GUARDIAN ANGEL ON THE ROAD: A SMART DETECTION AND RECOVERY NOTIFICATION SOLUTION

A PROJECT REPORT

Submitted by

A. SARMILA	-	410121243052
R. PREETHI	-	410121243042
K. SHAGUFTHA	-	410121243053
R. SUSHMA	-	410121243057

In partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

ADHI COLLEGE OF ENGINEERING AND TECHNOLOGY, SANKARAPURAM

MAY 2025

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
BONAFIDE CERTIFICATE

This is to certify that the project report entitled "SMART ACCIDENT DETECTION AND RECOVERY NOTIFICATION SOLUTION" submitted by Mr./Mrs_____with Reg.No._____to the department of **Artificial Intelligence and Data Science**, is a Bonafide record of work carried out under my supervision in partial fulfillment of the requirements for the award of the **Bachelor of Technology** degree in **Artificial Intelligence and Data Science** under **Anna University** during the academic year 2024–2025.

SIGNATURE

Mrs. L. MAHALAKSHMI,

ASSISSTANT PROFESSOR,

Artificial Intelligence and Data Science.

Adhi college of engineering and technology.

SIGNATURE

Dr. V. BELMER GLADSON,

HEAD OF THE DEPARTMENT,

Artificial Intelligence and Data Science.

Adhi college of engineering and technology.

Submitted for the Viva-Voice examination held on_____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We, express our gratitude and thanks to **Our Parents** first for giving health and a sound mind for completing this project. We give all the glory and thanks to our almighty **GOD** for showering upon, the necessary wisdom and grace for accomplishing this project.

It is our pleasant duty to express deep sense or gratitude to our Honorable Chairman **Dr. S. SARAN RAJ**, and our Beloved CEO Mam **Smt. SUJATHA MARAN**, for their kind encouragement.

We have unique pleasure in thanking our Principal **Dr.A.DEVARAJU, M.Tech., Ph.D**., for his unflinching devotion, which lead us to complete this project.

We express our faithful and sincere gratitude to our Head of the Department **Dr.V.BELMER GLADSON, M.E. Ph.D.,** Department of Artificial Intelligence and Data Science

We express our faithful and sincere gratitude to our Main-Project Coordinator **S. FEBEENA EZHIL JOTHI, M.Tech., Assistant Professor**, Department of Artificial Intelligence and Data Science.

We express our faithful and sincere gratitude to our guide **L. MAHALAKSHMI, M.E, Assistant Professor**, Department of Artificial Intelligence and Data Science for his valuable suggestions.

We render our thanks to all the **Faculties and Lab Technicians** of Department of Artificial Intelligence and Data Science for their timely assistance.

TABLE OF CONTENTS

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPTER 7

CHAPTER 8

CHAPTER 9

LIST OF FIGURES

LIST OF ABBREVIATION

SI. No	TITLE
1.	AI - Artificial Intelligence
2.	GMM - Gaussian Mixture Model
3.	GPS - Global Positioning System
4.	ML - Machine Learning
5.	CPU - Central Processing Unit
6.	ROI - Region of Interest
7.	FPS - Frames Per Second
8.	API - Application Programming Interface
9.	Geopy - Python Library for Geolocation
10.	BBox - Bounding Box
11.	YOLO - You Only Look Once

ABSTRACT

Computer vision-based accident detection through video surveillance has become a beneficial but daunting task. In this project, detection of road accidents is proposed. The proposed framework capitalizes on axis bounding box technique for accurate object detection followed by an efficient centroid based GMM algorithm for surveillance footage. The proposed framework provides a robust method to achieve a high Detection Rate and a low False Alarm Rate on general road-traffic CCTV surveillance footage. This framework was evaluated on diverse conditions such as broad daylight, low visibility, rain, hail, and snow using the proposed dataset. This framework was found effective and paves the way to the development of general-purpose vehicular accident detection algorithms in real-time. This project also use geopy library to capture the live location and we can send notification to the nearby police station and hospital with the snap of accident image. So by seeing the image they can take necessary resource allocation and the recovery is made very easy in less time. Alarm buzzer is also included to notify the nearby people.

KEYWORDS: Axis bounding box, Centroid-based GMM, Machine Learning, Image processing, Video Analytics, Real-Time System.

CHAPTER 1

INTRODUCTION

Expressways, highways and roads are becoming overcrowded with increasing of large number of vehicles. Intelligent transportation systems (ITS), applied to collect, cognize, and manage information about transportation flows from various sources, are emerging worldwide to make transportation more efficient, reliable, cleaner and safer. The requirement to detect, track, and count the moving vehicle is getting very important for traffic flow monitoring, planning, and controlling. The vehicle detections can be traditionally achieved through inductive loop detector, infrared detector, radar detector or video-based solution. Compared to other techniques, the video-based solutions based on surveillance camera mounted outdoor are easily influenced by environments such as weather, illumination, shadow, etc.

However, because video-based systems can offer several advantages over other methods such as traffic flow undisturbed, easily installed, conveniently modified, etc., they have drawn significant attention from researchers in the past decade. A traditional computer vision method for moving objet detection in video-based system is so-called “background subtraction”, or computing the difference between a background model and current frame, which demands to estimate a robust background to deal with the changing object. For this reason, in the case of vehicle detection on road, an adaptive rather than static background is needed for real-time road situations.

Regarding to real-time vehicle tracking system, the crucial issue is initiating a track automatically. Here we describe two systems in which the problem is attacked quiet differently. First one is virtual detector constructed with a set of rectangular regions in each frame. Because the camera is fixed, the virtual detector can be chosen to span each lane and the system then monitors changes in area of virtual detector that indicate the presence of a vehicle. The second one is blob tracking.

1.1 Back ground :

In this algorithm, a background model is generated for the scene. For each input image frame, the absolute difference between the input image and the background image is processed to extract foreground blobs corresponding to the vehicles on the road. Both above two approaches have

difficulty tackling shadows, occlusions, and large vehicles (e.g., trucks, trailers), all of which cause multiple vehicles to appear as a single region. To overcome the previous limitations in vehicle detection and tracking, we present an improved method in this paper to accurately separate the vehicle foreground from the adaptive background model through a combination of Otsu's thresholding method and moving cast shadow detection method . This paper is organized as the following.

The adaptive categorizing of background and foreground algorithm bundling with Otsu's method and shadow detection method. Object detection is a fascinating field in computer vision. It goes to a whole new level when we're dealing with video data. The complexity rises up a notch, but so do the rewards! We can perform super useful high-value tasks such as surveillance, traffic management, fighting crime, etc. using object detection algorithms.

Object detection is a fascinating field in computer vision. It goes to a whole new level when we're dealing with video data. The complexity rises up a notch, but so do the rewards!

We can perform super useful high-value tasks such as surveillance, traffic management, fighting crime, etc. using object detection algorithms. There are a number of sub-tasks we can perform in object detection, such as counting the number of objects, finding the relative size of the objects, or finding the relative distance between the objects. All these sub-tasks are important as they contribute to solving some of the toughest real-world problems.

Background subtraction is critical in many computer vision applications. We use it to count the number of cars passing through a toll booth. Background subtraction is a common computer vision task. We analyze the usual pixel-level approach. We develop an efficient adaptive algorithm using Gaussian mixture probability density. Recursive equations are used to constantly update the parameters and but also to simultaneously select the appropriate number of components for each pixel.

We know that a video is essentially a collection of image frames put together and played in a continuous stream. Therefore, it is noticeable that the vehicle changes coordinates in each successive frame and, hence, its location. Also, it can be noted that only the pixels representing

the vehicle will undergo changes in these consecutive frames. Thus, the frame differencing method aims to notice the changes in the pixel and location of the moving vehicle.

Image Thresholding works by assigning one of the two values based on a threshold to the pixels in a grayscale image. If the value of a pixel exceeds a threshold value, then it is assigned a particular value; else, it is given another value. The objective is to remove unwanted highlighted areas by obtaining a resultant binary image.

These traditional background subtraction techniques, while effective in controlled environments, often struggle in real-world road conditions due to the presence of dynamic backgrounds, lighting fluctuations, and partial occlusions. To address these limitations, adaptive algorithms that utilize probabilistic models like the **Gaussian Mixture Model (GMM)** have been introduced. These models allow the system to update the background representation continuously, adapting to scene changes and improving the detection of moving vehicles over time.

Moreover, integrating **Otsu's thresholding** helps in effective segmentation by automatically determining the optimal threshold between foreground and background pixels. When combined with **shadow detection algorithms**, this significantly reduces false positives caused by moving cast shadows, which are common in outdoor surveillance footage.

CHAPTER 2

LITERATURE REVIEW

Rajvardhan Rish, Sofiya Yede, Keshav Kunal, Nutan V Bansode proposed a system which states that the leading cause of deaths in road accidents is due to delay in medical help. This can be prevented by messaging the authorities and emergency contacts too on time. The system consists of GPS, GSM, accelerometer and Arduino. It alerts nearest hospital, police headquarters, family and friends during the time of mishap mainly by detecting changes in accelerometer. The system sends a google map link using GPS module and Arduino. The vehicle sets the flag bit of the Arduino UNO as an accident is identified until it detects abrupt deviation from the threshold values with the help of the measuring system detector. Throughout the accident, the device sets the effective sensitive value for measuring instrument detectors, unless a crash is observed. Once the accident or set bit is detected by the measuring instrument detector, Arduino activates the GSM module, which has a manually saved signal of the accident victim's emergency contact, and sends a pre-stored SMS to that contact.

Though this system works fine, it lacks the detection of rare minor accidents with no casualties. So, it will eventually result in waste of resources and time in the case of minor accidents. Furthermore, it uses Arduino UNO which is less powerful than the recent microcontrollers available in the market. Hence, we decided to only take the system architecture components which would be beneficial to our project in accuracy which are the following: GSM, GPS module, accelerometer.

Mr. S. Kailasam, Mr. Karthiga, Dr. Kartheeban, R.M. Priyadarshani, K Anithadevi states that due to lack of attention, Drowsiness, and drunk driving are the major causes of road accidents, this paper proposes preparing a system to prevent these circumstances. The proposed system herein aims at preventing and controlling accidents by using a Night Vision Camera. This system monitors the driver’s face when the car starts which mainly helps in observing continuously. It uses two functions: One to detect the eye blinking, second is for reading the blinking. Automatic driving and braking systems also combined with a controlling system using python programming. Speed is automatically reduced until the driver becomes alert and returns to consciousness. The proposed system alerts the driver depending on his state, and makes sure that he is not drowsy.

However, if the driver has a medical condition or blinks at an abnormal rate despite not being drowsy, the system will give a false alarm. In the worst case scenario, the driver happens to be in an accident, the system fails to detect the impact and contact the concerned authorities. Lastly the system would constantly consume power and drain the power supply since it monitors the driver continuously. Hence the outcome of not being able to identifying the actual accident scenario made us reject the idea of adding face recognition to our system as it would be costly, power- consuming and inefficient.

T Kalyani, S Monika, B Naresh, Mahendra Vucha have proposed three main components in this system namely vibration sensors, GPS and GSM module. When a vehicle meets with an accident, the vibration sensor will read the impact and Arduino will then compare it with the threshold value set in 6 the program. If the value exceeds the threshold value then GPS will generate the current location and GSM will send the alert message to respective authorities with the help of Arduino. Vibration sensor is used to detect the accident. One of the major gaps with this system is that it doesn't provide any solution to a minor accident situation where there isn't any serious damage or any casualties. So in such cases if the system doesn't provide a solution to this problem this will ultimately result in waste of resources and time. Also this system doesn't provide any medical history of the victim which eventually results in the delay which will ultimately hinder the cause.

Aarya D.S, Athulya C.K, Anas.P, Basil Kuriakose, Jerin Susan Joy, Leena Thomas proposed a system that states the vehicle accidents are one of the most leading causes of fatality. The period between the occurrence of an accident and the dispatch of emergency medical services to the accident site is a critical factor in accident survival rates.

Accident detection and messaging system will be stationed in vehicle itself which will be helpful during the time of accident as hospital, police and emergency contact can be informed immediately. The system is executed using GPS and GSM technology. A vibration sensor detects a collision using piezoelectric effect; which is the ability of certain materials to generate an electric charge when they are under mechanical stress. As soon as the collision is detected the GPS module locates the accident (latitude and longitude) and sends a message to the hospital and the emergency contact using the GSM module. The ambulance arrives to the location which is tracked by the GPS module and hence the victim is treated as soon as possible reducing the help time. In case if there

is a minor accident, the victim can press a switch (button) to prevent the emergency contacts from being alerted. This system comprises of Arduino, GPS, GSM and vibration sensor, which detects the accident and alerts the authorities immediately, it also combats false alarms by using a switch provided for the driver. However, the system does not provide the medical data and history of the victim and hence there could be a delay in the victim's treatment. We shall improvise our system in this scope.

Nagarjuna R Vatti, Prasanna Lakshmi Vatti, Rambabu Vatti, Chandrashekhar Garde have provided some shocking statistics of accident rates in India and also talked about the reasons of increase in road accidents that is caused by improper construction and low maintenance of the roads as well as overcrowding and increasing count of vehicles. Other than this, the youth succumb to their injuries on roads because of rash driving, drunken driving. The pre-existing types of accident prevention systems installed in cars are air bags, ABS. The projected system relies on the thought that the accident is detected by vibration and rotating mechanism sensors once a close review of current systems and literature surveys, and a message is straight away sent to the emergency contact numbers victimization the GSM module beside the situation found by the GPS module. If the vehicle gets any head-on collision the vibrations area unit created. If the vibrations exceed a threshold price they're detected and treated as a heavy accident case. The gyro detector can notice if the vehicle has toppled or atilt by an outsized angle. All told cases, the system can anticipate ten seconds. If button is ironed by the motive force among ten seconds the system considers that accident isn't serious and it resets back to traditional operation. During this system, the guts rate detector can notice the speed of the motive force only if the accident has occurred. By causation this data, the hospitals can get to grasp concerning the condition of the motive force and consequently they'll react to assist the motive force. They need to conduct experiments by implementing the system in a very toy automobile and determine that the system is functioning properly. This is by far the best and most efficient as well as cost friendly approach

Wen-Kai Tai, Hao-Cheng Wang, Cheng-Yu Chiang, Chin-Yueh Chien, Kevin Lai, and TsengChang Huang have stated that, traffic accidents are happening every day, still there is not an efficient way to investigate, record and measure such parameters with precision. The authors have come up with a system to extend the exactness and viability of knowledge retrieved in traffic accident investigation, shorten the investigation time, ensure the safety of investigators, decrease

social value, and improve the general service quality of traffic accident investigations. The planned system contains four phases: prepared, Record, Measure, Archive, and Edit phases. The intent of this system is merging the practicality of drones associated mobile devices to form a system that may collect and record proof each fleetly and expeditiously. If the investigator must sketch the accident diagram, he will use the functions of Accident Diagram App (ADA). This technique provides correct and correct readings, with the assistance of overhead pictures enamored by the assistance of drones; the gap measured is correct with average error of +-5cm.This paper provides an excellent system which could be able to detect and investigate traffic accidents happening in the city, and help save time. It uses drones to investigate the accident spot which collects overhead photos and provides a function ADA to do so efficiently. However, if the accident is very fatal, this system would not inform any hospital or such designated authorities.

Miss. Priyanka M. Sankpal, Prof. P. P. More have implied that this system aims at preventing accidents by detecting the driver's state i.e., if the driver is feeling drowsy or is not in the condition to drive. IR reflective obstacle sensors are used to detect the position of the eyes. If eyes are opened then the sensor gives the low signal to the controller, and if the eyes are closed till 4 sec then the sensor gives high signal to the controller. The PIC controller receives input from the sensor, and the controller gets the high or low signal and processes it to make the buzzer ON or OFF, depending on the buzzer the LCD displays "Driver Slept", if the buzzer is ON. Hence the driver is notified and a major mishap could be prevented from happening. Since drowsiness is a major cause of accidents, this system provides a very efficient way to reduce such incidents. Even if the system provides a way to prevent accidents from happening due to drowsiness, there could be other causes which may not be detected and since the system does not provide detecting the damage and notifying the emergency contacts, this could be a major drawback.

Usman Khalil, Tariq Javid, and Adnan Nasir have proposed a system for sending accident messages to emergency services using the Vehicular AD-Hoc Network (VANET). VANET assists these services in finding the best route to their destination by using the ABEONA algorithm, a traffic signal module. This function will significantly reduce the time it takes for an ambulance to arrive at the accident scene. VANET, eCALL mobile application, and ultrasonic sensor are among the system's features. Accident detection using VANET is the most effective approach of all because it not only detects an accident, but also provides the ambulance with the best path to get

to the accident site as quickly as possible. However, the system requires the machinery to be installed in both the vehicle and the ambulance, which is not always feasible. As a result, we determined that VANET will not be implemented in our scheme. In addition, if a cell phone is damaged in an accident, the application will be useless, rendering this solution useless. OEMs do not instal such detection systems in most low-end cars in developed countries.

Barış Guksa and Burcu Erkmen have proposed a system which would not only inform about the accident, but also detect if the driver is feeling sleepy and is not in a condition to drive, hence decreasing the possibility of an accident. To observe the driver's standing, the projected device uses a smartphone, which can do this with its front camera. The user is asked to require a photograph once the program is opened. Within the photograph, the situation of the face is set. Once the situation of the face is set, solely the face is cut and it's saved as a replacement photograph by the system. Then, the quantity of gap and shutting eyes in a very minute is saved within the system. If additional or less the quantity of eye gap and shutting in a very minute, that is completely different from the conventional, is detected, the motive force controls himself by causing him a voiced warning message speech “are you sleepy?” is aimed. The smartphone ought to have a front camera; associate ALS (Ambient light-weight Sensor), a gyro detector associated with a measuring system and a recipient IC (Integrated Circuit), which is required for GPS signals to be processed within the phone. The system proposed by the author could be efficient to prevent accidents from happening. As this system uses smartphone’s sensors for detection, it is mandatory that the driver has a smartphone, since smartphones are very common these days, they could be useful in such a system, but if the driver forgets to bring his smartphone, nonetheless does not carry a smartphone, this could render the whole system inefficient.

Based on data obtained from mobile sensors, this algorithm confirms an accident. This is a less expensive choice. Smartphone, GPS receiver, accelerometer, magnetometer, and gyroscope are the components used to implement the device proposed by **Harit S., Ravi K.R., Archana K**. It uses a collision index and a crash detection algorithm. A few common crash detection systems are also smartphone-based and use MEMS sensor data collected in real-time while driving. However, data obtained from smartphone builtin sensors may contain significant noise, and therefore most current solutions may reduce detection accuracy and lead to false positives. To address the limitations of using a smartphone as a data capture source, the proposed solution employs collision

index measurement, known false positive identification, and removal, as well as additional confirmation with position data from the smartphone. However, since this scheme does not have the victim's medical history, care could be delayed. This solution will not be able to offer what it promises in incidents where the smartphone is totally destroyed. As a result, a smartphone sensor-based application might not be able to match the precision of expensive in-vehicle systems fitted with high-end accident detection technology, as they have an obvious advantage of providing direct access to the vehicle.

sPrashant Kapri, Shubham Patane, Arul Shalom proposed a system which states that an accident might occur at an isolated area where humans are absent to report any mishap. Inbuilt hardware modules in luxury vehicles have recently been developed to detect and report accidents. Unfortunately, such devices are both costly and immobile.

They proposed a system in which the accidents are detected with the help of in-built sensors in the smartphone and physical context information. The system comprises of the a server and the software. The software acts as a sensing device as well as an interface for the third-party observers to contribute 3 information to the accident report. The software also uses Google maps on the smartphone for mapping purposes. The map will allow other drivers to plan their route intelligently around an accident, hereby reducing the congestion around that road via live update on map about accident. The client software can access the data from phone database such as a contact list to inform emergency contacts. The software will be installed in smartphone and will perform the task of detecting the accident and also exchanging information with server. The detection will completely depend on the sensors built in the phone which includes accelerometer sensor, GPS receiver, and microphone.

Although in-vehicle detection system provides essential information very quickly but unavailability of this system are restricted by their non- portability and costs, whereas smartphone provides a promising platform with same sensors at cheap price and portability benefits. The functionality of a smartphone will outperform that of a traditional in-vehicle accident detection device. Whilst the idea of adding a smartphone was great but the major drawback of this would be if the battery of the smartphone ran out or if Bluetooth on the smartphone was disabled during the time of the accident, then this system fails. Hence, we decided to not implement application software as it could potentially fail in major accidents which are the main reason for the

implementation of this project. Furthermore, if the device itself is damaged in a major accident, then there would be no detection making this implementation useless.

Adnan Bin Faiz, Ahmed Imteaj, Mahfuzulhoq Chowdhury have stated that, these days, smartphones have become an everyday requirement and this paper therefore proposes a device that uses 7 a smartphone to detect whether an accident using the sensors of the smartphone has occurred. The proposed system uses the GPS receiver of the phone to detect a sudden rapid change in deceleration that happens at the time of an accident. It also reads the change in pressure from the pressure sensor and the angle of tilt from an accelerometer sensor in a Smartphone. After detecting these three conditions, the android app would send an alert to the emergency contact. To prevent any false alarm, the system is provided with a switch which, if pressed, would prevent sending an alert to the emergency contact. This could be useful in case of wrong readings in the sensors, or if the accident is a minor one and not fatal. This device would require a constant internet connection in order to work smoothly. The system used in this paper can give systematic results and ensure the safety of the driver using smartphones which have become a daily necessity in today's life. Since the smartphone's sensors could provide false data sometimes, this system also comes with a switch, which could be used by the driver in case of wrong readings. Considering this system requires a constant internet connection, which may not be possible in remote areas, this may be a drawback in its working. In such a situation, the system could fail.

According to Manuel Fogue, Piedad Garrida, and Francisco J. Martinez vehicular networks would play a larger role in the Intelligent Transportation System sector, reducing road fatalities. The majority of its uses, such as traffic control, fleet management, and navigation. These would be dependent on vehicle-to-roadside infrastructure communication, or even vehicle-to-vehicle communication. The introduction of sensoring technologies on-board vehicles, as well as peer-to-peer mobile communication between vehicles, are expected to result in major safety improvements in the near future. Based on the concepts of data mining and information inference, the proposed system depicts a smart system that will automatically detect road accidents, alert them through vehicular networks, and estimate their severity. On-Board unit (OBU) sensors, data acquisition unit, processing unit, and wireless interfaces, V2V and V2I communications, Control Unit (CU), Information Discovery in Databases are among the libraries and applications used to implement

the system. Centered on V2V and V2I communications, the authors developed and implemented a prototype for automatic accident and assistance.

After a thorough selection of relevant characteristics, the findings revealed that vehicle speed is a critical factor in front accidents, but in side and rear-end collisions, the type of vehicle involved and the speed of the struck vehicle are more significant than speed itself. The studied classification algorithms do not show significant differences, but they do show that if we can identify accidents based on the types of impacts, we can significantly improve the system's accuracy, particularly in front crashes where the vehicle is typically the striking one. We disregarded use of KDD algorithm as it didn't provide significant difference and was inefficient as well as complex to implement. Vehicular networks will play an increasing role in the Intelligent Transportation Systems (ITS) area and hence reduce road fatalities. Most ITS applications, such as road safety, fleet management, and navigation. These will depend on communication between the vehicle and the roadside infrastructure (V2I), or even directly between vehicles (V2V). The integration of sensoring capabilities on-board of vehicles, along with peer-to-peer mobile communication among vehicles, forecast significant improvements in terms of safety in the near future.

Hossam M. Sherif, M.Amer Shedid, and Samah A. Senbel suggested a solution in which three key components, namely the Node algorithm, Router algorithm, and Coordinator algorithm, are required to facilitate the accident report. During a road accident, the Real Time Traffic Accident Detection System (RTTADS) can intelligently notify the accident site through a wireless interface. It will also notify the appropriate authorities. It will not only tell you how many people have died, but also what kind of emergency services are needed. RFID sensors, an RF module, a wireless module, a crash sensor, a rollover sensor, a fire alarm sensor, a weight sensor, and a microcontroller are among the system's hardware components. The proposed system would detect an accident in real time and send information to the supervisory programme about the accident site, vehicle speed (before the impact sensors are triggered), number of passengers in the vehicle, crash sensors that have been activated (front, back, right side, and left side), rollover sensor status, and fire alarm sensor status. The rule-based system scans the estimated number of ambulance cars required for injured patients, as well as whether or not a 5 fire truck is required. We liked the concept of using three separate algorithms to organise the process of sending a message, but it can

be easily implemented using a GPS and GSM module, and false alarms can be avoided using updated technology like the fingerprint sensor used in our proposed system.

Md. Syedul Amin, Jubayer Jalil, M. B. I. Reaz have stated that speed is one of the most important and basic risk factors in driving. Not only does it affect the severity of a crash, but it also raises the likelihood of a crash. If emergency service could get accident reports and reach it in time, more lives could have been saved. The GPS tracks the speed of a vehicle and, through a microcontroller chip, compares it with the previous speed every second. Whenever the speed is below the stated speed, an accident is believed to have occurred. Through using the GSM network, the device will then submit the accident position obtained from the GPS along with the time and speed. They propose to use the ability of a GPS receiver to track a vehicle's speed and detect an accident based on the speed monitored and send the position and time of the accident from the GPS data processed by a microcontroller to the Warning Service Center using the GSM network. Kinetic energy is converted into destructive forces that cause damage to the passengers as well as to the vehicle when an accident occurs. The system uses GPS, GSM and a microcontroller unit for hardware and the software components are accident detection algorithm, speed measurement, detection and reporting by MCU, and finally the data interpretation unit. It will also show the previous speed of the vehicle before committing the accident. This data will help the Alert Service Center to assess the severity of the accident basing on the speed and also it can initiate a voice call. The detailed algorithm sure reaps a good result but it is time consuming and hence, only necessary steps can be initiated keeping speed in mind.

Ms. Sharmila S. Gaikwad proposes an agent-based framework for the extremely complex and variable sense of healthcare emergency decision support. It emphasizes the value of using mobile agents to help the real-time deployment of an emergency service, rather than just hypotheses. A mobile agent is an autonomous program capable of transporting itself entirely under its own control between network nodes, carrying with it the data and execution status required to resume execution at the destination host from where it left off on the original host. Hence this agent makes a decision on when and where to move and how to go about the execution without consulting the user repeatedly. Along with mobility, they are also capable of performing dynamic and intelligent inference tasks during their execution which makes them a great approach not only in situations of urgent medical help but also in other areas like military to educational.

CHAPTER 3

PROBLEM STATEMENT

The exponential rise in vehicular traffic on roads, highways, and expressways has led to an increased occurrence of road accidents. A significant number of these accidents go unnoticed or are reported late, resulting in delayed emergency response, loss of life, and property damage. Existing traffic surveillance systems often lack real-time detection and notification capabilities, especially under challenging environmental conditions such as poor lighting, rain, or heavy traffic. Furthermore, many systems rely on traditional vehicle detection methods that are either sensor-based or non-adaptive to dynamic scenarios, leading to high false detection rates and limited effectiveness.

There is a pressing need for a smart, efficient, and real-time accident detection system that can not only detect accidents accurately but also notify the nearest emergency services instantly with the precise location and evidence of the incident. The proposed system aims to address these challenges by leveraging video surveillance, adaptive computer vision techniques, and real-time communication technologies to act as a "Guardian Angel" on the road.

In addition to delayed response times, another major concern with existing systems is their lack of adaptability to diverse environmental conditions such as low visibility, night-time scenarios, or adverse weather (rain, fog, snow). These limitations not only reduce the reliability of detection but also hinder the deployment of such systems in remote or high-risk areas. The absence of an integrated framework that combines intelligent detection, accurate tracking, environmental adaptability, and real-time alerting restricts the potential for scalable and effective road safety solutions. Hence, there is a critical need for a robust, environment-aware, and automated system that can act instantly as a digital first responder—minimizing human dependency and maximizing the chances of timely recovery and life-saving interventions.

CHAPTER 4

EXISTING SYSTEM

Current vehicle detection and accident monitoring systems mainly fall into two categories: **hardware sensor-based solutions** and **video-based surveillance systems**.

1. **Sensor-Based Systems**:These include inductive loop detectors, infrared sensors, and radar-based systems installed on roadways. While they provide reliable vehicle count and speed detection under controlled conditions, they have several limitations:
    - High installation and maintenance cost.
    - Limited coverage and inflexibility to adapt to changing road layouts.
    - No visual confirmation or evidence of incidents.
    - Inability to distinguish between types of incidents or vehicle behaviors.
2. **Video-Based Systems**: Surveillance cameras are used for monitoring traffic flow and detecting vehicle movement. Traditional computer vision methods, such as static **background subtraction**, are applied to detect moving objects. These systems offer the advantage of non-intrusiveness and ease of deployment but suffer from:
    - Sensitivity to lighting changes, shadows, and weather conditions.
    - Difficulty in separating foreground vehicles from complex backgrounds.
    - Challenges in accurate tracking due to occlusion or vehicle clustering.
    - Lack of real-time alert and location-based notification mechanisms

Despite improvements in video analytics, most existing systems do not provide an integrated end-to-end solution for **real-time accident detection, location capture, and automated notification** to emergency authorities. This creates a gap between surveillance and immediate actionable response—one that this project aims to bridge.

CHAPTER 5

PROPOSED SYSTEM

3.1 Overview

The proposed framework is designed to function as an automated accident detection and recovery notification system using real-time video surveillance. It is capable of identifying vehicular accidents through intelligent analysis of object motion patterns and behavior within video frames. The system is developed to be modular and scalable, making it suitable for deployment in smart cities, highways, toll plazas, and high-risk zones.

Unlike traditional systems that rely solely on sensors or manual surveillance, our framework combines machine learning with computer vision to detect critical incidents with high accuracy. Once an abnormal activity or accident is detected, the system triggers a series of automated responses—such as location tracking, image capturing, and alert notifications—to ensure that help is dispatched with minimal delay.

This proactive approach reduces the dependency on human monitoring, enhances situational awareness, and improves emergency response effectiveness. Additionally, the system adapts to different environmental conditions, which ensures reliability across various terrains, lighting scenarios, and weather situations.

3.2 Axis Bounding Box for Object Detection

The Axis-Aligned Bounding Box (AABB) approach plays a crucial role in detecting and tracking vehicles within video frames. It helps isolate each moving vehicle by enclosing it within a rectangular frame whose edges are aligned to the x and y axes of the video frame. This method is computationally efficient and helps in determining the object's location, width, and height over time.

Each bounding box is assigned a unique identifier, enabling the system to track multiple vehicles simultaneously. The trajectory, speed, and positional changes of these bounding boxes are

continuously monitored. If a bounding box abruptly stops or overlaps with another in a manner inconsistent with normal traffic behavior, it flags the possibility of an accident.

In cases of collision, the system observes the merging of two or more bounding boxes followed by stillness or disappearance—indicating vehicle impact and possible breakdown. This triggers the alert module for further investigation. This technique provides a reliable structure for object tracking and simplifies the complexity of analyzing motion in a busy road environment.

3.3 Centroid-Based GMM Algorithm for Surveillance

The Gaussian Mixture Model (GMM) is a probabilistic model that identifies the dynamic foreground from a stable background. In surveillance video, the background (road, trees, buildings) remains mostly static, while vehicles represent the foreground objects. The GMM continuously updates the background model to accommodate changes in lighting and shadows, making it highly effective for outdoor environments.

For each detected object, the **centroid** (geometric center) is calculated and tracked across successive frames. A sudden change in the centroid's path—like a rapid stop or erratic movement—can indicate abnormal behavior. This is especially useful in identifying sudden braking, skidding, or impact events.

By integrating centroid movement data with time-based thresholds, the system differentiates between regular slow traffic and actual accident scenarios. Moreover, the algorithm is fine-tuned to ignore noise like small debris, pedestrians, or shadows, reducing the rate of false positives.

The combination of GMM and centroid tracking provides a more stable and adaptive system capable of handling high-volume traffic data in real-time, under varying environmental conditions.

CHAPTER 6

METHODOLOGY

MODULE-1

INPUT:

- The video is input to the system.
- The collected video is read through opencv library.
- The system involves capturing of frames from the video.
- The input is from the cctv install in highway road or any road where there is a necessity to reduce the accident.
- In real time mostly the video is from the cctv.

MODULE-2

BACKGROUND LEARNING MODULE

- This is the second module in the system whose main purpose is to learn about the background in a sense that how it is different from the foreground.
- Furthermore as proposed system works on a video feed, this module extracts the frames from it and learns about the background. In a traffic scene captured with a static camera installed on the road side, the moving objects can be considered as the foreground and static objects as the background.
- MOG algorithms are used to learn about the background using the above mentioned technique.
- It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It uses a method to model each background pixel by a mixture of K Gaussian distributions (K = 3 to 5).

- The weights of the mixture represent the time proportions that those colours stay in the scene.
- The probable background colours are the ones which stay longer and more static.
- While coding, we need to create a background object using the function, cv2.createBackgroundSubtractorMOG().
- It has some optional parameters like length of history, number of gaussian mixtures, threshold etc.
- It is all set to some default values. Then inside the video loop, use backgroundsubtractor.apply() method to get the foreground mask.

MODULE-3

FOREGROUND EXTRACTION MODULE

- This module consists of three steps, background subtraction, image enhancement and foreground extraction. Background is subtracted so that foreground objects are visible. This is done usually by static pixels of static objects to binary 0.
- After background subtraction image enhancement techniques such as noise filtering, dilation and erosion are used to get proper contours of the foreground objects. The final result obtained from this module is the foreground.

MODULE-4

ACCIDENT DETECTION

- The third and the last module in the proposed system is accidenr detection.
- After applying foreground extraction module, proper contours are acquired.
- Features of these contours such as centroid, aspect ratio, area, size and solidity are extracted and are used for the classification of the vehicles.
- Bounding box is introduced around the foreground.

- The collision of two bounding box in respect to the position is calculated.
- From this the frame is classified as whether it is accident or normal vehicle movement.

MODULE-4

ALARM ALERT

- The first step would be to import the required packages into our Python code to use them in building the alarm.
- add some alarm tunes to the folder.
- This alarm helps the surrounding members to alert and helps to recover from the incident quickly.

MODULE-5

HOSPITAL MAIL ALERT

- The location is noted using geocoder and with the help of smtp the mail is sended to the particular hospital mail or rescue station.
- Geocoding is the process of **transforming a description of a location—such as a pair of coordinates, an address, or a name of a place—to a location on the earth's surface**.
- SMTP stands for Simple Mail Transfer Protocol.
- To allocate proper resource and to take proper legal action in correct time this module helps a lot.

SMTP

- It is used for sending messages to other computer users based on e-mail addresses.
- It provides a mail exchange between users on the same or different computers, and it also supports:

- It can send a single message to one or more recipients.
- Sending message can include text, voice, video or graphics.
- It can also send the messages on networks outside the internet.

- The main purpose of SMTP is used to set up communication rules between servers. The servers have a way of identifying themselves and announcing what kind of communication they are trying to perform. They also have a way of handling the errors such as incorrect email address. For example, if the recipient address is wrong, then receiving server reply with an error message of some kind.

GMM:

GMM is used to reconstruct the mean image after block processing. Specifically, the Gaussian distribution in GMM is initialized, and then k Gaussian distributions in the existing GMM are used to detect whether each pixel in the current image frame matches them. The pixel is considered to match the Gaussian distribution if the new pixel and the Gaussian distribution of GMM

$$\left|d_{ij}^{m,k}\right| \le \rho\sigma_{ij}^{m-1,k}$$

$$d_{ij}^{m,k} = G_{ij}^{m} - M_{ij}^{m-1,k}$$

where d is the absolute distance between the new pixel and the mean value of the k-th Gaussian distribution model, is the deviation threshold, m represents the standard deviation of the current image frame pixel, m G represents the pixel of the current image frame, m k 1, M represents the mean value of the current image frame pixel, i represents the number of rows of the current image frame, j represents the number of columns of the current image frame, k represents the number of Gaussian distributions, m represents the number of image frames.

If the new pixel matches the k-th Gaussian distribution model, the parameters of the Gaussian distribution model can be updated as follows:

$$w_{ij}^{m,k} = (1-\alpha)w_{ij}^{m-1,k} + \alpha$$

$$M_{ij}^{m,k} = (1-p)M_{ij}^{m-1,k} + pG_{ij}^{m}$$

$$\sigma_{ij}^{m,k} = \sqrt{(1-p)(\sigma_{ij}^{m-1,k})^2 + p\left(d_{ij}^{m,k}\right)^2}$$

Where represents the learning rate, which determines the updating speed of the background, m k, w represents the weight of pixels of the current image frame, p represents the update rate, the relationship of which with the other parameters can be expressed as: , m k p w . If the new pixel does not match any Gaussian distribution, a new Gaussian distribution is created to replace the Gaussian distribution with the smallest existing weight. The average of the newly created Gaussian distribution will be the average of the currently observed pixels, the standard deviation will be set to the maximum value of the initialization, and the weight will be set to the minimum value of the initialization. The weight of the other Gaussian distributions will be updated to the ones determined by following equation:

$$w_{ij}^{m,k} = (1-\alpha)w_{ij}^{m-1,k}$$

Based on, the priority of the k Gaussian distributions is determined from large to small. The higher the priority is, the more stable the Gaussian distribution is, and the closer it is to the background, so the first B Gaussian distributions are used to establish the background model.

$$B = \arg\min_b \langle \sum_{k=1}^{b} w_{ij}^{m,k} > \tau \rangle$$

Finding Contours

In this method, the shape of the vehicle is identified by setting boundaries around regions of interest (the vehicle) in the image frames. In the next steps, we can get the coordinates of these contours in the image frame and locate the vehicle.

Utilizing these methods selectively, we can easily locate the movements of vehicles in the video or image frame and detect them.

Image Thresholding

This method works by assigning one of the two values based on a threshold to the pixels in a grayscale image. If the value of a pixel exceeds a threshold value, then it is assigned a particular value; else, it is given another value. The objective is to remove unwanted highlighted areas by obtaining a resultant binary image.

Frame Differencing

We know that a video is essentially a collection of image frames put together and played in a continuous stream. Therefore, it is noticeable that the vehicle changes co-ordinates in each successive frame and, hence, its location. Also, it can be noted that only the pixels representing the vehicle will undergo changes in these consecutive frames. Thus, the frame differencing method aims to notice the changes in the pixel and location of the moving vehicle.

Architecture:

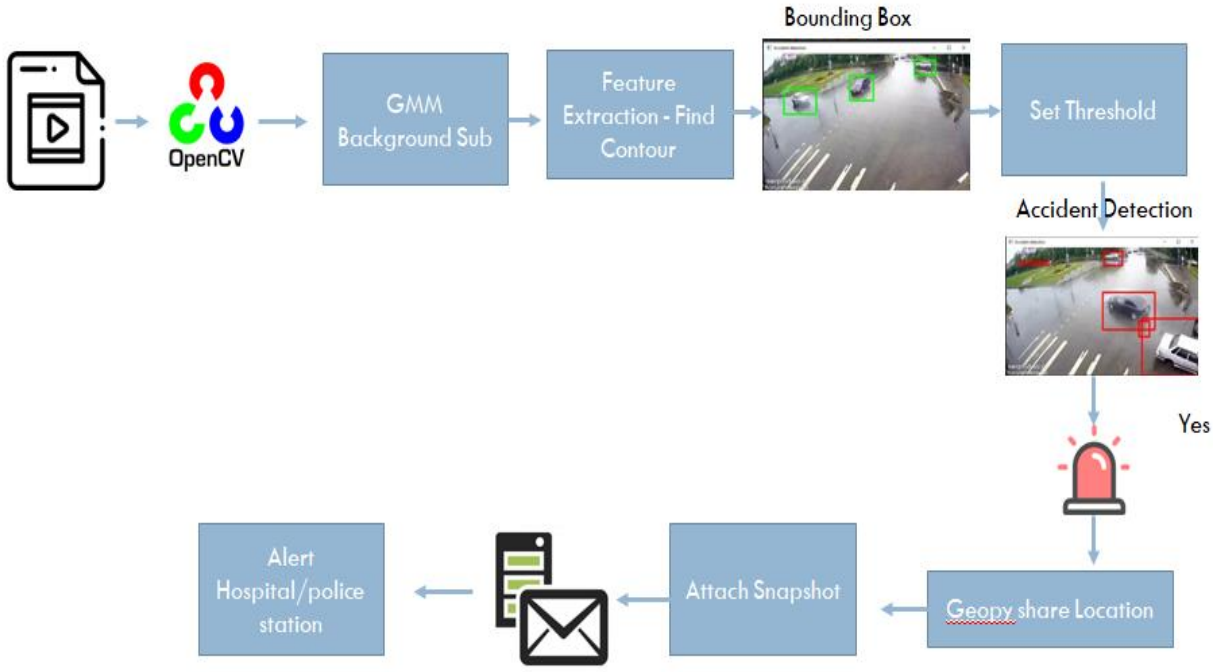

Fig 6.1: Architecture

Flow chart:

Fig 6.2: Flow chart

UML Diagram:

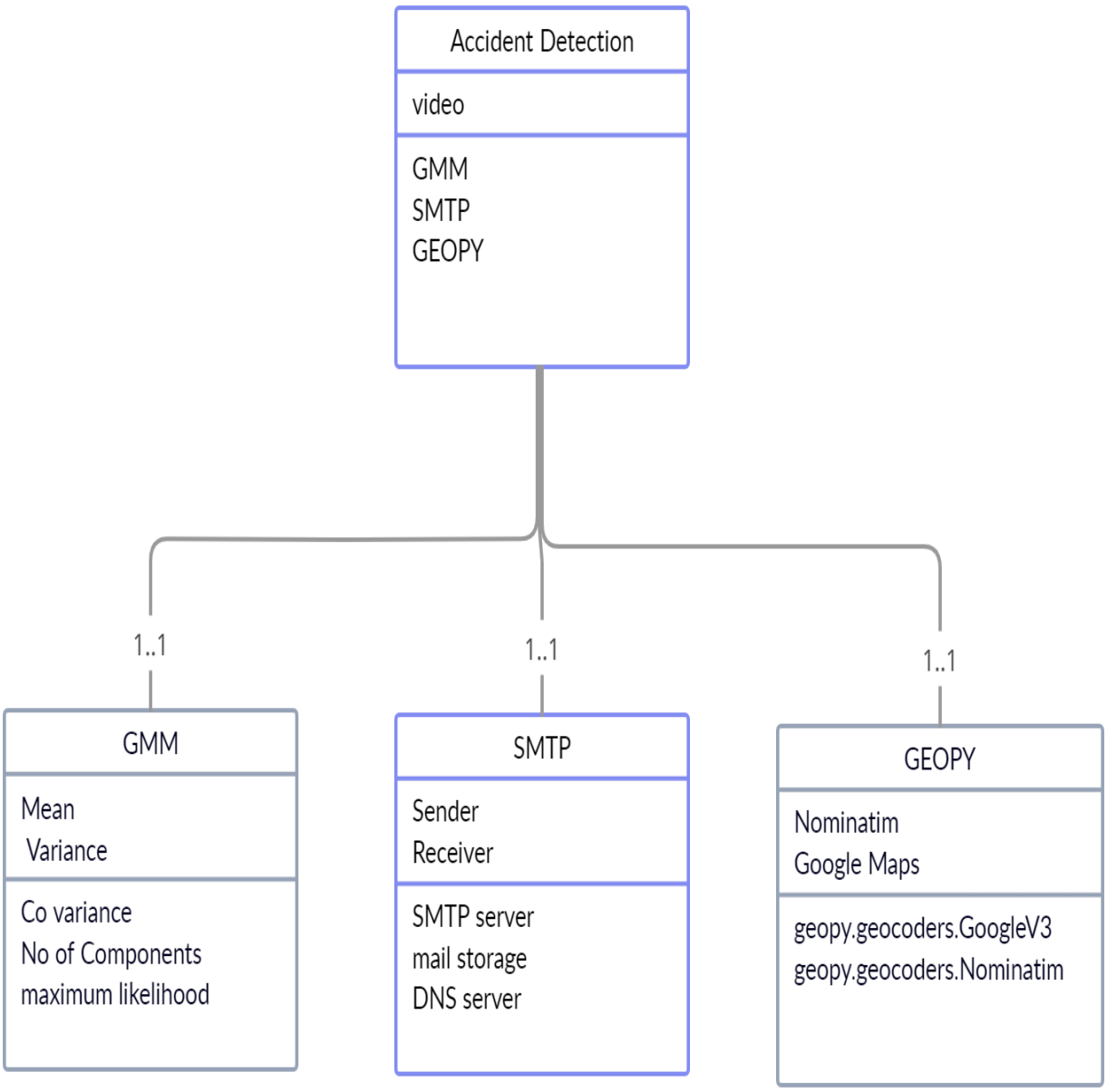

Fig 6.3:UML diagram

Usecase:

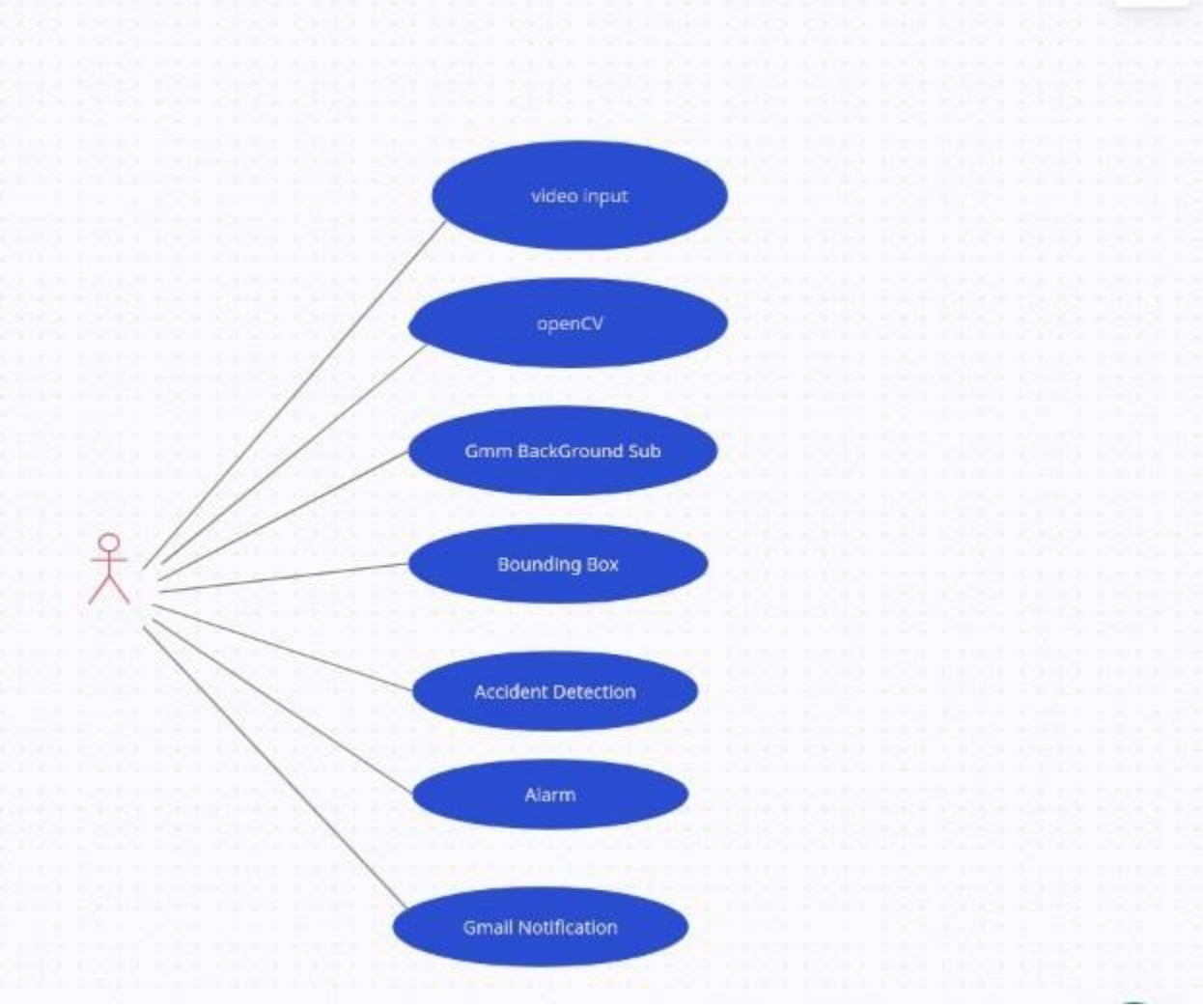

Fig 6.4: Use case

CHAPTER 7

REQUIREMENT SPECIFICATION

IMPLEMENTATION:

HOW TO INSTALL PYTHON IDE

Below is a step by step process on how to download and install Python on Windows:

Step 1) To download and install Python, visit the official website of Python https://www.python.org/downloads/ and choose your version. We have chosen Python version 3.6.3

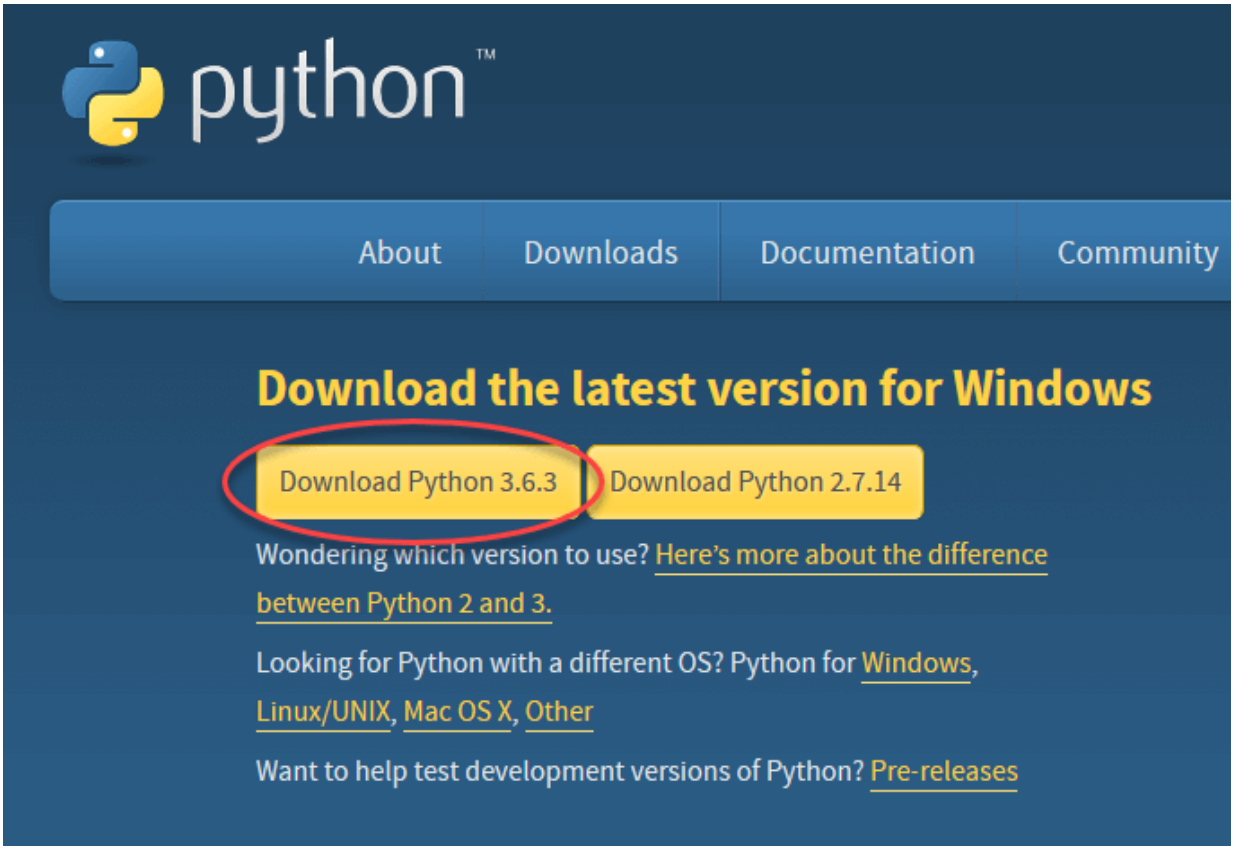

Fig 7.1:Installation of python

Step 2) Once the download is completed, run the .exe file to install Python. Now click on Install Now.

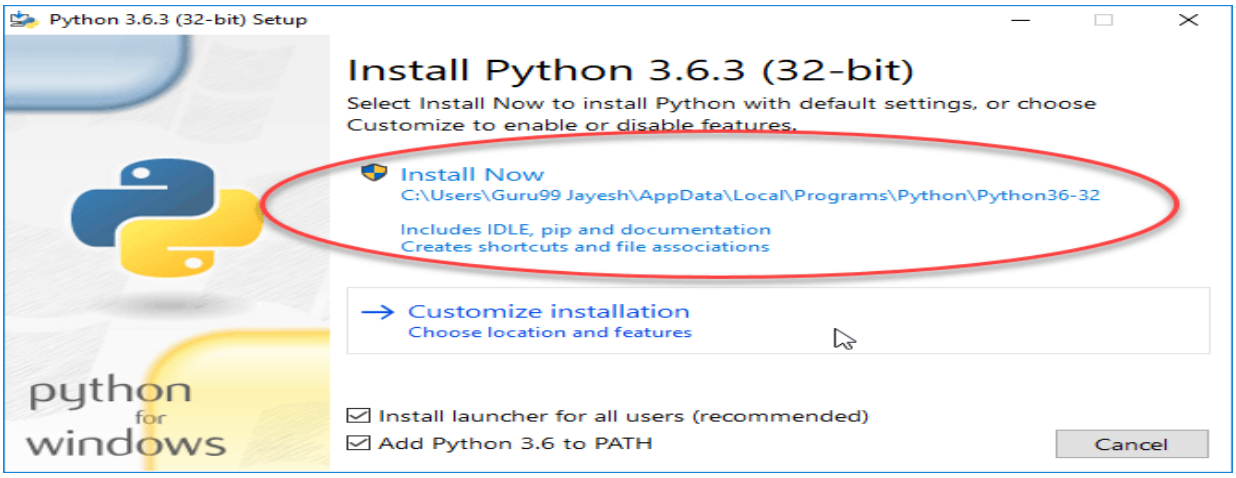

Fig 7.2: exe file link

Step 3) You can see Python installing at this point.

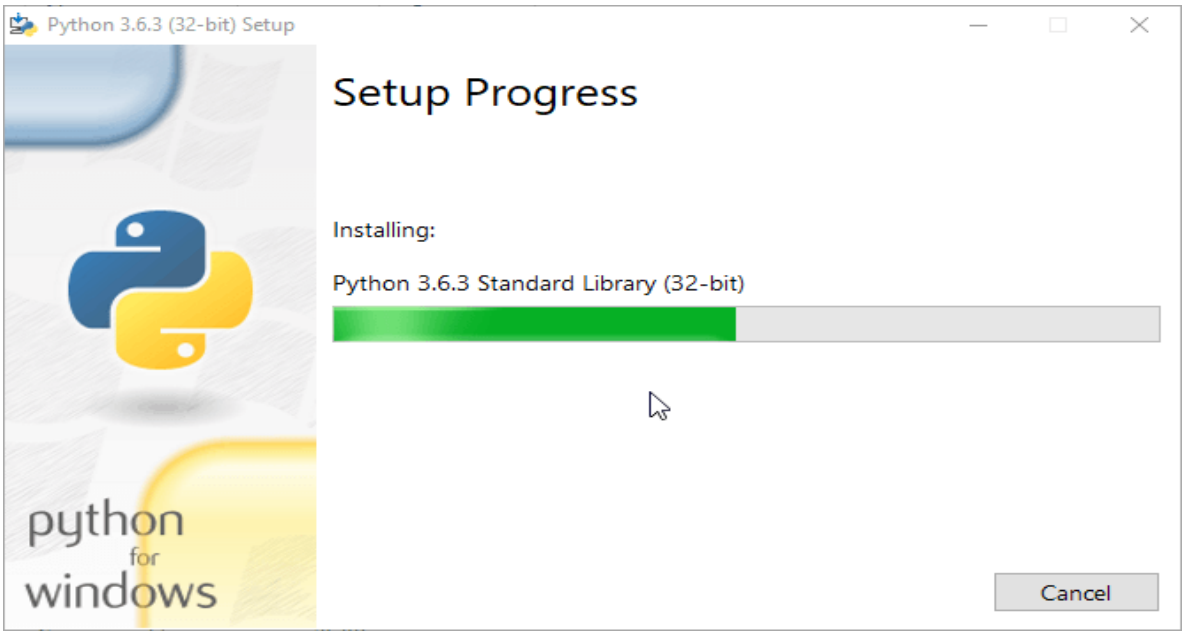

Fig 7.3: installation point

Step 4) When it finishes, you can see a screen that says the Setup was successful. Now click on "Close".

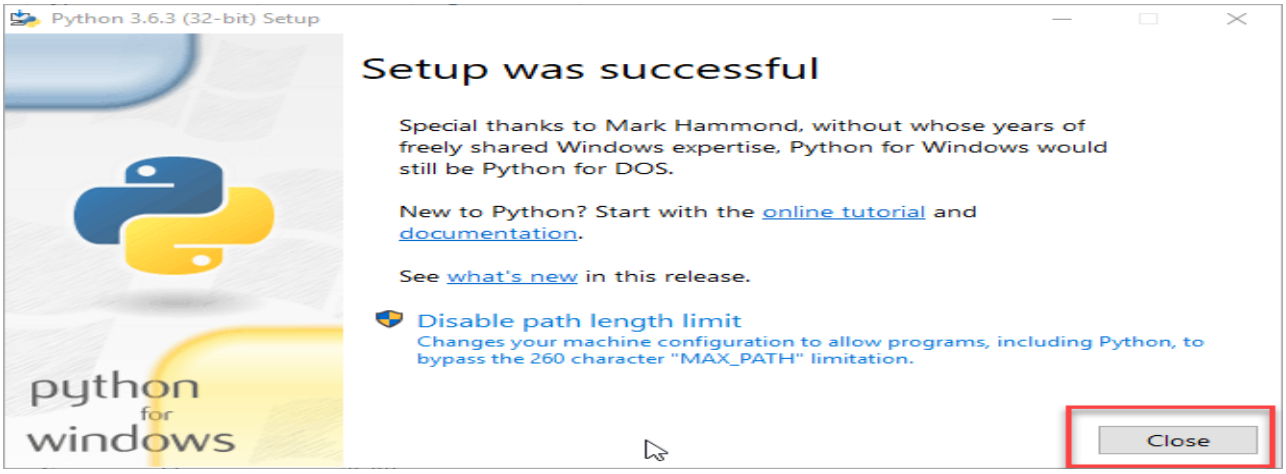

Fig 7.4: Successfully installed

HOW TO INSTALL PYCHARM

Here is a step by step process on how to download and install Pycharm IDE on Windows:

Step 1) To download PyCharm visit the website https://www.jetbrains.com/pycharm/download/ and Click the "DOWNLOAD" link under the Community Section.

Fig 7.5: Download Pycharm

Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click "Next".

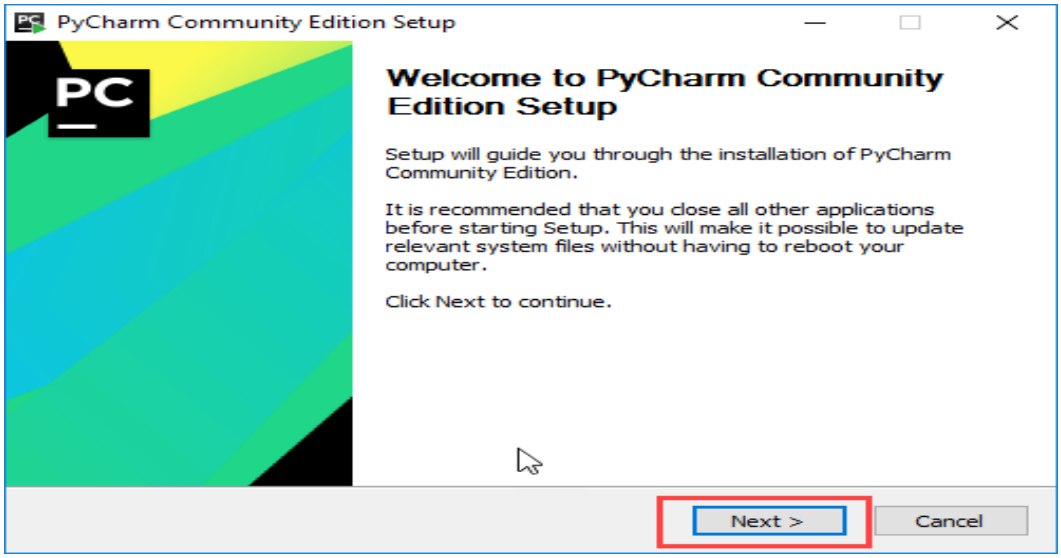

Fig 7.6: exe file

Step 3) On the next screen, Change the installation path if required. Click "Next".

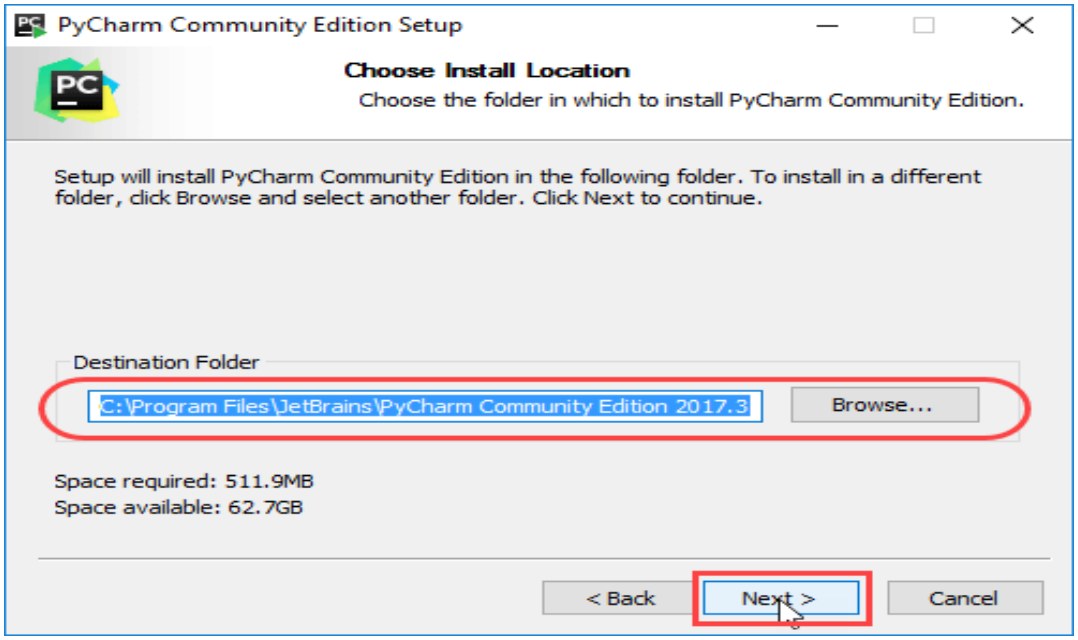

Fig 7.7:Location

Step 4) On the next screen, you can create a desktop shortcut if you want and click on “Next”.

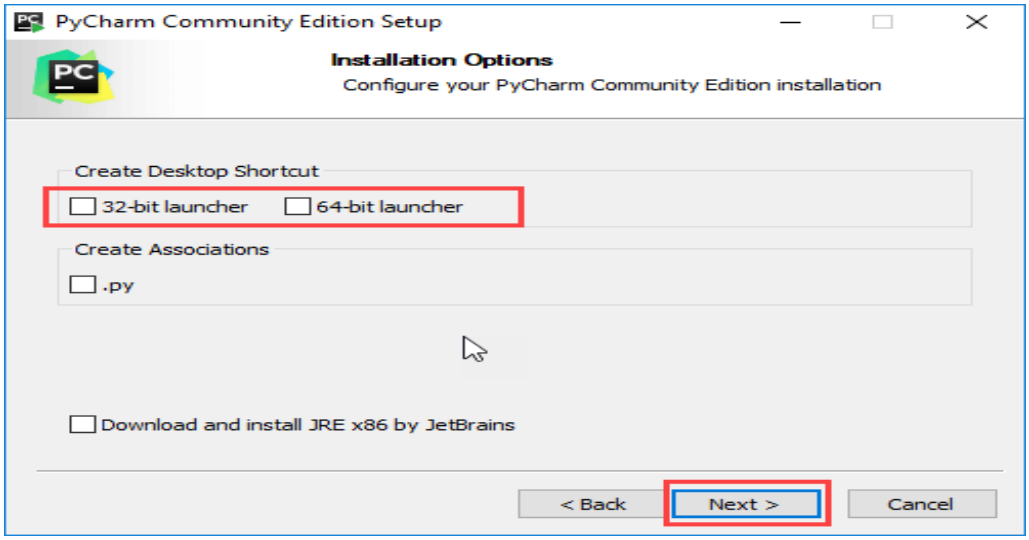

Fig 7.8:Shorcut

Step 5) Choose the start menu folder. Keep selected JetBrains and click on “Install”.

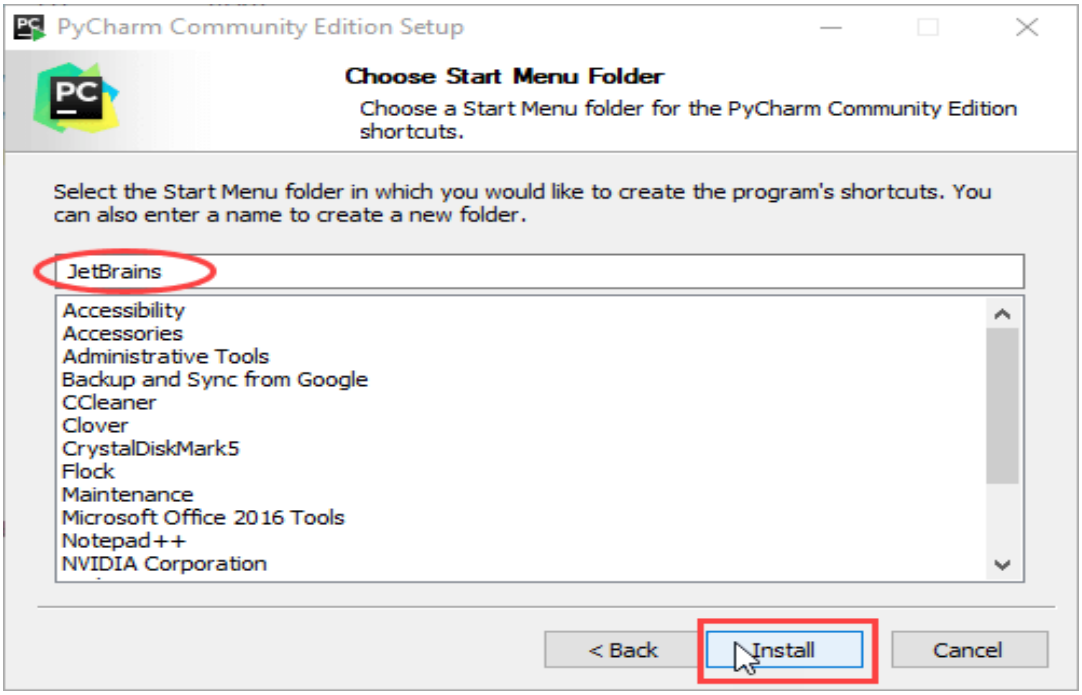

Fig 7.9: Menu

Step 6) Wait for the installation to finish.

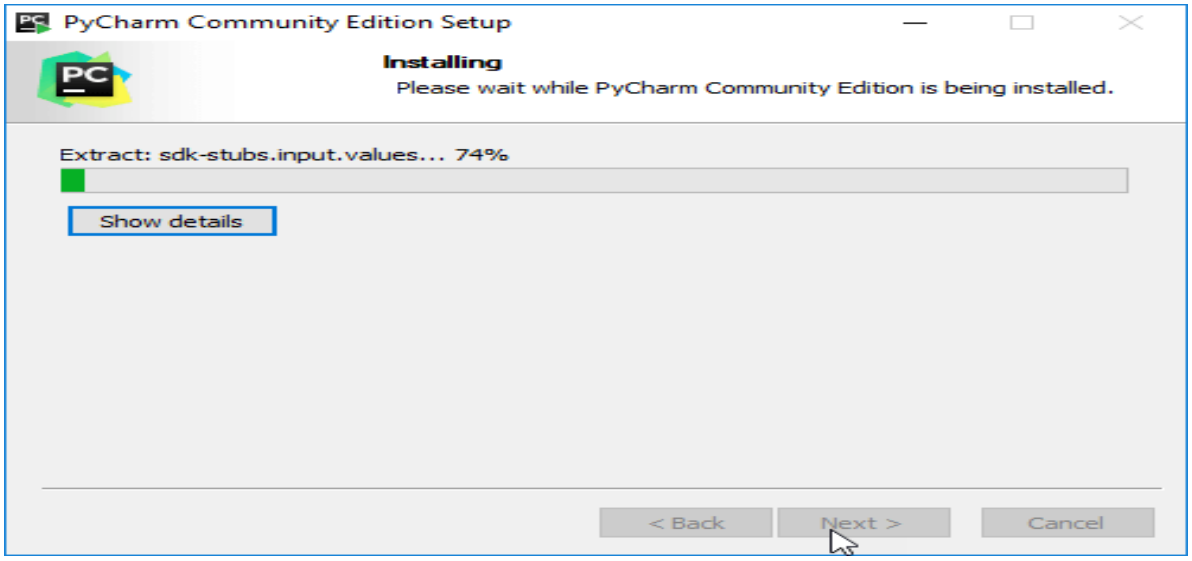

Fig 7.10:Installation finished

Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the "Run PyCharm Community Edition" box first and click "Finish".

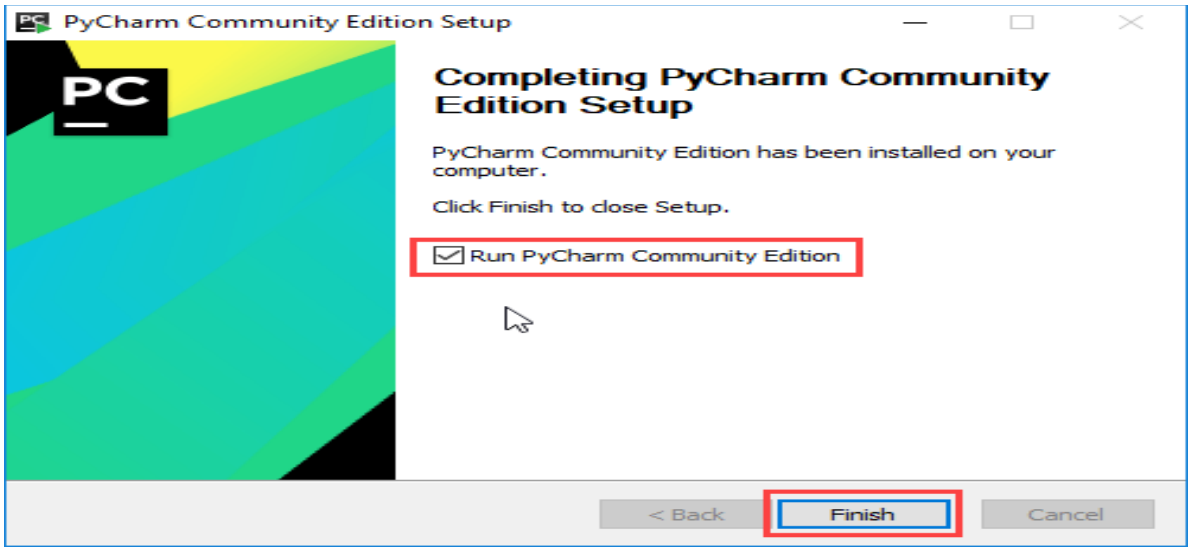

Fig 7.11: Pycharm installed

Step 8) After you click on “Finish,” the Following screen will appear.

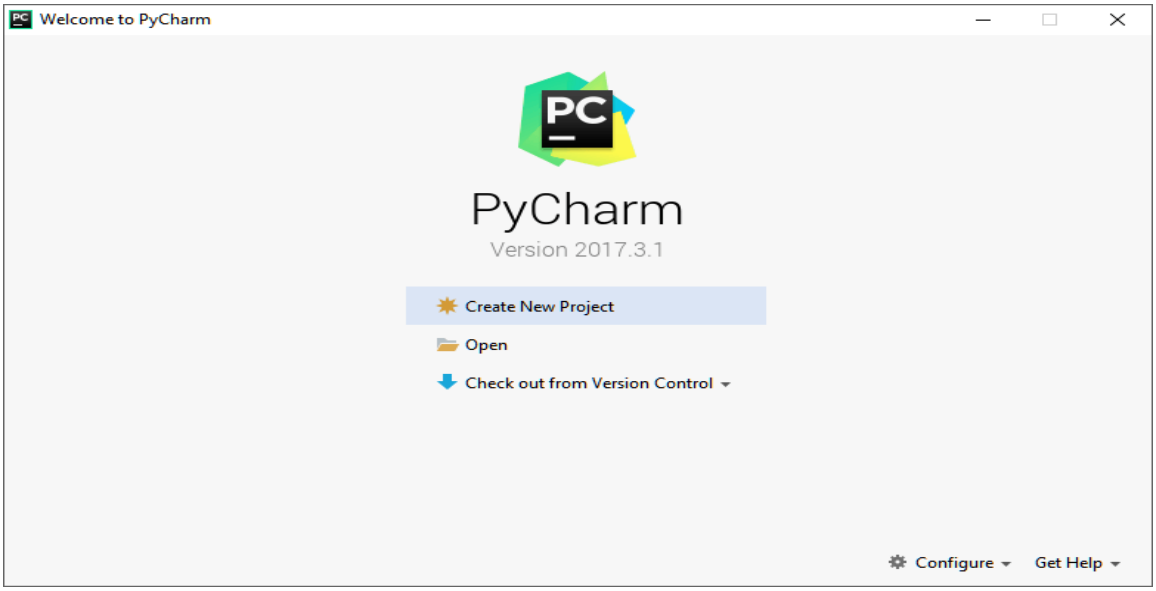

Fig 7.12: Final step

Click on Configure > Settings to open up settings in PyCharm

Search for “Project Interpreter”. My PyCharm looks like this

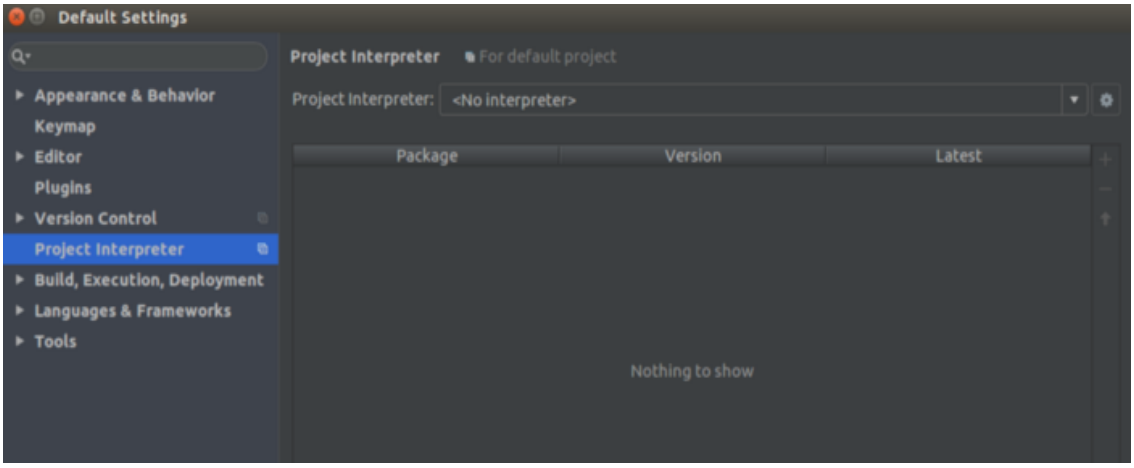

Fig 7.13: Search

Click on Add local via the settings on the right side

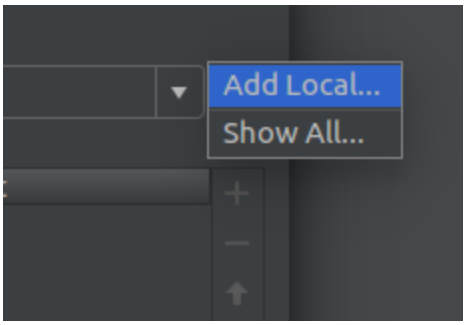

Fig 7.14: Add location

- Select “conda environment”
- Click on “Existing environment” and navigate to the environment that you want to use. Note that you have to select the bin/python file inside the conda environment for PyCharm to be able to recognise the environment
- Make sure to click the “Make available to all projects” if you want the interpreter to be used by multiple projects

Click ok and you are done

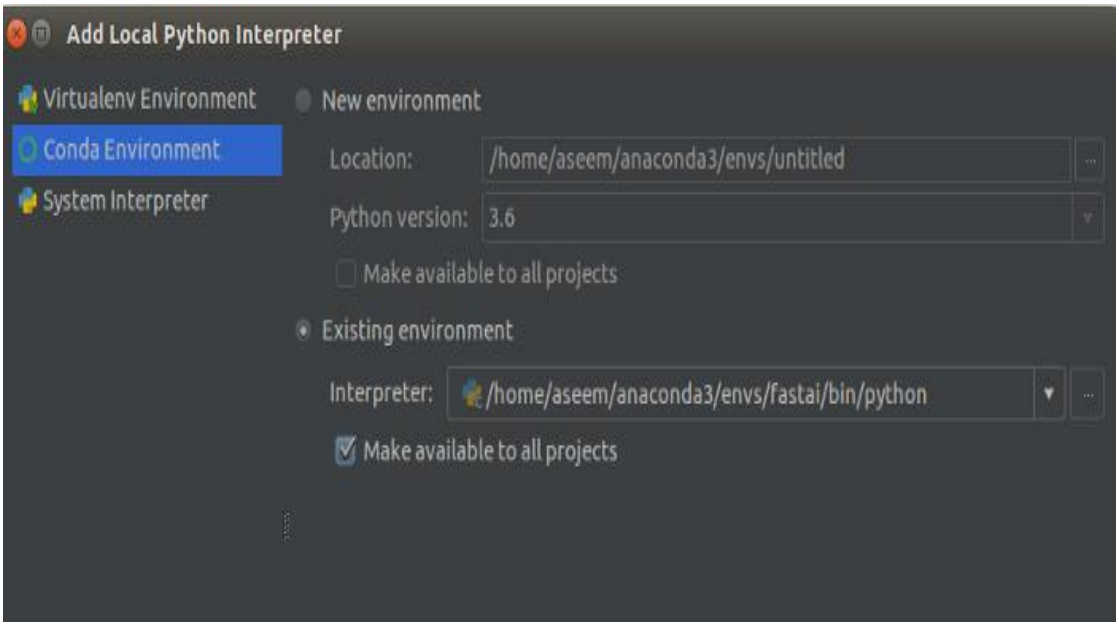

Fig 7.15: Done

CHAPTER 8

CODING AND TESTING

7.1 CODING

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The

conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important.Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

CODING:

```
import cv2
import time
from tkinter import *
from tkinter import messagebox
import pygame
import smtplib
pygame.init()
pygame.mixer.music.load("alarm.wav")
# Capturing Video
cap = cv2.VideoCapture("assets/i.mp4")
if not cap.isOpened():
    print("Initialising the capture...")
    cap.open("assets/test_35.mp4")
    print("Done.")
# Subtracting the background
subtractor = cv2.createBackgroundSubtractorMOG2(history=50, varThreshold=20)
# Text setting up
font = cv2.FONT_HERSHEY_SIMPLEX
# org
org = (40, 50)
```

```
    # fontScale
    fontScale = 0.8
    # Blue color in BGR
    color = (0, 0, 255)
    # Line thickness of 2 px
    thickness = 2
    res = 1
    arcount = 0
    count = -1
    while True:
       # Reading the frame
       res, frame = cap.read()
    if res == True:
          # Applying the mask
          mask = subtractor.apply(frame)
    # finding the contours
          contours, hierarchy = cv2.findContours(
             mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
       # Flag for accident detection
          flag = 0
for cnts in contours:
```

```
(x, y, w, h) = cv2.boundingRect(cnts)
 if w * h > 1000:
                if flag == 1:
                   # If accident detected change color if contours to red
                   cv2.rectangle(frame, (x, y), (x + w, y + h),
                                 (0, 0, 255), 3)
                else:
                   cv2.rectangle(frame, (x, y), (x + w, y + h),
                                 (0, 255, 0), 3)
      # If area of rectangle more than a threshold detect accident
            if w * h > 10000:
               area = w * h
            # Countint the number of frames for which the condition persists to refine the accident
      detection case
               arcount += 1
            # print(arcount)
            if arcount > 35:
               flag = 1
      if flag == 1:
            # If accident detected print Accident on the screen
            frame = cv2.putText(frame, "Accident Detected ", org, font,
```

```
                        fontScale, color, thickness, cv2.LINE_AA, False)
                try:
                    pygame.mixer.music.play()
                    s = smtplib.SMTP('smtp.gmail.com', 587)
    s.starttls()
     s.login("info.acdtest@gmail.com", "Dob11011993")
     message = "Accident Detected"
    s.sendmail("keralamanirajendran17@gmail.com", "dummyidnisha0@gmail.com", message)
    s.quit()
                    #messagebox.showerror("Accident", "Accident Detected")
except:
            pass
              # frame = cv2.putText(frame, "28° 35' 31.7040" N and 77° 2' 45.7836" E. ; ",
                            # (40, 80), font, fontScale, color, thickness, cv2.LINE_AA, False)
              # frame = cv2.putText(frame, "time:1600 hrs; ", (40, 100),
                            # font, fontScale, color, thickness, cv2.LINE_AA, False)
              # frame = cv2.putText(frame, "camera id:DWARKA_ROAD_22", (40, 120),
                            # font, fontScale, color, thickness, cv2.LINE_AA, False)
     cv2.imshow("Accident detection", frame)
    count += 1
if cv2.waitKey(33) & 0xff == 27:
```

```
            break
else:
        break
#2 addresss
#testing
#integrate
import os
from PIL import Image
import numpy as np
import cv2
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
vidcap = cv2.VideoCapture('assets/test_35.mp4')
count = 0
while(vidcap.isOpened()):
    ret, image = vidcap.read()
    length = int(vidcap.get(cv2.CAP_PROP_FRAME_COUNT))
    #print(length)
if (count < length) :
        if(int(vidcap.get(1)) % 50 == 0):
            print('Saved frame number : ' + str(int(vidcap.get(1))))
```

```
            cv2.imwrite("frame%d.jpg" % count, image)
            print('Saved frame%d.jpg' % count)
            count += 1
        else :
            count += 1
    else :
        if(int(vidcap.get(1)) % 50 == 0):
            print('Saved frame number : ' + str(int(vidcap.get(1))))
            cv2.imwrite("frame%d.jpg" % count, image)
            print('Saved frame%d.jpg' % count)
        break
vidcap.release()
allfiles=os.listdir(os.getcwd())
imlist=[filename for filename in allfiles if  filename[-4:] in [".jpg",".JPG"]]
w,h=Image.open(imlist[0]).size
N=len(imlist)
arr=np.zeros((h,w,3),np.float)
for im in imlist:
    imarr=np.array(Image.open(im),dtype=np.float)
    arr=arr+imarr/N
```

```
arr=np.array(np.round(arr),dtype=np.uint8)
image = Image.fromarray(arr,mode="RGB")
image.save("average.jpg")
image = cv2.imread('average.jpg', cv2.IMREAD_COLOR)
print(image.shape)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
image = image.reshape((image.shape[0] * image.shape[1], 3)) # height, width
print(image.shape)
# (25024, 3)
k = 5
clt = KMeans(n_clusters = k)
clt.fit(image)
for center in clt.cluster_centers_:
    print(center)
def centroid_histogram(clt):
    # grab the number of different clusters and create a histogram
    # based on the number of pixels assigned to each cluster
    numLabels = np.arange(0, len(np.unique(clt.labels_)) + 1)
    (hist, _) = np.histogram(clt.labels_, bins=numLabels)
  # normalize the histogram, such that it sums to one
    hist = hist.astype("float")
```

```
    hist /= hist.sum()
 # return the histogram
    return hist
hist = centroid_histogram(clt)
print(hist)
def plot_colors(hist, centroids):
    # initialize the bar chart representing the relative frequency
    # of each of the colors
    bar = np.zeros((50, 300, 3), dtype="uint8")
    startX = 0
 # loop over the percentage of each cluster and the color of
    # each cluster
    for (percent, color) in zip(hist, centroids):
        # plot the relative percentage of each cluster
        endX = startX + (percent * 300)
        cv2.rectangle(bar, (int(startX), 0), (int(endX), 50),
                    color.astype("uint8").tolist(), -1)
        startX = endX
        print(percent, color)
    # return the bar chart
    return bar
```

```
bar = plot_colors(hist, clt.cluster_centers_)

# show our color bar t

plt.figure()

plt.axis("off")

plt.imshow(bar)

plt.show()

import cv2

import time

from tkinter import *

from tkinter import messagebox

#from gps import *

import folium as folium

import geocoder as geocoder

import pygame

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.base import MIMEBase

from email import encoders

from geopy.geocoders import Nominatim

loc = Nominatim(user_agent="GetLoc")
```

```
# entering the location name
getLoc = loc.geocode("OMR,Chennai")
# printing address
print(getLoc.address)
# printing latitude and longitude
print("Latitude = ", getLoc.latitude, "\n")
print("Longitude = ", getLoc.longitude)
url_ = "https://maps.google.com/?q="+str(getLoc.latitude)+","+str(getLoc.longitude)
print(url_)
fromaddr = "info.acdtest@gmail.com"
toaddr = "dummyidnisha0@gmail.com"
# initialize dlib's face detector (HOG-based) and then create the
# facial landmark predictor
pygame.init()
pygame.mixer.music.load("alarm.wav")
# Capturing Video
cap = cv2.VideoCapture("0")
status=0
if not cap.isOpened():
    print("Initialising the capture...")
    cap.open("assets/test_0.mp4")
```

```
    print("Done.")
# Subtracting the background
subtractor = cv2.createBackgroundSubtractorMOG2(history=50, varThreshold=20)
# Text setting up
font = cv2.FONT_HERSHEY_SIMPLEX
# org
org = (40, 50)
# fontScale
fontScale = 0.8
# Blue color in BGR
color = (0, 0, 255)
# Line thickness of 2 px
thickness = 2
res = 1
arcount = 0
count = -1
while True:
    # Reading the frame
    res, frame = cap.read()
if res == True:
        # Applying the mask
```

```
            mask = subtractor.apply(frame)
        # finding the contours
            contours, hierarchy = cv2.findContours(
                mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
            # Flag for accident detection
            flag = 0
for cnts in contours:
                (x, y, w, h) = cv2.boundingRect(cnts)
if w * h > 1000:
                    if flag == 1:
                        # If accident detected change color if contours to red
                        cv2.rectangle(frame, (x, y), (x + w, y + h),
                                      (0, 0, 255), 3)
                    else:
                        cv2.rectangle(frame, (x, y), (x + w, y + h),
                                      (0, 255, 0), 3)
# If area of rectangle more than a threshold detect accident
                if w * h > 10000:
                    area = w * h
                # Countint the number of frames for which the condition persists to refine the accident
        detection case
```

```
                arcount += 1
            # print(arcount)
            if arcount > 35:
                flag = 1
    if flag == 1:
            # If accident detected print Accident on the screen
            frame = cv2.putText(frame, "Accident Detected ", org, font,
                                fontScale, color, thickness, cv2.LINE_AA, False)
            try:
                cv2.imwrite('img.jpg', frame)
                pygame.mixer.music.play()
                #pygame.mixer.music.play()
                status=1
    except:
                pass
    # messagebox.showerror("Accident", "Accident Detected")
    # frame = cv2.putText(frame, "28° 35' 31.7040" N and 77° 2' 45.7836" E. ; ",
                            # (40, 80), font, fontScale, color, thickness, cv2.LINE_AA, False)
            # frame = cv2.putText(frame, "time:1600 hrs; ", (40, 100),
                            # font, fontScale, color, thickness, cv2.LINE_AA, False)
            # frame = cv2.putText(frame, "camera id:DWARKA_ROAD_22", (40, 120),
```

```
                    # font, fontScale, color, thickness, cv2.LINE_AA, False)
 cv2.imshow("Accident detection", frame)
count += 1
 if cv2.waitKey(33) & 0xff == 27:
            break
    else:
        if status==1:
    # instance of MIMEMultipart
            msg = MIMEMultipart()
    # storing the senders email address
            msg['From'] = fromaddr
    # storing the receivers email address
            msg['To'] = toaddr
    # storing the subject
            msg['Subject'] = "Accident detected in jpr clg"
 # string to store the body of the mail
            body = "Accident Detected\n " + "Location " + url_
    # attach the body with the msg instance
            msg.attach(MIMEText(body, 'plain'))
    # open the file to be sent
            filename = "img.jpg"
```

```
        attachment = open("img.jpg", "rb")
# instance of MIMEBase and named as p
        p = MIMEBase('application', 'octet-stream') #to convert into Binary file
# To change the payload into encoded form
        p.set_payload((attachment).read())
# encode into base64
        encoders.encode_base64(p)
p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
# attach the instance 'p' to instance 'msg'
        msg.attach(p)
    # creates SMTP session
        s = smtplib.SMTP('smtp.gmail.com', 587)
# start TLS for security
        s.starttls()
# Authentication
        s.login(fromaddr, "Dob11011993")
# Converts the Multipart msg into a string
        text = msg.as_string()
# sending the mail
        s.sendmail(fromaddr, toaddr, text)
```

```
            # terminating the session
            s.quit()
            print("Mail Send")
        break
import cv2
import time
from tkinter import *
from tkinter import messagebox
import folium as folium
import geocoder as geocoder
import pygame
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
fromaddr = "info.acdtest@gmail.com"
toaddr = "dummyidnisha0@gmail.com"
from geopy.geocoders import Nominatim
# initialize dlib's face detector (HOG-based) and then create the
# facial landmark predictor
```

```
pygame.init()

pygame.mixer.music.load("alarm.wav")

# Capturing Video

cap = cv2.VideoCapture("assets/test_0.mp4")

#cap = cv2.VideoCapture("assets/12.mp4")

#cap = cv2.VideoCapture("assets/1.mp4")

if not cap.isOpened():

    print("Initialising the capture...")

    cap.open("assets/accident.mp4")

    print("Done.")

# Subtracting the background

subtractor = cv2.createBackgroundSubtractorMOG2(history=50, varThreshold=20)

# Text setting up

font = cv2.FONT_HERSHEY_SIMPLEX

# org

org = (40, 50)

# fontScale

fontScale = 0.8

# Blue color in BGR

color = (0, 0, 255)

# Line thickness of 2 px
```

```
thickness = 2

res = 1

arcount = 0

count = -1

while True:

    # Reading the frame

    res, frame = cap.read()


    if res == True:

        # Applying the mask

        mask = subtractor.apply(frame)

# finding the contours

        contours, hierarchy = cv2.findContours(

            mask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)

    # Flag for accident detection

        flag = 0

for cnts in contours:

            (x, y, w, h) = cv2.boundingRect(cnts)

  if w * h > 1000:

                if flag == 1:

                    # If accident detected change color if contours to red
```

```
                cv2.rectangle(frame, (x, y), (x + w, y + h),
                              (0, 0, 255), 3)
            else:
                cv2.rectangle(frame, (x, y), (x + w, y + h),
                              (0, 255, 0), 3)
    # If area of rectangle more than a threshold detect accident
            if w * h > 10000:
                area = w * h
            # Countint the number of frames for which the condition persists to refine the accident detection case
                arcount += 1
            # print(arcount)
            if arcount > 35:
                flag = 1
    if flag == 1:
            # If accident detected print Accident on the screen
            frame = cv2.putText(frame, "Accident Detected ", org, font,
                                fontScale, color, thickness, cv2.LINE_AA, False)
            try:
                cv2.imwrite('img.jpg', frame)
                pygame.mixer.music.play()
```

```
                g = geocoder.ip('me')
                print(g.latlng)
    location = g.latlng
    map = folium.Map(location=location, zoom_start=10)
    folium.CircleMarker(location=location, radius=50, color="red").add_to(map)
    folium.Marker(location).add_to(map)
    map
                map.save("map.html")
                # instance of MIMEMultipart
                msg = MIMEMultipart()
    # storing the senders email address
                msg['From'] = fromaddr
    # storing the receivers email address
                msg['To'] = toaddr
# storing the subject
                msg['Subject'] = "Subject of the Mail"
# string to store the body of the mail
                body = "Body_of_the_mail"
    # attach the body with the msg instance
                msg.attach(MIMEText(body, 'plain'))
```

```
                # open the file to be sent
                filename = "map.html"
                attachment = open("map.html", "rb")
        # instance of MIMEBase and named as p
                p = MIMEBase('application', 'octet-stream')
# To change the payload into encoded form
                p.set_payload((attachment).read())
# encode into base64
                encoders.encode_base64(p)
        p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
        # attach the instance 'p' to instance 'msg'
                msg.attach(p)
                filename = "img.jpg"
                attachment = open("img.jpg", "rb")
 # instance of MIMEBase and named as p
                p = MIMEBase('application', 'octet-stream')
# To change the payload into encoded form
                p.set_payload((attachment).read())
 # encode into base64
                encoders.encode_base64(p)
```

```
                p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
        # attach the instance 'p' to instance 'msg'
                msg.attach(p)
        # creates SMTP session
                s = smtplib.SMTP('smtp.gmail.com', 587)
        # start TLS for security
                s.starttls()
    # Authentication
                s.login(fromaddr, "Dob11011993")
        # Converts the Multipart msg into a string
                text = msg.as_string()
         # sending the mail
                s.sendmail(fromaddr, toaddr, text)
        # terminating the session
                s.quit()
                #messagebox.showerror("Accident", "Accident Detected")
    except:
                pass
            # frame = cv2.putText(frame, "28° 35' 31.7040" N and 77° 2' 45.7836" E. ; ",
                        # (40, 80), font, fontScale, color, thickness, cv2.LINE_AA, False)
            # frame = cv2.putText(frame, "time:1600 hrs; ", (40, 100),
```

```
                    # font, fontScale, color, thickness, cv2.LINE_AA, False)
            # frame = cv2.putText(frame, "camera id:DWARKA_ROAD_22", (40, 120),
                    # font, fontScale, color, thickness, cv2.LINE_AA, False)
 cv2.imshow("Accident detection", frame)
count += 1
 if cv2.waitKey(33) & 0xff == 27:
        break
else:
        break
```

OUTPUT:

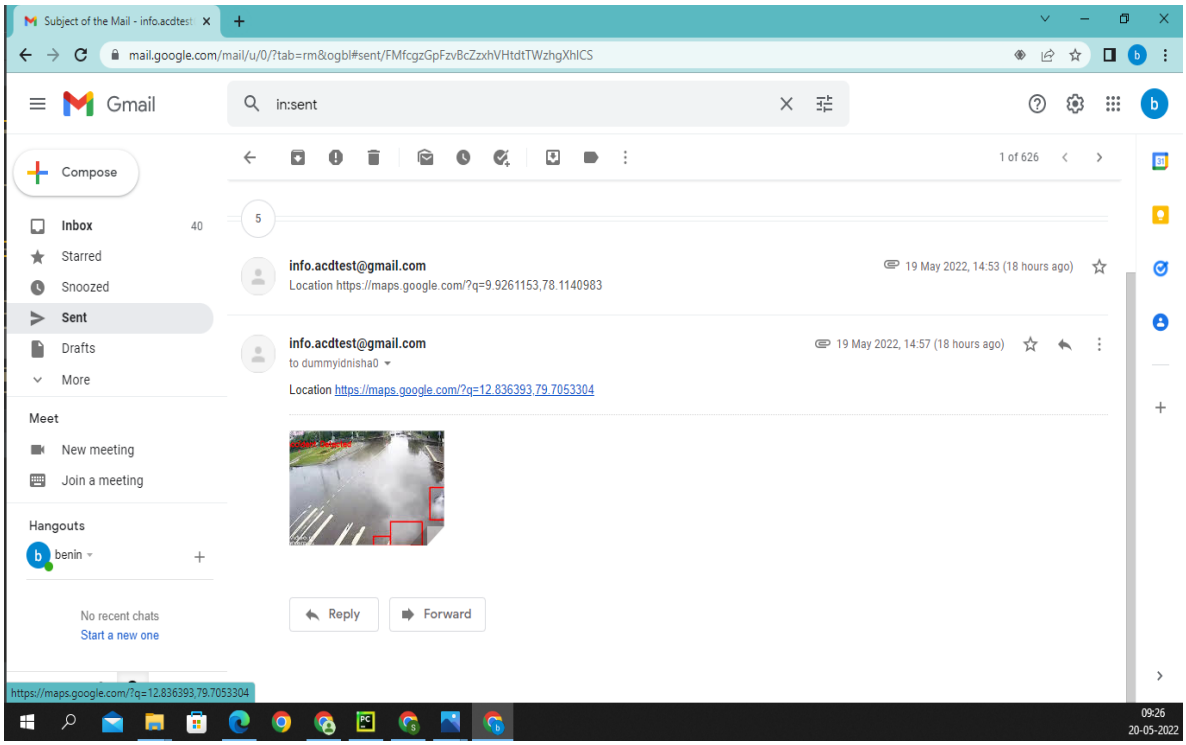

Fig 8.1:Mail sent

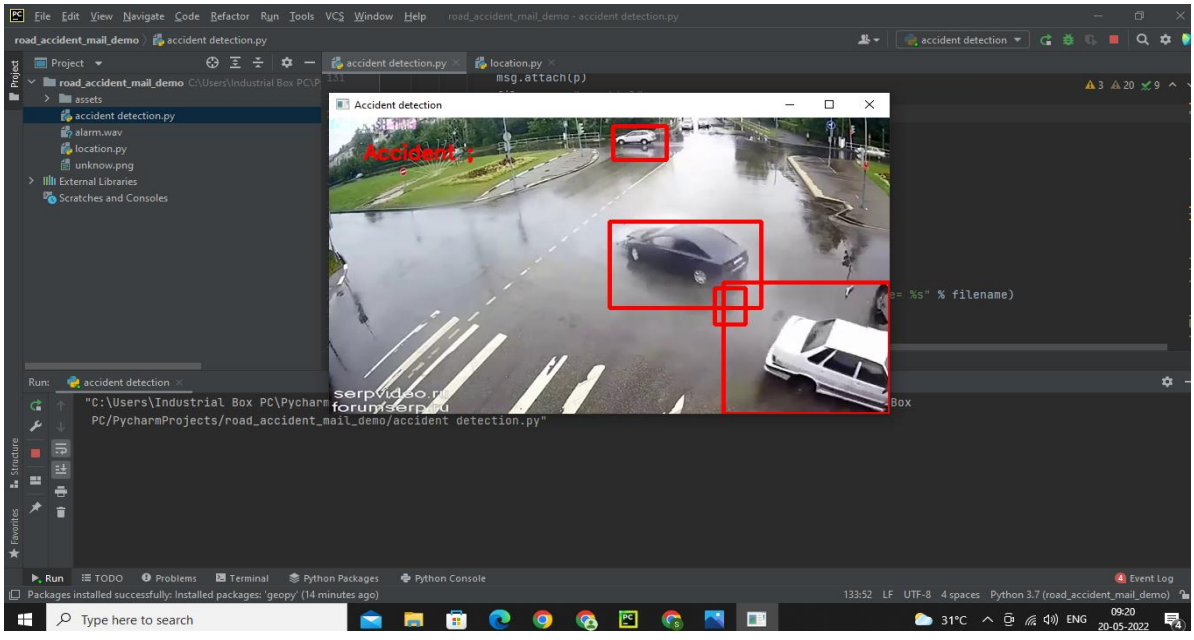

Fig 8.2:CCTV video

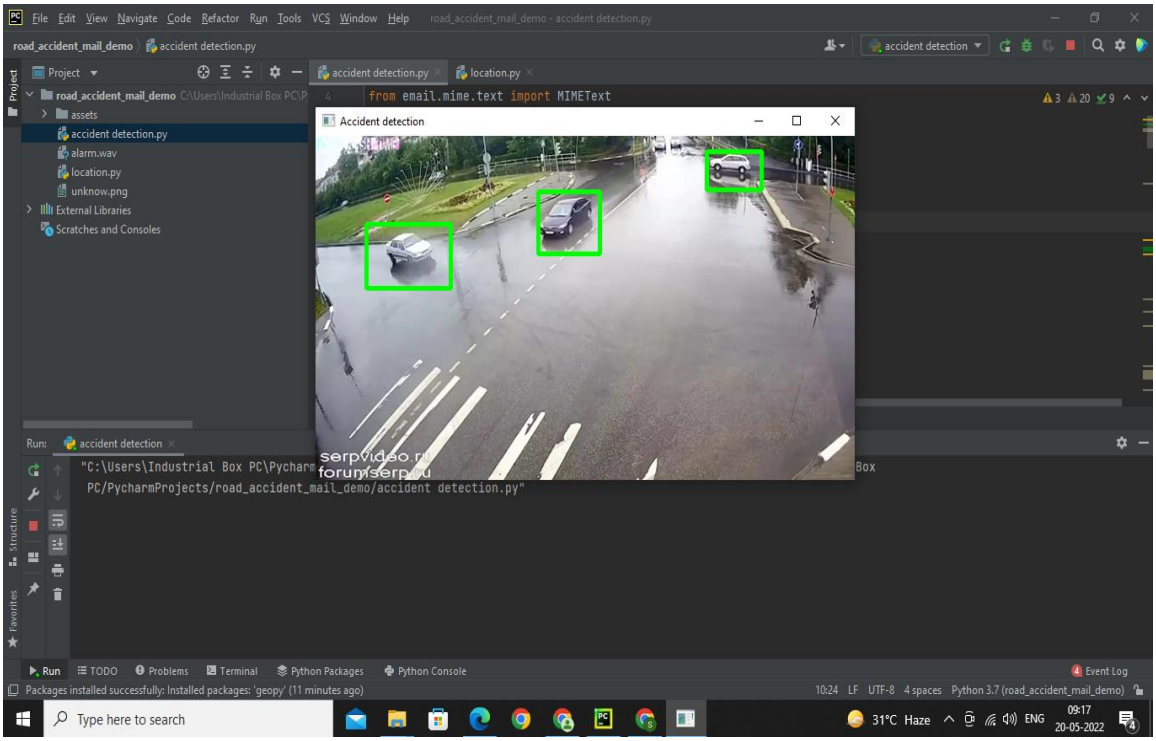

Fig 8.3: Analysing

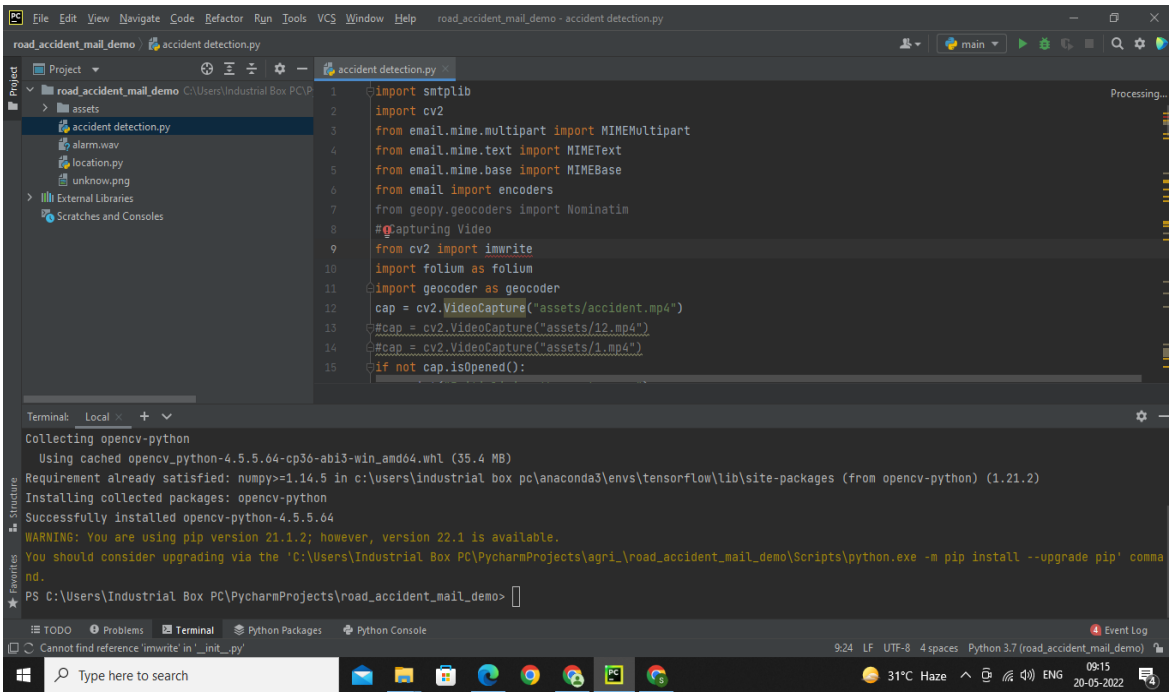

Fig 8.4:Coding running

CHAPTER 9

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The proposed solution is implemented on python, using the OpenCV bindings. The traffic camera footages from variety of sources are in implementation. A simple interface is developed for the user to select the region of interest to be analyzed and then image processing techniques are applied to detect the accident. Currently proposed system works with already captured videos but it can be modified to be used for processing live video streams. One of the limitations of the system is that it is not efficient at detection of occlusion of the vehicles which affects the counting as well as classification. This problem could be solved by introducing the second level feature classification such as the classification on the bases of color. Another limitation of the current system is that it needs human supervision for defining the region of interest. The user has to define an imaginary line where centroid of the contours intersects for the counting of vehicles hence the accuracy is dependent on the judgment of the human supervisor. Furthermore the camera angle also affects the system hence camera calibration techniques could be used for the detection of the lane for the better view of the road and increasing the efficiency. The system is not capable of detection of vehicles in the night as it needs the foreground objects to be visible for extraction of contour properties as well as features for the classification using SIFT features.

FUTURE ENHANCEMENT

- Advanced Feature Classification: Implementing a secondary classification based on vehicle colour or shape can improve detection accuracy in occluded conditions.
- Predictive Accident Analysis – Training machine learning models on historical data to predict high-risk accident zones and prevent collisions.
- IoT-Based Real-Time Alerts – Integrating IoT sensors to automatically send accident alerts to hospitals, police, and emergency services.
- Integration with Emergency Services – Connecting the system with ambulance dispatch and local authorities for faster response times.
- Camera Calibration Techniques: Using lane detection and calibration can improve system performance under varying camera angles.

REFERENCES

[1] Rajvardhan Rishi, Sofiya Yede, Keshav Kunal, Nutan V. Bansode," Automatic Messaging System for Vehicle Tracking and Accident Detection, Proceedings of the International Conference on Electronics and Sustainable Communication Systems,ICESC, 2020

[2] Mr S.Kailasam, Mr Karthiga, Dr Kartheeban, R.M.Priyadarshani, K.Anithadevi, "Accident Alert System using face Recognition",IEEE, 2019.

[3] Bharath P, Saravanan M, Aravindhan K," Literature Survey on Smart Vehicle Accident Prediction Using Machine Learning Algorithm ", 2019

[4] T Kalyani, S Monika, B Naresh, Mahendra Vucha, "Accident Detection and Alert System", 2019

[5] Aarya D.S, Athulya C.K, Anas.P, Basil Kuriakose, Jerin Susan Joy , Leena Thomas, "Accident Alert and Tracking Using Arduino", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 7, Issue 4, April 2018

[6] Nagarjuna R Vatti, PrasannaLakshmi Vatti, Rambabu Vatti, Chandrashekhar Garde, "Smart Road Accident Detection and communication System", 8 IEEE International Conference on Current Trends toward Converging Technologies, Coimbatore, India, 2018

[7] Wen-Kai Tai*, Hao-Cheng Wang, Cheng-Yu Chiang, Chin-Yueh Chien, Kevin Lai, and TsengChang Huang," RTAIS: Road Traffic Accident Information System", 20th International Conference on High Performance Computing and Communications, 2018

[9] Miss. Priyanka M. Sankpal, Prof. P. P. More "Accident Avoidance System Using IR Transmitter ", IJRASET, Volume 5 Issue IV, April 2017

[10] Usman Khalil, Tariq Javid, Adnan Nasir, "Automatic road accident detection techniques: A brief survey", in proceedings of IEEE Symposium on-Wireless Systems and Networks (ISWSN), 2017, 19- 22 Nov. 2017, Lahore, Pakistan

[11] Barış Guksa and Burcu Erkmen, "Smart Phone Application for Drowsiness Detection during Driving ", International Conference on Frontiers of Sensors Technologies, 2017.

[12] Harit Sharma, Ravi Kanth Reddy, Archana Karthik, "S-CarCrash: Real-time Crash Detection Analysis and Emergency Alert using Smartphone ", ICCVE, 2016

[13] sPrashant Kapri, Shubham Patane, Arul Shalom A, "Accident Detection & Alert System", IEEE, 2018 Bruno Fernandes, Vitor Gomes, Joaquim Ferreira and Arnaldo Oliveira, "Mobile Application for Automatic Accident Detection and Multimodal Alert", IEEE, 2015

[14] Adnan Bin Faiz, Ahmed Imteaj, Mahfuzulhoq Chowdhury, "Smart Vehicle Accident Detection and Alarming System Using a Smartphone", International Conference on Computer & Information Engineering, 2015

[15] Manuel Fogue, Piedad Garrido, Francisco J. Martinez,"A System for Automatic Notification and Severity Estimation of Automotive Accidents", IEEE, 2014

[16] Hossam M. Sherif, Hossam M. Sherif, Samah A. Senbel, "Real Time Traffic Accident Detection System using Wireless Sensor Network", International Conference of Soft Computing and Pattern Recognition, 2014

[17] Md. Syedul Amin, Jubayer Jalil, M. B. I. Reaz, "Accident Detection and Reporting System using GPS, GPRS and GSM Technology",ICIEV, 2012

[18] Sharmila Gaikwad, "Mobile Agents in Heterogeneous Database Environment for Emergency Healthcare System", at ITNG 2008 5th International Conference on Information Technology- New Generations April 7-9, 2008