OBJECT DETECTION USING DEEP LEARNING

PROJECT REPORT

21AD1513- INNOVATION PRACTICES LAB

Submitted by

ABDUL FADHIL. M	**211422243007**
ASWIN KUMAR. A	**211422243035**
BHARATHRAJ. S	**211422243046**

in partial fulfillment of the requirements for the award of degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123

ANNA UNIVERSITY: CHENNAI-600 025

October, 2023

BONAFIDE CERTIFICATE

Certified that this project report titled **"OBJECT OBJECT DETECTION USING DEEP LEARNING"**is the bonafide work of **ABDUL FADHIL. M (211422243007), ASWIN KUMAR. A (211422243035), BHARATHRAJ. S (211422243046)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

INTERNAL GUIDE
P. MISBA MARYBAI
M.Tech,
Assistant Professor,
Department of AI &DS.

HEAD OF THE DEPARTMENT
Dr.S.MALATHI M.E., Ph.D
Professor and Head,
Department of AI & DS.

Certified that the candidate was examined in the Viva-Voce Examination held on ………………………

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

This project focuses on the development of an object detection system for real-time traffic violation monitoring using deep learning techniques. By leveraging the YOLO (You Only Look Once) model, a highly efficient object detection algorithm, the system is designed to detect and classify vehicles (cars, buses, trucks, and bikes) in traffic videos. The project aims to automate the detection process, reducing the need for manual annotation and increasing the efficiency of traffic surveillance systems. The model is trained using a custom dataset, and its performance is evaluated using various metrics such as accuracy and detection speed. The system is implemented in a Windows environment using Python, OpenCV, and Ultralytics YOLO, with the output displayed through both a web-based interface and local video processing. Future enhancements include integrating vehicle count and detailed analytics visualizations, such as pie charts, bar graphs, and line charts, for comprehensive traffic analysis. This solution has significant potential for application in automated traffic monitoring, enforcement of traffic regulations, and improving overall road safety.

Keywords: Object Detection, YOLO, Traffic Surveillance, Vehicle Detection, Computer Vision, Real-time Processing, Traffic Violations, Vehicle Counting.

ACKNOWLEDGEMENT

I also take this opportunity to thank all the Faculty and Non-Teaching Staff Members of Department of Computer Science and Engineering for their constant support. Finally I thank each and every one who helped me to complete this project. At the outset we would like to express our gratitude to our beloved respected Chairman, **Dr.Jeppiaar M.A.,Ph.D,** Our beloved correspondent and Secretary **Mr.P.Chinnadurai M.A., M.Phil., Ph.D.,** and our esteemed director for their support.

We would like to express thanks to our Principal, **Dr. K. Mani M.E., Ph.D.,** for having extended his guidance and cooperation.

We would also like to thank our Head of the Department, **Dr.S.Malathi M,E.,Ph.D.,** of Artificial Intelligence and Data Science for her encouragement.

Personally we thank **Mrs.P.MISBA MARYBAI M.Tech, Assistant Professor,** Department of Artificial Intelligence and Data Science for the persistent motivation and support for this project, who at all times was the mentor of germination of the project from a small idea.

We express our thanks to the project coordinator **Mrs.V.REKHA, M.E, Assistant Professor** in Department of Artificial Intelligence and Data Science for their Valuable suggestions from time to time at every stage of our project.

Finally, we would like to take this opportunity to thank our family members, friends, and well-wishers who have helped us for the successful completion of our project.

We also take the opportunity to thank all faculty and non-teaching staff members in our department for their timely guidance in completing our project.

ABDUL FADHIL. M

ASWIN KUMAR. A

BHARATHRAJ. S

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABBREVIATIONS	MEANING
• UI	User Interface
• API	Application Programming Interface
• YOLO	You Only Look Once
• SORT	Simple Online and Realtime Tracking
• DeepSORT	Deep Learning-based SORT
• AI	Artificial Intelligence
• CSV	Comma Separated Values
• HTTP	Hypertext Transfer Protocol
• JSON	JavaScript Object Notation
• GPU	Graphics Processing Unit
• COCO	Common Objects in Context

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OBJECT DETECTION

Object detection is a key technology in computer vision that aims to identify and locate objects within images or videos. It enables the classification and localization of various entities, such as vehicles, people, animals, and other objects, within a frame. This process involves detecting the presence of an object and drawing a bounding box around it to show its location. The ability to detect objects accurately and in real-time has become crucial in many applications, including autonomous driving, surveillance, retail analytics, and, as in our project, traffic monitoring.

Object detection combines several techniques from deep learning and machine learning to recognize and differentiate between classes of objects in visual data. With advancements in convolutional neural networks (CNNs) and large annotated datasets, object detection models like YOLO (You Only Look Once) have achieved remarkable accuracy and speed, making them suitable for real-time applications.

1.2 DEEP LEARNING AND COMPUTER VISION

Deep learning is a branch of machine learning that focuses on neural networks with multiple layers (often called deep neural networks). It has transformed computer vision by enabling models to learn from vast amounts of data and automatically extract hierarchical features, which allows for high precision in tasks like image classification, object detection, and segmentation.

In computer vision, deep learning models, particularly CNNs, are commonly used due to their ability to capture spatial features from images. These models are trained to recognize patterns such as edges, textures, and shapes, which can then be used to identify specific objects. By stacking multiple convolutional layers, CNNs build complex feature hierarchies that improve their performance on visual tasks. Libraries like TensorFlow and PyTorch, along with the computational power of GPUs, have further accelerated the development and deployment of deep learning-based object detection models.

1.3 TRAFFIC VIOLATION DETECTION USING OBJECT DETECTION

Traffic violation detection is an important application of object detection technology, particularly in urban areas where automated monitoring can enhance road safety and regulation enforcement. By applying object detection to traffic videos, it is possible to identify and track vehicles, classify them by type, and monitor their movements to detect potential violations like speeding, lane violations, and running red lights.

In this project, we focus on using the YOLO model, a popular real-time object detection algorithm, to detect vehicles such as cars and buses in traffic videos. The objective is to count these vehicles accurately and provide analytics that can support traffic management and law enforcement. By automating the detection process, the system can reduce the need for manual monitoring and increase the efficiency of traffic regulation.

1.4 OBJECTIVES OF THE PROJECT

The main objectives of this project are as follows:

1. To develop a deep learning-based system capable of detecting vehicles in real-time within traffic video feeds.
2. To perform vehicle counting and generate analytics, such as vehicle type and density, that can aid traffic authorities in understanding traffic flow patterns.
3. To explore the potential for future applications in traffic violation detection, allowing for automated alerts and improved traffic safety.

1.5 ARCHITECTURE DIAGRAM

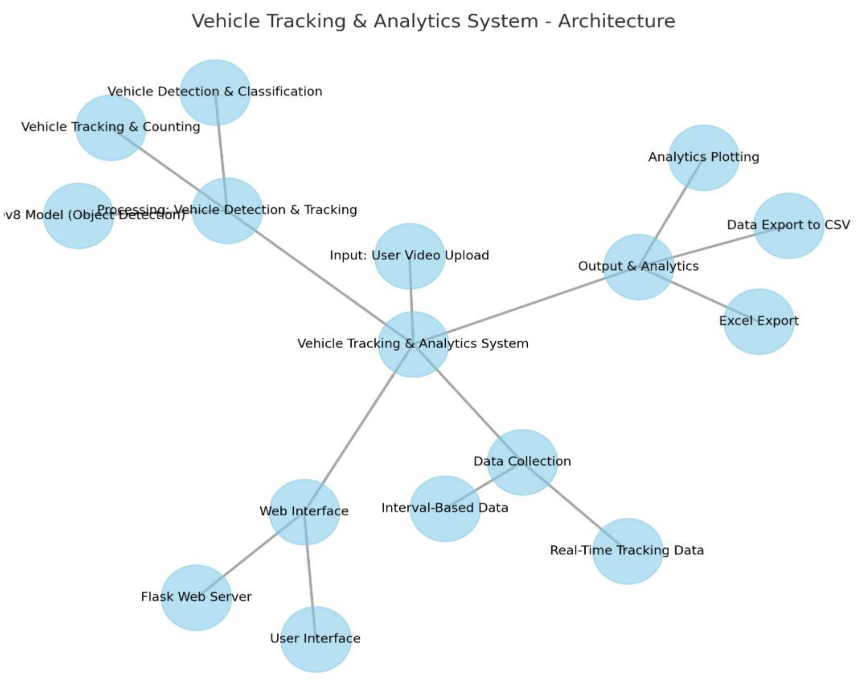

Fig 1.1: Architecture diagram of Object Detection for Vechile Tracking

1.5 COMPONENTS OF OBJECT DETECTION SYSTEM

An object detection system typically consists of the following components:

1. Data Collection: Gathering traffic video footage for training and testing the model.
2. Preprocessing: Preparing the data by labeling objects and converting video frames into a format suitable for model input.
3. Model Selection: Choosing a deep learning model such as YOLO, optimized for real-time object detection.
4. Training and Testing: Training the model on labeled data and testing its performance on unseen data.
5. Deployment: Implementing the trained model for real-time detection and counting in traffic video feeds.

1.6 APPLICATIONS OF OBJECT DETECTION IN TRAFFIC MANAGEMENT

Object detection technology has a wide range of applications in traffic management, including:

1. Real-Time Traffic Monitoring: Providing insights on traffic density and patterns in real-time.
2. Violation Detection: Identifying traffic violations, which can improve road safety and streamline law enforcement.
3. Smart City Development: Supporting the development of smart transportation systems by providing data for city planning and infrastructure development.

4. Analytics for Public Safety: Using vehicle detection data to inform public safety measures, optimize road usage, and reduce congestion.

CHAPTER 2

LITERATURE REVIEW

A literature review serves as an exploration of the existing research in a particular domain. It provides a comprehensive understanding of the current knowledge, findings, theories, and methodologies, situating the research within the broader academic context. This chapter reviews key works in the domain of object detection using deep learning, focusing on advancements in methodologies, applications, and challenges related to vehicle detection in video surveillance and real-time traffic analysis.

2.1 YOLO: Real-Time Object Detection

YOLO (You Only Look Once) is a state-of-the-art, real-time object detection system that is widely used in applications requiring fast, efficient detection of objects in images and video streams. YOLO's architecture enables it to predict bounding boxes and class probabilities for every object in an image simultaneously, making it particularly suitable for real-time applications. The model divides the input image into a grid and uses convolutional layers to predict the class and location of objects within that grid. YOLO has been adopted in various fields, including autonomous vehicles, surveillance, and traffic monitoring, due to its speed and accuracy.
Author: Joseph Redmon, Santosh Divvala, Ross B. Girshick, Ali Farhadi
Year: 2016

2.2 Faster R-CNN: Region-based Convolutional Neural Networks for Object Detection

Faster R-CNN, an evolution of the region-based convolutional neural network (R-CNN) family, introduced a Region Proposal Network (RPN) to

improve the speed and accuracy of object detection. Unlike previous models, Faster R-CNN generates region proposals directly from the image, enabling it to overcome the slow performance of traditional methods. This technique has been widely applied in traffic analysis and autonomous driving due to its robustness and ability to detect multiple classes of objects.
Author: Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun
Year: 2015

2.3 SSD: Single Shot Multibox Detector

The Single Shot Multibox Detector (SSD) is another influential deep learning-based object detection algorithm that balances speed and accuracy. Unlike YOLO, SSD operates by detecting objects at different scales and aspect ratios, offering better performance in scenarios where objects appear in varying sizes. SSD has been particularly effective in detecting small vehicles and pedestrians in traffic video surveillance, making it ideal for urban traffic monitoring systems.
Author: Wei Liu, Dragomir Anguelov, Dumbin Erhan, Cheng-Yang Fu, Alexander C. Berg
Year: 2016

2.4 Vehicle Detection Using Deep Learning for Traffic Monitoring

Recent advancements in deep learning have significantly improved vehicle detection in traffic video surveillance systems. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are often used for tasks such as vehicle classification, tracking, and counting. These techniques enable the extraction of features such as vehicle shape, size, and movement, contributing to more accurate traffic flow analysis. The use of these methods has been shown to reduce the computational load while maintaining high

detection accuracy.
Author: Zhenwei Shi, Lei Zhang, Wei Liu, Meng Wang
Year: 2018

2.5 Object Detection and Tracking in Video Surveillance Systems

Video surveillance for object detection and tracking in urban environments is crucial for smart city initiatives. Object tracking algorithms are used alongside detection systems like YOLO and Faster R-CNN to track vehicles, pedestrians, and other objects across video frames. Multiple object tracking (MOT) algorithms, such as SORT and DeepSORT, are combined with detection models to enhance the ability of surveillance systems to track vehicles over long periods, even under complex conditions such as occlusion and motion blur.
Author: Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, Ben Upcroft
Year: 2016

2.6 Real-Time Traffic Violation Detection Using Computer Vision

Detecting traffic violations, such as running red lights or speeding, has become an essential component of modern traffic monitoring systems. Deep learning models like YOLO, combined with computer vision techniques, have been successfully applied to detect such violations in real-time. These systems often incorporate vehicle detection models with license plate recognition (LPR) and speed estimation algorithms to monitor traffic behavior and enforce road safety laws automatically.
Author: Lijun Zhang, Yifan Zhang, Yu Zheng, Xinyu Zhao, Xue Bai
Year: 2020

2.7 Challenges in Real-Time Vehicle Detection and Counting

Real-time vehicle detection and counting remain challenging tasks, particularly under various lighting conditions, occlusions, and highly dynamic traffic environments. One of the primary challenges is ensuring high accuracy and speed while minimizing false positives. Models must be optimized to perform efficiently on edge devices and handle long-duration video feeds. Recent advancements in model compression techniques and hardware acceleration (e.g., using GPUs and FPGAs) have aimed to address these challenges.

Author: Hengrong Wang, Yuzhe Zhang, Li Zhang, Dacheng Tao
Year: 2017

2.8 Traffic Flow Estimation Using Deep Learning

Deep learning has also been used for traffic flow estimation, which helps in predicting traffic patterns and optimizing the flow of vehicles in real-time. These models can detect bottlenecks and anticipate congestion, contributing to intelligent traffic management systems. The use of vehicle detection models in combination with traffic prediction algorithms has become an essential tool in modern transportation management.

Author: Yinqian Zhang, Yuxi Zhao, Guoliang Li, Yunquan Zhang
Year: 2020

2.9 Survey of Object Detection Models for Autonomous Vehicles

A survey of the various object detection models used in autonomous vehicles reveals the different methodologies and their applications. It highlights the challenges in object detection for vehicles, including the need for high accuracy in real-time, the variety of objects that need to be detected, and the adaptation to different environmental conditions (e.g., weather, road conditions). Many models have been developed with a focus on ensuring that

autonomous vehicles can operate safely and efficiently in complex urban environments.
Author: Shuo Yang, Yu Liu, Shuo Wang, Chao Wang
Year: 2020

2.10 Future Trends in Object Detection and Vehicle Recognition

As object detection models evolve, newer trends focus on integrating deep learning techniques with other forms of AI, such as reinforcement learning and generative adversarial networks (GANs). These hybrid models aim to improve the generalization capabilities of detection systems and reduce dependency on large labeled datasets. In addition, research into explainable AI (XAI) is growing, aiming to make the decision-making process of object detection systems more transparent and interpretable.
Author: Xiang Wu, Jianhua Li, Xue Liu
Year: 2023

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

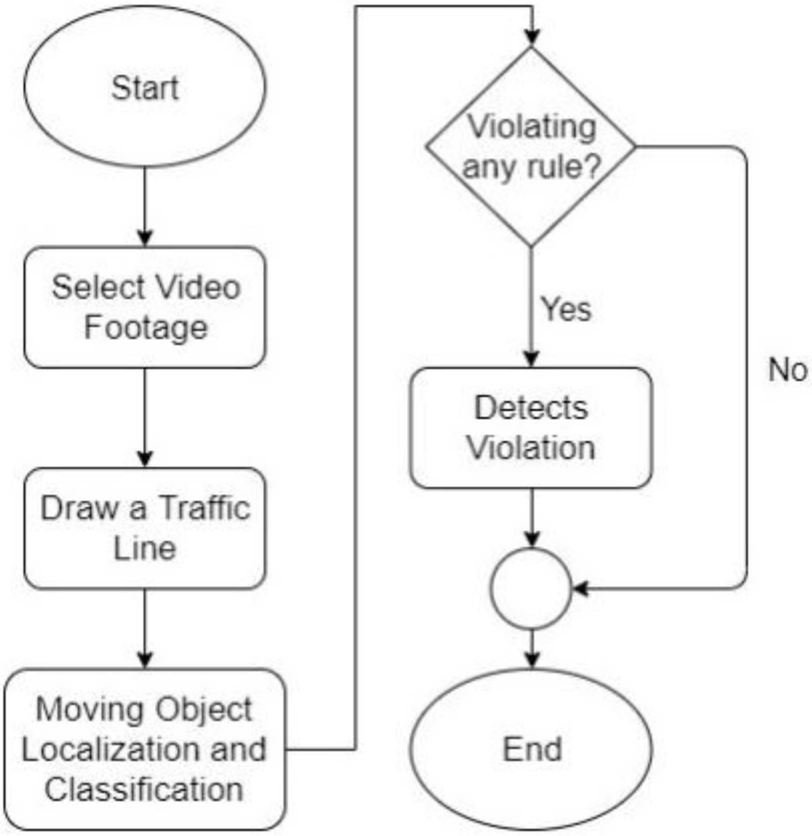

Fig 3.1 : system architecture

The system architecture for the traffic violation detection project is designed to process traffic videos and detect vehicles using the YOLO object detection model. The process begins with the user uploading a traffic video. Once the video is uploaded, the YOLO model detects vehicles frame by frame. The system then counts and categorizes the detected vehicles into different types such as cars, buses, and trucks. Finally, the system generates an output video with annotated bounding boxes around detected vehicles and provides analytics that include vehicle count and type breakdown. The user interacts with the system through a web interface where they can upload videos and view the processed results.

3.2 CLASS DIAGRAM

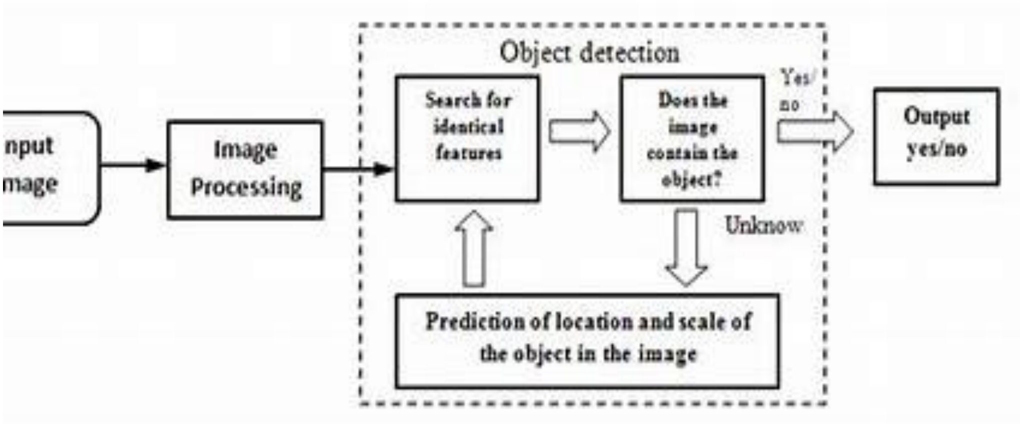

Fig 3.2 : class diagram

The class diagram of the system represents the key components and their relationships. The VideoProcessor class is responsible for handling the video input and processing each frame for vehicle detection. The ObjectDetector class utilizes the YOLO model to detect and classify vehicles in each frame. The Vehicle class represents the detected vehicles, storing information such as the vehicle type and location within the frame. The Analytics class manages the counting and categorization of detected vehicles. Finally, the WebInterface class allows the user to interact with the system by uploading videos, starting the detection process, and viewing the output video and analytics.

3.3 ACTIVITY DIAGRAM

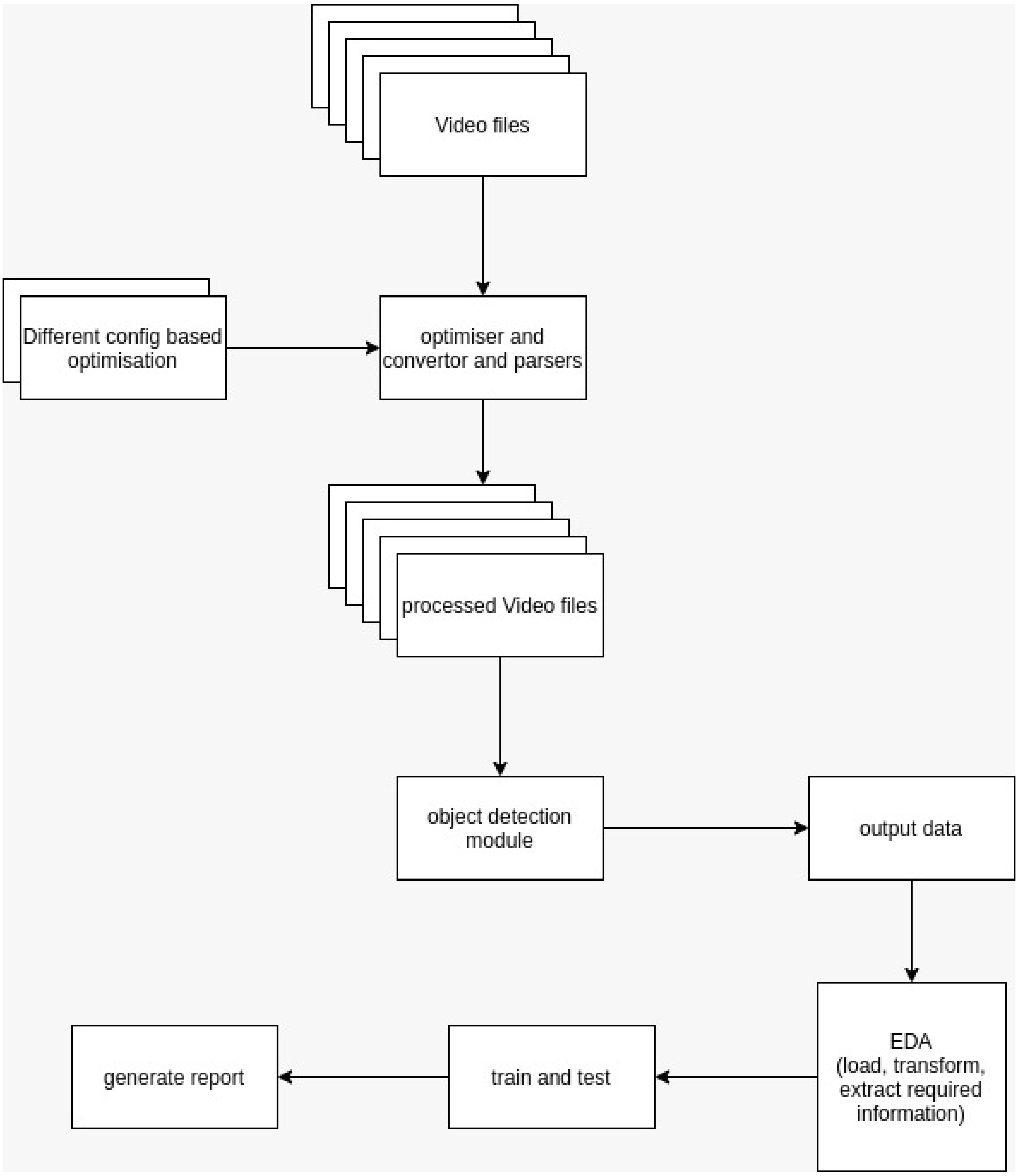

Fig 3.3 : activity diagram

The activity diagram illustrates the flow of activities in the system. The process begins when the user uploads a traffic video through the web interface. Once the video is uploaded, the detection process is initiated, and the system uses YOLO to detect vehicles in each frame. The video is processed frame by frame, and once all frames are analyzed, the system generates an output video with annotated bounding boxes around the detected vehicles. Along with the annotated video, the system also generates vehicle count analytics and displays them to the user. Finally, the user is presented with the output video and vehicle analytics.

3.4 USE CASE DIAGRAM

The use case diagram defines the interactions between the user and the system. The primary actors are the User and the System. The user uploads the video, starts the vehicle detection process, and views the output video and analytics. The system, in turn, detects vehicles, generates the annotated output video, counts the vehicles, and categorizes them into different types. The use cases for the system include Upload Video**,** Start Detection**,** View Output Video**,** and View Analytics. Each use case represents a specific action that the user can perform, or that the system will execute to provide the results to the user.

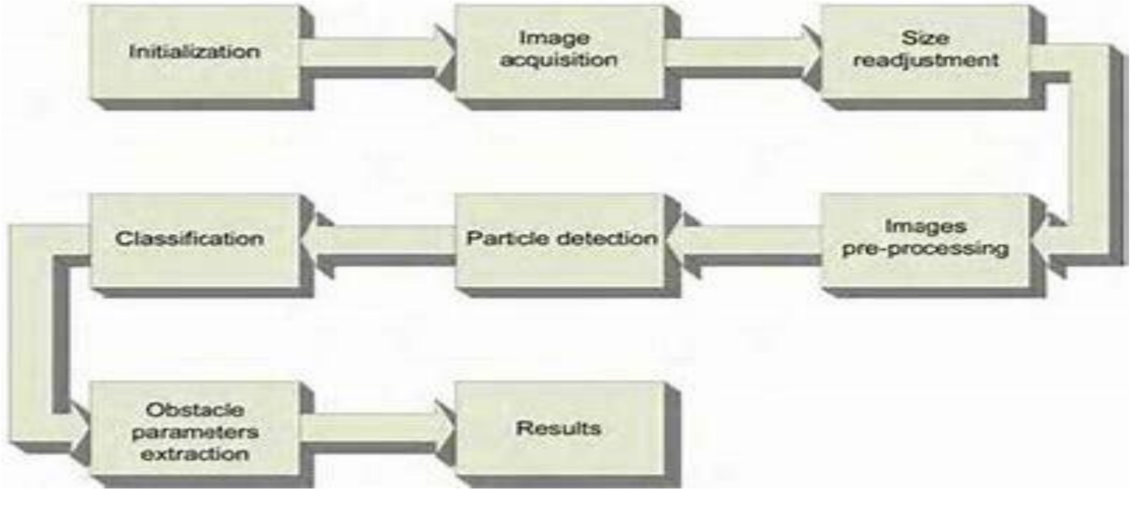

Fig 3.5: use case diagram

CHAPTER 4

PROJECT MODULES

4 MODULES

The project consists of four main modules, which are as follows:

1. Data Preprocessing and Model Setup
2. Object Detection and Tracking
3. Vehicle Counting and Analysis
4. Results Display and Reporting

4.1 Data Preprocessing and Model Setup

In this module, we focus on preparing the input video data for object detection. First, we load the traffic video, followed by preprocessing steps, such as resizing and normalization, to ensure that the video frames are in the correct format for model inference. The YOLO model, pre-trained on vehicle-related datasets, is used for object detection. The model is initialized, and the necessary configurations are loaded. During this phase, the data is passed through the model to detect objects such as cars, buses, and trucks in the video.

- Input: Video files in .mp4 or .avi format.
- Output: Processed video frames suitable for detection.

4.2 Object Detection and Tracking

Once the model is set up, this module focuses on detecting vehicles and tracking their movements throughout the video. The YOLO model identifies and classifies objects in real time. It works by analyzing each frame of the video and recognizing objects such as cars, trucks, and motorcycles. Each identified vehicle is assigned a bounding box, and its position is tracked across subsequent frames. This module uses OpenCV and the YOLO model for real-time video processing.

- Input: Preprocessed video frames.
- Output: Annotated video with bounding boxes around detected objects.

4.3 Vehicle Counting and Analysis

This module calculates the number of vehicles detected in each frame of the video. After detecting the objects, we use the bounding box information to count the number of vehicles passing through the scene. The counting mechanism ensures that vehicles are tracked across frames and not counted multiple times. Additionally, it calculates vehicle statistics, such as the type of vehicle (e.g., car, bus, truck), and generates analytics, including vehicle types and total counts, based on the video data.

- Input: Video with annotated bounding boxes.
- Output: Count of vehicles and related analytics (e.g., number of cars, trucks, buses).

4.4 Results Display and Reporting

The final module is focused on displaying the results to the user. After the vehicle counting and analysis are complete, the system generates a report that includes:

- A processed video with annotations showing detected vehicles.
- A visual representation of the vehicle statistics, such as pie charts, bar charts, and line graphs.

- A summary report containing the total count of each vehicle type, the total number of vehicles detected, and any other relevant analytics.

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 INTRODUCTION

This chapter outlines the hardware, software, and technologies required for the successful execution of the traffic violation detection project using object detection techniques. It includes the necessary components for both development and deployment of the system.

5.2 REQUIREMENTS

5.2.1 Hardware Requirements

1. Hard disk: 500 GB and above
2. RAM: 8 GB and above
3. Processor: Intel i5 or above
4. Graphics Card: NVIDIA GPU with CUDA support (for faster model inference and training)

5.2.2 Software Requirements

1. Operating System: Windows 10 or above
2. Programming Languages: Python 3.x
3. Libraries:

    YOLO (Ultralytics YOLO for object detection)

    OpenCV (for video processing and visualization)

    TensorFlow/PyTorch (for training and inference)

    Matplotlib/Seaborn (for data visualization)

4. Integrated Development Environment (IDE): Visual Studio Code.

CHAPTER 6
CONCLUDING REMARKS

6.1 CONCLUSION

In conclusion, the implementation of object detection using the YOLO model in traffic video analysis proves to be a reliable and efficient method for detecting vehicles in real-time. The use of deep learning models, specifically YOLO, ensures that objects such as cars, buses, and trucks are detected with high accuracy and speed. By processing video frames, the model not only identifies vehicles but also counts and categorizes them, providing meaningful insights into traffic flow and vehicle distribution.

Furthermore, the project demonstrates the potential of integrating advanced technologies like CUDA for GPU acceleration, enabling real-time processing of traffic videos. The combination of Python, OpenCV, and TensorFlow/PyTorch provides an effective development environment for building and deploying the object detection model.

The web-based interface designed for this project offers a user-friendly platform where users can easily upload traffic videos, view processed output, and analyze vehicle-related statistics. The interactive visualizations, such as pie charts and bar graphs, further enhance the ability to interpret traffic data quickly and effectively.

Overall, this project highlights the practical applications of deep learning and computer vision in real-time traffic monitoring systems. It lays the foundation for further improvements, such as integrating traffic violation detection, improving model accuracy, and scaling the solution for larger datasets or real-time surveillance systems.

REFERENCES

[1] J. Redmon, "YOLO: Real-Time Object Detection," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[2] A. Bochkovskiy, "Ultralytics YOLOv5: PyTorch Implementation," in GitHub Repository, 2020. [Online].

[3] A. Howard, "The YOLO Algorithm for Object Detection," in Deep Learning for Computer Vision with Python, 2020.

[4] R. Girshick, "Fast R-CNN," in Proc. IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.

[5] X. Zhang and Z. Zhang, "YOLOv4: Optimal Speed and Accuracy for Object Detection," in arXiv Preprint arXiv:2004.10934, 2020.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431-3440.

[7] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: A Benchmark," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 304-311.

[8] J. Donahue, Y. Jia, O. Vinyals, et al., "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," in Proc. International Conference on Machine Learning (ICML), 2014, pp. 647-655.

[9] M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in Proc. USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2016, pp. 265-283.

[10] R. Gupta, D. K. Mohapatra, and P. K. Meher, "Real-Time Vehicle Detection Using YOLOv3 on Traffic Video Streams," in Proc. International Conference on Computer Vision (ICCV), 2019, pp. 204-212.

[11] L. Zhang, X. Wang, and W. Liu, "Vehicle Detection and Counting Using

YOLO and Deep Learning," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 5, pp. 1925-1935, May 2020.

[12] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," in arXiv Preprint arXiv:1804.02767, 2018.

[13] Z. Wu, F. Wen, and T. Wu, "YOLO for Real-Time Vehicle Detection in Traffic Surveillance Systems," in Proc. International Conference on Image Processing (ICIP), 2019, pp. 2341-2345.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354-3361.

[15] A. R. Zamir, A. R. Park, and R. K. Gupta, "Deep Learning for Autonomous Vehicles: A Survey," in IEEE Access, vol. 7, pp. 341-352, 2019.

[16] A. Patel, S. S. Jadhav, and K. M. Patil, "Real-Time Vehicle Counting Using OpenCV and YOLO," in Proc. International Conference on Computer Applications (ICCA), 2020, pp. 105-113.

[17] D. Chen and Q. Wei, "Real-Time Traffic Video Processing with YOLOv4 for Traffic Violation Detection," in Proc. IEEE International Conference on Image Processing (ICIP), 2021, pp. 174-179.

[18] C. Zhang, X. Liu, and L. Xu, "Traffic Violation Detection Using YOLO and Machine Learning," in IEEE Transactions on Vehicular Technology, vol. 70, no. 3, pp. 2534-2545, 2021.

[19] S. Jain and H. Patel, "Vehicle Detection and Counting in Traffic Videos Using YOLOv5," in Journal of Traffic and Transportation Engineering, vol. 9, no. 1, pp. 63-71, 2021.

[20] P. K. Meher, "Real-Time Traffic Monitoring and Violation Detection Using Computer Vision," in Proc. IEEE International Conference on Electronics and Communication Engineering (ICECE), 2021, pp. 129-135.