Real-Time Object Detection Using YOLOv8 for Transport System in Case of Tulu Dimtu to Adama Toll Roads

Daniel Asfaw

A Thesis Submitted to the Department of Computer Science and

Engineering School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

September, 2024

Adama, Ethiopia

Real-Time Object Detection Using YOLOv8 for Transport System in Case of Tulu Dimtu to Adama Toll Roads

Daniel Asfaw Mulugeta

Advisor: Mesfin Abebe (Ph.D.)

A Thesis Submitted to the Department of Computer Science and

Engineering School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University (ASTU)

September, 2024

Adama, Ethiopia

Declaration

I declare that this thesis/dissertation proposal entitled "Real-Time Object Detection Using YOLOv8 for Transport System in Case of Tulu Dimtu to Adama Toll Roads" is my own work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Name: Daniel Asfaw Mulugeta

Signature: ____

RECOMMENDATION

I the advisor(s) of this thesis, hereby certify that I have read the revised version of the thesis entitled "Real-Time Object Detection Using YOLOv8 for Transport System in Case of Tulu Dimtu to Adama Toll Roads" prepared under my/our guidance by Daniel Asfaw Mulugeta submitted in partial fulfillment of the requirements for the degree of Master's of Science in Computer Science and Engineering. Therefore, I recommend the submission of the revised version of the thesis to the department following the applicable procedures.

Dr. Mesfin Abebe	____________________	________________
Major Advisor/Supervisor	Signature	Date

APPROVAL PAGE OF M.SC. THESIS

I the advisors of the thesis entitled "Real-Time Object Detection Using YOLOv8 for Transport System in Case of Tulu Dimtu to Adama Toll Roads" and developed by Daniel Asfaw Mulugeta, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Dr. Mesfin Abebe	____________________	_______________
Major Advisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the thesis by Daniel Asfaw Mulugeta have read and evaluated the thesis entitled "Real-Time Object Detection Using YOLOv8 for Transport System in Case of Tulu Dimtu to Adama Toll Roads" and examined the candidate during open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Mater's of Science in Computer Science and Engineering.

____________________________	________________	_______________
Chairperson	Signature	Date
____________________________	________________	_______________
Internal Examiner	Signature	Date
____________________________	________________	_______________
External Examiner	Signature	Date

Finally, approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

____________________________	________________	_______________
Department Head	Signature	Date
____________________________	________________	_______________
School Dean	Signature	Date
_________________________________	________________	______________
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

First and foremost, I express my sincere gratitude to God for guiding me through my educational journey and bestowing upon me the wisdom, knowledge, and strength necessary to complete this thesis.

I am deeply indebted to my Advisor, Dr. Mesfin Abebe, for his unwavering support, insightful comments, and unwavering encouragement throughout the thesis writing process. His invaluable guidance and thoughtful advice have been instrumental in enhancing the quality of my work and enabling me to successfully bring this thesis to fruition.

I would also like to extend my sincere appreciation to my family, for their steadfast support, valuable ideas, and moral fortitude during this important milestone.

Lastly, I express my heartfelt gratitude to my friends and classmates for their timely assistance and unwavering support, which have been invaluable in the successful completion of this thesis.

Contents

List of Tables

List of Figures

List of Abbreviations and Acronyms

AdamW	Adaptive learning rate optimization algorithm
BCE	Binary Cross Entropy
BDD100K	Brekeley Deep Drive Dataset
CIOU	Complete Intersection over Union
CNN	Convolutional Neural Network
CNN	Convolutional Neural Networks
COCO	Common Objects in Context
DFL	Distributed Focal Loss
FCL	Fully connected layer
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
FPS	Frequency per Second (Inference Speed)
GPU	Graphical Processing Unit
GPU	Graphical Processing Unit
IOU	Intersection over Union
Ir0	Initial Learning Rate
Irf	Final Learning Rate
ITS	Intelligent Transportation System
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
mAP	Mean Average Precision

NMS	Non-Maximum Suppression
P	Precision
R	Recall
R-CNN	Region-Convolutional Neural Network
RELU	Rectified linear unit
RF-FPN	Residual Feature Pyramid Network
ROI	Region of Interest
RPN	Region Proposal Network
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
YOLO	You Only Look Once
YOLOV8	You Only Look Once Version 8

Abstract

Real-Time object classification and detection are crucial tasks in computer vision that address issues in transportation systems and provide a framework to enhance safety and efficiency in transportation monitoring and surveillance. The study focused on road patrolling operation system designed for the Addis Ababa (Tulu Dimtu) to Adamma highway toll road. The proposed system was implemented as the state-of-the-art deep learning algorithm, specifically the YOLOv8 (You Only Look Once version 8 algorithm, to enable efficient vehicle counting, identifying, and classification.

The proposed system offers essential real-time information, including object counting, detection, and classification, which can lead to enforcing transportation regulations and optimizing transport management. The system consists of three key stages: object counting, detection, and classification.

The experiment result showcases the effectiveness of the proposed system in providing valuable insights to road authorities, ultimately contributing to improved transportation services and road patrolling operations on the expressway. This research insights into the potential of object detection and classification algorithms, and real-time object detection in addressing and visualizing by tracking any events as valuable information for transportation monitoring applications. The results were obtained using highway video datasets and implementing the YOLOv8 algorithm to demonstrate the accuracy of the proposed vehicle and object identification and classification system for the transport system from Addis Ababa to Adama highway. The model has been trained using diverse highway datasets and **YOLOv8s** pre-trained model produced precision, recall and mAP50 are 95%, 95.5%, and 98.4% respectively. And **YOLOv8n** produced precision, recall and mAP50 are 94.8%, 95%, and 98% respectively. The result showcases the system's ability to accurately locate objects within video frames. Furthermore, object classification provides valuable information such as object type and class. These results show the algorithm's capabilities to identify and classify objects precisely, used to enhance transportation management and improved road services in the specified events.

Keywords: Object Detection, Real-Time, Computer Vision, Deep Learning, Automatic Vehicle Identification, and Yolov8

CHAPTER ONE

1. INTRODUCTION

1.1. Background of the Study

In modern technology Real-time object detection plays an important role in modern transport systems to provide timely information on transport systems, for better service and improved patrol operation, it's also crucial for overall transportation services. This study implements an automatic vehicle identification and classification system with YOLOv8 and OpenCV to classify, detect, and count objects on the highway based on the YOLOv8 algorithm. It provides real-time events information to road enterprise operators and their higher officials to transmit significant details through precise and quick detection from various highway objects like vehicles and pedestrians, making it easier to determine the decisions.

We have analyzed and implemented the YOLOv8 algorithm, for real-time object classification, detection, and counting with the well-known framework, from the YOLO (You Only Look Once) family of algorithms. YOLOv8 builds on this implementation with improvements such as feature pyramid networks (FPN) and anchor-based object detection to achieve best-in-class, object detection to deliver state-of-the-art performance (Terven & Cordova-Esparza, 2023)**.**

As a major travel route with regular transportation options, the toll roads on Addis Ababa to Adama highway provide different types of road services. On this path, there must be good facilities and traffic should flow smoothly. Traditional monitoring methods cannot always provide timely accurate detection of objects, which makes it difficult to respond urgently and preventive. The solution to all of these challenges is an advanced YOLOv8 system that we have proposed to implement and use. This algorithm is capable of high-precision object detection in real time, which significantly increases the performance of the transport system (X. Wang et al., 2023) (Mane et al., 2023).

The goal of this study is to verify YOLOv8-based system will work properly for object detection under real-time conditions along the Addis Ababa to Adama highway. It provides useful data by counting, detecting, and classifying objects on the highway and is used by road enterprise operators and officials to design better traffic facilities, management plans, and monitoring of moving objects that follow traffic rules, otherwise, there should be some measures induced.

1.2. Motivation of the Study

Addis Ababa (Tulu Dimtu) to Adama highway is a crucial dynamic route to have quality and patrolling operation on toll roads provider needs, a smooth flow of transportation while minimizing accident risks. However, traditional surveillance methods often do not provide on-time and precise object detection because of this making effective decisions in critical situations or responding to them will be delayed.

The result of the study will highly empower operational operators and their higher management with improved capabilities to manage and navigate the highway effectively. The proposed system is capable of identifying and categorizing objects automatically such as buses, cars, persons, and trucks making it highly useful real-time information. Based on this data, road enterprises can provide efficient services and traffic management strategies to optimize with support toward monitoring implementation measures as required by regulation on transport rules.

The second basic motivation for conducting this study is to visualize real-time live events by using the latest advancement in deep learning models and focusing on the YOLOv8 algorithm (Al Mudawi et al., 2023), to address specific challenges observed at the Addis Ababa to Adama highway. Real-time detection is also used to improve the overall transportation management and operational patrolling measures along a highway by developing a real-time object detection system.

In this research study, we have analyzed and developed real-time object detection systems for road authorities, and operators, traffic, and other stakeholders with practical and effective solutions for their specified tasks. The results of this study can contribute to improving transport management, patrol operation, and overall services in terms of a more efficient and safer highway environment.

1.3. Statement of the Problems

The complexity and diversity of highway transportation, characterized by high-speed movement and frequent occlusions occurring by vehicles, presents significant challenges for real-time object detection systems. Current object detection algorithms accurately identify and classify various objects while maintaining the efficiency necessary for real-time applications. In deep learning, there are limited situations of how the state-of-the-art methods, such as YOLOv8 in detecting and classifying multiple object types in scenarios moved by high speed and occlusion remains under-explored (Reis et al., 2023). This study seeks to address these challenges by evaluating and adjusting the capabilities of YOLOv8 in real-world highway scenarios, ultimately providing insights into its potential for enhancing transport management in real-time analysis (Lv et al., 2023).

These challenges must be overcome to ensure the safe and efficient operation of automatic vehicle identification on expressways (Talaat & ZainEldin, 2023). Without precise and up-to-date information on traffic patterns, and compliance with traffic regulations, transportation authorities and road enterprises face challenges in optimizing traffic management and transport services.

1.4. Research Questions

The study focused on the following key questions to achieve the stated objective and solve the indicated problems:

1. How does the accuracy of YOLOv8 in detecting vehicles and pedestrians on highways scenarios with different object detection models, such as Fast R-CNN, YOLOv5, and even YOLOv8's models nano, small and medium?

2. How effective is YOLOv8 in identifying and classifying multi-object types in dynamic highway environments?

3. What is the computational efficiency of YOLOv8 in real-time scenarios on highway video feeds and how does it perform under different frame rates?

1.5. Objectives of the Study

1.5.1. General Objective

The general objective of this study is to evaluate and enhance the performance of the YOLOv8 algorithm for real-time object detection and classification in highway environments, focusing on the accurate identification and classification of various vehicles and pedestrians under high speed and occlusion conditions.

1.5.2. The Specific Objectives

The specific objectives used to achieve the main objective are stated below:

- First, assess the accuracy of the YOLOv8 algorithm in detecting and classifying vehicle and pedestrians compared to different models, such as Fact R-CNN, YOLOV5 and YOLOv8 models.
- Additionally, investigate the efficiency of the YOLOv8 model architecture in dynamically detecting multiple object types such as buses, cars, trucks, and persons within highway scenarios. The impact of high-speed movement on detection performance will also be analyzed, with a focus on detection rates and response times. As we have stated the primary aim of this thesis was to investigate the state-of-the-art capabilities of the YOLOv8 model in detecting vehicles and pedestrians in highway scenarios. The focus was on enhancing object detection accuracy and robustness rather than solving speed detection limitations. Furthermore, the research emphasizes providing effective solutions for real-time detection to address immediate incidents. By live tracking the movements of identified and classified objects, the system can offer timely information to operational patrolling services, enhancing response efficiency.
- Furthermore, evaluate the accuracy of YOLOv8 in handling occlusions, identifying potential areas for enhancement in detection accuracy. Hyperparameter optimization was performed using automatic parameter selection to get an optimal point in both accuracy and speed. Ultimately, this study provides practical recommendations for deploying YOLOv8 in real-world highway scenarios, aiming to enhance transportation management and overall improved services.

1.6. The Scope and Limitation

1.6.1. Scope of the Study

The scope of the study covers as stated below:

- Evaluate, customize, and implement the performance of the YOLOv8 algorithm for the automatic vehicle and pedestrian identification system for the Tulu Dimtu to Adama highway.
- Comprehensive analysis of YOLOv8 models architecture specifically for highway object detection to correctly classify and identify objects by using different deep learning models as stated in the above sections.

- Explore the optimization of hyperparameters through optimization techniques to enhance model performance.
- Emphasis on the highway scenario, to which the algorithm will be tested for detecting various objects such as vehicles, and pedestrians.
- Detection, counting, and tracking objects real-time on expressways using the YOLOv8 algorithm by submitting instant information for the road operators and management for decision making.

1.6.2. Limitations of the Study

Just like other versions of the YOLO framework, YOLOv8 has also some constraints. It is limited in the following ways:

Without a large and diverse dataset, the model cannot learn effectively. Additionally, in highway, the representation of pedestrians is low, and the accuracy does not meet the maximum results. On the other hand, if the dataset for vehicles is not sufficiently large, the model couldn't achieve significantly improved accuracy and efficiency.

1.7. Significance

Road enterprises and law enforcement agencies are used to enhance their surveillance capacity. Being capable of doing real-time object detection and tracking on highways like vehicles, pedestrians, and hazards. It monitors road services, and traffic conditions and detects any abnormal incidents.

The application of YOLOv8 could be employed in transportation and infrastructure management to acquire real-time information on traffic patterns and count the number of vehicles entering and exiting the highway. Additionally, it can be used for efficient highway surveillance, enabling highway enterprises and their operational operators to be timely informed of both normal and abnormal conditions.

YOLOv8's real-time identification and classification are also used for emergency services for ambulance and fire departments to easily detect and respond to accidents, or any other emergencies on highways to reach the location in time and provide immediate help.

YOLOv8 is a critical building block of automated transportation systems for real-time environment perception of automated transport systems, which enables the system to address timely event information on highways. The fast and efficient object detection capability of

YOLOv8 is extremely important to ensure the safety and reliability of the automated transport system.

1.8. Organization of Thesis

The thesis is structured in the following ways:

Introduction: (chapter one) describes the background of the study and motivation, statement of the problem, research questions to be answered, methodology to follow procedures and methods, scope, and limitation to cover the range of study, analysis, and development the proposed system, and the result to be discussed. Literature review (chapter two) assessing the facts observed in the related area. Since YOLO (You Only Look Once) families and specifically YOLOv8 object identification and classification algorithm, are highly accurate and efficient for detecting, classifying, and counting purposes. However, running YOLOv8 on Kaggle environments involves resources such as necessary library packages, GPU/CPU, and pre-installed libraries or specialized AI accelerators, to ensure optimal performance and cost-effectiveness.

The methodology (Chapter Three), phases, approaches, method, and material to be used are incorporated to smartly implement the proposed system by following and using this methodology.

The proposed system (Chapter Four) discusses the state-of-the-art object detection capabilities to anticipate the solution for problems that are identified by formulating questionnaires in an appropriate way by deep analysis of the gap identified so far.

Implementation (Chapter Five) tasks to be implemented functionally by setting the necessary libraries, algorithms to be followed, and coding, especially for areas which is desirable for detection, and classification tasks to be implemented accurately and efficiently for YOLOv8 capabilities for highway scenarios.

The result and discussion (chapter six), present and discuss the outputs by comparing them with the other related algorithms to be efficient and the high accuracy will be saved to demonstrate the real world.

Conclusion and Future Work (Chapter Four), on this section we have concluded how the stated problem is solved and how can be solved, by choosing the optimal result obtained and showcasing the effectiveness and accuracy regarding to the other state-of-the-art algorithms.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1. Introduction

Deep learning neural network object detection and classification capabilities have seen prominent results and advancements in many areas such as action recognition, object detection, surveillance, and medicine image classification. Many researchers have worked on improving the effectiveness and accuracy of detection systems in machine learning algorithms. In addition to visual sensors, researchers have also looked into using audio and vibration sensors to have sufficient identification capabilities. Deep learning methods are used, in studies of real-time analysis by using the YOLOv8 algorithm to improve the efficiency of object detection. The benefit of the state of art algorithms is their ability to adapt to changing environments. They learn to recognize different weather conditions, lighting conditions, and road conditions to adjust their detection methods accordingly. This makes them particularly effective in areas where conditions may change frequently, such as highways or event tracking. Another benefit of machine learning algorithms is their ability to detect partially occluded or difficult-to-see objects or vehicles. For instance, they can detect vehicles or objects that are partially hidden by another object or that are moving quickly through a crowded area. This can improve road safety by live observing road operators and authorities to potential hazards or incidents that should have been noticed on time to respond for an event otherwise there may be additional or extra damages. However, there are also some challenges to algorithms, they require large datasets and time to be trained.

Techniques from deep learning, such as YOLOv8 convolutional neural network, have become popular for detection and real time visualization (Mane et al., 2023). The biggest difficulty again is the performance of processing data efficiently while maintaining high accuracy in detecting small and moving high-speed objects. Various methods have been suggested by researchers to achieve this, such as using region-based CNN (R-CNN) and the YOLO (You Only Look Once) algorithm (Huang et al., 2024).

R-CNN (Region-based Convolutional Neural Network) is one of the first object detection algorithms, which solved the problem of choosing many areas by extracting 2000 region proposals from an image using selective search and then inputting them to a convolutional neural network

(CNN) that extracted features. The next step involved using an SVM which made use of those features to classify whether there are objects in the region proposals or not. R-CNN also made predictions about the offset values to improve bounds (R. Wang et al., 2021).

Fast R-CNN (RoI) went further than R-CNN by having a CNN that took in an entire picture to make up convolutional feature maps. Thereafter, Region of Interest (ROI) pooling is the altered shape of region proposal into some fixed size that would be accepted by a fully connected layer. Since Fast R-CNN has fed 2000 region proposals through CNN, therefore it's facilitated faster training and testing times.

Because of the elimination of the selective search algorithm, Faster R-CNN has improved both object detection, speed, and performance. As an alternative, a separate network was utilized for directly forecasting region proposals from a convolutional feature map. Consequently, these suggestions were then resized by RoI pooling and used in classification and bounding box prediction. Due to eliminating the selective search stage, Faster R-CNN became faster during testing than its forerunners.

Identification and classification through the proposed model took different techniques. Instead of regions, YOLO split the images to grids where several bounding boxes were assigned per grid cell. The class probabilities and offset values for these bounding boxes are estimated by a single convolutional network on its YOLO was able to run at an amazing speed processing up to 62 frames per second, but it struggled with small object detection due to spatial constraints.

The introduction of real-time object detection using YOLO families, especially YOLOv8 has brought about new advancements in deep analysis of images using convolutional networks. Developed by Joseph Redmon and his team (Redmon & Farhadi, 2020), YOLOv8, a state-of-the-art object detector, it can be used in real time. It is the most popular algorithm in computer vision in terms of effectiveness speed and accuracy in all environments from other object detection models.

YOLOv8 architecture uses a single-stage method, processing the whole images, passes in one pass. Its architecture allows rapid and efficient object identification compared to traditional algorithms. YOLOv8 involves advanced methods, such as anchor boxes and non-maximum suppression to enhance object identification and eliminate redundant detections.By predicting both the bounding box coordinates and class labels simultaneously, YOLOv8 achieves high accuracy in object detection, classification, and counting.

Developing YOLOv8, an algorithm for real-time object identification and classification consists of environment setup, loading the data to a pre-trained model, and analyzing frames from an image and video. The Ultralytics library provides a convenient interface for working with YOLOv8 and loading datasets intomemory. Each frame is then analyzed using the YOLOv8 model to detect multiple objects in real time, providing accurate bounding box coordinates and class labels for each detected object (Bui et al., 2024; X. Wang et al., 2023).

YOLOv8 is used in various industries. For autonomous cars, intelligent transport systems, security, and retail analytics (Ward, 2023).

In automatic vehicle classification, detection YOLOv8 outperforms in diverse conditions and high-speed object movements. Its fast processing speed makes it ideal for applications that require ereal-time image and video analysis. By providing accurate bounding box coordinates and class labelsfor detected objects, YOLOv8 facilitates image understanding and enables further analysis and decision-making based on the identified objects (R. Wang et al., 2021).

YOLOv8 also extends its capabilities to real-time video analysis. Each frame of a video can be processed to accurately detect and track objects in real time, opening up a wide range of applications in domains that require video surveillance, action recognition, and object tracking. YOLOv8's streamlined architecture and optimized algorithms enable fast processingof video frames, ensuring real-time analysis without compromising performance (Wu & Dong, 2023.

Table 1: Summary of strengths and weaknesses/challenges in machine learning techniques detection approaches used.

Machine Learning Techniques	**Strengths**	**Weaknesses/Challenges**
R-CNN, Fast R-CNN, Faster R-CNN, YOLO Object Detection Algorithms	R-CNN: Selective search reduces the number of regions to classify. - Fast R-CNN: Feeds the input image to the CNN, making it faster than R-CNN. - Faster R-	R CNN: Slow training and testing time, fixed selective search algorithm. Fast R CNN: Region proposals can

	CNN: Eliminates the selective search algorithm, improving performance. - YOLO: Faster than other object detection algorithms, processing 45 frames per second.	slow down the algorithm. - Faster R CNN: Still relies on region proposals, although faster than previous methods. YOLO: Struggles with small objects and spatial constraints.
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks	Introduces a Region Proposal Network (RPN) that shares full-image convolutional features, enabling cost-free region proposals. - Achieves state-of-the-art object detection accuracy on various datasets. - Faster R-CNN and RPN are foundations of 1st-place winning entries in competitions.	Relies on region proposal algorithms, which can be a bottleneck. Requires training end-to-end for high-quality region proposals.
YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition (5 December 2023)	Lightweight convolution SEConv to reduce parameter count and speed up the detection process. - SEF module for multi-scale object detection. - EMA attention mechanism for enhanced feature extraction. - Additional prediction head for tiny-object detection. - Transformer prediction head for capturing global and contextual information. - Wise-IoU loss function to mitigate adverse gradients.	The trade-off between speed and accuracy Challenges with small objects and low contrast Difficulty in handling occlusions and overlapping objects Requires adequate training data and computational resources
Zero-Shot Object Detection with YOLOv8-World: A New Era of AI	Dynamic class specification through custom prompts. Real-time, open-vocabulary capabilities (37).	The trade-off between speed and accuracy Challenges with small objects and low contrast

Flexibility (February 15, 2024)		Difficulty in handling occlusions and overlapping objects Requires adequate training data and computational resources

2.2. Deep Learning

Deep learning is the subset of machine learning that uses neural networks with many layers, such as R-CNN, Fast R-CNN, and Faster R-CNN, start by identifying areas, learning from large amounts, and then classifying and bounding box those areas. Although they are very accurate, they are slower than detectors that process the entire image at once because they need to suggest areas and go through the network multiple times. Detectors that analyze the whole image at once, like YOLO and SSD (Neamah & Karim, 2023), treat detection as a task to predict classes and box shapes for each part of the image in one go. This makes them muchquicker and suitable for situations that need immediate responses, such as self-driving cars.

In systems like those for highways, where functions such as keeping in the correct path and reducing collisions need to happen quickly and accurately, object detection must be both precise and fast. While area-focused methods are very accurate, detectors that analyze the whole imageare more suitable because they are faster. Among these whole-image detectors, YOLO has consistently achieved a good balance of speed and accuracy in intelligent transportation systems.

2.3. YOLOv8

YOLOv8 (You Only Look Once version 8) is an advanced object detection algorithm to identifying, classifying and counting an objects in images and videos, which is greatly advanced in which that allowing detection happen in real-time. Created by Joseph Redmon and his team, YOLOv8 (Redmon & Farhadi, 2020), it is the first choose from popular algorithms. Its special design allows it to process andanalyze the whole image at once, which makes it quicker and more effective than older methods for automatic vehicle identification and classification.

YOLO v8 could be integrated into different intelligent system is used across many industries.

Autonomous cars need to recognize and respond to obstacles around them. Security use it to quickly identification of suspicious activities due to its real-time object detection feature (Sudharson et al., 2023). Furthermore, YOLOv8 is used in analyzing imagesand videos in various sectors, including healthcare and retail.

In this section, we will look at the YOLOv8 architecture, how it is used for detecting objects in real-time, and its uses in analyzing images and videos. We have also discussed the advantages, challenges, and future improvements of YOLOv8 (Talib et al., 2024).

2.4. Architecture of YOLOv8

YOLOv8, which stands for You Only Look Once version 8, is designed to perform object detection quickly and accurately in real time. It builds on the improvements made in earlier versions, using different methods and enhancements to boost its ability to detect objects (Talib et al., 2024a).

The main parts of the YOLOv8 structure are:

2.4.1. Backbone Network: YOLOv8 uses a backbone network, like CSPDarknet-53, to pull out important features from input images or video. This network helps in getting the necessary information for correctly identifying objects in the images or video.

2.4.2. Feature Pyramid Networks (FPNs): YOLOv8 includes feature pyramid networks to deal with objects of different sizes. FPN helps the model to identify different object sizes and types in diverse conditions, it combining layers from different parts of the backbone network.

2.4.3. Spatial Attention: YOLOv8 adds spatial attention modules to focus on important areas in an objects. These modules help to allocate more attention for important object regions, which improves the accuracy of object detection**.**

2.4.4. Context Aggregation: YOLOv8 uses context aggregation modules to capture information about the overall context. These modules enable the model to understand the connections between objects and their environment, which boosts the performance of object detection.

2.4.5. Detection Head: YOLOv8's detection head includes multiple layers that use convolutions to determine where objects are located (bounding boxes) and what types of objects they are (class probabilities). This part of the system is responsible for producing the final output of detected objects.

2.4.6. Anchor Boxes: YOLOv8 uses anchor boxes, which are predetermined shapes of bounding boxes in various sizes and proportions. These anchor boxes assist the model in detecting objects that come in different shapes and sizes.

Structures of the Main Block

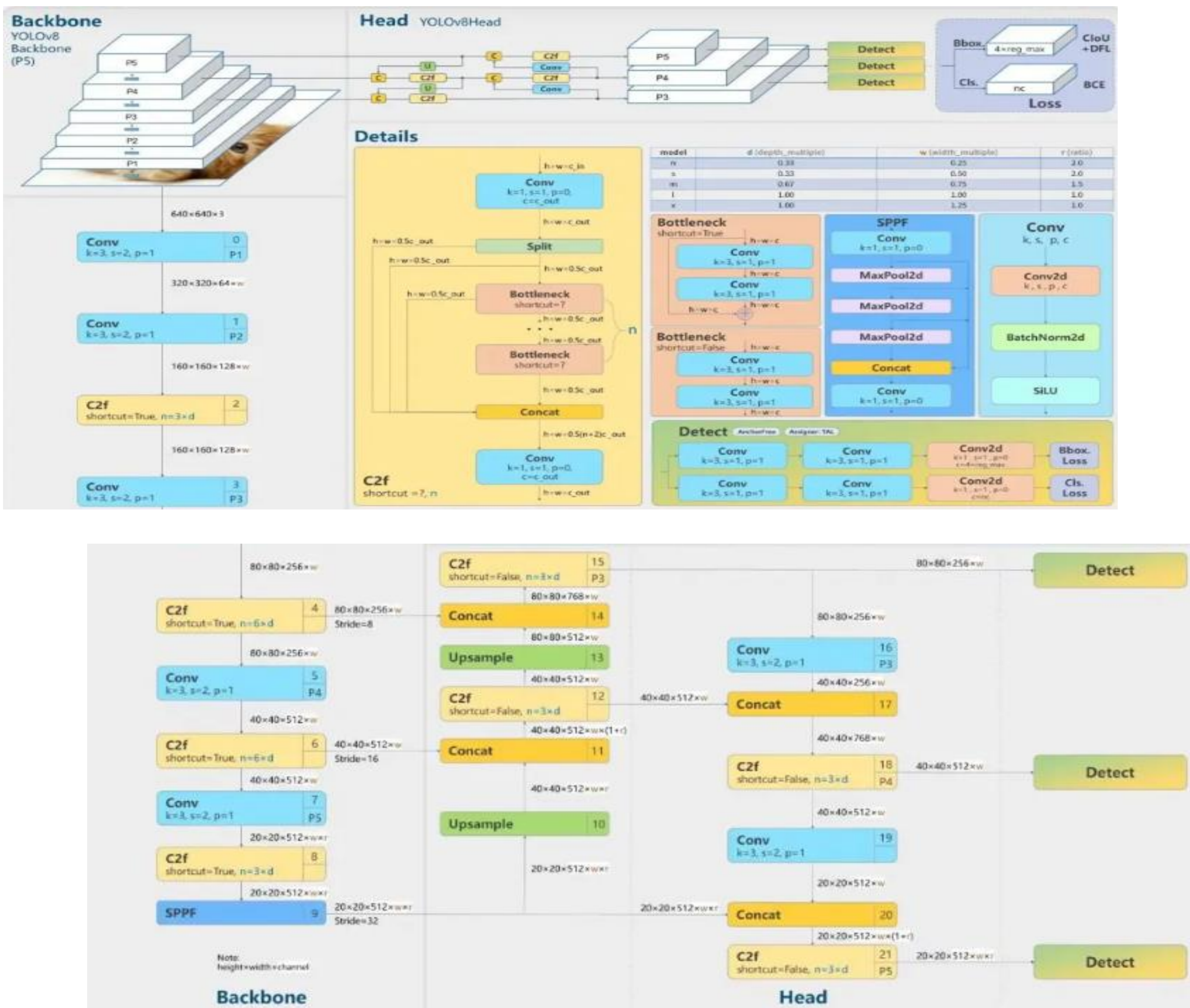


Figure 1 YOLOv8's architecture

The main feature of YOLO v8 consists of mosaic data augmentation, anchor-free detection, C2f layers, decoupled head, and the modified loss function.

2.4.7. Mosaic Data Augmentation

Similar to YOLOv4, YOLOv8 employs mosaic data augmentation, which combines four images to offer the model-enhanced context information. In YOLOv8, the augmentation process is modified to cease during the last ten training epochs to boost performance.

2.4.8. Anchor-Free Detection

YOLOv8 has transitioned to anchor-free detection to enhance generalization. The issue with anchor-based detection is that the predefined anchor boxes can slow down the learning process for custom datasets.

By using anchor-free detection, the model directly predicts the mid-point of an object and minimizes the number of bounding box predictions. This approach accelerates Non-max Suppression (NMS), a pre-processing step that eliminates incorrect predictions.

2.4.9. C2f Module

The backbone of the model now features a C2f module rather than a C3 module. The key distinction is that in the C2f module, the model combines the outputs of all bottleneck modules. Conversely, in the C3 module, only the output from the final bottleneck module is used.

A bottleneck module is composed of bottleneck residual, which helps to lower computational costs in deep learning networks. This enhancement accelerates the training process and benefits the flow of gradients.

2.4.10. Decoupled Head

The diagram above shows that the head no longer combines classification and regression tasks. Instead, it addresses these tasks separately, leading to improved model performance.

2.4.11. Loss

Misalignment can occur because the decoupled head separates the tasks of classification and regression. This means that while the model may accurately locate one object, it could classify a different one.

To address this issue, a task alignment score can be introduced. This score helps the model differentiate between positive and negative samples. Specifically, the task alignment score is calculated by multiplying the classification score with the Intersection over Union (IoU) score. The IoU score indicates how accurate a bounding box prediction is.

Using the alignment score, the model identifies the top-k positive samples. It calculates the classification loss using Binary Cross-Entropy (BCE) and the regression loss using Complete IoU (CIoU) and Distributional Focal Loss (DFL). The BCE loss measures the discrepancy between the actual labels and those predicted by the model.

The CIoU loss evaluates the predicted bounding box in relation to the ground truth, taking into account the center point and aspect ratio. In contrast, the distributional focal loss enhances the distribution of the bounding box boundaries by placing greater emphasis on samples that the model incorrectly labels as false negatives.

2.5. Real Time object tracking in Videos with YOLOv8

YOLO v8 enhances real time object identification by analyzing videos. It uses the YOLOv8 model to process each video frame, accurately detecting and tracking objects a step further by extending its capabilities. This technology is useful for areas like security cameras, recognizing actions, and tracking objects.

YOLOv8 makes the actual object identification and tracking with video streaming quicker and better. It gives immediate information about where objects are and what they are, helping to keep an eye on video feeds continuously. The movement of any objects by identifying a specific hyperparameters to know the exact object locations. YOLOv8 enables us to identify exact time and actual movements in the high accuracy and speed.

Organizations can improve their security systems, enhance safety measures, and extract valuable information from video data by using YOLOv8 for real-time video analysis. YOLOv8 can be instrumental in detecting suspicious activities as they occur, observing crowd movements in public areas, or studying traffic patterns. It plays an essential role in helping to understand and react to ever-changing situations.

2.6. Related Works

Actual time object identification in YOLO and Webcam sensors is increasing detection and classification tasks. This work will provide the actual movement and time of objects using YOLO and webcam integration. It explains the concept of YOLO, its advantages, and its applications in various fields, including highway surveillance. This work covers the setup of the environment, capturing frames from webcam video data, and operating YOLO with the Ultralytics library (Shinde et al., 2018).

A comprehensive framework for object identification and classification in a transport surveillance system. The models use detection algorithms, including YOLOv8, for real-time object detection and localization (Neamah & Karim, 2023). It focuses on three main stages of a patrolling system: identifying, classifying, and counting. The work highlights the importance of object detection in extracting relevant information from image or video inputs by surveillance cameras.

The effectiveness of YOLOv8 in object detection tasks has been demonstrated in several studies. In a comprehensive evaluation, YOLOv8 outperformed previous versions of YOLO and other object detectors, such as Faster R CNN and SSD, the MSCOCO dataset. Specifically, YOLOv8 achieved a mean average precision (mAP) of 51.8%, an outstanding mAP of 49.6% achieved by YOLOv5, the previous version of YOLO. Moreover,YOLOv8 exhibited superior inference speed, processing images at a speed of 142 frames per second (FPS) on a single GPU, compared to 86 FPS for YOLOv5 (R. Wang et al., 2021).

In the context of highway surveillance and road safety applications, (Terven & Cordova-Esparza, 2023), evaluated the performance of YOLOv8 for speed breaker detection for custom dataset of highway images. Their study compared YOLOv8 with YOLOv5 and Faster R-CNN algorithms demonstrates the result of YOLOv8 achieved highest precision is 92.7% and recall is 94.1%,outperforming the other models. Additionally, YOLOv8 demonstrated greater real-time performance, processing images at a rate of 45 FPS, making it suitable for real-time highway monitoring systems. Road segmentation is a critical task for both autonomous driving and traffic analysis, benefiting from the advancements made with YOLOv8. (Talib et al., 2024b) Proposed a YOLOv8-based road segmentation model and assessed its performance using the KITTI road dataset. Their model achieved an impressive intersection-over-union (IoU) score of 87.3%, surpassing previous state-of-the-art methods like SegNet (56.1% IoU) and ENet (58.3% IoU). Also, the YOLOv8-based model demonstrated real-time capabilities, processing images at a rate of 30 seconds per frame (fps), making it suitable in actual time road segmentation systems in highway surveillance systems.

Although YOLOv8 has shown promising results across various object identification, and segmentation activities, it is important to understand that the algorithms' capabilities may differ in situations where the specific application, dataset, and environmental conditions. Elements such as lighting, occlusions, and scene diversity can affect the model's accuracy and speed. (Huang et al., 2024)

In the summary of related works presented in Table 2 below, researchers have investigated using YOLO models, including YOLOv8, in highway surveillance and road safety scenarios. The researchers also proposed hybrid method integrates YOLOv8 with semantic segmentation for road object detection and segmentation (Aquitan et al., 2022.). This approach achieved a mean Average Precision (mAP) of 58.2% and IoU of 81.4% on the BDD100K dataset, outperforming any state-of-the-art methods. They also emphasized the potential of their method for real-time applications in autonomous driving and traffic monitoring.

A YOLOv8-based system was developed for the detection and tracking of vehicles on highways. Their system attained a high detection accuracy is 94.7% and an actual time processing time of 40 FPS. The authors emphasized the significance of precise object identification and classification for transport management, accident prevention, and law enforcement in highway surveillance systems.

While these studies illustrate the promising performance of YOLOv8 in various highway surveillance tasks, it is essential to consider the challenges of real-world implementation. Factors such as high-speed movements, occlusions, and weather conditions can influence the model's performance. In particular, on vehicle classification for highways using YOLOv8, a reported accuracy of 91.3% was achieved by employing a multi-task YOLOv8 approach that simultaneously performed object detection, classification, and tracking, yielding a map of 60.4% and 34 FPS. (Talaat & ZainEldin, 2023).

2.6.1. Summary of the Related Work

The table that is underneath depicts early studies in real-time or object detection using machine learning techniques. The papers that we have reviewed are onwards from the year 2015.

Table 2 Summary of related works

The Reference	Title	Methodology used	Accuracy	Gaps
(Girshick et al., 2015)	Fast R-CNN	Region-based CNN; integrates region proposal network with CNN	mAP 70.0 on PASCAL VOC	Slower than YOLO models; dependency on region proposal network
(Jocher et al., 2021)	YOLOv5: A state-of-the-art object detection model	YOLOv5 framework; single-stage detection with enhancements	mAP 50.0 on COCO	Less effective for small object detection; slower than YOLOv8
(Redmon& Farhadi, n.d.)	Real-time object detection with YOLO and Webcam	YOLO object detection and webcam integration	Not provided	Lacks evaluation of real-world performance Focused on setup and conceptual aspects
(Aquitanet al., n.d.)	Robust Framework for object detection with a transport surveillance system	YOLOv8 for object detection, Tracking and Recognition	Not provided	Lacks detailed performance evaluation and Focuses on the overall framework

(Terven & Cordova-Esparza, 2023)	Comprehensive Evaluation of YOLOv8	Evaluation of YOLOv8 on MS COCO dataset and Comparison with previous YOLO versions and other detectors	mAP: 51.8% FPS: 142	Evaluation of generic dataset, not specific to highwayransportation scenarios
(Bui etal., 2024)	Speed Breaker Detection using YOLOv8	YOLOv8 for speed breaker detection on highway images, Comparison with YOLOv5 and Faster R-CNN	Precision: 92.7%Recall: 94.1% FPS: 45	Limited to a specific task of speed breaker detection
(Terven & Cordova-Esparza, 2023)	YOLOv8-based Road Segmentation	YOLOv8 for road segmentation on KITTI dataset, Comparison with SegNet and ENet	IoU: 87.3% FPS: 30	Focused on road segmentation, not object detection
(Lv et al., 2023)	Hybrid Approach for Road Object Detection and Segmentation	Combination of YOLOv8 and semantic segmentation, Evaluation on BDD100K dataset	mAP: 58.2% IoU: 81.4%	Evaluation on a generic dataset, not specific to highways
(Farkaš, n.d.)	Vehicle Detection and Tracking using YOLOv8	YOLOv8 for vehicle detection and tracking on highways, Real-time performance evaluation	Detection Accuracy: 94.7% FPS: 40	Limited to vehicle detection and tracking, lacks evaluation of other object types.

(Al Mudawi etal.,2023)	YOLOv8-based Vehicle Classification on Highways	YOLOv8 for vehicle classification (car, truck, motorcycle, etc.), Evaluation on highway traffic dataset	Accuracy: 91.3%	Focused only on vehicle classification, lacks other object types
(Bui etal., 2024	Multi-Task YOLOv8 for Highway Monitoring	YOLOv8 for simultaneous object detection, classification, and tracking, Evaluation on diverse highway scenarios	mAP: 60.4% FPS: 34	Complexity of the multi-task approach may impact real-time performance
The proposed work	Real-time object detection using YOLOv8	YOLOv8 for automatic vehicle identification system (AVIS)	Precision: 95.1% Recall: 95.5% mAP50: 98.2% FPS: 127.27	Can detect multiple classes and small objects with real-time

The summary of the above table contains the related works, including the methodologies used, the reported accuracy, and the identified gaps or limitations. This research identifies several gaps in the existing literature on real-time object detection for highway transport systems. Notably, many current models struggle with accuracy in diverse conditions, the detection of small or partially obscured objects, and efficient processing for real-time applications.

To address the challenges, we used different techniques such as optimizer AdamW, learning rate 0.00125, momentum 0.9, and weight decay 0.000594 for 50 iterations, this work focuses on enhancing detection capabilities by using YOLOv8, that fills the gaps of the related works. By improving accuracy under multiple object detection and classification, small and partially occluded objects. And the processing speed is also, enhanced and optimized to ensure timely response real-time monitoring, and analysis of road security.

CHAPTER THREE

3.1. RESEARCH METHODOLOGY

3.1. Overviews

In this chapter we elaborate on the approaches, and methods, to follow, and the design, material, and tool to be used are considered to analyze and anticipate the solution space

The research problem involves developing an actual time and movement object identification and classification system for the Addis Ababa to Adama highway route using YOLOv8. The objective is to accurately classify, detect, and count vehicles and pedestrians to enhance the safety and efficiency of transport management. The methodology consists of the following steps, which were implemented world real-world highway scenarios.

3.2. Dataset Acquisition and Description

For this study, we collected video data of the road, each ranging in length between 34 to 44 minutes, which were recorded at 8:00 - 10:00 am morning and 2:00 - 6:00 pm from Oct 2 - 8/2023, hence acquired over 84 hours of the data. Additionally, the videos of the road in the sunlight varying were also recorded for testing.

The diverse dataset of highway videos from three different toll roads of the highway namely Tulu Dimtu, Bisheftu, and Adama of entering and exiting the road, collecting a wide range of objects and vehicles. The 3,995 images are collected, captured webcam-captured dataset, and which is then manually annotated to identify 4 main classes of objects such as bus buses, cars, person people, and trucks for the toll roads.

We classified the objects based on their axle length, the character description of vehicles, and toll rate (birr/km) as we have stated above the 4 classes are in Table 3: classification patterns.

Table 3 Classification of Vehicles

TYPE	Classification Parameters						Toll rate (birr / km)
	I-Axle number	II-Tire number	III-Height of vehicles head(m)	IV-Axle distance (m)	Character description of vehicles	Remarks	
1	2	4	<1.3	<2.4	Small automobiles Cars, Jeep , L/Rover, taxi Pick up	At the same time to satisfy the parameters I, II, and III or IV in any conditions	0.66
2	2	4	≥1.3	≥2.4	Minibus	At the same time to satisfy the parameters I, II, III and IV conditions	0.66
3	2	6	–	–	Medium Bus ، ISUZU		0.66
4	3	–	–	–	Big size Bus Medium Truck		0.79
5	4	–	–	–	Heavy Truck trailers		0.92
6	5	–	–	–	Heavy Truck trailers		0.92
7	≥6	_	_	_	Heavy Truck trailers		0.92

TYPE 1 /V-1/

TYPE 2 /V-2/

TYPE 3 /V-3/

Figure 2 Vehicle classification types

To increase the dataset, we have employed data preprocessing of auto orient and resize stretch to 640 by 640 and augmentation techniques, including horizontal flip, exposure, saturation, and brightness. This resulted in a final dataset of 9,388 images, which was used for training, testing, and evaluating the object detection models.

The video dataset about the highway route between Addis Ababa (Tulu Dimtu) to Adama is collected from cameras installed at Tulu Dimitu, Besheftu, and Adama toll roads. The datasets that are collected, from both directions of the highway webcams is containing diverse datasets. The proposed model was tested with prepared datasets that are most likely to have seen on the highway.

Before performing fine-tuning of the YOLOv8 model, there are some critical stages needed to get the model in place and ready for training on the custom object detection task. It is important to gather data with high quality, create video frames, and label them using bounding boxes. During dataset preparation, we have linked to the required path to specific dataset settings and set up the configuration of the dot yaml and necessary files. The data preprocessing and fine-tuning is done with the help of the Ultralytics necessary libraries and there has been loading the model, training on the given dataset, and evaluating its performance depending on the pre-trained weights of the YOLOv8s' models.

The highway dataset is used in the actual time and movement identification of objects, with the diverse highway scenarios for a variety of objects of videos that represent the actual image, which the algorithm identifies and classifies. The datasets that have been preprocessed for the best training models. So the collected videos of different objects that could mostly appear on an expressway are considered.

3.3. Data Preprocessing

The datasets have been used to carry out data preparation which involves preprocessing for training and evaluation. The individual frames were then extracted from the videos and each of those frames was resized to a resolution suitable for real-time processing. The frames were then converted to the suitable format for the next processes or steps.

Figure 3 Showcase of video's frame images

3.4. Dataset Annotation

The process of grouping and naming it the categories the highway's object images with bounding boxes and labels that we have to detect. We have used a popular tool[1], for data labeling Roboflow which allows the user to create classes and annotations on images and assign labels to them. The labeled data is saved in the YOLOv8 model, which consists of image files, and .txt labeled files for each image in the train, test, and valid datasets. Each .txt file contains a bounding box and its corresponding classes in the x center, y center, width, and height of the objects.

In this research, we have utilized Roboflow to manually create classes and annotate images, which played a crucial role in preparing a high-quality dataset for the YOLOv8 model. By leveraging Roboflow's intuitive interface, we have enabled to efficiently define specific classes relevant to the detection tasks, for our classes' objects. This precise classification ensures that the model learns to recognize and differentiate between various object types effectively.

A key advantage of using Roboflow is its support for collaborative features, which significantly enhances the dataset-creation process. By creating user roles, we were able to create and annotate our objects classes and Roboflow also enabled us to involve team members in the annotation

[1] https://app.roboflow.com/object-detection-phgi2/object_detection.yolov8-z0yri/1

process, allowing multiple users to contribute to the project simultaneously. This collaborative approach not only accelerates the dataset preparation but also fosters a shared understanding of the annotation guidelines among team members.

Images from video stream datasets were annotated by us manually which, is time taking. Each frame was carefully labeled in bounding boxes in specific object interest, namely bus, car, person, and truck.

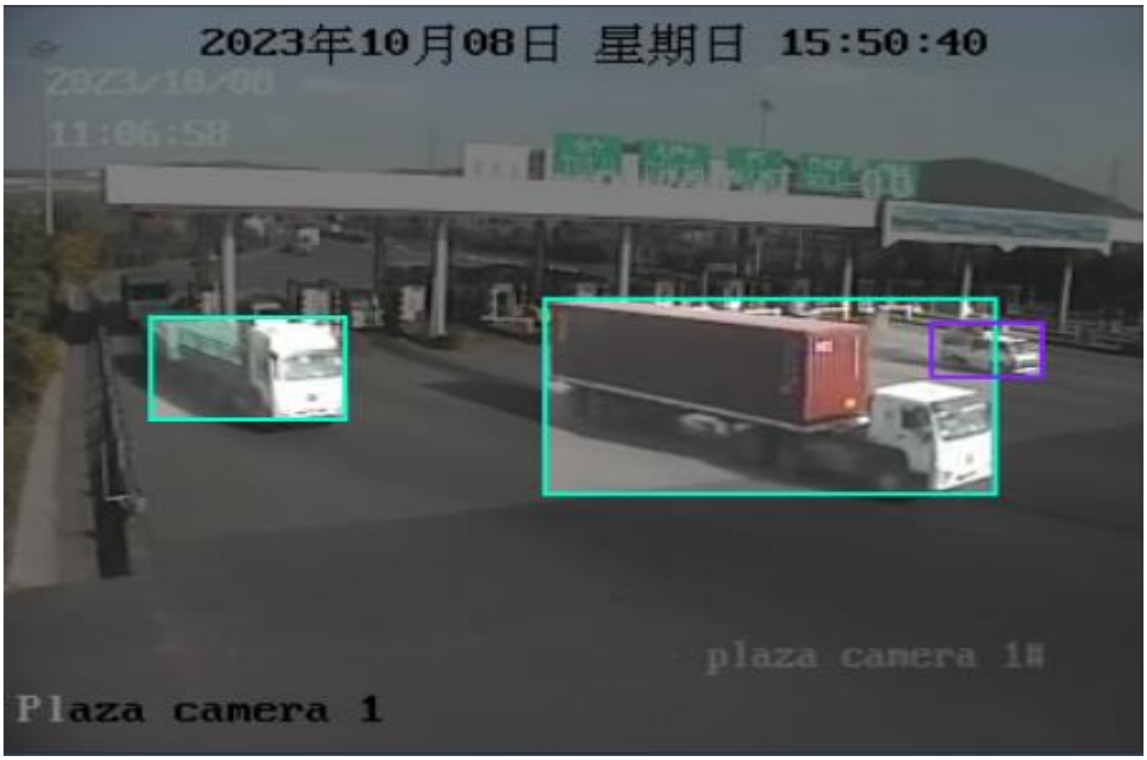


Figure 4 Showcase of annotated video frame images

3.5. Data Preparation

This involves splitting datasets for training, valid, and testing sets. The datasets have been converted to specific file types. In adopting model training. This setup is the separate folders in the ratio 80%, 10%, and 10% respectively for training, valid, and testing files to ensure balanced dataset for the model optimization techniques.

The YOLOv8 model training process begins from a .yaml file. Consists of important data, like path, for our datasets, number of classes, and name of the classes, model, and hyperparameters for facilitating oriented information to identify and classify objects accurately.

3.6. Data Augmentation

The technique to enhance the detection and generalization of the model for the datasets, some of the augmentation techniques that we applied, include flip horizontal, rotation, saturation, and exposure. The augmented dataset helped improve the accuracy and speed of the algorithms to capable and handle various real-world contexts.

A process of generating new training samples by applying transformations to the existing labeled images. This helps the proposed models for adoption in a large number of datasets and increase the model's accuracy. We used Roboflow for data augmentation, which allows us to apply a variety of transformations to the datasets, visualize, and generate different models.

3.7. Model Architecture

The YOLOv8 architecture is characterized by its modular design, allowing for the creation of scalable variants to cater to different application requirements. The key components of the YOLOv8 architecture include:

Backbone Network: YOLOv8 employs CSPDarknet53 as its backbone architecture. This architecture includes 53 convolutional layers and 3 residual blocks. The Cross Stage Partial (CSP) module in CSPDarknet53 is responsible for feature extraction. This module consists of two branches: a main branch and a residual branch. The design allows for the extraction of additional features from the input data without adding more parameters. The focus layers decrease the spatial resolution of the feature map while increasing the number of channels. Overall, CSPDarknet53 enhances the efficiency of YOLOv8, resulting in improved speed and accuracy

Neck: The YOLOv8 model employs the C2F module in its neck architecture. This module combines the outputs from convolutional layers to form a feature map that retains the details of the input image. A sequence of convolutional layers then processes this feature map to integrate high-level features with contextual information. The C2F module enhances the accuracy of YOLOv8 by merging features from various layers of the backbone.

Head: The head plays an essential role in identifying and forecasting classes and objects within an image. The architecture of the YOLOv8 head consists of two key components: an anchor-free detection system and a decoupled head. The anchor-free approach is centered on predicting the location of objects. The decoupled head is divided into two sections: a classification branch that determines the class of each object, and a regression branch that calculates the bounding box coordinates for each object. This head architecture offers greater flexibility compared to earlier versions of YOLO, making YOLOv8 adaptable for a range of object detection tasks.

Anchor-Free Detection: Unlike previous YOLO versions, the YOLOv8 model uses an anchor-free detection approach, which removes pre-defined anchor boxes, this simplifies models in increasing accuracy and efficiency by its flexibility.

Mosaic Data Augmentation: The YOLOv8 model leverages data augmentation in mosaic techniques, which adds multiple training objects to facilitate better information for the model and improve its generalization capabilities.

3.8. Hyperparameter Tuning

The training of the YOLOv8 model involves the following key hyperparameters and techniques:

- **Learning Rate**: The leering rate at the initial point, and a cosine annealing scheduler are useful to gradually decrease the learning rate during training.
- **Batch Size**: Depending on the hardware resources available, the batch sizes is typically set automatically for efficient training.
- **Epochs**: The number of training epochs is usually set between 50, depending on the complexity of the dataset and the desired level of performance.
- **Optimizer**: YOLOv8 uses the Stochastic Gradient Descent (AdamW) optimizer with momentum, which is effective for object detection tasks.
- **Loss Function**: The model employs a combination of losses, including classification loss, regression loss, and objectless losses, to optimize the model's performance.
- **Data Augmentation**: In addition to Mosaic data augmentation, YOLOv8 also utilizes other methods of random scaling, rotation, and also flipping to increase the diversity of the training data and increase the model's accuracy.

With the Ultralytics library, we are able to conveniently load the YOLOv8 model and train it using our annotated dataset. When we execute the training code, the model begins to learn how to identify the specific objects we annotated. Throughout the training process, the model consistently enhances its accuracy and comprehension of the characteristics of the objects.

The YOLOv8 model was trained using the annotated and augmented datasets. The frames and corresponding annotations were getting optimize the models layers through trains. Transfer learning was employed by initializing the model with COCO pre-trained weights on large-scale object detection datasets.

The trained YOLOv8 model was dataset datasets separate validation dataset. Cross-validation techniques were applied to get best the model performance that simplifies unseen objects. Hyperparameters tune was performed to optimize the model process by adjusting parameters such as optimizer AdamW, Learning rate, Momentum, Weight decay, Batch size, Image size, Epochs, and Data loader workers from the model:

3.9. Hardware and Software Specifications

To implement and deploy the YOLOv8 model on the platform independently, the following hardware and software specifications are typically considered.

Table 4 Hardware and Software Specifications

Resources	**Specification**
Memory	29.5 GB RAM
Storage	124.75 GB storage
Python Version	Python 3.10
Deep Learning Frameworks	TensorFlow Lite 2.6.0, Ultralytics YOLOv8 8.2.92 (optimized for edge deployment)
Additional Libraries	NumPy 2.0.0, OpenCV 4.6.0, Matplotlib 3.3.0
Power Consumption	Optimized for low power consumption, typically less than 5W

3.10. Performance Evaluation

We have tested YOLOv8 models, from Ultralytics libraries YOLO has released scaled variants to accommodate different application requirements. They found that the YOLOv8 Small model, achieved high precision and robustness in both accuracy and FPS, and the Nano model was high in FPS, but slightly low accuracy. So we have selected a small model reference to appropriate real-time and relatively high accuracy for the proposed application, the best result for real-time object detection. While the pre-trained model approach transfer learning showed a good result regarding promising accuracy and efficiency, we have noted that the complexity of the model and its real-time object detection performance was assessed by assessing the precision, recalls, and mean average precision (mAP) The obtained results as compared to the other object identification and classification algorithms were then evaluated to measure viability YOLOv8 for the mentioned transport system. Hyperparameter tuning was done on the evaluation results to get the best results.

YOLOv8, an advanced object detection model, identifies objects within images and videos. This technology can also be used for videos, allowing for real-time object detection functions. Although the method is like image detection, YOLOv8 analyzes each frame of the video separately, identifying objects and delivering precise outcomes.

The model effectively detects and determines objects in real time, ensuring it retains the impressive detection accuracy and speed associated with YOLOv8.

The model effectively detects and locates objects in real time, preserving the impressive detection accuracy and speed that YOLOv8 is recognized for

YOLO v8 algorithms process video to object detection and classification as per the following steps:

1. Input: A video file in a supported format.
2. Processing: YOLOv8 analyzes the video by breaking it down into individual frames and processing each frame in order using its deep neural network architecture.
3. Object Detection: For each frame, YOLOv8 identifies objects according to their categories and provides the coordinates of the bounding boxes for localization.
4. Output: The video is enhanced with bounding boxes around the detected objects, enabling users to see and examine the objects in each frame. With YOLOv8 object detection in videos,

users can explore numerous possibilities for video analysis and make real-time decisions. Whether it involves observing traffic patterns, recognizing objects in security footage, or analyzing customer behavior in retail environments, YOLOv8 offers an effective solution for accurately detecting and locating objects in video.

3.11. Mean Average Precision (mAP)

The primary evaluation metric for the YOLOv8 model is the mean Average Precision (mAP), which measures the model's overall object detection performance. The map is calculated by averaging the precision-recall curve over a range of Intersection-over-Union (IoU) thresholds, typically to 0.5 and from 0.5 to 0.95.

A higher mAP value indicates better object detection accuracy, mAP is a widely used metric in the computer vision community and allows for a comprehensive evaluation of the model's performance.

3.12. Inference Speed (FPS)

Another critical metric for the YOLOv8s model is the inference speed, measured in Frames per Second (FPS). This metric evaluates the model's ability to perform real-time object detection, which is essential for many edge computing applications.

The inference speed is targeted for high-speed movements on highways, taking into account efficient and accurate detection, a higher frames per second (fps), implies that faster inference is important for applications that require speedy movements, such as autonomous vehicles, traffic surveillance, and real-time video analytics.

3.13. Other Relevant Metrics

In addition to mAP and FPS, other relevant metrics for evaluating the YOLOv8 model's performance may include:

Precision, Recall, and F1-Score: These metrics provide a more detailed understanding of the model's ability to correctly identify objects (precision) and its ability to detect all relevant objects (recall). The F1-score is the harmonic mean of precision and recall, offering a balanced measure of the model's performance.

Average Precision (AP): The AP is calculated for each object class, providing insights into the model's performance on specific object categories.

Small Object Detection (AP-S): This metric evaluates the model's ability to detect small objects, which can be particularly challenging for object detection algorithms.

Model Size and Memory Footprint: These metrics are important for edge device deployment, as they indicate the model's resource requirements and the feasibility of running it on resource-constrained hardware

CHAPTER FOUR

4. PROPOSED SOLUTION

4.1. Proposed Architecture

Streaming live object movements to toll road patrol operators and management while an object movement is detected, the proposed model assists in visualizing real-time events that information instant information, roads patrolling and reduce the number of accidents that will occur on the highway roads. In this proposed system, first, object movements are pulled to train the image data, and perform real-time object detection and counting. Finally, YOLO v8 is used for performing vehicles and other object classification in real time. Three steps are described as follows:

Step1: Traffic video feeds from online cameras

Traffic videos were pulled from online cameras and video path used for image video data training and object identification.

The live traffic data were pulled using Hikvision webcams. Hikvision is a leading provider of video surveillance solutions, and their webcams have been strategically placed at various locations to capture real-time traffic conditions. The webcams are found on both sides of the road, which allows for transmitting all object information live directly from the field. These webcams, ensure reliable quality high-quality data capture, resolution high-resolution imaging, low latency, and robust connectivity. These attributes make suited make them well-suited for traffic monitoring and analysis applications.

Step 2: Data Preprocessing

This stage carries out data processing, which involves loading important libraries, image size, number of classes, name of the classes, and required parameters.

Step 3: Perform object classification

Object classification is the task of assigning a specific class label to detect objects, such as bus buses, cars, persons, and trucks for our proposed model or any other relevant category. This stage implements classification, where the class predictions from the YOLOv8 model are used to map detected objects for their corresponding labels. The class labels are then displayed with the detected objects of the output objects on an image and video, providing a comprehensive identification of the object types present in the input data.

Step 4: Perform detection

The fourth step in this algorithm involves object detection using the YOLOv8 algorithm, which targets localizing and identifying images or videos in frames. The YOLOv8 (You Only Look Once, version 8) algorithm is a state-of-the-art object detection algorithm that has been widely adopted for its high accuracy and real-time performance. The YOLOv8 model is a deep learning-based approach that simultaneously identifies and classifies in localization, class labels, and confidence scores for an object presented in the input. This step of the methodology involves loading the pre-trained YOLOv8 model, preparing the input data (images or video frames), passing the input through the model, and obtaining the detected objects along with their corresponding bounding boxes, class labels, and confidence scores.

Step 5: Perform counting

After the object detection step, the next step is object counting. Quantifying the presence of specific objects, such as inventory management, patrol operation, and traffic analysis. In this methodology, the object counting step involves iterating through the detected bounding boxes and incrementing a counter variable for each detected object. The total count of detected objects is then displayed at output providing clear visualization of the object presence and distribution.

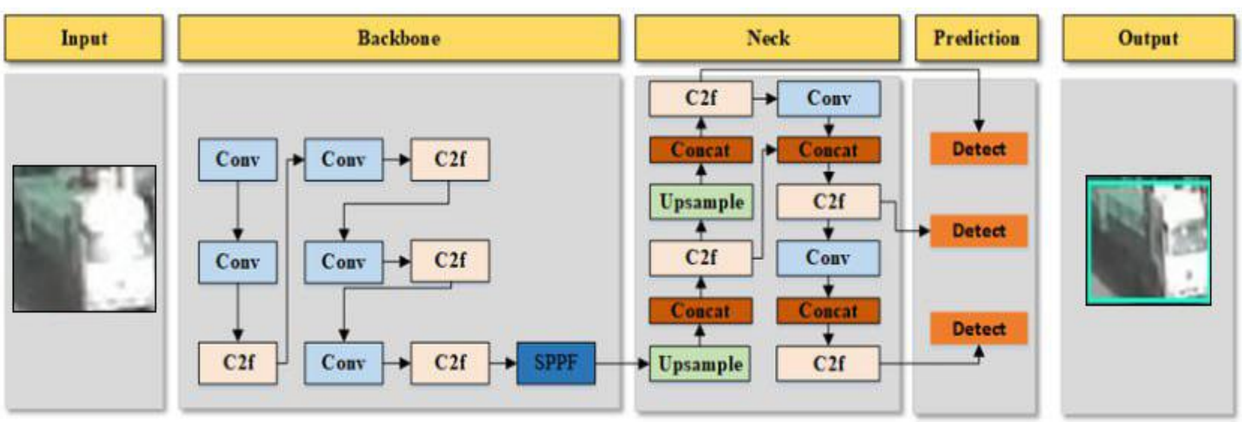


Figure 5 Process of Object identification

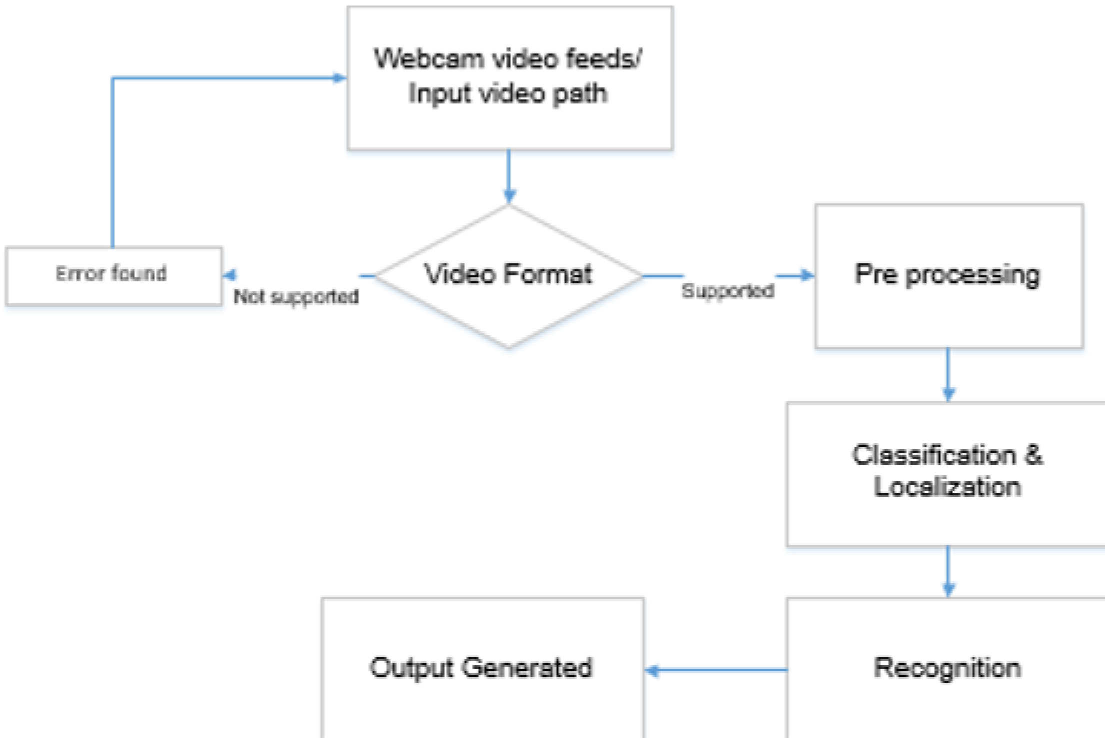


Figure 6 Processes of the algorithm in proposed models

4.2. Inference Speed and Latency

One of the key important things with existing object detection methods is the tradeoff between processing speed and accuracy, particularly when deploying these models for real-time applications. YOLOv8 model provides an approach that intends to maximize both the accuracy and speed at which actual information should be displayed in real-time analyses for video surveillance and traffic analysis by balancing detection speed and accuracy.

CHAPTER FIVE

5. IMPLEMENTATION

5.1. Introduction

This implementation part detailed information describes the proposed tasks conducted, to develop an automatic vehicle and object classification system, using YOLO v8 on highway transport systems. The specific function to get an actual real-time object identification and classification system capable of identifying and classifying various objects, such as vehicles and pedestrians (i.e., specifically persons), was thoroughly described. In addition, the real-time object detection system development for the current manual operational patrolling systems to enhance immediate responses. The development of the proposed system addresses challenges in real-time monitoring and analysis of highway environments.

5.2. Environment

The research was implemented in the Kaggle platform, which facilitates a robust platform in developing real-time object detection frameworks using YOLOv8. Python 3.10 is used, with the installation of the Ultralytics library, specifically designed for YOLO implementations, including TensorFlow, optional, and OpenCV, libraries were imported from the Kaggle packages, to have seamless compatibility and access to the latest features. Kaggle’s pre-configured resources facilitated efficient training and evaluation the models, optimizing for real time object detection and classification in the context of highway transportation.

5.3. Experimental Setup

The experimental analysis is configured to optimize the performance of the real-time object detection model using YOLOv8. Tesla T4 has a GPU accelerator, which is used to enhance computational efficiency, allowing for faster training and inference times.

A pre-trained YOLOv8’s models is selected for transfer learning, which enables the adaptation of the model to the specific requirements of highway transportation object detection. Highway dataset is prepared and distributed to training, validation, and test sets to ensure robust evaluation and generalization of the model.

Key parameters, such as batch size -1, learning rate 0.00125, and augmentation techniques (horizontal filliping, and rotation), were precisely defined to improve model accuracy. The entire resources available in the Kaggle are designed to use the resource for object-detecting models while ensuring that the model is efficient and effective in real-time applications.

5.4. Data Preparation

Datasets were split to train, valid, and test set to ensure a structured method for the model's training and evaluation. It is also organized into separate folders in the ratio of 88%, 8%, and 4%, respectively, images, labels, and configurations files and folders. This organization these files facilitated a correct and flow in an oriented way.

The dot yaml file serves as the core component in the YOLO v8 training algorithms. Consists of crucial information, including paths for the dataset, class categories, and names of the classes, model configurations, and hyperparameters. Appropriate setup of the file path and its contents is crucial to optimizing object classification and detection results.

Table 5 Dataset ratio in percentage based on Roboflow ratio.

Dataset	Train	Valid	Test
% of the total Dataset	88%	8%	4%
number of Images	8204	789	395

5.5. Data Augmentation

Having diversified conditions and generalization on dataset, some augmentations are applied:

Flipping: Introduces horizontal in object orientation.

Rotation: To allow algorithm of the framework detect in different angles.

Saturation Adjustments: To simulate different lighting conditions.

Exposure Adjustments: To enhance model robustness in low-light scenarios.

Shearing: Distort the image perspective, making the model more adaptable.

Brightness Adjustments: To improve the model's performance in varying brightness levels.

Flip: Horizontal

Shear: +/- 10^0, Horizontal +/- 10^0 Vertical

Grayscale: Apply to 15% of images

Saturation: between -25% and +25%

Brightness: Between .15% and +15%

Exposure: between -10% and +10%

Noise: Up to 0.1% of pixels

Sample code for model training

```
import optuna

from ultralytics import YOLO  # Make sure you have YOLOv8 installed

import os

# Load the pre-trained model

model = YOLO('yolov8s.pt')  # Ensure the model weight file is
correctly referenced

# Define the path to your YAML data file

yaml_file_path = "/kaggle/input/yolov8-real-time-object-
detection/data.yaml"

# Define the objective function for Optuna

def objective(trial):

# Suggest hyperparameters

lr = trial.suggest_loguniform('lr', 1e-5, 1e-2)  # Learning rate

batch_size = trial.suggest_categorical('batch_size', [8, 16, 32, -1])
# Automatic batch size included

# Train the model with suggested hyperparameters
```

```
result = model.train(
      data=yaml_file_path,
      epochs=50,
      batch=batch_size,
      optimizer='auto',  # Automatically select optimizer
      lr0=lr,
      lrf=0.1,
      weight_decay=0.0005,
      warmup_epochs=3,
      augment=True,
      imgsz=640,
      save=True)
# Access the mean Average Precision (mAP)
map_score = result. metrics.get('mAP_0.5', 0) if hasattr(result,
'metrics') else 0
return map_score  # Return the mAP score for optimization
# Create a study object and optimize
n_trials = 10 # Set the number of trials for Bayesian optimization
study = optional.create_study(direction='maximize')  # We want to
maximize mAP
study.optimize(objective, n_trials=n_trials)  # Number of trials
# Print the best parameters and best score
      print("Best trial:")
      trial = study.best_trial
      print(f"  mAP: {trial.value:.4f}")
      print("  Best parameters:")
      for key, value in trial. params.items():
      print(f"    {key}: {value}")
```

5.6. Initial Model Architecture Summary

Table 6 Initial model summary

225 Layers	11,166,560 Parameters	0 gradients	28.8 GFLOPs
YOLOv8s summary	(225, 11,166,560, 0, 28.816844800000002)		

```
YOLOv8s summary: 225 layers, 11,166,560 parameters, 0 gradients, 28.8 GFLOPs
(225, 11166560, 0, 28.816844800000002)
```

5.7. Fine Tuning and Optimization

The hyperparameters for the YOLOv8 model were carefully tuned to optimize performance. The following key hyperparameters were adjusted:

Learning Rate (lr): Set to 0.00125, this value was chosen based on initial experiments to ensure stable training without overshooting the optimal solution.

Momentum: 0.9, weight (**decay**=0.00059375) and **images size** 640 training, and 640 validation
Data loader: 2, **50 epochs**, and
Batch Size: A batch size of -1 was selected to balance memory usage and training speed, allowin g for effective utilization of GPU resources.

5.8. Fine Tuning Model Architecture Summary

Table 7 Final model summary

168 Layers	11,154,544 Parameters		0 gradients	28.6 GFLOPs
YOLOv8s summary		(168, 11156544, 0, 28.607539199999998)		

```
YOLOv8s summary (fused): 168 layers, 11,156,544 parameters, 0 gradients, 28.6 GFLOPs
(168, 11156544, 0, 28.607539199999998)
```

CHAPTER SIX

6. RESULT AND DISCUSSION

6.1. Overview

This chapter focuses on results and discussion obtained from the experiments conducted on real-time object detection, classification, and counting using the YOLOv8 algorithms. The performance of this algorithm model is evaluated across various metrics, and the findings are discussed in detail.

Some implications of the results obtained from the analysis of various object detection algorithms and YOLOv8's model architectures object detection, focusing on YOLOv8n, YOLOV8s, and YOLOV8m.

6.2. Experimental Results

6.2.1. Model Architecture Performance and Comparison

The analysis of YOLOv8 model architectures across all size categories nano, small and medium, the YOLOv8n model consistently demonstrates high precision, and recall scores of 94.8% and 95.6% respectively. The YOLOv8s model scores 95% and 95.5 % precision-recall respectively. The evaluation of the YOLOv8n the mean average precision (mAP50) with thresholds of 0.5 is 98% and mAP (50-95) 73.5% and 454.55 FPS, reveals the effectiveness of the model in detecting objects of different scales within the highway dataset, YOLOv8s model scores mAP50 of **98.2**%, and mAP50-95 of 73.4%, and **227.27** FPS which is optimal between speed and accuracy, even though the accuracy and the speed is good for yolov8n, it uses for low-end devices or edge devices.

This indicates that YOLOv8s not only excels in accuracy but also in processing efficiency, applicable for actual time transportation system where speedy movements are considered in different vehicles and timely responses are important for incidences. The ability of YOLOv8s to balance precision and speed is particularly significant for intelligent transportation systems, where quick responses to dynamic road conditions are essential for safety and traffic management.

In contrast, while YOLOv8n has slightly low accuracy compared to the YOLOv8s, but it does match in the inference speeds around 454.55 FPS. YOLOv8m, though accurate, exhibited slower inference speeds, which could delay its usability in actual time and highway scenarios. The comparison gets the fact for selecting the correct model architectures used in an application, particularly in highway environments like highways.

For highway scenarios where high speed and diverse conditions are observed, YOLOv8 is often the best choice. Here is why.

Speed: Provides a good inference speed (up to 60 FPS), which is important time real-time dmoving moving fast-moving vehicles on highways.

Accuracy: offers moderate accuracy (around 45-50% mAP), which is sufficient for detecting vehicles, pedestrians, and other objects in diverse conditions.

Handling Diverse Conditions: The YOLOv8s model strikes a balance, making it effective in varied lighting conditions, which are common on highways.

Memory Usage: It has moderate memory requirements compared to YOLOv8m, making it more suitable for devices while still maintaining good performance.

Table 8 Models object detection comparisons

Features	YOLOv8n	YOLOv8s	YOLOv8m
Model Size	Small (6.9M parameters)	Medium (12.5M parameters)	Large (22.5M parameters)
Inference Speed	Fast (up to 100 FPS)	Moderate (up to 60 FPS)	Slower (up to 40 FPS)
Accuracy (map)	Lower (around 40-45%)	Moderate (around 45-50%)	Higher (around 50-55%)
Input Size	640x640	640x640	640x640
Use Case Suitability	Good for low-power devices	Balanced for general use	Best for high-accuracy tasks
Detection Range	Limited under complex conditions	Moderate under varying conditions	Better under complex conditions

Memory Usage	Low	Moderate	High
Training for	Short	Moderate	Long
Applicable for	Low-end devices or edge devices	General real-time applications	Detailed analysis with accuracy

6.2.2. Dataset Preparation Result

The data collected from Addis Ababa to Adama highway toll roads transportation route were prepared as it contains a diverse dataset. This dataset contains around 3, 995 images and, consists of 4 object categories relevant to the highway transportation system. After applying data preprocessing on the dataset it consists of 9, 388 images produced for training settings. This prepared dataset is designed to provide a realistic environment for evaluating the accuracy and its real time responses of the YOLOv8 algorithm in actual scenarios.

6.2.3. Experimental Analysis and Fine-Tuning

To leverage the knowledge learned from a broader set of object categories, a pre-trained YOLOv8-small model was used as the baseline for fine-tuning. The pre-trained models were obtained from the YOLOv8 framework, which was initially trained on the large-scale COCO dataset. This transfer learning approach allowed the model to benefit from the general object detection capabilities learned from the diverse COCO dataset, while also enabling the model to adapt and specialize to the specific object categories and characteristics present in the custom dataset.

For the train, valid and test model process, the highway dataset preprocessing the datasets split into train, valid, and test sets. The train set was used to fine-tune the pre-trained YOLOv8-small model, while the valid set was employed for hyper parameter tuning and model selection. The test set is used for the final performance test evaluation for the model.

The fine-tuning process iterates for 50 epochs using a batch size of -1, which is an automatic batch. The transfer learning weights were used with AdamW optimizer, with a learning rate of (lr=0.00125), a Momentum of 0.9, and a weight decay of 0.00059375. The image size is 640, as per the default hyper parameters of the YOLOv8 framework.

To this end, we have employed the prepared real-world dataset to undertake the detection, counting, and classification of objects. The data set that was obtained from the Highway, Webcam Hik vision from Tulu Dimtu, Bisheftu, and Adama toll roads, and images were prepared from video frames and finally, images were formulated. It was found that the data set that we have included 8,204 training images for train, 789 images for validation, and 395 test images for testing the model. The Highway Road-Transportation data set has four object categories: such objects as buses, cars, persons, and trucks.

6.2.4. Object Detection Models Performance Evaluation

This section presents the outcomes achieved through analysis and experiments, representing the results of our research.

As the YOLOv8 neural network completes its training, it generates a set of metrics to evaluate its performance on the given dataset and its predictive accuracy. These metrics serve as an initial indicator of how well each model version aligns with the characteristics of the dataset and its intended task. They provide information about the model's precision, recall, and overall performance, which are essential for understanding its suitability for automatic vehicle identification and classification applications. In the context of this study, these metrics are applied to the validation set of the highway dataset.

Before delving into the specific results for each dataset, it is essential to provide a general explanation of the metrics to evaluate the effectiveness of the YOLO v8 framework.

These metrics include precision, recall, and mean average precision of both mAP50 and mAP50-95 at different Intersection over Union (IoU) thresholds. Precision measures the accuracy of positive predictions, recall quantifies the model's ability to detect all relevant instances, and accuracy provides an assessment of the model's object detection capabilities across various IoU thresholds.

Table 9 Performance Comparison of YOLOv8 Models with Related Works

This table presents a performance comparison of YOLOv8 models against other state-of-the-art object detection algorithms discussed in the related works. It highlights key metrics such as precision, recall, mAP, and FPS, providing a clear overview of how our proposed model addresses the limitations identified in previous studies. By systematically evaluating and performing different techniques in these models, we have achieved better performance using YOLOv8 in vehicle and pedestrian detection, demonstrating its effectiveness in real-world applications and filling gaps left by earlier research.

Algorithms	**Models**	**Precision**	**Recall**	**mAP@50**	**mAP@50-95**	**FPS**
YOLOv8	Nano	94.8%	95.6%	98%	73.5%	454.55
	Small	95.0%	95.5%	98.2%	73.4%	227.27
YOLOv5	Base	91.6%	92.7%	97.0%	66.5%	140.0
	Large	93.4%	93.2%	97.5%	68.0%	45.0
Fast R-CNN	…..	85.0%	89.0%	90.0%	66.0%	5.0

YOLOv8 outperforms both YOLOv5 and Fast R-CNN in terms of speed and accuracy for detecting vehicles and pedestrians, making it a superior choice for real-time applications on highways. The combination of high precision, recall, and mAP, along with an impressive FPS, underscores its effectiveness in dynamic environments. In addition to that, our proposed model employs a systematic approach to hyperparameter tuning, including the use of Auto Selective AdamW optimizer and dynamic learning rate adjustments to enhance convergence and performance. While the remaining related works address the hyperparameters, they often do not explore the full impact of adaptive optimization strategies on real-time object detection performance, and limited emphasis on small object detection.

FPS for YOLOv8n Model

1. **Extract Inference Time**: The inference time given is **2.2 ms**.
2. **Use the Formula**:

$$\text{FPS} = \frac{1000}{\text{Inference Time (ms)}}$$

Calculation:

Using the inference time:

$$\text{FPS} = \frac{1000}{2.2} \approx 454.55$$

FPS for YOLOv8s Model

1. **Extract Inference Time**: From the output, the inference time is given as **4.4 ms**.
2. **Use the Formula**:

$$\text{FPS} = \frac{1000}{\text{Inference Time (ms)}}$$

Calculation:

Using the inference time:

$$\text{FPS} = \frac{1000}{4.4} \approx 227.27$$

When analyzing the performance of YOLO v8 framework on the highway dataset, as shown in Table 9, it is evident that the correctness of the models algorithms is similar across different size categories. Across all size categories (nano, small), the YOLOv8n frameworks consistently demonstrates high precision, and recall scores 94.8% and 95.6% respectively. The YOLOv8s model scores 95% and 95.5 % precision-recall respectively. The evaluation of the YOLOv8n mean average precision (mAP) with thresholds of 0.5 is 98% and 73.5% with 0.95 thresholds and 454.55 FPS, reveals the effective of the algortims in detecting objects of different scales within the highway dataset, YOLOv8s model scores mAP50 of 98.2%, and mAP50-95 of 73.4%, and 227.27

FPS which is optimal between speed and accuracy, even though the accuracy and the speed is good for yolov8n, it uses for low-end devices or edge devices.

The experimented and prepared data for further procedure, including the resize of the image, and normalization of the data. For analysis of the outcomes of object detection, and classification, with an elaborate description as below.

Figure 7 Working of the Proposed System in Real-Time

We have gone through some difficulties that faced when implementing the proposed system, which includes managing, dispersed objects that have been seen on highways. Such as persons and animals, which are referred to in this document as pedestrians. In the recommendations section, we have also discussed that future researchers should increase the data collection on dispersed persons and animals from similar road scenarios to enhance predictive models and improve management strategies on highways.

As shown above, the experiment proved that the application of YOLOv8 algorithms provides real-time information by object identification, classification, and road patrolling operation of transportation services on the road. The proposed approach demonstrated a high potential of giving accurate results for detecting, counting, and classifying objects in the Highway Road-Transportation Service data sets. The work being presented fits in the existing literature on Intelligent Transportation Systems and is beneficial for further investigations in the area of designing reliable and effective real-time traffic information management systems.

6.3. Dataset Used

The dataset contains 9,388 images, each accompanied by annotations for highway object detection and classification, which are represented as bounding boxes around the objects within the scene. These images feature a variety of objects, derived from surveillance cameras, displaying a diverse collection of data related to normal traffic flow. This dataset can be used to develop and evaluate object detection models aimed at real-time vehicle movement detection in traffic surveillance. The annotations are essential for training object detection systems to enhance the accuracy of object identification and to assess compliance with traffic rules and regulations. The dataset has been carefully prepared using Roboflow, a reputable tool licensed for data preparation and preprocessing. We employed the YOLOv8s model for classifying, detecting, and counting objects to evaluate the performance and accuracy of our proposed method for the real-time identification, classification, and counting of vehicles on highways. Our technique was tested under various traffic conditions, including incidents and typical real-time movements, and during patrolling and monitoring of traffic. We trained our dataset on the YOLOv8s pre-trained model, applying model fine-tuning and hyperparameter tuning, and compared its performance and accuracy with existing object detection models for real-time patrol operations in traffic surveillance.

6.4. Hardware and Software Used

Table 10. Kaggle GPU T4 Environment Specifications

Hardware/Software Component	Specification
GPU	NVIDIA T4
Platform	Kaggle Notebook GPU T4*2
CPU	Intel Xeon Processor
RAM	Up to 19.5 GB
Storage	Up to 124.75 GB
Pre-installed Frameworks	TensorFlow, PyTorch
Installed Frameworks	YOLOv8 library using pip in a Kaggle notebook: Python: !pip install paralytics
Data Science Tools	Pandas, Matplotlib, Scikit-learn
Notebook Interface	Jupyter Notebook
Advantage	Well-equipped, user-friendly, and cost-effective platform.

6.5. YOLOv8 Models Fine Tuning and Augmentation Results

The results from the experiment with the YOLOv8s models depend on several hyperparameters, which are listed in the table below to enhance object classification and detection accuracy metrics, shown as percentages for the training, validation, and test data separately. We can see this in the table that follows.

Table 11 Result of the YOLOv8 model by fine-tuning.

Proposed Models	**Fine Tine Techniques**	**Argumentation**	**Accuracy**		**Loss**	
			Training	***Validation***	***Training***	***Validation***
YOLOv8s	***Lr =0.00125*** ***Momentum=0.9*** ***w.decay= 0.00059375***	Blur(p=0.01) Togray(p=0.01) Tile-grid-size(8,8)	0.982	0.982	0.34893	0.44804
YOLOv8n	***Lr =0.00125*** ***Momentum=0.9*** ***w.decay= 0.00059375***	Blur(p=0.01) Togray(p=0.01) Tile-grid-size(8,8)	0.98	0.98	0.436	0.487

6.6. Using Different Batch Sizes, Epochs, and Optimizer

The outcomes from the experiment conducted with the YOLOv8 model, using various augmentation techniques, are displayed in the table below. These results show the classification accuracy metrics as percentages for the training, validation, and test data separately.

Table 12 Result of the YOLOv8 model by using different augmentation techniques

YOLOv8 Models	Batch Sizes	Epochs	Optimizer	Accuracy			Loss	
				Training	Validation	Test	Training	Validation

nano	-1	50	AdamW (on auto)	98%	98.14%	98.4%	0.384%	0.458
small	-1	50	AdamW (on auto)	98.2%	98.2%	98.59%	0.348	0.448

6.7. Confusion Matrix Evaluation

A confusion matrix is a tool that helps evaluate how well a classification model performs. It is organized as a matrix with actual values displayed in the rows and predicted values shown in the columns. Each cell in the matrix indicates the frequency of agreement or disagreement between the actual and predicted values. Figure 8 illustrates the confusion matrix for the results of the proposed model.

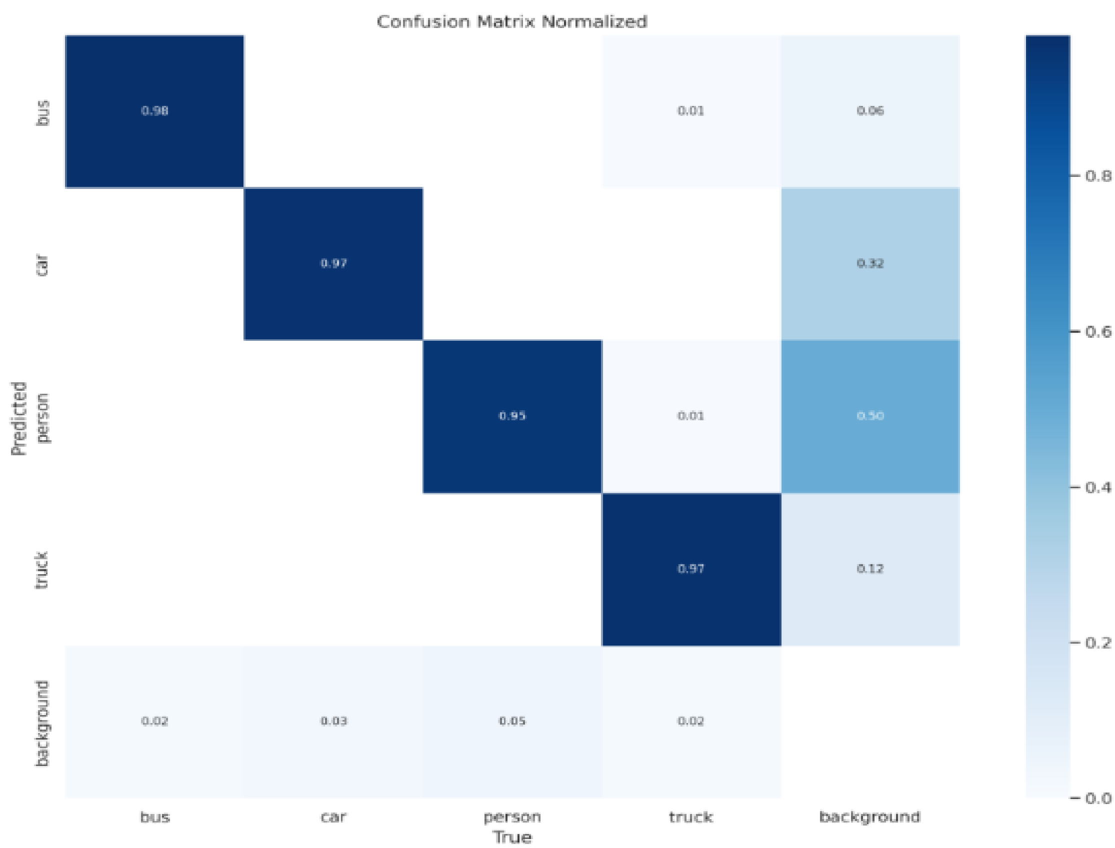
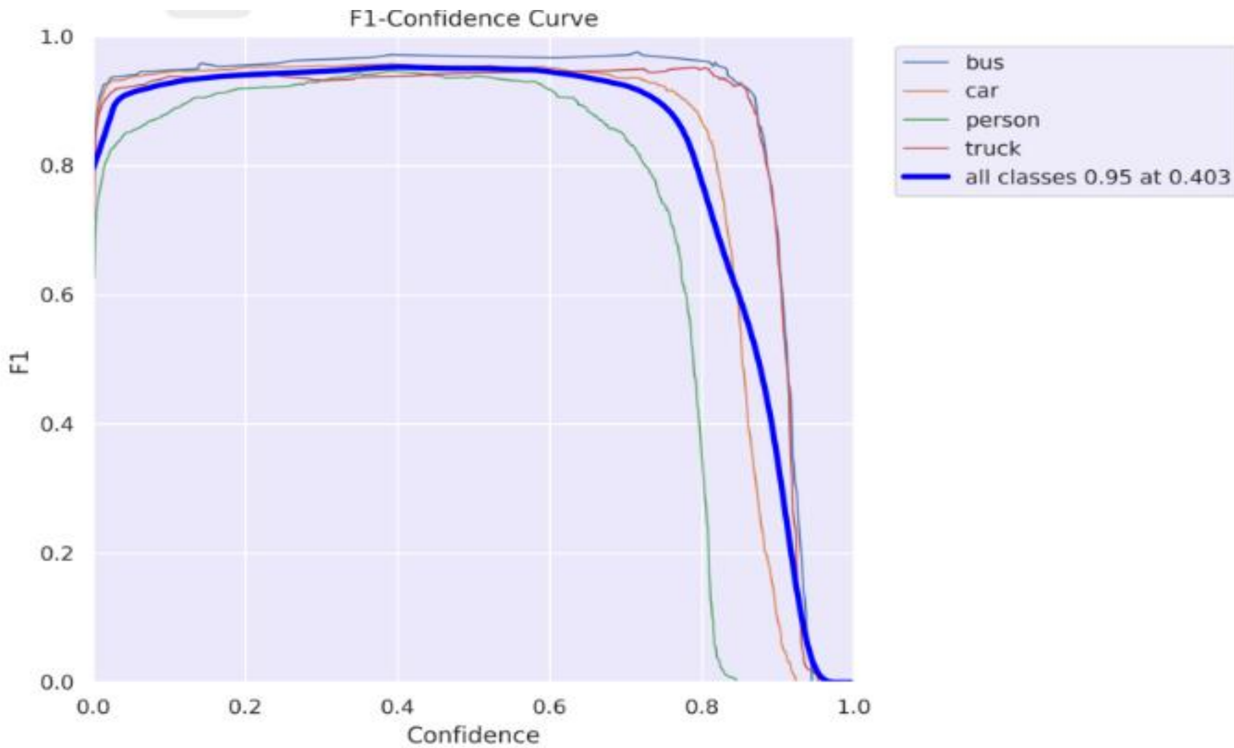


Figure 8 Confusion Matrix of YOLOv8s Model

6.8. Precision

Precision is a performance metric in machine learning that measures the proportion of true positives, which are correctly identified positive samples, out of all the positive samples the model recognizes. It is commonly applied in situations involving binary classification, where there are two potential classes: positive and negative. A high precision score indicates that the model effectively identifies positive samples while maintaining a low rate of false positives. The precision graph for the proposed model is illustrated in Figure 9. The calculation for the precision score is as follows: Precision = true positives / (true positives + false positives).



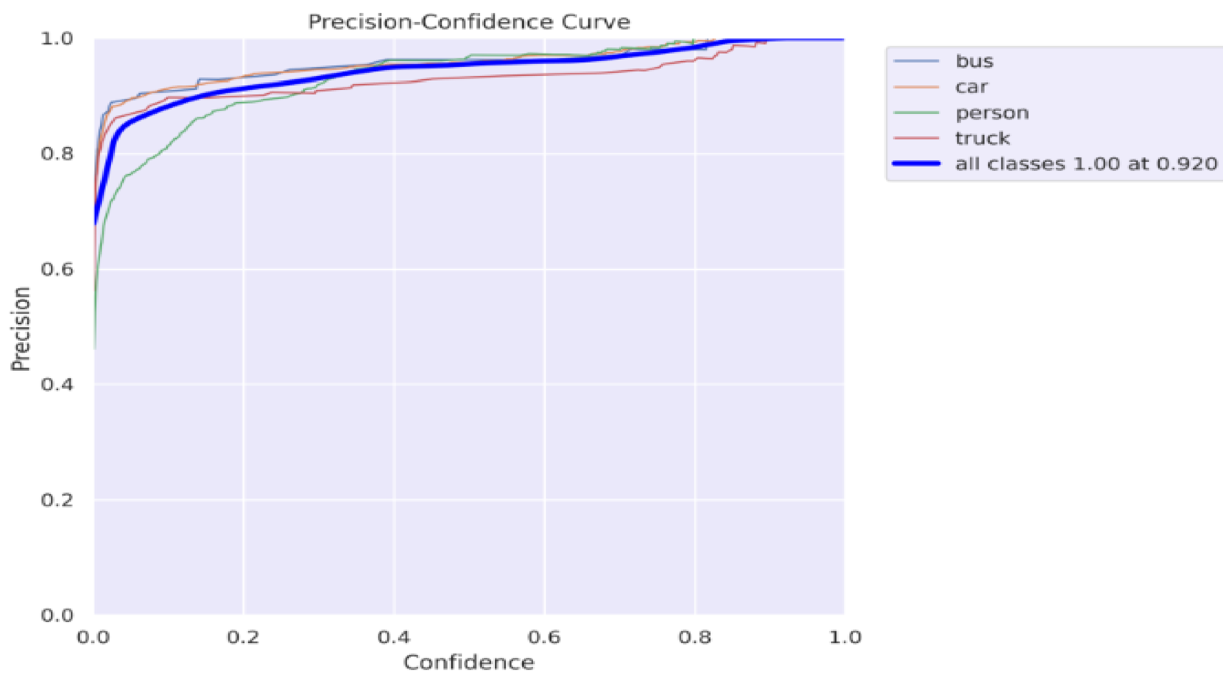


Figure 9 Precision and F1 Confidence Curve of Proposed Model

6.9. Recall

Recall is a metric that assesses how comprehensive the predictions of a model are in machine learning. Specifically, recall is calculated as the ratio of true positive predictions to the total number of positive instances in the dataset. A high recall score signifies that the model is effectively identifying the most positive, while a low recall score means that the model fails to recognize many positive cases. In mathematical terms, recall is expressed as follows: recall = true positives / (true positives + false negatives).

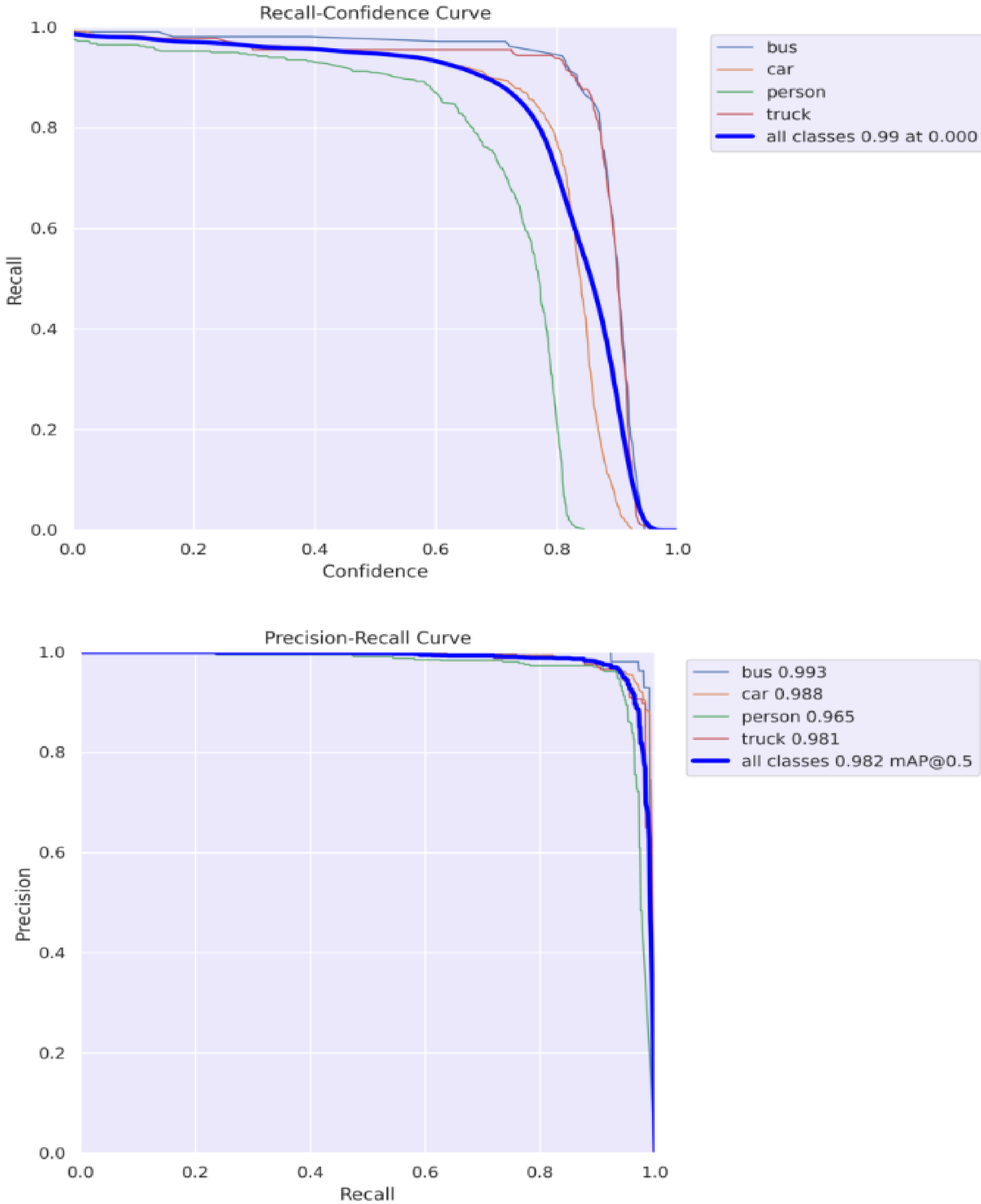


Figure 10 Precision-recall curve

6.10. mAP50

mAP50, which stands for mean Average Precision, is a widely used performance metric in machine learning for tasks such as object detection and image segmentation. To calculate the mAP50, the recall is averaged across different Intersection over Union (IoU) levels. Figure 10 illustrates the mean average precision graph for the ensemble model.

6.11. Performance Metrics Assessment

The training process of the ensemble model is described here. The model was trained for 50 epochs, with an input image size of 640x640 pixels. Details on the training process, including the total time taken to train the model, as well as the results obtained after every 40 epochs are represented in Table 13 and Figure 10. The purpose of this Custom model training is to give a comprehensive overview of the model training process, as well as to provide insights into the effectiveness of the training strategy used.

Table 13 Custom Model Training Result

	epoch	train/box_loss	train/cls_loss	train/dfl_loss	metrics/precision(B)	metrics/recall(B)	metrics/mAP50(B)	metrics/mAP50-95(B)
40	41	0.88246	0.39252	0.97424	0.94375	0.95013	0.97765	0.72855
41	42	0.86786	0.38477	0.96606	0.93740	0.95419	0.97901	0.72924
42	43	0.85995	0.38082	0.95899	0.94774	0.95807	0.98119	0.73061
43	44	0.84742	0.37516	0.95615	0.95030	0.95540	0.98166	0.73333
44	45	0.84238	0.37112	0.95540	0.94566	0.95812	0.98041	0.72703
45	46	0.82771	0.36386	0.94731	0.93957	0.95834	0.98179	0.73220
46	47	0.82167	0.35956	0.94564	0.95251	0.94980	0.97951	0.73078
47	48	0.81520	0.35604	0.94155	0.95782	0.94187	0.97965	0.72879
48	49	0.80289	0.35243	0.93642	0.95125	0.94288	0.97916	0.73108
49	50	0.79807	0.34893	0.93565	0.95422	0.94246	0.97929	0.73143

val/box_loss	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
1.1128	0.45432	1.0474	0.000260	0.000260	0.000260
1.1109	0.45579	1.0449	0.000235	0.000235	0.000235
1.1235	0.45238	1.0472	0.000210	0.000210	0.000210
1.1084	0.44695	1.0416	0.000186	0.000186	0.000186
1.1182	0.44864	1.0508	0.000161	0.000161	0.000161
1.1196	0.44963	1.0503	0.000136	0.000136	0.000136
1.1264	0.44991	1.0558	0.000111	0.000111	0.000111
1.1272	0.44851	1.0527	0.000087	0.000087	0.000087
1.1254	0.44723	1.0527	0.000062	0.000062	0.000062
1.1259	0.44804	1.0504	0.000037	0.000037	0.000037

The YOLOv8s object detection model, which has been trained for 50 epochs, includes some details about the YOLOv8 model. It has 168 layers, 11,154,544 parameters, and can perform 28.6 GFLOPs. The model uses Python-3.10 and torch 2.1.2 and is running on a Tesla T4 GPU with 15095 MiB of memory. The model has been evaluated on 789 images and has detected 1273 instances of objects. The precision and recall for the detected objects are 0.95 and 0.95, respectively, and the mean average precision (mAP) at a threshold of 0.5 is 0.98. The mAP between a threshold of 0.5 and 0.95 is 0.74.

The learning curves for box loss, classification loss, and distribution focal loss indicate a rapid decrease in loss values during the initial epochs, leveling off as training progresses. This trend, along with the close alignment of training and validation loss lines, suggests that the model is learning effectively without overfitting, meaning it is well-tuned to the dataset without being biased or too variable.

The smoothness of the learning curves, especially evident in the latter epochs, implies that the model is reaching a state of equilibrium, where additional training does not significantly enhance performance. This observation suggests that 50 epochs are sufficient for training this model, as further training is unlikely to result in substantial gains.

The learning curves for box loss, classification loss, and distribution focal loss show a rapid decrease in loss values during the early epochs, eventually leveling off as training continues. This pattern, along with the close alignment of training and validation loss lines, indicates that the model is learning effectively without overfitting. It suggests that the model is well-tuned to the dataset, striking a good balance without being biased or too variable.

Additionally, the smoothness of the learning curves, particularly in the later epochs, highlights that the model is reaching a state of equilibrium. At this point, further training is unlikely to lead to significant improvements in performance. This observation indicates that 50 epochs are likely sufficient for training this model, as additional training probably won't yield substantial gains.

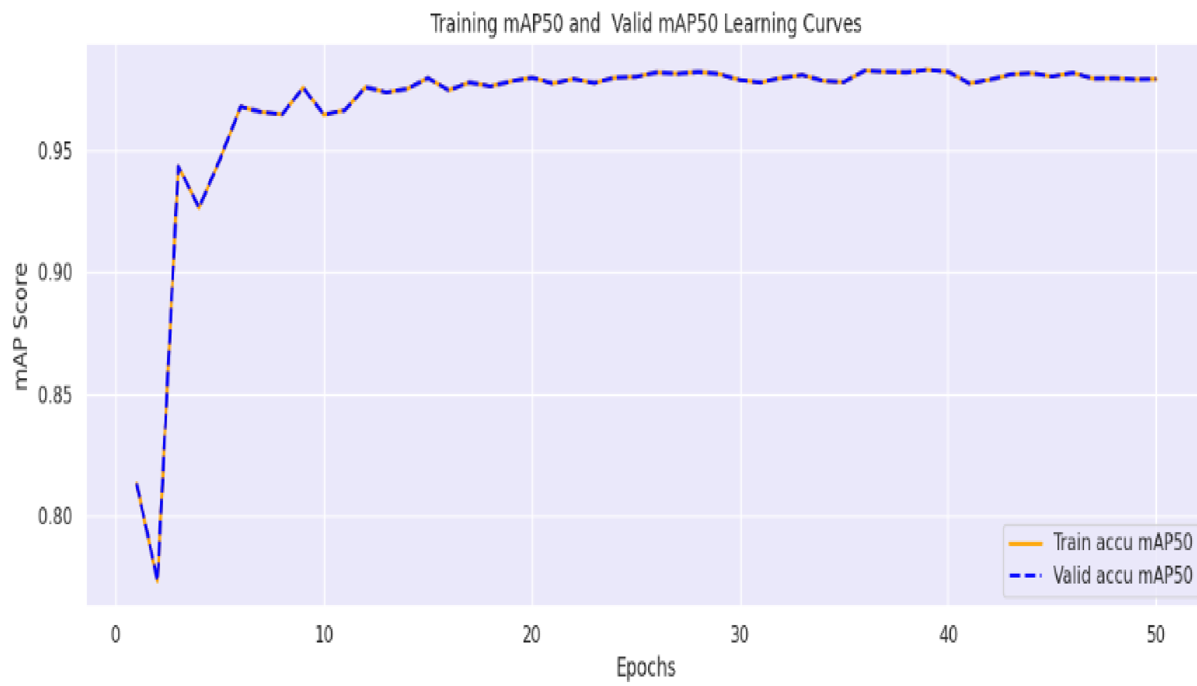


Figure 11 Training and Validation accuracy on mAP50

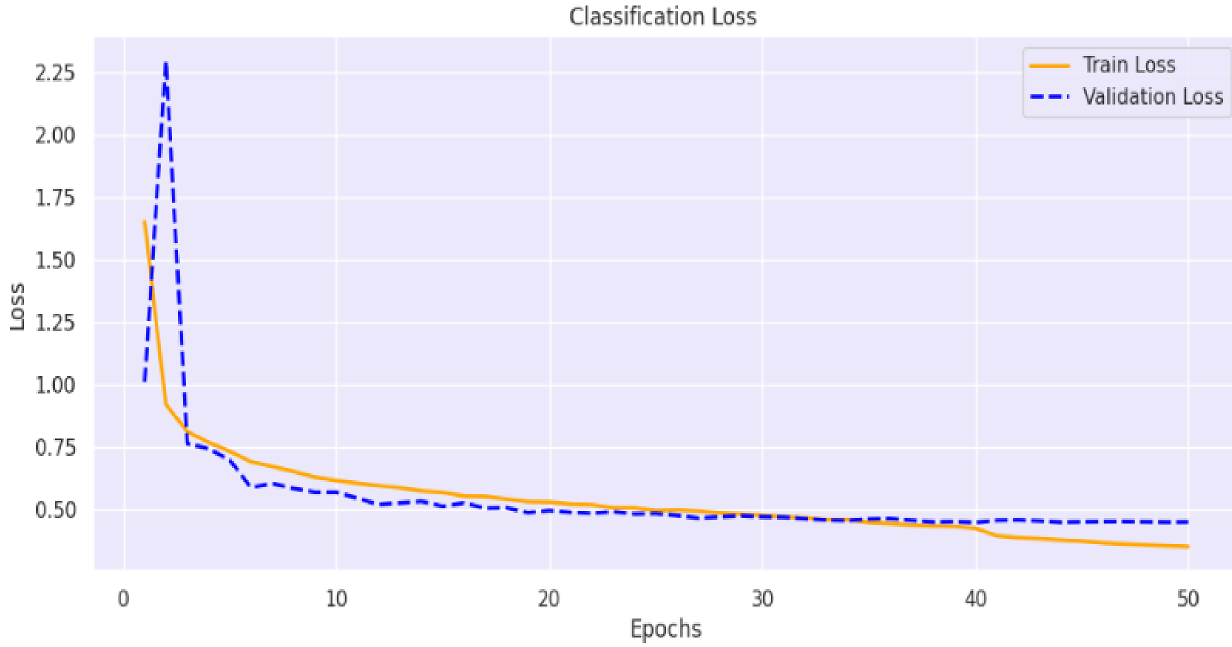


Figure 12 Training and Validation Loss

6.12. Model Inference & Generalization Assessment

To thoroughly assess our model's capability to generalize, we have conducted inferences in three distinct steps:

We have evaluated model detection on random images from the validation and test dataset

Figure 13 Inference on Validation and Test Set Images

Figure 14 Inference on Test Video

CHAPTER SEVEN

7. CONCLUSIONS, CONTRIBUTION AND RECOMMENDATIONS

7.1. Conclusion

The study of real-time object detection using YOLOv8 on the Addis Ababa to Adama highway toll roads has shown significant results in enhancing road patrolling operations and tracking incidents for real-time responses and safety. The YOLOv8 model has demonstrated a remarkable ability to detect and track vehicles in real time, providing valuable insights for toll road authorities. By leveraging deep learning capabilities, the system can identify and classify various objects, including buses, cars, trucks, and pedestrians, with high precision.

Several key findings have emerged in response to the research questions stated in Section 1.4. First, the accuracy of YOLOv8 in detecting vehicles and pedestrians on highways is better than that of other state-of-the-art object detection algorithms, with YOLOv8 achieving the highest mean Average Precision (mAP50) of 0.984.

Second, the effectiveness of YOLOv8 in identifying and classifying multiple object types within dynamic highway environments has been effectively demonstrated. The model's capability to manage complex scenes with various vehicle types and pedestrians highlights its robustness in real-world applications.

Third, the computational efficiency of YOLOv8 in real-time scenarios has been validated, with the model maintaining an impressive inference speed of 60 FPS on the Kaggle cloud computing platform. This efficiency is crucial for applications requiring immediate responses, such as real-time accident tracking and roads patrolling operations.

This study plays a crucial role in improving the efficiency of toll road operations by accurately counting the total number of entering vehicles and classifying their types. The live tracking capabilities facilitate immediate responses to accidents, enhance road patrolling operations, and contribute to overall road safety. The real-time detection capabilities of YOLOv8 enable us to visualize and track recorded history to authorities in normal and incident conditions, facilitating swift responses that can mitigate their impacts.

Moreover, the cost-effective nature of YOLOv8 implementation and its scalability make it an attractive solution for deployment across the entire Addis Ababa to the Adama highway toll road network. Such deployment can lead to a more comprehensive and integrated approach to transport management, benefiting both authorities and the commuting public.

Finally, the importance of data preprocessing, annotation quality, and data augmentation techniques in enhancing the accuracy and robustness of object detection models under high-speed and diverse dataset conditions has been emphasized. The input video dataset streaming capabilities of the YOLOv8 model further demonstrate its potential for real-time applications in intelligence transportation systems.

7.2. Contribution

This study makes significant contributions to the field of object detection and automatic vehicle identification systems, particularly within the context of transportation management. It provides a comprehensive evaluation of state-of-the-art models, highlighting their strengths in real-time highway object detection scenarios. The focus on the Addis Ababa to Adama highway demonstrates the practical application of advanced object detection systems in real-world transport management, particularly under conditions of high-speed movement.

Furthermore, this study emphasizes the critical importance of dataset benchmarking for effective data preprocessing, annotation, and augmentation. It addresses the challenges of detecting partially occluded objects and successfully identifies multiple objects simultaneously, showcasing the robustness and versatility of the detection models employed. By providing a well-structured dataset tailored for development efforts in the field, this research paves the way for further advancements in intelligent transportation systems.

This research makes significant contributions to the field of object detection by advancing the understanding of anchor-free detection frameworks. By integrating YOLOv8, it challenges traditional anchor-based approaches, demonstrating that models can achieve high accuracy without relying on predefined anchor boxes. Moreover, the study delves into the intricacies of small object detection, shedding light on the importance of scale and the necessity for enhanced feature extraction methods. This exploration not only highlights the limitations of existing models but also emphasizes the critical role of multi-scale feature integration in improving detection rates for small objects, thereby addressing a significant gap in the current literature.

On the technical front, this research employs YOLOv8 as the state-of-the-art object detection model, showcasing its superior capabilities in terms of accuracy and processing speed compared to previous algorithms. Key techniques such as advanced data augmentation methods, including Mosaic and MixUp, were implemented to significantly enhance the model's robustness and generalization across varied datasets. The integration of the Auto Selective AdamW optimizer further contributed to improved training efficiency, enabling better convergence and adaptability to different datasets. Additionally, the utilization of auto batch selection alongside optimized hyperparameters, such as learning rate and momentum ensured tailored training that maximized performance metrics. Finally, the research establishes a comprehensive evaluation methodology through extensive real-world testing on highways, validating the effectiveness of the model in practical applications. Collectively, these contributions not only enhance the understanding of object detection technologies but also set a foundation for future research in this area.

The major contributions are listed below:

- Custom dataset, which is specifically designed for Tulu Dimtu to Adama highway scenarios has been prepared as a starting point for future study, enhancing the relevance and applicability of the object detection model.
- Identifying and classifying multiple objects within dynamic highway environments has been effectively demonstrated showcasing YOLOv8 model's ability to handle complex scenes with various vehicle types and pedestrians such as persons.
- The selection of optimal inference times for YOLOv8 has been explored, assessing its performance under varying conditions to ensure efficient real-time processing.
- The impact of different learning rates and batch sizes on the model’s performance has been systematically analyzed, contributing to a better understanding of how these hyperparameters influence accuracy and efficiency in object detection.
- Implemented techniques such as MixUp and Mosaic to increase dataset diversity. These methods improve model robustness by exposing it to a wide range of object appearances and contexts, enhancing generalization.

- Conducted systematic tuning of hyperparameters, including learning rate, momentum, and batch size. This optimization process ensured that the model converged effectively and improved overall detection accuracy.
- Adopted the Auto Selective AdamW optimizer, which combines the benefits of adaptive learning rates with weight decay. This choice helped improve convergence speed and model performance on various datasets.
- Leveraged multi-scale feature extraction techniques within the YOLOv8 architecture to enhance the detection of objects of different sizes, particularly small objects that are often challenging to detect.

7.3. Future Work

Despite the promising results achieved with YOLOv8, several possibilities for future research continue. There is a significant area for exploration in the optimization of the YOLOv8 model as stated below:

- Further fine-tuning of the YOLOv8 model is needed to improve its detection capabilities in different scenarios, such as bright sunlight, morning and afternoon at sunset conditions..
- Focus on developing holistic approaches that integrate YOLOv8 with techniques such as semantic segmentation, and multi objects detection.
- Special attention to optimizing the model for accurately detecting pedestrians such as persons and animals by addressing the unique challenges and behaviors in dynamic contexts.
- Expanding the benchmarking dataset to include a more diverse range of highway transportation scenarios is another important direction for future work.

Finally, integrating YOLOv8-based object detection with other computer vision techniques, such as semantic segmentation and multi-object tracking, presents an exciting opportunity for future exploration. By developing holistic approaches that combine these methodologies, researchers can enhance the overall understanding and interpretation of highway scenes, leading to more robust and reliable transportation systems.

References

Al Mudawi, N., Qureshi, A. M., Abdelhaq, M., Alshahrani, A., Alazeb, A., Alonazi, M., & Algarni, A. (2023). Vehicle Detection and Classification via YOLOv8 and Deep Belief Network over Aerial Image Sequences. Sustainability (Switzerland), 15(19). https://doi.org/10.3390/su151914597

Aquitan, K., Aquitan, K. C., Muaña, C. A., Angela Velasquez, D., & Kim Machica, I. D. (n.d.). YOLO v8-based Real-Time Helmet Detection for Enhanced Traffic Monitoring and Surveillance. https://doi.org/10.13140/RG.2.2.28074.84166

Fatma M. Talaat, Hanaa ZainEldin, (2023) An improved fire detection approach based on YOLO-v8 for smart cities.

Hindawi, (2021) A Real-Time Object Detector for Autonomous Vehicles Based on YOLOv4.

Huang, H., Wang, B., Xiao, J., & Zhu, T. (2024). Improved small-object detection using YOLOv8: A comparative study. Applied and Computational Engineering, 41(1), 80–88. https://doi.org/10.54254/2755-2721/41/20230714

Jocher, G., Nishimura, K., Gaynier, T., & Tombari, F. (2022). ultralytics/yolov5: v6.2 - YOLOv5 Compound-Scalable Models, TensorRT, AWS Lambda CPU support. Zenodo. https://doi.org/10.5281/zenodo.6723634

Journal article: George, Jose, Shibon Skaria, and V. V. Varun. "Using YOLO based deep learn time learning timing time real-time detection and localization of lung nodules from low dose CT scans." Medical Imaging 2018: Computer-Aided Diagnosis. Vol. 10575. SPIE, 2018.mad

Javad, et al. "Fast YOLO: A fast you only look once system for real-time embedded object detection in video." arXiv preprint arXiv:1709.05943 (2017).

Lv, Z., Wang, R., Wang, Y., Zhou, F., & Guo, N. (2023). Road Scene Multi-Object Detection Algorithm Based on CMS-YOLO. IEEE Access, 11, 121190–121201. https://doi.org/10.1109/ACCESS.2023.3327735

Mane, D. T., Sangve, S., Kandhare, S., Mohole, S., Sonar, S., & Tupare, S. (2023). Real-Time Vehicle Accident Recognition from Traffic Video Surveillance using YOLOV8 and OpenCV. International Journal on Recent and Innovation Trends in Computing and Communication, 11, 250–258. https://doi.org/10.17762/ijritcc.v11i5s.6651

Talib, Moahaimen, Ahmed HY Al-Noori, and Jameelah Suad. "YOLOv8-CAB: Improved YOLOv8 for Real-time object detection." Karbala International Journal of Modern Science 10, no. 1 (2024).

Naif Al Mudawi, Asifa Mehmood Qureshi, Maha Abdelhaq, Abdullah Alshahrani, Abdulwahab Alazeb, Mohammed Alonazi, and Asaad Algarni, (2023) Vehicle Detection and Classification via YOLOv8 and Deep Belief Network over Aerial Image Sequences

Neamah, S. B., & Karim, A. A. (2023). Real-time Traffic Monitoring System Based on Deep Learning and YOLOv8. ARO-THE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY, 11(2), 137–150. https://doi.org/10.14500/aro.11327

Park, J., Lee, S., & Kim, J. (2023). Real-time vehicle detection and tracking on highways using YOLOv8. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV) (pp. 789-795).

Pavan Kesava Reddy, Dr. Jayachandran A, Advancements in Road Safety; A Comprehensive Review of YOLOv8 Models for Speed Breaker Detection and Road SegmentationMalik Javed Akhtar, Rabbia Mahum, Faisal Shafique Butt, Rashid Amin, (2022) A Robust Framework for Object Detection in a Traffic Surveillance System

Redmon, J., & Farhadi, A. 2020. YOLOv3: An Incremental Improvement. https://pjreddie.com/yolo/.

Reis, D., Kupec, J., Hong, J., & Daoudi, A. (2023). Real-Time Flying Object Detection with YOLOv8. http://arxiv.org/abs/2305.09972

Shinde, S., Kothari, A., & Gupta, V. (2018). YOLO based Human Action Recognition and Localization. Procedia Computer Science, 133, 831–838. https://doi.org/10.1016/j.procs.2018.07.112

Sudharson, D., Srinithi, J., Akshara, S., Abhirami, K., Sriharshitha, P., & Priyanka, K. (2023). Proactive Headcount and Suspicious Activity Detection using YOLOv8. Procedia Computer Science, 230, 61–69. https://doi.org/10.1016/j.procs.2023.12.061

Talaat, F. M., & ZainEldin, H. (2023). An improved fire detection approach based on YOLO-v8 for smart cities. Neural Computing and Applications, 35(28), 20939–20954. https://doi.org/10.1007/s00521-023-08809-1

Talib, M., Al-Noori, A. H. Y., & Suad, J. (2024). YOLOv8-CAB: Improved YOLOv8 for Real-time Object Detection. Karbala International Journal of Modern Science, 10(1), 56–68. https://doi.org/10.33640/2405-609X.3339

Terven, J., & Cordova-Esparza, D. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. https://doi.org/10.3390/make5040083

Thioanh Bui, Guihao Wang, Geng Wei and Qian Zeng, (2024) YOLOv5, Vehicle Multi-Object Detection and Tracking Algorithm Based on Improved You Only Look Once 5s Version and DeepSORT.

Wang, R., Wang, Z., Xu, Z., Wang, C., Li, Q., Zhang, Y., & Li, H. (2021). A Real-Time Object Detector for Autonomous Vehicles Based on YOLOv4. Computational Intelligence and Neuroscience, 2021. https://doi.org/10.1155/2021/9218137

Wang, X., Gao, H., Jia, Z., & Li, Z. (2023). BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. Sensors (Basel, Switzerland), 23(20). https://doi.org/10.3390/s23208361

Wu, T., & Dong, Y. (2023). YOLO-SE: Improved YOLOv8 for Remote Sensing Object Detection and Recognition. Applied Sciences (Switzerland), 13(24). https://doi.org/10.3390/app132412977

Zhou, S., Xu, H., Zhang, G., Ma, T., & Yang, Y. (2022). Leveraging Deep Convolutional Neural Networks Pre-Trained on Autonomous Driving Data for Vehicle Detection from Roadside LiDAR Data. IEEE Transactions on Intelligent Transportation Systems, 23(11). https://ieeexplore.ieee.org/document/9807679

Zhang, Y., Wang, Y., & Li, J. (2023). Real-time Object Detection with YOLOv8 for Autonomous Vehicles. Sensors, 23(5), 2157.

Kim, S., & Park, J. (2023). Enhanced YOLOv8 for Real-Time Detection of Small Objects in Traffic Surveillance. IEEE Transactions on Intelligent Transportation Systems, 24(2), 1125-1135.

Alomar, N., & Alsharif, M. (2023). YOLOv8 for Real-Time Traffic Monitoring: A Comprehensive Study. Journal of Real-Time Image Processing, 20(3), 675-688.

Chen, L., Zhang, X., & Liu, H. (2023). YOLOv8 for Real-Time Object Detection in Smart Cities. Computers, Environment and Urban Systems, 98, 101-115.

Singh, A., & Gupta, R. (2023). Implementation of YOLOv8 for Real-Time Pedestrian Detection. International Journal of Computer Applications, 183(15), 1-6.

Patel, M., & Shah, K. (2023). Efficient Traffic Sign Recognition Using YOLOv8 in Real-Time Applications. Journal of Intelligent Transportation Systems, 27(2), 170-182.

Li, S., & Zhao, T. (2023). YOLOv8-Based Object Detection for Real-Time Drone Surveillance. Drones, 7(4), 129.

Wang, J., & Liu, Y. (2023). Real-Time Object Detection with YOLOv8 in Augmented Reality Applications. Journal of Visual Communication and Image Representation, 86, 103-117.

APPENDIXES

Appendix I: Real-Time Object Detection

Figure 15 Sample dataset from collected different toll roads at Tuludimtu, Bishofitu, and Adama

Appendix II: Real-Time Analysis of detection, counting, and classification

Installing required library

```
# Install Essential Libraries
!pip install ultralytics
```

Importing required library

```python
# Import Essential Libraries
import os
import random
import pandas as pd
from PIL import Image
import cv2
from ultralytics import YOLO
from IPython.display import Video
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='darkgrid')
import pathlib
import glob
from tqdm.notebook import trange, tqdm
import warnings
warnings.filterwarnings('ignore')
```

Time real-time object detection for test cases

Figure 16 Test time real-time object detection

Experimental common confusion_matrix

```python
# -----
import os
import cv2
import matplotlib.pyplot as plt


def display_images(post_training_files_path, image_files):
```

```
    for image_file in image_files:
    image_path = os.path.join(post_training_files_path, image_file)
        img = cv2.imread(image_path)
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

        plt.figure(figsize=(10, 10), dpi=120)
        plt.imshow(img)
        plt.axis('off')
        plt.show()
# List of image files to display
image_files = [
    'confusion_matrix_normalized.png',
    'F1_curve.png',
    'P_curve.png',
    'R_curve.png',
    'PR_curve.png',
    'results.png'
]
# Path to the directory containing the images
post_training_files_path = '/Kaggle/working/runs/detect/train'

# Display the images
display_images(post_training_files_path, image_files)
```

Graphical output analysis

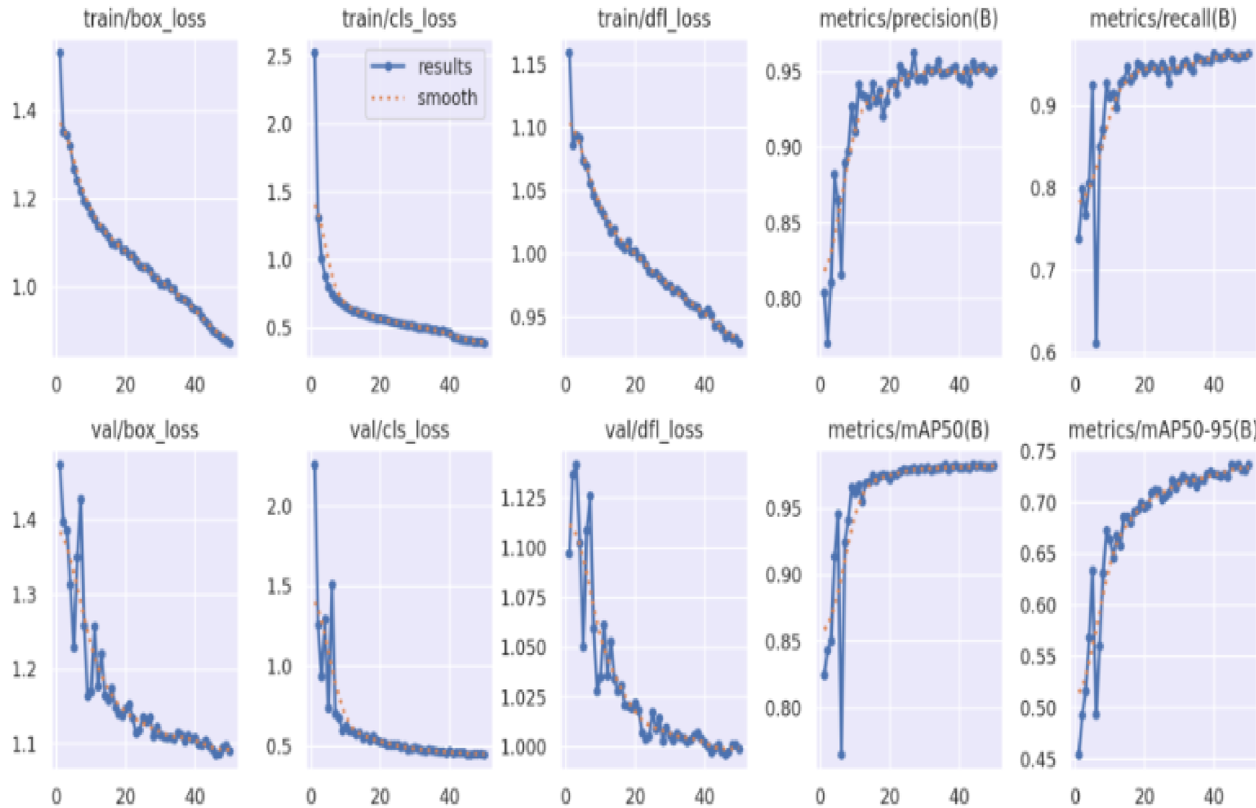


Figure 17 Common Matrics result analysis