DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

ROAD LANE LINES, OBSTACLES DETECTION USING MACHINE LEARNING

By
Likhith M – ENG19CS0155
Likith S – ENG19CS0156
Sonia Reddy GS - ENG19CS0316
Srujana V - ENG19CS0320
Batch no - **17**

Under the supervision of

Dr. Revathi V
Associate Professor
Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,

(2022-2023)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II work titled **"ROAD LANE LINES, OBSTACLES DETECTION USING MACHINE LEARNING"** is carried out by **Likhith M (ENG19CS0155), Likith S (ENG19CS0156), Sonia Reddy GS (ENG19CS0316), Srujana V (ENG19CS0320),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2022-2023**.

Dr. Revathi V	**Dr Girisha G S**	**Dr. Udaya Kumar Reddy K R**
Associate Professor Dept. of CS&E, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner	**Signature of Examiner**
1.	
2.	

DECLARATION

We, **Likhith M (ENG19CS0155), Likith S (ENG19CS0156), Sonia Reddy GS (ENG19CS0316), Srujana V (ENG19CS0320),** are students of eigth semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled **"ROAD LANE LINES, OBSTACLES DETECTION USING MACHINE LEARNING"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2022-2023**.

Student **Signature**

Name1: Likhith M
USN : ENG19CS0155
Name2: Likith S
USN : ENG19CS0156
Name3: Sonia Reddy GS
USN : ENG19CS0316
Name4: Srujana V
USN : ENG19CS0320

Place : Bangalore
Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Girisha G S, Department Chairman****,* ***Computer Science and Engineering****,* ***Dayananda Sagar University,*** *for providing right academic guidance that made our task possible.*

We would like to thank our guide ***Dr. Revathi V, Associate Professor****,* ***Dept. of Computer Science and Engineering****,* ***Dayananda Sagar University****, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our ***Project Coordinator Dr. Meenakshi Malhotra*** *and* ***Dr, Pramod Naik*** *as well as all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Sl.no	Abbreviations	Full form
1	Open CV	Open-source Computer Vision
2	YOLO V4	You Only Look Once Version4
3	LaneRTD	Lane Real Time Detection
4	ROI	Region Of Interest
5	FPS	Frames Per Second

FIGURES AND TABLES

ABSTRACT

In today's world road safety is becoming a major concern since the number of vehicles used by people have been increasing at a dreadful rate and at the same time there are higher number of accidents caused due to many obstacles faced on roads and also due to the driver's recklessness by taking incorrect lanes and disturbing the other side of the road which is not meant to be taken while driving. The proposed system detects the road lane lines as well as the obstacles such as potholes on the road using computer-based technologies. It mainly aims for the betterment of already proposed models for lane detection by enhancing the already used models and also including new techniques which can detect the lanes clearly and precisely. Here, a model is introduced through which it is possible to detect the lanes on straight roads and curved roads too using OpenCV that is "Open-Source Computer Vision". So, now potholes on roads can be detected mainly using the YOLOv4 technique which is used for real-time object detection. The whole process is tested on real time videos for more accurate vision and compatibility. Through this system it can reduce the number of accidents to certain extent, further future implementations and goals will be discussed at the end.

CHAPTER 1 INTRODUCTION

1.1. INTRODUCTION

When people drive, they use their eyes to decide where to go. The lines on the road that show others where the lanes are act as our constant reference for where to steer the vehicle. Naturally, one of the first things in implementation part would be in developing a self-driving car to automatically detect lane lines using an algorithm. In this project detection of lane lines in images using Python and OpenCV is used. OpenCV means "Open-Source Computer Vision" which is a package that has many useful tools for analyzing images. Road lane Line and Pothole Detection. This is a unique Project mainly focusing on detecting the road line using open CV and Machine Learning. In this project mainly the focus is on detecting the road lanes which would help for autopilot cars and also there will be a working of Pothole detection so that the car is not damaged when the road is not proper. The tools used are color selection, region of interest selection, Grayscaling, Gaussian smoothing, Canny Edge Detection and Hough Transform line detection. The goal is to piece together a pipeline to detect the line segments in the image, then average/extrapolate them and draw them onto the image for display. Once a working pipeline is ready, later it can be tried on the video stream. Potholes are formed due to wear and tear and weathering of roads. They cause not only discomforts to citizens but also deaths due vehicle accidents. The US records more than 2000 fatal accidents per year due to potholes and bad road conditions. There are numerous use cases of this detection system. For example, a civic authority can detect and locate potholes and assess their magnitudes so that they can plan for repairs. Cameras are also installed on moving vehicles can detect potholes in real time and help drivers avoid potholes.

1.2. OBJECTIVES

- The goal is to create a model of road lane line detection and also to create a better model which can detect the potholes in the road in real time without compromising accuracy.
- The final model is expected to be useful for finding the potholes with their proper location, alongside generating the warning for the riders also and keeping the generated data.

1.3. SCOPE

Our main focus is to detect the road lanes and potholes so that there won't be any threat of accidents for the autopilot vehicle as the lanes are detected and even potholes also due to which the machine can be trained so that it won't cross the lane and adjust the position, also because of detecting the pothole the vehicle will slow down or shift accordingly as per requirements.

Chapter 2 PROBLEM DEFINITION

Lane Line detection is a critical component for self-driving cars and also for computer vision in general. This concept is used to describe the path for self-driving cars and to avoid the risk of getting in another lane. The pothole detection typically involves designing an algorithm or system that can automatically detect and locate potholes from images or video footage captured by cameras mounted on vehicles or other devices. The ultimate goal of pothole detection is to improve road safety by identifying and repairing potholes in a timely manner, reducing the risk of accidents and damage to vehicles.

Chapter 3 LITERATURE SURVEY

[1] Malik Haris, Adam Glowacz, "Lane Line Detection Based on Object Feature Distillation," Artificial Intelligence, 8 May 2021.

The paper discusses lane detection, which is defined as a problem of pixel-level classification. The authors propose a method to improve the ability of the network to detect lane lines without increasing computational cost. The method involves adding a decoder branch to recover lane line boundary refinement information, which is then used to generate a lane line segmentation probability map. The feature pixel classification prediction and pixel-level distillation loss function are applied to the network, and the direct up-sampling branch learns from the probability map generated by the decoder branch. This approach improves the F1Measure and makes the algorithm better adapted to various types of road scenes.

The proposed method is evaluated on several scenarios, including straight lines, curves, backlit scenes, and vehicle occlusion scenes. The experimental results show that the algorithm has good detection accuracy, strong robustness, and higher detection speed. The authors also found that incorporating prior geometric information between lane lines effectively improves the performance of lane detection.

In future work, the authors plan to fuse the lane detection and classification results with a forward collision warning strategy. This approach will be helpful in assisting drivers to avoid collisions caused by accidental lane deviation, fatigue, or driving under the influence of controlled substances in complex or straight road environments structured.

Overall, the paper proposes a novel method to improve the ability of the network to detect lane lines, which is critical for autonomous vehicles and driver assistance systems. The experimental results demonstrate the effectiveness of the proposed method, and future work will focus on integrating the lane detection and classification results with other advanced driving assistance systems.

[2] Mamta Joshi, Ashutosh Vyas, “Comparison of Canny edge detector with Sobel and Prewitt edge detector using different image formats”, In: International Journal of Engineering Research & Technology (IJERT) IJERT ISSN: 2278-0181, June 2021

So, now it can be observed from the numerical results that the sobel operator and prewitt operator shows the highest average PSNR (peak-to-signal-ratio). On the other hand, canny operator shows the least average PSNR (peak-to-signal-ratio) among the other three operators.

Usually, the edge detector with the least PSNR will have highest edge detection capabilities. In this case canny edge has the least average MSE (Mean-square-error) and least average PSNR (peak-to-signal-ratio). We also observed that in less complex images canny edge shows maximum output and vice versa. This is because canny operator is more suitable for detecting weak edges and therefore, has a higher chance of detecting false edges. Canny edge detector is more efficient in nature compared to other detectors.

So, finally to produce a clear image we need to detect more accuracy of edges but using prewitt and sobel operator we cannot produce more accuracy of edges which will make the detection part easier.

[3] P. S. Ezekiel, O. E. Taylor & D. J. S. Sako, “Smart System for Potholes Detection Using Computer Vision with Transfer Learning”, In: International Journal of Innovative Science and Research Technology Volume 6, Issue 7, July ‑ 2021

Here, we used OpenCV technique where we first start collecting and gathering images of potholes. Later, these images were pre-processed using image annotation and data augmentation. We also labelled each image so that it will be easy for identification. Then using transfer- learning we downloaded a YOLOv3 pre-trained weight file and here we use a web cam for detection of potholes. In YOLOv3, Darknet53 is used as the backbone to extract features from an input image. The average precision for small objects are improved using YOLOv3 but it is not much compatible for medium and large objects. YOLOv3 cannot be used in sensitive domains.

[4] Bin Liu, Hongzhe Liu, Jiazheng Yuan, "Lane Line Detection based on Mask R-CNN," Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology, April 2019.

A lane detection algorithm based on Convolutional Neural Network is proposed. Jun Li et al used CNN and a Recurrent Neutral Network (RNN) to the detect lane boundary in a standard form than any other previous ones. Here CNN works on providing the geometric information about the structure of the lane and the same output information is used by RNN in order to detect the lanes.

Usually Traditional vision-based methods detect lane lines based on the camera image characteristics and features such as color gradient, histogram or edge where as, Visual-based solutions can be divided into two main categories.

One is feature-based method that allows to distinguish feature points of lane lines according to road characteristics such as color, gradient or edge. Another kind of model-based method establishes the mathematical model of road structure. They use the geometric coordinates of cameras and lanes as input parameters and depend on their accuracy. For the purpose of determining the parameters, the initial configuration information is merged with the feature points of lane lines extracted from road images.

[5] Hyunwoo Song, Kihoon Baek and Yungcheol Byun, "Pothole Detection Using Machine Learning", In: Advanced Science and Technology Letters Vol.150 (AST 2018), pp.151-155, Feb 2018

In this paper, detection is done using a smart phone and classification is performed using Transfer Learning, timely alerts to drivers are given to identify potholes and humps avoid accidents. Ultrasonic sounds are used to identify the potholes and humps. Utilized smart phone as a sensor to acquire moment information and sensed data will be fed into a classifier to detect the status of a road.

[6] Mingfa Li, Yuanyuan Li, and Min Jiang, "Lane Detection Based on Connection of Various Feature Extraction Methods", In: Hindawi Advances in Multimedia Volume 2018, Article ID 8320207, August 2018

The main motive here is to improve the efficiency and accuracy of real-time lane detection. There are 2 main steps involved: (1) image pre-processing and (2) the establishment and matching of line lane detection model.

In image pre-processing not only process the image itself but we also do colour feature extraction and edge feature extraction. In order to reduce the noise factor in the process of motion and tracking, after extracting the colour features of the image, we need to use Gaussian filter to smooth the image. Then, the image is obtained by binary threshold processing and also a process called morphological closure.

After pre-processing do the colour transformation, then the colour extraction and later adding colour extraction in pre-processing. After this adding of edge detection in Pre-processing takes place. Currently, here only the Hough transformation is used to detect straight lanes to track lane and do not develop advanced lane detection methods. In the future, there will be a more advanced lane detection approach to improve the performance.

[7] Zhong-xun Wang, Wenqi Wang, "The research on edge detection algorithm of lane", In: Wang and Wang EURASIP Journal on Image and Video Processing (2018) 2018:98, Oct 2018.

Based on the intelligent perception of the vehicle's driving environment, when the driver operates improperly, the system will automatically warn the driver through voice and other forms to correct the current erroneous driving operation. The system will also automatically replace the driver's control of the vehicle in special circumstances such as the driver's sleep or poor mental status, thereby avoiding traffic accidents. When natural and non-natural factors change, it is easy for drivers to make wrong driving judgments

on specific road condition information which leads to traffic accident. If the vehicle can be judged incorrectly by the driver at this time, the correct judgment of the road conditions and timely warning to guide the driver's wrong operation will be able to effectively avoid the occurrence of traffic accidents. In the edge detection stage, the Kirsch algorithm has obvious superiority, and this paper upgrades on the basis of the original classical algorithm, which makes the image processing speed more than twice the original, expands the scope of use, and improves the detection and identification of lane lines. Speed, and while speeding up, does not affect the detection of lane lines.

Through the improved Hough transform and Kirsch operator, the robustness and adaptability of the detection results are enhanced, the redundant information of the operator is reduced, the computational complexity of the algorithm is reduced by the matrix operation, and the detection of lane lines is improved.

[8] Wael Farag, Zakaria Saleh, "Road Lane-Lines Detection in Real-Time for Advanced Driving Assistance Systems", In: International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, Nov 2018.

In this paper, a vision-based approach capable of reaching a real-time performance in the detection and tracking of structured road boundaries with a slight curvature, and is robust enough in the presence of shadow conditions, is proposed.
In this system, the lane detection is done based on edge detection. The captured image is transformed to a new mapping which is based on bird's eye view of the road. In this view, the lane boundaries or dashed lines appear very close to vertical lines with contrast color on a black background. An adaptive filtering method is employed to detect and isolate vertical line segments that can be interpolated to construct longer lane lines.
All possible forms that lane markings can take place in an image are parametrized is a collection of shapes. Then, an evaluation function is constructed to give a numerical value to how well a particular lane shape/marking is matching the pre-specified parametrized lane forms. Then, the maximum value of this function, at a particular position in the image, is used to highlight that a lane is detected.
Hough transformation is used and applied to edge images to detect the lane boundaries with the assumption that the lane lines are long enough with smooth curving. In this paper, the captured RGB color image of the camera has a size of 960x540 pixels. This image is then converted to grayscale, as the first step of the LaneRTD (real time detection), in order to lower the processing time. The next step the algorithm does is lowering the noise in the grayscale image, to avoid incorrect edge detection, by applying the Gaussian Blur algorithm.

The Hough transformation is then used to detect potential straight-line segments, in the edged image, that can be later part of the lane boundary. The search for these line segments is only done within the ROI. In this paper, a fast and reliable lane-lines detection and tracking technique is developed, presented thoroughly and given the name “LaneRTD (real time detection”.

[9] Seung-Ki Ryu, Taehyeong Kim, and Young-Ro Kim, "Image-Based Pothole Detection System for ITS Service and Road Management System," Mathematical Problems in Engineering, vol. 2017, p. 10, September 2015.

Here, several efforts have been made for developing a technology that can automatically detect and recognize potholes, which may contribute to the improvement in survey efficiency and pavement quality through prior investigation and immediate action.
Existing methods for pothole detection can be divided into vibration-based methods, three-dimensional (3D) reconstruction-based methods, and vision-based methods.
These vision-based methods are cost-effective when compared with 3D laser scanner methods. Because of the distorted signals generated by noise in collecting image and video data it is difficult to accurately detect a pothole using these methods.
So, a pothole detection method using various features in 2D images is proposed for improving the existing pothole detection method and accurately detecting a pothole. Also, the performance of the proposed method is compared with that of the existing method for several conditions such as road, recording, and brightness.

[10] EMIR BUZA, SAMIR OMANOVIC, ALVIN HUSEINOVIC, "Pothole Detection with Image Processing and Spectral Clustering," Recent Advances in Computer Science and Networking, pp. 48-53, 2013.

The paper proposes a novel unsupervised method for detecting and estimating potholes on roads. The method is based on image processing and spectral clustering, a technique used for identifying structures in data using the spectral properties of a pairwise similarity

matrix. The first phase of the proposed method involves detecting frames with defects, which are then analyzed using the spectral clustering approach.

The method utilizes inexpensive and widely available equipment, such as off-the-shelf digital cameras, mounted on passenger vehicles for video acquisition. The data collected from the grayscale images is analyzed using histogram-based techniques to identify regions of interest. To test the efficacy of the proposed method, the researchers conducted experiments on 50 different pothole images. The detection accuracy was calculated manually, and the results showed that the proposed method was able to identify all potholes in the given dataset. Additionally, the surface estimation accuracy was found to be 81%, indicating that the method is suitable for rough estimation of potholes. The researchers further highlight that the proposed method is cost-effective, making it a viable solution for detecting potholes in regions where resources are limited. The results of the experiments demonstrate that the proposed method detected potholes with reasonable accuracy.

In conclusion, the paper proposes a new unsupervised method for detecting and estimating potholes on roads using image processing and spectral clustering. The method was tested on 50 different pothole images, and the results showed that it was able to detect potholes with reasonable accuracy. The proposed method is cost-effective and utilizes inexpensive equipment, making it a viable solution for detecting potholes in regions where resources are limited. Future work will focus on improving the estimation of potholes when the road has several cracks.

Chapter 4 PROJECT DESCRIPTION

4.1 SYSTEM DESIGN

Lane Line detection is a critical component for self-driving cars and also for computer vision in general. This concept is used to describe the path for self-driving cars and to avoid the risk of getting in another lane. In this Project, building a machine learning project takes place to detect lane lines in real-time.

This is done using the concepts of computer vision that is using OpenCV library. To detect the correct lane the detection of white markings on both sides of the lane must happen. Using computer vision techniques in Python, the identification of road lane lines will take place in which autonomous cars must run.

This will be a critical part of autonomous cars, as the self-driving cars should not cross the lane which may lead its direction in opposite lane to avoid accidents.

To detect white markings in the lane, first, the masking for the rest part of the frame will be done. This is done using frame masking.

Potholes are formed due to wear and tear and weathering of roads. They cause not only discomforts to citizens but also deaths due vehicle accidents.

This describes an approach of building a real time detection system. There are numerous use cases of this detection system. For example, a civic authority can detect and locate potholes and assess their magnitudes so that they can plan for repairs.

Figure 4.1: Original road lane line image

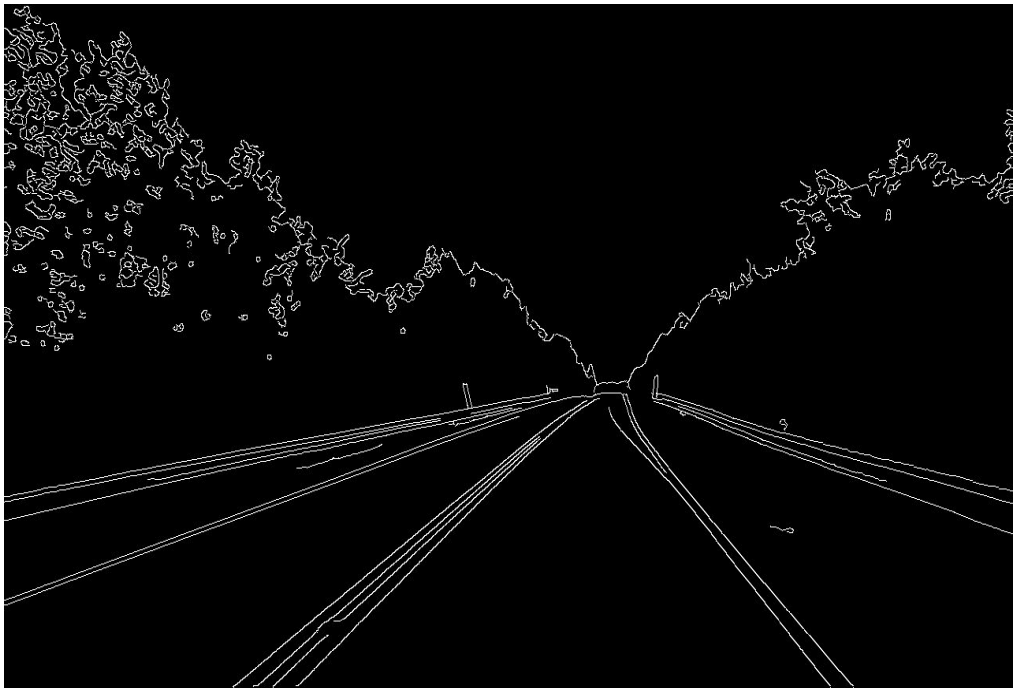

Figure 4.2: Road Lane Lines after applying Canny edge detector

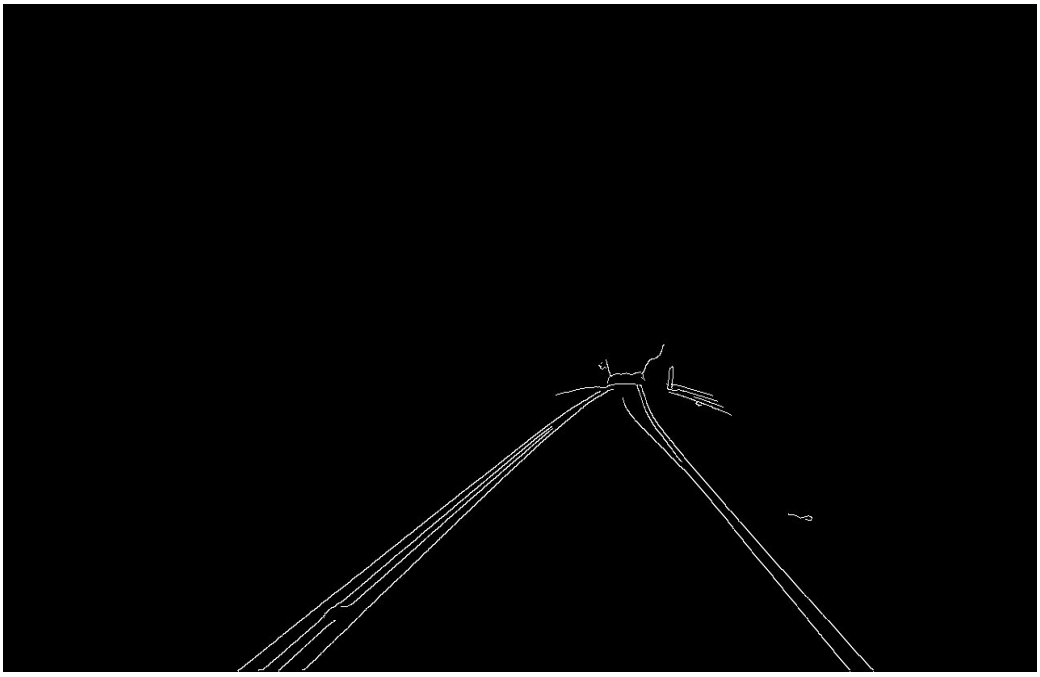

Figure 4.3: Road Lane Lines after applying Region Of Interest

Figure 4.4: Road Lane Lines after applying Hough Transformation

Figure 4.5: Potholes on road (during day)

Figure 4.6: Potholes on road (during night)

So, the use of YOLOv4(You Only Look Once Version4) algorithm is used for the project. This algorithm is a one stage detection model which use basically "VGG 16" Convolutional Neural Network architecture for training and Softmax actication function for classification on darknet-53. So, in this project there is a model created for pothole detection using yolov4(tiny- for real time detection) architecture with a mix of self-annotated data and a little pre-annotated Roboflow data.

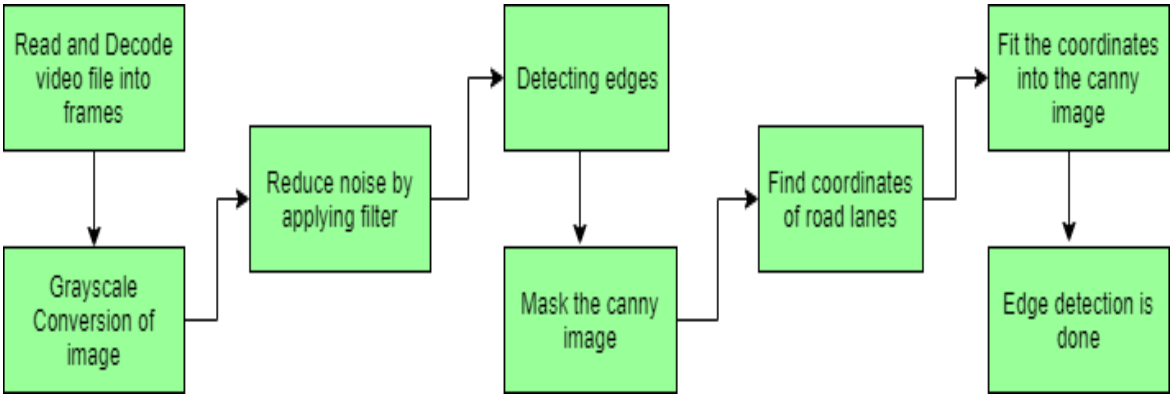

Figure 4.7: Block Diagram for road-lane line detection system

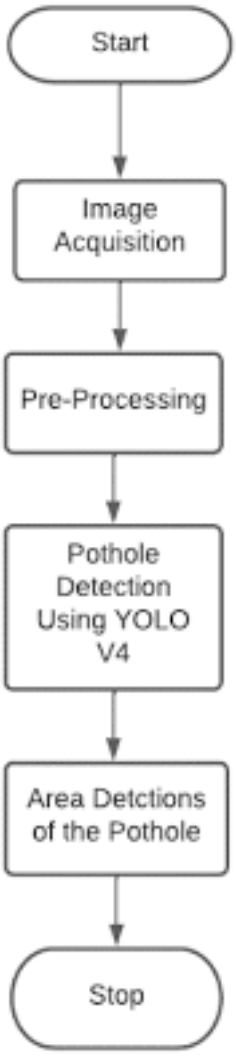

Figure 4.8: Block Diagram for pothole detection system

4.2 ASSUMPTIONS AND DEPENDENCIES

- The model works accurately.
- The lane markings are clearly visible which makes us easy to detect.
- The model can assist drivers to go on correct lane.

Chapter 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS

- Systems are heavily software based, often relying on probabilistic methods and object models.
- Describe the path for self-driving cars and to avoid the risk of getting in another lane.
- Detect the white markings on both sides on the lane.
- Conversion of pixels to a line in Hough Transform space.
- Process each frame of video to detect lane.

5.2 NON-FUNCTIONAL REQUIREMENTS

- The application should be reliable since it is dealing with a medical condition.
- The application should be fast and responsive.
- The application should be rolled out with regular patchwork and updates.
- The application should be compatible with all kinds of smartphones.

5.3 HARDWARE AND SOFTWARE REQUIREMENTS

- PyCharm IDE.
- Installation of all required libraries.
- 2gb RAM
- 200mb Storage for smooth functioning

Chapter 6 METHODOLOGY

Most of the principles, concepts and important topics used which are used to focus through the research papers, the source was IEEE papers and from these research papers more information and ideas about Lane detection that is Hough Transform, region of interest and pre-existing models such as YOLO V4 and OpenCV were used. After this there were more deeper concepts to be seen through and learnt such as grayscaling, blob, region of interest and converting the images in the form of matrix form so that a clear image can be obtained. The main code structure, algorithm implementation, images, storing of the input format and modules helpful from journals, research papers, algorithm uses in real life and stackoverflow.

So, these are the final steps followed for road lane line detection:

I. Capturing and decoding video file frame by frame.
II. Conversion of the Image to GrayScale.
III. Applying filters to reduce noise in video frames.
IV. Edge Detection Using Canny Edge detection method.
V. Finding the region of interest and working on that part.
VI. Detecting lanes using Hough line transform.

These are the final steps followed for pothole detection system:

I. Download and prepare the pothole detection dataset in the required YOLOv4 format.
II. Download the YOLOv4-Tiny pre-trained weights.
III. Modify the configuration file for the YOLOv4-Tiny model to carry out fixed-resolution training.
IV. Training YOLOv4-Tiny model with fixed-resolution images.
V. Modify the YOLOv4-Tiny configuration file to carry out multi-resolution training.
VI. Training YOLOv4-Tiny model with multi-resolution images.
VII. Download the YOLO4 pre-trained weights.
VIII. Modify the configuration file for the YOLOv4 model for multi-resolution training and train the model.

IX. Carry out one final training experiment by modifying the configuration file and training with fixed-resolution images.

X. Run inference on real-world pothole detection videos and analyze the results.

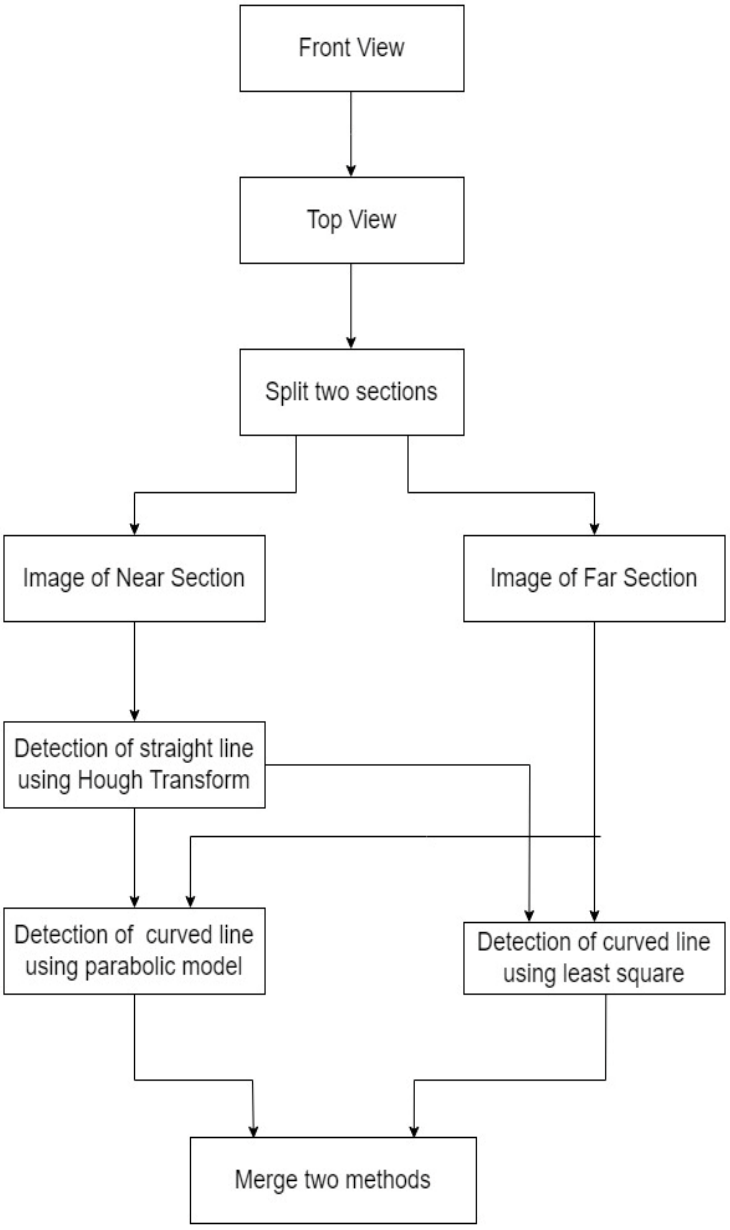

Figure 6.1: Flow Diagram of the Road Lane Line Detection system.

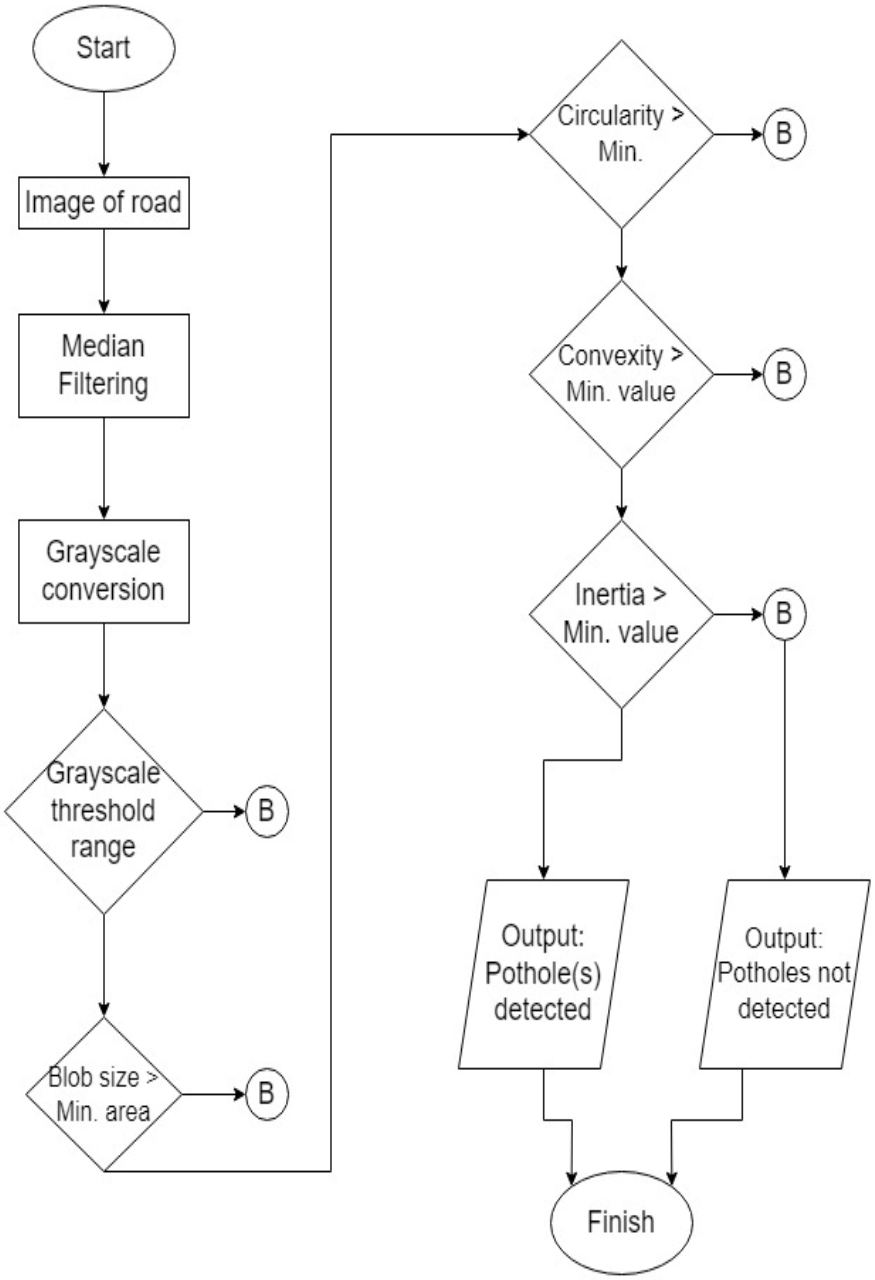

Figure 6.2: Flow diagram of the pothole detection system.

6.1 Comparison of Sobel edge detector, Prewitt edge detector and Canny edge detector

Edge detection is a type of image segmentation techniques which is used to simplify the image data to minimize the amount of data to be processed. Edge detection is important in image processing for object detection, so it important to have a good understanding of edge detection methods. Edges are used to characterize boundaries of an image and are therefore considered for prime importance in image processing. Edge detection filters out useless data, noise while preserving the important structural properties in an image. The comparative analysis of Sobel, Prewitt and Canny Image Edge Detection methods is presented. It has been shown that the Canny's edge detection algorithm performs better than Sobel, and Prewitt edge detectors.

i. Sobel Operator:

The Sobel operator consists of a pair of 3×3 convolution kernels as shown in Figure 7.1. One kernel is simply the other rotated by 90°.

-1	0	1
-2	0	2
-1	0	1

G_x

1	2	1
0	0	0
-1	-2	-1

G_y

Figure 6.3: Sobel Mask

These kernels are designed to respond maximally to edges which are vertical and horizontal relative to the pixel grid, one kernel for each of the two perpendicular orientations. The kernels can be applied separately to the input image. This produces separate measurements of the gradient component in each direction (call these Gx and Gy).

After this, these can be combined together to find the absolute magnitude of the gradient at each point and the orientation of that gradient. The gradient magnitude is given by:

$$|G|=|Gx| +|Gy|$$

The angle of orientation of the edge (relative to the pixel grid) giving rise to the spatial gradient is given by:

$$\theta=\arctan(Gy/Gx)$$

ii. Prewitt's operator

Prewitt operator is similar to the Sobel operator and is used for detecting vertical and horizontal edges in images. It is a fast method for edge detection. The prewitt edge detector is a good method to estimate the magnitude and direction of an edge. The prewitt operator is limited to 8 possible directions, however most direction estimates are not much more accurate. This gradient based edge detector is estimated in the 3x3 neighborhood for 8 directions. All the eight convolution masks are calculated. The convolution mask with the largest module is then selected. The convolution masks of the Prewitt detector are given below in figure 7.3.

1	1	1
0	0	0
-1	-1	-1

-1	0	1
-1	0	1
-1	0	1

Figure 6.4: Prewitt's Mask

iii. Canny Edge Detector: Canny edge detection is one of the basic algorithms used in iris recognition. The algorithm uses a multi-stage process to detect a wide range of edges in images. The Canny edge detector is based on the gradient magnitude of a smoothed image: local maxima of the gradient magnitude that are high are identified as edges. The motivation for Canny's edge operator was to derive an 'optimal' operator in the sense that it,

- Minimizes the probability of multiply detecting an edge.
- Minimizes the probability of failing to detect an edge.
- Minimizes the distance of the reported edge from the true edge.

The first two of these criteria address the issue of detection, that is, given that an edge is present will the edge detector find that edge (and no other edges). The third criterion addresses the issue of localization that is how accurately the position of an edge is reported. There is a tradeoff between detection and localization – the more accurate the detector the less accurate the localization and vice versa [6]. The canny edge detection first removes noise from image by smoothening. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (non-maximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

A block diagram of the canny edge detection algorithm is shown in Figure 7.3.

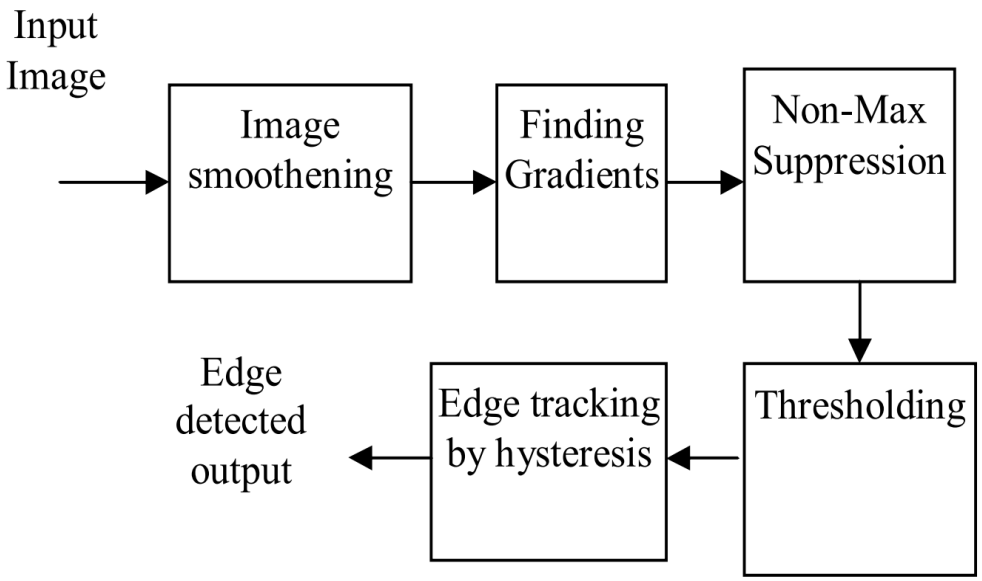

Figure 6.5: Block diagram of the stages of the Canny edge detector

The stages of canny edge detection are:

1. Smoothing: By applying Gaussian filter, smoothing of and image is done to reduce noise.

2. Finding gradients: The edges have been marked where the gradients of the image is having large magnitudes.

3. Non-maximum suppression: Only local maxima have been marked as edges.

4. Thresholding: Potential and actual edges are determined by thresholding.

5. Edge tracking by hysteresis: Edges that are not connected with strong edges have been suppressed.

a. Detection using Sobel edge detector

It returns edges at those points where the gradient of the image is maximum. Fig.6 displays the results of applying the Sobel method to an image.

Figure 6.6: Image after Sobel edge detection

b. Detection using Prewitt edge detector

The Prewitt method finds edges using the Prewitt approximation to the derivative. It returns edges at those points where the gradient of the image is maximum. The output image is given in figure 7.5.

Figure 6.7: Image after Prewitt edge detection

c. Detection using Canny edge detector

The Canny method finds edges by looking for local maxima of the gradient of the image. The gradient is calculated using the derivative of the Gaussian filter. The method uses two thresholds to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. The resultant image is shown in figure 7.6.

Figure 6.8: Image after canny edge detection

Finally, after analysis we can say that canny edge detector is able to detect maximum number of edges. Canny edge detector gives good results for detection of horizontal and vertical edges .it is also able to detect the circular edges and edges at the corner. Prewitt and Sobel show good results for detecting horizontal and vertical edges.

Chapter 7 EXPERIMENTATION

7.1 SOFTWARE DEVELOPMENT

Majorly 3 works were involved:

(i) One is extracting the frames of the video

(ii) Grayscaling each image and

(iii) Detecting the lanes according to the grayscaled images.

Here is the code which is used for capturing the frames as seen here that our input is given as a mp4(video) ret, frame = cap.read() frame = lanesDetection(frame) cv.imshow('Lanes Detection', frame) in this line of the code the whole video is getting converted to frames and in future, will use the same frames for grayscaling and using it of region of interest.

The code snippet related to this is given below:

```
def videoLanes():
    cap = cv.VideoCapture('./img/Lane.mp4')
    while(cap.isOpened()):
        ret, frame = cap.read()
        frame = lanesDetection(frame)
        cv.imshow('Lanes Detection', frame)

        if cv.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv.destroyAllWindows()
```

As told before the major task was to work on grayscaling, in this below code the images are scaled in particular width and height and the images have been moved in grayscale format and then Hough transform is used to get lanes in the image and the images with lines were returned as a result.

The code snippet is given below:

```
region_of_interest_vertices = [
```

```
    (200, height), (width/2, height/1.37), (width-300, height)
]
gray_img = cv.cvtColor(img, cv.COLOR_RGB2GRAY)
edge = cv.Canny(gray_img, 50, 100, apertureSize=3)
cropped_image = region_of_interest(
    edge, np.array([region_of_interest_vertices], np.int32))

lines = cv.HoughLinesP(cropped_image, rho=2, theta=np.pi/180,
                threshold=50, lines=np.array([]), minLineLength=10, maxLineGap=30)
image_with_lines = draw_lines(img, lines)
# plt.imshow(image_with_lines)
# plt.show()
 return image_with_lines
```

This was about how the lanes have been detected but now the discussion is about the potholes as mentioned here certain data is tested and trained in YOLO V4(You Only Look Once Version 4) algorithm where the data or images are checked if it is matching with the frame and if yes then, opting out the distance and dimensions here and putting out the notification and beep sound.

The code snippet is given below:

```
def loc():
                g = geocoder.ip('me')
                return g.latlng
            def record():
                try:
                    time2 = datetime.datetime.now().strftime(' %H-%M-%S')
                    l = loc()
                    lat, long = l[0], l[1]
                    mydb = client["Data"]
                    inf = mydb.pothole
```

```
        #for rec in inf.find():
            #if lat not in rec['latitude']:
        rd = {'date': time1, 'time': time2, 'latitude': lat, 'longitude': long}
        inf.insert_one(rd)
    except:
        pass
```

Chapter 8 TESTING AND RESULTS

8.1 RESULTS

The obtained results show that our algorithm performs very well under various criteria and different categories.

After the use of the required algorithms, these are the results obtained accordingly.

The road lane markings are shown for both:

(a) Straight lanes

(b) Curved paths

This helps drivers to follow the correct lane while avoiding taking the wrong routes.

During day light as shown below it is clearly seen that the lanes are detected with high precision and accuracy as seen here.

The algorithm Canny edge detection has proven to be very effective in detecting lines with clarity by giving clear edges, removing noise from the image and increasing the contrast of the image too.

Now the discussion is about Hough Tranformation, the main reason Hough Lines are used is, to detect whether the edges detected are the actual lines or not. So, Hough transform in any way needs to detect the edges in order to provide a more efficient and evident results.

Also, there is a need to remove or eliminate noise before applying Hough Tranform to that image which is a very important step before the completion.

When there are various diversions or deviations also Hough Transform algorithm proved to be very efficient.

Let us look into the pictorial representation of details in detail obtained finally.

Fig 8.1 – Green lines indicating lane lines for straight path

Fig 8.2 – Lane line detection for curved paths

Fig 8.3 – Sample pothole image

Fig 8.4 – Pothole detection (square box)

Fig 8.5 – Pothole a larger one along with area and frames per second

Fig 8.6(a) – Pothole detection during night time

Fig 8.6(b) – Pothole detection during night time

8.2 DISCUSSION OF RESULTS

The sample image shows the lane lines but not accurately and precisely in figure 8.1.
Here after applying all the algorithms and techniques as a result it is a filtered image that is only detecting the lane lines and not the unnecessary part to its left and right which is seen in the image 8.2 is detected. This tested image shows the lane markings (light green) which has been detected, it is a straight path as shown in figure 8.2.

Previously only Canny edge detection was used but after using Hough Transformation, the detection of the edges present on the road were the actual lane lines or not was identified and also results for curved paths have been obtained. The image as a result is as shown above in figure 8.3.

Now moving on to the next part of the model, here the obtained results show that potholes are being detected on road along with the area of a pothole, the frames per second are also calculated and indicated as shown in figure 8.4.

This is done using YOLO v4 algorithm which detects the potholes more accurately.
The image is taken in the daylight as seen. This tested image shows the pothole being detected, along with the area in square feet(sq.ft) and frames per second(FPS) that is 21 towards the left corner on the top.

As seen in figure 8.5 the result obtained is a box shaped structure wherein the pothole is being detected. So, when the image of a road is taken as a whole, the exact part indicating the pothole which makes it easy for the driver to avoid it and choose the other path.

This is a very efficient method since the accuracy obtained also is high with precise frame per second values.

The average FPS(frames per second) obtained in our model is 28 which is considered to be effective.

So, till now it was seen with the outcome that is the detection of potholes in daylight, but here the model was developed to detect in night time also since the path would not be as clear as during the day time. After testing and training the results obtained are as shown in figure 8.6.

Therefore, the model has been implemented during day and night times also.

Chapter 9 CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

The lane detection has proved to be an efficient technique to prevent accidents in Intelligent Transportation Systems. The system uses various techniques such as image processing, computer vision, and machine learning algorithms to detect and track the road lane lines.

The accuracy and reliability of road lane line detection systems have improved significantly over the years due to advancements in technology.

Main motive is to detect the road line using Open-source Computer Vision and Machine Learning. Mainly the focus is on detecting the road lanes which would help for autopilot cars to drive safely and we would work on Pothole detection so that the car is not damaged when the road is not proper. So, finally the algorithms and tools used were the Region of Interest, Selection of the color, Canny edge detection, Hough transformation algorithm, Gray Scaling which detected both straight lines and curved lines in one picture.

The model can be implemented in real life also, and the results obtained are with more efficiency and accuracy where 28 frames per second has been obtained as the maximum frames using YoloV4 tiny algorithm in pothole detection.

First part was detecting the lanes and drawing the lanes as per the colour required.

After that process was completed next, certain steps were followed to detect the potholes Data preparation, training and detection thus the whole process of pothole detected has been completed. Together, the lane line detection and potholes detection has been achieved.

Overall, road lane line detection systems have become an integral part of modern vehicle safety systems, and their continued development and improvement will help to make our roads safer for everyone.

9.1 SCOPE FOR FUTURE WORK

The model here has successfully detected the lanes and potholes and it can also detect the distance of the pothole from the camera, but this also has few limitations which needs to be processed and worked up in the future and it has been listed out those pointers where the work in this project in the future can be carried out.

More improvements can be made by taking into the consideration the blur images which is obtained from the camera.

At night it is common that the lanes are only visible till the car light hits the road through headlight but in this case when the image is optimized, and the result video is generated the far and near ratio of the video gets disturbed and the detection for lane lines which are too far, this is also one of the criteria to be considered and worked on. In this case another category such as detecting the lanes using Raised pavement marker which could be very useful in detecting far away lanes and perfect accuracy at night as well.

The potholes are detected perfectly but a marker could be created once at a time so, in the future it is possible to implant multiple markers with the distance and detections at a time.

Chapter 10 REFERENCES

[1] Malik Haris, Adam Glowacz, "Lane Line Detection Based on Object Feature Distillation," Artificial Intelligence, 8 May 2021.

[2] Mamta Joshi, Ashutosh Vyas, “Comparison of Canny edge detector with Sobel and Prewitt edge detector using different image formats”, In: International Journal of Engineering Research & Technology (IJERT) IJERT ISSN: 2278-0181, June 2021

[3] P. S. Ezekiel, O. E. Taylor & D. J. S. Sako, “Smart System for Potholes Detection Using Computer Vision with Transfer Learning”, In: International Journal of Innovative Science and Research Technology Volume 6, Issue 7, July ‑ 2021

[4] Bin Liu, Hongzhe Liu, Jiazheng Yuan, "Lane Line Detection based on Mask R-CNN," Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology, April 2019.

[5] Hyunwoo Song, Kihoon Baek and Yungcheol Byun, “Pothole Detection Using Machine Learning”, In: Advanced Science and Technology Letters Vol.150 (AST 2018), pp.151-155, Feb 2018

[6] Mingfa Li, Yuanyuan Li, and Min Jiang, “Lane Detection Based on Connection of Various Feature Extraction Methods”, In: Hindawi Advances in Multimedia Volume 2018, Article ID 8320207, August 2018

[7] Zhong-xun Wang, Wenqi Wang, “The research on edge detection algorithm of lane”, In: Wang and Wang EURASIP Journal on Image and Video Processing (2018) 2018:98, Oct 2018

[8] Wael Farag, Zakaria Saleh, “Road Lane-Lines Detection in Real-Time for Advanced Driving Assistance Systems”, In: International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, Nov 2018

[9] Seung-Ki Ryu, Taehyeong Kim, and Young-Ro Kim, "Image-Based Pothole Detection System for ITS Service and Road Management System," Mathematical Problems in Engineering, vol. 2017, p. 10, Sept 2015.

[10] EMIR BUZA, SAMIR OMANOVIC, ALVIN HUSEINOVIC, "Pothole Detection with Image Processing and Spectral Clustering," Recent Advances in Computer Science and Networking, pp. 48-53, 2013.

Chapter 11 SAMPLE CODE

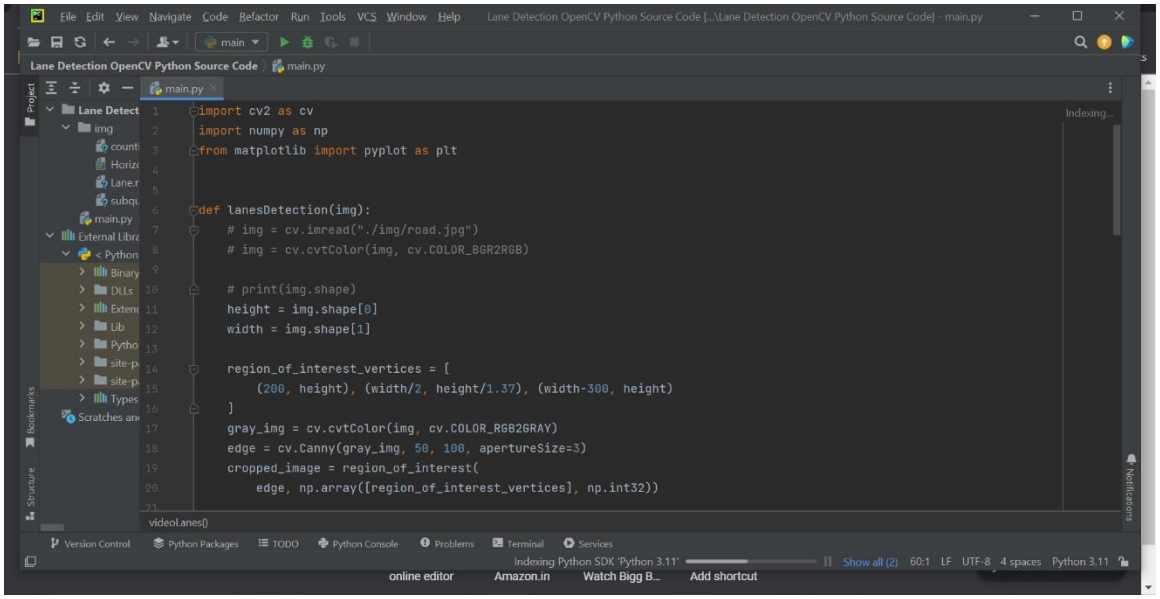

Fig 11.1(a) – Code for Grayscaling and Canny edge detector(road lane lines)

This is the code for road lane line detection which shows the, Grayscaling of the image, Canny Edge detector part

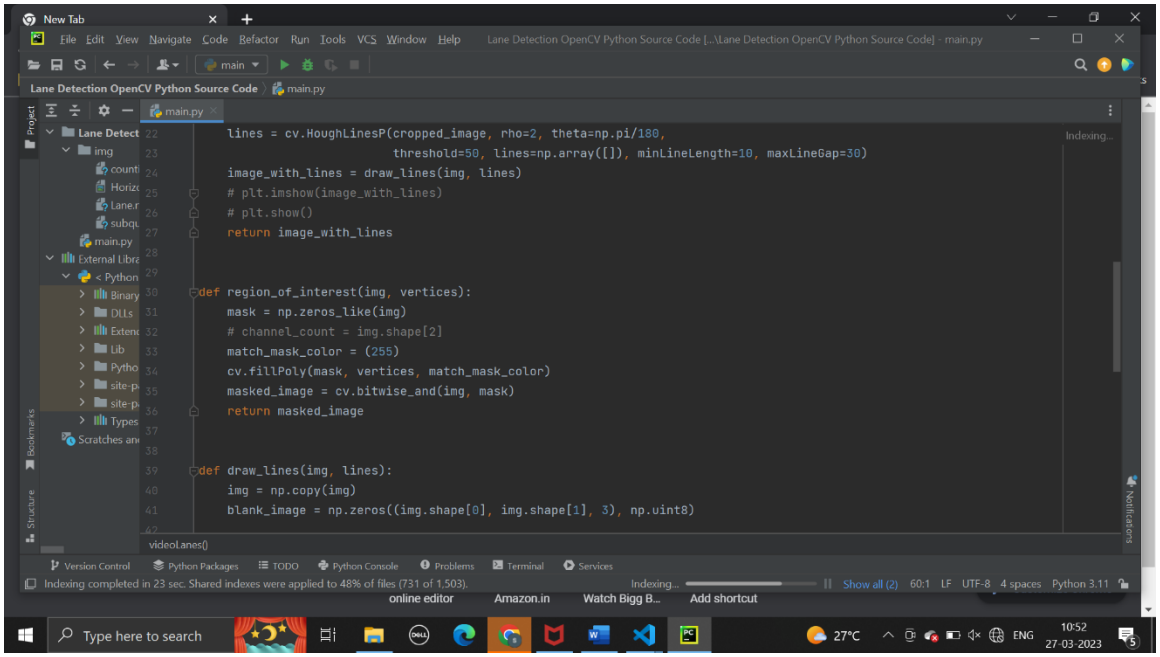

Fig 11.1(b) – Code for Region of Interest and color masking part

This is the code for road lane line detection which shows the Region of Interest and colour masking part

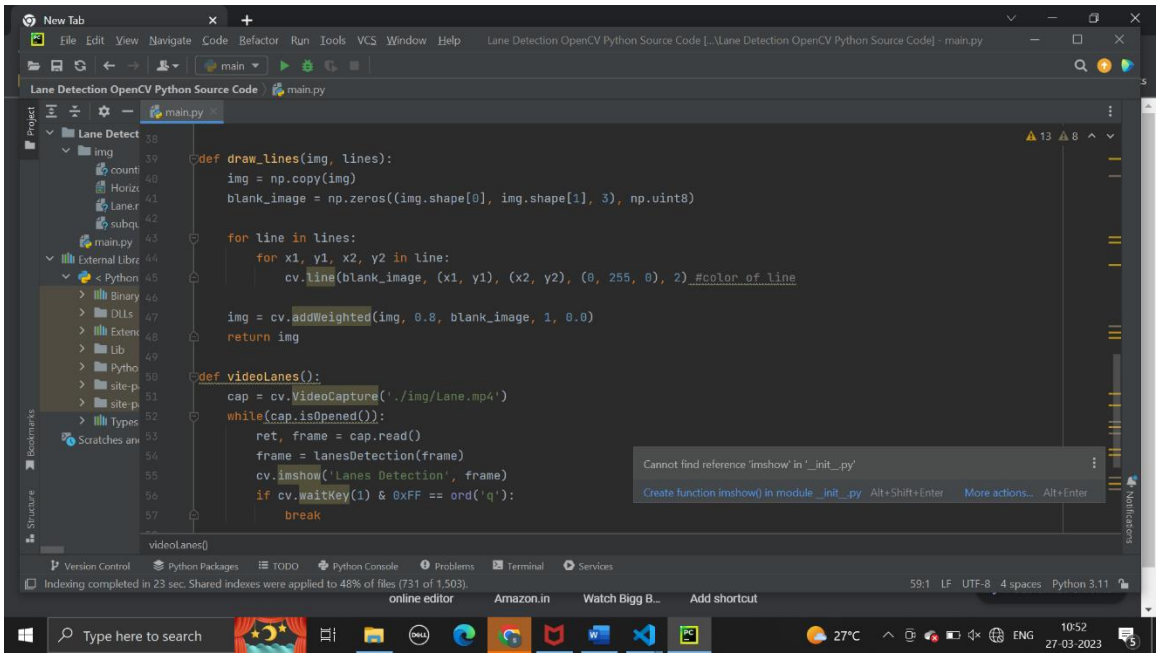

Fig 11.1(c) – Code for drawing lines and adding weights.

This is the code for drawing the road lane lines and adding the weights to it.

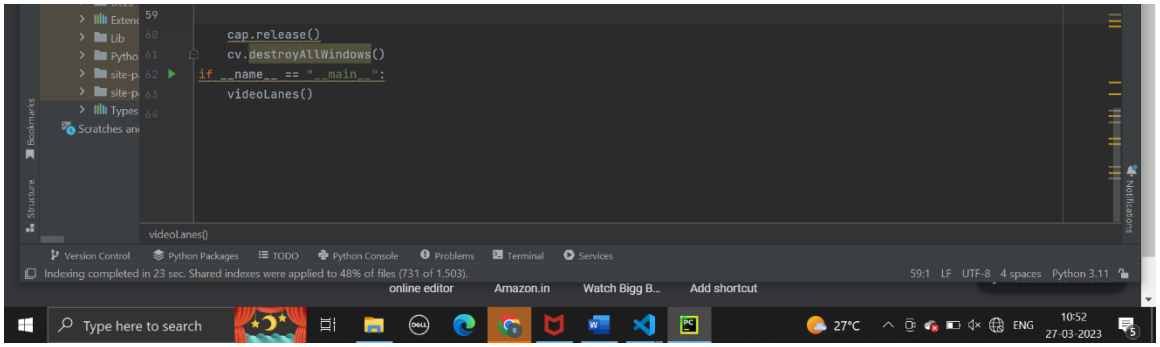

Fig 11.1(d) – Code for exiting the code.

This is the code for exiting out of the code

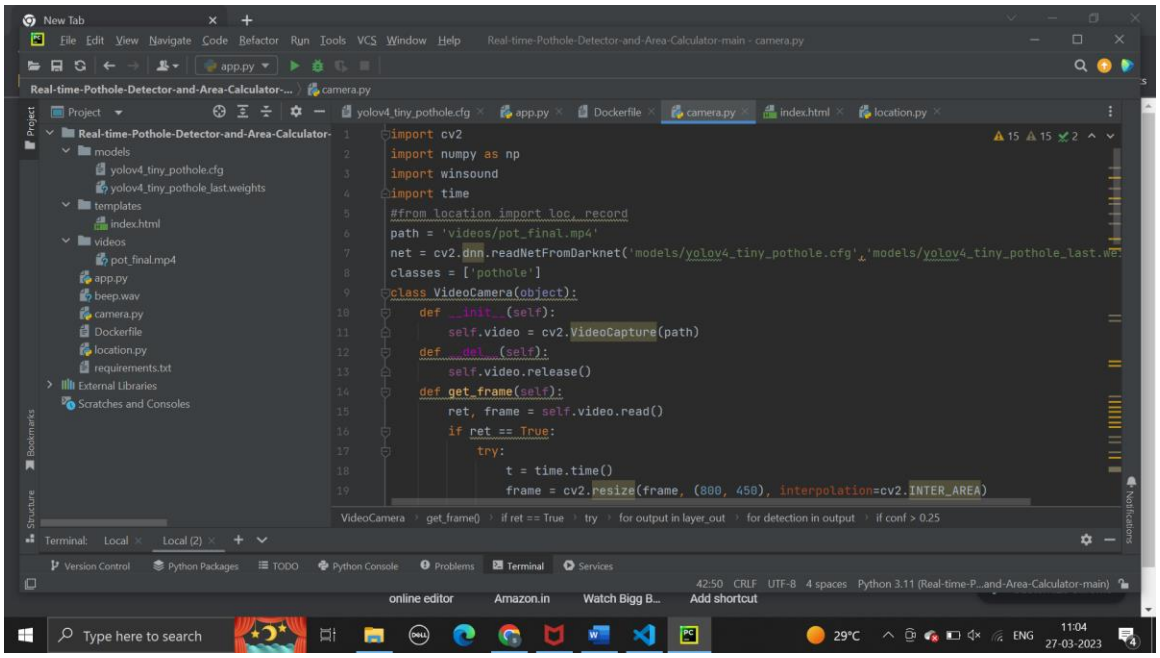

Fig 11.2(a) – Code for importing libraries for pothole detection

This is the importing of libraries part.

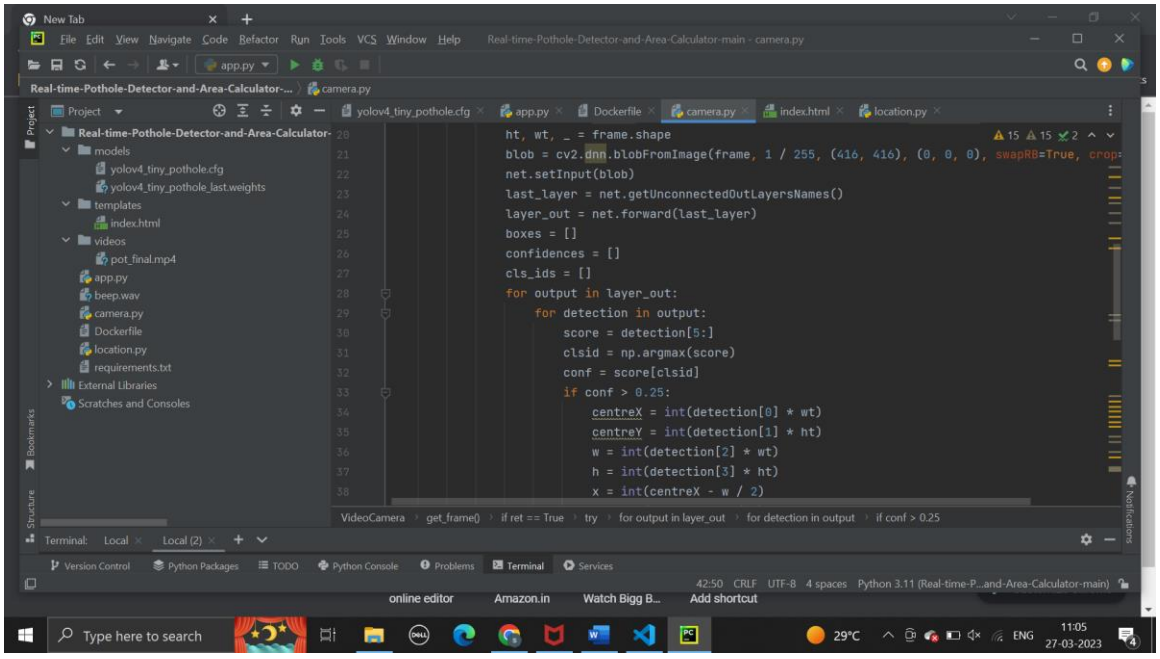

Fig 11.2(b) – Code for blob size and making box

This code is for the blob size and making the box shaped images

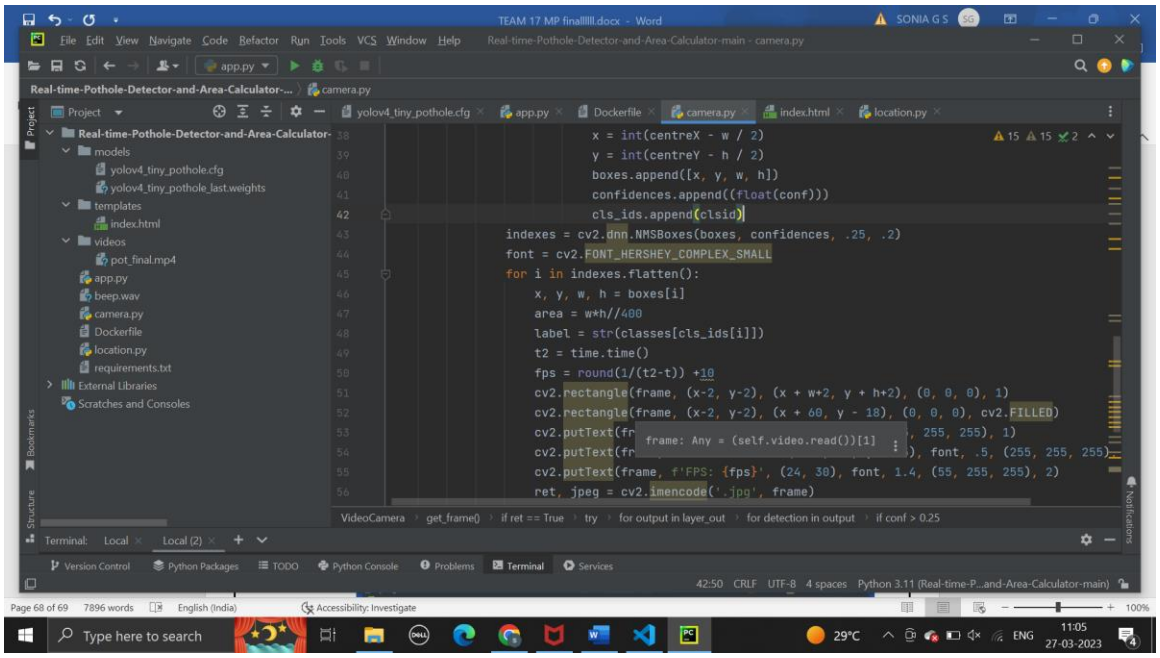

Fig 11.2(c) – Code for making the rectangle shaped box

This is for the dimensions for the rectangle shaped box where pothole fits

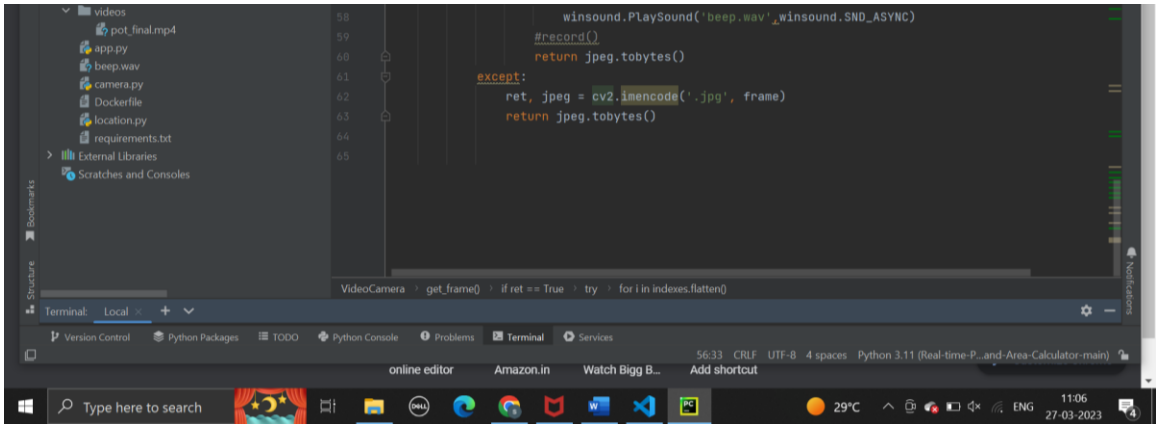

Fig 11.2(d) – Code for converting image of pothole from jpeg to bytes

This is for converting the image from jpeg to bytes

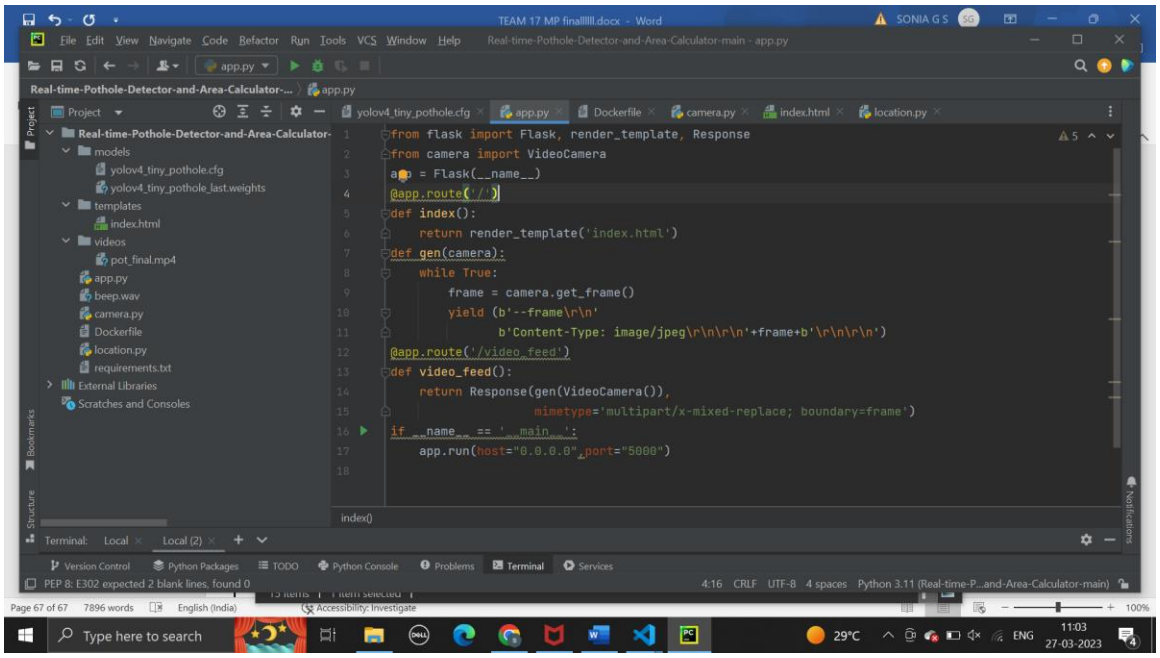

Fig 11.2(e) – Code for Grayscaling and Canny edge detector

This is the main part because the port number and html part is included.

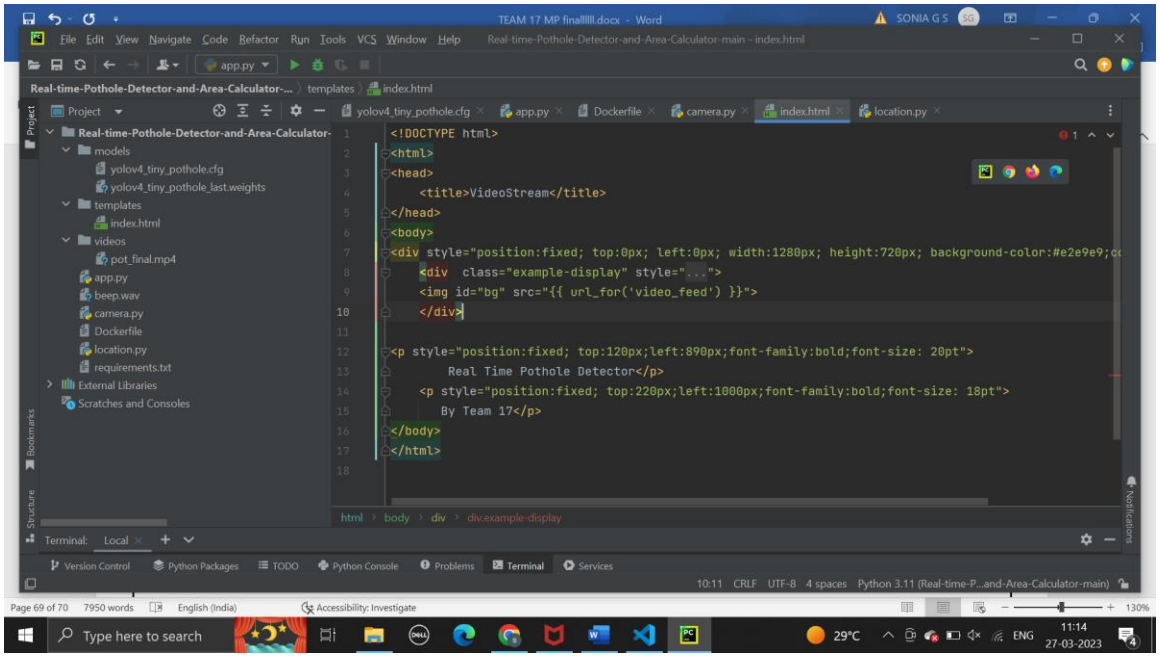

Fig 11.2(f) – Code for designing the layout

This part of code is for the HTML, that is the outer layout.

FUNDING AND PUBLISHED PAPER DETAILS

Second International conference on Advanced Trends in Mechanical & Automation Science (ATMA -2023)

Paper No. & Topic: ATMA23-EC03, Road Lane Lines, Obstacles Detection Using Machine Learning.

Our paper will be published in **Journal-Elsevier-Materials Today Proceedings (**Materials Today Proceedings is indexed in Scopus, Web of Science, and CPCI).

GITHUB LINK (SOURCE CODE)

https://github.com/soniareddygs/Team_17_Road_Lane_Lines-Obstacles_Detection_using_Machine_Learning