Real-time Traffic Analysis and Parking Safety with YOLOv8

A PROJECT REPORT

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

BACHELOR OF TECHNOLOGY

By

IJAS AHAMMED [SCM20CS063]
MEENAKSHY R NAMBIAR [SCM20CS081]
P.K. PRANAV SATHISH [SCM20CS098]
TOM GEORGE KAPPIL [SCM20CS127]

SCMS SCHOOL OF ENGINEERING AND TECHNOLOGY
(Affiliated to APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY)
VIDYA NAGAR, PALISSERY, KARUKUTTY
ERNAKULAM - 683576

May 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCMS SCHOOL OF ENGINEERING AND TECHNOLOGY
Vidya Nagar, Karukutty, Ernakulam - 683582

BONAFIDE CERTIFICATE

This is to certify that the report entitled *'Real-time Traffic Analysis and Parking Safety with YOLOv8'*, submitted by Ijas Ahammed, Meenakshy R Nambiar, P.K. Pranav Sathish and Tom George Kappil, to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Guide	**Project Coordinator**	**Head Of Department**
Ms. Bini Omman	Dr. Arshey M	Dr. MANISH T I
Assistant Professor	Associate Professor	Professor
Department of CSE	Department of AI & DS	Department of CSE
SSET	SSET	SSET

DECLARATION

We undersigned hereby declare that the project report on *'Real-time Traffic Analysis and Parking Safety with YOLOv8'*, submitted for partial fulfilment of the requirements for the award of the degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under the supervision of Ms. Bini Omman. This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for the award of any degree, diploma, or similar title of any other University.

PLACE: Ernakulam
DATE: 12/04/24

SIGNATURE
Ijas Ahammed
Meenakshy R Nambiar
P.K. Pranav Sathish
Tom George Kappil

ACKNOWLEDGEMENT

We are greatly indebted to Dr. Anitha G Pillai, Principal, SCMS School of Engineering and Technology, Ernakulam and Dr. Manish T I, Head of Department, Department of Computer Science and Engineering, SCMS School of Engineering and Technology, who wholeheartedly granted us the permission to conduct this project. We would like to thank our guide Ms. Bini Omman and our coordinator Dr. Arshey M., for their valuable inputs and mentoring, without which this project would not have been completed successfully. We would like to express our sincere gratitude to all the teachers of the Department of Computer Science who gave us moral and technical support. We would like to thank the supporting staff in the computer lab whose dedicated work kept the lab working smoothly thus enabling us to have access to various resources which helped us understand more about the project topic. We would also like to extend our gratitude to our friends and family for providing us with the necessary resources and support.

ABSTRACT

This project introduces an all-encompassing solution for real-time traffic analysis and parking safety assessment, harnessing cutting-edge computer vision algorithms. Our system boasts a user-friendly interface that seamlessly integrates live highway surveillance camera feeds, providing a comprehensive overview of traffic conditions. By capitalizing on the YOLOv8 architecture, our system adeptly identifies, categorizes, tracks, quantifies, and gauges the velocity of vehicles within the monitored regions. Moreover, it constructs heatmaps, depicting traffic density across lanes over varying time intervals, thus aiding in the analysis of traffic trends and facilitating predictive studies.

Additionally, a dedicated module addresses parking safety concerns, utilizing a specialized model to determine if vehicles are parked correctly in on-road parking slots. This functionality empowers traffic authorities to effectively monitor parked vehicles and implementing necessary safety measures to follow the rules and mitigating potential risks.

CONTENTS

List of Figures

List of Tables

ABBREVIATIONS

MOT Multiple Object Tracking

YOLO You Only Look Once

SORT Simple Online and Realtime Tracking

ROI Region OF Interest

TIA Traffic Impact Analysis

TP True Positive

FP False Positive

TN True Negative

FN False Negative

CAB Context Attention Block

COCO Common Objects in Context

SA Spatial Attention

C2f Coarse-to-Fine

Chapter 1

INTRODUCTION

1.1 Overview

In big cities, keeping traffic moving smoothly and making sure parking is safe are really important for everyone's safety and convenience. But with so many cars on the roads and parking spaces filling up fast, it's not easy to manage everything. That's why we're using the latest technology to tackle these challenges.

Our project is all about creating a system that can watch traffic and parking spots in real-time using cameras on highways and streets. We've made it easy to use, so anyone can understand what's going on. We're using YOLOv8 (You Only Look Once - version 8) to help us in detecting, classifying, tracking, and estimating the speed of vehicles within the monitored areas. We even make heat maps to show which lanes have the most traffic at different times.

Additionally, a dedicated module addresses parking safety concerns, utilizing a specialized model to ascertain the appropriateness of vehicle placement in on-road parking slots. This functionality empowers traffic authorities to effectively monitor parked vehicles and implement requisite safety measures, ensuring adherence to regulations and mitigating potential risks.

Through this project, we aim to contribute to the advancement of intelligent transportation systems, providing actionable insights for enhancing traffic management and ensuring safer road environments.

1.2 Objectives

- Develop a user-friendly system for real-time traffic analysis and parking safety assessment: The project aims to create an accessible system that can analyze traffic data and assess parking safety in real-time.
- Integrate live highway surveillance camera feeds seamlessly into the system: Incorporating live camera feeds from highways into our system for continuous monitoring.
- Utilize YOLOv8 architecture for accurate vehicle detection, classification, tracking, and speed estimation: Leveraging YOLOv8 architecture, we aim to accurately detect, classify, track, and estimate the speed of vehicles within monitored areas.
- Generate heatmaps to visualize traffic distribution over time: The project will involve creating heatmaps to visually represent traffic distribution across lanes over different time periods, aiding in comprehensive traffic analysis.
- Implement a dedicated module for parking safety assessment: We will develop a specific module to assess whether vehicles are correctly parked in on-road parking spots, enhancing safety measures.
- Conduct comparative analysis with SORT, DEEPSORT, and ByteTrack algorithms: We will compare our system's performance with leading algorithms to evaluate its effectiveness and identify areas for improvement.
- Contribute to advancing intelligent transportation systems: Through this project, we aim to contribute to the development of intelligent transportation systems by providing valuable insights for traffic management and ensuring safer road environments.

1.3 Problem Statement

The challenge is the inefficiency of existing traffic management systems, lacking real-time capabilities and accurate vehicle detection. To address this, there's a pressing need for an integrated system that can provide real-time traffic analysis and parking safety assessment, ensuring efficient traffic management and safer roads.

1.4 Organization of Report

The report is divided into five chapters. The overview, objectives and problem statement are covered in the introductory part of the report. A variety of reviews of relevant literature are included in the second chapter. Chapter Three goes on the specific methodology of the functioning of the suggested system. The system architecture and use case diagrams are discussed in the fourth chapter.The experimental results and discussions are presented in the fifth chapter. In the last chapter, the project's conclusion and future scope are also mentioned. The references are provided at the end of the pages.

Chapter 2

LITERATURE REVIEW

"Ng, J.J., Goh, K.O.M. and Tee, C., 2023. Traffic Impact Assessment System using Yolov5 and ByteTrack. Journal of Informatics and Web Engineering, 2(2), pp.168-188." [1] In this, the Traffic Impact Assessment (TIA) system integrates YOLOv5 and ByteTrack for vehicle detection, classification, and real-time object tracking, addressing safety and congestion concerns through effective traffic management strategies. Expounding on related research encompassing methods like SORT, DeepSORT, StrongSORT, and ByteTrack, the document elucidates their performance and accuracy. It outlines the system's implementation steps, including camera connection, region definition for vehicle counting, and detection parameter configuration, alongside its capacity to exhibit object information, tracking results, and experimental outcomes such as accuracy tests. Concluding with future prospects like vehicle speed calculations and traffic violation detection, the system's robustness and potential for advancements are underscored. Experimental assessments involving two-minute videos, Video 1 and Video 2, at 24 FPS and 1280×720 resolution, demonstrate the system's performance using GPU-accelerated YOLOv5 detection on an NVIDIA RTX 2060 Super model. Comparisons with ByteTrack, OC-SORT, and StrongSORT in tracking accuracy and time highlight its effectiveness, augmented by a model warm-up process optimizing response times.

"Bathija, A. and Sharma, G., 2019. Visual object detection and tracking using yolo and sort. International Journal of Engineering Research Technology, 8(11), pp.345-355." [2] The paper underscores the importance of visual object detection and tracking using YOLO and SORT algorithms, emphasizing the challenges associated with tracking moving objects across consecutive video frames. It presents an analysis of the tracking-by-detection approach, employing YOLO for detection and SORT for tracking. The focus lies on training a custom image dataset for six specific classes using YOLO and employing this model for object tracking with the SORT algorithm. Addressing the complexity of visual tracking in computer vision, factors such as target deformations, illumination variations, and rapid movement are discussed, highlighting the necessity for accurate object detection systems in applications like autonomous vehicles and smart video surveillance. The process of visual object tracking is elucidated, detailing the utilization of YOLO

for detection and SORT for tracking in Python. Technical aspects of YOLO, including its speed and accuracy in object detection, and the procedure for training a custom dataset are explored. The implementation of SORT, an online tracker utilizing the Hungarian algorithm and Kalman filter, is discussed. A quantitative analysis of the proposed system is provided, encompassing parameters such as true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). Experimental results outline the testing of the system on various videos, offering insights into the achieved accuracy, precision, and recall. Additionally, the potential for future work is discussed, including training the system for additional classes, implementing different object detectors and trackers, and expanding the scope of object detection and tracking across various domains and objects.

"Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W. and Wang, X., 2022, October. Bytetrack: Multi-object tracking by associating every detection box. In European conference on computer vision (pp. 1-21). Cham: Springer Nature Switzerland." [3] The paper introduces a novel Multiple Object Tracking (MOT) method called BYTE, aimed at improving tracking accuracy by associating almost every detection box. It proposes a simple yet effective data association method, BYTE, to recover objects from low-score detection boxes and filter out background detections, surpassing other state-of-the-art tracking methods. The introduction of ByteTrack, a robust tracker combining the high-performance detector YOLOX with the BYTE method, achieves state-of-the-art performance on MOT17, MOT20, HiEve, and BDD100K tracking benchmarks, exhibiting high MOTA, IDF1, and HOTA scores, and running at 30 FPS on a single V100 GPU. The rationale behind BYTE emphasizes its ability to enhance tracking accuracy by recovering objects from low-score detection boxes and utilizing every detection box in the matching process, showcasing its robustness and generalization across different trackers. Furthermore, an analysis of BYTE's performance using lightweight detection models illustrates its stability and effectiveness, even with lightweight detectors. Experiments on tracklet interpolation demonstrate its ability to improve MOTA and IDF1 scores, enhancing tracking accuracy. Moreover, insights into the impact of different training data on ByteTrack's performance highlight its capability to achieve high accuracy with minimal training data. Overall, the study offers a comprehensive evaluation of the proposed methods, showcasing their effectiveness and robustness across various datasets and scenarios.

"Talib, M., Al-Noori, A.H. and Suad, J., 2024. YOLOv8-CAB: Improved YOLOv8 for Real-time object detection. Karbala International Journal of Modern Science, 10(1), p.5" [4] The study intro-

duces the YOLOv8-CAB model, enhancing object detection accuracy, especially for small objects, by integrating the Context Attention Block (CAB), modifying the Coarse-to-Fine (C2F) block, and Spatial Attention (SA). With a mean average precision of 97% on small objects, a 1% increase over conventional models, the YOLOv8-CAB demonstrates innovation and potential for real-time object detection advancements. Evaluation on the COCO dataset highlights superior feature extraction and contextual information preservation, outperforming standard YOLO models and state-of-the-art techniques. The manuscript addresses challenges in object detection, emphasizing the model's significance in efficient detection, particularly for small and geometric objects. Detailed experimental analysis and results underscore the model's capability across various object categories, positioning YOLOv8-CAB as a promising advancement in real-time object detection techniques.

Chapter 3

METHODOLOGY

3.1 Proposed System

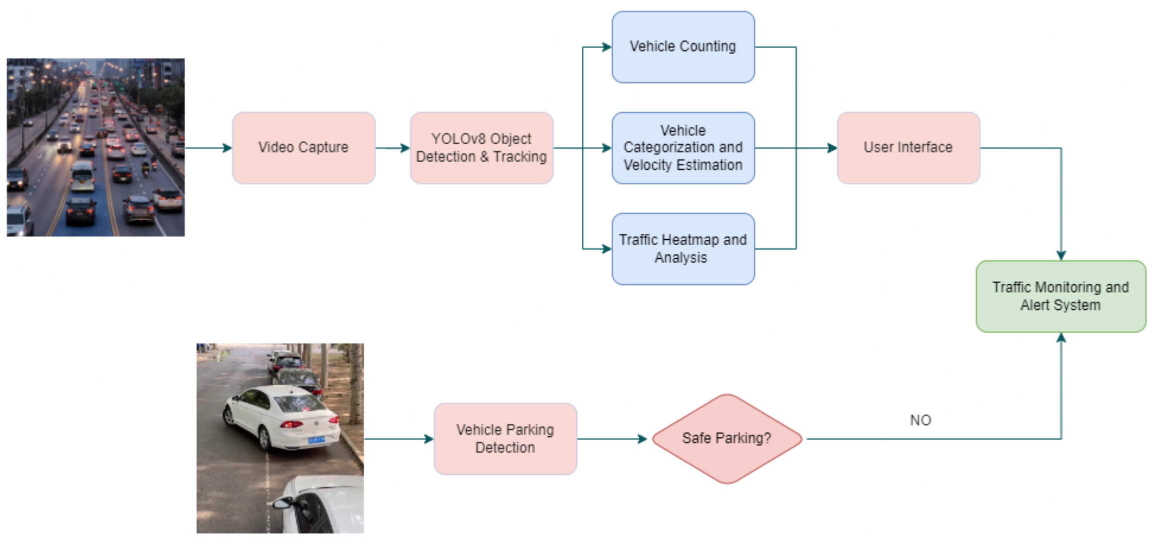

Figure 3.1: Proposed System

- **Video Capture:** This module captures and preprocesses the incoming video feed, making it ready for further analysis.

- **YOLOv8 Object Detection and Tracking:** This is the core component of the system, which utilizes the YOLOv8 architecture to detect, classify, and track vehicles in the video feed.

- **Vehicle Counting:** This module keeps track of the number of vehicles detected in the video feed.

- **Vehicle Categorization and Velocity Estimation:** This component categorizes the detected vehicles based on their types (e.g., cars, trucks, motorcycles) and estimates their velocities.

- **Traffic Heatmap and Analysis:** This module creates heatmaps that visualize the traffic density across different lanes and time intervals, enabling traffic trend analysis and predictive studies.
- **Parking Safety Assessment (Vehicle Parking Detection):** This dedicated module utilizes a specialized model to assess if vehicles are parked correctly in designated on-road parking slots, aiding in parking safety monitoring.
- **User Interface:** This component provides a user-friendly interface that displays the processed video feed, heatmaps, vehicle counts, and other relevant information for traffic monitoring and analysis.
- **Traffic Monitoring and Alert:** This component observes traffic conditions by analyzing processed data and produces visual representations. Additionally, it includes an alert or notification system that activates when the vehicle parking detection module identifies improperly parked vehicles.

3.2 Dataset

We utilized the UA-DETRAC dataset for our project. The UA-DETRAC dataset comprises 100 videos, meticulously selected from over 10 hours of image sequences captured by a Canon EOS 550D camera at 24 distinct locations. To ensure variability, data collection occurred at different sites with varying illumination conditions and shooting angles. The videos were recorded at a frame rate of 25 frames per second (fps), with JPEG image resolution set at 960×540 pixels.

Due to its extensive size, we worked with approximately 1000 images from this dataset. These images were manually annotated using the Roboflow tool, and the classes we focused on were car, truck, bus, and ambulance. We divided the dataset into training and testing sets using an 80%/20% split ratio, respectively. The training set facilitated the training of the YOLOv8 model to recognize vehicles accurately. Subsequently, the testing set enabled us to evaluate the final performance of the trained model on unseen data, providing an unbiased assessment of its effectiveness in real-world scenarios.

Furthermore, for the assessment of parking safety, we developed a custom dataset. We extracted frames from a publicly available video and utilized them to train another YOLOv8 model in a similar fashion.

3.3 Vehicle Tracking using SORT

In this project, we employed the SORT (Simple Online and Realtime Tracking) algorithm for vehicle tracking. SORT is a widely used real-time object tracking algorithm that utilizes a combination of bounding box detection and a Kalman filter for tracking objects across frames.

3.3.1 SORT Methodology

Object Detection: The YOLO model is used to detect objects (vehicles) in each frame and generate bounding boxes with associated detection scores.

Bounding Box Association: SORT associates the detected bounding boxes in the current frame with existing tracks from the previous frame, based on the Intersection over Union (IoU) metric and a predefined threshold.

Track Initialization and Termination: New tracks are initialized for unassociated bounding boxes, while existing tracks are terminated if they have not been associated with any detections for a certain number of frames.

Kalman Filter: A Kalman filter is employed to predict the next state of each track, considering the object's motion and accounting for potential occlusions or missed detections.

3.3.2 SORT Advantages and Limitations

Advantages:

- Simple and computationally efficient
- Real-time performance
- Handles occlusions and missed detections to some extent

Limitations:

- Unique IDs assigned to objects may change across frames
- Tracking performance can degrade in complex scenarios with frequent occlusions or crowded environments

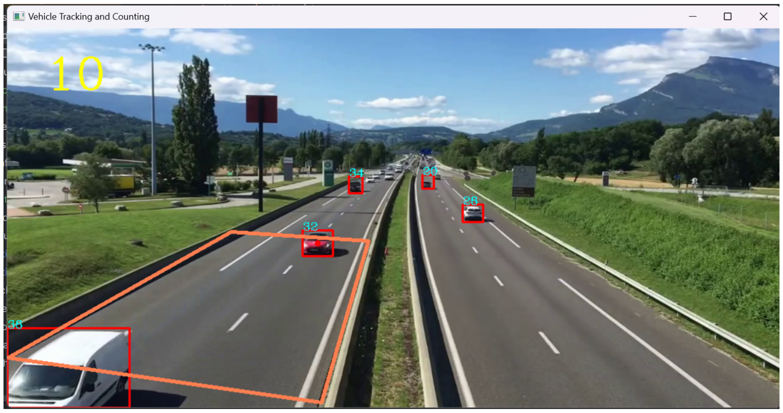

Figure 3.2: (a) Vehicle tracking using sort

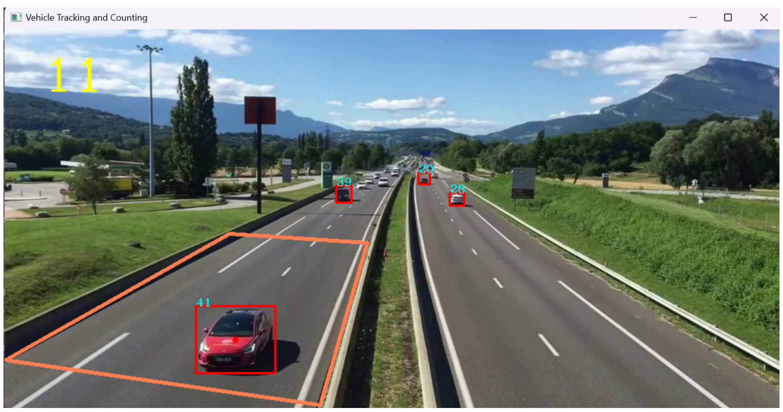

Figure 3.3: (b) Vehicle tracking using sort

3.4 Vehicle Tracking using ByteTrack

ByteTrack is a more recent and advanced Multiple Object Tracking (MOT) algorithm that leverages byte feature embeddings for robust and consistent object tracking.

3.4.1 ByteTrack Methodology

Object Detection: Similar to SORT, the YOLO model is used to detect objects (vehicles) in each frame and generate bounding boxes with associated detection scores.

Feature Extraction: ByteTrack extracts byte feature embeddings for each detected object, representing its visual appearance and characteristics.

Bounding Box Association: The algorithm associates the detected bounding boxes in the current frame with existing tracks from the previous frame, using a combination of the Intersection over Union (IoU) metric and the similarity between the byte feature embeddings.

Track Initialization and Termination: New tracks are initialized for unassociated bounding boxes, and existing tracks are terminated if they have not been associated with any detections for a certain number of frames.

Track Management: ByteTrack employs various strategies to handle occlusions, track re-identification, and track termination, ensuring robust and consistent object tracking.

3.4.2 ByteTrack Advantages and Limitations

Advantages:

- Assigns unique and consistent IDs to objects throughout the entire frame sequence
- Improved tracking accuracy and robustness, especially in complex scenarios with occlusions and crowded environments
- Utilizes byte feature embeddings for more reliable object re-identification

Limitations:

- Computationally more expensive compared to SORT
- May require additional computational resources for real-time performance

Chapter 4

SYSTEM DESIGN

4.1 System Architecture

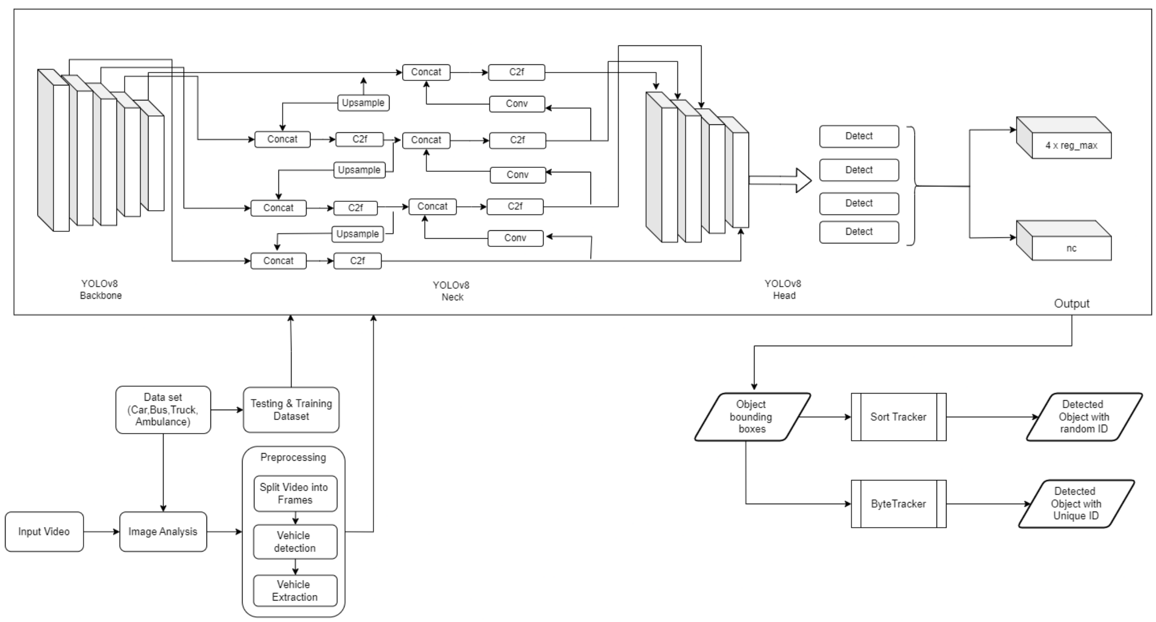

Figure 4.1: System Architecture

The design of our traffic monitoring system revolves around the integration of several key components, including the YOLOv8 architecture for object detection, data preprocessing for efficient handling of input video streams, conversion of input videos into frames for analysis, and the generation of bounding boxes as output. Additionally, vehicle tracking is implemented using two distinct Multiple Object Tracking (MOT) algorithms: SORT (Simple Online and Realtime Tracking) and ByteTrack.

4.1.1 YOLOv8 Architecture

The core component of our system is built upon the YOLOv8 architecture. YOLOv8, short for You Only Look Once version 8, is a state-of-the-art real-time object detection system. It employs a single convolutional neural network (CNN) to simultaneously predict bounding boxes and classify objects within those boxes. By leveraging YOLOv8, our system can accurately detect and classify vehicles in real-time video streams, providing a foundational framework for subsequent analysis.

4.1.2 Data Preprocessing

Prior to input into the YOLOv8 architecture, incoming video streams undergo preprocessing. This step involves tasks such as noise reduction, frame normalization, and resolution adjustments to optimize the data for subsequent analysis. Data preprocessing ensures that the input to the object detection model is of high quality and consistency, enhancing the accuracy and efficiency of the detection process.

4.1.3 Video to Frames Conversion

To facilitate real-time analysis, input video streams are converted into individual frames. This process involves extracting each frame from the video stream and passing it through the object detection pipeline independently. By converting videos into frames, our system can analyze each frame in isolation, enabling efficient parallel processing and real-time detection of vehicles.

4.1.4 Output Bounding Boxes

Upon detection of vehicles within each frame, our system generates bounding boxes to delineate the spatial extent of each detected object. These bounding boxes provide valuable information about the location and size of vehicles within the video stream, enabling downstream analysis such as vehicle counting, categorization, and tracking.

4.1.5 Vehicle Tracking with MOT Algorithms

SORT is a widely-used online tracking algorithm known for its simplicity and efficiency. It operates by assigning unique IDs to detected objects in the initial frame and then maintains object identities across subsequent frames using a combination of motion prediction and data association techniques.

While SORT does not provide unique IDs throughout the entire frame sequence, it offers reliable object tracking in real-time scenarios.

On the other hand, ByteTrack is a more recent MOT algorithm that utilizes byte feature embeddings for tracking. Unlike SORT, ByteTrack assigns unique IDs to objects throughout the entire frame sequence, ensuring consistent and accurate object tracking. This approach enhances tracking accuracy and robustness, particularly in complex scenarios with occlusions and crowded environments.

Chapter 5

RESULTS

5.1 Evaluation Metrics

In the field of object detection, various metrics are used to evaluate the performance of models. The following metrics were obtained for the YOLOv8 model in this project:

5.1.1 Precision (B)

Precision measures the fraction of true positive detections out of the total positive detections made by the model. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Where:

- TP (True Positives): The number of correctly detected objects.
- FP (False Positives): The number of incorrect detections or false alarms.

A high precision value indicates a low rate of false positives, meaning the model is accurate in identifying relevant objects. The obtained precision score of 0.92925 (or 92.9%) is excellent, suggesting that out of all the objects detected as positive by the model, approximately 92.9% were correctly identified.

5.1.2 Recall (B)

Recall measures the fraction of true positive detections out of the total actual positive instances in the data. It is calculated as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Where:

- TP (True Positives): The number of correctly detected objects.

- FN (False Negatives): The number of actual objects that were missed or not detected by the model.

A high recall value indicates a low rate of false negatives, meaning the model is able to detect most of the relevant objects. The obtained recall score of 0.93228 (or 93.2%) is outstanding, indicating that the model correctly detected approximately 93.2% of the actual positive instances (e.g., vehicles) present in the data.

5.1.3 mAP50 (B)

mAP50 (mean Average Precision at IoU=0.5) is a widely used metric for object detection tasks. It measures the average precision of the model at an Intersection over Union (IoU) threshold of 0.5, which means that a detection is considered a true positive if its IoU with the ground truth bounding box is greater than or equal to 0.5. A higher mAP50 value indicates better object detection and localization performance. The obtained mAP50 score of 0.95709 (or 95.7%) is exceptional, demonstrating the model's high accuracy in detecting and localizing objects at a reasonable IoU threshold of 0.5.

5.1.4 mAP50-95 (B)

mAP50-95 is another variation of the mAP metric, which calculates the average precision over multiple IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05. This metric provides a more comprehensive evaluation of the model's performance across different levels of localization accuracy. A higher mAP50-95 value indicates better object detection and localization performance across a range of IoU thresholds. The obtained mAP50-95 score of 0.76653 (or 76.6%) is very good, suggesting that the model maintains reasonably high precision even at higher IoU thresholds, which require more precise localization of the objects.

Table 5.1: Metrics

Metric	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)
Value	0.92925	0.93228	0.95709	0.76653

Table 5.2: Training Loss

Loss Component	train/box_loss	train/cls_loss	train/dfl_loss
Value	0.93933	0.491	1.0175

Table 5.3: Validation Loss

Loss Component	val/box_loss	val/cls_loss	val/dfl_loss
Value	0.82451	0.47661	1.0112

5.2 Evaluation Curves

5.2.1 F1 Curve

The F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics. The F1 curve plots the F1 score against different threshold values used for classification. It helps to visualize the trade-off between precision and recall as the classification threshold varies.

5.2.2 Precision Curve

The precision curve plots precision values against different threshold values used for classification. It illustrates how the precision of the model changes as the classification threshold varies.

5.2.3 Recall Curve

The recall curve plots recall values against different threshold values used for classification. It shows how the recall of the model changes as the classification threshold varies.

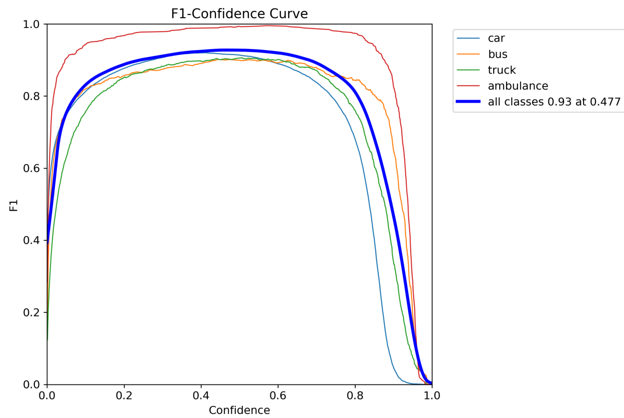

Figure 5.1: F1 Curve

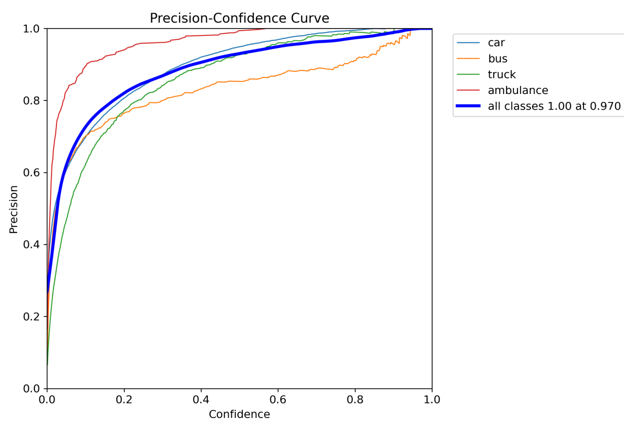

Figure 5.2: Precision Curve

5.2.4 Precision-Recall (PR) Curve

The precision-recall (PR) curve plots precision values against recall values for different threshold values used for classification. It provides insights into the trade-off between precision and recall at various classification thresholds.

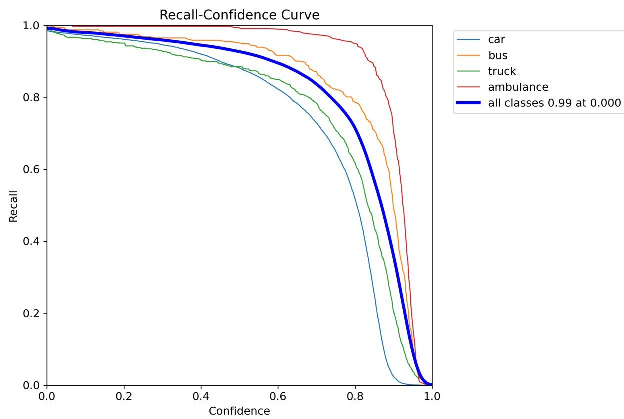

Figure 5.3: Recall Curve

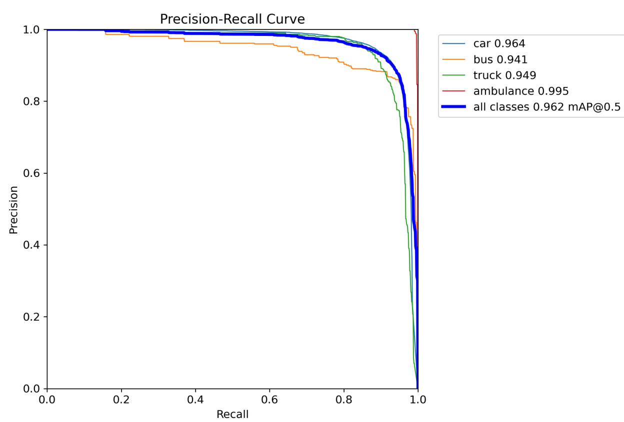

Figure 5.4: Precision-Recall Curve

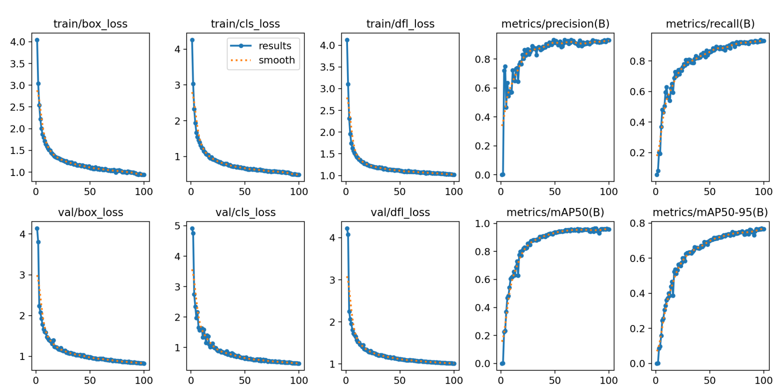

Figure 5.5: Results

5.3 Experimental Results

A user-friendly interface was developed using React.js and Flask API to facilitate video input for the vehicle tracking and counting model. This interface allows users to seamlessly upload videos to the system for analysis, providing a convenient way to interact with the application.

Implementation: The user interface was designed with simplicity and ease of use in mind. Users can upload videos directly from their devices through the browser interface. The uploaded videos are then processed by the backend Flask API, which handles the vehicle tracking and counting tasks using the YOLOv8 model.

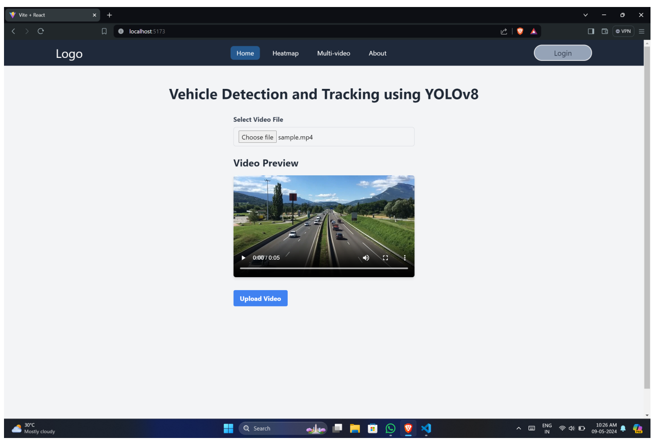

Figure 5.6: React app

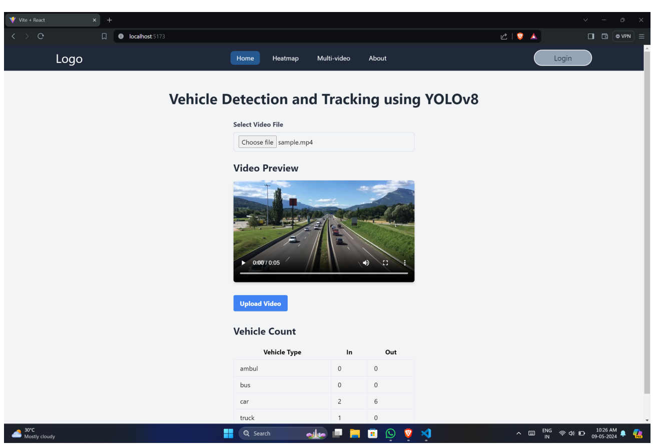

Figure 5.7: App Displaying results

5.3.1 Line Counter Method

The line counter method involves drawing a line on the frames, and when a detected vehicle crosses this line, counts are incremented for the respective vehicle type (car, truck, bus, ambulance). This method allows us to track the flow of vehicles and monitor traffic movement.

Implementation: For each frame, the YOLOv8 model detects vehicles and their bounding boxes. A line is drawn on the frame, and as vehicles cross this line, their counts are updated accordingly.

Advantages:

- Provides real-time tracking of vehicles crossing a specific point
- Differentiates between vehicle types for more detailed analysis

Limitations:

- Requires accurate placement of the line for precise counting
- May not capture vehicles that change direction before crossing the line

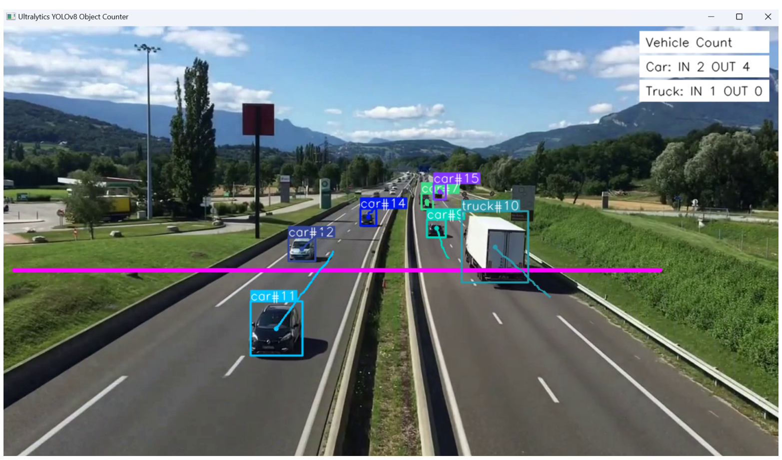

Figure 5.8: Line Counter

5.3.2 Polygon Counter Method

In the polygon counter method, a region of interest (ROI) is defined on the frames, and when a detected vehicle enters this region, counts are incremented. This method allows for more flexible counting in areas with irregular traffic patterns.

Implementation: Similar to the line counter method, the YOLOv8 model detects vehicles and their bounding boxes. A polygonal region is defined as the ROI, and as vehicles enter this region, their counts are updated.

Advantages:

- Offers flexibility in defining counting areas
- Suitable for monitoring traffic in complex environments

Limitations:

- Requires careful selection of the ROI to avoid counting unwanted objects
- May lead to overlapping counts if vehicles traverse multiple ROIs

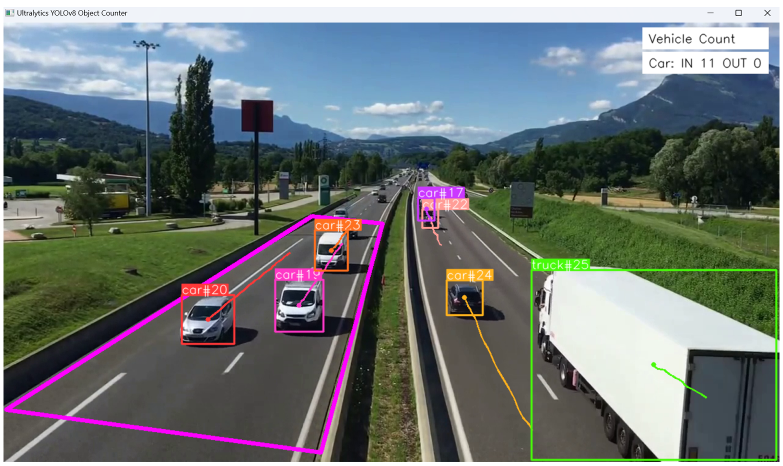

Figure 5.9: Polygon Counter

5.3.3 Heatmap Generation

Heatmap generation using YOLOv8 involves analyzing the distribution of detected vehicles across different regions of the frame. By visualizing the density of vehicle detections, heatmaps provide insights into traffic patterns and congestion areas.

Implementation: After detecting vehicles using YOLOv8, the distribution of vehicle detections is analyzed. Heatmaps are generated to highlight areas with high vehicle density.

Advantages:

- Offers a visual representation of traffic flow and congestion
- Helps identify areas with high traffic volume for further analysis

Limitations:

- Requires accurate vehicle detection for reliable heatmap generation
- Interpretation may be subjective and influenced by parameters such as color mapping

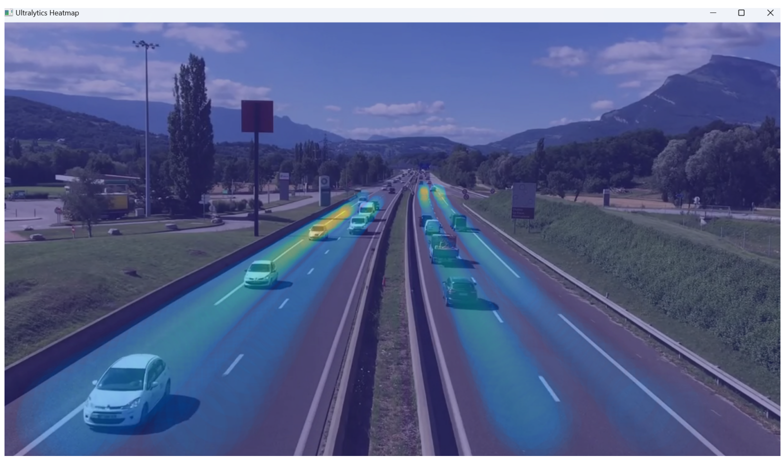

Figure 5.10: Heatmap

5.3.4 Parking Safety Assessment

In addition to vehicle tracking and counting, the system also incorporates a parking safety assessment module to address concerns related to on-road street parking. Leveraging the YOLOv8 segmentation model for car detection, combined with OpenCV for line detection, the system determines whether vehicles are parked correctly within designated parking spaces.

Implementation: The YOLOv8 segmentation model is utilized to detect cars in the video frames. A line is drawn using OpenCV to demarcate the boundary of the parking space. If a detected car is completely inside the designated area or has passed the line, it is classified as ”parked” and no action is taken. However, if any part of the car extends beyond the line, indicating incorrect parking, it is labeled as ”Not Parked”, and an alert is generated to notify authorities or relevant stakeholders.

Advantages:

- Enhances safety by ensuring vehicles are parked correctly within designated spaces
- Provides real-time monitoring and alerts for improper parking, facilitating timely intervention

Limitations:

- Accuracy may be affected by variations in vehicle sizes and orientations
- Limited to detecting parking violations based on line boundaries, may not capture all instances of improper parking

Figure 5.11: Parking Safety (a) Not Parked

By integrating parking safety assessment into the vehicle tracking and counting system, the overall effectiveness of traffic management and safety enforcement efforts is enhanced. Future enhancements could include the implementation of more advanced algorithms for precise parking detection and automated enforcement mechanisms to address parking violations effectively.

Figure 5.12: Parking Safety (b) Parked

Chapter 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

In conclusion, this project presents a comprehensive solution for real-time traffic analysis and parking safety assessment through the integration of cutting-edge computer vision algorithms. Our system, powered by the YOLOv8 architecture, efficiently identifies, categorizes, tracks, and quantifies vehicles in monitored regions. The user-friendly interface seamlessly integrates live highway surveillance camera feeds, providing a holistic view of traffic conditions.

Furthermore, the generation of heatmaps enables the visualization of traffic density across lanes over different time intervals, facilitating in-depth analysis of traffic patterns and predictive studies. Additionally, the dedicated parking safety module enhances monitoring capabilities by assessing vehicle parking compliance, empowering traffic authorities to enforce safety regulations effectively.

Overall, our system offers a valuable tool for traffic management authorities to monitor and analyze traffic dynamics and ensure parking safety, contributing to enhanced road safety and efficiency.

6.2 Future Scope

While the current system provides robust functionality for real-time traffic analysis and parking safety assessment, several avenues for future enhancement and expansion exist:

- **Integration of Advanced Algorithms:** Explore the integration of more advanced object detection and tracking algorithms to improve accuracy and robustness, potentially incorporating machine learning techniques for anomaly detection.

- **Enhanced User Interface:** Continuously refine and enhance the user interface to improve user experience and accessibility, incorporating features such as customizable dashboards and real-time alerts.

- **Integration with Smart City Initiatives:** Explore integration with smart city initiatives to leverage data from various sources, including IoT sensors and traffic management systems, for comprehensive traffic management and urban planning.
- **Predictive Analytics:** Develop predictive analytics capabilities to forecast traffic patterns, congestion hotspots, and parking demand, enabling proactive decision-making and resource allocation.
- **Expand Parking Safety Features:** Enhance the parking safety module to include features such as automated violation detection and enforcement, leveraging machine learning algorithms for improved accuracy.
- **Deployment in Diverse Environments:** Test and deploy the system in diverse urban and rural environments to assess its adaptability and scalability across different traffic scenarios and geographical regions.

Bibliography

[1] Ng, J.J., Goh, K.O.M., & Tee, C. (2023). Traffic Impact Assessment System using Yolov5 and ByteTrack. *Journal of Informatics and Web Engineering*, 2(2), 168-188. doi:10.33093/jiwe.2023.2.2.13

[2] Bathija, A., & Sharma, G. (2019). Visual object detection and tracking using YOLO and SORT. *IT Professional*, 8(11), 345-355. doi:10.17577/IJERTV8IS110343

[3] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022, October). Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision* (pp. 1-21). Cham: Springer Nature Switzerland. doi:10.1007/978-3-031-20047-2_1

[4] Talib, M., Al-Noori, A.H., & Suad, J. (2024). YOLOv8-CAB: Improved YOLOv8 for Real-time object detection. *Karbala International Journal of Modern Science*, 10(1), 5. doi:10.33640/2405-609X.3339

[5] Sriharsha Vikruthi, Dr. Maruthavanan Archana, Dr. Rama Chaithanya Tanguturi (2023). A Novel Framework for Vehicle Detection and Classification Using Enhanced YOLO-v7 and GBM to Prioritize Emergency Vehicle. *IJISAE*, 12(1s), 302–312.

[6] Lin, Cheng-Jian, & Jyun-Yu Jhang. (2022). Intelligent traffic-monitoring system based on YOLO and convolutional fuzzy neural networks. *IEEE Access*, 10, 14120-14133. doi:10.1109/ACCESS.2022.3147866

[7] J. Zhao et al. (2022). Improved vision-based vehicle detection and classification by optimized YOLOv4. *IEEE Access*, 10, 8590-8603. doi:10.1109/ACCESS.2022.3143365

[8] https://github.com/ultralytics/ultralytics