffic control system using RFID technology, Year 2019, IEEE

Author's : Vijayaraman P, P Jesu Jayarin**.**

Abstract : Vijayaraman P, P Jesu Jayarin propose a system that estimates the size of traffic in highways by using RFID. The RFID reader modules have to be mounted along side the roads. The RFID tags/labels need to be affixed/stuck to the vehicles. The tags/labels will be fixed to the license plates of the vehicles aiming for the best reachable probability of the label getting read by the reader. Then the RFID readers determine the count of vehicles by reading the RFID tags present on the license plates and appropriate timing for green light is given.

Methodologies :RFID tags are attached to every vehicle and RFID readers are placed beside road which reads the RFID tag and determine density.

Advantages : Can scan multiple devices simultaneously. Emergency vehicle detection is faster.

Disadvantages : Unauthorized devices may be able to read and even change data on RFID tags and signals can interfere with each other.

2.3 Smart Traffic Control System Based on Image Processing , Year 2017,IJRASET

Author's : Chirag Thakkar, Rajesh Patil.

Abstract : Chirag Thakkar, Rajesh Patil propose a system that estimates the size of traffic in highways by using Image Processing. The image sequences from a camera are analyzed using various edge detection and object counting methods to obtain the most efficient technique. Subsequently, the number of vehicles at the intersection is evaluated and traffic is efficiently managed. The paper also proposes to implement a real-time emergency vehicle detection system. In case an emergency vehicle is detected, the lane is given priority over the others. The key point of this paper is the technique which is used for edge detection. The authors have given the comparison of various edge detection techniques and conclude that canny edge detection is the best method for edge detection.

Methodologies : Captures images of real time traffic perform image-preprocessing on these images and then uses canny edge detection technique to determine density.

Advantages : Implementation is cheaper.Faster Processing.

Disadvantages : Does not give preference to larger vehicles and calculation of vehicle density is not accurate.

2.4 Intelligent traffic light control system based on traffic environment using deep learning , Year 2020, IARJSET.

Author's : Ananya Bansal, Pramod Goyal,Vaibhav Kashyap.

Abstract : Ananya Bansal, Pramod Goyal,Vaibhav Kashyap propose a system that estimates the size of traffic in highways by using Image Processing. The image sequences from a camera are analyzed using various edge detection and object counting methods to obtain the most efficient technique. Subsequently, the number of vehicles at the intersection is evaluated and traffic is efficiently managed. The paper also proposes to implement a real-time emergency vehicle detection system. In case an emergency vehicle is detected, the lane is given priority over the others. The key point of this paper is the technique which is used for edge detection. The authors have given the comparison of various edge detection techniques and conclude that canny edge detection is the best method for edge detection.

Methodologies : Capture images and uses cnn to determine density by detecting and classify each vehicle.
Advantages : Acheives good accuracy and it can give preference to larger vehicles.
Disadvantages : Accuray will be less in case of higher density roads because it can detect multiple vehicles as single vehicle and also it requires more processing time .

3.SYSTEM REQUIREMENT SPECIFICATION

3.1 Functional Requirements

3.1.1 Image acquisition

In image processing it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed.

3.1.2 Edge Detection

Among the key features of an image i.e. edges, lines, and points. We have used edge in our present work which can be detected from the abrupt change in the gray level. An edge essentially demarcates between two distinctly different regions, which means that an edge is the border between two different regions i.e. the edge of the cars in our case. ¬ Edge detection methods locate the pixels in the image that correspond to the edges of the objects (cars) seen in the image. ¬ The result is a binary image with the detected edge pixels. ¬ Common algorithms used are Sobel Edge Detection Technique, Perwitt Edge Detection, Roberts Edge Detection Technique, Zerocross Threshold Edge Detection Technique and Canny Edge Detection Technique.

In our project we used Canny Edge Detection Technique because of its various advantages over other edge detection techniques.

3.1.3 Canny edge detection

Canny edge detection algorithm developed by John F. Canny in 1986. Usually, in Matlab and OpenCV we use the canny edge detection for many popular tasks in edge detection such as lane detection, sketching, border removal.

The basic steps involved in this algorithm are:

- Noise reduction using Gaussian filter
- Gradient calculation along the horizontal and vertical axis
- Non-Maximum suppression of false edges
- Double thresholding for segregating strong and weak edges
- Edge tracking by hysteresis

Noise reduction using Gaussian filter

This step is of utmost importance in the Canny edge detection. It uses a Gaussian filter for the removal of noise from the image, it is because this noise can be assumed as edges due to sudden intensity change by the edge detector. The sum of the elements in the Gaussian kernel is 1, so, the kernel should be normalized before applying as convolution to the image. In this tutorial, we will use a kernel of size 5 X 5 and sigma = 1.4, which will blur the image and remove the noise from it. The equation for Gaussian filter kernel is

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Gradient calculation

When the image is smoothed, the derivatives Ix and Iy are calculated w.r.t x and y axis. It can be implemented by using the Sobel-Feldman kernels convolution with image as given:

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Sobel Kernels

after applying these kernel we can use the gradient magnitudes and the angle to further process this step. The magnitude and angle can be calculated as

$$|G| = \sqrt{I_x^2 + I_y^2},$$
$$\theta(x, y) = arctan\left(\frac{I_y}{I_x}\right)$$

Gradient magnitude and angle

Non-Maximum Suppression

This step aims at reducing the duplicate merging pixels along the edges to make them uneven. For each pixel find two neighbor's in the positive and negative gradient directions, supposing that each neighbor occupies the angle of pi /4, and 0 is the direction straight to the right. If the magnitude of the current pixel is greater than the magnitude of the neighbors, nothing changes, otherwise, the magnitude of the current pixel is set to zero.

Double Thresholding

The gradient magnitudes are compared with two specified threshold values, the first one is lower than the second. The gradients that are smaller than the low threshold value are suppressed, the gradients higher than the high threshold value are marked as strong ones and the corresponding pixels are included in the final edge map. All the rest gradients are marked as weak ones and pixels corresponding to these gradients are considered in the next step.

Edge Tracking using Hysteresis

Since a weak edge pixel caused by true edges will be connected to a strong edge pixel, pixel W with weak gradient is marked as edge and included in the final edge map if and only if it is involved in the same connected component as some pixel S with strong gradient. In other words, there should be a chain of neighbor weak pixels connecting W and S (the neighbors are 8 pixels around the considered one). We will make up and implement an algorithm that finds all the connected components of the gradient map considering each pixel only once. After that, you can decide which pixels will be included in the final edge map.

3.1.4 Background subtraction

Background subtraction is a way of eliminating the background from image. To achieve this we extract the moving foreground from the static background.

Background Subtraction has several use cases in everyday life, It is being used for object segmentation, security enhancement, pedestrian tracking, counting the number of visitors, number of vehicles in traffic etc. It is able to learn and identify the foreground mask.

In OpenCV we have 3 algorithms to do this operation –

BackgroundSubtractorMOG – It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm.

BackgroundSubtractorMOG2-It is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It provides better adaptability to varying scenes due illumination changes etc.

BackgroundSubtractorGMG – This algorithm combines statistical background image estimation and per-pixel Bayesian segmentation.

In this project we have used BackgroundSubtractorMOG2 algorithm for doing background subtraction.

3.1.5 BackgroundSubtractorMOG2

It is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It is based on two papers by Z.Zivkovic, "Improved adaptive Gausian mixture model for background subtraction" in 2004 and "Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction" in 2006. One important feature of this algorithm is that it selects the appropriate number of gaussian distribution for each pixel. (Remember, in last case, we took a K gaussian distributions throughout the algorithm). It provides better adaptibility to varying scenes due illumination changes etc.

As in previous, we have to create a background subtractor object. Here, you have an option of selecting whether shadow to be detected or not. If detectShadows = True (which is so by default), it detects and marks shadows, but decreases the speed. Shadows will be marked in gray color.

3.2 Software Requirements

3.2.1 PyCharm

PyCharm is the most popular IDE used for Python scripting language. This chapter will give you an introduction to PyCharm and explains its features.

PyCharm offers some of the best features to its users and developers in the following aspects −

- Code completion and inspection
- Advanced debugging
- Support for web programming and frameworks such as Django and Flask

Features of PyCharm

Besides, a developer will find PyCharm comfortable to work with because of the features mentioned below −

Code Completion

PyCharm enables smoother code completion whether it is for built in or for an external package.

SQLAlchemy as Debugger

You can set a breakpoint, pause in the debugger and can see the SQL representation of the user expression for SQL Language code.

Git Visualization in Editor

When coding in Python, queries are normal for a developer. You can check the last commit easily in PyCharm as it has the blue sections that can define the difference between the last commit and the current one.

Code Coverage in Editor

You can run .py files outside PyCharm Editor as well marking it as code coverage details elsewhere in the project tree, in the summary section etc.

Package Management

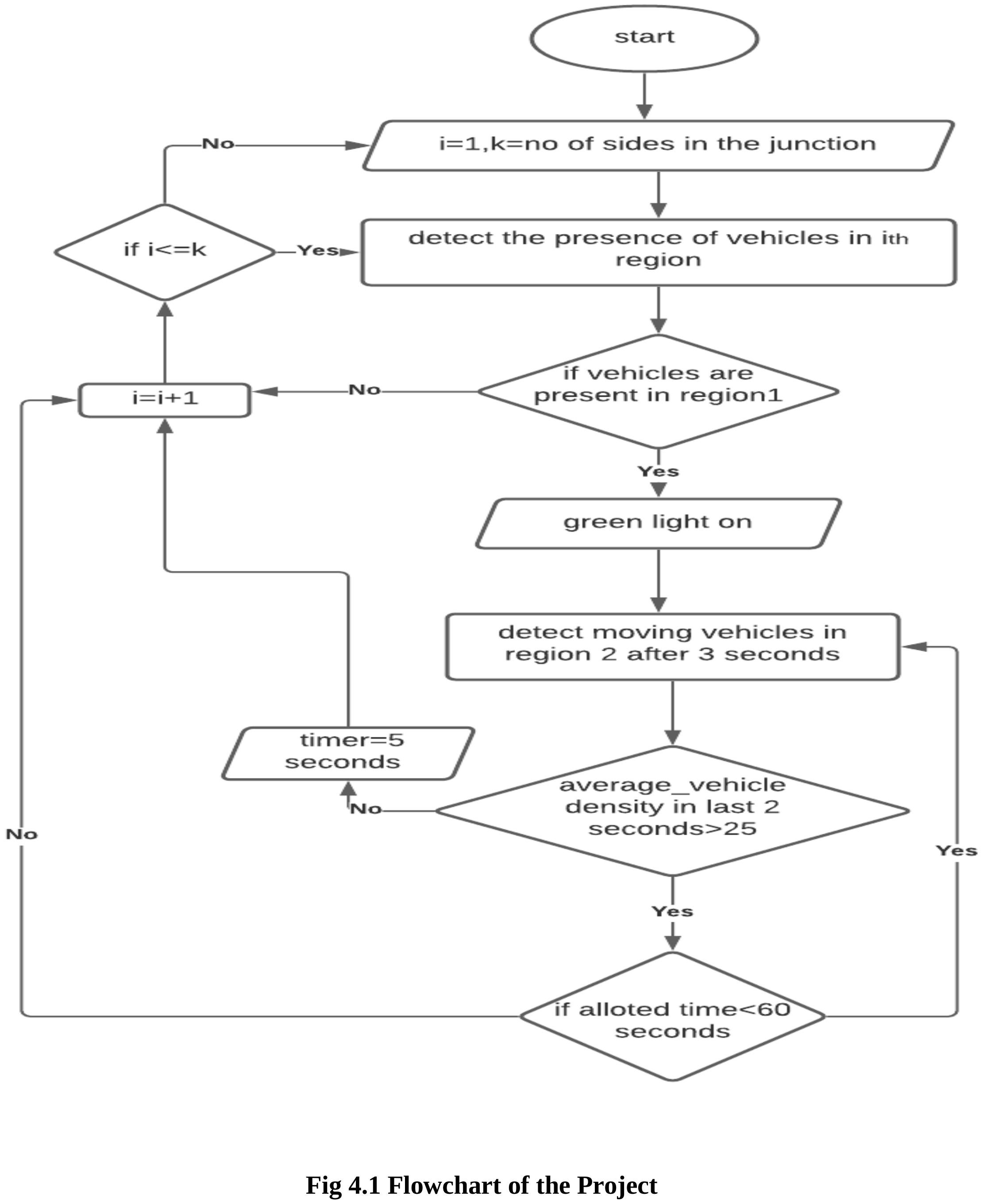
All the installed packages are displayed with proper visual representation. This includes list of installed packages and the ability to search and add new packages.

3.3 Hardware Requirements

- A USB based web camera to capture video of the traffic on road.
- The hardware module consists of an Arduino board used to control LEDs representing the red and green lights.
- A timer module is used to display the remaining time.

4. METHODOLGY

4.1 SYSTEM ARCHITECTURE



start
i=1,k=no of sides in the junction
detect the presence of vehicles in ith region
if i<=k
Yes
No
i=i+1
if vehicles are present in region1
No
Yes
green light on
detect moving vehicles in region 2 after 3 seconds
timer=5 seconds
average_vehicle density in last 2 seconds>25
No
Yes
if alloted time<60 seconds
No
Yes

Fig 4.1 Flowchart of the Project

4.2 PROCEDURE:

Phase1:

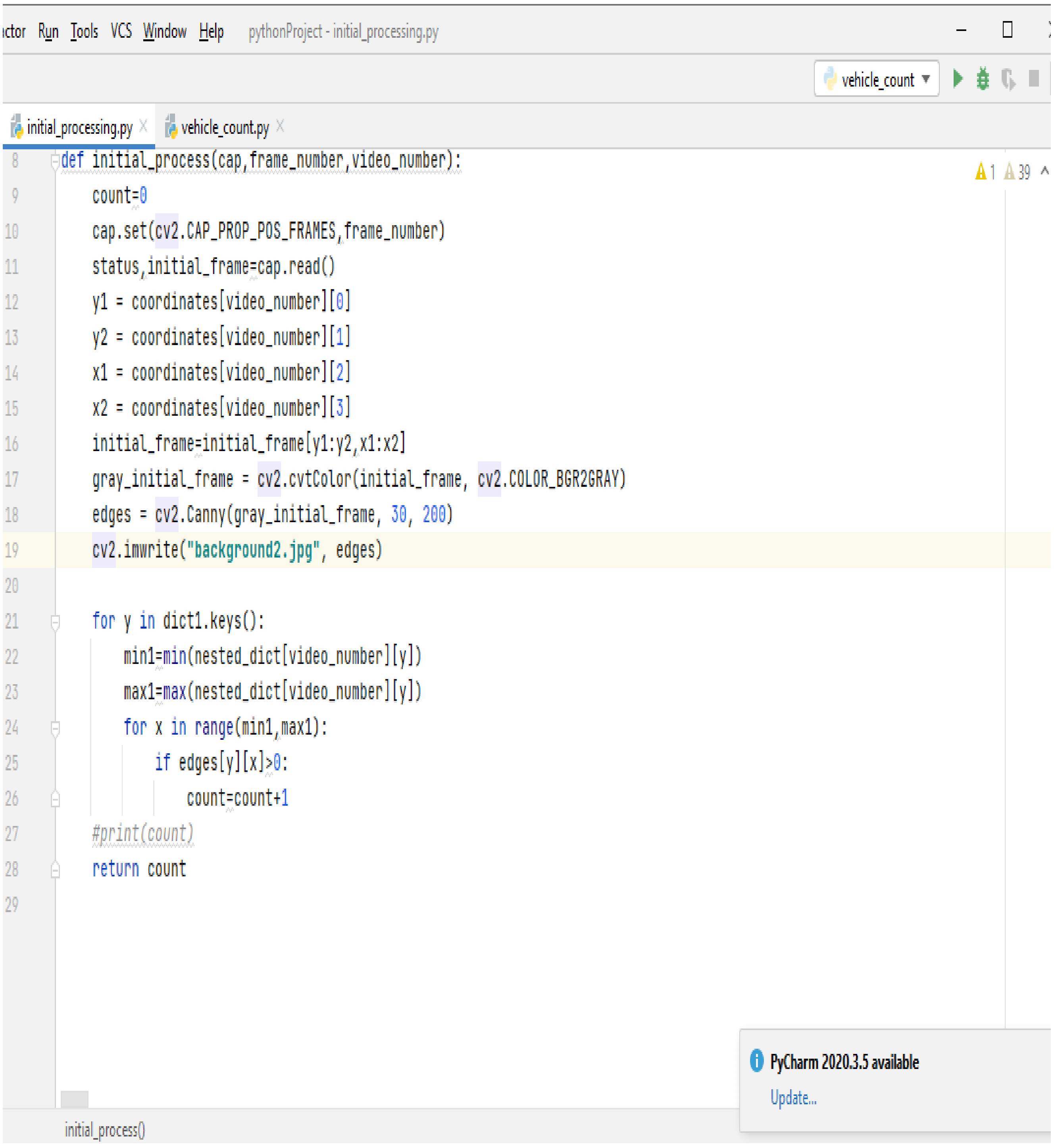
- Initially image acquisition is done with the help of web camera
- Applying Canny Edge Detection in Region 1 to detect the edges in the image.
- If vehicles are detected in region 1 then assign green light to that side otherwise skip to next side.
-

Phase 2:

- After 3 seconds detect moving vehicles in region 2 using Background substractionMOG2
- Calculate density of each vehicle in region 2 in each frame and also simultaneously calculate average vehicles density in the last 2 seconds.
- If average vehicle density in last 2 seconds is less than minimum average density then give 5 seconds timer and go to next side.
- Even if the timer reaches 60 seconds go to next side.

5.IMPLEMENTATION

5.1 Code for canny edge detection in region

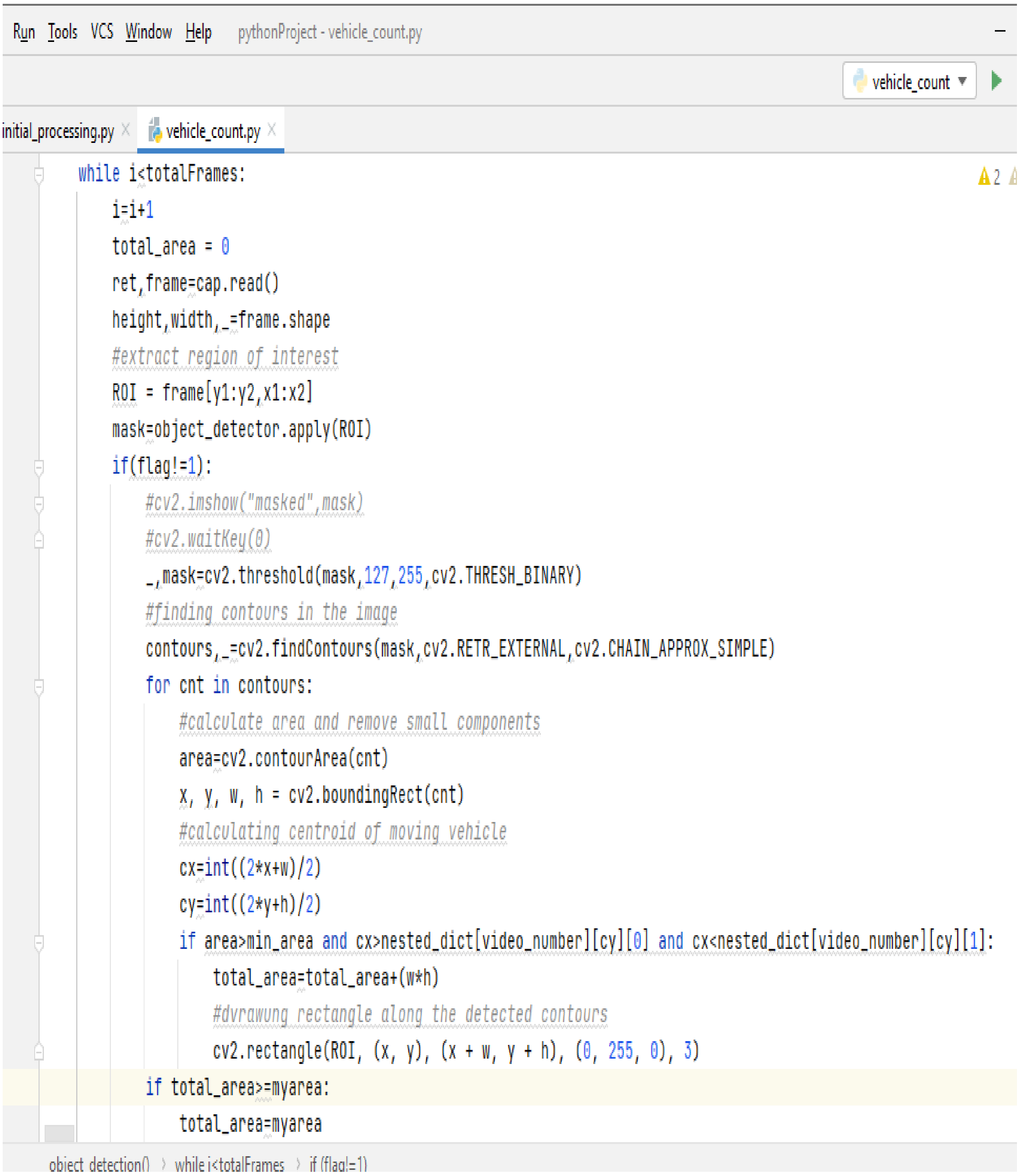

ctor Run Tools VCS Window Help pythonProject - initial_processing.py
vehicle_count
initial_processing.py
vehicle_count.py
PyCharm 2020.3.5 available
Update...
initial_process()

```
def initial_process(cap,frame_number,video_number):
    count=0
    cap.set(cv2.CAP_PROP_POS_FRAMES,frame_number)
    status,initial_frame=cap.read()
    y1 = coordinates[video_number][0]
    y2 = coordinates[video_number][1]
    x1 = coordinates[video_number][2]
    x2 = coordinates[video_number][3]
    initial_frame=initial_frame[y1:y2,x1:x2]
    gray_initial_frame = cv2.cvtColor(initial_frame, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray_initial_frame, 30, 200)
    cv2.imwrite("background2.jpg", edges)

    for y in dict1.keys():
        min1=min(nested_dict[video_number][y])
        max1=max(nested_dict[video_number][y])
        for x in range(min1,max1):
            if edges[y][x]>0:
                count=count+1
    #print(count)
    return count
```

Fig 5.1 Code for canny edge detection in region 1

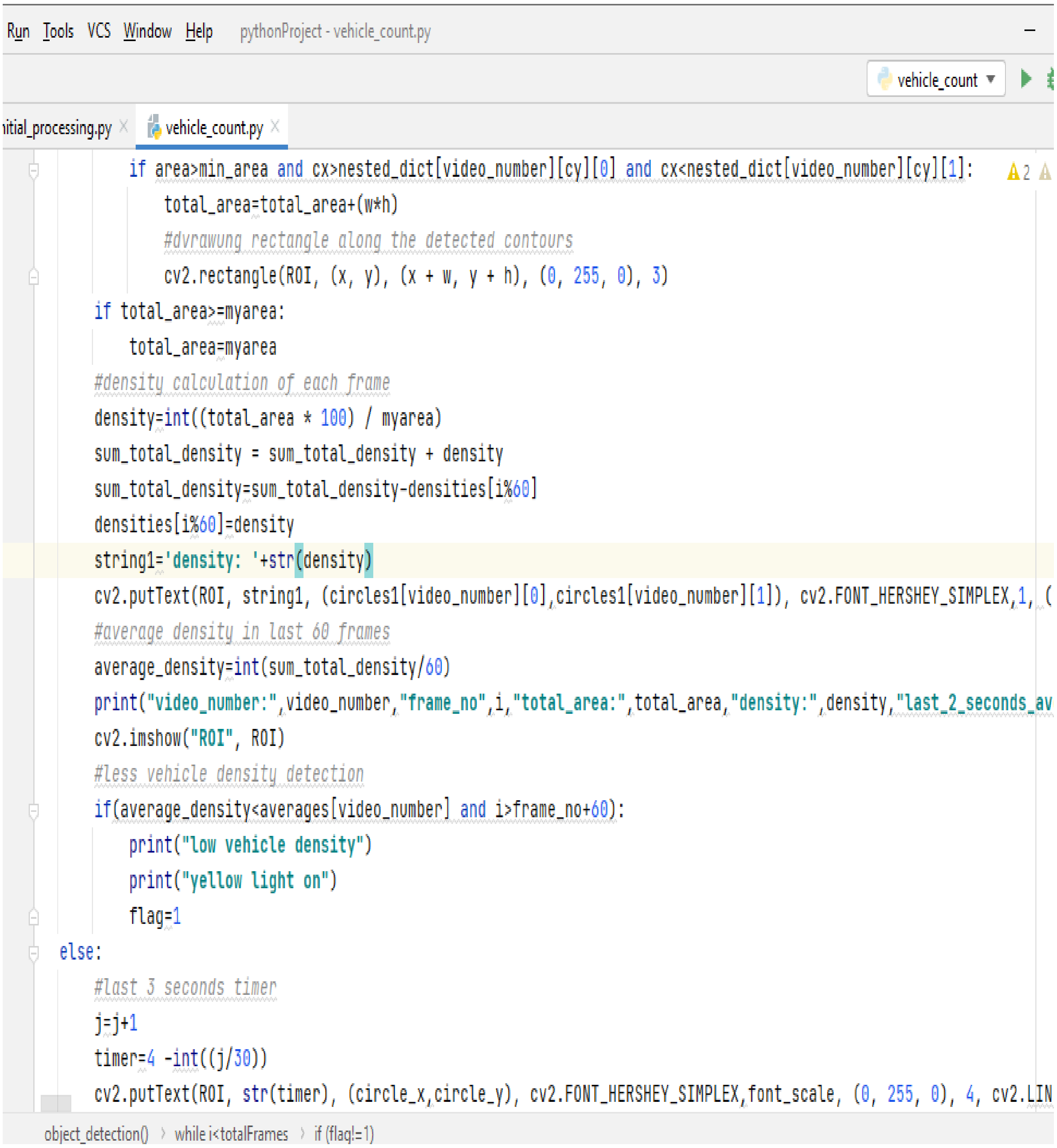
5.2 Code for moving vehicle detection in region 2



```
while i<totalFrames:
    i=i+1
    total_area = 0
    ret,frame=cap.read()
    height,width,_=frame.shape
    #extract region of interest
    ROI = frame[y1:y2,x1:x2]
    mask=object_detector.apply(ROI)
    if(flag!=1):
        #cv2.imshow("masked",mask)
        #cv2.waitKey(0)
        _,mask=cv2.threshold(mask,127,255,cv2.THRESH_BINARY)
        #finding contours in the image
        contours,_=cv2.findContours(mask,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
        for cnt in contours:
            #calculate area and remove small components
            area=cv2.contourArea(cnt)
            x, y, w, h = cv2.boundingRect(cnt)
            #calculating centroid of moving vehicle
            cx=int((2*x+w)/2)
            cy=int((2*y+h)/2)
            if area>min_area and cx>nested_dict[video_number][cy][0] and cx<nested_dict[video_number][cy][1]:
                total_area=total_area+(w*h)
                #dvrawung rectangle along the detected contours
                cv2.rectangle(ROI, (x, y), (x + w, y + h), (0, 255, 0), 3)
        if total_area>=myarea:
            total_area=myarea
```

Fig 5.2 Code for moving vehicle detection in Region 2

5.3 Code for vehicle density detection in region 2

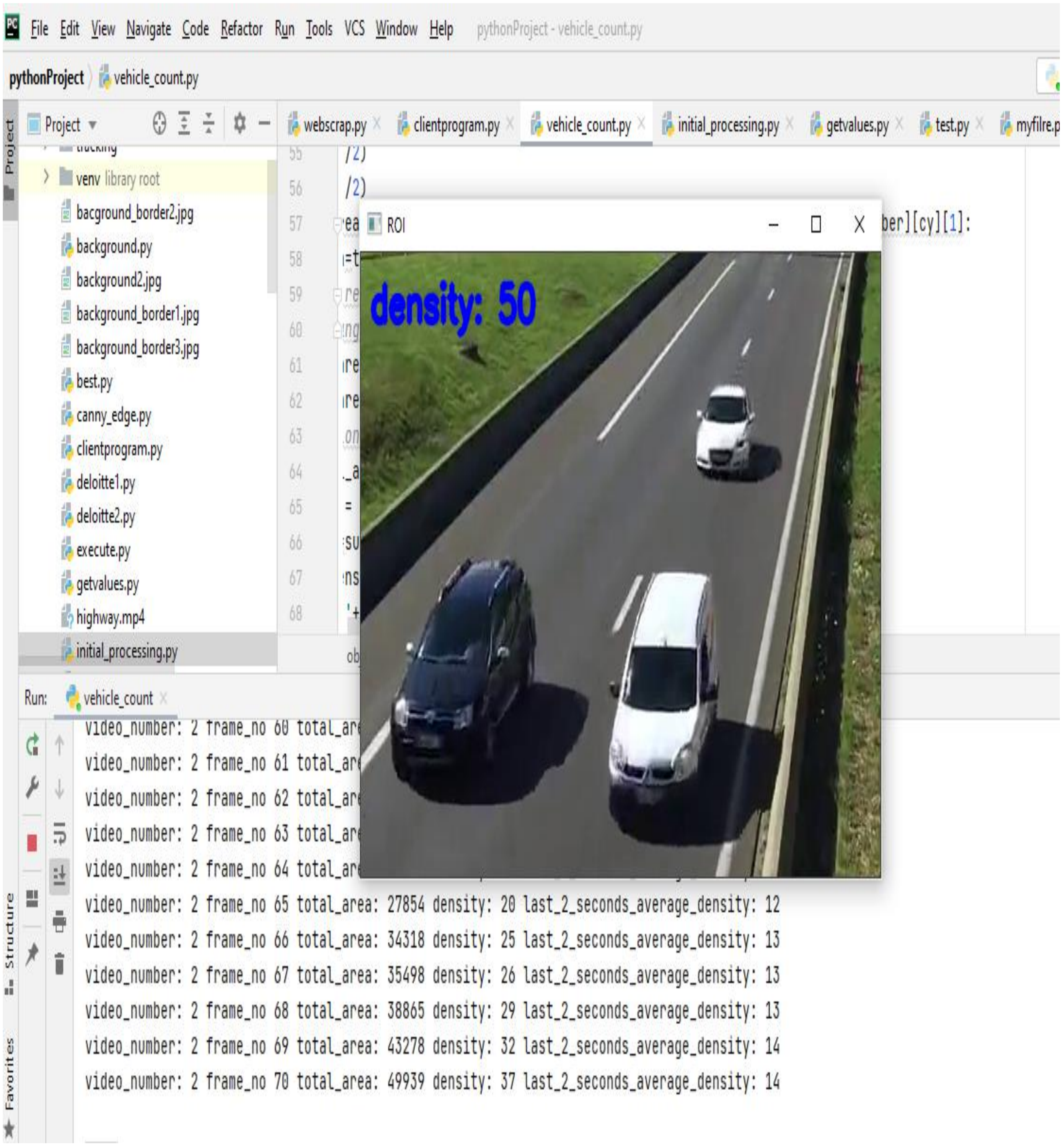

Run Tools VCS Window Help pythonProject - vehicle_count.py
vehicle_count
itial_processing.py
vehicle_count.py
object_detection() › while i<totalFrames › if (flag!=1)

```
        if area>min_area and cx>nested_dict[video_number][cy][0] and cx<nested_dict[video_number][cy][1]:
            total_area=total_area+(w*h)
            #dvrawung rectangle along the detected contours
            cv2.rectangle(ROI, (x, y), (x + w, y + h), (0, 255, 0), 3)
    if total_area>=myarea:
        total_area=myarea
    #density calculation of each frame
    density=int((total_area * 100) / myarea)
    sum_total_density = sum_total_density + density
    sum_total_density=sum_total_density-densities[i%60]
    densities[i%60]=density
    string1='density: '+str(density)
    cv2.putText(ROI, string1, (circles1[video_number][0],circles1[video_number][1]), cv2.FONT_HERSHEY_SIMPLEX,1, (
    #average density in last 60 frames
    average_density=int(sum_total_density/60)
    print("video_number:",video_number,"frame_no",i,"total_area:",total_area,"density:",density,"last_2_seconds_av
    cv2.imshow("ROI", ROI)
    #less vehicle density detection
    if(average_density<averages[video_number] and i>frame_no+60):
        print("low vehicle density")
        print("yellow light on")
        flag=1
else:
    #last 3 seconds timer
    j=j+1
    timer=4 -int((j/30))
    cv2.putText(ROI, str(timer), (circle_x,circle_y), cv2.FONT_HERSHEY_SIMPLEX,font_scale, (0, 255, 0), 4, cv2.LIN
```

Fig 5.3 Code for vehicle density calculation in region 2

6. RESULTS

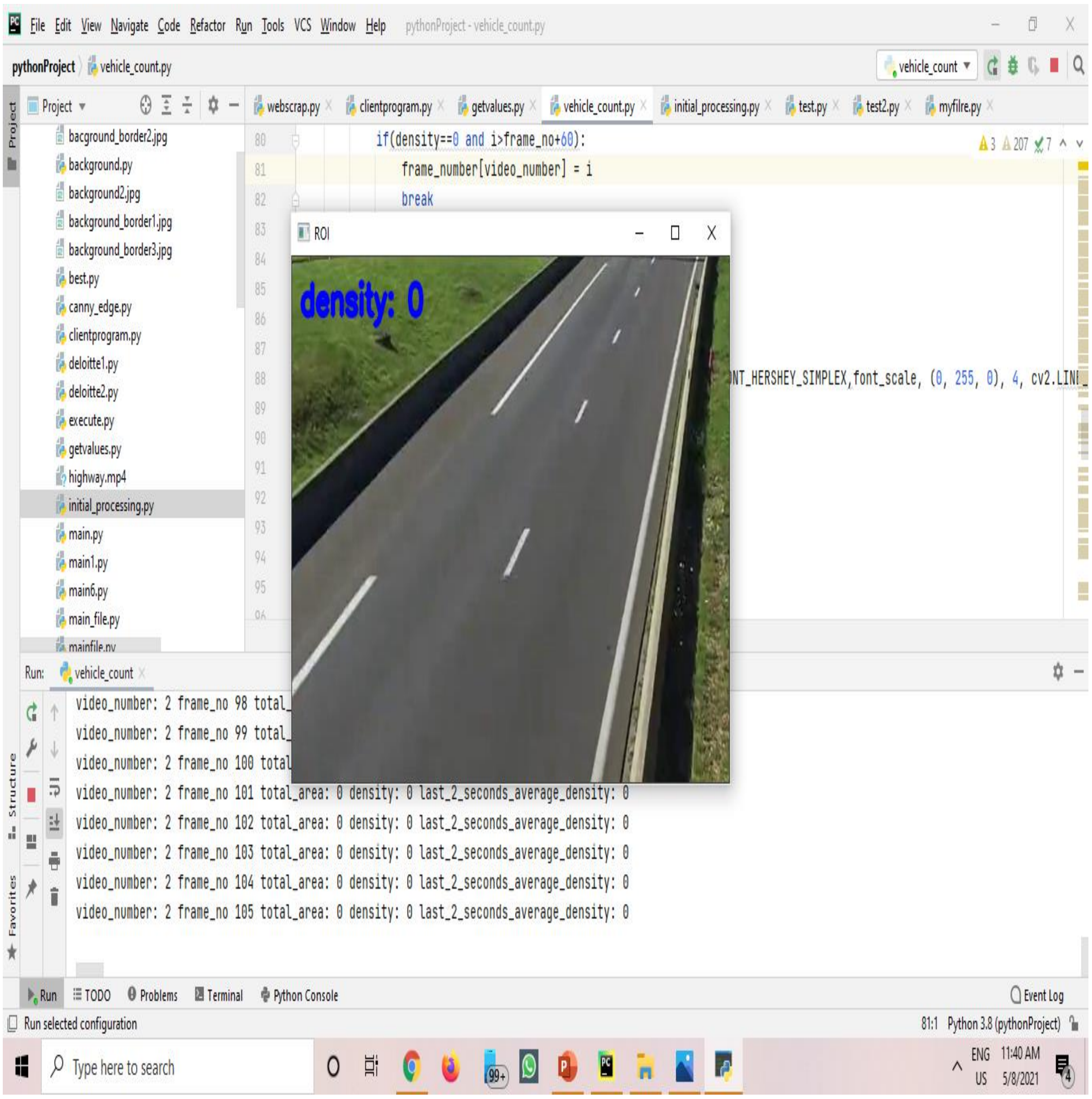
6.1 Density of moving vehicles



```
video_number: 2 frame_no 60 total_ar
video_number: 2 frame_no 61 total_ar
video_number: 2 frame_no 62 total_ar
video_number: 2 frame_no 63 total_ar
video_number: 2 frame_no 64 total_ar
video_number: 2 frame_no 65 total_area: 27854 density: 20 last_2_seconds_average_density: 12
video_number: 2 frame_no 66 total_area: 34318 density: 25 last_2_seconds_average_density: 13
video_number: 2 frame_no 67 total_area: 35498 density: 26 last_2_seconds_average_density: 13
video_number: 2 frame_no 68 total_area: 38865 density: 29 last_2_seconds_average_density: 13
video_number: 2 frame_no 69 total_area: 43278 density: 32 last_2_seconds_average_density: 14
video_number: 2 frame_no 70 total_area: 49939 density: 37 last_2_seconds_average_density: 14
```

Fig 6.1 Density of moving vehicles in region 2

6.2 Low vehicle density detection



```
if(density==0 and i>frame_no+60):
    frame_number[video_number] = i
    break
```

```
NT_HERSHEY_SIMPLEX,font_scale, (0, 255, 0), 4, cv2.LINE_
```

```
video_number: 2 frame_no 98 total_
video_number: 2 frame_no 99 total_
video_number: 2 frame_no 100 total
video_number: 2 frame_no 101 total_area: 0 density: 0 last_2_seconds_average_density: 0
video_number: 2 frame_no 102 total_area: 0 density: 0 last_2_seconds_average_density: 0
video_number: 2 frame_no 103 total_area: 0 density: 0 last_2_seconds_average_density: 0
video_number: 2 frame_no 104 total_area: 0 density: 0 last_2_seconds_average_density: 0
video_number: 2 frame_no 105 total_area: 0 density: 0 last_2_seconds_average_density: 0
```

Fig 6.2 Low vehicle density detection

6.3 Canny Edge detection to detect presence of vehicles in region 1

Canny-edge detected image | **Original image**

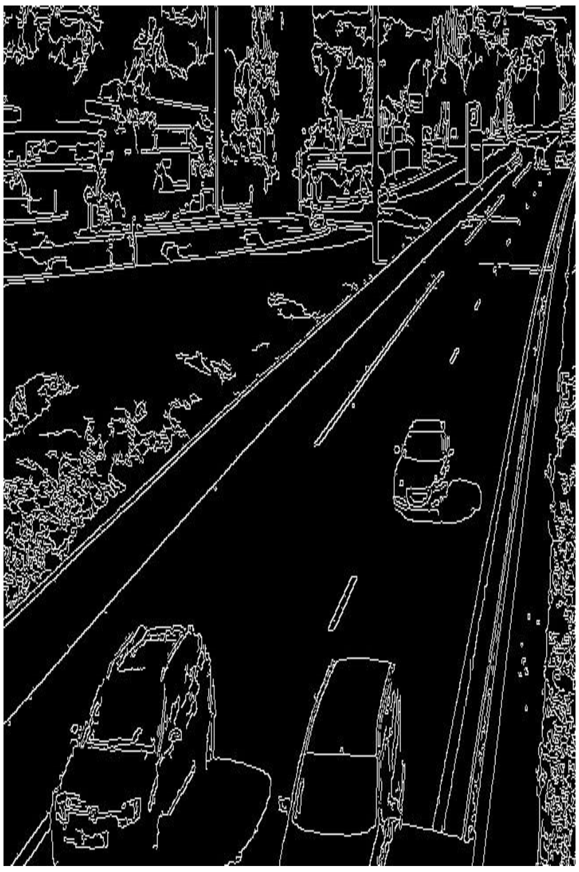


Fig 6.3 Canny edge detected image

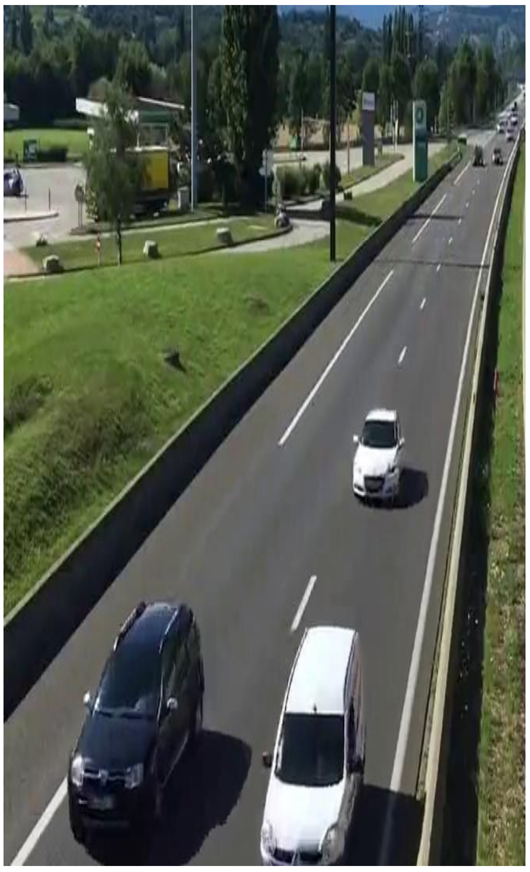


Fig 6.4 Real time image of region 1

7. CONCLUSION

- In this paper a method for estimating the traffic density using Image Processing is presented. This is done by using the images of each lane in order to calculate the traffic density .
- The advantages of this method include such benefits as use of image processing over sensors, low cost, easy setup and relatively good accuracy and speed.
- Because this method has been implemented using Image Processing and Python software, production costs are low while achieving high speed and accuracy
- "Traffic control using image processing" technique that we propose overcomes the limitations of the earlier (in use) techniques used for controlling the traffic.
- Earlier in automatic traffic control use of timer had a drawback that the time is being wasted by green light on the empty. The technique we proposed avoids this problem.

BIBLIOGRAPHY

[1] Mr. Yogesh Shinde , Miss. Hemlata Powar. “Intelligent Traffic Light Controller Using IR Sensors for Vehicle Detection” in IEEE[2018]

[2] Vijayaraman P, P Jesu Jayarin. “Intelligent traffic control system using RFID technology” in IEEE [2019]

[3] Chirag Thakkar, Rajesh Patil. “Smart Traffic Control System Based on Image Processing” in IJRASET[2017]

[4] Ananya Bansal, Pramod Goyal, Vaibhav Kashyap “Intelligent traffic light control system” based on traffic environment using deep learning , IARJSET[2020]