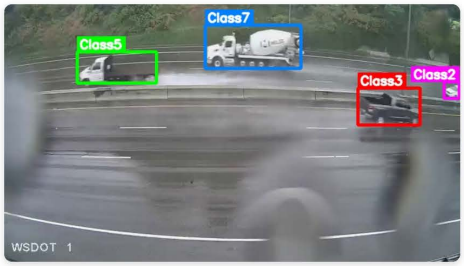
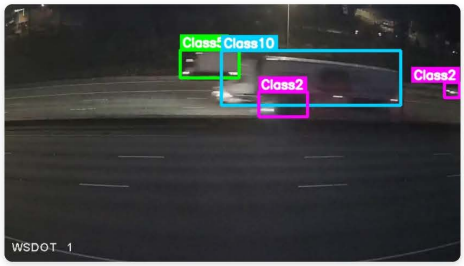
Cost-Effective Traffic and Roadway Data Collection Using an Edge-Based Comprehensive Sensing System

WA-RD 961.1

Yinhai Wang, PhD, P.E.
Chenxi Liu, PhD
Yan Shi
Kehua Chen

December 2025





Washington State Department of Transportation
Office of Research & Library Services

WSDOT Research Report

Research Report
Agreement T1461, Task AI
WSDOT Vehicle Classification

Cost-Effective Traffic and Roadway Data Collection Using an Edge-Based Comprehensive Sensing System

by

Yinhai Wang, PhD, P.E.
University of Washington

Yan Shi, Research Assistant
University of Washington

Chenxi Liu, PhD
University of Washington

Kehua Chen, Postdoctoral Researcher
University of Washington

Department of Civil and Environmental Engineering
University of Washington, Box 352700
Seattle, Washington 98195

Washington State Transportation Center (TRAC)
University of Washington, Box 359446
University Tower, 4333 Brooklyn Ave NE
Seattle, Washington 98195-9446

Washington State Department of Transportation
Technical Monitors
Natarajan Janarthanan, Ph.D.

Prepared for

The State of Washington
Department of Transportation
Julie Meredith, Secretary

December 2025

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
WA-RD 961.1		

4. Title and Subtitle	5. Report Date
Cost-Effective Traffic and Roadway Data Collection Using an Edge-Based Comprehensive Sensing System	December 2025
	6. Performing Organization Code
7. Author(s)	**8. Performing Organization Report No.**
Yinhai Wang, Yan Shi, Chenxi Liu, and Kehua Chen	
9. Performing Organization Name and Address	**10. Work Unit No. (TRAIS)**
Washington State Transportation Center University of Washington, Box 359446 University Tower, 4333 Brooklyn Ave NE Seattle, Washington 98195-9446	**11. Contract or Grant No.** Contract T1461-AI
12. Sponsoring Organization Name and Address	**13. Type of Report and Period Covered**
Washington State Department of Transportation Transportation Building, MS 47372 Olympia, Washington 98504-7372 14 Shervin Jahangirnejad, Research Coordinator, Shervin.Jahangirnejad@wsdot.wa.gov	Research report **14. Sponsoring Agency Code**

15. Supplementary Notes

16. Abstract

This project developed and evaluated a cost-effective, camera-based system for FHWA 13-bin vehicle classification using roadside video, edge computing, and deep learning. Traditional vehicle classification methods, such as inductive loops, pneumatic tubes, and manual counts, are often expensive to deploy and maintain and provide limited classification detail, particularly under challenging traffic, weather, and lighting conditions. To address these limitations, a portable roadside sensing platform was designed that integrates video cameras with edge computing hardware for on-site, near–real-time vehicle detection and classification. The system operates effectively in daytime, nighttime, and rainy conditions while minimizing data transfer requirements. At the core of the platform is an enhanced YOLOv11 deep learning model incorporating a mixture-of-experts (MoE) architecture. The model uses a shared backbone with multiple expert heads tailored to different environmental conditions, supported by a weather encoder and feature-wise linear modulation (FiLM) to adapt to variations in illumination, glare, and wet pavement. Model training and evaluation were conducted using a curated dataset of more than 20,000 vehicle instances annotated into the 13 FHWA vehicle classes and collected at WSDOT-relevant sites under diverse conditions. The training pipeline addressed class imbalance and visibility challenges through data augmentation, nighttime image enhancement, and class-weighted loss functions. The proposed system achieved an overall classification accuracy of up to 97.9 percent, with strong performance for heavy-duty vehicles and rare FHWA classes, even under adverse weather and lighting conditions.

17. Key Words	18. Distribution Statement
Vehicle detection and classification, deep learning; and multi-expert learning	No restrictions.

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified.	Unclassified.	NA	NA

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Washington State Department of Transportation or Federal Highway Administration. This report does not constitute a standard, specification, or regulation.

CONTENTS

Section *Page*

ABBREVIATIONS LIST xi
EXECUTIVE SUMMARY xiii
KEY PRODUCTS xv
1.0 INTRODUCTION 1
1.1 Background 1
1.2 Key Challenges 5
2.0 LITERATURE REVIEW 10
2.1 Overview 10
2.2 Existing Vehicle Classification and Detection Algorithm 11
2.2.1 Pneumatic Tube and Loop Detectors 12
2.2.2 Weight-In-Motion Systems 13
2.2.3 Machine Vision Systems 14
2.2.4 LIDAR Systems 20
2.3 Summary 21
3.0 SENSORS DEPLOYMENT AND DATA COLLECTION 22
3.1 System and Data Overview 22
3.2 Sensors Deployment 23
3.3 Public Datasets 27
3.4 Data Collection Using the MUST Sensor 29
3.5 Dataset Annotation 31
3.6 Dataset Analysis and Imbalance Handling 32
3.7 Summary 35
4.0 LEARNING METHODS FOR TRAFFIC VEHICLE CLASSIFICATION 38
4.1 Overview 38
4.2 YOLOv11 Baseline Architecture 39
4.3 Data Preprocessing and Augmentation 40
4.4 Imbalance Dataset Handling 43
4.5 Training Configuration and Optimization 44
4.6 Evaluation and Validation Strategy 44
4.7 Results and Summary 45
5.0 MODEL DEPLOYMENT 50

5.1 Overview 50
5.2 Deployment Pipeline 51
5.3 Model Quantization 53
5.4 Pruning Optimization 56
5.5 Runtime Integration 58
5.6 Summary 60
6.0 CHALLENGES IN ADVERSE VISIBILITY 62
6.1 Overview 62
6.1.1 Nighttime Low-Light Challenges 62
6.1.2 Rainy Weather Challenges 62
6.1.3 Proposed Solution 63
6.2 Existing Methods 64
6.2.1 Hardware-Oriented Methods 64
6.2.2 Classical Digital Enhancement Techniques 65
6.2.3 Deep Learning-Based Low-Light Enhancement 65
6.2.4 Rain and Weather-Specific Restoration 66
6.2.5 Domian Adaption and Data-Driven Solutions 67
6.3 Results from Existing Enhancement and Adaptation Methods 68
6.3.1 Nighttime Image Enhancement 68
6.3.2 Day-to-Night Data Augmentation 70
6.3.3 Fine-Tuning for Nighttime Domain Adaptation 72
6.4 Summary 75
7.0 MIXTURE-OF-EXPERTS (MOE) FRAME ARCHITECTURE 77
7.1 Overview 77
7.2 Architecture Design of the Mixture-of-Experts Framework 79
7.2.1 Shared Backbone 80
7.2.2 Expert Modules 80
7.2.3 Environment Encoding 82
7.3 Loss Function 83
7.4 Summary 85
8.0 RESULTS AND ABLATION STUDY 87
8.1 Results 87
8.2 Routing Mechanism Ablation 92
8.3 FiLM Modulation Ablation 94

8.4 Loss Function Ablation 95
8.5 Cross-Site Generalization Study 97
9.0 CONCLUSIONS AND FUTURE WORK 98
9.1 Conclusion 98
9.1.1 Development of a High-Quality, Multi-Condition Dataset 98
9.1.2 Image Enhancement and Nighttime Adaptation Pipeline 99
9.1.3 Adaptation to Rainy Weather and Challenging Visibility 99
9.1.4 Unified Mixture-of-Experts (MoE) Framework 99
9.1.5 Extensive Ablation Studies and Model Validation 100
9.1.6 Generalization to New Deployment Sites 100
9.2 Future Work 101
10.0 REFERENCES 102

LIST OF FIGURES

Figure 1. FHWA 13 vehicle category classification [4] ---------- 7

Figure 2. Similar appearance. Right: Class 3 light truck pulling a trailer [4]. Left: Class 8 truck pulling a trailer [4].---------- 8

Figure 3. Pneumatic tube detectors ----------12

Figure 4. Weight-in-motion systems ----------14

Figure 5. Two-stage methods [19]. ----------15

Figure 6. R-CNN algorithm [14]. ----------16

Figure 7. YOLO structure [21] ----------19

Figure 8 Vehicle classification with YOLO ----------20

Figure 9. The location of the MUST sensor. ----------24

Figure 10. The MUST sensor. ----------25

Figure 11. The installed MUST sensor. ----------26

Figure 12 Two viewpoints of MUST sensor. ----------30

Figure 13. Annotation process. ----------32

Figure 14 Data distribution in one-hour classes ----------33

Figure 15. Sampling methods. ----------34

Figure 16. Vehicle classification sampling method. ----------34

Figure 17 Image-level class distribution of the preprocessed dataset. ----------35

Figure 18. Additional images from the Yakama Intersection. ----------37

Figure 19. Offline training process. ----------38

Figure 20. YOLO V11 structure. ----------40

Figure 21. Examples of augmented samples. ----------42

Figure 22. Robust performance in occlusion scenarios. ----------47

Figure 23. Robust performance with multi-unit vehicles. ----------48

Figure 24. Robust performance with different appearance within the same class. --------49

Figure 25. Robust performance across similar appearances in different classes----------49

Figure 26. Final product. ----------51

Figure 27. Deployment pipeline. ----------52

Figure 28. Video processing. ----------60

Figure 29. Comparison of nighttime image enhancement models on real-world traffic footage. ----------70

Figure 30. Example of day-to-night translation using CycleGAN. The nighttime images are generated by translating daytime images using the CycleGAN model. ----------71

Figure 31. Example results of nighttime vehicle detection. ----------74

Figure 32. Overview of the proposed MoE model. ----------78

Figure 33. The architecture of MoE model. ----------79

Figure 34. Visual comparison of input images (top) and extracted feature maps (bottom). ----------91

Figure 35. Qualitative results of our MoE detection model. ----------92

LIST OF TABLES

Table 1. FHWA vehicle classification definitions [4] 6

Table 2. Merged classes. .....**Error! Bookmark not defined.**

Table 3. Dataset comparison. 37

Table 4. Performance of the proposed model. 46

Table 5. Per-class comparison of mAP across different methods. 89

Table 6. MoE architecture ablation. 94

Table 7. Loss function ablation. 96

ABBREVIATIONS LIST

ADAS	Advanced driver-assistance system
ARM	Accumulated route mileage
ASCE	American Society of Civil Engineers
CNN	Convolutional neural network
COCO	Common objects in context
CSV	Comma-separated value
CUDA	Compute Unified Device Architecture
CV	Computer vision
CSP	Cross-stage partial
DT	Decision tree
DCNN `	Deep convolutional neural network
DL	Deep learning
DNN	Deep neural network
DOT	Department of transportation
FHWA	Federal Highway Administration
FPN	Feature Pyramid Network
FPS	Frames per second
FiLM	Feature-wise Linear Modulation
GPS	Global Positioning System
GPU	Graphic processing unit
ONNX	Open Neural Network Exchange
IoU	Intersection over union
ITS	Intelligent transportation systems
LiDAR	Light detection and ranging
MAE	Mean absolute error
mAP	Mean average precision
MUST	Mobile Unit for Sensing Traffic
MoE	Mixture-of-experts
NAS	Neural architecture search
NB	Naive Bayes

PAN	Path Aggregation Network
PTR	Permanent traffic recorders
PTZ	Pan-tilt-zoom
PTQ	Post-training quantization
QAT	Quantization-aware training
R-CNN	Region-based convolutional neural networks
RGB	Red-green-blue
RIAC	Retinex-based Image Adaptive Correction
ROI	Region of interest
RPN	Region proposal network
STAR Lab	Smart Transportation Applications & Research Laboratory
SVM	Support vector machine
WIM	Weigh-in-motion
WSDOT	Washington State Department of Transportation
YOLO	You Only Look Once
YOLOv11	You Only Look Once 11th version

EXECUTIVE SUMMARY

This project developed and demonstrated a cost-effective system for FHWA 13-bin vehicle classification using roadside video, edge computing, and deep learning. The work was motivated by the Washington State Department of Transportation's (WSDOT's) growing need for high-resolution vehicle classification to support planning, freight analysis, pavement and bridge design, and performance monitoring. The work was motivated by WSDOT's growing demand for high-resolution vehicle counting across complex traffic conditions. Traditional systems (e.g., loops, tubes, and manual counts) can be expensive to deploy, difficult to maintain, and often provide limited class details, especially under challenging weather and lighting conditions.

To address these limitations, we designed a camera-based, edge-deployed sensing platform coupled with an enhanced YOLOv11 deep learning model. The sensor package is portable and can be mounted temporarily at roadside sites to record and process traffic in daytime, nighttime, and rainy conditions. Video data are processed directly on an edge processor located in the field, reducing data transfer requirements and enabling near–real-time vehicle classification.

The core of the system is a customized YOLOv11 detector integrated with a mixture-of-experts (MoE) architecture that adapts to different environmental conditions. The model uses a shared backbone and multiple expert heads tuned for specific weather–lighting scenarios (e.g., clear daytime, nighttime, rain). A weather encoder and feature-wise linear modulation (FiLM) layer enables the model to leverage contextual information—such as brightness, glare, or wet pavement—to improve robustness. This design supports accurate classification of both common and rare FHWA 13-bin classes, including multi-trailer trucks and other heavy-duty vehicles.

Model development relied on a curated dataset of more than 20,000 vehicle instances annotated into the 13 FHWA classes. Data were collected using the portable sensor platform at real-world WSDOT-relevant sites under diverse conditions, including daytime, nighttime, and rain. To address class imbalance and visibility challenges, the training pipeline incorporated data augmentation, nighttime image enhancement, and class-weighted loss functions.

The resulting MoE-based detector achieved an overall classification accuracy of up to 97.9 percent, with particularly strong performance for heavy truck classes and rare FHWA bins, even under adverse weather and lighting conditions. This level of performance suggested that the system can provide high-quality traffic classification data suitable for planning, designing, and monitoring applications.

The full solution was deployed and tested on edge hardware for real-time inference, demonstrating its potential as a practical alternative to manual counts and traditional hardware-based systems. In addition, the project delivered a scalable data pipeline, interactive visualization dashboard, and a fully annotated 13-bin dataset, supporting future deployments and technology transfer to WSDOT and other transportation agencies.

Overall, this research provided a flexible, scalable, and cost-effective solution for automated FHWA 13-bin vehicle classification in real-world traffic environments. This will enable more detailed and reliable data to support data-driven transportation planning, infrastructure management, and safety analysis for state and regional agencies.

KEY PRODUCTS

The research produced multiple practical products, including the following:

- **Portable data collection platform**

  A field-deployable camera and edge-computing system for recording, synchronizing, and processing vehicle video data under daytime, nighttime, and rainy conditions.

- **Edge-based 13-bin classification software**

  A deep-learning–based software solution, built on an enhanced YOLOv11 + MoE architecture, for automated detection and FHWA 13-bin vehicle classification on edge hardware.

- **Curated multi-condition dataset**

  A fully annotated dataset across 13 FHWA classes, captured in diverse traffic, weather, and lighting conditions.

- **Training and deployment pipeline**

  A reproducible model training and deployment pipeline, including custom loss functions, data augmentation strategies, nighttime enhancement methods, and support for edge inference.

1.0 INTRODUCTION

1.1 BACKGROUND

The condition of roadway infrastructure in the United States has raised significant national concern in recent years. According to the 2021 American Society of Civil Engineers (ASCE) Infrastructure Report Card, 43 percent of public roadways are rated in poor or mediocre condition [1]. The report also gave the nation's overall infrastructure a grade of C-, indicating that many assets are deteriorating faster than they can be repaired. The road network faces a substantial backlog of maintenance and rehabilitation needs, and continued deterioration threatens safety, mobility, and economic competitiveness.

The Federal Highway Administration (FHWA) has noted that traffic volumes have increased steadily over the past decades, placing additional stress on pavements, bridges, and highway systems [2]. At the same time, transportation agencies nationwide are faced with budget constraints that limit their ability to maintain and upgrade their networks. Making effective, equitable, and timely investment decisions requires high-quality traffic data—including data on detailed vehicle classifications, traffic volumes, roadway conditions, and operational characteristics.

However, traffic data quality and coverage remain uneven across the United States, particularly in rural areas and locations without permanent sensors. For example, FHWA's Traffic Monitoring Guide highlights that many states rely on a small number of permanent traffic recorders (PTRs) to collect continuous, high-accuracy vehicle classification data, while most locations rely on short-term counts that often lack detailed classification accuracy [3]. PTRs are expensive to install and maintain, and they require reliable power, communication infrastructure,

and safe maintenance access. As a result, agencies typically deploy them sparingly, leaving large portions of the roadway network without complete traffic information.

Washington state is no exception. The Washington State Department of Transportation (WSDOT) manages an extensive highway system that includes urban freeways, suburban arterials, and a large network of rural highways. Yet, like many states, WSDOT's traffic sensor coverage is limited to a fraction of the roadway network, with rural and remote areas having the least sensor availability. This creates "data black holes"—locations where reliable volume, classification, and roadway-condition data are missing. Such gaps make it difficult to support planning, maintenance prioritization, safety studies, freight analysis, and compliance with federal reporting requirements.

Additionally, many existing sensors—including inductive loops, microwave radar sensors, and video image processors—are not capable of producing FHWA 13-category vehicle classification, which is required for applications such as pavement design, freight movement analysis, weight impact studies, and long-range planning [3]. While PTRs can provide this information, the number of these sensors statewide is limited, and expanding PTR deployments is difficult because of cost, maintenance needs, and operational constraints.

In locations without PTRs or permanent sensors, WSDOT must rely on manual short-term traffic counts to collect vehicle classification data. These field activities can be hazardous, especially along high-speed rural highways or at locations with narrow shoulders, challenging geometry, or high truck volumes. In addition to safety risks, manual counts provide only a brief snapshot of traffic conditions rather than continuous data, which limits their usefulness for long-term trend analysis and decision-making.

In parallel, the field of machine learning and computer vision has advanced rapidly, creating new opportunities for automated traffic data collection. Deep-learning–based object detection and classification models are now capable of identifying vehicles from video data with high accuracy. Edge computing hardware has also matured, enabling real-time processing directly at the roadside without the need for high-bandwidth communication. These advances offer promising potential for portable, low-cost, and scalable traffic sensing solutions. However, several key barriers remain.

One major challenge is the lack of publicly available, high-quality FHWA 13-bin vehicle classification datasets. Unlike general vehicle detection datasets—such as COCO, KITTI, or Waymo—there are no public datasets that follow the FHWA 13-class structure or that contain multi-trailer trucks, specialized heavy-duty vehicles, and the full range of combinations necessary for engineering applications. This absence of data makes it difficult to train, benchmark, and validate machine-learning models for the detailed classification required by transportation agencies.

Additionally, roadway traffic exhibits a long-tail distribution of vehicle types. Common categories, such as two-axle passenger cars, appear extremely frequently, whereas rare categories, such as multi-axle and multi-trailer trucks, appear infrequently even over long observation periods. This imbalance presents a significant challenge for machine learning: models tend to overfit to dominant classes and struggle to accurately recognize rare but critical truck types. These heavy vehicle categories, although less frequent, have the greatest impact on pavement deterioration, freight modeling, and structural design. Addressing this long-tail imbalance requires customized algorithms, specialized data augmentation, and carefully designed training strategies.

Given these challenges, WSDOT needs a cost-effective, flexible, and highly accurate traffic sensing solution capable of extending data coverage into locations where traditional sensors are impractical. The system must operate reliably across a variety of environmental conditions—daytime, nighttime, and rain—while running efficiently on edge hardware. It should reduce the need for manual fieldwork, support more consistent statewide data collection, and integrate smoothly with existing WSDOT equipment and workflows.

The primary goal of this research was to develop a robust, portable, and edge-based traffic-sensing system dedicated to FHWA 13-bin vehicle classification. The solution was designed to address WSDOT's need for accurate, scalable, and cost-effective vehicle class data by combining modern machine-learning algorithms with an efficient edge hardware platform. The system focuses exclusively on automated 13-bin classification and is engineered to be easy to deploy, maintain, and operate across diverse roadway environments. The project included the development of an enhanced deep-learning classification model, an optimized deployment pipeline for edge devices, and a portable data collection unit capable of operating in locations with limited communication infrastructure.

To achieve this goal, the research objectives included the following aspects:

Propose an ML-based approach for realizing cost-effective traffic and roadway data collection. ML technology has the capability to effectively learn and extract knowledge from large amounts of data. Specifically, deep-learning-based object detection, localization, classification, tracking, and counting would be utilized in this project because of its outstanding capability to learn deep representative features with high accuracy and efficiency.

Incorporate the algorithm in an edge-based comprehensive sensing system MUST. This means the whole data process would be finished on edge devices without the need to

transfer data back to a server. One existing problem here would be deploying the model efficiently on edge devices while considering computational restrictions. Also, the system should function well in locations with poor communication infrastructure or safety concerns due to geometry and volume constraints.

Collect and categorize important traffic information accurately and automatically. This information would include classified vehicle volumes (FHWA 13-bin classification), road surface conditions, visibility, and other relevant data. It is important to note that this system would focus only on the data collection of vehicles, not pedestrians or cyclists.

To sum up, the expected outcomes of this project were a traffic data sensing and processing system deployed in a mobile unit. This system should be capable of performing 13-category vehicle classification and helping researchers collect information where traditional methods cannot be used for safety or functional limitations. The collected traffic data would support data-driven decision-making and more advanced technology development in related fields.

1.2 KEY CHALLENGES

The FHWA 13-bin vehicle classification system, developed in the 1980s, remains the national standard for federal reporting and state-level traffic monitoring. It categorizes vehicles on the basis of axle count, axle arrangement, and vehicle configuration. Table 1 summarizes the official FHWA definitions. Figure 1 shows the vehicle category classification.

Table 1. FHWA vehicle classification definitions [4]

Class Group	Class Definition	Class Includes	Number of Axles
1	Motorcycles	Motorcycles	2
2	Passenger Cars	All cars Cars with one-axle trailers Cars with two-axle trailers	2, 3, or 4
3	Other Two-Axle Four-Tire Single-Unit Vehicles	Pick-ups and vans Pick-ups and vans with one- and two-axle trailers	2, 3, or 4
4	Buses	Two- and three-axle buses	2 or 3
5	Two-Axle, Six-Tire, Single-Unit Trucks	Two-axle trucks	2
6	Three-Axle Single-Unit Trucks	Three-axle trucks Three-axle tractors without trailers	3
7	Four or More Axle Single-Unit Trucks	Four-, five-, six- and seven-axle single-unit trucks	4 or more
8	Four or Fewer Axle Single-Trailer Trucks	Two-axle trucks pulling one- and two-axle trailers Two-axle tractors pulling one- and two-axle trailers Three-axle tractors pulling one-axle trailers	3 or 4
9	Five-Axle Single-Trailer Trucks	Two-axle tractors pulling three-axle trailers Three-axle tractors pulling two-axle trailers Three-axle trucks pulling two-axle trailers	5
10	Six or More Axle Single-Trailer Trucks	Multiple configurations	6 or more
11	Five or Fewer Axle Multi-Trailer Trucks	Multiple configurations	4 or 5
12	Six-Axle Multi-Trailer Trucks	Multiple configurations	6
13	Seven or More Axle Multi-Trailer Trucks	Multiple configurations	7 or more

Diagram of the FHWA 13-class vehicle classification system, illustrating vehicle categories defined by axle count and configuration for engineering and traffic monitoring purposes.

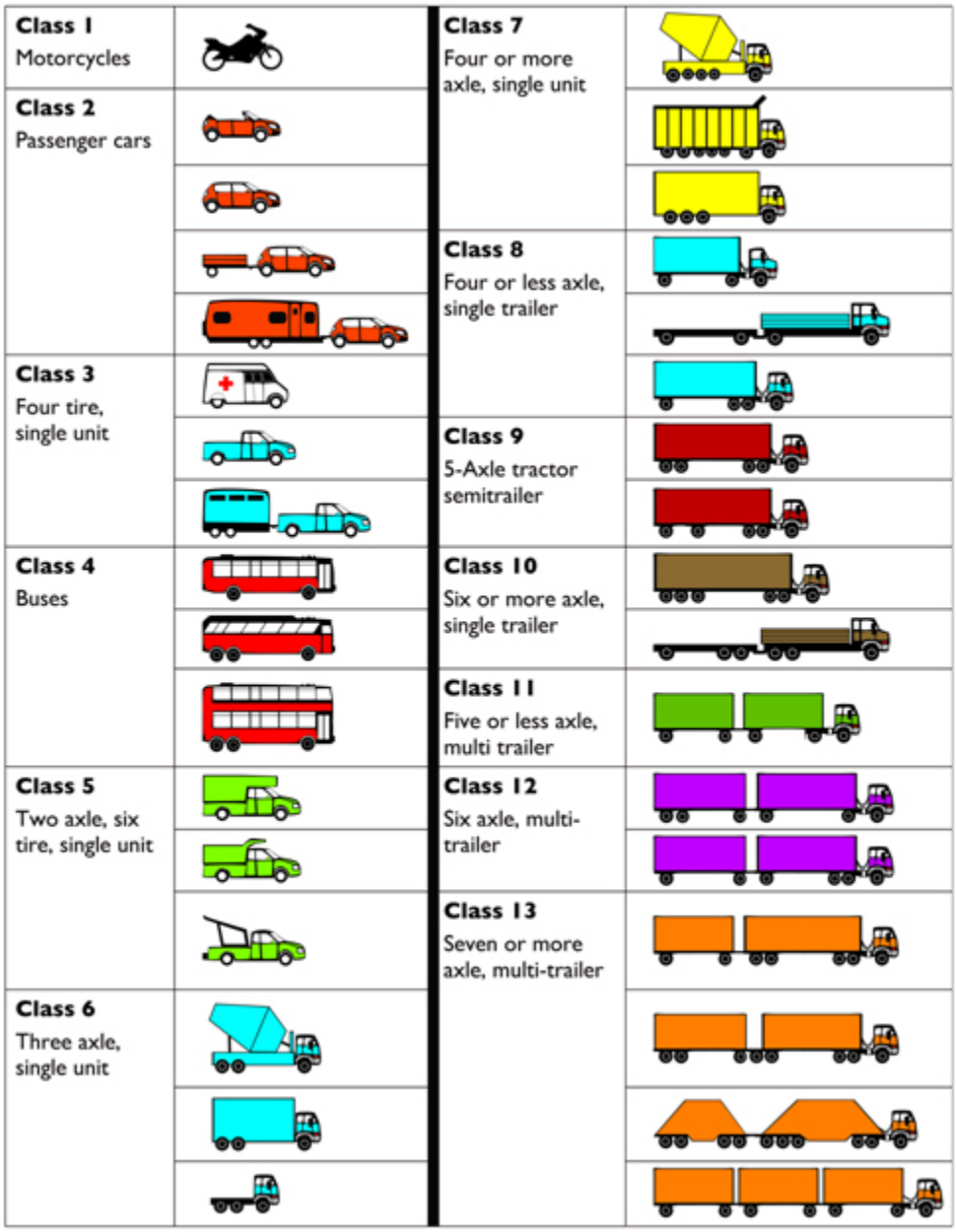


Figure 1. FHWA 13 vehicle category classification [4]

Despite its long-standing use and engineering relevance, applying the FHWA 13-bin system to video-based machine learning presents several significant challenges. These challenges

stem from the complexity of vehicle configurations, variability in traffic conditions, and limitations in available data.

The challenges involved in the FHWA vehicle classification system are as follows:

- **Similarity in Appearance:** Different vehicle types can appear similar, such as two-axle trailers in Class 3 and two-axle trucks towing trailers in Class 8 (as shown in Figure 2), complicating classification with a single-view camera.

Figure 2. Similar appearance. Right: Class 3 light truck pulling a trailer [4]. Left: Class 8 truck pulling a trailer [4].

- **Occlusion in Images or Videos**: Images or videos, particularly in high-traffic scenarios, may encounter occlusion problems. For example, a large truck masking a smaller one can obstruct the complete view of the vehicle's axle information. This presents challenges for classification systems based on axle features.
- **High-Speed Travel Ambiguity**: In scenarios involving high-speed motion, the task of discerning whether a trailer and its towing vehicle constitute separate entities poses a significant challenge. This issue notably affects the accurate classification of vehicles within categories 3, 8, 9, and 10.
- **Uneven and Long-Tailed Datasets:** The effectiveness of machine learning models in classifying vehicles is hampered by limited data on rarer vehicle types, such as those in

classes 11, 12, and 13. These larger trucks are significantly less common than vehicles in other classes, leading to a long-tail, uneven distribution in the dataset.

- **Axle Number Focus:** In the FHWA classification system, the number of axles plays a vital role, yet it represents just a small component of a vehicle's overall structure. While humans can easily identify the number of axles, machines face a significant challenge in pinpointing the exact location and distinguishing axles from other parts. Developing machine learning algorithms that effectively focus on and accurately identify axle features presents a notable challenge in this domain.
- **Environmental Variability:** Environmental variability adds another layer of complexity. Daytime, nighttime, and rainy conditions dramatically affect video quality, introducing shadows, glare, headlight bloom, reflections, and reduced visibility. In nighttime environments, dark vehicle undersides make axles nearly invisible, while rain creates reflections on wet pavement that obscure wheel boundaries. These factors degrade the visibility of structural features—such as axles, tires, trailer gaps, and hitch points—that are essential for accurate FHWA classification, making robust performance across conditions extremely difficult.

2.0 LITERATURE REVIEW

2.1 OVERVIEW

Recent advances in computer vision and machine learning have greatly expanded the capabilities of video-based traffic sensing, enabling automatic detection and classification of vehicles under diverse environmental conditions. A substantial body of research has explored techniques ranging from object detection and fine-grained vehicle classification to image enhancement and transfer learning for adverse weather. These developments provide a strong foundation for automated traffic monitoring; however, they also reveal significant limitations when applied to the FHWA 13-bin classification framework.

Existing object detection models—such as YOLO, Faster R-CNN, RetinaNet, and transformer-based detectors—achieve real-time performance and high accuracy in identifying broad vehicle categories. Yet most studies focus on general classes (e.g., car, truck, bus) and do not attempt to distinguish the axle-based or configuration-specific categories required for FHWA reporting. Similarly, research on vehicle classification has progressed from simple appearance-based methods to deep-learning models capable of fine-grained categorization, but publicly available datasets rarely include the multi-axle and multi-trailer configurations that dominate the upper FHWA bins. As a result, prior classification work has limited applicability to engineering needs involving pavement loading, freight modeling, or axle-based analysis.

To improve performance in challenging visual conditions, a growing line of research has examined image enhancement for nighttime and adverse weather, including low-light enhancement, dehazing, and reflection suppression. These approaches help restore missing visual cues such as axle shadows, trailer boundaries, and wheel features. However, many enhancement

networks are computationally expensive and difficult to deploy on edge hardware, limiting their real-world utility for field-based traffic sensing.

Another closely related area is transfer learning and domain adaptation across environmental conditions. Performance degradation between daytime and nighttime, or between clear and rainy weather, is well documented in detection and classification tasks. Methods such as weather-invariant feature learning, style transfer, and mixture-of-experts architectures aim to reduce this gap by adapting models to variable conditions. These techniques show promise, particularly for roadside systems exposed to Washington state's diverse weather and lighting environments.

Collectively, the literature demonstrates significant progress but also highlights clear gaps: most existing models do not address the fine-grained axle-based distinctions of the FHWA 13-bin system; few datasets represent rare heavy-duty truck classes; and environmental robustness remains a challenge for real-world deployment. These limitations underscore the need for a specialized, integrated approach that combines advanced detection, fine-grained classification, visibility enhancement, and cross-weather adaptation.

The following sections provide a detailed review of each of these areas—object detection, vehicle classification, image enhancement, and transfer learning—to establish the technical foundation for the system developed in this project.

2.2 EXISTING VEHICLE CLASSIFICATION AND DETECTION ALGORITHM

To understand the technological landscape for vehicle classification and detection, it is important to examine the commonly used sensing approaches and assess how well they align with the requirements of FHWA's 13-bin classification system. Existing methods span a broad range of sensing technologies, including pneumatic tubes, inductive loops, weight-in-motion

(WIM) systems, machine vision, and light detection and ranging (LiDAR). These technologies are widely used in traffic monitoring, yet their suitability varies significantly when evaluated against the need for accurate axle-based classification and fine-grained category distinctions required by FHWA reporting.

2.2.1 Pneumatic Tube and Loop Detectors

Pneumatic tubes were among the earliest vehicle-sensing technologies, first deployed in the 1920s and still widely used for short-term traffic data collection today [4][9][13]. When a vehicle passes over the tube, changes in air pressure allow the system to detect axle counts and approximate axle spacing, as illustrated in Figure 3 Although pneumatic tubes are useful for basic axle-based counts, their temporary installation, vulnerability to wear, and poor durability under high-volume or high-speed conditions limit their suitability for highway-scale monitoring.

Illustration of pneumatic tube detectors used for traffic monitoring, showing how axle counts are measured through air pressure changes when vehicles pass over the tubes.

Figure 3. Pneumatic tube detectors

Inductive loop detectors provide a more permanent alternative. Embedded within the pavement, loops detect changes in magnetic fields as vehicles pass overhead. They can estimate vehicle length [9][10][11] and, when deployed in dual-loop configurations, can measure vehicle

speed [12]. While loops operate continuously and at relatively low cost, they struggle to separate closely spaced vehicles in congested conditions. Furthermore, neither pneumatic tubes nor loop detectors provide the detailed structural information needed to distinguish multi-unit trucks, tractor–trailer combinations, or multi-trailer configurations—categories that are essential for FHWA's 13-bin system.

2.2.2 Weight-In-Motion Systems

Weight-in-motion (WIM) systems play an important role in traffic data collection and axle-based vehicle classification. As described in prior studies [14], WIM systems are designed specifically to measure dynamic axle loads for vehicles in motion. Typically installed within pavement or bridge structures (as shown in Figure 4), WIM systems integrate multiple sensors, digital cameras, and computational units to collect axle-weight data and classify vehicles accordingly. Various detection techniques used in WIM systems have been detailed in the literature [15][16].

Despite their value for freight monitoring and pavement analysis, WIM systems are constrained by high installation and maintenance costs and function only at fixed locations. Their infrastructure requirements make them impractical for widespread deployment on local or rural roadways, particularly where flexible, mobile data collection is needed.

Figure 4. Weight-in-motion systems

2.2.3 Machine Vision Systems

Machine vision has emerged as the most prevalent and scalable sensor technology for vehicle classification. Video-based systems can capture rich visual information—including color, texture, shape, and semantic features—from traffic scenes, making them well-suited for large-scale deployment. Vehicle classification approaches using machine vision can be broadly categorized into prior-knowledge methods and deep learning methods.

Prior knowledge Methods

Traditional machine-vision approaches typically follow a two-step process: hypothesis generation and hypothesis verification. In the first step, candidate vehicle regions of interest (ROIs) are identified. In the second step, feature-based criteria are used to confirm whether an ROI contains a vehicle. Common techniques include the following:

Color features: Vehicle color distributions often differ from background scenes. Using color channels or threshold segmentation can help isolate vehicles from their surroundings.

Edge features: Vehicles exhibit regular geometric shapes and strong linear edge structures. Vertical and horizontal edge patterns provide cues for locating and identifying vehicles.

Prior-knowledge methods are computationally efficient and straightforward to implement, but their performance is highly sensitive to environmental variability, occlusions, and complex traffic scenes. These limitations restrict their usefulness for FHWA 13-bin classification, which requires detailed and robust structural discrimination.

Deep Learning Methods

Deep learning, especially convolutional neural networks (CNNs), has become the dominant approach for vehicle detection and classification. Deep-learning-based methods can be grouped into two-stage and one-stage detection frameworks.

Two-Stage Methods: The two-stage method is focused on creating proposals for vehicle regions and subsequently identifying vehicles within these proposed zones, as shown in Figure 5 Prominent examples of this method include Region-Based Convolutional Neural Networks (R-CNN) [17], Spatial Pyramid Pooling Network (SPP-Net) [18], Fast R-CNN [16], and Faster R-CNN [20]. These methods have substantially improved classification accuracy.

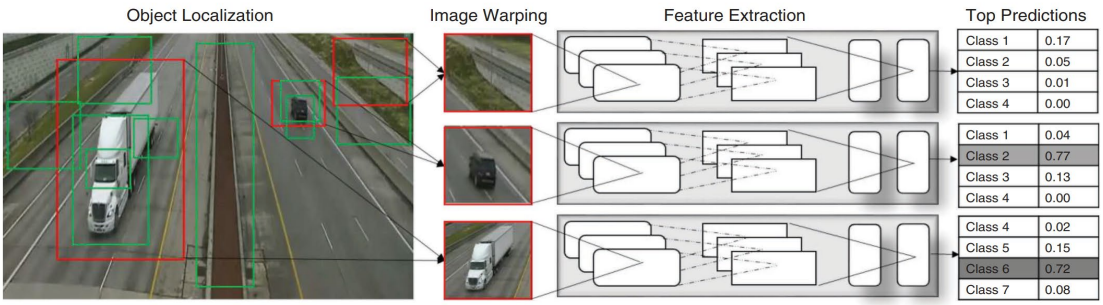


Figure 5. Two-stage methods [19].

The R-CNN [17] algorithm (as shown in Figure 6) starts with a sparse sampling of the original input image, focusing on selected candidate regions. It then employs CNN to extract features from these regions, followed by the classification of this extracted information using a support vector machine (SVM). R-CNN has shown notable enhancement in detection accuracy and a reduction in the limitations that typically accompany traditional target detection algorithms.

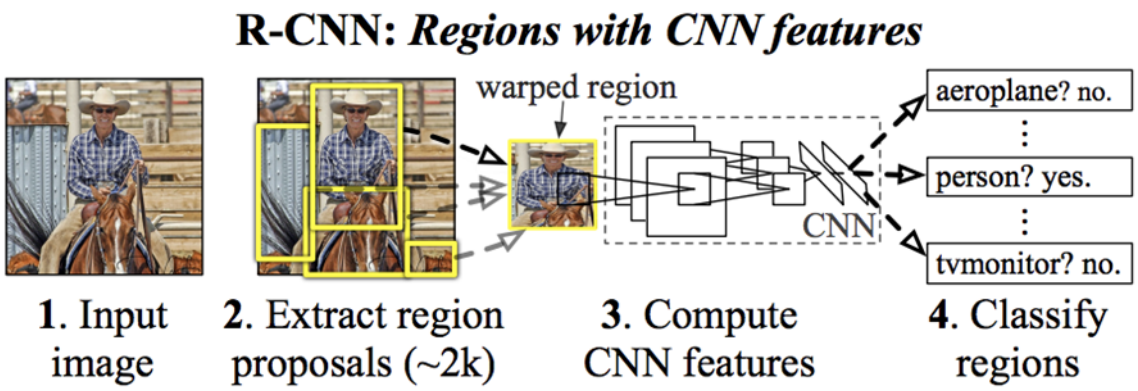


Figure 6. R-CNN algorithm [14].

SPP-Net [18] introduces a spatial pyramid pooling operation between the convolution and full connection layers. This process extracts the features of the input images once, creating images of a fixed scale. With the optimization of the SPP network, Fast R-CNN [19] adopts a multi-task loss function, replacing the SVM for classification. This allows for both classification and frame regression to be trained within the same network framework.

However, the traditional method of generating candidate regions, which relies on CPU processing, can limit the overall system speed. This limitation was addressed with the advent of Faster R-CNN [20], which enables end-to-end target detection. Faster R-CNN first generates multiple target candidate regions, then extracts their features using region mapping and spatial pyramid pooling. Subsequently, it identifies various target attributes through multiple branches

in the CNN, working in parallel. A significant advancement of Faster R-CNN over Fast R-CNN is the accelerated speed of region generation, owing to the features of the Region Proposal Network (RPN). It classifies detection targets by first assessing whether the candidate box meets the feature requirements and then applying the multi-task loss function for further processing.

Ji-qing Luo et al. [21] developed an advanced model using the Faster R-CNN framework, optimized with neural architecture search (NAS) and feature enrichment, for effective detection of multi-scale vehicles in traffic scenes. Their methodology begins with a Retinex-based Image Adaptive Correction algorithm (RIAC) to enhance image quality in the dataset, reducing shadow and illumination issues. They then implement NAS on the Faster R-CNN's backbone network, enabling optimal extraction of multi-layer features through improved cross-layer connections. Additionally, their model incorporates an object feature enrichment technique, merging multi-layer features with contextual information to enhance vehicle target identification, particularly in challenging scenarios such as small-scale or heavily occluded targets. Tested on the UA-DETRAC dataset, their approach achieves a 70 percent classification accuracy for diverse vehicle types including vans, cars, buses, and others.

Adu-Gyamfi et al. [19] devised a vehicle classification system utilizing a deep convolutional neural network (DCNN). This system is adept at swiftly and precisely extracting vehicular features. Distinct from other methodologies, this DCNN model is initially pre-trained with an auxiliary dataset [22] and subsequently fine-tuned using domain-specific data acquired from the Virginia and Iowa DOT closed-circuit television (CCTV) camera databases. The process commences with localization, employing an algorithm named Selective Search to generate class-independent region proposals. These proposals are then processed through the DCNN, which extracts unique feature descriptors after resizing the regions to a uniform square

dimension (256 × 256). The extracted features for each region are subsequently classified using a linear SVM scoring system. The classification aligns with FHWA's 13 vehicle types, but some classes are merged as shown in Table 2. The system demonstrated classification accuracy ranging from 82 percent to 100 percent across seven classes.

Table 2. Merged classes

FHWA Classification	**Proposed Video-Based Classification**
Class 1. Motorcycles	Class 1. Motorcycles
Class 2. Passenger cars	Class 2. Passenger cars
Class 3. Pickups and vans	Class 3. Pickups and vans
Class 4. Buses	Class 4. Buses
Class 5. Single-unit, 2-axle trucks	Class 5. Single-unit trucks (FHWA classes 5, 7, 7)
Class 6. Single-unit, 3-axle trucks	Class 5. Single-unit trucks (FHWA classes 5, 7, 7)
Class 7. Single-unit, 4-axle trucks	Class 5. Single-unit trucks (FHWA classes 5, 7, 7)
Class 8. Single-trailer, 3- or 4-axle trucks	Class 6. Single-trailer trucks (FHWA classes 8, 9, 10)
Class 9. Single-trailer, 5-axle trucks	Class 6. Single-trailer trucks (FHWA classes 8, 9, 10)
Class 10. Single-trailer, 6+-axle trucks	Class 6. Single-trailer trucks (FHWA classes 8, 9, 10)
Class 11. Multi-trailer, 5- or fewer axle trucks	Class 7. Multi-trailer trucks (FHWA classes 11, 12, 13)
Class 12. Multi-trailer, 6-axle trucks	Class 7. Multi-trailer trucks (FHWA classes 11, 12, 13)
Class 13. Multi-trailer, 7- or more axle trucks	Class 7. Multi-trailer trucks (FHWA classes 11, 12, 13)

One Stage Methods: The one-stage detection algorithm streamlines the process of vehicle detection by omitting the step of generating vehicle region proposals and integrating both vehicle identification and detection into a unified network. In comparison to two-stage methods, one-stage methods are much faster and simpler. Since it bypasses the region proposal stage, the overall pipeline is more straightforward, which can make it easier to implement and optimize (as shown in Figure 7). Notable examples of one-stage methods include YOLO V1[23], SSD [24],

YOLO V2[25], YOLO V8 [26], YOLO V11[29]. The YOLO series, particularly YOLO, YOLO V2, and YOLO V11 stand as prominent representatives of the single-stage target detection algorithm. These algorithms transform the task of target detection into a regression problem, facilitating end-to-end identification within the detection process.

The YOLO network operates by dividing the input image into a grid system. Each grid is responsible for detecting the target object whose center point lies within that specific grid. The network then uses anchor frames within each grid to replace the candidate frames used in two-stage detection networks. This approach significantly enhances the detection rate of the network and is especially beneficial for tasks requiring high real-time detection efficiency. Figure 8 shows the result of vehicle classification using YOLO.

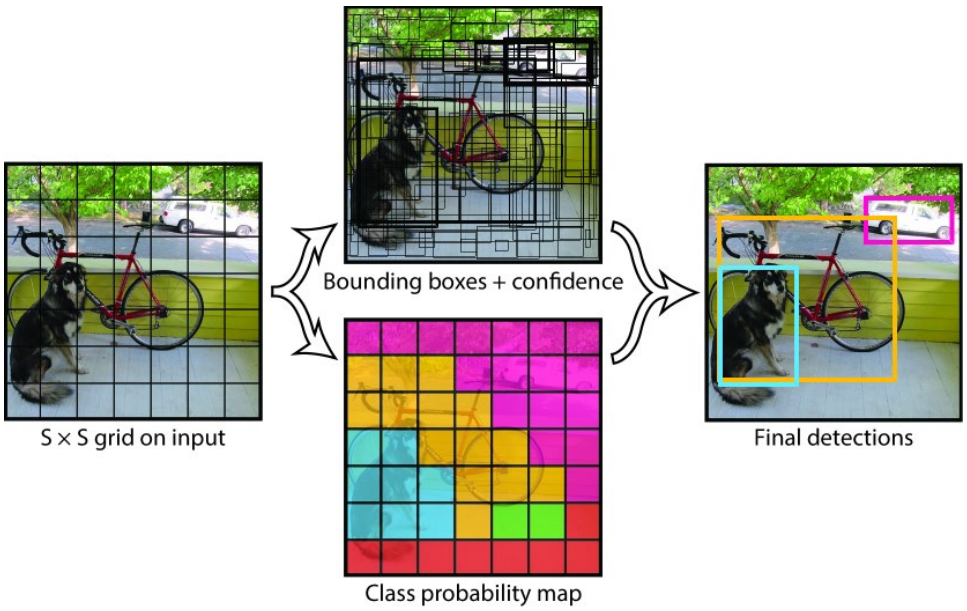


Figure 7. YOLO structure [21]

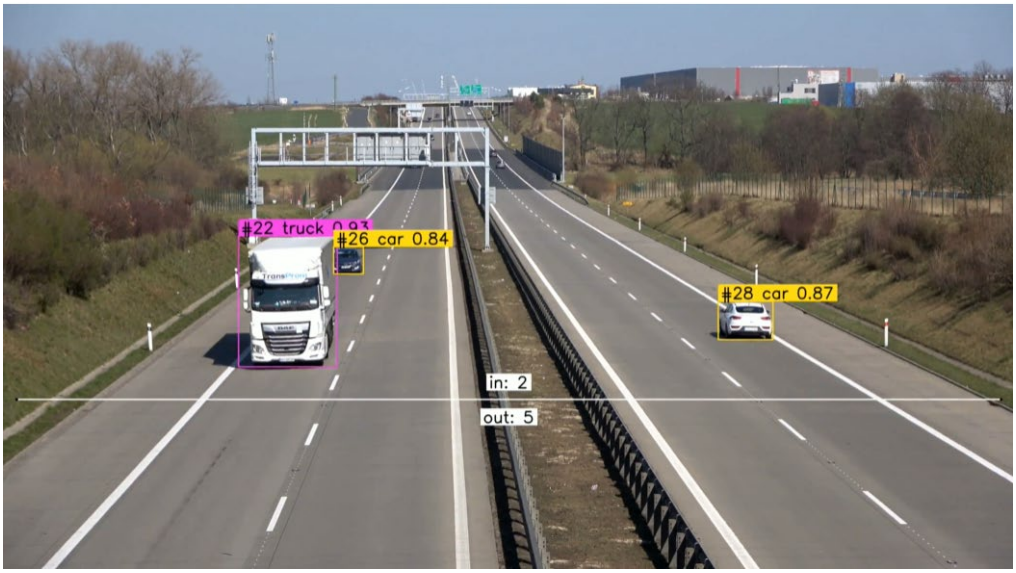


Figure 8 Vehicle classification with YOLO

2.2.4 LIDAR Systems

In the realm of vehicle classification systems, light detection and ranging (LIDAR) sensors stand out for their ability to emit eye-safe laser lights and record environmental reflections, such as roads, passing vehicles, and vegetation. These data are then used to extract vital features such as the size and shape of passing cars, with LIDAR proving particularly adept at identifying vehicle shapes because of its high precision.

Asborno et al.'s research [28] focused on the classification of truck body types using LIDAR technology. Their system employs two roadside LIDAR units, defining two crucial features: the duration that a vehicle remains in front of the unit and an array of vehicle body points that delineate the shape of the truck. These features are inputs to various classifiers, including decision tree (DT), artificial neural network (ANN), SVM, and Naive Bayes (NB), to facilitate vehicle classification. Deployed at an Interstate location, the system successfully classified vehicles into five distinct truck body types (five-axle tractor-trailers: van and

container, platform, low-profile trailer, tank, and hopper and end dump) with an impressive accuracy of up to 96 percent.

This LIDAR-based system offers advantages over camera-based technologies, particularly in terms of reduced susceptibility to environmental factors and enhanced adaptability in diverse weather conditions. However, a significant challenge in LIDAR-based systems is the issue of vehicle occlusion, and their precision is less than optimal when detecting stationary or slow-moving vehicles, particularly in measuring lateral distances.

2.3 SUMMARY

A number of sensing and classification technologies have been explored for traffic monitoring, each with specific strengths and limitations. Pneumatic tubes, loop detectors, and WIM systems offer axle-related measurements but lack the scalability and granularity needed for FHWA's 13-bin categories. Machine-vision approaches, while more flexible and cost-effective, typically focus on broad vehicle types rather than detailed axle-based classifications. Efforts such as Adu-Gyamfi et al. [22] demonstrated early attempts at FHWA-aligned classification but still merged categories and achieved only partial coverage of the 13-bin system.

These limitations underscore the need for a dedicated, fine-grained machine-vision solution capable of classifying all 13 FHWA vehicle categories with high accuracy—especially for rare multi-axle and multi-trailer classes—within real-world, high-volume traffic environments.

3.0 SENSORS DEPLOYMENT AND DATA COLLECTION

3.1 SYSTEM AND DATA OVERVIEW

The development of an FHWA 13-bin vehicle classification system requires a sensing platform capable of capturing high-quality visual information and a dataset that reflects the full diversity of truck and trailer configurations present on Washington state highways. Because existing public datasets do not include axle counts, trailer structures, or multi-trailer combinations, and lack the heavy-vehicle variety necessary for FHWA's engineering requirements, this project relies on a customized data collection approach supported by a dedicated roadside sensing unit. This chapter provides an overview of the sensing system used in this study, a review of publicly available datasets and their limitations, and the data collection methodology adopted to create a dataset suitable for axle-based vehicle classification.

At the core of the sensing system is the Mobile Unit for Sensing Traffic (MUST), a portable, edge-computing sensor platform deployed along Interstate 5. The MUST unit is equipped with a pan–tilt–zoom camera that enables flexible adjustment of viewing angles and supports high-resolution video recording across different traffic and environmental conditions. Its roadside mounting position allows for clear visibility of lower vehicle structures, including wheels and axles, which is essential for distinguishing among the 13 FHWA vehicle classes. The system's ability to perform local processing further improves data quality by ensuring stable recording even in situations where communication infrastructure is limited.

A detailed examination of commonly used public vehicle datasets shows that none provide the axle-level or configuration-specific details required for FHWA-compliant classification. Datasets such as Stanford Cars [5], CompCars [6], BDD100K [7], and UA-DETRAC [8] primarily focus on broad vehicle categories or consumer-level identification and

do not include heavy-duty truck types, multi-axle configurations, or articulated trailer combinations. As such, they are inadequate for tasks involving pavement design, freight analysis, and detailed roadway engineering. This gap underscores the need to build a custom dataset that captures all FHWA classes under real-world operating conditions.

To address this need, a structured data collection process was conducted using the MUST unit. Video recordings were captured during controlled time windows to minimize occlusion and ensure clear visibility of vehicle axles and trailer configurations. Data were collected from multiple viewpoints to capture both side-profile and oblique angles, providing complementary information about vehicle geometry and articulation. The resulting dataset offers high-resolution, multi-angle imagery suitable for fine-grained axle-based classification and serves as the foundation for model training and evaluation.

Together, the sensing system, public dataset review, and custom data collection process provide the necessary framework for developing an accurate and robust FHWA 13-bin vehicle classification model. The sections that follow describe each of these components in detail, beginning with the deployment of the MUST sensor along Interstate 5.

3.2 SENSORS DEPLOYMENT

To collect high-quality data for FHWA 13-bin vehicle classification, a Mobile Unit for Sensing Traffic (MUST) sensor was deployed along Interstate 5 (I-5). The selected deployment location is situated at latitude 47.01725, longitude –122.90585, a segment of I-5 known for its steady flow of mixed freight vehicles. This site provides an advantageous vantage point that offers clear visibility of a wide variety of truck types, including single-unit trucks, tractor–trailers, and multi-trailer combinations. Diversity of vehicle configurations is essential for

developing and validating a machine-vision-based classifier targeting all FHWA 13 categories. The roadway environment at the deployment location is shown in Figure 9.

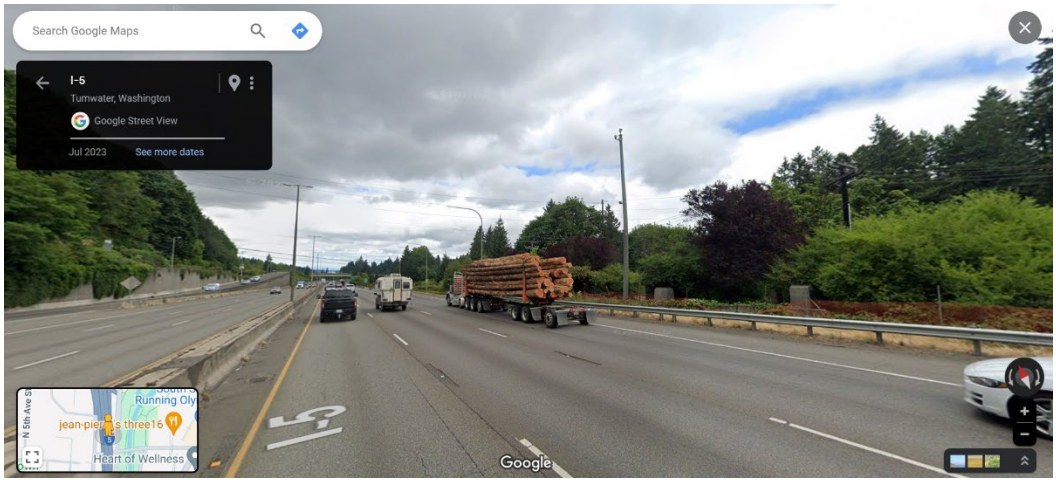


Figure 9. The location of the MUST sensor.

The MUST unit (Figure 10) is a portable, edge-computing traffic sensing platform designed for rapid deployment, continuous operation, and local data processing. It integrates a Pan-Tilt-Zoom (PTZ) camera for flexible viewing control, wireless communication modules (Bluetooth and Wi-Fi) for local access and configuration, and environmental sensors such as temperature and humidity probes to record contextual conditions during data collection. For the purposes of this study, the PTZ camera is the primary sensing component, providing high-resolution video suitable for axle-based vehicle classification.

Figure 10. The MUST sensor.

A key advantage of the MUST unit is its ability to perform on-device processing using edge-based AI hardware. This capability minimizes bandwidth requirements and enables efficient collection of high-resolution videos under varied environmental conditions, including daytime, nighttime, and rain. Since FHWA 13-bin classification relies on distinguishing fine-grained structural features—such as axles, trailer articulation points, wheel spacing, and tractor configuration—the high-quality imagery and stable local processing provided by MUST are essential for reliable model development.

For roadside deployment, the MUST sensor is mounted securely to an existing pole infrastructure using industrial-grade stainless steel hose clamps. The installation height ranges from 20 to 30 feet, selected to ensure an optimal downward viewing angle that captures both the overall vehicle profile and the wheel/axle region clearly. A higher installation position provides greater coverage of multiple lanes, reduces occlusion, and minimizes motion blur for high-speed vehicles. At the same time, the height is low enough to allow the camera to resolve axle-level details—critical for FHWA-based vehicle classification. The installed MUST sensor is shown in Figure 11.

Figure 11. The installed MUST sensor.

For power, the system uses an outdoor-rated AC-to-DC transformer, converting standard 120V AC to the required DC input for the sensing unit. This configuration allows deployment on utility poles or roadside infrastructure equipped with a power outlet. Following installation, the PTZ camera is calibrated to establish a stable field of view covering two to three lanes of mainline traffic, ensuring consistent vehicle visibility and minimizing occlusions.

The specific placement of the MUST unit at roadway level rather than at an overhead gantry was intentional. FHWA 13-bin classification requires visibility of axles, wheel spacing, and tractor–trailer geometry—features best observed from a shallow angle along the side profile of the vehicle. A roadside mounting position offers superior visibility of the lower vehicle structure, which overhead cameras often fail to capture. This placement supports the collection of more detailed imagery, improving classification accuracy for heavy-duty trucks and multi-trailer combinations, which are historically difficult to classify.

In summary, the deployment of the MUST sensor along I-5 provides a strategically positioned, high-quality video dataset essential for developing a fine-grained FHWA 13-bin vehicle classifier. The combination of optimal sensor placement, stable edge-based recording, and diverse truck traffic ensures that the collected data are representative of real-world operating conditions and well suited for machine-learning-based model development.

3.3 PUBLIC DATASETS

In this section, we review several widely used public vehicle datasets and evaluate their suitability for the Federal Highway Administration's (FHWA) 13-bin vehicle classification system. Because FHWA classification relies on axle configuration, trailer articulation, and vehicle length, it requires datasets that provide detailed structural information—attributes rarely emphasized in mainstream computer-vision research. Many public datasets focus on general-purpose vision tasks or consumer-level identification and therefore lack the engineering relevance needed for fine-grained, axle-based classification. The following discussion highlights the content and limitations of major public datasets relative to FHWA requirements.

The Stanford Cars Dataset [5] contains 16,185 images covering 196 classes of passenger cars. These classes are defined by differences in vehicle make, model, and production year, and the images are primarily collected from curated web sources such as car-sale listings. While this dataset is valuable for fine-grained consumer vehicle recognition, it includes only passenger cars and does not contain heavy trucks, buses, trailers, or multi-axle commercial vehicles. It also lacks any annotation related to axle count, vehicle length, or trailer structure. As a result, the dataset has limited applicability to engineering-oriented tasks that require FHWA's 13-bin classification.

The CompCars Dataset[6] also focuses heavily on consumer vehicles. Its web subset includes hundreds of makes and over a thousand models, while its surveillance subset contains approximately 50,000 front-view images recorded by traffic cameras. Despite its scale, the dataset does not include heavy trucks, tractor–trailers, or multi-trailer combinations—vehicle types that dominate the upper FHWA classes. Furthermore, the available images do not provide axle information or trailer configurations, and the surveillance data are restricted to front-view imagery, making it impossible to distinguish axle groups or determine vehicle length. Thus, although CompCars is useful for brand-level identification, it is not designed for FHWA classification.

The BDD100K dataset [7] is one of the largest and most diverse driving video datasets, containing 100,000 high-definition video sequences recorded under various lighting, weather, and roadway conditions. It includes annotations for broad vehicle categories such as car, truck, and bus and is widely used in autonomous-driving research. However, because all heavy trucks are labeled simply as "truck," the dataset does not differentiate between single-unit trucks, tractor–trailers, tandem-axle trucks, or multi-trailer combinations. It also does not provide axle counts or length-related attributes. These limitations make the dataset unsuitable for detailed engineering analysis or FHWA-level classification.

The UA-DETRAC dataset [8] provides over 140,000 real-world traffic frames collected from 24 urban locations and includes more than one million bounding-box annotations. The vehicle categories in this dataset are limited to cars, buses, vans, and others. Although UA-DETRAC is a valuable benchmark for multi-object detection and tracking, its labeling structure does not support the fine-grained distinctions required by FHWA. It contains no information about trailers, axle groupings, articulated vehicles, or heavy truck subclasses such as three-axle

trucks, four-axle tractors, or multi-trailer configurations. Because of this, the dataset does not align with the detailed classification needs of roadway engineering or pavement design.

In summary, existing public vehicle datasets can generally be grouped into two themes: datasets that focus on general object detection with broad vehicle categories, and datasets that emphasize consumer-level distinctions such as make and model. Neither category provides the axle-level, length-based, or configuration-specific information required for FHWA's 13-bin vehicle classification. The absence of heavy-duty truck diversity, lack of multi-trailer vehicles, and limited representation of real-world roadside viewpoints further restrict their applicability. Consequently, these datasets cannot support the development of a reliable engineering-grade classifier for state transportation agency needs. To overcome these limitations, it is necessary to collect a custom dataset under real Washington state roadway conditions, capturing diverse vehicle types across all 13 FHWA bins, including rare heavy-truck categories and varied environmental conditions. The following section describes the data collection methodology used to build such a dataset using the MUST sensor.

3.4 DATA COLLECTION USING THE MUST SENSOR

As discussed in the previous section, existing public datasets do not provide the axle-level, configuration-specific categories required for FHWA's 13-bin vehicle classification system. To address this gap and develop a dataset tailored to Washington state roadway conditions, we conducted our own data collection using the Mobile Unit for Sensing Traffic (MUST) sensor.

The data collection process was designed to maximize the visibility of vehicle axles, trailers, and structural configurations while minimizing occlusion and labeling ambiguity. To achieve this, video data were captured twice daily—once at 8:00 AM and once at 9:00 PM—with

each session lasting one hour. These specific time windows were selected to coincide with relatively low traffic density, which improves the clarity of vehicle profiles and reduces the likelihood of vehicles obscuring each other. From each hour of video, approximately 7,200 image frames are extracted, followed by a quality-control procedure to filter out frames where vehicles are partially obstructed or not fully within the field of view.

The MUST sensor supports remote configuration, allowing operators to adjust viewing angle and orientation without on-site intervention. To capture the structural details needed for FHWA 13-bin classification, data were collected from two distinct viewpoints. The first viewpoint is parallel to the roadway, providing a clear and unobstructed view of truck wheels and axles—critical information for distinguishing between tractor–trailers, multi-unit trucks, and multi-trailer configurations. The second viewpoint uses a tilted oblique angle, which better captures the front profile, overall vehicle geometry, and trailer articulation. Together, these two perspectives provide complementary visual information, improving the diversity and robustness of the resulting dataset. Figure 12 illustrates the two viewpoints used during data collection.

Figure 12 Two viewpoints of MUST sensor.

This dual-angle strategy is particularly important for FHWA 13-bin classification, where differences between vehicle classes are often subtle and depend on features such as axle

groupings, fifth-wheel connections, and trailer combinations. By combining side-profile and oblique-view imagery, the collected dataset more accurately captures the variability of real-world truck configurations. This approach also enhances the generalization capability of downstream machine-learning models by exposing them to multiple viewing geometries and environmental conditions.

In summary, the MUST-based data collection procedure ensures the acquisition of high-resolution, multi-view, and low-occlusion vehicle imagery that is well suited for developing a fine-grained FHWA 13-bin classification model. The planned timing, multi-angle strategy, and careful frame selection process collectively contribute to the quality and usability of the dataset for model training and evaluation.

3.5 DATASET ANNOTATION

After collecting the raw video data from the MUST sensor, a standardized annotation process was performed to create high-quality training samples for FHWA 13-bin classification. The first step involved selecting frames that contain fully visible and unobstructed vehicles, ensuring that axle configuration, trailer articulation, and vehicle geometry were clearly captured. From each video segment, frames were extracted and evaluated, and only those meeting visibility and clarity criteria were retained for annotation.

For each selected frame, bounding boxes were manually drawn around all trucks to identify their spatial location within the image. Following this, each vehicle was assigned an FHWA class label based on axle count, axle arrangement, and vehicle type. This consistent annotation protocol (as shown in Figure 13) ensures that all samples follow the same interpretation rules and reflect the structural distinctions embedded in the FHWA 13-bin standard. A total of approximately 4,000 images were selected from thousandsof processed

frames. These images were carefully curated to represent a diverse range of truck types, orientations, and configurations, prioritizing samples where vehicles are centered, fully visible, and free from motion blur or occlusion.

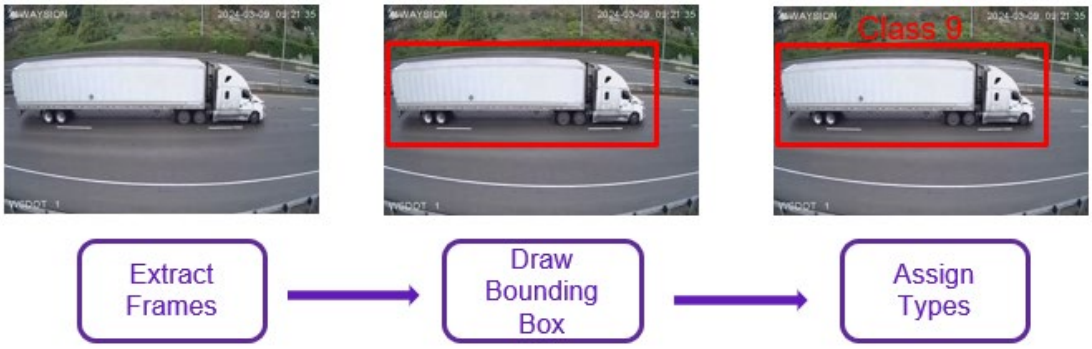


Figure 13. Annotation process.

This annotation strategy ensures a high-quality dataset tailored specifically to the needs of axle-based classification. By focusing on clarity, structural visibility, and representativeness, the resulting dataset provides a reliable foundation for training machine-learning models capable of distinguishing between all 13 FHWA vehicle categories.

3.6 DATASET ANALYSIS AND IMBALANCE HANDLING

Following the annotation process, the resulting dataset was analyzed to understand its distribution across FHWA vehicle classes. As shown in Figure 14, the collected roadway data exhibit a long-tailed distribution characteristic of real-world traffic flows. Passenger cars (Class 2) and two-axle, four-tire single-unit vehicles (Class 3) dominate the dataset, together accounting for more than 90 percent of the observed vehicles. Class 2 includes 4,823 samples, while Class 3 contains 1,988 samples. The remaining 11 FHWA classes collectively represent less than 10 percent of the dataset, reflecting the low frequency of heavy-duty trucks and multi-trailer combinations on the corridor. This severe imbalance presents a challenge for training

classification models, as underrepresented classes may be misclassified or under-learned without targeted mitigation strategies.

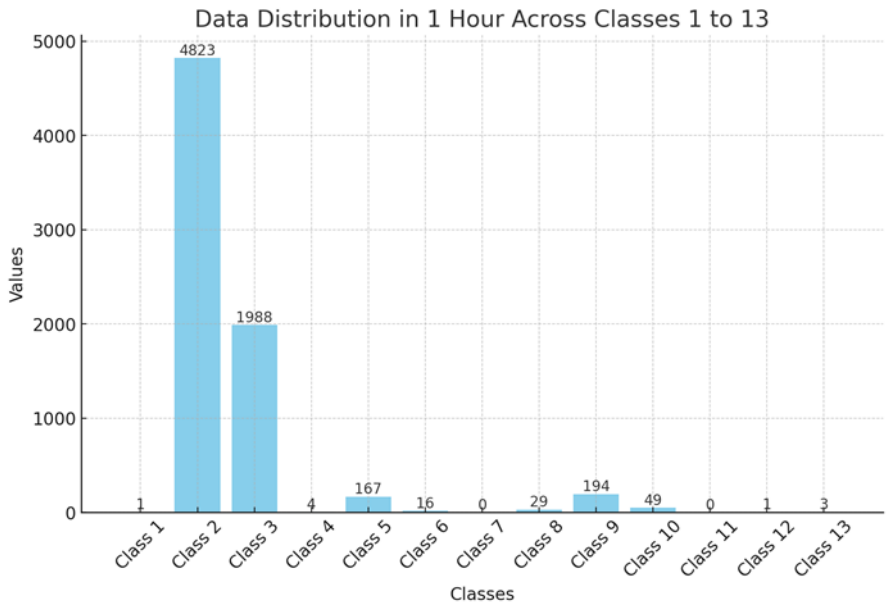


Figure 14 Data distribution in one-hour classes

To address this issue, a combination of under-sampling and over-sampling techniques was employed, as illustrated in figures 15 and 16. For classes with very large numbers of samples—primarily Class 2 and Class 3—only a selected subset of images was retained to reduce their dominance while preserving scene variability. This under-sampling strategy prevents the model from overfitting to common classes. Conversely, for rare classes such as single-trailer and multi-trailer trucks, additional samples were generated through over-sampling, including selecting frames that capture varied orientations, distances, and vehicle movements. This enhances their representation in the dataset and ensures the model is exposed to sufficient examples of each truck type.

Figure 15. Sampling methods.

This combined resampling approach results in a more balanced dataset (as shown in Figure 17 Image-level class distribution of the preprocessed dataset that promotes equitable learning across all FHWA categories. By reducing the skew toward common vehicle types and increasing the visibility of rare but engineering-critical classes, the dataset becomes more suitable for training robust machine-learning models capable of performing accurate 13-bin classification in real-world conditions.

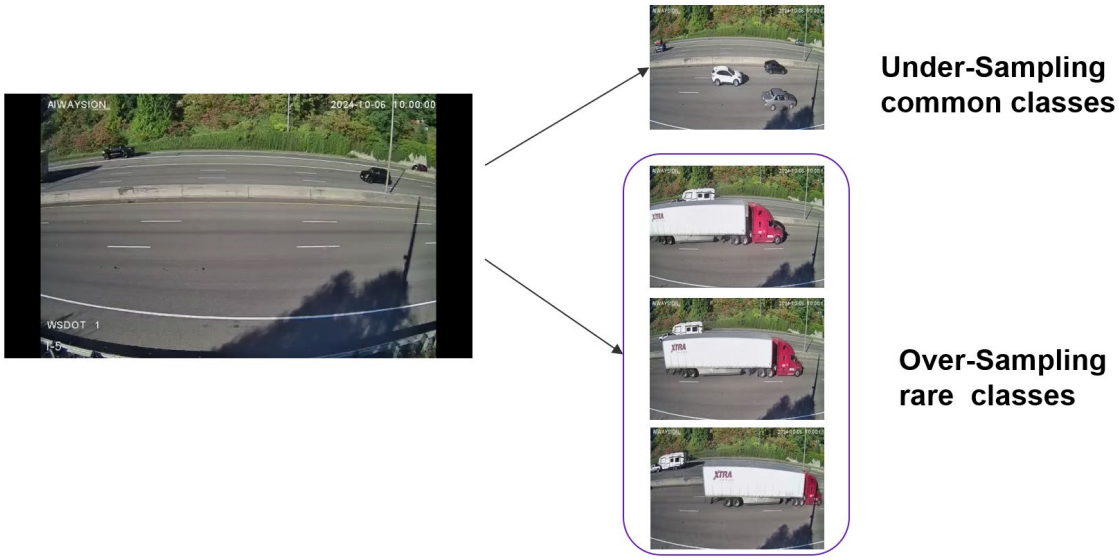


Figure 16. Vehicle classification sampling method.

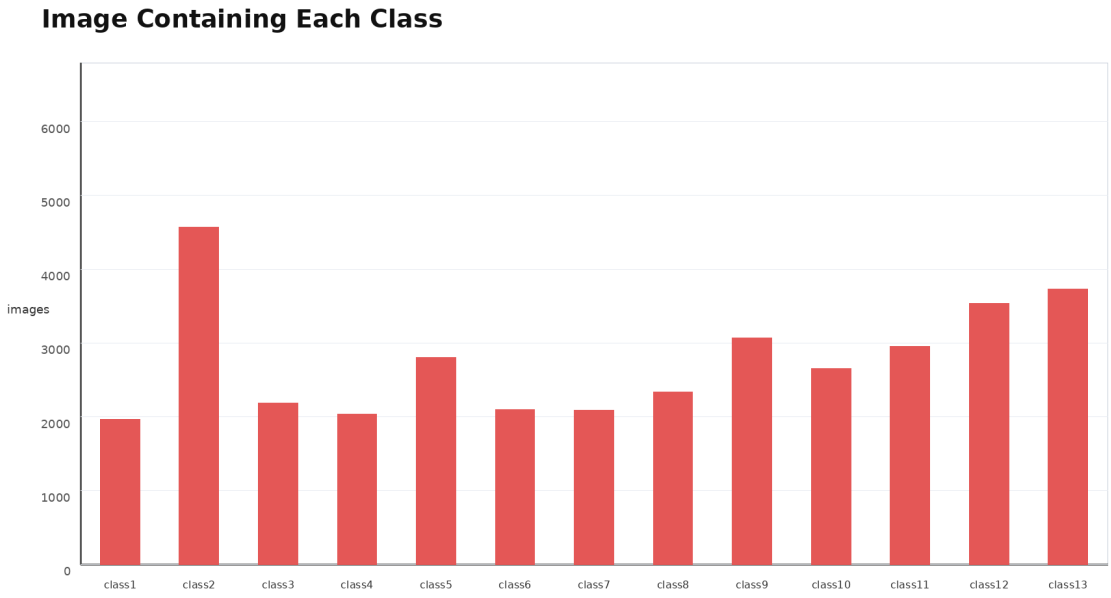


Figure 17 Image-level class distribution of the preprocessed dataset.

3.7 SUMMARY

Existing public vehicle detection datasets, such as Stanford Cars [5], CompCars [6], BDD100K [7], and UA-DETRAC [8], have supported general object detection research but fall short for transportation-specific applications. In particular, they do not adhere to the 13-bin vehicle classification scheme defined by the Federal Highway Administration (FHWA), which is widely used for traffic monitoring, infrastructure planning, and regulatory enforcement. Furthermore, few datasets provide comprehensive coverage across diverse environmental conditions—such as rain, nighttime, and other low-visibility scenarios—thereby limiting their generalizability in real-world deployments.

We address these limitations by introducing a new fine-grained vehicle detection dataset for transportation analytics. It adheres to the FHWA 13-bin vehicle classification standard and captures a wide range of environmental conditions. The dataset was collected using a fixed

roadside camera system installed along Washington state's I-5 highway which records vehicle activity under varied lighting and weather conditions throughout the day. To enhance spatial diversity and improve generalization, we additionally include 1,000 daytime images collected (as shown in Figure 18. Additional images from the Yakama Intersection) at the Yakama intersection in Washington state using a MUST sensor camera system We selected images that included rare vehicle classes to improve representation of underrepresented FHWA categories in the training set.. Although these supplementary images are limited to daytime scenes, they introduce variation in roadway geometry, traffic composition, and background context—factors that improve robustness across deployment sites. The complete dataset contains 7,000 images, distributed as follows: 4,000 in clear daytime, 1,500 during rain, and 1,500 at night. Each image is annotated with bounding boxes and class labels covering both common passenger vehicles and heavy-duty trucks, resulting in 20000 total vehicle instances. Here, a vehicle instance refers to a single annotated bounding box within an image. Because the dataset is constructed for frame-based detection, the same physical vehicle may appear in multiple frames and therefore be counted as multiple instances if it is visible across consecutive images. These repeated instances arise naturally from sequential video frames rather than from artificial duplication of identical images, and they often capture the same vehicle under different positions, scales, and viewpoints.

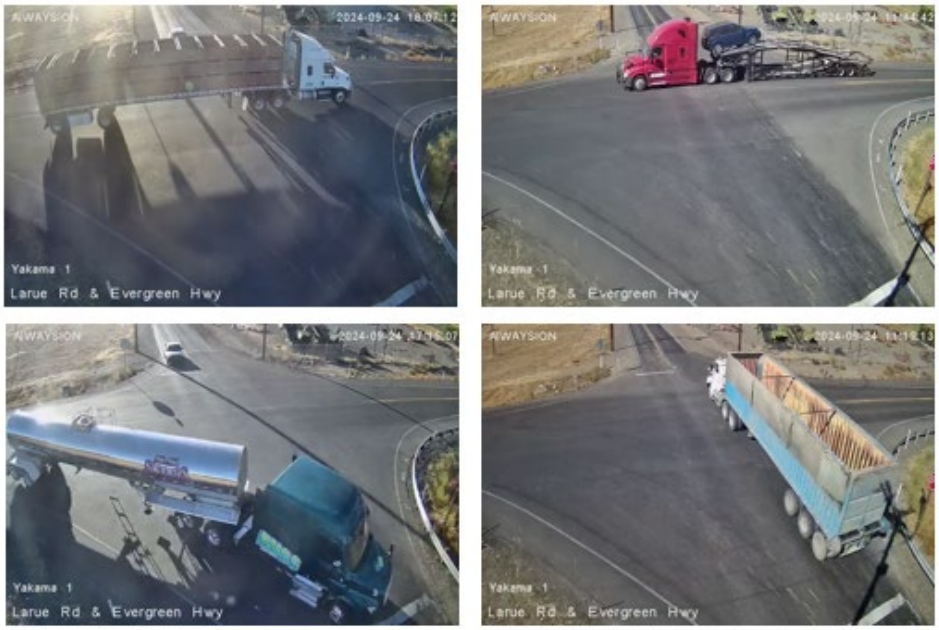


Figure 18. Additional images from the Yakama Intersection.

To ensure annotation quality, each image was manually labeled by trained annotators using a custom labeling tool. Bounding boxes and class labels were assigned based on the FHWA 13-bin taxonomy, guided by a visual reference to ensure consistency across annotators. All annotations were reviewed in a two-stage quality control process: first by a second annotator and then by a domain expert, who resolved ambiguities and corrected inconsistencies. This rigorous process ensures reliable fine-grained labels, particularly for visually similar classes such as single-unit and combination trucks. Table 3 shows the dataset comparison.

Table 3. Dataset comparison.

Dataset	FHWA 13-Bin	Diverse Weather	Day/Night
UA-DETRAC [8]	✗	✗	✗
BDD100K [7]	✗	✓	✓
CompCars [6]	✗	✗	✗
Ours	✓	✓	✓

4.0 LEARNING METHODS FOR TRAFFIC VEHICLE CLASSIFICATION

4.1 OVERVIEW

This chapter presents an overview of the end-to-end methodology used to develop the FHWA 13-bin vehicle classification system, spanning data collection, annotation, model training, and final deployment on the MUST edge sensor. Figure 19 provides a high-level illustration of the entire pipeline. Traffic data are first captured by the MUST sensor at the roadside and transmitted to a cloud-based back-end server for storage. Frames are then extracted from the recorded videos, and a curated subset of images with clear and unobstructed vehicle visibility is selected for annotation. Each vehicle within these images is labeled according to the FHWA 13-bin classification standard to ensure consistent and engineering-relevant ground truth. These annotated images form the foundation of the custom dataset used for model development.

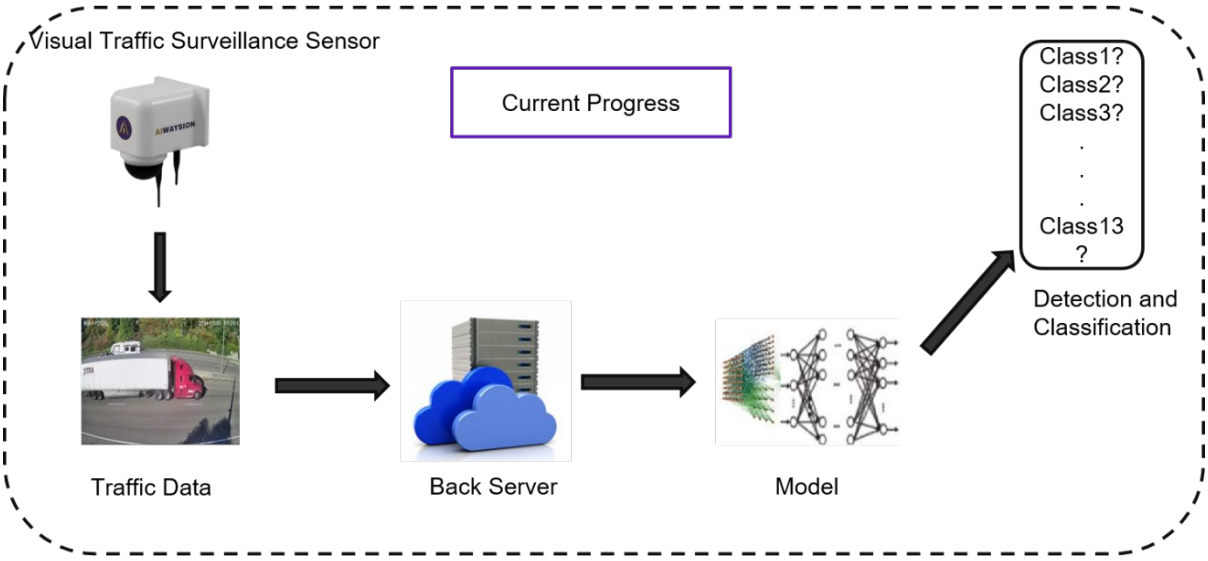


Figure 19. Offline training process.

Once the dataset has been prepared, it is used to train an advanced deep-learning–based detection and classification model. The training process focuses on enabling the model to learn subtle structural differences between FHWA vehicle classes, including axle counts, trailer

configurations, and single-unit versus combination-unit distinctions. Figure 19 illustrates the final system architecture, where the trained model is deployed directly onto the MUST edge device. By running inference locally on the sensor, the system is capable of performing real-time vehicle detection and FHWA 13-bin classification without the need for high-bandwidth communication or remote computing support. The edge deployment ensures fast processing, reduced latency, and greater resilience in field environments.

The following sections provide detailed descriptions of each stage of the pipeline, including the model training and evaluation approach.

4.2 YOLOV11 BASELINE ARCHITECTURE

YOLOv11 [27][29][30] was selected as the foundational architecture due to its favorable balance of detection accuracy, speed, and computational efficiency. As the latest iteration of the YOLO family, YOLOv11 introduces improvements in backbone design, feature fusion, and detection head configuration that make it particularly appropriate for edge-based traffic sensing.

The backbone of YOLOv11(as shown in Figure 20) builds on a CSP (Cross-Stage Partial) architecture, incorporating optimized residual blocks, improved gradient flow, and enhanced representational capacity. These design improvements allow the backbone to capture fine-grained structural details, such as wheel patterns and axle spacing, which are essential for FHWA classification. The neck employs a modified Feature Pyramid Network (FPN) combined with a Path Aggregation Network (PAN), enabling multi-scale fusion of low-level and high-level features. This fusion mechanism ensures that the model remains sensitive to small but critical structural cues, such as secondary trailers or small axle groupings, while also maintaining awareness of overall vehicle shape.

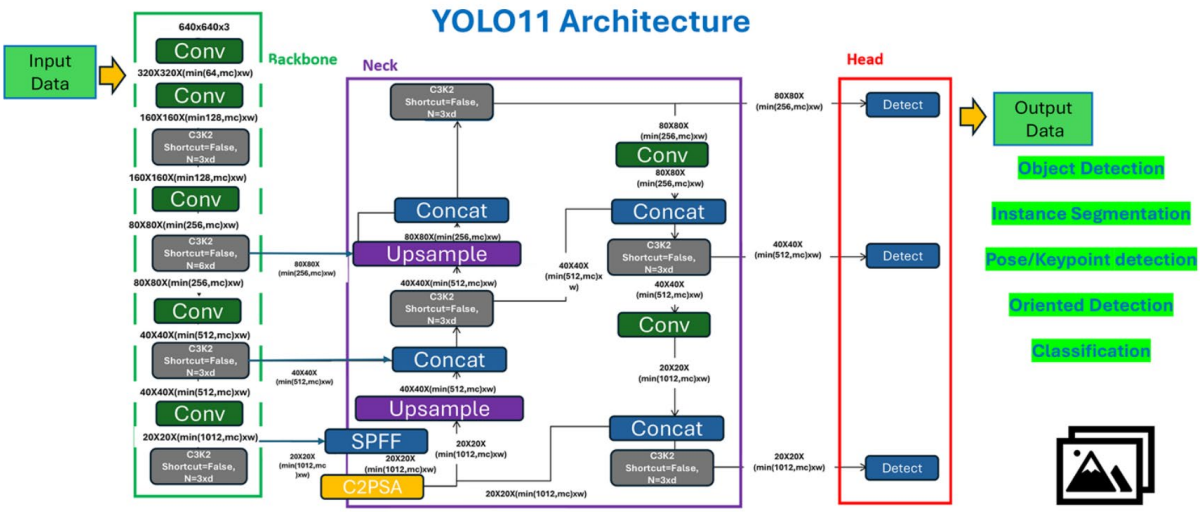


Figure 20. YOLO V11 structure.

The detection head of YOLOv11 adopts an anchor-free design with decoupled branches for classification and regression. The anchor-free approach reduces computational overhead and improves generalization because it avoids reliance on predetermined anchor box priors, which often fail to align with irregular vehicle geometries. The decoupled head further separates classification from bounding-box regression, allowing each task to optimize its features independently. This is particularly important for FHWA classification, where the bounding box must accurately frame the vehicle body while the classification branch must focus on axle-related details located near the lower portion of the vehicle.

Together, these architectural characteristics make YOLOv11 a strong baseline for FHWA 13-bin vehicle classification, offering robustness, high resolution feature extraction, and fast inference suitable for real-time performance on the MUST edge sensor.

4.3 DATA PREPROCESSING AND AUGMENTATION

Prior to training, all annotated images were resized and normalized to match the input requirements of YOLO V11 architecture. Each image was first scaled to the model's input

resolution while preserving aspect ratio through letterboxing. This approach prevents distortion of vehicle geometry, which is essential when learning axle spacing and trailer articulation.

A comprehensive data augmentation strategy was implemented to improve robustness under varied environmental and operational conditions. Augmentation techniques included geometric transformations such as random horizontal flipping, scaling, rotation, and cropping. Random horizontal flipping mirrors the image along the vertical axis, effectively reversing the left–right orientation of the scene. This operation allows the model to learn features that are invariant to camera viewpoints and lane positioning. For example, vehicles traveling in one direction (e.g., northbound) may appear as traveling in the opposite direction after flipping; however, this transformation increases the diversity of training samples and helps the model generalize to different camera placements and roadway layouts. Additional augmentations such as scaling, rotation, and cropping simulate variations in camera angle, zoom level, and partial occlusion, which are frequently encountered in roadside traffic videos. Photometric augmentations were also applied, including adjustments to brightness, contrast, saturation, and color temperature. These adjustments simulate diverse lighting conditions such as early morning shadows, midday glare, dusk transitions, and nighttime illumination. Additional augmentations targeted weather effects, including light blurring to simulate motion or rain-induced distortions, gamma correction to mimic low-light nighttime scenes, and haze-based manipulations to simulate fog or mist. These augmentations are critical because the MUST dataset covers a wide range of lighting and weather conditions, and the model must remain resilient across all scenarios. See examples in Figure 21.

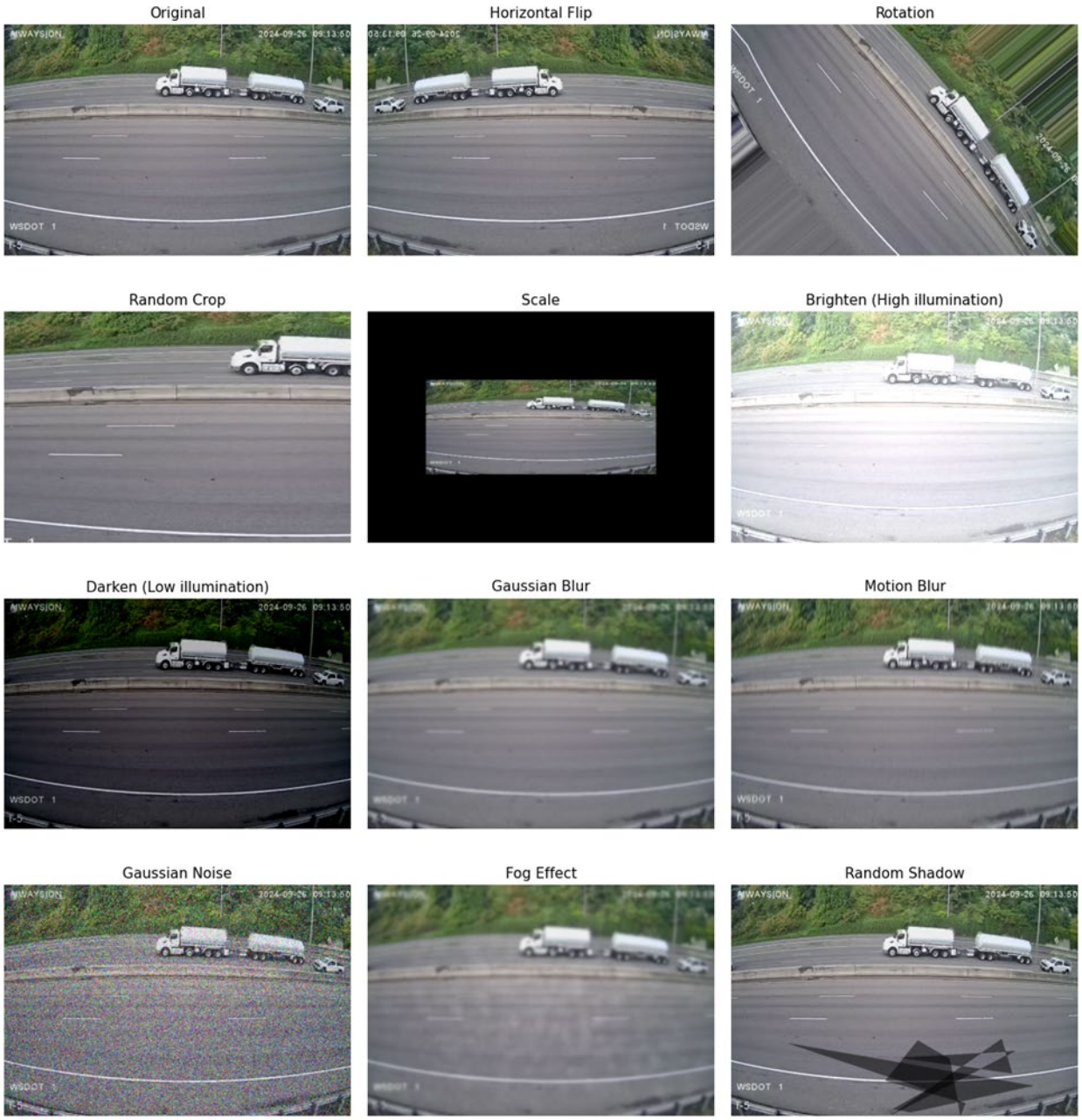


Figure 21. Examples of augmented samples.

Finally, a multi-view training strategy was employed to incorporate images from both camera viewpoints collected by the MUST sensor. One viewpoint provides a low-side profile that emphasizes wheels, axles, and trailer joints, while the oblique viewpoint provides broader geometric context. Training on both views simultaneously encourages the model to learn complementary cues, improving classification accuracy regardless of camera placement.

4.4 IMBALANCE DATASET HANDLING

The dataset collected from the roadside environment exhibits a highly imbalanced, long-tailed distribution. Most samples consist of Class 2 and Class 3 vehicles, while heavy-duty trucks and multi-trailer combinations—essential for FHWA compliance—are severely underrepresented. Without intervention, a model trained on such data would become biased toward dominant classes and fail to recognize rare but critical vehicle configurations.

To mitigate this challenge, we introduced a class-weighted loss function for the classification branch of YOLO V11. Each class was assigned a weight inversely proportional to its frequency, ensuring that rare classes contribute a proportionally larger gradient during back-propagation. This weighting scheme forces the model to focus on underrepresented classes, such as Classes 7 through 13, which include multi-axle single-unit trucks, tractor–trailers, and multi-trailer combinations.

In addition to loss weighting, we applied targeted sampling strategies. Over-sampling was used to increase the presence of rare vehicle classes in each training epoch. Rather than simply repeating identical samples, images from underrepresented classes were sampled more frequently and combined with data augmentation techniques to introduce variations in viewpoint, scale, and illumination. This strategy increases the effective diversity of training samples while maintaining class balance. Conversely, under-sampling was applied to extremely common classes to prevent them from dominating the training batches. The combination of these strategies produces a more stable gradient distribution and improves the model's ability to learn discriminative features across all FHWA categories.

These imbalance-mitigation techniques collectively ensure that the model maintains high performance even on rare, high-impact vehicle categories, which are crucial for engineering and freight analysis.

4.5 TRAINING CONFIGURATION AND OPTIMIZATION

The model was trained using an AdamW optimizer, selected for its stable convergence properties and ability to generalize well across nonlinear classification tasks. An initial learning rate was set based on the batch size and dynamically adjusted using a cosine annealing scheduler with warm-up iterations. The warm-up phase gradually increases the learning rate during the initial epochs, preventing early divergence and improving the stability of weight updates.

Training was conducted over multiple epochs until convergence criteria were met. Mixed-precision training was employed to accelerate computation and reduce GPU memory usage. Batch normalization layers were tuned using momentum adjustments to improve stability under augmented data distributions. Regularization strategies such as weight decay were applied to prevent overfitting, especially considering the small size of certain FHWA categories.

Throughout training, periodic validation was conducted using a held-out validation set consisting of 20 percent of the dataset. Validation metrics included class-wise accuracy, mean average precision (mAP), confusion matrix analysis, and per-class sensitivity. Early stopping mechanisms were implemented to terminate training when validation performance plateaued, ensuring optimal model efficiency.

4.6 EVALUATION AND VALIDATION STRATEGY

To evaluate the model's performance, we conducted comprehensive validation across all 13 FHWA categories. Each class was analyzed individually to ensure that the model learned the structural distinctions necessary for correct classification. Metrics included per-class accuracy,

mAP at multiple IoU thresholds, and interpretability assessments such as confusion matrices. Particular emphasis was placed on evaluating performance in rare classes, which often pose the greatest challenge.

Qualitative analyses were also performed to assess model robustness in complex scenarios. Model predictions were inspected on images involving occlusion, overlapping vehicles, multi-unit configurations, and visually similar vehicle classes. Special attention was paid to borderline cases such as tractor–trailer vs. multi-trailer combinations and five-axle vs. seven-axle trucks, where the classification depends on subtle visual cues.

This combination of quantitative and qualitative evaluation provides a holistic understanding of model performance and ensures its readiness for real-world deployment.

4.7 RESULTS AND SUMMARY

The training methodology described above significantly enhances YOLOv11's ability to perform FHWA 13-bin classification. Through improved data augmentation, multi-view training, long-tail imbalance mitigation, and optimized hyperparameters, the model demonstrates strong robustness across all vehicle types and environmental conditions. These targeted improvements enable the model to overcome common challenges such as occlusions, intra-class appearance variability, and fine-grained axle-based distinctions, laying the foundation for reliable edge deployment in the next phase of the project.

Table 4 presents the results of the current model's performance, demonstrating its effectiveness compared to other models. The YOLOv11-based model achieved a high average accuracy of 0.985 across all classes, outperforming the company with best result's accuracy of 0.96.

Table 4. Performance of the proposed model.

Class	**Accuracy**
Class1	0.995
Class2	0.981
Class3	0.982
Class4	0.959
Class5	0.991
Class6	0.984
Class7	0.995
Class8	0.968
Class9	0.977
Class10	0.988
Class11	0.995
Class12	0.995
Class13	0.995
All Classes	**0.985**
Company with the best result	0.96

In addition to the quantitative results, we present qualitative results showcasing the model's robust performance under various challenging scenarios. As shown in Figure 22, the model accurately detects and classifies objects even when trucks obstruct each other, either moving in the same direction or in different directions.

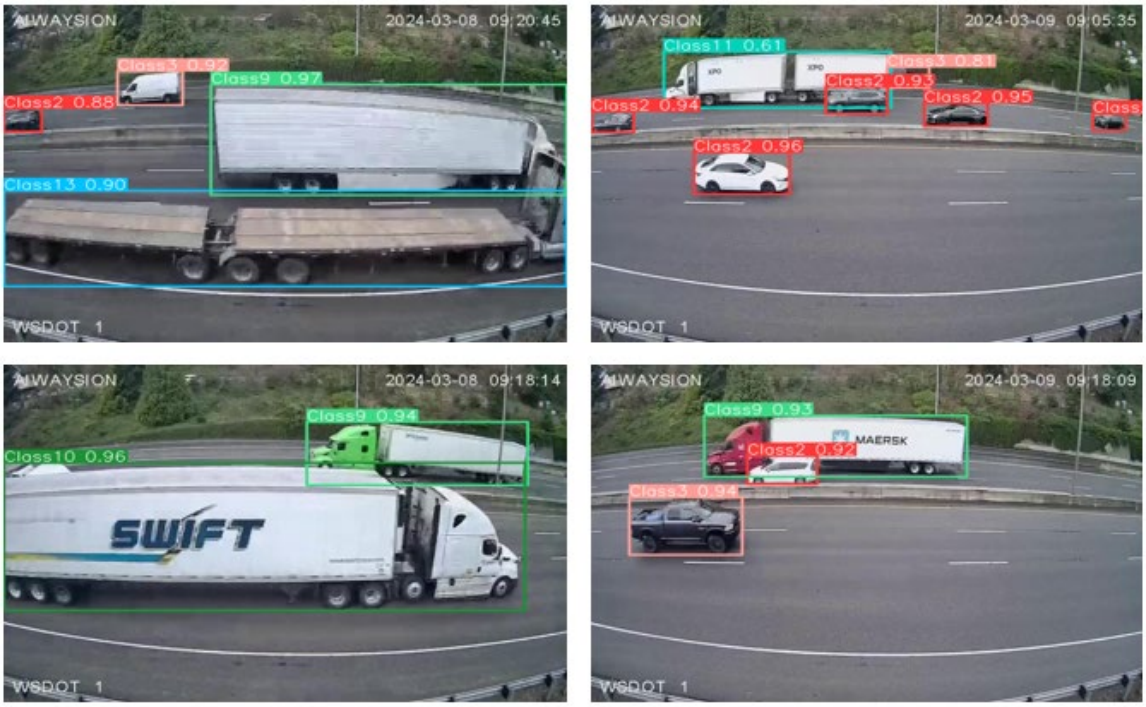


Figure 22. Robust performance in occlusion scenarios.

Figure 23 highlights the model's improved performance in identifying multi-unit vehicles. Earlier training iterations often misclassified multi-unit trucks as multiple separate vehicles. For example, a truck carrying multiple cars was previously detected as individual vehicles. However, the updated model accurately identifies such cases as a single unit, as shown in the first image. Similarly, in scenarios where a trailer is towing a tractor, the model now recognizes them as a unified entity rather than distinct components, demonstrating significant advancements in understanding and processing multi-unit vehicles.

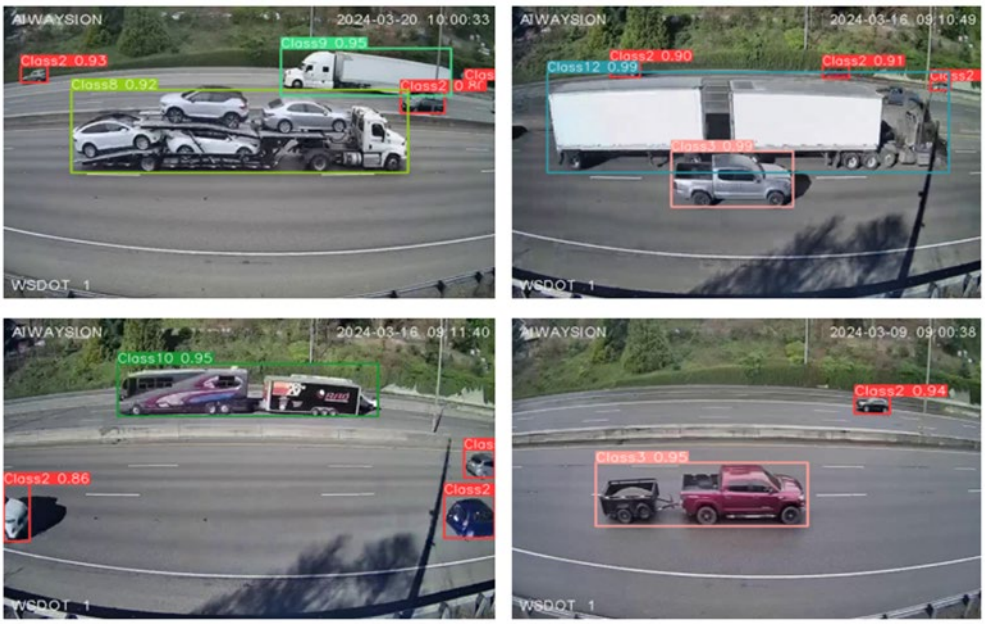


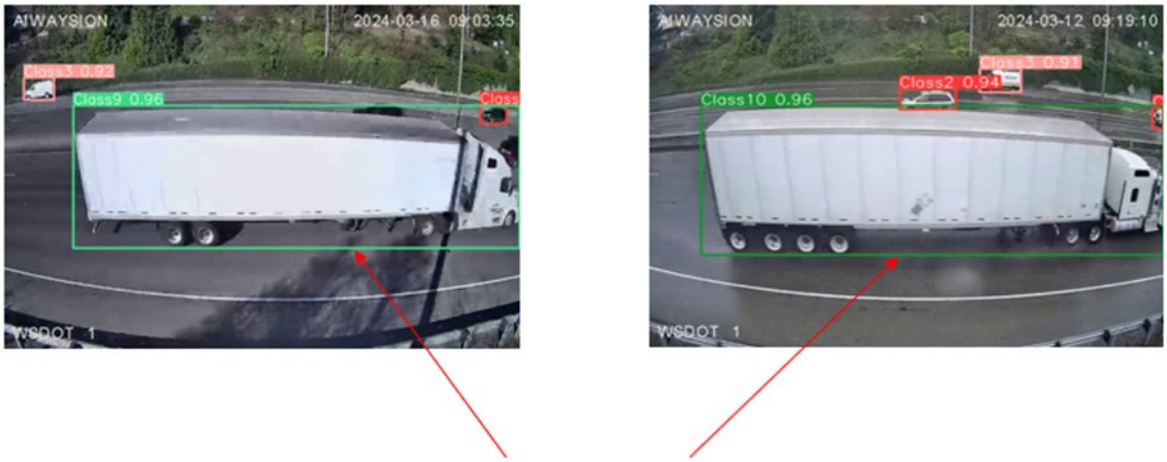
Figure 23. Robust performance with multi-unit vehicles.

Figure 24 showcases examples of vehicles from the same class (Class 10) but with distinct physical appearances. One image features a flatbed trailer, while the other depicts a box truck. Despite these differences, the model accurately classifies both as Class 10, demonstrating its robust ability to generalize across varying appearances within the same class. This performance underscores the model's reliability in diverse real-world scenarios.

Figure 25 illustrates a pair of images showing vehicles from different classes but with very similar appearances. In these examples, the truck on the left belongs to Class 9 due to having five axles, while the truck on the right is classified as Class 10 because it has seven axles. Despite their nearly identical appearances, with the key distinguishing feature being the number of axles, the model successfully classifies them correctly. This demonstrates the model's precision in detecting subtle differences and its ability to handle fine-grained classification challenges effectively.

Same Class But Different Appearance

Figure 24. Robust performance with different appearance within the same class.



Different Classes But Similar Appearances

Figure 25. Robust performance across similar appearances in different classes

5.0 MODEL DEPLOYMENT

5.1 OVERVIEW

Figure 26 illustrates the final product of our system: a complete end-to-end FHWA 13-bin vehicle classification solution deployed directly on the MUST edge sensor. After training the model with a carefully curated and annotated dataset, the next essential step is transitioning the model from a high-performance development environment into a lightweight, real-time inference engine suitable for embedded hardware. The MUST sensor is designed for field deployment in diverse roadway environments and is equipped with limited computational resources compared to cloud servers or GPUs. Therefore, the deployment process must ensure high efficiency, low latency, and reliable operation under continuous real-time workloads.

The deployed system performs all detection and classification tasks locally. This edge-based processing architecture allows the sensor to classify each detected vehicle into one of the 13 FHWA categories without relying on remote servers or high-bandwidth communication links. By keeping inference on the device, the system minimizes latency, enhances robustness in areas with weak connectivity, and ensures consistent real-time performance. The following subsections describe the deployment pipeline, model optimization procedures, and the real-time detection workflow that enables continuous traffic monitoring in the field.

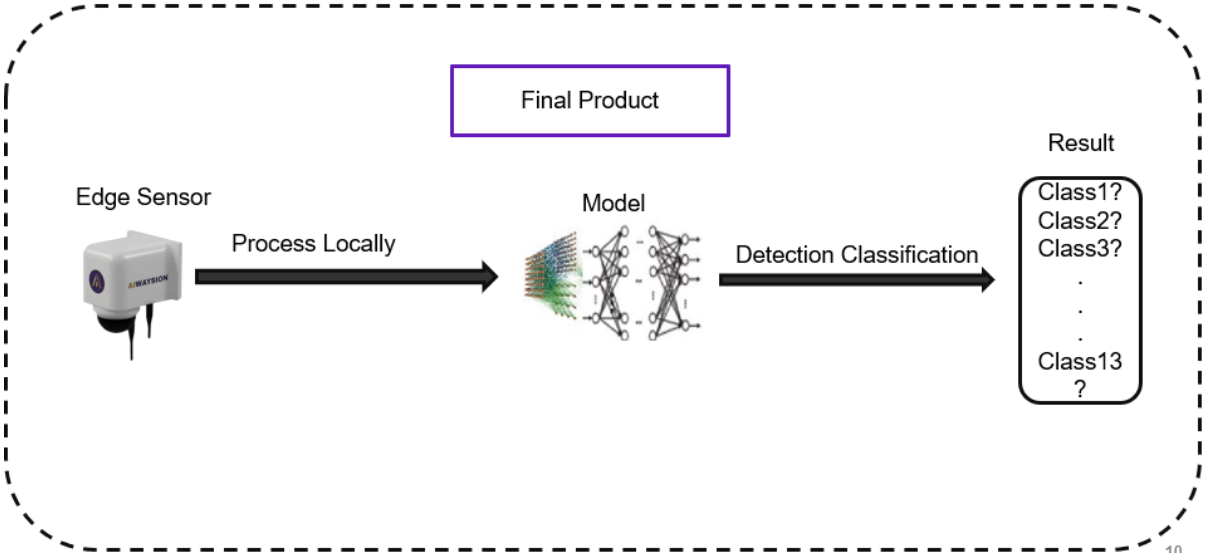


Figure 26. Final product.

5.2 DEPLOYMENT PIPELINE

The overall deployment workflow is shown in Figure 27. The process begins by exporting the trained model from the training environment. Deep-learning models are typically trained in frameworks such as PyTorch, which are optimized for flexibility and research experimentation but are not directly suited for efficient edge inference. To improve cross-platform compatibility and reduce runtime overhead, the trained YOLOv11-based model is first converted into the Open Neural Network Exchange (ONNX) format. ONNX serves as an intermediate, framework-agnostic representation that describes the model as a computation graph consisting of standardized operators. This allows the same model to be executed on a wide range of hardware and software platforms, including CPU-only embedded systems, inference engines, and custom accelerators commonly used in edge sensors. By separating model description from the original training framework, ONNX avoids the need to bundle the full training environment on the device and simplifies long-term maintenance.

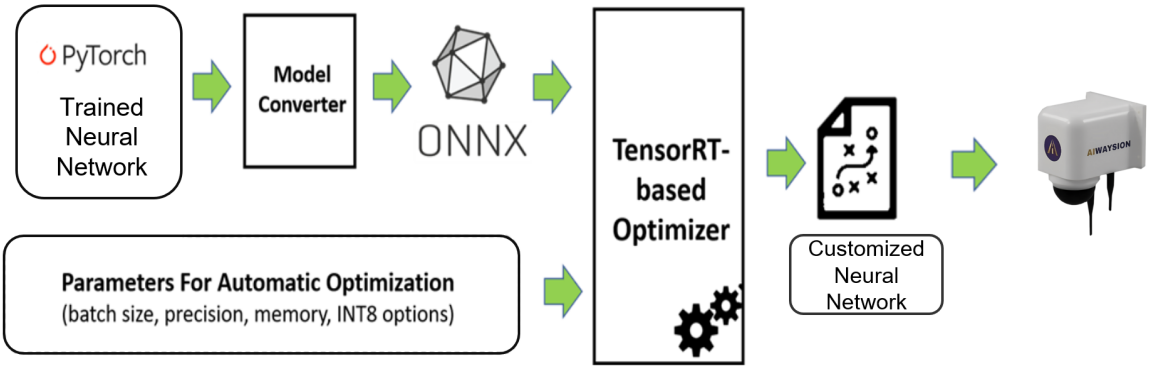


Figure 27. Deployment pipeline.

Once the model is converted into ONNX, it undergoes a comprehensive optimization process. In this stage, the computation graph is analyzed and simplified to remove unnecessary complexity. Graph simplification includes operations such as constant folding, where expressions that involve only constants are precomputed offline, and dead-node elimination, where nodes that do not contribute to the final output are removed. Operator fusion combines multiple sequential operations—such as a convolution followed by batch normalization and a non-linear activation—into a single fused operator. This reduces memory access, improves cache locality, and decreases the number of kernel launches, resulting in faster execution on the edge hardware. Memory layout refinement adjusts how tensors are stored in memory (for example, choosing between NCHW and NHWC formats) to match the preferred access patterns of the underlying processor. This reduces unnecessary data rearrangement and improves throughput. In parallel, profiling tools are used to measure inference time, CPU utilization, and memory consumption for each part of the graph. Profiling helps identify performance bottlenecks and guides targeted optimization so that the model can meet the real-time requirements of continuous traffic monitoring.

The final step in the pipeline involves packaging and installing the optimized model onto the MUST edge sensor. During this stage, the runtime environment is configured to ensure that the necessary inference engine, ONNX runtime libraries, and system dependencies are correctly installed and version-aligned. The model file, configuration parameters, and auxiliary scripts are then deployed to the device, often within a containerized or sandboxed environment to improve portability and reliability. After installation, the model is validated against on-device data streams by running test sequences and comparing the outputs with those obtained in the development environment. This validation step checks not only accuracy but also latency, stability under continuous operation, and resource usage. By following this structured deployment pipeline—from conversion, to optimization, to on-device validation—the model transitions smoothly from training to real-world operation while maintaining both accuracy and efficiency on the MUST edge sensor.

5.3 MODEL QUANTIZATION

One of the most important optimization techniques applied during deployment is model quantization. In the original training environment, the YOLOv11 model is represented using 32-bit floating-point numbers (FP32) for both weights and activations. This representation is convenient for training because it provides high numerical precision and stable gradients, but it is computationally expensive and memory intensive for edge devices. Quantization reduces the numerical precision of the model to formats such as 16-bit floating point (FP16) or 8-bit integer (INT8), thereby decreasing the memory footprint and accelerating arithmetic operations. For example, when converting from FP32 to INT8, each parameter requires only one quarter of the original storage, which directly translates into lower memory bandwidth requirements and faster data movement through the processor.

In the context of the MUST sensor, where compute, memory, and power are constrained, these reductions are especially important. Many embedded CPUs and NPUs provide hardware acceleration for low-precision integer operations but have limited support for high-precision floating-point math. By representing weights and activations in INT8 where possible, the model can take advantage of these low-precision instruction sets, resulting in significantly faster inference. At the same time, some parts of the network—such as the first and last layers or layers that are particularly sensitive to precision—may be kept in FP16 or FP32 to preserve numerical stability. This mixed-precision strategy allows the system to benefit from quantization while avoiding large drops in accuracy.

Two main quantization strategies were considered: post-training quantization (PTQ) and quantization-aware training (QAT). PTQ is applied after the model has been fully trained in FP32. In PTQ, the trained model is analyzed using a representative calibration dataset, and scaling factors are computed to map floating-point values to a lower-precision range (for example, mapping FP32 values into the INT8 range of –128 to 127). These scale factors can be computed per-tensor or per-channel, with per-channel quantization offering better accuracy for convolutional filters at the cost of slightly more complex computation. PTQ is simple to apply and does not require retraining, making it attractive when training resources or time are limited. However, PTQ can sometimes lead to noticeable accuracy loss if the model is highly sensitive to numerical perturbations.

Quantization-aware training takes a different approach by simulating quantization effects during the training process itself. In QAT, the forward pass includes “fake quantization” operations that mimic INT8 behavior while the weights are still stored and updated in floating-

point form during backpropagation. This allows the model to adapt to the quantization noise and learn weight configurations that are inherently more robust to low-precision representation. QAT typically yields better accuracy than pure PTQ, particularly for challenging tasks or fine-grained classifications such as FHWA 13-bin vehicle categories. The trade-off is that QAT requires additional training cycles and more careful tuning of the training process.

For this project, we evaluated both PTQ and QAT using a held-out validation set that reflects the long-tailed distribution of FHWA classes. Initial experiments with full INT8 post-training quantization showed substantial gains in inference speed but also revealed slight degradations in classification accuracy for some rare heavy-truck classes. To mitigate this, we adopted a selective quantization strategy. Layers that contribute most to final decisions—for example, the detection head and certain intermediate feature maps critical for axle and trailer representation—were quantized using higher precision (FP16 or kept as FP32), while the majority of convolutional layers in the backbone and neck were converted to INT8. This selective approach retains the speed benefits of low-precision arithmetic where the model is more tolerant to approximation, while preserving higher precision in components that are most sensitive to small changes in activation values.

In addition to layer-wise decisions, we also tuned the calibration procedure used in PTQ. A diverse calibration set was constructed to include vehicles from all FHWA classes, especially multi-axle trucks and multi-trailer combinations. This ensured that the quantization scale factors were derived from a value distribution representative of real deployment conditions, rather than being dominated by the much more frequent passenger car and light truck classes. Calibration experiments were iteratively refined by examining per-class accuracy and confusion patterns before and after quantization. This iterative process helped identify specific layers and activation

ranges where quantization caused problematic shifts, allowing us to fine-tune quantization parameters or exclude those layers from INT8 conversion.

Overall, the final quantization configuration represents a compromise between speed and accuracy tailored to the constraints of the MUST sensor. The resulting model is significantly smaller and faster than its original FP32 counterpart, enabling real-time inference on the edge device, while maintaining high classification performance across all 13 FHWA classes. This careful combination of PTQ, selective mixed precision, and calibration ensures that quantization serves as an enabler of real-time field deployment rather than a source of unacceptable accuracy loss.

5.4 PRUNING OPTIMIZATION

In addition to quantization, model pruning plays a critical role in reducing computational load and improving real-time performance on the MUST sensor. Pruning refers to the process of removing redundant or low-contributing parameters from the neural network while maintaining overall accuracy. Modern deep-learning models often contain significant structural redundancy, where many channels, filters, or weights contribute minimally to the final prediction. By removing these components, the model becomes smaller, faster, and more energy-efficient—properties that are essential for resource-limited edge deployments.

In this project, we applied structured pruning, which removes entire convolutional channels, filters, or blocks rather than individual weights. Structured pruning is preferred over unstructured pruning because it produces compact model architectures that map efficiently onto embedded hardware. For example, removing entire channels from a convolution layer not only reduces the number of multiply–accumulate (MAC) operations but also decreases memory

access, which is often a major bottleneck in edge computation. Structured pruning also avoids creating sparse weight matrices, which many edge processors handle inefficiently.

The pruning process begins by analyzing the weight distributions and activation statistics of each layer. Layers with many near-zero activations or low-magnitude filter weights often indicate redundancy. We used a combination of magnitude-based pruning (removing filters with small L1/L2 norms) and activation-based pruning (removing channels that consistently produce weak or uninformative activations during training). These statistics were gathered from multiple datasets, including rare heavy-vehicle classes, to ensure that pruning decisions did not disproportionately affect performance on underrepresented categories.

After identifying candidate channels or filters for removal, we progressively pruned small portions of the network in multiple rounds. This incremental approach helps avoid sudden drops in accuracy and allows the model to recover through fine-tuning. After each pruning stage, the model was retrained for several epochs to allow remaining weights to compensate for the removed parameters. This step was crucial for FHWA 13-bin classification, where small structural cues—such as axle spacing or trailer articulation—are easily lost if the network becomes overly compact.

The final pruned model showed a significant reduction in computational complexity (measured in FLOPs) and memory usage, without compromising classification accuracy. Combined with quantization, pruning contributed to a noticeable reduction in inference time per frame, enabling the MUST sensor to consistently maintain real-time performance even during periods of high traffic volume. The pruned and quantized structure represents a highly optimized version of the original YOLOv11, ideally suited for continuous roadside deployment.

5.5 RUNTIME INTEGRATION

Effective deployment on the MUST edge sensor requires more than model optimization—it requires careful runtime integration, ensuring that the model interacts efficiently with the device's computational resources, memory hierarchy, and video-processing pipeline. Edge systems often face performance bottlenecks not from model complexity alone, but from input/output operations, buffering delays, or inefficient threading. Therefore, a significant part of deployment involves tuning the system-level components that support real-time inference.

The MUST sensor uses a lightweight, embedded inference engine capable of executing ONNX models with minimal overhead. During integration, we configured the inference runtime to use pre-allocated memory pools, preventing dynamic memory allocation during inference. This eliminates memory fragmentation and reduces latency spikes, both of which are critical for maintaining a stable frame rate. Pre-allocation also ensures deterministic behavior under continuous operation, which is necessary for embedded deployments where memory resources are limited.

Threading configurations were also carefully tuned. Because the MUST sensor uses a multi-core CPU but lacks a dedicated GPU, inference speed is highly dependent on efficient CPU parallelism. We experimented with different combinations of thread pools, including dedicated threads for model execution, video decoding, and tracking. A pipeline-parallel architecture was adopted, where the streamer, detector, and tracker run concurrently rather than sequentially. This prevents bottlenecks and ensures that the detector always receives frames as soon as they are available.

To optimize data flow, we implemented zero-copy memory sharing between modules whenever possible. For example, frames decoded by the video streamer are passed to the detector

without duplicating the pixel buffer, significantly reducing memory bandwidth consumption. Similarly, the detector outputs (bounding boxes and class labels) are passed directly to the tracker using in-memory queues with lock-free synchronization, improving throughput and reducing inter-process delays.

Profiling tools were used extensively to measure system-level performance, including per-frame processing time, CPU utilization, memory footprint, and queue backlogs. These metrics helped identify subtle bottlenecks such as inefficient image resizing operations, sub-optimal tensor layouts, or thread contention. Targeted optimizations—such as switching to hardware-accelerated resize operations or adjusting batch sizes—were then applied. Finally, the entire pipeline was validated under long-duration stress tests to ensure that the system could operate continuously for hours without memory leaks, thread deadlocks, or thermal throttling.

Through careful runtime integration and system-level optimization (as shown in Figure 28) , the MUST sensor is now capable of running the pruned and quantized YOLOv11 model in real time, processing each frame as it arrives and delivering accurate FHWA 13-bin classifications directly on the roadside device. This combination of model-level and system-level optimization ensures robust and reliable performance in diverse traffic environments.

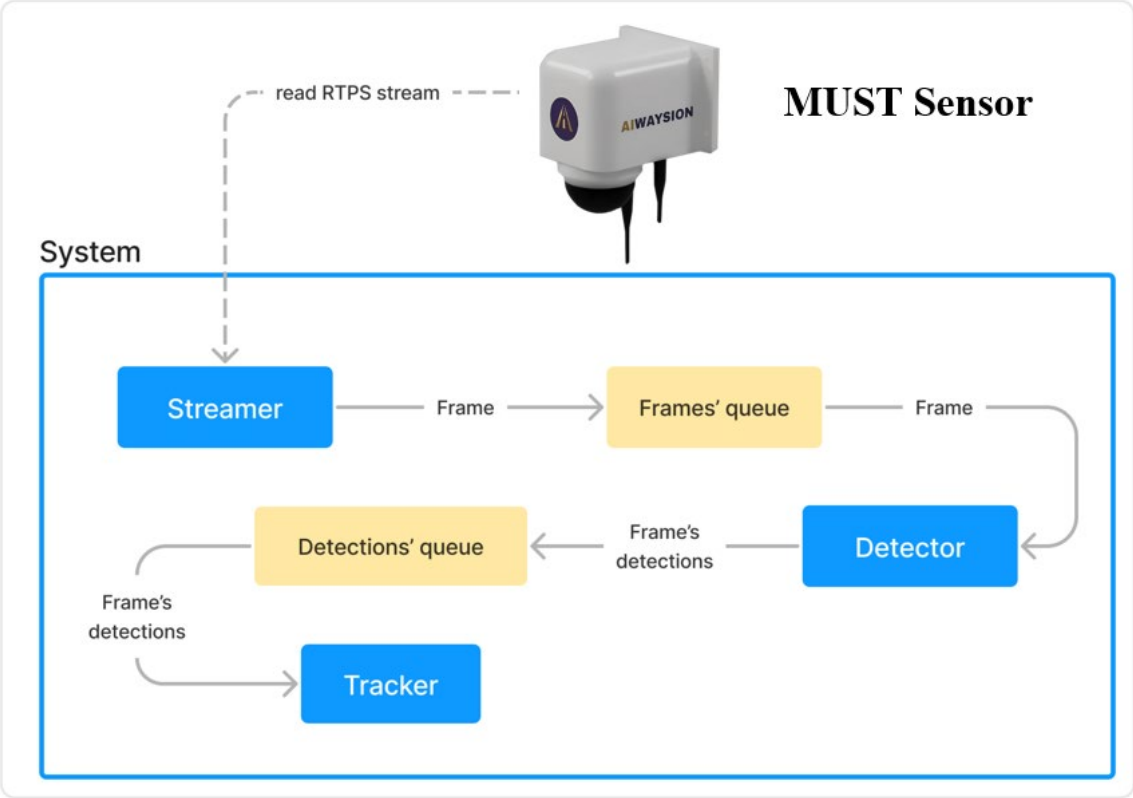


Figure 28. Video processing.

5.6 SUMMARY

In summary, the deployment of the FHWA 13-bin vehicle classification model onto the MUST edge sensor required a comprehensive integration of model-level and system-level optimization techniques. Starting from the conversion of the trained YOLOv11 model into the ONNX format, the deployment pipeline systematically incorporated graph simplification, operator fusion, memory-layout refinement, and performance profiling to prepare the model for efficient execution on embedded hardware. Advanced compression methods—including structured pruning and mixed-precision quantization—further reduced computational complexity

and memory footprint while preserving the fine-grained accuracy required for axle- and trailer-based classification.

Beyond model optimization, careful attention was devoted to runtime integration on the MUST device. The deployment included configuring inference engines, tuning threading strategies, optimizing memory usage, and refining dataflow between video acquisition, detection, and tracking modules. These system-level improvements ensured stable, low-latency performance and enabled the sensor to deliver real-time classification results under continuous roadside operation.

Together, these efforts produced a highly efficient, pruned, and quantized YOLOv11 model capable of providing reliable FHWA 13-bin classification directly on the MUST edge sensor without reliance on cloud computing. The resulting system is robust, fast, and field-ready, laying the technical foundation for large-scale, cost-effective, and high-resolution traffic monitoring across the Washington state highway network.

6.0 CHALLENGES IN ADVERSE VISIBILITY

6.1 OVERVIEW

Traffic camera systems frequently operate under varying environmental and lighting conditions, many of which introduce significant visual degradation that impacts both human interpretation and machine-based vehicle classification. Among these, nighttime low-light environments and rainy weather represent two of the most challenging scenarios for reliable automated perception. These conditions reduce overall visibility, distort structural features, and introduce artifacts that directly affect the performance of image-based detection and classification algorithms.

6.1.1 Nighttime Low-Light Challenges

Nighttime scenes are fundamentally constrained by insufficient ambient illumination, causing key vehicle features—such as contours, edges, color patterns, and reflective surfaces—to become faint or invisible. The reduced signal-to-noise ratio leads to:

- amplified sensor noise,
- low contrast across the entire frame,
- color distortion, and
- motion blur due to longer exposure times.

Additionally, headlights introduce glare, halos, and overexposed regions that obscure nearby vehicles and pavement markings. As a result, traditional classification models trained on well-lit daytime imagery often fail to generalize to nighttime traffic conditions.

6.1.2 Rainy Weather Challenges

Rain presents a different but equally difficult set of problems. Raindrops adhering to the camera lens act as physical obstacles that refract and distort incoming light, creating blurred or

warped regions. Water splashes from moving vehicles and reflective puddles generate transient occlusions. Heavy rainfall also reduces ambient visibility and increases atmospheric scattering, resulting in:

- softened edges,
- blurred vehicle outlines,
- strong reflections on wet pavement, and
- inconsistent lighting across the scene.

These distortions degrade both the spatial and temporal coherence of video frames, making vehicle shapes harder to distinguish and reducing detection accuracy.

6.1.3 Proposed Solution

Since our system is intended for statewide deployment, robustness under such adverse conditions is not optional—it is essential for ensuring accurate identification of large trucks, multi-trailer vehicles, passenger cars, and all other FHWA vehicle classes.

To address these challenges, we developed a multi-stage strategy designed to enhance visibility, expand data diversity, and strengthen the model's adaptability. First, we optimized camera-level settings such as exposure and shutter speed to improve raw nighttime capture quality. Next, we incorporated state-of-the-art image enhancement algorithms—including low-light enhancement and deweathering methods—to recover scene details lost due to darkness, glare, motion blur, or rain-induced distortion. To overcome the scarcity of nighttime data, we generated synthetic nighttime imagery via CycleGAN[42]-based day-to-night translation, significantly expanding the training distribution. These enhanced and augmented datasets were then used to fine-tune the classification model, enabling it to better handle visually degraded scenes. Finally, we introduced a unified mixture-of-experts (MoE) framework that dynamically

routes each input to a specialized expert branch—daytime, nighttime, or rainy weather—ensuring that the system adapts in real time to changing conditions. Together, these components form a robust, scalable solution capable of delivering consistent performance across diverse and challenging environmental scenarios.

6.2 EXISTING METHODS

Improving image quality under low-light and rainy-weather conditions has been an active research topic for many years, particularly in fields such as surveillance, autonomous driving, and transportation monitoring. These methods vary considerably in terms of complexity, underlying principles, and the types of degradations they address. Broadly, the existing approaches can be grouped into hardware-oriented methods, algorithmic enhancement techniques, and data-driven domain adaptation strategies. While these categories are conceptually distinct, they are often used together in practical systems.

6.2.1 Hardware-Oriented Methods

Early traffic monitoring systems primarily relied on hardware-level adjustments to cope with challenging visibility. The most common approach is to modify camera exposure and shutter speed to allow more light into the sensor. In nighttime settings, longer exposure times brighten the scene but introduce motion blur when vehicles travel at highway speeds. Increasing sensor gain can also improve brightness, but at the cost of amplified noise and reduced sharpness—issues that are especially problematic for classification tasks relying on fine-grained details.

Some camera installations also employ optical solutions, such as hydrophobic coatings or protective visors, to prevent raindrops from sticking to the lens. While these measures help reduce lens obstruction, they cannot fully eliminate distortions caused by heavy rain or water

streaks. More advanced systems use infrared or near-infrared illumination to compensate for low-light environments. These systems provide additional visibility without increasing motion blur, but they require specialized sensors and are significantly more expensive, making them impractical for large-scale deployment.

6.2.2 Classical Digital Enhancement Techniques

Before deep learning became mainstream, image enhancement largely relied on handcrafted algorithms designed to manipulate pixel intensity distributions. Histogram equalization and gamma correction are two common examples. These methods can quickly boost contrast and brightness, but they often over-amplify noise in very dark images. Retinex-based techniques—which were inspired by human visual perception—attempt to separate illumination from reflectance. While more sophisticated than global adjustments, classical Retinex algorithms struggle when illumination is extremely uneven or when noise dominates the scene.

These traditional methods are computationally lightweight and can operate in real time, but their effectiveness deteriorates significantly in the presence of complex artifacts such as headlight glare, heavy noise, or raindrop occlusions.

6.2.3 Deep Learning-Based Low-Light Enhancement

Recent advancements[33][37][38] in deep learning have led to a new generation of image enhancement models that are far more capable in visually degraded conditions. Retinex-inspired deep networks (e.g., RetinexFormer[32]) combine the interpretability of classical decomposition with the flexibility of learned illumination maps. These models tend to preserve structural information well, making them suitable for traffic imagery where lane markings, vehicle edges, and shadow boundaries are important.

Another group of methods—represented by Zero-DCE[31] and Zero-DCE++[42]—learns pixel-wise adjustment curves directly from data. These models require no paired training examples and are extremely fast, making them appealing for real-time systems. However, their simplicity sometimes limits their ability to recover details when the original signal is extremely weak.

GAN-based approaches such as EnlightenGAN [35]aim to produce more natural-looking enhancements by training a generator and discriminator jointly. These models often improve visual aesthetics, but they may introduce subtle artifacts that can interfere with downstream detection tasks if not carefully tuned.

More recently, Transformer-based models such as Restormer[32] and LLFormer [41] have demonstrated state-of-the-art performance in low-light restoration. Their attention mechanisms allow them to capture long-range dependencies, making them particularly strong at handling situations where the illumination varies significantly across the scene. Although these models produce high-quality results, they generally require more computational resources[36][39].

6.2.4 Rain and Weather-Specific Restoration

Low-light conditions are not the only challenge; rainy weather introduces a distinct set of degradations that are fundamentally different from illumination problems. Rain streak removal algorithms attempt to model streaks as high-frequency noise patterns and suppress them using multi-scale CNNs or recurrent frameworks. However, these methods do not address raindrops on the lens—an artifact that behaves more like a refractive obstruction than a noise component. Raindrop removal networks, which are often GAN-based, attempt to reconstruct occluded

content using contextual cues from surrounding regions. Their performance varies depending on the shape and size of the droplets.

In heavy rain or fog, visibility loss is dominated by scattering effects. Dehazing algorithms model the scene as a combination of direct transmission and atmospheric light. While originally designed for fog and haze, these methods can partially restore clarity in rainy scenes where scattering effects are significant.

Although each weather-specific method addresses a different type of degradation, no single algorithm is capable of handling all challenges simultaneously—particularly when rain and low-light conditions occur together.

6.2.5 Domian Adaption and Data-Driven Solutions

In recent years, domain adaptation has become a powerful tool for dealing with nighttime and rainy-weather imagery. Instead of relying solely on enhancement algorithms, data-driven approaches attempt to adapt the model itself to the target domain. One widely adopted strategy involves generating synthetic nighttime images using unpaired image-to-image translation techniques such as CycleGAN [42]. These models can transform daytime footage into realistic nighttime scenes while preserving the semantic content of the original images. This dramatically increases data diversity and reduces the need for large-scale nighttime data collection.

Other self-supervised or zero-reference methods, such as SCI, learn enhancement mappings without requiring clean reference images. These models rely on intrinsic consistency or structural priors to guide learning, making them more flexible in real-world deployments where labeled data is scarce.

Finally, fine-tuning the detection model directly on a mixture of real, enhanced, and synthetic images often yields the best results. By exposing the model to the full spectrum of

degraded conditions, it becomes more resilient to variations in illumination, weather, and scene composition.

6.3 RESULTS FROM EXISTING ENHANCEMENT AND ADAPTATION METHODS

To better understand the limitations of nighttime traffic footage and to identify the most effective strategies for improving visual quality, we evaluated a broad set of existing image enhancement and domain adaptation techniques. These methods were tested on our traffic video dataset, and their effects on downstream vehicle detection and classification were assessed in detail.

6.3.1 Nighttime Image Enhancement

Nighttime scenes in our dataset typically suffer from extremely low contrast and significant sensor noise, which obscure key structural cues such as vehicle edges, contours, and lane boundaries. To address these issues, we expanded our enhancement pipeline and tested eleven state-of-the-art low-light enhancement methods representing a wide range of algorithmic principles. These include curve-based enhancement (Zero-DCE++[8]), Retinex-inspired decomposition models (Kind++ [34]and RSFNet [37]), GAN-based illumination correction (EnlightenGAN[35]), recursive and flow-based models (DRBN[9] and LLFlow[40]), and several recent Transformer architectures such as Restormer[32] and LLFormer [41].

In practice, we observed that these models produce noticeably different types of enhancement. Transformer-based approaches—particularly Restormer and LLFormer—were the most consistent in recovering vehicle outlines and suppressing heavy noise, especially in scenes with severe illumination imbalance or strong color distortion. Lighter-weight models like Zero-DCE++ offered significant computational efficiency and real-time potential but occasionally failed to restore details in the darkest regions, leading to overly smooth textures or incomplete structural recovery. Self-supervised techniques such as Zero-IG and SCI produced stable results

without requiring paired data, making them suitable for large-scale deployment where obtaining reference images is impractical.

A visual comparison of these enhancement methods applied to our real nighttime footage is shown in Figure 29. Overall, enhanced images exhibited clearer lane boundaries, brighter vehicle surfaces, and improved contrast—features that directly benefited our downstream classification task.

However, enhancement also introduced several challenges. In some cases, aggressive illumination boosting amplified sensor noise, especially in low-photon areas, resulting in grainy textures or color blotches. Retinex-based models occasionally produced halo artifacts around headlights or reflective surfaces, and GAN-based methods sometimes altered the scene's global color distribution, which could mislead feature extractors trained on unaltered data. Furthermore, inconsistencies across enhancement methods may introduce domain shifts if not handled carefully during training. These limitations highlight the importance of choosing enhancement algorithms that balance detail recovery with noise suppression, and of integrating robust preprocessing strategies to prevent the propagation of enhancement-induced artifacts into downstream perception modules.

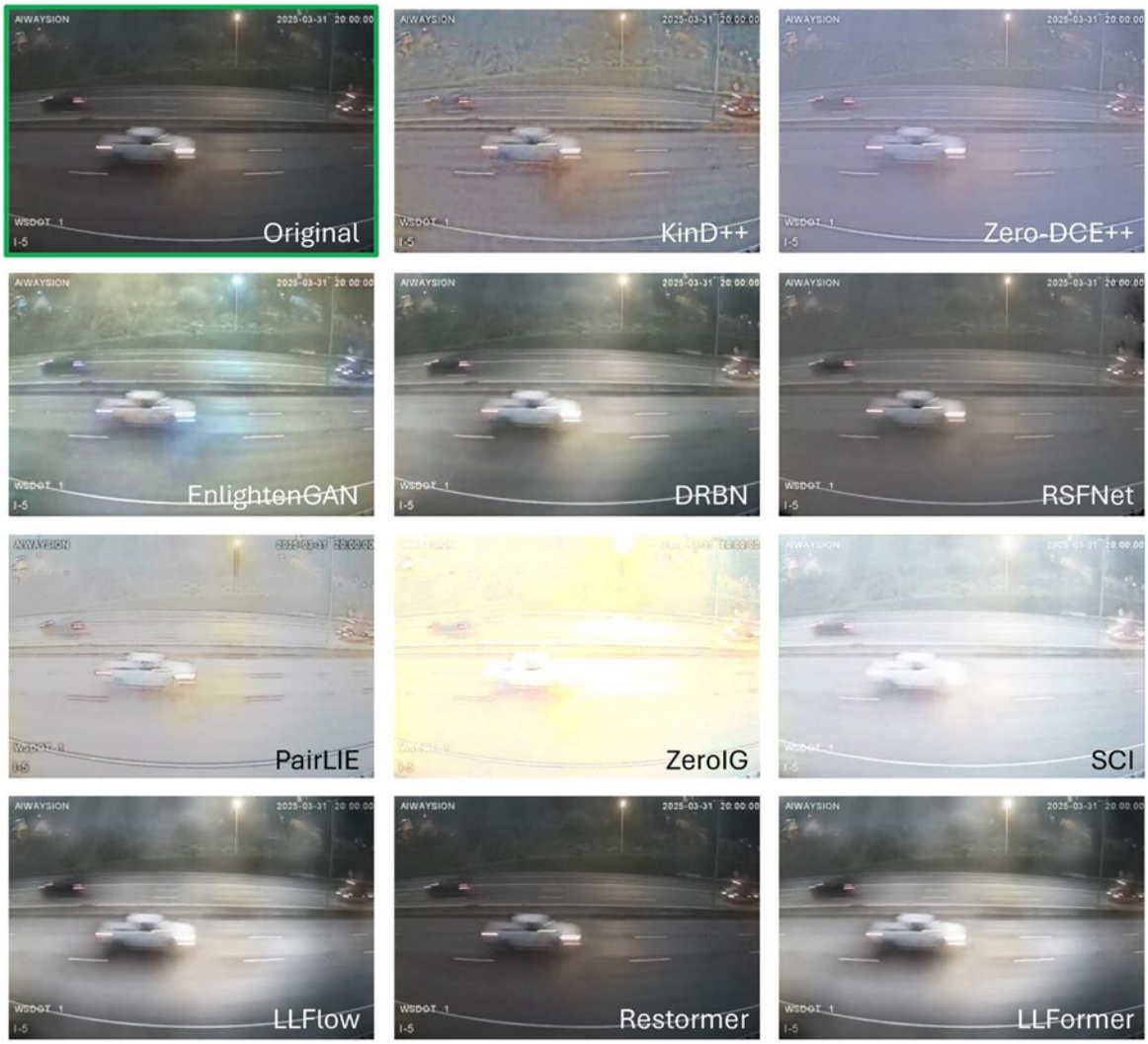


Figure 29. Comparison of nighttime image enhancement models on real-world traffic footage.

6.3.2 Day-to-Night Data Augmentation

Although enhancement improves nighttime imagery, the performance of a classifier also depends on the diversity of nighttime samples available during training. Because nighttime traffic data is inherently limited—particularly for rare FHWA categories and uncommon roadway configurations—we adopted a domain translation strategy to enrich the dataset. Specifically, we used CycleGAN to generate synthetic nighttime images from existing daytime

videos. This approach leverages unpaired image-to-image translation, allowing the model to learn nighttime appearance characteristics without requiring aligned day–night image pairs.

CycleGAN preserves the geometric and semantic structure of the original daytime scenes—vehicle locations, lane configurations, object boundaries—while introducing realistic nighttime cues such as darker global luminance, localized illumination sources, sharper headlight falloff, color shifts, and sensor-like noise. These properties are crucial for preserving label consistency: the synthetic nighttime frames remain perfectly aligned with the original annotations, ensuring that all bounding boxes and category labels remain valid after translation.

The generated images significantly broadened the variation of nighttime appearances in our dataset. As shown in Figure 30, the synthetic outputs capture many visual properties characteristic of real nighttime traffic, including uneven illumination across the road surface, glare around headlights, and muted color contrast. Importantly, CycleGAN[42] also introduces natural diversity in environmental conditions, producing outputs that resemble twilight, early night, and late-night scenes depending on the input context. This variability makes them well suited for training a classifier that must operate across a wide range of lighting conditions.

Figure 30. Example of day-to-night translation using CycleGAN. The nighttime images are generated by translating daytime images using the CycleGAN model.

When combined with real nighttime samples and enhanced images, the augmented dataset enabled the classification model to better generalize across lighting variations and

headlight conditions. The classifier became more robust to shadows, reflections, and inconsistent illumination that would otherwise hinder recognition.

However, CycleGAN-based augmentation also introduces certain limitations. Because the model optimizes primarily for visual realism rather than strict physical accuracy, some translations may introduce subtle distortions—such as exaggerated color shifts, artificial textures, or over-smoothing in low-light regions. In a few cases, small objects or fine vehicle details may become partially blurred or attenuated, which could influence classifier learning if not carefully controlled. Additionally, generative hallucinations (e.g., faint artifacts or inconsistent lighting patterns) occasionally appear in low-contrast areas. Despite these imperfections, the benefits of increased nighttime diversity outweighed these drawbacks when supported by real and enhanced imagery during training.

Overall, the integration of synthetic nighttime data substantially strengthened the model's resilience to nighttime variability, contributing to smoother performance across both common and challenging illumination scenarios.

6.3.3 Fine-Tuning for Nighttime Domain Adaptation

After expanding the dataset with both real and synthetic nighttime images, we fine-tuned our vehicle classification model to better handle the unique challenges presented by low-light conditions. This fine-tuning process involved adapting a pretrained baseline model—originally trained on well-lit daytime images—to our enhanced low-light dataset. Because the feature distribution of nighttime imagery differs substantially from that of daytime traffic scenes, the model required careful retraining rather than simple direct transfer. We refined the training pipeline by adjusting learning rate schedules, applying targeted data augmentations, and incorporating loss-balancing strategies to prevent overfitting to high-contrast headlights or noise-

dominant regions. These adjustments ensured that the model learned to extract meaningful structure from visually degraded inputs while retaining its previously learned daytime representations.

The fine-tuned model was trained using a mix of original nighttime images, enhanced versions produced by multiple illumination-correction methods, and synthetic nighttime scenes generated through day-to-night translation. This heterogeneous training set exposed the network to a wide spectrum of lighting conditions, noise patterns, headlight intensities, and object appearances. As a result, the model became more resilient to environmental variability: it learned to detect vehicles not only in moderately illuminated nighttime scenes but also in situations with extreme contrast, backlighting, or partial occlusions. The presence of multiple enhancement techniques additionally encouraged robustness to diverse image artifacts, since the model had to generalize across varying enhancement styles rather than rely on any single preprocessing method.

The fine-tuned model achieved a mAP@0.5 of 0.95 on our nighttime test set, demonstrating strong performance even in scenes containing glare, motion blur, and severely underexposed regions. Figure 31 illustrates several successful nighttime detections, highlighting the model's ability to correctly classify vehicles despite challenges such as intense headlight flare, reflective surfaces, distorted colors, and low-signal sensor noise. In many cases, the model was able to identify vehicles whose outlines were barely visible in the raw footage, indicating that the combined enhancement-augmentation pipeline effectively restored features essential for recognition.

Figure 31. Example results of nighttime vehicle detection.

These results confirm the value of integrating enhancement, generative augmentation, and task-specific fine-tuning into a cohesive nighttime adaptation pipeline. However, they also revealed an important limitation: despite the notable gains in nighttime performance, the resulting model specialized strongly in low-light scenarios. Applying this fine-tuned nighttime model to daytime or rainy-weather data led to decreased accuracy due to mismatched visual characteristics and domain-specific feature biases. Consequently, we were forced to maintain separate model variants for different environmental conditions—daytime, nighttime, and rainy scenes—to ensure optimal performance. This requirement increased operational complexity, especially for real-world deployment where conditions change dynamically and models must switch seamlessly.

The need for multiple specialized models reduces scalability, complicates model selection at runtime, and introduces additional storage and maintenance overhead. These constraints motivated our transition toward a unified multi-expert architecture capable of handling diverse conditions within a single adaptable framework. The next subsection describes how we designed and integrated this architecture to overcome the limitations of maintaining separate models while preserving high performance across all lighting and weather domains.

6.4 SUMMARY

In summary, the enhancement, data augmentation, and fine-tuning strategies presented in this chapter significantly improved the model's performance under challenging nighttime conditions. The use of diverse enhancement methods provided richer visual signals, enabling the model to recognize vehicles even in scenes with extreme low illumination, glare, or heavy noise. The CycleGAN-generated nighttime images effectively addressed the scarcity of real nighttime data, while the fine-tuning process successfully integrated these varied inputs into a more robust feature representation. Together, these components formed a comprehensive nighttime adaptation pipeline that delivered strong and reliable detection accuracy.

However, despite these benefits, the approach also introduced several limitations. The combined use of multiple enhancement techniques and generative models increased computational overhead, making the overall pipeline slower and less suitable for real-time inference on resource-constrained edge devices. Additionally, the need to maintain separate models for different environmental conditions—daytime, nighttime, and rainy weather—added operational complexity and reduced system scalability. Managing multiple specialized models is impractical in large-scale deployments and complicates real-time switching between conditions.

To overcome these challenges, we aim to reduce computational redundancy while maintaining high accuracy across diverse scenarios. This motivation leads to the next chapter, where we introduce a unified MoE architecture designed to dynamically adapt to varying environmental conditions. The MoE framework enables efficient sharing of knowledge across domains while activating only the most relevant experts during inference, making it well-suited for edge deployment.

7.0 MIXTURE-OF-EXPERTS (MOE) FRAME ARCHITECTURE

7.1 OVERVIEW

Traffic imagery collected under different environmental conditions—daytime, nighttime, and rainy weather—exhibits substantial variation in illumination, noise levels, contrast, and visual clarity. These differences impose conflicting requirements on a single deep learning model, making it difficult for one network to perform optimally across all domains. Earlier attempts to address this problem involved training separate models for each condition, but this approach introduces significant deployment challenges, including increased storage requirements, manual model switching, and reduced scalability on edge devices.

The(MoE framework offers a principled solution to these limitations. Originally developed to enhance model expressiveness and computational efficiency, MoE architectures combine multiple specialized expert networks with a learnable gating module that dynamically selects or weights the most relevant experts for each input. This design enables domain-specific specialization within a unified model, allowing different experts to focus on their respective environmental conditions—such as nighttime low-light features or rain-induced distortions—without interfering with each other's learning.

As shown in Figure 32, in this project we adopted an MoE architecture to consolidate daytime, nighttime, and rainy-scene expertise into a single adaptive system. The framework provides several key advantages:

- **Unified deployment** without maintaining multiple separate models.
- **Real-time adaptability**, as the gating module routes each input to the most suitable experts.

- **Computational efficiency**, since only a subset of experts is activated per frame—ideal for edge devices.
- **Scalability**, allowing new experts to be added for additional weather or lighting conditions.
- **Improved robustness**, particularly in ambiguous or transitional environments.

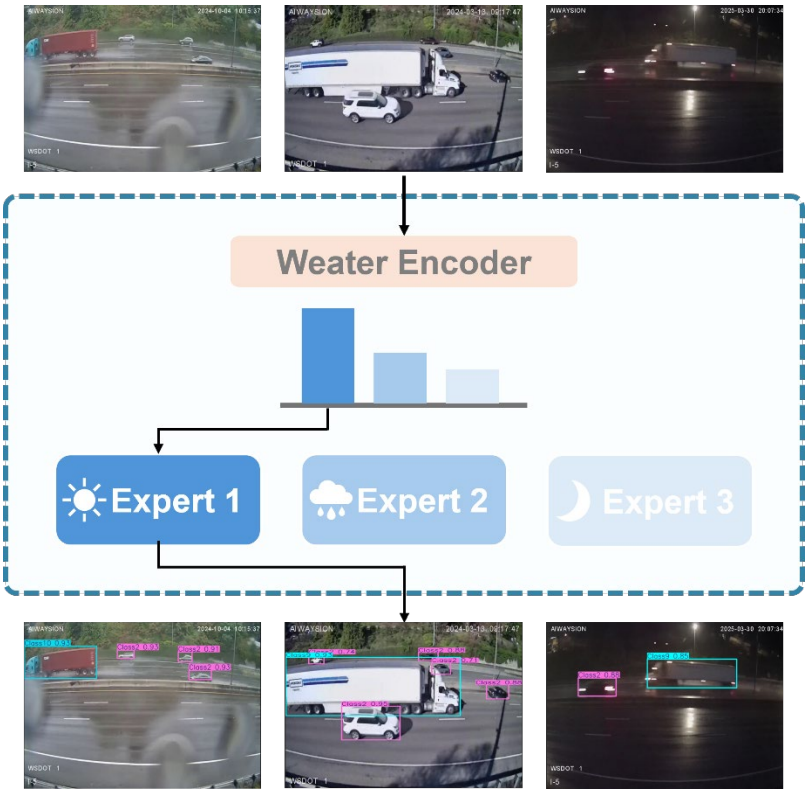


Figure 32. Overview of the proposed MoE model.

This chapter details the design of our MoE structure, the training strategy used to balance expert learning, and the resulting improvements in generalization across diverse environmental conditions. The MoE architecture represents a critical step toward building a scalable, efficient, and unified model capable of operating reliably in real-world traffic scenarios.

7.2 ARCHITECTURE DESIGN OF THE MIXTURE-OF-EXPERTS FRAMEWORK

The MoE architecture (as shown in Figure 33) used in this project integrates a shared backbone, a set of structurally identical expert networks, and a FiLM-based conditional modulation mechanism. Although the experts share exactly the same architecture, they learn distinct parameterizations driven by the statistics of their corresponding environmental domains. This design preserves architectural simplicity while enabling specialization through data-driven optimization. More importantly, the MoE structure addresses a fundamental limitation of single-model approaches: a single set of parameters cannot simultaneously satisfy the conflicting representational demands posed by daytime brightness, nighttime low-light noise, and rainy-scene occlusions. By allowing experts to diverge in parameter space while maintaining a unified computational pipeline, the architecture provides a principled way to capture diverse environmental characteristics without incurring the complexity, redundancy, or deployment burden associated with training separate models.

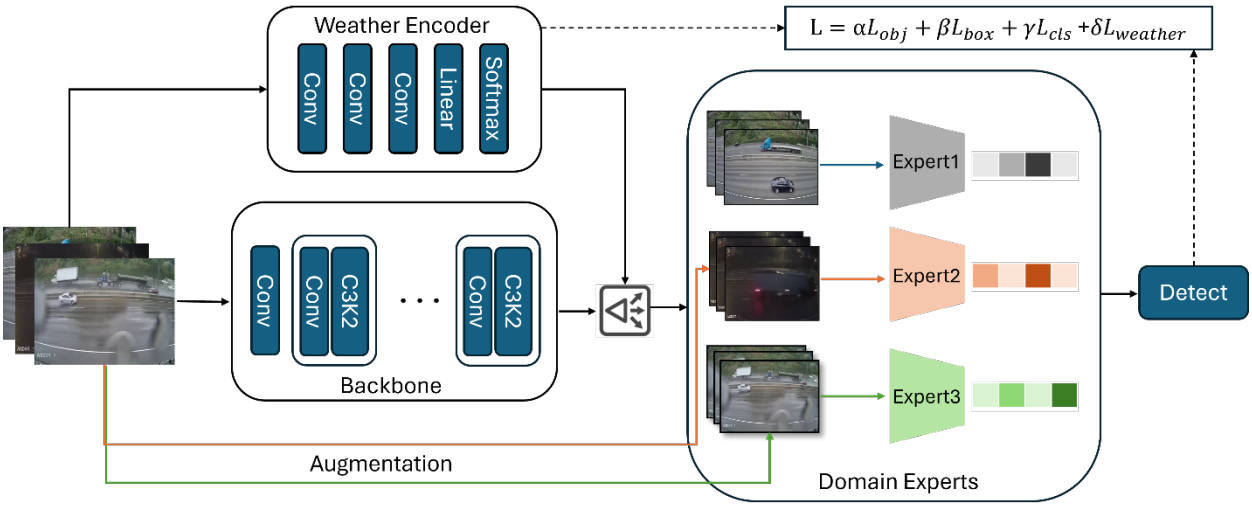


Figure 33. The architecture of MoE model.

7.2.1 Shared Backbone

Each input frame x is first processed by a shared backbone network f_{θ_b}, producing an intermediate feature representation:

$$F = f_{\theta_b}(x), F \in \mathbb{R}^{C \times H \times W} \quad \text{Eq. 1}$$

The shared backbone follows a YOLO-style architecture, incorporating a sequence of convolutional blocks, CSP (Cross-Stage Partial) modules, and spatial pyramid components to efficiently capture multi-scale information. Such backbones are designed to progressively compress spatial resolution while enriching semantic content, enabling the extraction of features that remain stable across diverse lighting and weather conditions. In particular, the early layers emphasize local edge and texture information, whereas deeper layers encode higher-level structures such as vehicle shapes, lane geometry, and motion-related contours. By centralizing this computation within a single backbone, the architecture avoids redundant feature extraction across experts and ensures consistent low-level processing. This design also reduces computational load—especially important for edge-device deployment—while preventing experts from learning conflicting representations of shared visual primitives. The resulting backbone features thus form a robust and unified foundation upon which the specialized experts can apply condition-specific refinements.

7.2.2 Expert Modules

Following backbone extraction, the unified feature representation **F**is passed through multiple expert modules:

$$E_k = (F; \theta_k), k \in \{day, night, rain\} \quad \text{Eq. 2}$$

Each expert implements the same functional architecture $\varepsilon(\cdot)$:

$$E_k(F; \theta_k) = \varepsilon(F; \theta_k) \quad \text{Eq. 3}$$

where only the parameters θ_k differ across experts. In practice, each expert follows a YOLO-style head structure, consisting of lightweight convolutional blocks, CSP or bottleneck modules, and multi-scale feature refinement layers. This design mirrors the standard YOLO detection head's ability to effectively process hierarchical spatial features while keeping computational overhead modest.

Because the experts are structurally identical, the MoE framework benefits from architectural uniformity, simplified deployment, and reduced engineering complexity. However, despite having the same architecture, the experts quickly diverge during training due to domain-specific gradients and loss signals:

- **The daytime expert** becomes more responsive to high-frequency structures, sharp contours, and illumination-rich textures. YOLO's convolutional blocks naturally amplify detailed spatial information, enabling this expert to exploit well-lit conditions and strong contrast.
- **The nighttime expert** adapts toward suppressing sensor noise, stabilizing low-SNR features, attenuating glare from headlights, and emphasizing coarse, shape-based cues. The YOLO-style feature processing pipeline encourages robustness by propagating smoothed, noise-tolerant representations through its bottleneck layers.
- **The rainy expert** learns to handle raindrop occlusions, streak artifacts, surface reflections, and reduced visibility. Its refinement layers adjust to ambiguous local textures, focusing on geometric cues and mid-level patterns that remain reliable under weather-induced degradation.

Although the experts share identical YOLO-inspired architectures, their parameter sets θ_k diverge significantly as each expert learns the statistical regularities of its target domain. This

controlled diversification ensures that each expert becomes a domain specialist while still maintaining consistency with the shared computational design.

7.2.3 Environment Encoding

Environmental conditions such as daytime brightness, nighttime low-light noise, and rain-induced distortions significantly influence visual appearance in traffic imagery. To ensure that the MoE framework can adaptively respond to these variations, the system incorporates a weather encoding module that converts high-level visual statistics into a compact environmental representation. This encoding serves as the foundation for modulating expert behavior in a principled and computationally efficient manner.

After the shared backbone produces the intermediate feature representation **F**, a global summary of the scene is extracted using spatial pooling or feature aggregation. This summary vector is then processed by a shallow multilayer perceptron (MLP):

$$\boldsymbol{z} = \boldsymbol{g}_{\emptyset}(\boldsymbol{F}) \qquad \text{Eq. 4}$$

where g_{ϕ}denotes the weather encoder. Despite its simplicity, this shallow MLP is sufficient to capture key environmental cues—such as average brightness, noise level, contrast variation, and texture stability—that correlate strongly with different operating conditions. Because the MLP is lightweight and fully differentiable, it can be trained jointly with the rest of the network, allowing the underlying representation to self-organize around meaningful environmental distinctions without requiring manual labels.

The resulting embedding **z**is then used to generate Feature-wise Linear Modulation (FiLM) parameters for each expert:

$$(\gamma_k, \beta_k) = h_{\phi_k}(z) \qquad \text{Eq. 5}$$

where h_{ϕ_k} is a small projection network that maps environmental context into channel-wise scaling and shifting factors. These FiLM parameters condition each expert's output:

$$\check{F}_k = \gamma_k \odot E_k(F) + \beta_k \qquad \text{Eq. 6}$$

allowing the system to dynamically emphasize or suppress expert features depending on the inferred environment.

FiLM-based conditioning is used because it provides a highly efficient and expressive mechanism for modulating convolutional features. Unlike explicit routing—where a gating network dynamically selects which expert branch should process the input— or attention-based mechanisms, FiLM introduces minimal computational overhead and integrates naturally with YOLO-style expert architectures, where channel-wise modulation directly influences the strength of learned patterns. Furthermore, FiLM supports smooth interpolation between experts, enabling robust performance in ambiguous or transitional scenarios—such as dawn, dusk, or light rain—where strict domain boundaries do not exist. Through the combination of a shallow weather encoder and FiLM-based modulation, the MoE architecture achieves fine-grained, real-time adaptation to diverse environmental conditions without increasing model complexity or compromising inference speed.

7.3 LOSS FUNCTION

The loss function plays a central role in guiding how the MoE framework learns to handle the combined challenges of class imbalance, environmental variation, and expert specialization. Although the architecture determines the pathways through which information flows, it is ultimately the loss configuration that shapes how the backbone, weather encoder, and expert branches cooperate to produce reliable predictions across diverse operating conditions. In our setting, the training objective jointly optimizes detection accuracy and auxiliary weather

classification, enabling both the detection heads and the routing mechanism to evolve in a consistent manner.

The overall loss is defined as

$$L_{total} = L_{det} + \lambda_{weather} \cdot L_{weather} \quad \text{Eq. 7}$$

where L_{det} corresponds to the YOLO-style detection loss and $L_{weather}$ is a cross-entropy term supervising the weather encoder. The auxiliary weather supervision helps the encoder learn meaningful distinctions between daytime, nighttime, and rainy scenes, which is essential for stable routing during inference.

The detection loss follows standard object detection practice and includes box regression, object prediction, and classification terms. For each spatial location (i, j), the loss is computed as

$$L_{det} = \sum_{i,j}[\lambda_{box} \cdot L_{CIoU}(b_{ij}, \hat{b}_{ij}^{k}) + \lambda_{obj} \cdot L_{BCE}(o_{ij}, \hat{o}_{ij}^{k}) + \lambda_{cls} \cdot w_{c_{ij}} L_{CE}(p_{ij}, \hat{p}_{ij}^{k})] \quad \text{Eq. 8}$$

Here, L_{CIoU} encourages accurate bounding box geometry, L_{BCE}handles object prediction, and L_{CE}measures classification error. Because the FHWA vehicle dataset is naturally long-tailed—with a few classes such as light-duty vehicles appearing far more frequently than heavy multi-trailer categories—we apply class-dependent weights $w_{c_{ij}}$ to counteract this imbalance. In practice, these weights help prevent the detector from collapsing toward high-frequency classes. When dealing with severe imbalance or noisy nighttime labels, focal loss:

$$L_{Focal}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \quad \text{Eq. 9}$$

can optionally replace the standard classification term, placing more emphasis on difficult or ambiguous examples.

The weather classification loss is a simple cross-entropy applied to the output of the weather encoder. Although small in magnitude, this term is important: it stabilizes the routing mechanism, reinforces the association between environmental cues and expert behavior, and

improves generalization in mixed or transitional conditions. Without this supervision, the encoder may drift during training, leading to experts receiving mismatched gradients and degrading specialization.

All components—the backbone, weather encoder, FiLM conditioning layers, and expert decoders—are trained jointly using AdamW with a cosine learning-rate schedule. This unified optimization strategy ensures that detection performance, expert specialization, and routing consistency develop together rather than competing, ultimately leading to a more stable and adaptable all-weather detection system.

7.4 SUMMARY

This section presented a unified, all-weather vehicle detection framework built upon a mixture-of-experts (MoE) architecture with FiLM-based conditional modulation. Motivated by the substantial appearance variation across daytime, nighttime, and rainy environments, the proposed system addresses a long-standing challenge in traffic perception: enabling a single model to maintain stable performance under diverse and often adverse operating conditions. By combining a shared YOLO-style backbone, structurally identical expert modules, and a lightweight weather encoder that produces FiLM modulation parameters, the framework allows domain-specific specialization to emerge naturally while preserving architectural simplicity and computational efficiency.

Through end-to-end training with a joint detection–weather classification loss, the system learns both accurate vehicle representations and reliable environmental cues. The detection loss encourages robust localization and classification across the 13 FHWA vehicle categories, while the auxiliary weather loss stabilizes the routing mechanism and ensures consistent expert

activation. Together, these components enable the model to adapt dynamically as environmental conditions shift, and to improve prediction reliability during transitional or ambiguous scenes.

Overall, this work contributes a flexible, scalable, and practical approach to all-weather vehicle detection. The modularity of the MoE design opens the door to further extensions, including the addition of new experts tailored to fog, snow, or glare; incorporation of temporal cues for improved stability; or integration with larger multimodal systems for comprehensive traffic sensing. As intelligent transportation systems increasingly rely on vision-based perception, the principles demonstrated here offer a promising path toward models that can operate reliably and safely in the unpredictable conditions of real-world environments.

8.0 RESULTS AND ABLATION STUDY

This section presents a comprehensive evaluation of the proposed (MoE) framework. The purpose of these experiments is not only to quantify raw detection accuracy but also to understand why the system works: which architectural components are essential, which loss functions are most suitable, and how well the model generalizes to new deployment conditions. Because the MoE architecture introduces several interacting modules—namely the Weather Encoder, expert routing, FiLM modulation, and class-imbalance handling—it is important to examine each one in isolation. Ablation studies allow us to separate the contribution of individual components and verify that the observed improvements stem from meaningful model behavior rather than incidental increases in model size or complexity. The results presented here follow the 13-class FHWA taxonomy and were conducted using the expanded nighttime and rainy-weather datasets described in earlier sections.

8.1 RESULTS

We evaluate the proposed mixture-of-experts detection framework on our benchmark dataset, which includes fine-grained vehicle annotations across three distinct environmental domains: clear, rain, and night. To ensure balanced and fair evaluation, each domain is randomly partitioned into 80 percent for training, 10 percent for validation, and 10 percent for testing, while preserving the original class distribution within each split.

We adopt standard COCO-style object detection metrics to evaluate detection accuracy:

- mAP@0.5: Mean Average Precision at an IoU threshold of 0.5, commonly used to benchmark localization performance.

- mAP@[.5:.95]: The average mAP across IoU thresholds from 0.5 to 0.95 (in steps of 0.05), providing a stricter and more comprehensive evaluation of both localization and classification accuracy.

To validate the effectiveness of our approach, we compare against a diverse set of competitive baselines:

- YOLOv8 (m, n) [27][29]: A recent high-performance one-stage detector. We include both medium (m) and nano (n) variants to reflect different model capacities. This baseline lacks explicit domain adaptation.
- RT-DETR (l, x)[43]: A transformer-based real-time detection model that refines dense queries for improved performance. Both large (l) and extra-large (x) variants are evaluated.
- YOLOv11 (m, n) [27][29] : Our backbone and detection head baseline with a single shared decoder across all domains. This configuration isolates the effect of expert routing by removing domain specialization.
- Ours (Full): The complete proposed framework with a shared backbone, domain-specific expert decoders, and a context encoder for hard expert routing.

Table 5 shows the overall detection performance in terms of mAP@50 and mAP@50:95 across all classes and environmental conditions. Our proposed method consistently outperforms all baselines, achieving the highest scores in both metrics. The superior average performance across all classes highlights the effectiveness and robustness of our approach under varying scenarios.

Table 5. Per-class comparison of mAP across different methods.

Method	**Ours**	**RT-DETR-X**	**RT-DETR-L**	**YOLOv11-M**	**YOLOv11-N**	**YOLOv8-M**	**YOLOv8-N**
Class1 mAP@50	0.995	0.995	0.797	0.995	0.995	0.995	0.995
Class1 mAP@50:95	0.827	0.897	0.590	0.849	0.812	0.830	0.751
Class2 mAP@50	0.972	0.967	0.969	0.977	0.977	0.976	0.975
Class2 mAP@50:95	0.858	0.833	0.823	0.863	0.847	0.841	0.852
Class3 mAP@50	0.928	0.896	0.893	0.929	0.930	0.925	0.917
Class3 mAP@50:95	0.834	0.792	0.789	0.834	0.827	0.814	0.814
Class4 mAP@50	0.995	0.995	0.946	0.995	0.995	0.995	0.995
Class4 mAP@50:95	0.941	0.960	0.915	0.960	0.947	0.940	0.952
Class5 mAP@50	0.968	0.907	0.927	0.968	0.959	0.972	0.951
Class5 mAP@50:95	0.914	0.856	0.870	0.907	0.879	0.901	0.876
Class6 mAP@50	0.986	0.995	0.981	0.986	0.995	0.995	0.990
Class6 mAP@50:95	0.946	0.924	0.940	0.924	0.928	0.960	0.926
Class7 mAP@50	0.987	0.952	0.958	0.970	0.981	0.995	0.993
Class7 mAP@50:95	0.941	0.911	0.896	0.931	0.922	0.926	0.928
Class8 mAP@50	0.935	0.918	0.876	0.950	0.906	0.882	0.882
Class8 mAP@50:95	0.887	0.877	0.845	0.892	0.870	0.847	0.847
Class9 mAP@50	0.982	0.949	0.952	0.982	0.986	0.972	0.977
Class9 mAP@50:95	0.933	0.897	0.901	0.922	0.923	0.903	0.917
Class10 mAP@50	0.990	0.939	0.931	0.971	0.990	0.990	0.974
Class10 mAP@50:95	0.955	0.897	0.883	0.929	0.939	0.935	0.930
Class11 mAP@50	0.995	0.995	0.995	0.928	0.831	0.995	0.995
Class11 mAP@50:95	0.995	0.995	0.995	0.928	0.831	0.995	0.968
Class12 mAP@50	0.995	0.905	0.995	0.995	0.988	0.981	0.988
Class12 mAP@50:95	0.974	0.870	0.940	0.955	0.964	0.956	0.973
Class13 mAP@50	0.995	0.995	0.995	0.995	0.995	0.995	0.983

Method	Ours	RT-DETR-X	RT-DETR-L	YOLOv11-M	YOLOv11-N	YOLOv8-M	YOLOv8-N
Class13 mAP@50:95	0.976	0.973	0.979	0.962	0.955	0.960	0.967
All mAP@50	0.979	0.955	0.940	0.972	0.964	0.978	0.970
All mAP@50:95	0.922	0.899	0.874	0.912	0.896	0.910	0.900

Figure 34 illustrates visual comparisons between original traffic scenes (top row) and the corresponding feature representations extracted by our model (bottom row) under different environmental conditions—clear daytime, nighttime, and rainy weather. These feature maps demonstrate the model's ability to extract robust and discriminative representations across diverse domains. For instance, during nighttime (center column), the model enhances contours and suppresses background noise, allowing it to retain vehicle geometry despite low illumination. Under rainy conditions (right column), the model captures essential shape and texture information while minimizing the impact of water streaks and motion blur. These qualitative results indicate that our multi-expert framework, guided by the weather encoder, effectively learns domain-specific representations that remain consistent for fine-grained vehicle detection, thereby improving overall robustness and generalization under challenging conditions.

Figure 34. Visual comparison of input images (top) and extracted feature maps (bottom).

To further illustrate the robustness of our approach, Figure 35 shows detection results under various environmental conditions using real-world roadside camera data. Each column highlights a different scenario: daytime (left), nighttime (middle), and rainy weather (right). The proposed multi-expert model accurately detects a wide range of vehicle classes—including trucks, tankers, sedans, and specialty vehicles—even in the presence of adverse visual factors like glare, blur, and rain streaks. These results underscore the effectiveness of our condition-aware expert specialization and scene-adaptive routing in handling environmental variability and class imbalance in real-world settings.

Figure 35. Qualitative results of our MoE detection model.

8.2 ROUTING MECHANISM ABLATION

To understand whether the MoE framework truly benefits from environment-aware expert selection, we evaluated three routing configurations and analyzed how each one

influences overall performance. This ablation is necessary because routing is the core mechanism that decides which expert processes a given image; without isolating its contribution, it would be impossible to determine whether the model's improvements come from meaningful decision-making or simply from increased parameter count.

The first configuration keeps the full Weather Encoder and uses learned environmental embeddings to guide expert selection. This represents the intended behavior of the MoE system and allows us to measure the benefit of accurate, semantically meaningful routing. The second configuration removes routing entirely, effectively collapsing the architecture into a single-expert baseline. This version provides a reference point for what the model can achieve without any expert specialization. The third configuration freezes the routing module at random initialization. This preserves the overall parameter budget and architecture but prevents the routing unit from learning useful environmental cues. Comparing the random-routing model against the learned-routing model helps us determine whether improvements are due to genuine expert selection or simply to added model capacity.

As shown in Table 6, the results clearly show why this ablation is essential. Both the baseline configuration without routing and the configuration with random, untrained routing perform almost identically, achieving roughly 0.915–0.916 mAP@50. The similarity between these two outcomes confirms that simply adding a routing module does not improve detection accuracy. In contrast, when routing is fully learned, performance rises to 0.951 mAP@50. This sizable improvement demonstrates that the routing module contributes meaningful information and that the MoE system benefits specifically from the Weather Encoder's ability to provide accurate environmental context. By isolating the routing mechanism in this way, the ablation

confirms that performance gains stem from the model's ability to choose appropriate experts rather than from architectural complexity alone.

Table 6. MoE architecture ablation.

Weather Mode	FiLM Enabled	Result
None	No	0.915
Random	No	0.916
Robust	No	0.951
None	Yes	0.940
Random	Yes	0.949
Robust	Yes	**0.979**

8.3 FILM MODULATION ABLATION

FiLM modulation is designed to adapt feature representations based on environmental cues generated by the Weather Encoder. However, because FiLM relies directly on the routing embedding, it is important to test whether FiLM improves performance only when routing is accurate or whether it provides benefits even in the absence of meaningful environmental information. This ablation is necessary to determine whether FiLM is acting as a principled conditioning mechanism or merely as an additional nonlinear transformation unrelated to context.

We evaluated FiLM under three routing conditions: robust learned routing, no routing, and random routing. These comparisons help us identify the degree to which FiLM depends on high-quality context vectors to operate correctly.

As shown in Table 6, the results show that FiLM does not operate independently of routing quality. When paired with robust routing, FiLM delivers a substantial improvement, raising accuracy from 0.951 to 0.979 mAP@50. This confirms that when the environmental

embedding is accurate, FiLM effectively tailors feature representations to the corresponding weather condition. In contrast, when routing is either absent or random, FiLM actually reduces performance. This behavior illustrates why the ablation is critical: without correct environmental inputs, FiLM conditions features incorrectly, leading to degraded detection performance. The study therefore demonstrates that FiLM and routing are tightly coupled components and must be jointly optimized to achieve maximum benefit.

8.4 LOSS FUNCTION ABLATION

Because the dataset exhibits notable class imbalance—particularly among heavy-vehicle categories under nighttime and rainy conditions—it is important to investigate how different loss formulations influence the MoE model's performance. Ablation is necessary here to determine whether observed accuracy gains arise from the architecture itself or from the choice of loss function. By isolating the loss, we can understand which formulation best supports stable optimization and balanced performance across both common and rare vehicle classes.

Four losses were evaluated using the same architecture (robust routing with FiLM modulation):

The standard BCE loss treats all classes equally:

$$\mathcal{L}_{\text{BCE}} = -[y\log\, p + (1-y)\log\,(1-p)] \qquad \text{Eq. 10}$$

where $y$is the ground-truth label and $p$is the predicted probability.

To explicitly handle class imbalance, BCE can be augmented with class-specific weights w_c:

$$\mathcal{L}_{\text{WBCE}} = -w_c[y\log\, p + (1-y)\log\,(1-p)] \qquad \text{Eq. 11}$$

Rare classes receive larger weights, ensuring their gradients are not overwhelmed by dominant classes.

Focal Loss down-weights easy samples using a modulating term $(1-p)^{\gamma}$:

$$\mathcal{L}_{\text{Focal}} = -(1-p)^{\gamma}\, y\log\, p - p^{\gamma}(1-y)\log\,(1-p) \qquad \text{Eq. 12}$$

where γcontrols how strongly easy examples are suppressed.

Soft Focal Loss smooths the hard weighting effect by replacing the binary labels with soft targets $\tilde{y}$:

$$\mathcal{L}_{\text{SoftFocal}} = -(1-p)^{\gamma}\, \tilde{y}\log\, p - p^{\gamma}(1-\tilde{y})\log\,(1-p) \qquad \text{Eq. 13}$$

Soft targets help reduce training instability but still rely on the same down-weighting of easy samples.

As shown in Table 7, weighted BCE is the most effective, achieving 0.979 mAP@50. The other three losses cluster at 0.943 mAP@50, suggesting that dynamic reweighting strategies such as Focal Loss offer little advantage for this dataset. The performance of Weighted BCE demonstrates that explicitly compensating for class imbalance through fixed per-class weights is more effective than relying on adaptive mechanisms. This ablation confirms that the model benefits from a loss function that directly addresses class frequency disparities and supports consistently strong performance across both common and rare vehicle types.

Table 7. Loss function ablation.

Loss Function	**Weather Mode**	**FiLM**	**Result**
BCE	Robust	Yes	0.943
Weighted BCE	Robust	Yes	**0.979**
Focal Loss	Robust	Yes	0.943
Soft Focal Loss	Robust	Yes	0.943

8.5 CROSS-SITE GENERALIZATION STUDY

To determine whether the optimized MoE model is suitable for real-world deployment, we evaluated its performance at a completely separate site with different roadway geometry. This experiment is essential for assessing generalizability, since a model that performs well only on its training site would be unsuitable for large-scale deployment across diverse traffic environments.

The MoE configuration used in this evaluation combines robust routing, FiLM modulation, and the Weighted BCE loss. Despite the substantial differences in environmental characteristics at the new location, the model achieves the same 0.979 mAP@50 reported at the original site, requiring only minimal fine-tuning. This result demonstrates that the Weather Encoder generalizes beyond the conditions it was trained on, that expert selection remains stable, and that the architecture is not overfitted to site-specific lighting or geometric constraints. The study confirms that the model is ready for deployment at additional locations with little adaptation effort.

9.0 CONCLUSIONS AND FUTURE WORK

This section provides a comprehensive summary of the project's contributions and outlines future directions to further advance the system. Over the course of the project, we developed a complete vehicle classification framework capable of operating in real-world, state-level transportation environments. The system evolved from initial daytime-only classification models into a robust, adaptive platform designed to handle diverse environmental conditions—including nighttime, rain, and other sources of visual degradation commonly encountered in highway traffic video.

The work completed across all tasks demonstrated the feasibility and value of using modern machine learning techniques, combined with targeted data augmentation and domain-specific optimizations, to achieve reliable vehicle classification performance at scale. The final system integrates image enhancement, domain adaptation, and a unified MoE model, providing both high accuracy and operational efficiency.

9.1 CONCLUSION

Across the full duration of the project, we achieved several major technical milestones that significantly advance the state of practice in traffic video analytics.

9.1.1 Development of a High-Quality, Multi-Condition Dataset

We collected and curated a diverse dataset spanning daytime, nighttime, and rainy-weather conditions across multiple locations. The dataset includes the 13 FHWA vehicle classes and captures real operational variability—such as headlight glare, low-light environments, raindrop occlusions, splash effects, varied pavement surfaces, and mixed traffic density. This dataset became the foundation for all enhancement, adaptation, and training efforts.

9.1.2 Image Enhancement and Nighttime Adaptation Pipeline

To address low-light conditions, we assembled and evaluated an extensive suite of state-of-the-art enhancement algorithms, including Retinex-based, curve-based, GAN-based, flow-based, and Transformer-based models. We also incorporated CycleGAN-generated day-to-night images to expand nighttime training diversity. Through systematic fine-tuning, the nighttime model achieved mAP@0.5 = 0.95, confirming the effectiveness of the enhancement–augmentation–adaptation strategy.

9.1.3 Adaptation to Rainy Weather and Challenging Visibility

We collected real-world rainy-weather footage and fine-tuned the model to handle rain-induced distortions. The refined model achieved 97.9 percent accuracy across all FHWA classes under rainy conditions, demonstrating strong resilience to both short-term (raindrops, reflections) and long-term (reduced contrast) degradations.

9.1.4 Unified Mixture-of-Experts (MoE) Framework

A major contribution of this project was the development of a unified MoE framework designed to replace separate daytime, nighttime, and rainy models. The MoE architecture includes the following:

- a shared backbone for general feature extraction
- specialized daytime, nighttime, and rainy-weather expert branches
- a lightweight Weather Encoder for real-time condition recognition
- FiLM-based modulation for fine-grained feature adjustment.

This architecture preserves the accuracy of specialized models while maintaining the efficiency and maintainability of a single unified system.

9.1.5 Extensive Ablation Studies and Model Validation

Through systematic experimentation, we identified the optimal configurations of the following:

- expert routing
- FiLM feature modulation
- class imbalance handling (weighted BCE)
- weather-classification supervision.

Routing and FiLM together improved performance from 0.951 to 0.979 mAP@50, confirming the architectural choices. Weighted BCE proved essential for stabilizing performance across rare FHWA classes.

9.1.6 Generalization to New Deployment Sites

To ensure statewide scalability, the final MoE system was deployed and evaluated at an additional site with different roadway geometry and ambient conditions. After minimal fine-tuning, the model maintained its 0.979 mAP@50 performance, confirming strong cross-location generalization and readiness for broader operational use.

The project successfully delivered the following:

- A unified, environment-adaptive model
- High accuracy under multiple challenging conditions
- Strong generalization across locations
- A scalable framework suitable for future expansion.

This integrated system represents a significant improvement over traditional single-condition models and offers a practical, deployment-ready solution for vehicle classification in real-world transportation networks.

9.2 FUTURE WORK

Although the MoE framework demonstrated strong performance and readiness for deployment, several natural extensions can further enhance its robustness and operational value. These directions do not reflect missing components but rather opportunities to broaden the system's applicability as WSDOT deployments continue to expand.

A key area for future improvement is the incorporation of additional environmental conditions beyond the nighttime and rainy scenarios evaluated in this project. Real-world traffic cameras frequently encounter fog, snow, glare, and other forms of low visibility. The modular structure of the MoE architecture is well suited for such extensions, as new experts can be added with minimal modification to the core system.

Another natural direction is the continued expansion to additional WSDOT sites. As the system is deployed more broadly, long-term monitoring will provide valuable insights into model stability under varying hardware configurations and roadway geometries. This will help refine maintenance strategies and ensure consistent statewide performance.

Finally, the system may benefit from adding lightweight temporal cues to further stabilize predictions during occlusions or rapid lighting changes. Such enhancements would build on the already strong performance of the current model and help reduce frame-to-frame variability in challenging segments of video.

Overall, the proposed MoE framework is technically mature and deployment-ready, and the future work outlined above represents natural, incremental expansions rather than fundamental limitations of the current system.

10.0 REFERENCES

[1] American Society of Civil Engineers (ASCE). (2021). 2021 Report Card for America's Infrastructure. ASCE. https://2021.infrastructurereportcard.org/

[2] Federal Highway Administration (FHWA). (2023). Traffic Data and Monitoring Overview. U.S. Department of Transportation. https://www.fhwa.dot.gov/policyinformation/

[3] Federal Highway Administration (FHWA). (2013). Traffic Monitoring Guide. U.S. Department of Transportation. https://www.fhwa.dot.gov/policyinformation/tmguide/tmg_2013/

[4] Hallenbeck M E, Selezneva O I, Quinley R. Verification, refinement, and applicability of long-term pavement performance vehicle classification rules[R]. United States. Federal Highway Administration. Office of Infrastructure Research and Development, 2014.

[5] Krause J, Deng J, Stark M, et al. Collecting a large-scale dataset of fine-grained cars[J]. 2013.

[6] Yang L, Luo P, Change Loy C, et al. A large-scale car dataset for fine-grained categorization and verification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3973-3981.

[7] Yu F, Chen H, Wang X, et al. Bdd100k: A diverse driving dataset for heterogeneous multitask learning[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 2636-2645.

[8] Wen L, Du D, Cai Z, et al. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking[J]. Computer Vision and Image Understanding, 2020, 193: 102907.

[9] Murrugarra R., Wallace W., Wojtowicz J. Task 30: Data Fusion Methodology; Technical Report 10-06. Center for Infrastructure, Transportation and the Environment, Department of Civil and Environmental Engineering, Rensselaer Polytechnic Institute; Troy, NY, USA: 2010.

[10] Mocholí-Salcedo A., Arroyo-Núñez J.H., Milián-Sánchez V.M., Palomo-Anaya M.J., Arroyo-Núñez A. Magnetic Field Generated by the Loops Used in Traffic Control Systems. IEEE Trans. Intell. Transp. Syst. 2017;18:2126–2136. doi: 10.1109/TITS.2016.2632972.

[11] Meta S., Cinsdikici M.G. Vehicle-classification algorithm based on component analysis for single-loop inductive detector. IEEE Trans. Veh. Technol. 2010;59:2795–2805. doi: 10.1109/TVT.2010.2049756.

[12] Wu L., Coifman B. Improved vehicle classification from dual-loop detectors in congested traffic. Transp. Res. Part C Emerg. Technol. 2014;46:222–234. doi: 10.1016/j.trc.2014.04.015.

[13] Gajda J., Mielczarek M. Automatic vehicle classification in systems with single inductive loop detector. Metrol. Meas. Syst. 2014;21:619–630. doi: 10.2478/mms-2014-0048.

[14] Peters R.J. Australian Road Research Board Proceedings. The National Academies of Sciences, Engineering and Medicine; Washington, DC, USA: 1986. Culway, an Unmanned and Undetectable Highway Speed Vehicle Weighing System.

[15] Roh H.J., Sharma S., Sahu P.K. Modeling snow and cold effects for classified highway traffic volumes. KSCE J. Civ. Eng. 2016;20:1514–1525. doi: 10.1007/s12205-015-0236-0.

[16] Romanoschi S.A., Momin S., Bethu S., Bendana L. Development of traffic inputs for new Mechanistic-Empirical Pavement Design Guide. Transp. Res. Rec. 2011:142–150. doi: 10.3141/2256-17.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[19] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[20] S. Ren et al., "Faster R-CNN: Towards real-time object detection with region proposal networks," in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99.

[21] Ji-qing Luo, Hu-sheng Fang, Fa-ming Shao, Yue Zhong, Xia Hua, "Multi-scale traffic vehicle detection based on faster R–CNN with NAS optimization and feature enrichment", Defence Technology, Volume 17, Issue 4, 2021, Pages 1542-1554, ISSN 2214-9147,2020.10.006.

[22] Y. O. Adu-Gyamfi, S. K. Asare, A. Sharma, and T. Titus, ''Automated vehicle recognition with deep convolutional neural networks,'' Transp. Res. Rec., vol. 2645, no. 1, pp. 113–122, 2017.

[23] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[24] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.

[25] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

[26] Jacob Solawetz and Francesco. What is yolov8? the ultimate guide., 2023. 04-30-2023. https://blog.roboflow.com/yolov8-tracking-and-counting/

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 779–788.

[28] M. I. Asborno, C. G. Burris, and S. Hernandez, ''Truck body-type classification using single-beam LiDAR sensors,'' Transp. Res. Rec., J. Transp. Res. Board, vol. 2673, no. 1, pp. 26–40, Jan. 2019.

[29] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," 2023. Available: https://github.com/ultralytics/ultralytics

[30] Sapkota, Ranjan & Flores-Calero, Marco & Qureshi, Rizwan & Badgujar, Chetan & Nepal, Upesh & Poulose, Alwin & Zeno, Peter & Vaddevolu, Uday Bhanu Prakash & Khan, Sheheryar & Shoman, Maged & Yan, Hong & Karkee, Manoj. (2025). YOLO advances to its genesis: a decadal and comprehensive review of the You Only Look Once (YOLO) series. Artificial Intelligence Review. 58. 10.1007/s10462-025-11253-3.

[31] Guo C, Li C, Guo J, et al. Zero-reference deep curve estimation for low-light image enhancement[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1780-1789.

[32] Cai Y, Bian H, Lin J, et al. Retinexformer: One-stage retinex-based transformer for low-light image enhancement[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2023: 12504-12513.

[33] Zhou S, Li C, Change Loy C. Lednet: Joint low-light enhancement and deblurring in the dark[C]//European conference on computer vision. Cham: Springer Nature Switzerland, 2022: 573-589.

[34] Zhang Y, Zhang J, Guo X. Kindling the darkness: A practical low-light image enhancer[C]//Proceedings of the 27th ACM international conference on multimedia. 2019: 1632-1640.

[35] Jiang Y, Gong X, Liu D, et al. Enlightengan: Deep light enhancement without paired supervision[J]. IEEE transactions on image processing, 2021, 30: 2340-2349.

[36] Yang W, Wang S, Fang Y, et al. From fidelity to perceptual quality: A semi-supervised approach for low-light image enhancement[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 3063-3072.

[37] Saini S, Narayanan P J. Specularity factorization for low-light enhancement[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 1-12.

[38] Fu Z, Yang Y, Tu X, et al. Learning a simple low-light image enhancer from paired low-light instances[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 22252-22261.

[39] Ma L, Ma T, Liu R, et al. Toward fast, flexible, and robust low-light image enhancement[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 5637-5646.

[40] Wang Y, Wan R, Yang W, et al. Low-light image enhancement with normalizing flow[C]//Proceedings of the AAAI conference on artificial intelligence. 2022, 36(3): 2604-2612.

[41] Wang T, Zhang K, Shen T, et al. Ultra-high-definition low-light image enhancement: A benchmark and transformer-based method[C]//Proceedings of the AAAI conference on artificial intelligence. 2023, 37(3): 2654-2662.

[42] Chu C, Zhmoginov A, Sandler M. Cyclegan, a master of steganography[J]. arXiv preprint arXiv:1712.02950, 2017.

[43] Wang S, Xia C, Lv F, et al. RT-DETRv3: Real-time end-to-end object detection with hierarchical dense positive supervision[C]//2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE, 2025: 1628-1636.

Title VI Notice to Public

It is the Washington State Department of Transportation's (WSDOT's) policy to assure that no person shall, on the grounds of race, color, or national origin, as provided by Title VI of the Civil Rights Act of 1964, be excluded from participation in, be denied the benefits of, or be otherwise discriminated against under any of its programs and activities. Any person who believes his/her Title VI protection has been violated, may file a complaint with WSDOT's Office of Equity and Civil Rights (OECR). For additional information regarding Title VI complaint procedures and/or information regarding our non-discrimination obligations, please contact OECR's Title VI Coordinator at (360) 705-7090.

Americans with Disabilities Act (ADA) Information

This material can be made available in an alternate format by emailing the Office of Equity and Civil Rights at wsdotada@wsdot.wa.gov or by calling toll free, 855-362-4ADA(4232). Persons who are deaf or hard of hearing may make a request by calling the Washington State Relay at 711.