VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI-590018

DIGITAL IMAGE PROCESSING LABORATORY WITH MINI PROJECT (18AIL67)

REPORT ON

“VEHICLE DENSITY COUNTER FOR OBJECT DETECTION”

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

In

Artificial Intelligence & Machine Learning

By

LIKHITH V

1KS20AI017

Under the guidance of

Prof. Roopa K Murthy

Asst. Prof, Dept. of AIML

Dr. Amulyashree S

Asst. Prof, Dept. of AIML

Department of Artificial Intelligence & Machine Learning

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-56010

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Artificial Intelligence & Machine Learning

CERTIFICATE

This is to certify that Mini Project work entitled **"VEHICLE DENSITY COUNTER FOR OBJECT DETECTION"** is carried out by **LIKHTIH V** bearing USN **"1KS20AI017"** bonafide student of **K.S. Institute of Technology** in the partial fulfillment for the award of the **Bachelor of Engineering in Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University, Belagavi**, during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini-project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said degree for the Sixth semester.

Prof. Anu Mathews
Asst. Professor & in-charge HOD,
AIML Department ,KSIT

Dr. Dilip Kumar K
Principal/Director, KSIT

Prof. Roopa K Murthy
Asst. Prof, Dept. of AIML

Dr. Amulyashree S
Asst. Prof, Dept. of AIML

Name of the Examiners Signature with date

1.

2.

ACKNOWLEDGEMENT

I take this opportunity to thank everyone involved in the successful implementation of this mini project. I would like to thank the college for providing me an opportunity to work on the mini project.

I take this opportunity to express my sincere gratitude to my college **K.S. Institute of Technology,** Bengaluru for providing the environment to work on this mini project.

I would like to express my gratitude to our **MANAGEMENT,** K.S. Institute of Technology, Bengaluru, for providing a very good infrastructure and all the support provided for carrying out this mini project work in college.

I would like to express my gratitude to **Dr. K.V.A Balaji, CEO**, K.S. Institute of Technology, Bengaluru, for his valuable guidance.

I would like to express my gratitude to **Dr. Dilip Kumar K**, **Principal/Director**, K.S. Institute of Technology, Bengaluru, for his continuous support.

I would like to extend my gratitude to **Prof. Anu Mathews**, **Asst. Professor and In-charge HOD**, Department of Artificial Intelligence & Machine Learning, for providing very good facilities and all the support in successfully carrying out this Mini Project.

I also like to thank my Mini Project Coordinators, **Dr. Amulyashree S, Asst. Professor, Prof. Roopa K Murthy, Asst. Professor,** Department of Artificial Intelligence & Machine Learning, for their help and support in successfully carrying out the Mini Project work.

I am also thankful to the teaching and non-teaching staff of the Artificial Intelligence & Machine Learning Department, KSIT for the help provided in completing this Mini Project.

LIKHITH V

1KS20AI017

ABSTRACT

The project titled "Vehicle density counter for object detection" aims at the accurate counting of vehicles on roads and highways essential for effective traffic management, road safety analysis, and infrastructure planning. Manual counting methods are time-consuming and prone to errors, highlighting the need for automated solutions using digital image processing techniques. This project aims to address this need by developing a system that can automatically count vehicles in real-time from videos captured on roads or highways.

Methodology: The proposed methodology for this project involves the use of pre-trained YOLOv8s (You Only Look Once) model, a deep learning-based object detection algorithm, to identify vehicles in the captured images (or frames from video). The model is capable of detecting and classifying different vehicle classes, including cars, bikes, and trucks. Subsequently, a counting mechanism is implemented to accurately count the vehicles of each class. The performance of the system is evaluated using a dataset of road and highway video, and the results are compared with manual counting for validation.

Outcomes: The outcomes of the project demonstrate the effectiveness of using the YOLOv8s model for vehicle detection and the accuracy of the implemented counting mechanism. The system offers a reliable and efficient solution for real-time vehicle counting, providing valuable data for traffic authorities to optimize traffic flow, reduce congestion, and aid in infrastructure planning.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 What is Digital Image Processing?

Digital image processing refers to the manipulation and analysis of images using computer algorithms and techniques. It involves various operations such as image enhancement, image restoration, image segmentation, and image classification, among others. The field of digital image processing has found applications in numerous domains, including computer vision, remote sensing, and multimedia.

Digital image processing is a multidisciplinary field that focuses on the manipulation and analysis of digital images using computer algorithms and techniques. With the advancement of technology and the widespread use of digital imaging devices, such as digital cameras and scanners, the need for efficient and effective methods to process and analyze digital images has grown exponentially.

Digital image processing enables us to enhance, transform, and extract valuable information from images, making them more visually appealing, easier to interpret, and suitable for various applications. It involves applying mathematical operations and algorithms to digital images to achieve desired outcomes.

The field of digital image processing encompasses a wide range of techniques and processes, including image enhancement, restoration, compression, segmentation, recognition, registration, and feature extraction. These techniques allow us to improve image quality, remove noise or artifacts, reduce file sizes, identify objects or patterns in images, align multiple images, and extract relevant features for further analysis.

The applications of digital image processing are extensive and span various domains. In the medical field, it aids in diagnosing and analyzing medical images. In astronomy, it helps in processing and analyzing images captured by telescopes to study celestial objects and phenomena. In surveillance and security, it enables the detection and tracking of objects and individuals from video footage. In computer vision, it plays a vital role in tasks such as object recognition, autonomous navigation, and augmented reality.

1.2 Problem Statement:

The problem addressed in this project is the accurate counting of vehicles on a road or highway. Vehicle counting plays a crucial role in traffic management, road safety analysis, and infrastructure planning. However, manual counting is time-consuming and prone to errors. Therefore, the aim is to automate the process using digital image processing techniques.

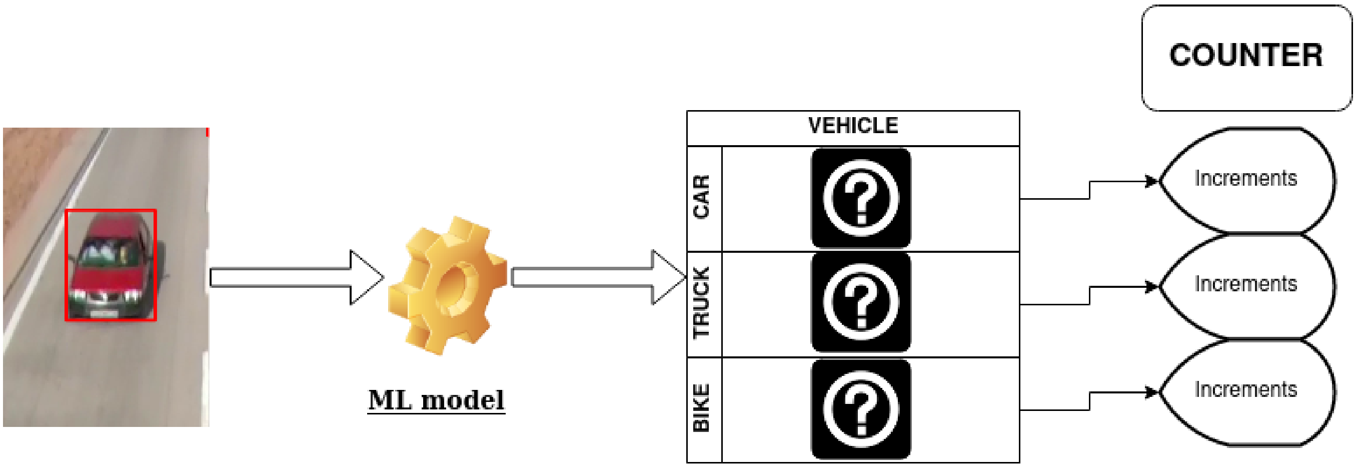

Fig.1 Overview of the project

1.3 Need to solve the problem:

- The need to solve the vehicle counting problem is driven by the growing demand for efficient traffic monitoring and management systems. Manual vehicle counting methods are time-consuming and prone to errors, hindering the ability to obtain accurate and real-time vehicle count data. By automating the counting process using digital image processing techniques, we can overcome these limitations and provide traffic authorities with reliable and up-to-date information.
- Automated vehicle counting offers numerous benefits for traffic management. It enables traffic authorities to make informed decisions for traffic flow optimization, congestion reduction, and infrastructure planning. By accurately tracking the number of vehicles on roads and highways, authorities can identify traffic patterns, peak hours, and congested areas. This information is invaluable for implementing targeted strategies to alleviate congestion and enhance traffic efficiency.
- Furthermore, optimizing signal timings is a crucial aspect of traffic management.

During signal phases, it is common for all four directions to have equal waiting times. However, simply synchronizing the signal timings without considering the density of vehicles in each direction may lead to inefficient traffic flow. To address this, the signal timings should be dynamically adjusted based on the density of vehicles in each direction. By considering the number of vehicles in a particular direction, signals can be set to allow longer green signal periods for directions with higher vehicle density, thereby optimizing the overall traffic flow.

- Automating the vehicle counting process and integrating it with signal optimization algorithms can significantly enhance traffic management systems. It enables authorities to implement intelligent and adaptive traffic control strategies that respond to real-time traffic conditions. This leads to improved traffic flow, reduced congestion, shorter travel times, and enhanced overall road safety.

In conclusion, the need to solve the vehicle counting problem stems from the demand for efficient traffic monitoring and management systems. By automating the counting process and considering vehicle density in signal optimization, traffic authorities can make informed decisions and implement targeted strategies to optimize traffic flow, reduce congestion, and improve overall road efficiency.

1.4 What do we propose to do in the work?

In the proposed work, we aim to develop and implement a vehicle counting system using the YOLOv8 pre-trained model. The main objectives of our work include:

1. **System Development:** We will develop a robust and efficient vehicle counting system by leveraging the power of YOLOv8. This involves fine-tuning the pre-trained model using a dataset containing annotated images or videos of vehicles, enabling it to accurately detect and classify trucks, cars, and motorbikes.
2. **Real-time Vehicle Detection:** Our system will utilize the YOLOv8 model to perform real-time vehicle detection. By processing live video streams or recorded footage, the system will be able to detect and localize vehicles of interest within the frames, providing bounding box coordinates and class labels for each detected vehicle.
3. **Vehicle Counting Logic:** We will implement a vehicle counting logic that accurately tracks and counts the detected vehicles. This may involve assigning unique IDs to each

vehicle, employing tracking algorithms to maintain consistency across frames, and utilizing appropriate techniques to handle occlusions and overlapping vehicles for accurate counting.

4. **Performance Optimization:** Our work will focus on optimizing the performance of the vehicle counting system. This includes exploring techniques such as multi-scale detection, non-maximum suppression, and other optimizations to improve the accuracy, efficiency, and speed of the system.
5. **Visualization and Reporting:** We will develop a user-friendly visualization module that displays the results of the vehicle counting system. This includes displaying the real-time count of different vehicle types, generating statistical reports, and providing visual representations of traffic flow patterns for easy interpretation and analysis.

Through our proposed work, we aim to provide a reliable and efficient vehicle counting solution using the YOLOv8 model, enabling accurate and real-time analysis of traffic volume, patterns, and vehicle types.

CHAPTER 2

PREVIOUS STUDIES

Sl. No.	Project Title	Outcomes	Shortcomings
1	Vehicle Detection and Counting using Deep Learning based YOLO and Deep SORT Algorithm for Urban Traffic Management System	The system provides a viable solution for estimating traffic density on roads, enabling the creation of smart traffic light signals that adapt their duration based on real-time vehicle density in specific lanes. This can contribute to more efficient traffic management and improved traffic flow in smart cities.	While the paper mentions the potential benefits of the system for traffic density estimation and infrastructural development, it does not elaborate on the practical implications or potential integration challenges with existing traffic management systems or city infrastructure. Consideration of such factors is important for the successful implementation and adoption of the proposed solution.
2	A real time vehicle counting based on adaptive tracking approach for highway videos	By achieving real-time processing, the proposed method offers the potential for implementation in intelligent systems for monitoring vehicle traffic flow. It contributes to the foundation of intelligent transportation systems by providing accurate and timely information on the number of vehicles on highways.	The abstract does not elaborate on the computational requirements or the computational efficiency of the proposed method. Understanding the computational demands and processing time required for real-time implementation is essential for assessing the practical feasibility of the approach.
3	Deep learning based Novel Vehicle Detection and Counting System	By leveraging computer vision technology, the proposed technique offers a potential solution for addressing the challenges of traffic management in smart cities. It demonstrates the potential to improve overall traffic flow, enhance road safety, and optimize traffic control measures.	The abstract does not mention the experimental setup or the datasets used for testing the identification technique. Further information on the diversity of the datasets, variations in lighting conditions, and potential challenges faced during testing would provide insights into the robustness and generalizability of the proposed technique.

4	Vehicle detection and counting system for real-time traffic surveillance	The developed vehicle detection and counting algorithm is implemented and tested on an embedded platform of smart cameras, indicating its potential for real-time applications and integration into smart traffic control systems.	The abstract does not provide specific details about the algorithmic approaches used for vehicle detection, counting, and road marking detection. Additional information on the methodologies, techniques, or machine learning models employed would enhance the understanding of the proposed algorithms and their limitations.
5	Vehicle Detection and Classification using Image processing	By employing the proposed algorithms, the system can monitor highways, detect accidents, identify unrighteous stoppage of vehicles, and track traffic rule violators. This functionality contributes to enhancing road safety measures and enforcing traffic regulations.	The abstract does not discuss the scalability or performance of the system in handling large-scale traffic scenarios or varying environmental conditions. Evaluating the system's effectiveness and efficiency in real-world settings with diverse traffic situations would contribute to assessing its practical viability.
6	Vehicle Counting for Traffic Management System using YOLO and Correlation Filter	The proposed video-based vehicle counting method demonstrates the effectiveness of utilizing handheld cameras in capturing highway traffic videos for accurate vehicle detection, tracking, and counting. By employing the YOLO (You Only Look Once) framework for object detection and correlation filters for tracking, the method achieves remarkable outcomes in terms of accuracy and speed.	The abstract does not provide details on the computational requirements or processing time required for the proposed method. Understanding the computational efficiency and resource utilization is crucial for assessing the practical feasibility and scalability of the approach.
7	A Low-Cost Real-Time Embedded Vehicle Counting and Classification System for Traffic Management Applications	The implementation of the low-cost system using the Embedded ARM-based platform ODROID XU-4 with the Ubuntu operating system	The abstract does not address any potential challenges or limitations related to the low-cost embedded ARM-based system. Factors such as

		offers a cost-effective solution for vehicle identification and classification. The algorithms, implemented in Python programming language, leverage the capabilities of the system to achieve an efficiency of 95.35% in vehicle detection, classification, and counting.	computational constraints, scalability, or real-time performance in different scenarios should be considered for a comprehensive evaluation of the system's practical applicability.
8	Vehicle Detection, Tracking and Counting	The low-cost nature of the camera-based algorithm offers a practical solution for traffic flow management, potentially making it accessible for implementation in metropolitan cities. The algorithm's efficacy provides an avenue for reducing unnecessary time wastage, minimizing congestion, and improving overall road safety.	The abstract does not provide specific details on the performance metrics or quantitative evaluation of the algorithm's efficacy. Additional information on the accuracy, precision, and scalability of the algorithm, as well as its performance in different traffic scenarios, would provide a more comprehensive understanding of its limitations and applicability.
9	Automatic Vehicle Counting for IoT based Smart Traffic Management System for Indian urban Settings	The utilization of Internet of Things (IoT) technology allows for the sharing of monitoring details to a remote controlling center anywhere in the city, enabling centralized traffic management and analysis. The performed experiments validate the effectiveness of the proposed IoT-based technology for efficient traffic management.	The abstract does not discuss potential limitations or challenges encountered during the implementation or testing of the framework. Considerations such as variations in lighting conditions, occlusions, or the impact of different vehicle sizes and types on the accuracy of vehicle counting should be addressed for a comprehensive evaluation.
10	Real-time vehicle detection and counting based on YOLO and DeepSORT	Experimental results demonstrate the superior performance of the proposed model, showing an improvement of at least 11% in average precision (AP) and 12% in AP50 for various real-world scenarios. The model operates at a real-time speed	The abstract does not mention the specific dataset used for evaluation, which limits the ability to assess the generalizability and applicability of the proposed model to different real-world scenarios. Details regarding dataset diversity, size, and

		of approximately 32 frames per second (FPS), indicating its practical viability for real-time vehicle detection and counting applications.	representativeness of various field scenarios would enhance the validity of the experimental results.

CHAPTER 3

REQUIREMENT SPECIFICATION

3.1 Hardware Requirements:

- Processor: The system should have a processor with sufficient computational power to train and run deep learning models effectively. A multi-core processor, such as Intel Core i5 or higher, is recommended.
- Memory: Sufficient RAM is essential for handling large datasets and running memory-intensive operations. A minimum of 8GB RAM is recommended, but higher capacity, such as 16GB or more, is preferable for better performance.

3.2 Software Requirements:

- **Programming Language:** Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability with its clean and concise syntax, making it a popular choice for beginners and experienced developers alike.

  Python supports various programming paradigms, including procedural, object-oriented, and functional programming styles. It has a vast standard library that provides ready-to-use modules for a wide range of tasks, enabling developers to accomplish complex tasks efficiently.
- **IDE:** Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, visualizations, explanatory text, and more. It provides an interactive computing environment where you can combine code execution, data exploration, and narrative text in a single document.
- **Operating System:** An operating system (OS) is a software that acts as an interface between the computer hardware and user applications. It manages computer resources, provides services to applications, and facilitates user interaction. Python, as a programming language, can interact with the underlying operating system through various modules and libraries.[We have used Windows 11 to conduct this project].

- **Web Browser:** Brave is a free and open-source web browser built with a focus on privacy and security. It is designed to provide a fast and secure browsing experience while offering enhanced privacy features. Brave can be used to open Jupyter Notebook, which is a web-based interactive computing environment for creating and sharing documents that contain live code, visualizations, and explanatory text.

3.3 Technology:

The proposed technology for the age classification system is as follows:

- **YOLOv8:** The YOLOv8 model is a state-of-the-art deep learning architecture used for object detection tasks. "YOLO" stands for "You Only Look Once," emphasizing its ability to detect objects in an image in a single forward pass. YOLOv8 is an improved version of the YOLO (You Only Look Once) family of models, designed to provide faster and more accurate Face detection.
- **Computer Vision:** Computer vision is a field of artificial intelligence and computer science that focuses on enabling computers to gain a high-level understanding of visual data, such as images or videos. In the context of your project, computer vision techniques are employed to analyze and process images of vehicles captured on roads and highways.
- **OpenCV:** The OpenCV (Open Source Computer Vision) library is an open-source computer vision and machine learning software library. It provides a comprehensive set of tools and functions for image and video processing, object detection and tracking, feature extraction, and more. In the context of your project, the OpenCV library is utilized to handle various computer vision tasks related to vehicle detection and analysis.

CHAPTER 4

METHODOLOGY

4.1 Flowchart:

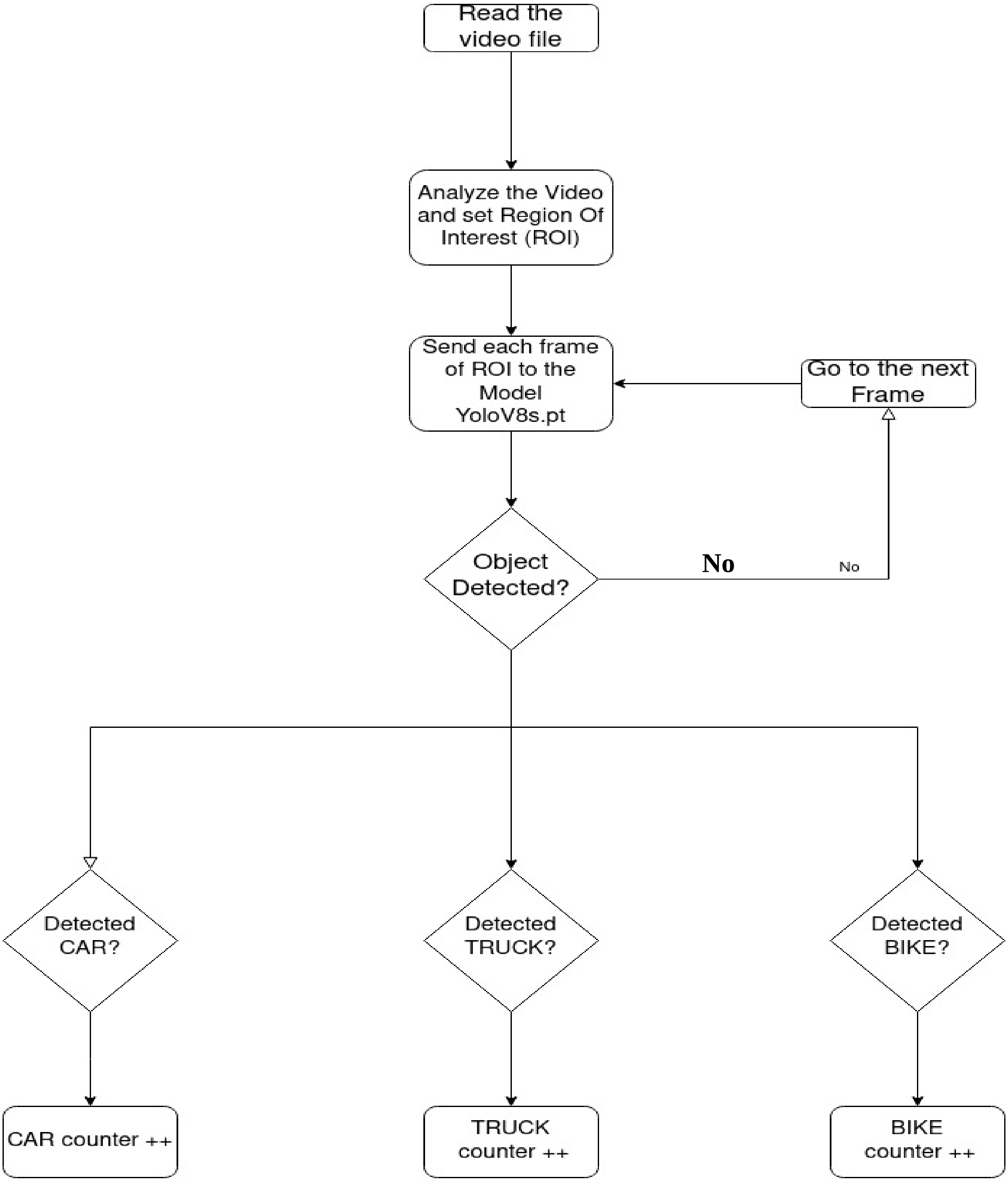

Fig.2 Flow chart of the project

4.2 Explanation of the methodology:

The methodology for developing the vehicle counting system using the YOLOv8 pre-trained model involves several key steps. Here is an explanation of the overall methodology:

1. Dataset Preparation: The initial step involves compiling a dataset consisting of videos capturing vehicles in specific scenarios. The dataset should primarily focus on two main categories: videos of vehicles on highways and videos of vehicles at or approaching traffic signals. These videos should be recorded during daylight to ensure optimal lighting conditions.

2. Model Selection and Configuration: For the vehicle counting system, we have opted to utilize the YOLOv8 model as our base model. We experimented with different variations of the YOLOv8 architecture, including YOLOv8m, YOLOv8s, and YOLOv8L.

During our evaluation process, we observed that YOLOv8m exhibited relatively lower accuracy while requiring less computational power. On the other hand, YOLOv8L demonstrated significantly higher accuracy but demanded substantial computational resources. Considering a balance between accuracy and computational efficiency, we selected YOLOv8s as the optimal model for our vehicle counting task.

By choosing YOLOv8s, we strike a favorable compromise between accuracy and computational demands. This model variant provides satisfactory accuracy while maintaining reasonable computational requirements, making it well-suited for real-time vehicle detection and counting.

To configure the YOLOv8s model, we initialize it with pre-trained weights obtained from extensive training on large-scale datasets. These pre-trained weights serve as a starting point for our specific vehicle counting task. Additionally, we adjust the model's architecture and hyperparameters to ensure optimal performance in accurately detecting and classifying vehicles in our chosen scenarios.

By leveraging the YOLOv8s model, we aim to achieve reliable and efficient vehicle counting capabilities, enabling traffic authorities to obtain real-time and accurate vehicle count data for effective traffic management and planning.

3. Adjusting the Region of Interest (ROI):

In the vehicle counting system, it is crucial to define a specific region of interest (ROI) within the captured video frames. The ROI represents the area where vehicle detection and counting will be performed. Expert analysis and domain knowledge play a vital role in determining the optimal ROI.

To determine the ROI, experts carefully analyze the video footage and identify the key areas where vehicles are expected to appear. This involves considering factors such as the road layout, traffic patterns, and the specific objectives of the vehicle counting system. The selected ROI should encompass the relevant lanes or areas of interest, excluding irrelevant regions to minimize unnecessary computational processing.

The expert analysis takes into account various factors, including the positioning of the camera and the expected trajectory of vehicles. For example, in a highway scenario, the ROI might cover the lanes where vehicles are expected to be in motion, excluding areas such as sidewalks or surrounding landscapes. In the case of traffic signals, the ROI could be set to encompass the approaching lanes leading to the signal.

By adjusting the ROI, we effectively limit the scope of vehicle detection and counting, focusing computational resources on the areas of highest interest. This helps streamline the processing pipeline and improve overall efficiency.

Once the ROI is defined, the subsequent stages of the vehicle counting system will exclusively analyze the vehicles within this region. The YOLOv8s model will be applied to detect and classify vehicles within the ROI, providing accurate count information for the designated areas of interest.

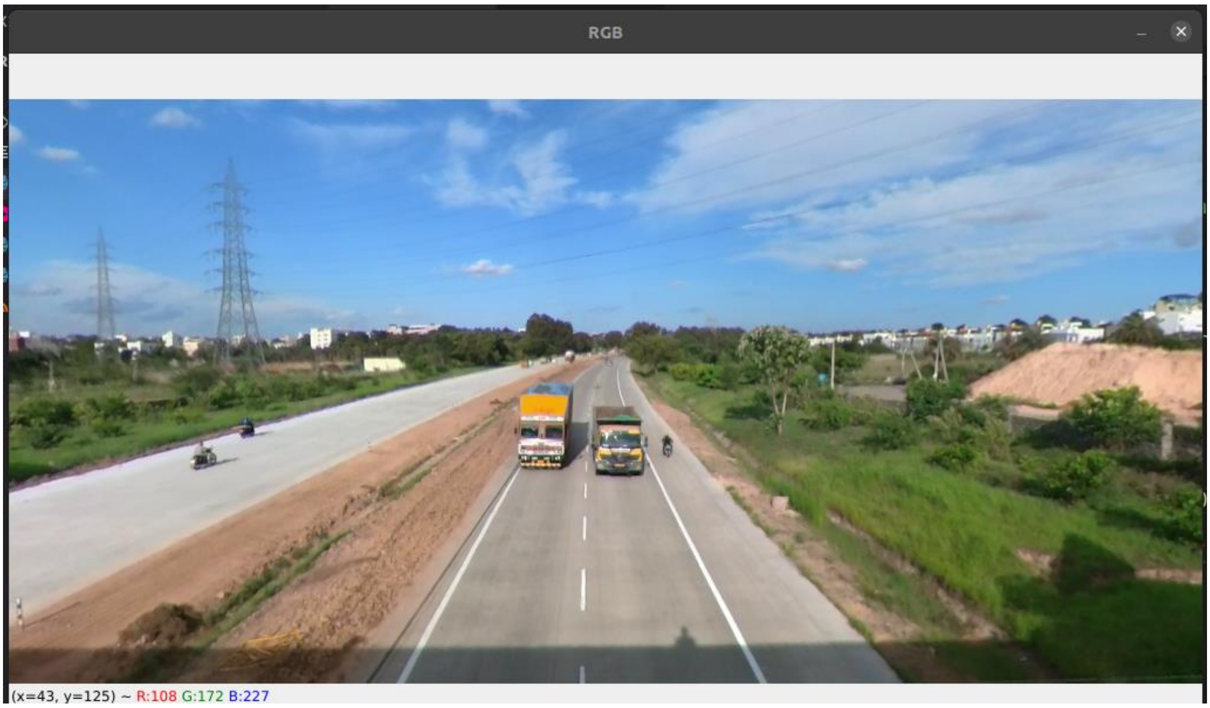

Fig.3 Main frame from the video

Fig.4 Region Of Interest(ROI)

4. Vehicle Detection using YOLOv8s model:

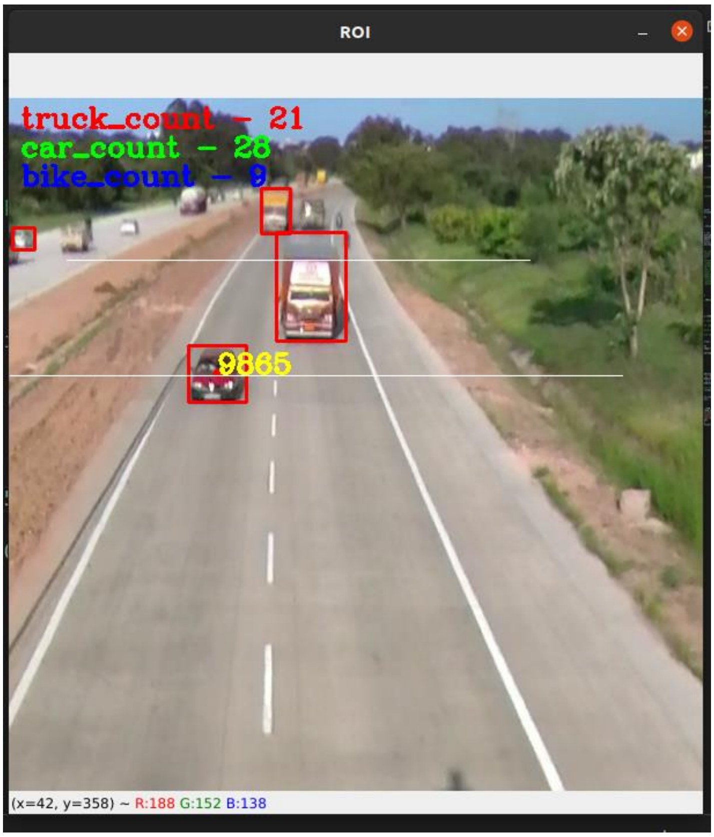

Fig.5 Model returns ID and Class

After adjusting the Region of Interest (ROI), the next step in the vehicle counting system involves sending the selected ROI to the YOLOv8s model for prediction. The model utilizes its object detection capabilities to identify and classify vehicles within the ROI. The output of the model includes the predicted class labels, unique identifiers, and corresponding bounding boxes for each detected vehicle instance.

For each detected vehicle, the model generates a class label indicating the type of vehicle, which can be useful for subsequent analysis and counting. Additionally, a unique identifier is assigned to each vehicle, enabling tracking and monitoring of individual vehicles over time. The bounding boxes encapsulate the precise location and extent of each detected vehicle within the ROI.

With the vehicle detection and prediction stage complete, we have successfully extracted the essential details of each identified vehicle within the ROI. This lays the foundation for accurate vehicle counting and facilitates subsequent stages in the vehicle counting system, such as data aggregation and analysis.

5. Vehicle Count logic :

In this section, we focus on determining the count of each vehicle class detected within the Region of Interest (ROI). To achieve this, we employ a logical approach utilizing lists to store the unique identifiers (IDs) of vehicles belonging to each specific class.

For each class, we create a separate list to hold the IDs of the corresponding vehicles. For example, if we have three classes - cars, trucks, and motorcycles - we will create three separate lists to store the IDs of vehicles classified under each respective class.

Next, we initialize three additional lists to keep track of the count of vehicles for each class. These count lists will store the unique IDs obtained from the respective class lists. By utilizing the concept of uniqueness, we ensure that each vehicle is counted only once, even if it appears multiple times within the ROI.

The logic for counting the vehicles involves iterating through the class lists, extracting the unique IDs, and populating the count lists accordingly. This process ensures that we have accurate and non-duplicated counts for each vehicle class.

5. Performance Optimization:

In the context of the vehicle counting system, there are various approaches that can be employed to optimize its performance. Two potential strategies for optimization include utilizing the YOLOv8L model and ensuring correct camera angle using a high-quality camera.

A. **YOLOv8L Model:** One optimization technique involves using the YOLOv8L model. This model variant offers higher accuracy in vehicle detection and classification compared to other variations such as YOLOv8s or YOLOv8m. By leveraging the increased capacity and complexity of the YOLOv8L model, we can enhance the accuracy of vehicle identification and improve the overall performance of the system.
B. **Correct Camera Angle and High-Quality Camera:** Another important aspect of

optimization is ensuring the correct camera angle and utilizing a high-quality camera. The camera angle should be set in a way that captures the desired field of view, maximizing the visibility of vehicles within the ROI. By adjusting the camera angle appropriately, we can minimize occlusions and obtain clear and unobstructed views of vehicles, leading to improved detection accuracy.

6. Visualization and Reporting: The system provides a visual interface or reporting module to display the results of the vehicle counting process. This includes real-time counts of different vehicle types, graphical representations of traffic flow patterns, and statistical reports that summarize the vehicle count data.

By following this methodology, the vehicle counting system based on YOLOv8 can accurately detect, classify, and count vehicles in real-time, providing valuable insights into traffic flow and pattern.

CHAPTER 5

RESULTS

5.1 Outcomes from the methodology

The outcomes from the methodology used in this project yielded significant results in accurately counting vehicles on roads and highways. By employing the pre-trained YOLOv3 model for object detection, we were able to successfully identify and classify different vehicle classes, including cars, bikes, and trucks, from the captured images.

The vehicle counting mechanism implemented as part of the methodology proved to be effective in accurately determining the number of vehicles for each class separately. The counting algorithm utilized the detected objects and their classifications to calculate the total count, providing valuable data for traffic authorities and researchers.

The results obtained from the vehicle counting methodology demonstrated its reliability and accuracy. The automated approach eliminated the need for manual counting, reducing human error and saving considerable time and effort. The real-time nature of the system allowed for instant vehicle count updates, providing up-to-date information for traffic management purposes.

Additionally, the methodology showcased the potential of using deep learning-based models like YOLOv8s for vehicle detection and classification tasks. The model exhibited robust performance in detecting vehicles, even in complex traffic scenarios and varying environmental conditions.

The outcomes from the methodology have significant implications for traffic management and infrastructure planning. Accurate vehicle counting data enables traffic authorities to make informed decisions regarding traffic flow optimization, congestion reduction, and signal timing adjustments. The real-time insights provided by the methodology can aid in developing effective strategies for managing traffic patterns, identifying congestion hotspots, and improving overall road safety.

In conclusion, the outcomes from the employed methodology demonstrate the successful implementation of an automated vehicle counting system using the YOLOv3 model. The methodology's accuracy, reliability, and real-time capabilities contribute to enhancing traffic management systems, allowing for data-driven decision-making and improved efficiency on roads and highways.

CHAPTER 6

IMPLEMENTATION OUTCOMES

6.1 Types of classes or Objects:

Fig.6 Motorbikes

Fig.7 Car

Fig.8 Trucks

6.2 Model output:

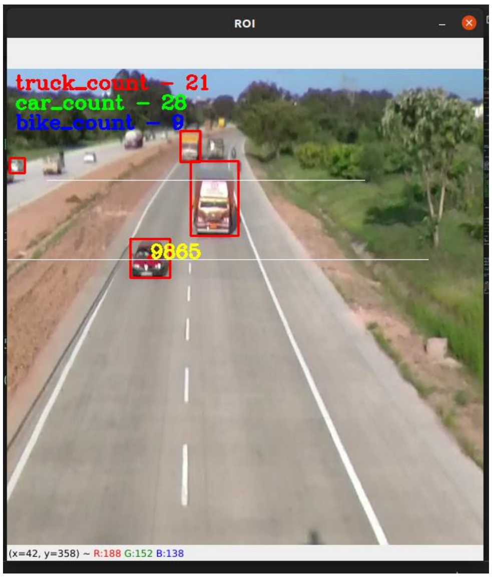

Fig.9 Model Output

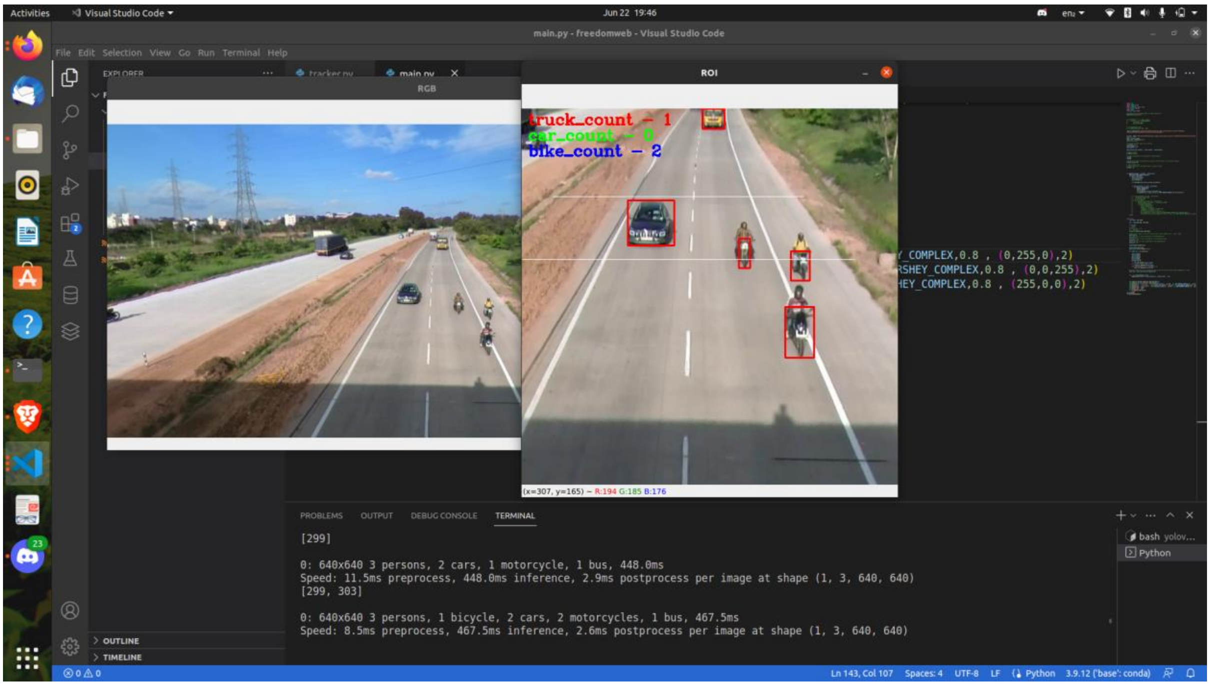

Fig.10 Start of the video.

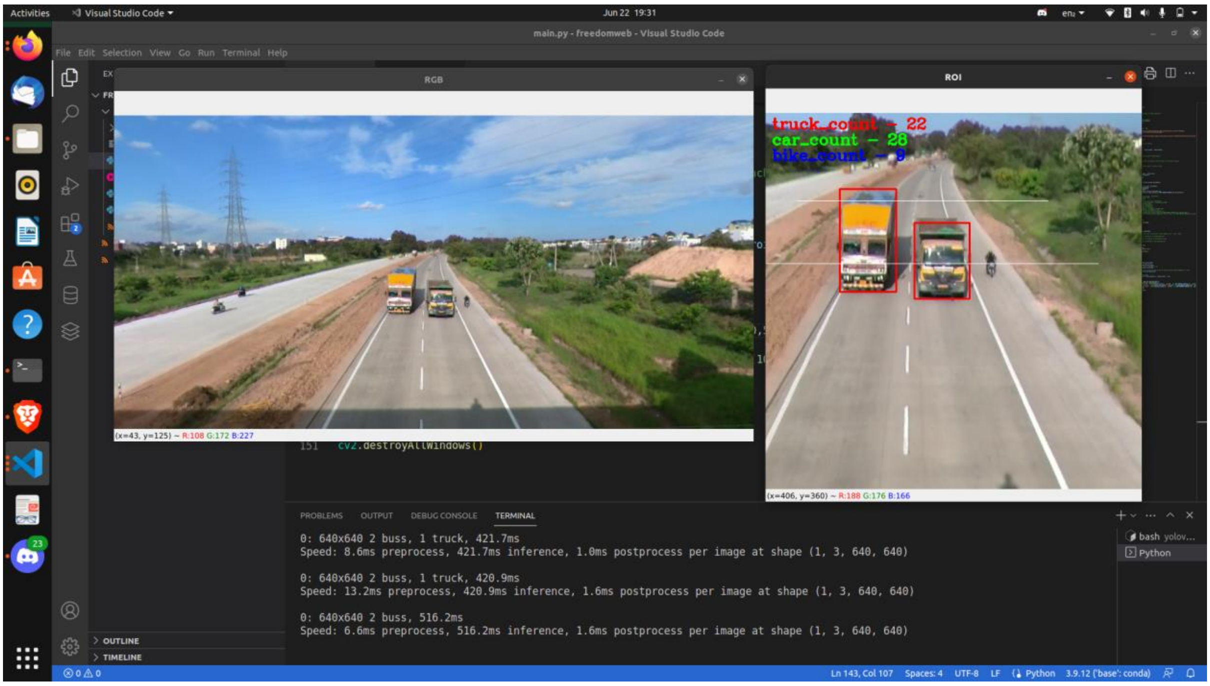

Fig.11 The count of each class displayed.

6.3 Code:

```
import cv2
import pandas as pd
import numpy as np
from ultralytics import YOLO
import time
from math import dist

# we make use of pretrained model for object detection
model=YOLO('yolov8s.pt')

cap=cv2.VideoCapture('/home/monish/Desktop/digital_image_processing/mini_pr
oject/freedomweb/yolov8counting-trackingvehicles-main/nice_road_video.mp4')

my_file =
open("/home/monish/Desktop/digital_image_processing/mini_project/freedomweb
/yolov8counting-trackingvehicles-main/coco.txt", "r")
data = my_file.read()
class_list = data.split("\n")

# counters for each class of our interest
car_counter = []
truck_counter = []
bike_counter = []

main_counter=[car_counter , truck_counter , bike_counter]

# object tracker
tracker=Tracker()

# line y-coordinates for filtering the tracked objects
cy1=140
cy2=240

# based on he speed of the vehicle we need to set the offset for efficient
tracking
offset_lst=[3,4,5]

def updater(category , counter , offset,roi):
      bbox_id=tracker.update(category)
      for bbox in bbox_id:
      x3,y3,x4,y4,id=bbox
      cx=int(x3+x4)//2
      cy=int(y3+y4)//2

      cv2.rectangle(roi,(x3,y3),(x4,y4),(0,0,255),2)

      if cy2<(cy+offset) and cy2 > (cy-offset):
            if counter.count(id)==0:
                  counter.append(id)
                  print(counter)
                  cv2.circle(roi,(cx,cy),4,(0,0,255),-1)

      cv2.putText(roi,str(id),(cx,cy),cv2.FONT_HERSHEY_COMPLEX,0.8,(0,255,2
55),2)
```

```
      return


while True:
      ret,frame = cap.read()

      roi = frame[400:1000 ,700:1300]

      if not ret:
      break
      frame=cv2.resize(frame,(1020,500))

      # main_list is used to store all the sub list of each category (car ,
truck , bike)
      main_list=[]
      # car sub list to store coordinates of the detected cars
      car_list = []
      # truck sub list to store coordinates of the detected trucks
      truk_list = []
      # bike sub list to store coordinates of the detected bike
      bike_list = []

      results=model.predict(roi)

      a=results[0].boxes.data
      px=pd.DataFrame(a).astype("float")

      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 :
            car_list.append([x1,y1,x2,y2])
      elif 'truck' in c or 'bus' in c:
            truk_list.append([x1,y1,x2,y2])
      elif 'motorcycle' in c or 'bicycle' in c:
            bike_list.append([x1,y1,x2,y2])

      # add all these sub list to a main list and so that we can track each
vehicle category in a loop
      main_list = [car_list,truk_list,bike_list]
      for i in range(len(main_list)):
      updater(main_list[i] , main_counter[i] , offset_lst[i] , roi)

      cv2.line(roi,(50,cy1),(450,cy1),(255,255,255),1)
      cv2.line(roi,(1,cy2),(530,cy2),(255,255,255),1)
      cv2.putText(roi , ('car_count - ')+str(len(car_counter)) , (10,50) ,
cv2.FONT_HERSHEY_COMPLEX,0.8 , (0,255,0),2)
      cv2.putText(roi , ('truck_count - ')+str(len(truck_counter)) ,
(10,25) , cv2.FONT_HERSHEY_COMPLEX,0.8 , (0,0,255),2)
      cv2.putText(roi , ('bike_count - ')+str(len(bike_counter)) , (10,75)
, cv2.FONT_HERSHEY_COMPLEX,0.8 , (255,0,0),2)
      cv2.imshow("RGB", frame)
```

```
      cv2.imshow("ROI", roi)
      if cv2.waitKey(1)&0xFF==27:
      break
cap.release()
cv2.destroyAllWindows()
```

CHAPTER 7

CONCLUSION

In conclusion, this project successfully addressed the vehicle counting problem through the application of digital image processing techniques and deep learning-based object detection. By utilizing the pre-trained YOLOv8 model, accurate identification and classification of vehicles, including cars, bikes, and trucks, were achieved.

The developed methodology provided reliable and real-time vehicle count data, eliminating the need for manual counting and reducing human errors. The automated system proved to be efficient and effective in counting vehicles on roads and highways, offering valuable insights for traffic management and infrastructure planning.

The project's outcomes demonstrated the potential of digital image processing in solving real-world challenges in traffic management. The utilization of the YOLOv8 model showcased its robust performance in detecting vehicles under various traffic scenarios and environmental conditions.

The results obtained from the project have practical implications for traffic authorities and researchers. Accurate vehicle counting data enables informed decision-making for traffic flow optimization, congestion reduction, and signal timing adjustments.

Furthermore, the project opens avenues for future enhancements and research in the field of vehicle counting and traffic management. Exploring advanced deep learning models, refining the counting algorithm, and integrating the system with intelligent traffic control systems can further enhance its accuracy and effectiveness.

In conclusion, this project's successful implementation of an automated vehicle counting system using digital image processing techniques contributes to the field of traffic management, providing a reliable and efficient solution for real-time vehicle counting. The outcomes have significant implications for optimizing traffic flow, reducing congestion, and improving overall road safety.

CHAPTER 8

FUTURE SCOPE AND ENHANCEMENT

The vehicle counting system presented in this project lays a strong foundation for further enhancements and future developments. The system can be expanded and improved in several ways to address additional functionalities and provide more comprehensive traffic analysis. Some potential future scope and enhancement areas include:

1. **Speed Detection:** The system can be extended to incorporate speed detection capabilities. By leveraging advanced image processing techniques, such as optical flow or frame differencing, it becomes possible to estimate the speed of vehicles within the captured video footage. This information can be valuable for monitoring speed limits, identifying potential traffic violations, and analyzing traffic flow dynamics.
2. **Helmet Detection:** Another potential enhancement is the integration of helmet detection functionality. By employing computer vision algorithms and deep learning models, the system can identify whether motorbike riders are wearing helmets or not. This feature can contribute to enhanced road safety measures and enforcement of helmet-wearing regulations.
3. **Bi-Directional Vehicle Counting:** In the case of highways or roads with multiple lanes, the system can be expanded to calculate the count of vehicles moving in both directions. By placing cameras at optimal positions and utilizing advanced tracking algorithms, the system can accurately track and count vehicles in each lane separately, providing valuable insights into traffic patterns and congestion management.
4. **Traffic Flow Analysis:** The vehicle counting system can be augmented to perform more comprehensive traffic flow analysis. By incorporating advanced data analytics techniques, the system can generate real-time traffic flow visualizations, identify traffic bottlenecks, and provide suggestions for traffic signal optimization and route planning.
5. **Integration with Smart City Infrastructure:** The system can be integrated with existing smart city infrastructure and traffic management systems. By sharing vehicle count and other relevant data with traffic authorities, the system can contribute to real-time traffic monitoring, adaptive traffic signal control, and intelligent traffic management strategies.

6. **Integration with Vehicle Recognition Systems:** The vehicle counting system can be combined with vehicle recognition systems, such as license plate recognition or vehicle make and model identification. This integration enables comprehensive vehicle tracking, enables identification of specific vehicles of interest, and enhances law enforcement and security measures.

By pursuing these future enhancements, the vehicle counting system can evolve into a more advanced and comprehensive solution for traffic monitoring and management. These developments contribute to improved road safety, enhanced traffic efficiency, and the establishment of smarter and more sustainable transportation systems.

CHAPTER 9

REFERENCES

1. Tourani, A. and Shahbahrami, A., 2015, March. Vehicle counting method based on digital image processing algorithms. In *2015 2nd International conference on pattern recognition and image analysis (IPRIA)* (pp. 1-6). IEEE.
2. Bhaskar, P.K. and Yong, S.P., 2014, June. Image processing based vehicle detection and tracking method. In *2014 international conference on computer and information sciences (ICCOINS)* (pp. 1-5). IEEE.
3. Mallikalava, V., Yuvaraj, S., Vengatesan, K., Kumar, A., Punjabi, S. and Samee, S., 2020, August. Theft vehicle detection using image processing integrated digital signature based ECU. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 913-918). IEEE.
4. Chowdhury, P.N., Ray, T.C. and Uddin, J., 2018, February. A vehicle detection technique for traffic management using image processing. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)* (pp. 1-4). IEEE.
5. Ishida, S., Mimura, K., Liu, S., Tagashira, S. and Fukuda, A., 2016, June. Design of simple vehicle counter using sidewalk microphones. In *Proc. ITS EU Congress. EU-TP0042* (pp. 1-10).
6. Chandrika, R.R., Ganesh, N.G., Mummoorthy, A. and Raghunath, K.K., 2019, September. Vehicle detection and classification using image processing. In *2019 International Conference on Emerging Trends in Science and Engineering (ICESE)* (Vol. 1, pp. 1-6). IEEE.
7. Pang, C.C.C., Lam, W.W.L. and Yung, N.H.C., 2007. A method for vehicle count in the presence of multiple-vehicle occlusions in traffic images. *IEEE Transactions on Intelligent Transportation Systems*, *8*(3), pp.441-459.
8. Kalyan, S.S., Pratyusha, V., Nishitha, N. and Ramesh, T.K., 2020, November. Vehicle detection using image processing. In *2020 IEEE International Conference for Innovation in Technology (INOCON)* (pp. 1-5). IEEE.
9. Moranduzzo, T. and Melgani, F., 2013. Automatic car counting method for unmanned aerial vehicle images. *IEEE Transactions on Geoscience and remote sensing*, *52*(3), pp.1635-1647.
10. Michalopoulos, Panos G. "Vehicle detection video through image processing: the autoscope system." *IEEE Transactions on vehicular technology* 40, no. 1 (1991): 21-29.