100+ Years of Glorious history inscribed in the yeomen service to the field of education
Poojya Dr. Sharnbaswappa Appaji
Mahadasohi Peethadhipathi
Sharnbasveshwar Samsthana
President,Sharnbasveshwar Vidya Vardhak Sangha
Chancellor, Sharnbasva University
Poojya Matoshri Dr. Dakshayini S. Appa
Chairperson,
Sharnbasveshwar Vidya Vardhak Sangha
Member of BOG Sharnbasva University
Poojya Chiranjeevi Doddappa Appa
Mahadasohi 9th Peethadipatigalu
Sharnbasveshwar Samsthana, Kalaburagi
Estd. 2017
ಸ್ಥಾಪನೆ : 2017
Poojya Matoshri Godutai Avvaji
Poojya Doddappa Appa
Founder President
Sharnbasveshwar Vidya Vardhak Sangha

Centenary Celebrated Sharnbasveshwar Vidya Vardhak Sangha's

ಶರಣಬಸವ ವಿಶ್ವವಿದ್ಯಾಲಯ

Sharnbasva University

Kalaburagi - 585103, Karnataka - India

ಕಲಬುರಗಿ 585 103 ಕರ್ನಾಟಕ - ಭಾರತ

Phone / Fax No. 08472-277852, 277853, 277854, 277855 www.sharnbasvauniversity.edu.in - email : Sharnbasvauniversity@gmail.com

A PROJECT REPORT ON

"VEHICLE MOVEMENT ANALYSIS AND INSIGHT GENERATION IN COLLEGE CAMPUS USING EDGE AI"

Submitted to

INTEL Unnati Industrial Training Program 2024.

Submitted By

STUDENT NAME: ACHYUT **USN: SG21ECE005**

STUDENT NAME: PRATIK **USN: SG21ECE056**

STUDENT NAME: BHAGYASHREE K **USN: SG21ECE022**

STUDENT NAME: POOJA DEODURG **USN: SG21ECE052**

Project Mentor:

DR. SHARANU KORISHETTY

INTRODUCTION

Artificial Intelligence (AI) is a technical system which has the ability to mimic human intelligence as characterized by behaviours such as sensing, learning, understanding, decision-making, and acting. Owing to the availability of powerful computing hardware (GPUs and specialist architectures) and of large amounts of data, AI solutions especially Machine Learning (ML) and more specifically Deep Learning (DL) have found numerous and widespread applications over the past two decades (such as image recognition, fault detection or automated driving functions). Due to their reliance on large amounts of data, most current AI solutions require large-scale cloud data centres for computationally demanding processing tasks

While some advanced driver-assistance systems (ADAS), such as a lane keeping assistant or cruise control, are already commercially available. For several additional vehicle automation functions sufficiently efficient and reliable performance must still be developed and implemented before human drivers can be replaced by AI (in all operating domains). Transferring driving tasks successively from human to AI drivers and meeting all requirements with respect to sensing (scene understanding), decision-making and acting, presents a complex technological challenge with respect to both AI hardware and software models. Today it is clear that besides AI, the connectivity vehicle-to-vehicle (V2V) and between vehicles and infrastructure (V2I) will be key to deploying automated vehicles, since it provides the basis for the coordination of vehicles.

DATASET DESCRIPTION

VEHICLE MOVEMENT ANALYSIS AND INSIGHT GENERATION IN COLLEGE CAMPUS USING EDGE AI

Parking Space Detection:

To detect parking spaces, you can use image processing techniques. One approach is to initialize a parking map with rectangles representing parking spots. You can do this manually or automatically by detecting white marker lines through color or line detection.

OpenCV provides tools for contour detection, edge detection (such as Canny edge detection), and other image processing functions that can help identify parking spaces.

Counting Empty and Occupied Spaces:

Once you've detected parking spaces, you can track them over time. When a car enters or leaves a spot, update the occupancy status accordingly.

Vehicle Speed Estimation:

To estimate vehicle speed, you'll need timestamps of when a car enters and exits a parking spot. Measure the distance between these timestamps (known distance between spots) to calculate speed.

- Upgrade pip with mentioned command below.

```
pip install --upgrade pip
```

- Install libraries with mentioned command below.

```
pip install -r requirements
pip install pandas
pip install numpy
pip install ultralytics
pip install opencv python
pip install pickle
pip install cvzone
pip install matplotlib.pyplot
```

Problem Defination

- finding out the empty parking spaces in a car-parking-lot.
- finding out the speed and no of vehicle.

solution

- Extracting the parking lot coordinates form the image by car_park_coordinate_generator.py script.
- Then use this coordinates to processing every car parking space individualy.
- Implementing digital iamge processing techniques to find out the empty and occupied parking spaces.
- drawing the reults into the image.

Used The Concepts

- OOP concepts
- Opencv High Level GUI Programming
- Opencv Basic Image Processing
- Doc String
- Python Type Annotation

METHODOLOGY

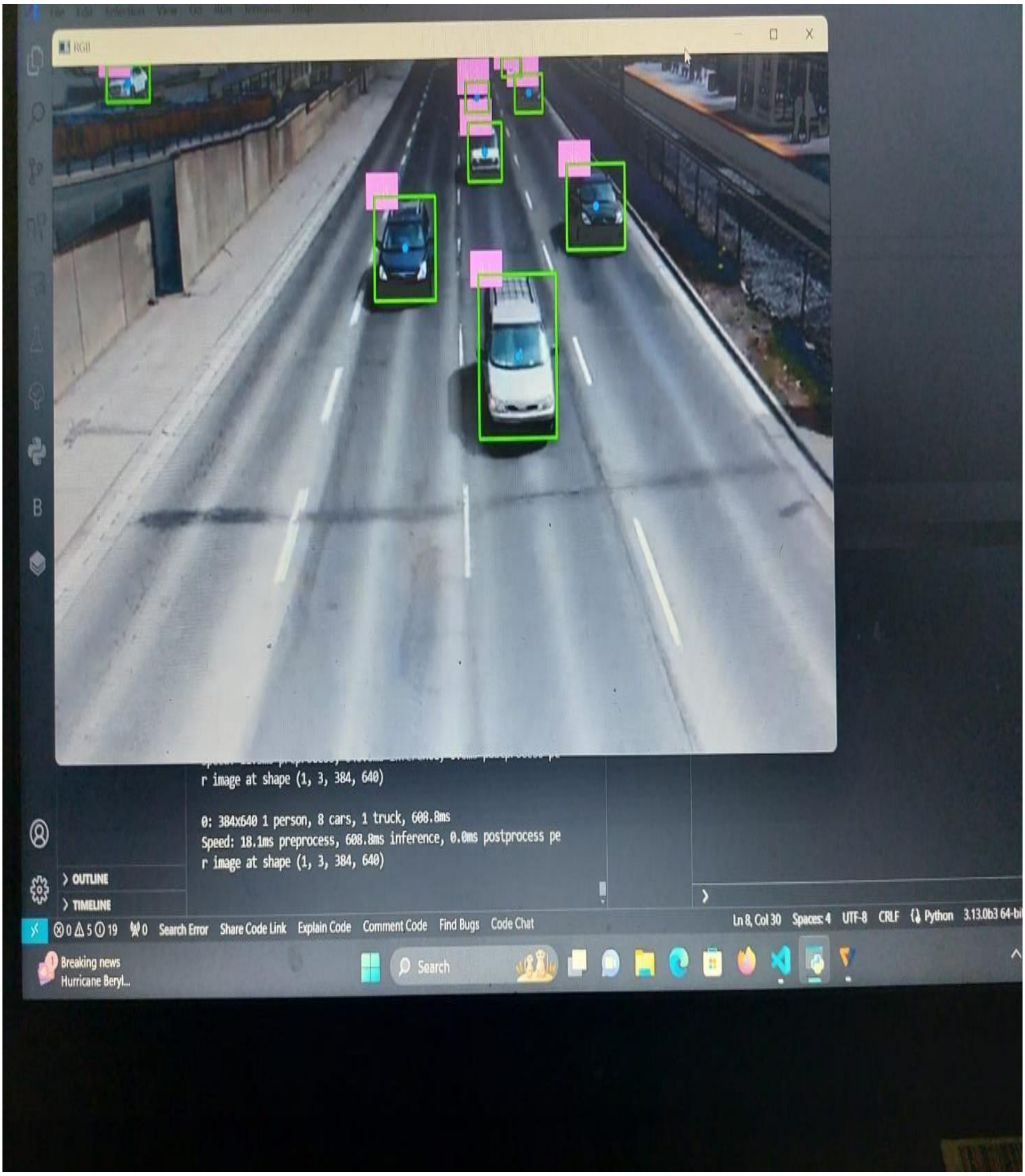
In this project we present our views on the state-of-the-art and future technology milestones in the Edge AI domain. We have put forward an interesting way for the analysis of a vehicle movement using the Edge AI technology and how it makes life easier and smart in the coming decade.

Initially we have used python as a tool for running the project and then uploaded a video and images of movement of vehicles. Then we run the developed code in the VS Code and observe the output. So as mentioned earlier when we run the code in VS Code and upload the videos and images in it, we can observe that the areas with no vehicles are marked with green boundary and the areas with vehicles are marked with red boundary.

We can also get the number of vehicles approaching and leaving the area present in the uploaded video or image respectively.

Hence by this technology i.e; Edge AI we can determine the presence or absence of the vehicles and also number of vehicles approaching or leaving the area without any requirement of manpower by making life smarter by right use of technology by taking it as a boon and not as a bane.

RESULTS AND DISCUSSION



```
r image at shape (1, 3, 384, 640)

0: 384x640 1 person, 8 cars, 1 truck, 608.8ms
Speed: 18.1ms preprocess, 608.8ms inference, 0.0ms postprocess pe
r image at shape (1, 3, 384, 640)
```

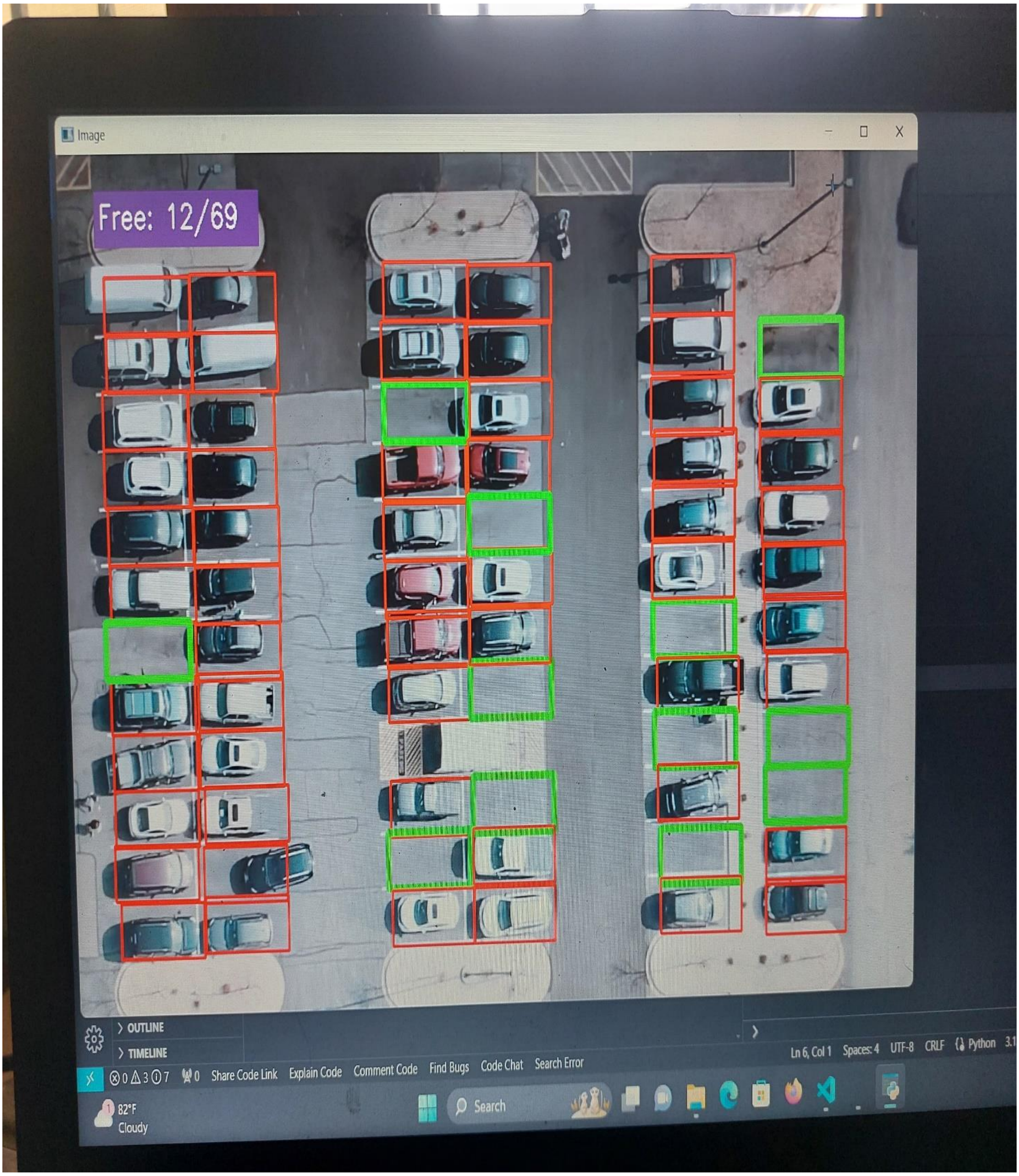


Image
Free: 12/69
OUTLINE
TIMELINE
Share Code Link
Explain Code
Comment Code
Find Bugs
Code Chat
Search Error
Ln 6, Col 1
Spaces: 4
UTF-8
CRLF
Python
82°F
Cloudy
Search

CONCLUSION

Smart Parking System using Python and OpenCV:

This system is built on the Edge Detection method, which recognizes parking spaces with filled and unfilled spots. It aims to eliminate human labor requirements by providing real-time results for intelligent parking spot recognition.

In this project we present our views on the state-of-the-art and future technology milestones in the Edge AI domain. We have put forward an interesting way for the analysis of a vehicle movement using the Edge AI technology and how it makes life easier and smart in the coming decade.

With the proliferation of mobile devices and a wealth of rich application services, the Internet of vehicles (IoV) has struggled to handle computationally intensive and delay-sensitive computing tasks. To substantially reduce the latency and the energy consumption, application work is offloaded from a mobile device to a remote cloud or a nearby mobile edge cloud for processing. Compared with remote clouds, mobile edge clouds are located at the edge of the network. Therefore, mobile edge computing (MEC) has the advantages of effectively utilizing idle computing and storage resources at the edge of the network and reducing the network transmission delay. In addition, mobile devices are increasingly moving toward intelligence. To satisfy the service experience and service quality requirements of mobile users, the vehicle Internet is transforming into the intelligent vehicle Internet. Artificial intelligence (AI) technology can adapt to rapidly changing dynamic environments to provide multiple task requirements for resource allocation, computational task scheduling, and vehicle trajectory prediction. This article introduces IoV from three aspects, namely, MEC, AI and the advantages of combining the two, and analyses the corresponding architecture and implementation technology. The application of MEC and AI in IoV is analysed and compared with current approaches. Finally, several promising future directions in the field of IoV are discussed.

CODE

1. Code for parking space counting

```
import cv2
import numpy as np
import pickle

rectW,rectH=107,48

cap=cv2.VideoCapture('carPark.mp4')

with open('carParkPos','rb') as f:
    posList=pickle.load(f)
frame_counter = 0
def check(imgPro):
    spaceCount=0
    for pos in posList:
        x,y=pos
        crop=imgPro[y:y+rectH,x:x+rectW]
        count=cv2.countNonZero(crop)
        if count<900:
            spaceCount+=1
            color=(0,255,0)
            thick=5
        else:
            color=(0,0,255)
            thick=2

        cv2.rectangle(img,pos,(x+rectW,y+rectH),color,thick)
    cv2.rectangle(img,(45,30),(250,75),(180,0,180),-1)
    cv2.putText(img,f'Free:
{spaceCount}/{len(posList)}',(50,60),cv2.FONT_HERSHEY_SIMPLEX,0.9,(255,25
5,255),2)

while True:
    _,img=cap.read()
    if frame_counter == cap.get(cv2.CAP_PROP_FRAME_COUNT):
        frame_counter = 0 #Or whatever as long as it is the same as next line
        cap.set(cv2.cv.CV_CAP_PROP_POS_FRAMES, 0)
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    blur=cv2.GaussianBlur(gray,(3,3),1)

Thre=cv2.adaptiveThreshold(blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv
2.THRESH_BINARY_INV,25,16)
```

```
        blur=cv2.medianBlur(Thre,5)
        kernel=np.ones((3,3),np.uint8)
        dilate=cv2.dilate(blur,kernel,iterations=1)
        check(dilate)

        cv2.imshow("Image1",img)
        cv2.waitKey(10)
```

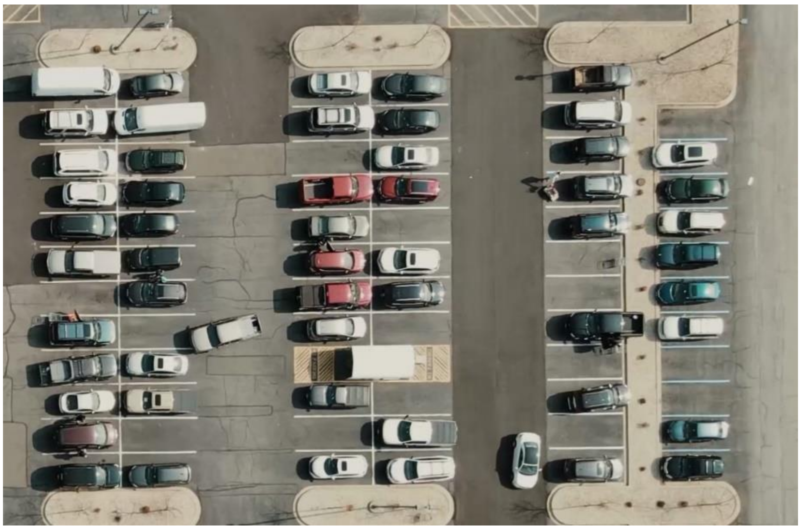
2. Code for vehicle counting and detection (main code)

```
import cv2
import numpy as np
from ultralytics import YOLO
import pandas as pd
import cvzone
from tracker import Tracker

model = YOLO("yolov10s.pt")

def RGB(event, x, y, flags, param):
    if event == cv2.EVENT_MOUSEMOVE:
        point = [x, y]
        print(point)

cv2.namedWindow('RGB')
cv2.setMouseCallback('RGB', RGB)

cap=cv2.VideoCapture('cars.mp4')
my_file = open("coco.txt", "r")
data = my_file.read()
class_list = data.split("\n")

tracker=Tracker()
cy1=425
```

```
offset=6

listcardown=[]
count=0
while True:
    ret,frame = cap.read()
    count += 1
    if count % 2 != 0:
        continue
    if not ret:
       break
    frame = cv2.resize(frame, (1020, 600))

    results = model(frame)
    a = results[0].boxes.data
    px = pd.DataFrame(a).astype("float")
    list=[]
    for index, row in px.iterrows():
        x1 = int(row[0])
        y1 = int(row[1])
        x2 = int(row[2])
        y2 = int(row[3])
        d = int(row[5])
        c = class_list[d]
        if 'car' in c:
           list.append([x1,y1,x2,y2])
    bbox_idx=tracker.update(list)
    for bbox in bbox_idx:
        x3,y3,x4,y4,id=bbox
        cx=int(x3+x4)//2
        cy=int(y3+y4)//2
        cv2.circle(frame,(cx,cy),4,(255,0,0),-1)
        cvzone.putTextRect(frame,f'{id}',(x3,y3),1,1)
        cv2.rectangle(frame,(x3,y3),(x4,y4),(0,255,0),2)

#       if cy1<(cy+offset) and cy1>(cy-offset):
#          cvzone.putTextRect(frame,f'{id}',(x3,y3),1,1)
#          cv2.rectangle(frame,(x3,y3),(x4,y4),(0,0,255),2)
#          if listcardown.count(id)==0:
#             listcardown.append(id)

  #  cv2.line(frame,(343,425),(961,425),(255,255,255),1)
  #  cardown=len(listcardown)
   # cvzone.putTextRect(frame,f'Cardown:-{cardown}',(50,60),2,2)
      cv2.imshow("RGB", frame)
    # Break the loop if 'q' is pressed
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

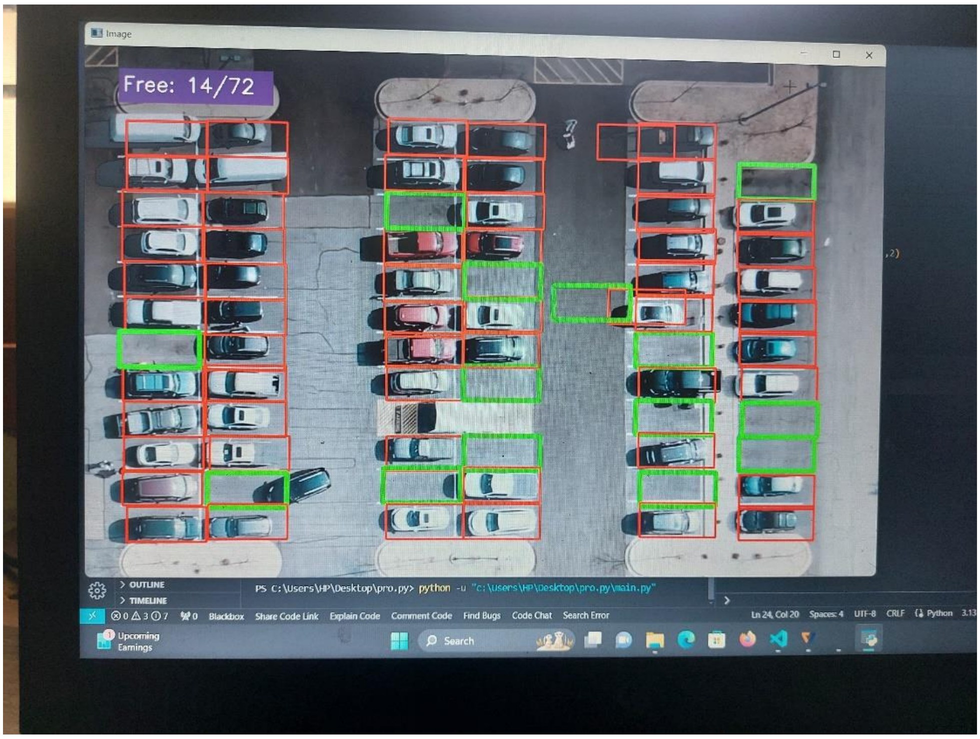
SOLUTION FEATURES

1. Data preprocessing



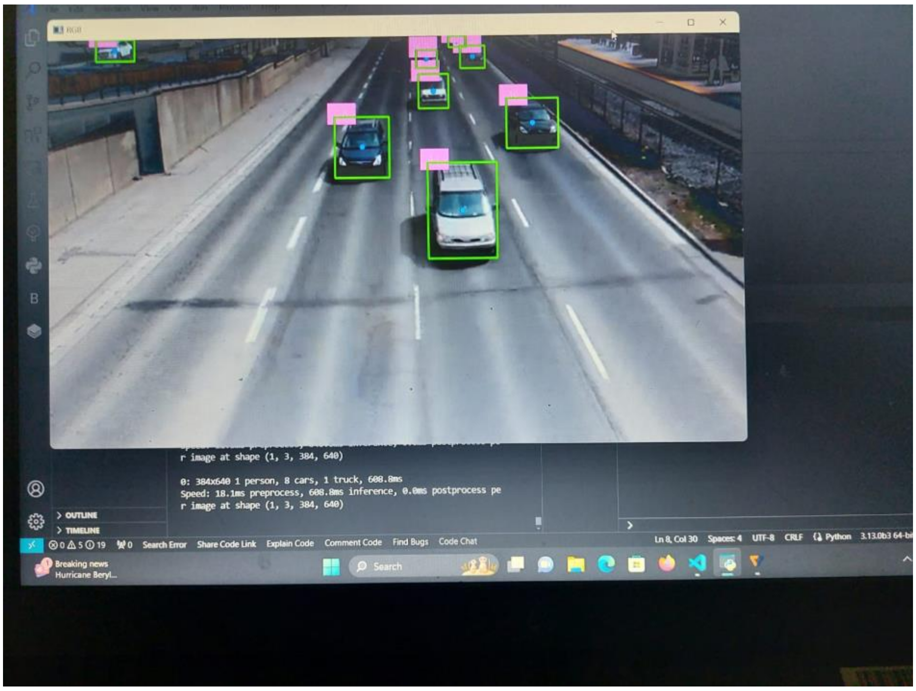
2. Vehicle movement analysis

3. Parking occupancy monitoring



```
PS C:\Users\HP\Desktop\pro.py> python -u "c:\Users\HP\Desktop\pro.py\main.py"
```

4. Vehicle movement analysis



```
r image at shape (1, 3, 384, 640)

0: 384x640 1 person, 8 cars, 1 truck, 608.8ms
Speed: 18.1ms preprocess, 608.8ms inference, 0.0ms postprocess pe
r image at shape (1, 3, 384, 640)
```

TEAM CONTRIBUTION

The contribution of vehicle movement analysis and insight generation in a college campus using edge ai. This project is obediently completed with the contribution of team members ,but also there are few cases where individuals worked hard for a particular work such as

1. Achyut and Pratik worked on the resource related to coding and algorithms and to present the working of project.

2. Pooja and Bhagyashree worked on the report and studied the whole project given a best presentation in report.