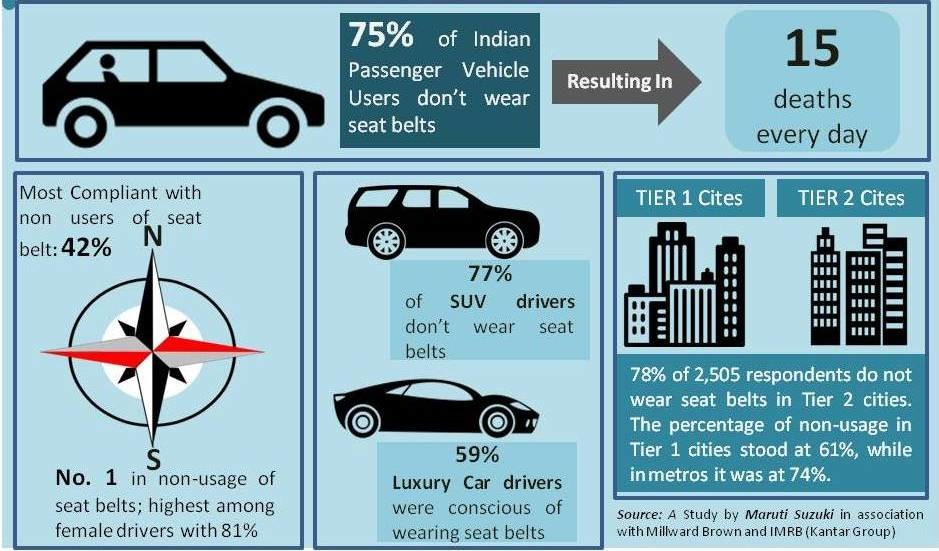
|  |
| --- |
| **BuckleUp AI:**  **The Seat Belt Detection System**  **MINI PROJECT SYNOPSIS**  *Submitted by*  **KRATIKA (2300680140067)**  **RITIK JINDAL (2300680140100)**  *in partial fulfillment for the degree of*  **MASTER OF COMPUTER APPLICATION**  *Under the guidance*  *Of*  **PROJECT COORDINATOR**  MR. ROOBAN AGRAWAL  (Professor)   MCA DEPARTMENT  and  **Associate Professor (Dr.) Satish Kumar Soni**  **HEAD (MCA DEPARTMENT)**  *Submitted to*  **Department of Master of Computer Application,**  **MEERUT INSTITUTE OF ENGINEERING AND TECHNOLOGY, MEERUT (U.P.) - 250005**  and   **DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW**   **UNIVERSITY IN LUCKNOW, UTTAR PRADESH**   2024-25 |

|  |
| --- |
| TABLE OF CONTENT   * Introduction * Problem Statement * Literature Review/ Earlier Work Done * Project Plan * Project Research * Methodology * Roles & Modules * Conclusion * Key Technologies * BuckleUp AI Working * Use Case Diagram * Data Flow Diagram * ER Diagram * BuckleUp AI: Sensors for Seat Belt Detection * Python Code using OpenCV * Screenshots of Sample Outputs * Bibliography |

INTRODUCTION

As a result of Road Traffic Accidents around the world, the lives of about 1.35 million people are cut short annually. About 20 to 50 million people experience nonfatal injuries due to these accidents and many are disabled permanently. Road traffic accidents are causing significant economic impacts on victims, thereby to the whole nation, by costing around 3% loss of the gross domestic product . Thus, traffic accident has emerged as a topic of discussion and analyzing traffic accident data, becoming a major concern for researchers in search of coherent methods for road accidents forecasting. The main aim of accident data analysis is to identify the factors affecting road traffic accident occurrences, thus mitigate the main issues in the area of road safety. The effectiveness of accident prevention methods depends mostly on the genuineness of the gathered and estimated data and the suitability of the analysis methods. The selection of the right data analysis method will help in uncovering the reasons for accident occurrence at a particular zone or location of study and to predict with reasonable precision, the probability of accident occurrence per day or the relative safety of various categories of road users in that region.

**A STUDY REVEALS THAT 75% VEHICLE USERS DON’T WEAR SEAT BELTS**



Road accidents are one of the leading causes of death in the country. According to the Ministry of Road Transport and Highways, in 2016, 1.5 lakh people died in road accidents. Vehicle Users not wearing seat belts are more prone to death. A total of 5,638 people died in 2016 due to non-usage of seat belts. As per the study conducted by Maruti Suzuki India Limited, 75% of Indian Passenger Vehicle users (drivers, co-drivers & rear) don’t wear seat belts leading to an average of 15 deaths per day.  
  
Whereas India’s just 25% of the population comply with seat belt usage; Europe & US comply with 98% & 85% respectively in similar regards. The study also revealed zone-wise figures in which South ranks no.1 in non-usage of seat belts, and non-usage of seat belts among female drivers was highest at 81 percent compared to male drivers at 68 percent. The Northern Zone was the most compliant with 42 percent of non-users.  
  
SUV drivers were the most defaulters with 77% when it came to not wearing seat belts. Whereas, luxury car drivers have the highest compliance rate of 59% in this regard. When we look city-wise, 78% of 2,505 respondents do not wear seat belts in Tier 2 cities. The percentage of non-usage in Tier 1 cities stood at 61%, while in metros it was at 74%.

PROBLEM STATEMENT

**Objective:**

* To improve vehicle safety by ensuring all passengers are wearing seat belts.

**Key Problems:**

* High rate of injuries and fatalities in accidents due to nonuse of seat belts.
* Current systems only detect if a seat is occupied, not if the seat belt is properly fastened.

**Goal:**

* Develop an accurate and reliable system to detect and notify about improper or non-use of seat belts in real-time.

LITERATURE REVIEW/ EARLIER WORK DONE

* **Early Systems:** Sensor-based (pressure/weight) systems to detect seat occupancy and buckle engagement.
* **Infrared & Ultrasonic Sensors:** Enhanced detection but limited by lighting/environmental conditions.
* **Computer Vision & Machine Learning:** Algorithms using image processing and deep learning (CNNs) to verify correct seat belt use.
* **IoT Integration:** Real-time data sharing with cloud servers for alerts and monitoring, especially useful for fleet management.
* **Conclusion:** Advanced systems aim for higher accuracy, real-time performance, and integration with broader safety monitoring.

PROJECT PLAN

**Timeline:**

* Phase 1: Research and Requirement Analysis (Weeks 1-2)
* Phase 2: System Design and Component Selection

(Weeks 3-4)

* Phase 3: Development of Detection Algorithm (Weeks 5-7)
* Phase 4: Testing and Refinement (Weeks 8-10)
* Phase 5: Final Integration and Project Review

(Weeks 11-12)

**Milestones:**

* Literature review completion, Algorithm implementation, Initial Testing, Final Project Demo.
* Deliverables: Design document, functional prototype, final report, and presentation.

PROJECT RESEARCH METHODOLOGY

* **Step 1: Data Collection:** Gather images/videos of seat belt use cases (correct/incorrect).
* **Step 2: Algorithm Development:** Use image processing techniques and CNNs to detect seat belt position on passengers.
* **Step 3: Sensor Integration:** Combine vision-based detection with physical sensors to enhance reliability.
* **Step 4: Testing and Validation:** Evaluate system performance under different lighting and seating conditions.
* **Step 5: IoT Integration:** Enable real-time data sharing and alert notifications to mobile devices or in-car displays.

ROLES & MODULES

***MEMBER A: Kratika***

* Responsible for project planning, scheduling, and team coordination.
* Research and Development.
* Literature review, previous work analysis, and initial system design.

***MEMBER B: Ritik Jindal***

* Testing and Quality Assurance
* Testing, validation of the system, and refinement for real-world conditions.
* IoT Integration and Alerts

CONCLUSION

**Summary:**

* The project aims to create an effective, real-time seat belt detection system that enhances vehicle safety through the integration of computer vision, sensor technology, and IoT.

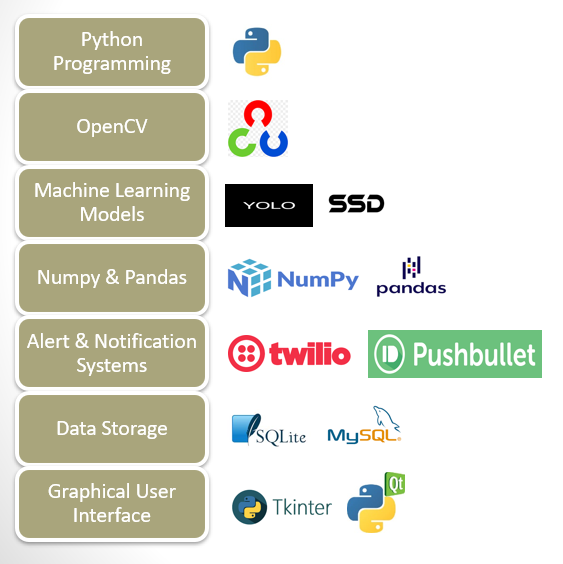
**Expected Outcome:**

* A reliable detection system that can be deployed in real-world scenarios, ensuring improved passenger safety compliance.

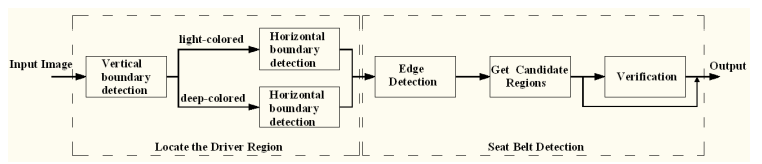
**Future Scope:**

* Potential for further improvements in accuracy, speed, and user interface to make the system more adaptable and accessible in commercial vehicles

KEY TECHNOLOGIES

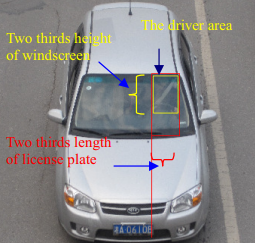


BuckleUp AI WORKING

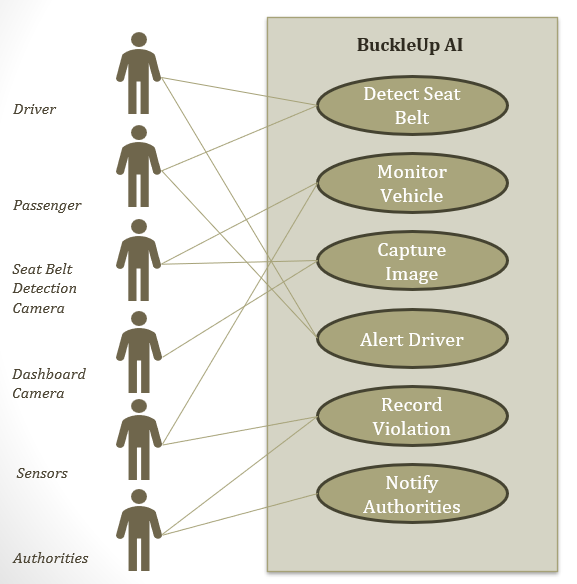


**BuckleUp AI works on two main procedures:**

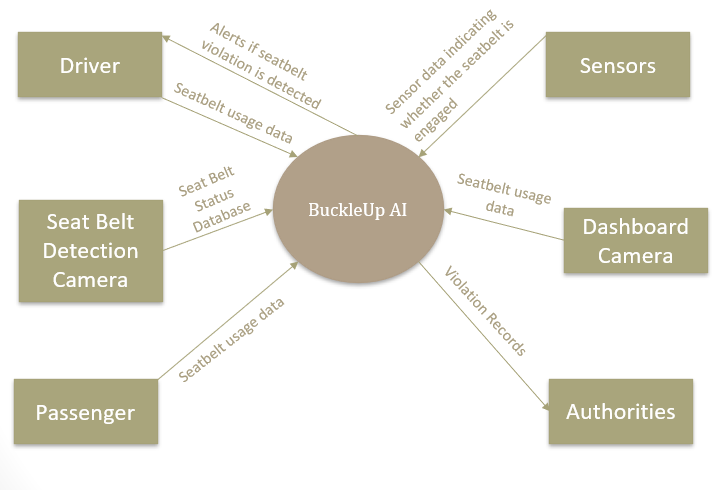
* The first one is to obtain the boundary of the driver area where the potential seat belt area will be detected. The vertical boundaries of the driver’s area are first obtained according to the license plate's position, and the windscreen's horizontal boundaries are detected to detect the ones of the driver’s area.
* The second one is to detect the edges of the seat belt.



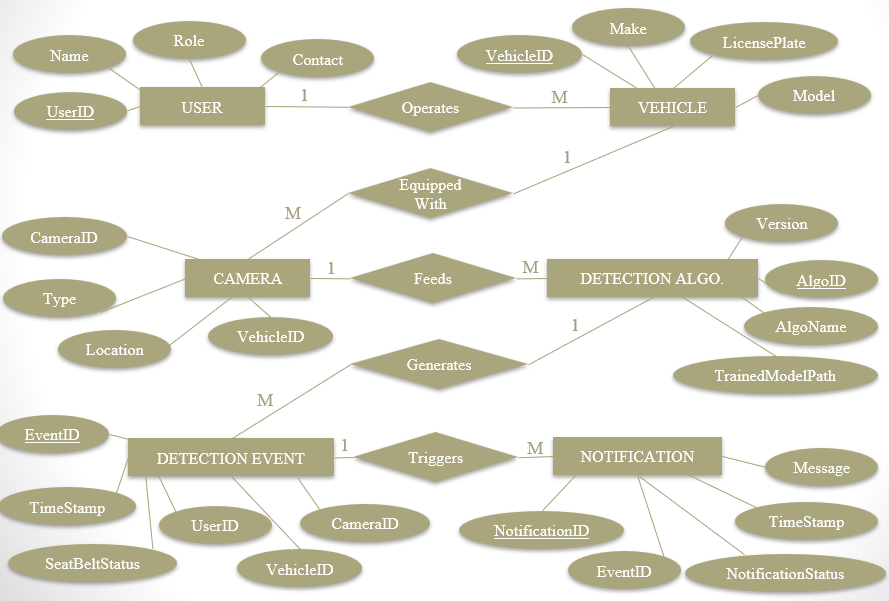
The Driver Area Position in Vehicle

USE CASE DIAGRAM

DATA FLOW DIAGRAM

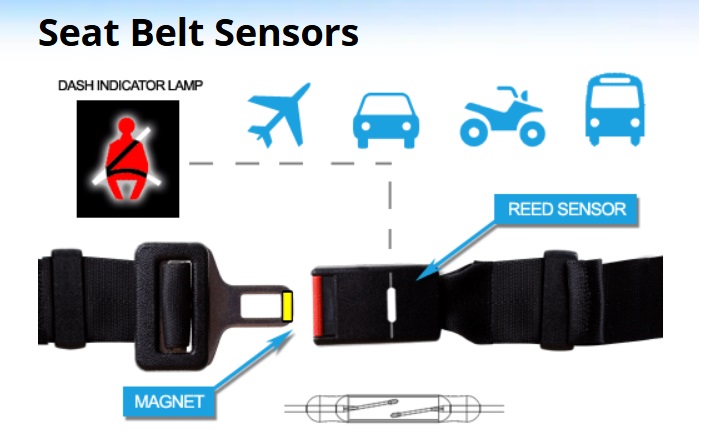


ER DIAGRAM



BuckleUp AI:

SENSORS FOR SEAT BELT DETECTION



As we all know, seat belts are now used by all airlines and in almost all automotive vehicles with the possibility that they may become mandatory for buses and trains as well. In recent years the sensor has been discovered by safety belt designers as being perhaps the best and most reliable way to detect when a seat belt has been engaged.

FEATURES OF THE SEAT BELT SENSORS:

* Ability to activate and control the switching function from only one side of the buckle.
* Designed for SPST (Form A) and SPDT (Form C) switching.
* Designed to handle high-shock environments.
* Designed to operate in dirty environments.
* Dynamically tested contacts.
* Hermetically sealed.
* Millions of reliable switching operations.
* Reliable switching.

PYTHON CODE USING OpenCV

This code is to detect seatbelts in a moving vehicle with the help of Speed Detecting Cameras.

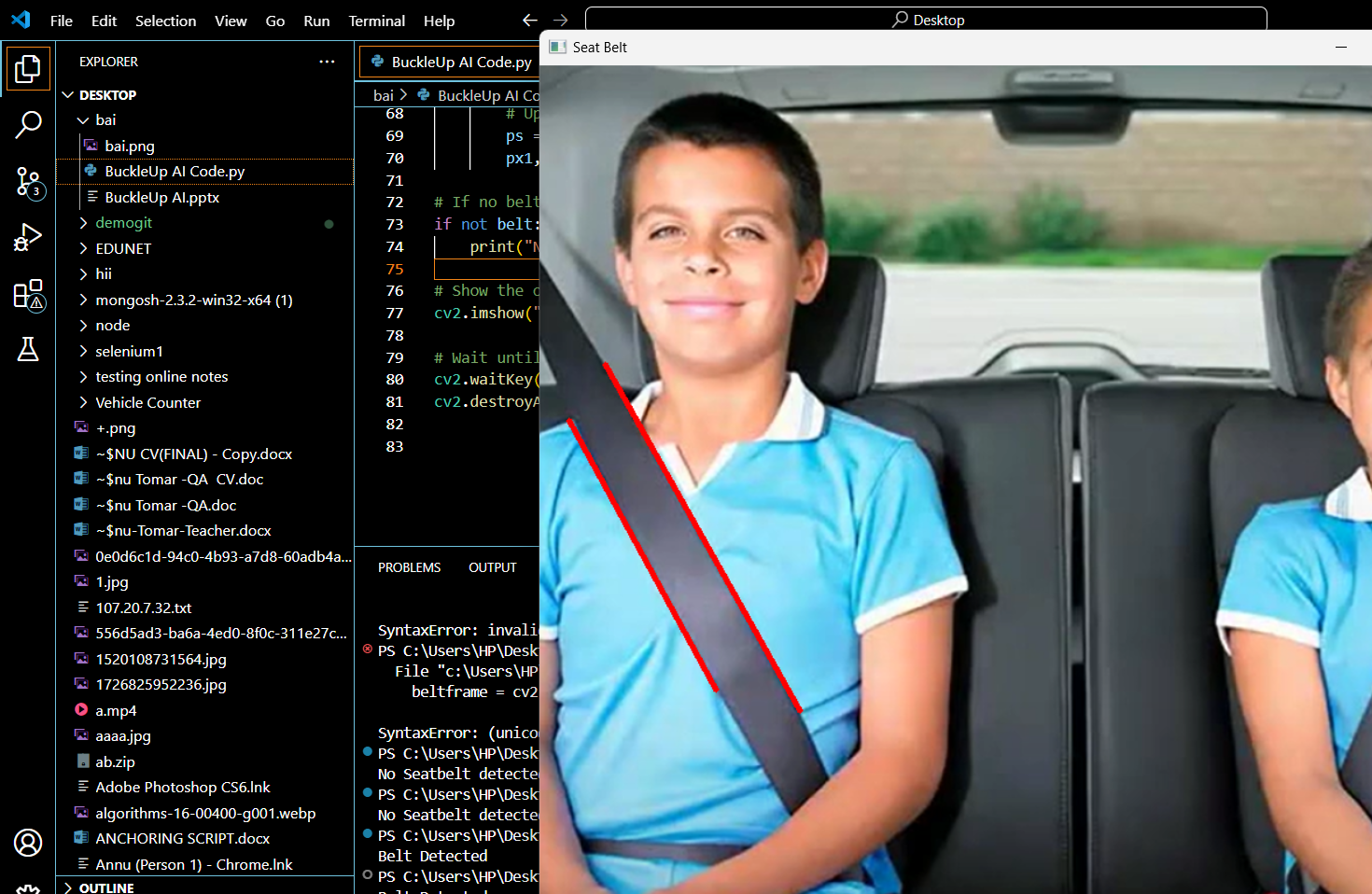
import cv2  
import numpy as np  
import imutils

#Slope of line  
def Slope(a,b,c,d):  
    return (d - b)/(c - a)

# Reading Image  
beltframe = cv2.imread("Test.jpg")  
   
# Resizing The Image  
beltframe = imutils.resize(beltframe, height=800)  
   
#Converting To GrayScale  
beltgray = cv2.cvtColor(beltframe, cv2.COLOR\_BGR2GRAY)  
  
# No Belt Detected Yet  
belt = False  
   
# Bluring The Image For Smoothness  
blur = cv2.blur(beltgray, (1, 1))  
   
# Converting Image To Edges  
edges = cv2.Canny(blur, 50, 400)

# Previous Line Slope  
ps = 0  
   
# Previous Line Co-ordinates  
px1, py1, px2, py2 = 0, 0, 0, 0  
  
  
# Extracting Lines  
lines = cv2.HoughLinesP(edges, 1, np.pi/270, 30, maxLineGap = 20, minLineLength = 170)  
   
# If "lines" are not Empty  
if lines is not None:  
    # Loop line by line  
    for line in lines:  
   
        # Co-ordinates Of Current Line  
        x1, y1, x2, y2 = line[0]  
      # Slope Of Current Line  
        s = Slope(x1,y1,x2,y2)  
          
        # If the Current Line's Slope Is Greater Than 0.7 And Less Than 2  
        if ((abs(s) > 0.7) and (abs (s) < 2)):  
  
  
            # And Previous Line's Slope Is Within 0.7 To 2  
            if((abs(ps) > 0.7) and (abs(ps) < 2)):  
  # And Both The Lines Are Not Too Far From Each Other  
                if(((abs(x1 - px1) > 5) and (abs(x2 - px2) > 5)) or ((abs(y1 - py1) > 5) and (abs(y2 - py2) > 5))):  
   
                    # Plot The Lines On "beltframe"  
                    cv2.line(beltframe, (x1, y1), (x2, y2), (0, 0, 255), 3)  
                    cv2.line(beltframe, (px1, py1), (px2, py2), (0, 0, 255), 3)  
  
  
  
  
  
            # And Previous Line's Slope Is Within 0.7 To 2  
            if((abs(ps) > 0.7) and (abs(ps) < 2)):  
  # And Both The Lines Are Not Too Far From Each Other  
                if(((abs(x1 - px1) > 5) and (abs(x2 - px2) > 5)) or ((abs(y1 - py1) > 5) and (abs(y2 - py2) > 5))):  
   
                    # Plot The Lines On "beltframe"  
                    cv2.line(beltframe, (x1, y1), (x2, y2), (0, 0, 255), 3)  
                    cv2.line(beltframe, (px1, py1), (px2, py2), (0, 0, 255), 3)  
   
                    # Belt Is Detected  
                    print ("Belt Detected")  
                    belt = True  
  
  # Otherwise Current Slope Becomes Previous Slope (ps) And Current Line Becomes Previous Line (px1, py1, px2, py2)              
        ps = s  
        px1, py1, px2, py2 = line[0]  
  
  
  
if belt == False:  
    print("No Seatbelt detected")  
   
# Show The "beltframe"  
cv2.imshow("Seat Belt", beltframe)

SCREENSHOT OF SAMPLE OUTPUT

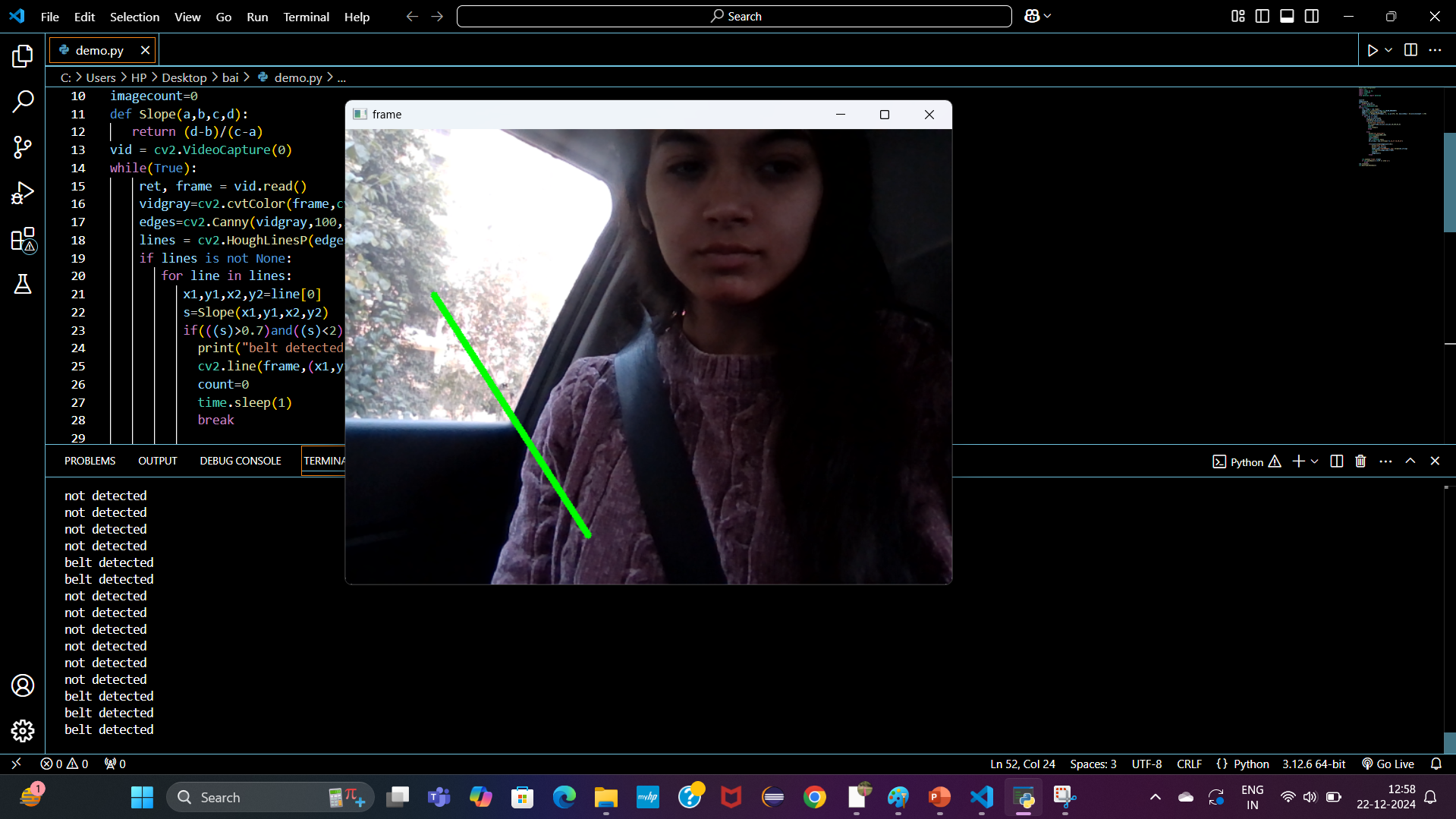


This code can detect seatbelts with the help of an inbuilt camera inside the vehicle.

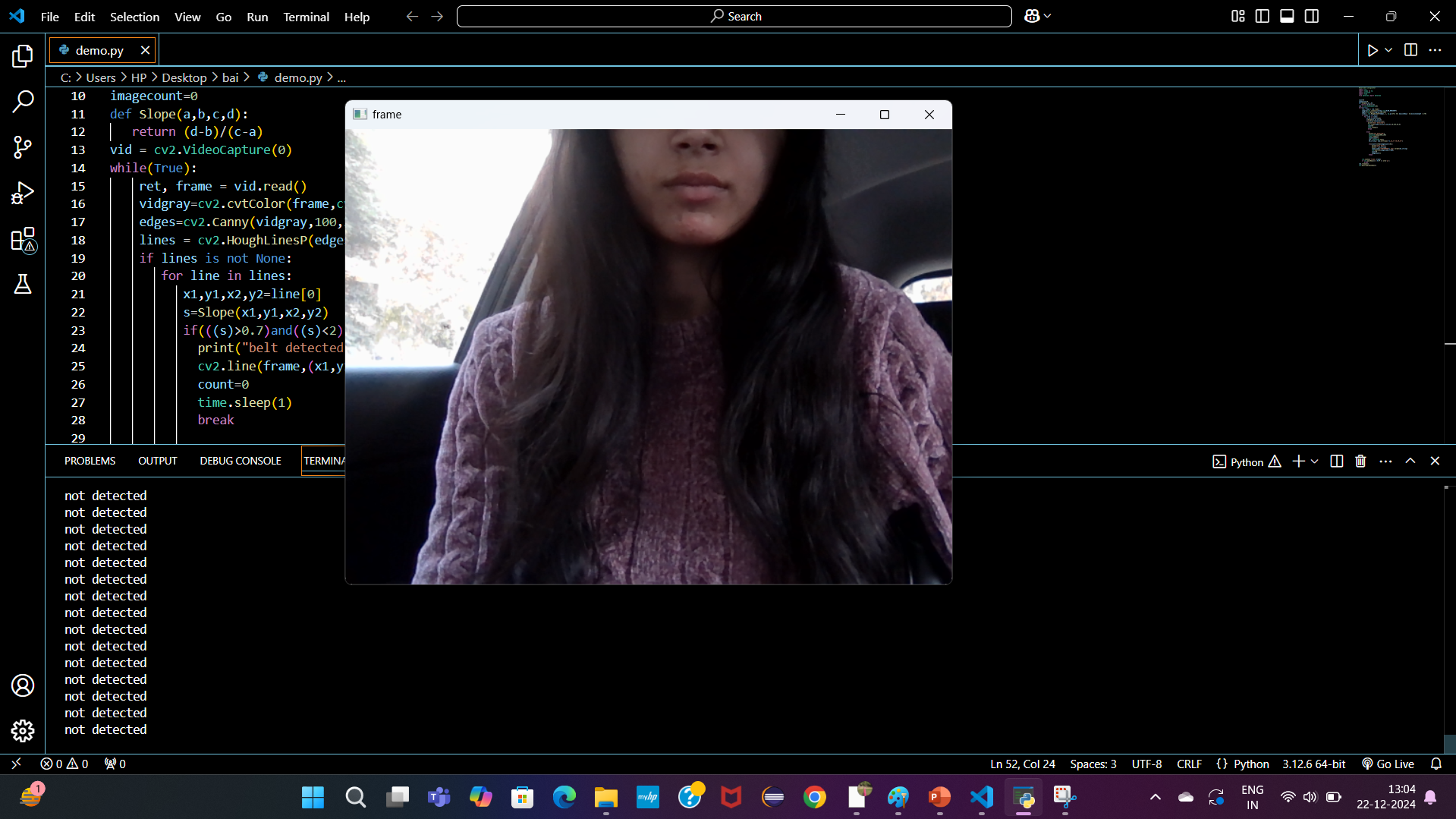
#seat belt invigilator  
import cv2  
import numpy as np  
import winsound  
import time  
from datetime import datetime  
  
count=0  
imagecount=0  
def Slope(a,b,c,d):  
   return (d-b)/(c-a)  
vid = cv2.VideoCapture(0)  
while(True):  
    ret, frame = vid.read()  
    vidgray=cv2.cvtColor(frame,cv2.COLOR\_BGR2GRAY)  
    edges=cv2.Canny(vidgray,100,200)  
    lines = cv2.HoughLinesP(edges, 1, np.pi/270, 30, maxLineGap = 20,minLineLength = 170)  
    if lines are not None:  
       for line in lines:  
          x1,y1,x2,y2=line[0]  
          s=Slope(x1,y1,x2,y2)  
          if(((s)>0.7)and((s)<2)):  
            print("belt detected")  
            cv2.line(frame,(x1,y1),(x2,y2),(0,255,0),6)  
            count=0  
            time.sleep(1)  
            break

          else:  
             print("not detected")  
             winsound.Beep(2000,100)  
             time.sleep(1)  
             count=count+1  
             now = datetime.now()  
             dt\_string = now.strftime("%d\_%m\_%Y %H\_%M\_%S")  
              
             if((count==15)&(imagecount==0)):  
                 print("wear belt")  
                 print("",dt\_string)  
                 image\_name="TS07GW8668{}.png".format(dt\_string)  
                 cv2.imwrite(image\_name,frame)  
                 count=0     
                 imagecount=1    
             break  
              
                  
    cv2.imshow('frame’, frame)       
    if cv2.waitKey(1) & 0xFF == ord('q'):  
       break    
vid.release()  
cv2.destroyAllWindows()

SCREENSHOTS OF SAMPLE OUTPUTS



SEAT BELT IS DETECTED IN THE ABOVE ATTACHED SCREENSHOT



SEAT BELT IS NOT DETECTED IN THE ABOVE ATTACHED SCREENSHOT

BIBLIOGRAPHY

For the Successful completion of my project. I have taken help from the following website links:

1. www.google.com

2. OpenCV Documentation https://docs.opencv.org/.

3. Bradski, Gary, and Adrian Kaehler Learning OpenCV

4. Computer Vision in C++ with the OpenCV Library. O'Reilly Media, 2016.

5. TensorFlow and Keras Documentation <https://www.tensorflow.org/>.

6. National Highway Traffic Safety Administration (NHTSA)

7. Department of Transportation and Road Safety Studies

THANK YOU!