RESEARCH

UDOT

Keeping Utah Moving

Utah Department of Transportation - Research Division
4501 South 2700 West - P.O. Box 148410 - SLC,UT 84114-8410

Report No. UT-26.03

BENCHMARKING COMPUTER VISION-BASED APPROACHES TO DERIVE ENGINEERING-ORIENTED CONDITION FROM EXISTING UDOT ASSETS DATA

Prepared For:

Utah Department of Transportation
Research & Innovation Division

Final Report
February 2026

DISCLAIMER

The authors alone are responsible for the preparation and accuracy of the information, data, analysis, discussions, recommendations, and conclusions presented herein. The contents do not necessarily reflect the views, opinions, endorsements, or policies of the Utah Department of Transportation or the U.S. Department of Transportation. The Utah Department of Transportation makes no representation or warranty of any kind, and assumes no liability therefore.

ACKNOWLEDGMENTS

The authors acknowledge the Utah Department of Transportation (UDOT) for funding this research, and the following individuals from UDOT on the Technical Advisory Committee for helping to guide the research:

- Abdul Wakil, UDOT Asset Engineer for Maintenance
- Chris Whipple, UDOT Safety Programs Engineer
- Benjamin Maughan, UDOT Asset Management Analytics
- Sean Berry, UDOT Data Analytics, Software Administrator
- Scott Jones, DTS Information Technology Director
- Ryan Ferrin, UDOT Statewide Maintenance Engineer
- Shawn Lambert, UDOT Director of Construction
- Kevin Nichol, UDOT Research Project Manager

TECHNICAL REPORT ABSTRACT

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
UT- 26.03	N/A	N/A

4. Title and Subtitle	5. Report Date
Benchmarking Computer Vision-Based Approaches to Derive Engineering-Oriented Condition from Existing UDOT Assets Data	February 2025
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Shailaja Ratna Pandey, Mohsen Zaker Esteghamati	

9. Performing Organization Name and Address	10. Work Unit No.
Utah State University Department of Civil and Environmental Engineering Old Main Hill Logan, Utah, 84321	5H094 30H
	11. Contract or Grant No. 25-8073

12. Sponsoring Agency Name and Address	13. Type of Report & Period Covered
Utah Department of Transportation 4501 South 2700 West P.O. Box 148410 Salt Lake City, UT 84114-8410	Final July 2024 to Dec 2025
	14. Sponsoring Agency Code PIC No. UT24.206

15. Supplementary Notes

Prepared in cooperation with the Utah Department of Transportation and the U.S. Department of Transportation, Federal Highway Administration

16. Abstract

Condition assessment of how transportation infrastructure supports safe and reliable road and highway operation. Departments of Transportation across the country rely heavily on manual inspections, which are time-consuming and costly. This study evaluated whether modern computer vision (CV) methods can support traffic sign condition assessment along Utah highways. High-resolution roadway images collected using a camera-mounted vehicle were curated and annotated for three sign types (regulatory, warning, and guide) and four defect conditions (fading, delamination, missing letters/symbols, and broken signs) based on the Manual on Uniform Traffic Control Devices (MUTCD) standards. This study compared two different CV algorithms of YOLO11 and RT-DETR for traffic-sign detection and defect classification. Overall, the CV models showed promising performance for defect cases where an adequate number of training data existed. For example, for fading, YOLO11 and RT-DETR achieved 75% F1 on the validation. Binary classification of delamination (i.e., delamination versus no delamination) yielded similar performance for both models (68% F1). In contrast, the models showed poor performance to identify missing letters/symbols due to texture overlap with delamination and a limited number of annotated sign images with such defects. The results suggested that data quality and label definition had a greater impact on model performance than the choice of algorithms for the studied models.

17. Key Words	18. Distribution Statement	23. Registrant's Seal
Traffic sign condition assessment, Computer vision, Object detection, YOLO11, Vision transformers, MUTCD	Not restricted. Available through: UDOT Research Division 4501 South 2700 West P.O. Box 148410 Salt Lake City, UT 84114-8410 www.udot.utah.gov/go/research	N/A

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	69	N/A

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

CV	Computer Vision
AI	Artificial Intelligence
DOT	Department of Transportation
HOG	Histogram of Oriented Gradients
SVM	Support Vector Machine
HOG+C	HOG plus Color Histograms
DNN / DNNs	Deep Neural Network(s)
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
YOLO	You Only Look Once
YOLO11	YOLO model version (v11)
RT-DETR	Real-Time Detection Transformer
mAP	Mean Average Precision
TT100K	Traffic sign dataset (TT100K)
GAN	Generative Adversarial Network(s)
ViT	Vision Transformer(s)
GTSDB	German Traffic Sign Detection Benchmark
GTSRB	German Traffic Sign Benchmark
BTSC	Belgium Traffic Sign Classification
rMASTIF	Croatian Traffic Sign dataset
RIECNN	Real-Time Enhanced CNN
LiDAR	Light Detection and Ranging
UAV	Unoccupied/Unmanned Aerial Vehicle(s)
ADAS	Advanced Driver-Assistance Systems

EXECUTIVE SUMMARY

Departments of Transportation (DOTs) in the US require periodic, reliable condition data to keep roads safe and allocate maintenance funds efficiently. However, most DOTs still rely on manual visual inspection methods, which require inspectors to physically visit sites and make judgments based on the guidelines and their engineering experience. This process is time-consuming and costly, and it exposes inspectors to risk. In addition, the inspection process scales poorly across large road networks with many different safety assets. The problem is particularly more urgent for traffic signs because loss of sign visibility and legibility can directly affect how drivers and pedestrians interpret roadway information.

This study focused on whether computer vision (CV) models can support scalable condition assessment of traffic signs in Utah using highway images captured by a vehicle-mounted roof camera and a Manual on Uniform Traffic Control Devices (MUTCD)-aligned labeled dataset. The dataset covered three sign types, regulatory, warning, and guide, and four defect conditions, fading, delamination, missing letters/symbols, and broken signs. The work followed two goals: (i) to develop an automated pipeline that processed large-scale imagery to detect signs and classify multiple defect types, and (ii) to compare CV models for the given task based on engineering-oriented metrics. To meet these objectives, the study implemented a framework that preprocessed these highway images, then trained and tested You Only Look Once (YOLO11) and Real-Time Detection Transformer (RT-DETR) models using identical training, validation, and test splits, and finally evaluated results with per-class precision, recall, F1-score, normalized confusion matrices, and qualitative inference examples.

The results showed that for sign-type classification, YOLO11 achieved F1-scores of 84% for regulatory and guide signs and 76% for warning signs on the test set. For fading, both models achieved identical validation F1-score of 75%, but YOLO11 generalized better on the test set with an F1-score of 67% as compared to RT-DETR with an F1-score of 50%. For three-class delamination severity, results suggested modest class separation. The performance of both models was highest for the dominant delamination class of "<25%" with F1-score being 56% for YOLO11 and 55% for RT-DETR, whereas the F1-score was lower for the "no delamination" class (29%/42%) and the ">25%" delamination (25% for both). Overall, the results indicate that

distinguishing severity near the 25% threshold remains challenging under the current labeling and data distribution. However, reformulating the delamination as a binary problem (i.e., delamination versus no delamination) improved performance for both models and yielded similar F1-scores of 68% for the delamination class. Missing letters/symbols remained challenging for both detectors (F1 = 20% for YOLO11 and 24% for RT-DETR), consistent with texture overlap with delamination. Broken-sign classification in this dataset primarily included small, localized damage on the sign face, and the test split contained only three positive broken-sign instances. As a result, the F1-score was highly sensitive to a small number of errors, which limited the reliability of the reported performance. Overall, both models achieved their highest performance for fading, with consistent results across the validation and test sets.

Across all defect tasks, the study found that labeled data limitation, rather than model architecture, accounted for models' poor performance. Overlapping and coexisting defects, subjective severity thresholds (e.g., <25% versus >25% delamination), visually subtle and rare damage, and dense scenes with multiple signs capped performance. Strategies such as targeted sampling with offline augmentations and class weighting were applied to address class imbalance, but they did not fully resolve training issues due to the lack of sufficient images with clear defect evidence and definitions.

Based on these results and observations, the study indicates that CV can support traffic sign condition assessment when sufficiently large, consistently labeled training data are available. The strongest performance was achieved for defects with clear visual cues and adequate representation in the dataset (e.g., fading). Therefore, future work should prioritize improving training data by collecting targeted data on rare and subtle defects from close-range views, defining and annotating defects consistently, with definitions and annotations agreed upon by a committee of engineers and inspectors, and by publishing the dataset, annotation rules, and training code to enable replication and continued improvement. In conclusion, this study establishes a practical benchmark for assessing traffic sign conditions from highway imagery and identifies the data and labeling improvements needed to support future monitoring workflows.

1.0 INTRODUCTION

1.1 Problem Statement

Condition assessment of transportation infrastructure is important in order to ensure safety, reliable mobility, optimize maintenance management, and reduce the likelihood of accidents. Conventional methods include a periodic manual visual inspection method where engineers/inspectors physically visit the site, assessing the condition based on the standard guideline. However, this process is labor-intensive, costly, time-consuming, and difficult to scale for large roadway networks.

Traffic signs form an important safety asset in the transportation infrastructure, impacting visibility and legibility for roadway users, from drivers to pedestrians. Degraded sign conditions reduce legibility, thereby increasing the likelihood of delayed or incorrect road-user decisions and elevating safety risks. Therefore, identifying and documenting the condition of these signs becomes of utmost importance. Nevertheless, with the traditional approach, this process can be time-consuming and inconsistent, especially along high-speed and hard-to-access corridors. Therefore, this study addresses that gap by applying computer vision (CV) algorithms to automatically detect traffic signs from the highway imagery and assess their condition, with the goal of providing a scalable and data-driven framework that can support frequent monitoring and proactive maintenance for transportation agencies such as the Utah Department of Transportation (UDOT).

1.2 Objectives

This project aims to address the following objectives:

1. Develop automated and reliable CV-based methods to process and analyze large-scale highway imagery across Utah to detect traffic signs and assess their condition across multiple defect types.

2. Evaluate and compare the performance of multiple CV algorithms in terms of the Manual on Uniform Traffic Control Devices (MUTCD) guidelines; engineering-oriented

condition assessment, including multi-defect classification performance; and suitability for deployment in agency workflows.

1.3 Scope

This research focuses on condition assessment of highway traffic signs in Utah using CV. The primary dataset consists of high-resolution highway images obtained from Pathway (Pathway Services, 2025) in collaboration with UDOT for four regions across Utah (Ogden, Salt Lake, Orem, and Richfield). The study considers regulatory, warning, and guide signs with condition labels capturing four defect types: fading, delamination, missing letters/symbols, and broken signs. The main tasks include preparing and cleaning the dataset, generating inputs suitable for the two automated detection and assessment pipelines: a Convolutional Neural Network (CNN)-based, You Only Look Once (YOLO) model, and a transformer-based Real-Time Detection Transformer (RT-DETR) model. These models are trained and evaluated on the prepared dataset to compare their performance across these defects and to assess their suitability for large-scale deployment in transportation asset management.

1.4 Outline of Report

This report is organized into six chapters. Chapter 1 introduces the study by outlining the background and motivation for automated traffic sign condition assessment, stating the research problem and objectives, and defining the scope of the study. Chapter 2 reviews existing computer vision research in civil infrastructure, with emphasis on traffic sign detection and recognition. Chapter 3 presents the dataset development workflow, including data collection, preprocessing, annotation, and exploratory analysis. Chapter 4 describes the model development and evaluation, including training and validation procedures, performance metrics, and comparative results for sign detection and defect classification. Chapter 5 summarizes the key findings and discusses the original research objectives and their implications for transportation asset management. Finally, Chapter 6 offers recommendations and guidance for the practical implementation of automated traffic sign condition assessment and suggests directions for future research.

2.0 LITERATURE REVIEW

2.1 Overview

This section provides background on transportation infrastructure condition assessment and emphasizes the importance of well-maintained traffic signs within highway networks. The section then provides motivation for automated, CV-based approaches by outlining the limitations of current manual inspection practices, which helps formulate the research problem and states the objectives and scope of the study, with a focus on Utah highways. Finally, this section also reviews how other researchers have applied similar CV frameworks to assess infrastructure, identifies gaps and limitations in the existing literature on traffic sign condition assessment, and explains how the present study addresses those gaps.

2.2 Background

Traffic signs are significant transportation safety assets within roadway networks that provide safer traffic environments through regulating, warning, or guiding road users (Koyuncu & Amado, 2008). The effectiveness of these signs, however, depends on whether they are clearly visible and legible (Khalilikhah & Heaslip, 2016). A faded, deteriorated, or damaged sign, regardless of any kind, that impairs readability can contribute to confusion and an increased risk of a crash. Traditionally, these assessment processes have relied on manual inspections, in which engineers physically visit sites and evaluate their condition using prescribed guidelines (e.g., MUTCD) and engineering judgment. While this method has long been the industry standard, it is inherently labor-intensive, time-consuming, and costly. Furthermore, inspecting hard-to-reach or hazardous locations introduces significant safety risks to inspectors (Seo et al., 2015).

Traffic signs possess distinct visual features, such as standardized shapes, colors, and high-contrast symbols (Greenhalgh & Mirmehdi, 2012), making them well-suited for automated detection. Recent advances in sensing technologies have created opportunities to improve inspection efficiency and safety. With the growing availability of high-resolution image and video data, enabled by drones (Seo et al., 2018), Light Detection and Ranging (LiDAR) (X. Zhang et al., 2025), and other sensing technologies, there is an increasing interest in automating

asset condition assessment using CV algorithms. Images and videos are the two major modes of data analyzed by CV, which captures visual information similar to that obtained by human inspectors (Spencer et al., 2019). CV-based approaches can then translate this visual data into pixel-level information to detect patterns and extract features with high precision. Such a vision-based approach, combined with cameras and unattended aerial vehicles (UAVs), offers the potential for rapid, automated inspection and monitoring of civil infrastructure condition assessment (Zhou et al., 2023).

2.2.1The Role of CV in Civil Engineering

In recent years, CV has transformed many technology-driven domains, such as autonomous driving, robotics, and surveillance, and its potential for solving challenges in civil engineering is now being fully realized. By enabling automated extraction of visual features from images and videos, CV algorithms can replicate key aspects of human inspection with enhanced speed, consistency, and safety. These systems can significantly reduce the operational burden on inspectors by offering ease of deployment, greater flexibility, lower inspection costs, and minimized exposure to hazardous environments (Feng & Feng, 2018).

The applications of CV in civil engineering are both diverse and rapidly expanding. Examples include post-disaster assessments (Khajwal et al., 2023), where UAVs collect aerial imagery of damaged infrastructure; automated defect detection in bridges and pavements (Koch et al., 2015); quality control during construction (Ettalibi et al., 2024); and progress monitoring through real-time video analysis (Yang et al., 2023). These technologies not only streamline inspection processes but also offer high-resolution data that supports more informed decision-making (Hoskere et al., 2018). A growing body of research has been focusing on an automated post-disaster damage assessment pipeline using aerial or satellite imagery. For instance, a study by Soleimani et al., (2024) proposed a framework that estimates socioeconomic loss metrics at the individual building level, showcasing the potential to enhance the rapid aerial damage assessment models, which ultimately leads to a more efficient and targeted disaster response and recovery efforts. Another related study introduced an uncertainty-aware ensemble of deep segmentation models that uses post-hazard satellite imagery to segment and classify building damage. This framework supports rapid initial damage assessments prior to expert dispatch or in

regions without capacity for large-scale assessments (Soleimani-Babakamali et al., 2023). These studies support the viability of a CV-based, image-driven condition assessment framework, which motivates similar approaches for transportation infrastructure.

2.2.2 CV in Transportation Asset Management

Within the transportation domain, CV has been widely applied to tasks such as traffic monitoring and control, incident detection and management, road condition monitoring, autonomous driving, and traffic sign recognition (Dilek & Dener, 2023). Among these applications, traffic sign detection and classification have received significant attention due to their critical role in promoting roadway safety, guiding road user behavior, and ensuring compliance with regulatory standards (Gonzalez et al., 2011). Several studies have utilized hand-crafted feature extraction methods and non-linear models such as histogram of gradients (HOG), and support vector machine (SVM), among many others. Houben et al., (2013) proposed traffic sign detection approaches such as the Viola-Jones detector based on Haar features, and a linear classifier relying on HOG descriptors. Similarly, Balali & Golparvar-Fard (2016) evaluated the performance of three CV algorithms 1) Haar-like features with Cascade classifiers, 2) HOG with multiple one-versus-all SVM classifiers, and 3) a variant of the second method with histograms of colors concatenated to HOG (HOG+C) to detect and classify traffic signs in the presence of cluttered backgrounds and occlusions. One problem with these methods is their limited representation power, which can lead to an overlap between the classes of objects, thus adversely degrading classification performance (Habibi Aghdam et al., 2016). In recent years, this limitation has motivated a shift toward deep neural networks (DNNs), which learn task-specific feature hierarchies directly from data rather than relying on manually designed features. Within this family, convolutional neural networks (CNNs) have become a dominant choice. However, traffic scenes often involve clutter, occlusion, and wide variations in scale, and these conditions benefit from stronger modeling of global context. This need has also encouraged the adoption of transformer-based architectures and hybrid CNN–transformer designs for traffic sign detection and recognition.

Over the years, researchers have proposed a variety of CNN- and transformer-based models for traffic sign detection, recognition, and condition assessment under diverse operational

conditions. For example, Zhu et al., (2016) developed a two-stage traffic sign detection and recognition framework that uses a fully convolutional network (FCN) to guide traffic sign proposals and a CNN classifier to recognize traffic signs. The FCN-guided approach replaces standard EdgeBoxes, which helps their system generate discriminative candidate regions, enabling fast, and accurate detection in complex traffic scenarios. Another study introduced a traffic sign detection method based on YOLOv5 and Swin-Transformer that applies lightweight feature enhancement, channel attention and adaptive feature fusion to improve small sign detection. The model was evaluated on the TT100K and DFG datasets and reported mAP values above 75%, on both datasets, indicating high accuracy and real-time performance (Qian & Wang, 2024). Beyond improving recognition accuracy, several studies have focused on architectural modifications to address multi-scale and small-object challenges in traffic sign detection. To address this, Wang et al. (2023) proposed an improved feature pyramid model, named AP-FPN, integrated with the YOLOv5 model, which strengthens multi-scale feature representation and improves detection performance while maintaining real-time speed. Similarly, another study utilized a lightweight model for traffic sign detection based on real-time detection transformer (TSD-DETR). The network includes a feature extraction module composed of multiple convolutional blocks that extract multi-scale features, together with a small-object detection module and a detection head that extracts shallow features to improve detection of small signs, thereby improving detection accuracy (L. Zhang et al., 2025). These studies highlight the importance of multi-scale feature representation for reliable real-time traffic sign detection.

Another key challenge for the sign detection and recognition is maintaining robust performance under challenging weather conditions, which can significantly degrade model accuracy. This issue was addressed in a study that proposed a CNN-based challenge classifier, an encoder-decoder enhancement network (Enhance-Net), and two separate CNN architectures for sign detection and classification, which effectively ensured the enhancement of the sign regions and was shown to improve detection and recognition performance under challenging conditions (Ahmed et al., 2022). Fredj et al. (2023) developed a CNN-based framework for traffic sign recognition using two typical networks, VGG16 and optimized LeNet for Tunisian road signs under diverse and challenging illumination conditions. Their model achieved 99.6% accuracy on the base dataset and 99.05% accuracy when evaluated at different challenging scenarios, which

outperformed several existing methods. Similarly, another study introduced a Real-Time Enhanced CNN (RIECNN) for traffic sign recognition that uses multiple, diverse traffic sign datasets and outperforms state-of-the-art architectures in terms of recognition rate and execution time. The experimental results show that the accuracy achieved is 99.75% for German Traffic Sign Benchmark (GTSRB), 99.25% for the Belgium Traffic Sign Classification (BTSC) and 99.55% for the Croatian Traffic Sign (rMASTIF), showcasing the model's robustness in traffic sign recognition despite brightness and contrast variations (Abdel-Salam et al., 2022). Together with illumination-focused recognition frameworks, such methods demonstrate that handling adverse imaging conditions is critical for robust traffic sign perception.

These studies demonstrated feasibility of traffic sign detection and classification with modern CV models under well-annotated training datasets. However, the database for traffic signs is limited across the world, making it a challenge. To mitigate this, Dewi et al. (2021) employed Generative Adversarial Networks (GANs) to generate realistic synthetic traffic sign images. Experiments with YOLO, YOLOv3, and YOLOv4 showed that the model achieved higher recognition performance when trained on mixed real–synthetic data than those trained on real images alone. Alongside data augmentation and synthesis, recent work has also examined alternative architectures to cope with limited and imbalanced datasets. A study investigated Vision Transformers (ViTs) for traffic sign detection and found that with the small dataset and high class imbalance, these conventional models offer limited performance gain. To address this, the authors proposed a hybrid Pyramid Transformer that uses a hierarchical architecture with atrous convolutions to capture both local and global information and learn a multi-scale feature representation for traffic signs of varying sizes. This proposed model achieved a mean average precision of 77.8%, outperforming existing state-of-the-art methods on the German Traffic Sign Detection Benchmark (GTSDB) (Manzari et al., 2022).

Despite progress in automated inspection technologies, the existing literature has primarily emphasized object detection and recognition, such as detecting the presence of traffic signs or evaluating retroreflectivity (Güney et al., 2022; Spencer et al., 2019). For example, Steele et al. (2023) demonstrated a LiDAR-based method for retroreflectivity assessment with potential for predictive maintenance, whereas González et al. (2011) developed the VISUAL Inspection of Signs and panEls (VISUALISE) system for automatic traffic sign inspection at a

highway, albeit this framework was limited to visibility and presence detection. While effective for recognition and navigation, these methods do not evaluate sign condition or defects, leaving degradation modes such as fading, delamination, missing symbols, and breakage largely unexplored. This gap underscores the need for a more comprehensive, multi-defect condition assessment framework, which is the focus of this study.

To address this gap, this study goes beyond detection to classify traffic signs both by functional type (Regulatory, Warning, and Guide) and by condition state, incorporating four critical defects: missing letters/symbols, delamination, fading, and broken signs, as defined in the MUTCD guideline. This is implemented using YOLO11 (Ultralytics, 2024), which is a recent advancement in the YOLO real-time object detection series (Redmon et al., 2016). YOLO's architecture enables fast and accurate detection and classification, making it particularly valuable for transportation asset management, where large volumes of roadside imagery from mobile mapping systems, dashcams, or UAVs must be processed efficiently and reliably (Lin et al., 2024). In addition to this CNN-based baseline, the study evaluates a Transformer-based detector, RT-DETR. RT-DETR is an end-to-end object detection architecture that uses a CNN backbone with an efficient hybrid encoder and a transformer decoder with auxiliary prediction heads (Zhao et al., 2024a). The two models are compared in terms of their defect-detection and condition-classification performance, inference speed, and practical suitability for large-scale deployment in transportation asset management workflows.

Table 2.1: Summary of cited traffic sign literature, including application, approach, model type, and key takeaway

Author	Application	Approach	Type	Main takeaway
González et al. (2011)	Traffic sign/panel inspection	CV pipeline mounted on vehicle; automated inspection at driving speeds	Traditional CV inspection system	Shows early "inspection-at-scale" systems, but targets visibility/presence-like inspection rather than multi-defect condition assessment.
Houben et al. (2013)	Traffic sign detection benchmarking	Baselines include Viola–Jones (Haar) and HOG-based linear classifier	Hand-crafted + classical ML baselines	Establishes classic hand-crafted baselines and evaluation culture for sign detection under real-world scenes.
Balali et al. (2016)	Sign detection/classification in clutter/occlusion	Haar-like + Cascade, HOG + one-vs-all SVM, and HOG+Color	Hand-crafted + SVM / cascade	Reviews hand-crafted features for traffic sign detection under clutter and occlusion
Zhu et al. (2016)	Detection + recognition pipeline	FCN-guided proposals + CNN classifier (two-stage)	CNN (two-stage)	Demonstrates early deep pipelines that replace proposal heuristics with learned proposal generation.
Qian & Wang (2024)	Small traffic sign detection	YOLOv5 + Swin-Transformer hybrid; feature fusion/attention	Hybrid CNN–Transformer detector	Provides an example of hybrid designs for small-object signs.
Wang et al. (2023)	Multi-scale traffic sign detection	Replace YOLOv5 FPN with AF-FPN to improve multi-scale performance	CNN detector with improved neck/FPN	Multi-scale feature pyramids drive gains for small and far signs in highway scenes.
Zhang et al. (2025)	Small traffic sign detection	"TSD-DETR" style: CNN feature extractor + transformer detection design tuned for small objects	Transformer-based detector (DETR family)	Proposed DETR-style model can be adapted for small sign detection.

Ahmed et al. (2022)	Robust detection under challenging weather	Challenge classifier + Enhance-Net (encoder–decoder enhancement) + detection/classification modules	CNN-based robustness framework	Addresses weather/visibility challenges and motivates why raw highway imagery quality limits learnability.
Fredj et al. (2023)	Traffic sign recognition	CNN recognition with VGG16 and optimized LeNet on Tunisian signs	CNN classifier	Reinforces recognition success under uncontrolled environment such as weather conditions, complex background, etc.
Abdel-Salam et al. (2022)	Traffic sign recognition	RIECNN (image enhancement + CNN) for real-time recognition	CNN classifier + enhancement	Shows recognition can hit very high accuracy with the right preprocessing, and enhancement and benchmark datasets.
Dewi et al. (2021)	Data scarcity mitigation	GAN-generated synthetic traffic signs + YOLO variants	Data synthesis + CNN detector	Limited datasets push researchers toward augmentation/synthesis strategies.
Manzari et al. (2022)	Traffic sign detection with imbalance/small data	Vision transformer analysis; propose Pyramid Transformer with multi-scale design	Transformer / hybrid ViT-style	ViTs may not win on small, imbalanced sign datasets without architectural changes, such as the one proposed in the study: Pyramid Transformer
Steele et al. (2023)	Retroreflectivity / asset management	Mobile LiDAR-based retroreflectivity evaluation for predictive maintenance	LiDAR-based measurement + analytics	Proposes methodology to measure retroreflectivity of traffic signs as a condition indicator
Güney et al. (2022)	Traffic sign detection	YOLOv5 implementation for sign/road object detection	YOLOv5 based detector	Targets detecting signs/road objects for Advanced Driver Assistance Systems (ADAS)

2.3 Summary

This chapter reviewed the current state of CV applications in civil infrastructure and transportation, with a particular focus on traffic sign detection, recognition, and condition assessment. It outlined how CV evolved from handcrafted rules and mathematical models to data-driven learning with DNNs. Researchers now use CV as a practical tool to replace or supplement manual visual inspections in civil engineering, including post-disaster damage assessments and structural condition monitoring.

The chapter highlighted applications of CV in the transportation domain. It summarizes how researchers apply CNN- and transformer-based models for traffic sign detection and recognition, and how these models handle multi-scale signs in real-time settings. The chapter also described major challenges in this process, such as a large variation in sign scale and the need to maintain robust performance under adverse lighting and weather conditions, and it showed how different studies addressed these issues through improved architectures and enhancement frameworks. In addition, the chapter discussed work that uses synthetic images and data augmentation to mitigate limited dataset size and class imbalance, especially when training modern deep models for traffic sign tasks.

Overall, the reviewed studies demonstrate that data-driven CV methods can achieve accurate automatic traffic sign detection and recognition, thereby reducing the time, labor, expertise, and safety burdens associated with manual inspection. However, most existing research focuses on detection and recognition on curated benchmarks and gives relatively little emphasis to multi-defect, condition-focused assessment that aligns with MUTCD-style defect categories. Therefore, in response to these gaps, the next chapter describes the data collection and dataset preparation for four defect types (fading, delamination, missing letters or symbols, and broken signs) and compares two different CV models, YOLO11 (CNN-based) and RT-DETR (transformer-based), to determine which model better captures engineering-relevant defects and offers greater potential for long-term deployment in transportation asset management.

3.0 DATA COLLECTION AND PROCESSING

3.1 Overview

This chapter describes the highway imagery data used in this study and its geographic coverage of the Utah roadway network. It explains how the study identified and labeled traffic signs by type and defect category, and how quality control was applied to the annotations. The chapter then discusses the preprocessing steps used to transform raw images into model-ready inputs, including dataset preparation, splitting into training, validation, and test sets, and the annotation process performed.

3.2 Data Collection

The primary dataset for this project consists of high-resolution highway images obtained from Pathway (Pathway Services, 2025) in collaboration with UDOT. These images were collected using a specialized inspection vehicle equipped with visual and sensor systems that capture data at standard highway speeds. The study uses data from four regions in Utah: Ogden, Salt Lake, Orem, and Richfield. Figure 3.1 summarizes the distribution of traffic signs rated good, fair, or poor condition across these regions. Overall, signs rated 'good' account for over 90% of the inventory, whereas 'poor' signs represent less than 2%. This skewed distribution indicates that severely deteriorated signs are rare in the dataset, creating a pronounced class-imbalance challenge for subsequent defect detection and classification.

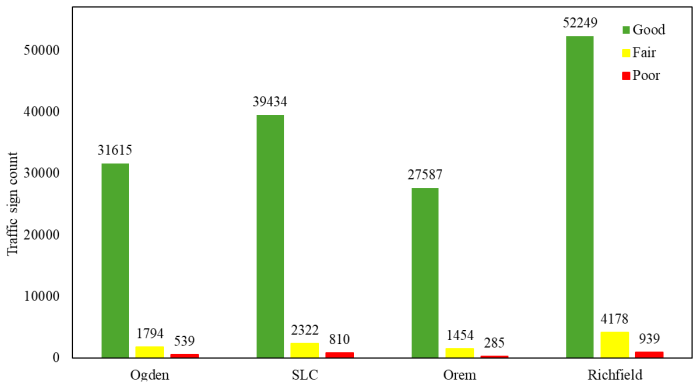

Figure 3.1: Distribution of traffic sign condition ratings (good, fair, poor) across four UDOT regions in Utah based on Pathway Services data (2025)

Raw images were collected at high resolution, with typical dimensions ranging from approximately 8250 × 2198 pixels to 9000 × 2176 pixels, capturing the full view of the roadway and roadside assets (Figure 3.2). To reduce computational expenses during training while preserving sufficient detail, each image was split into three segments, yielding segment sizes of about 2750 × 2198 pixels or 3000 × 2176 pixels, depending on the original frame dimensions (Figure 3.3). Images with no visible traffic signs were either discarded or selectively retained as background images. Retaining a subset of background images improves model generalizability by training the network to distinguish between sign-containing and sign-free roadway scenes. From the curated dataset (sign-containing images and selected background scenes), a subset of images was carefully selected for the study and partitioned into training and validation sets at 85% / 15% to ensure adequate representation of all traffic sign classes. The training set helps the model learn from these labeled examples, whereas the validation set guides hyperparameter tuning and model selection by evaluating performance on a separate subset not used for training. Finally, a small independent test set, which is held out from training and tuning, was reserved for final evaluation.

Figure 3.2: Example of a raw high-resolution roadway image showing the full highway scene and traffic signs

Figure 3.3: Example of a split segment derived from the original high-resolution roadway image

Traffic signs were categorized into three types: regulatory, warning, and guide, while four defect conditions were recorded for each type: missing letters/symbols, delamination, fading, and broken signs. These classifications follow the criteria outlined in the MUTCD (Federal Highway Administration, 2023), which are further summarized in Table 3.1. Using this labeling scheme, each image was manually annotated in Label Studio (Tkachenko et al., 2020), with the resulting annotations exported as an XML file. These XML files store the bounding box coordinates and corresponding labels, serving as structured input for training and evaluating the CV models.

Table 3.1: Traffic Sign Condition Annotation Guidelines (Per MUTCD Standard)

Defects	Good		Fair		Poor	
	MUTCD	**Adopted Label**	**MUTCD**	**Adopted Label**	**MUTCD**	**Adopted label**
Letter and Symbol	Sign messages are readable; all intended letters and symbols are present.	**No missing letters/ symbols**	-	-	Missing or illegible any letters or symbols.	**Missing letters/ symbols**
Delamination	Sign face has no delamination (peeling) of legend or background material.	**No delamination**	Sign face has less than 25% delamination of legend or background material.	**<25% delamination (legend/background)**	Sign face has more than 25% delamination of legend or background material.	**>25% delamination (legend/background)**
Fading	-	**No fading**	-	**Fading**	-	**-**
Broken	-	**Not broken**	-		-	**Broken sign**

Table 3.1 summarizes the condition-annotation scheme used in this study. For each defect, the MUTCD was interpreted and mapped to discrete severity levels (good, fair, poor), and short labels were adopted for use in the annotation and modeling pipeline as follows:

- **Letters and symbols:** A sign is in good condition if all intended letters and symbols are present and legible (i.e., “No missing letters/symbols”). Any missing or illegible letter or symbol moves the defect directly to the poor category (i.e., “Missing letters/symbols”). The MUTCD does not define an intermediate fair level for this defect, thus this defect is treated as binary.

- **Delamination:** A sign with no peeling of legend or background material falls in the good category (i.e., “No delamination”). If delamination affects less than 25% of the legend or background area, it is classified as fair (i.e., “<25% delamination”). If delamination exceeds 25%, it is classified as poor (i.e., “>25% delamination”).
- **Fading:** A sign is labeled as good when no noticeable fading is present (i.e., “No fading”), and as fair when fading is clearly visible (i.e., “Fading”). The guideline does not specify a separate poor level for this defect, thus it is also treated as binary.
- **Broken:** Signs with no structural damage are good (i.e., “Not broken”), and any break or fracture of the sign panel is labeled as poor (i.e., “Broken sign”). As the guideline does not provide an intermediate fair level, this defect classification is binary.

The “Adopted label” column in Table 3.1 lists the exact labels used during annotation. These descriptive labels allow engineers to better understand the extent to which a deficiency is present on a sign (e.g., “Missing letters/symbols”) rather than only seeing an overall rating (i.e., good/fair/poor), and support better prioritization of maintenance and replacement. Figure 3.4 shows several examples of the multi-label annotations that capture both traffic sign type (regulatory, warning, or guide) and its corresponding condition attributes (missing letters/symbols, delamination, fading, and broken) in accordance with the MUTCD guidelines.

(a)

(b)

(c)

Figure 3.4: Examples of multi-label annotations for traffic signs, including (a) a regulatory sign, (b) a warning sign, and (c) a guide sign, each with its corresponding condition attributes.

3.3 Summary

This chapter presented the highway imagery dataset and its coverage across the Utah roadway network. It described how the study identified traffic signs, assigned sign-type and defect labels, and applied quality checks to keep annotations consistent. The chapter also outlined the preprocessing workflow that converts raw imagery into model-ready inputs, including dataset organization and the creation of training, validation, and test splits.

The study utilized high-resolution roadway images from Pathway Services in collaboration with UDOT. The labeling scheme followed MUTCD guidance and includes three sign types: (i) regulatory, (ii) warning, and (iii) guide and four defect categories: (i) missing letters/symbols, (ii) delamination, (iii) fading, and (iv) broken signs. The study treated missing letters/symbols, fading, and broken as binary defects, since the MUTCD does not define an intermediate level, and assigned delamination severity using three levels based on the affected sign-face area (no delamination, <25%, and >25%). Annotations were performed in Label Studio and exported as XML files containing bounding boxes and labels, which helped store coordinates and defect/type labels for model training and testing.

4.0 MODEL TRAINING AND EVALUATION

4.1 Overview

This chapter presents the modeling workflow for traffic-sign defect classification. It explains how the study handled class imbalance, trained and tuned YOLO11 and RT-DETR models, and evaluated performance using precision, recall, F1-score, confusion matrices, and bar plots.

4.2 Class Imbalance

Based on the annotated dataset, the frequency of different traffic sign types and defects across the training, validation, and test splits is summarized in Figures 4.1 and 4.2, respectively. Figure 4.1 shows that guide signs have the highest number of instances across the training, validation, and test sets, while regulatory signs have the fewest. Figure 4.2 further shows that several condition labels are unevenly distributed, with defect cases occurring less frequently than their corresponding non-defect labels. This skew motivates the class-imbalance considerations, as these distributions directly affect model training and evaluation.

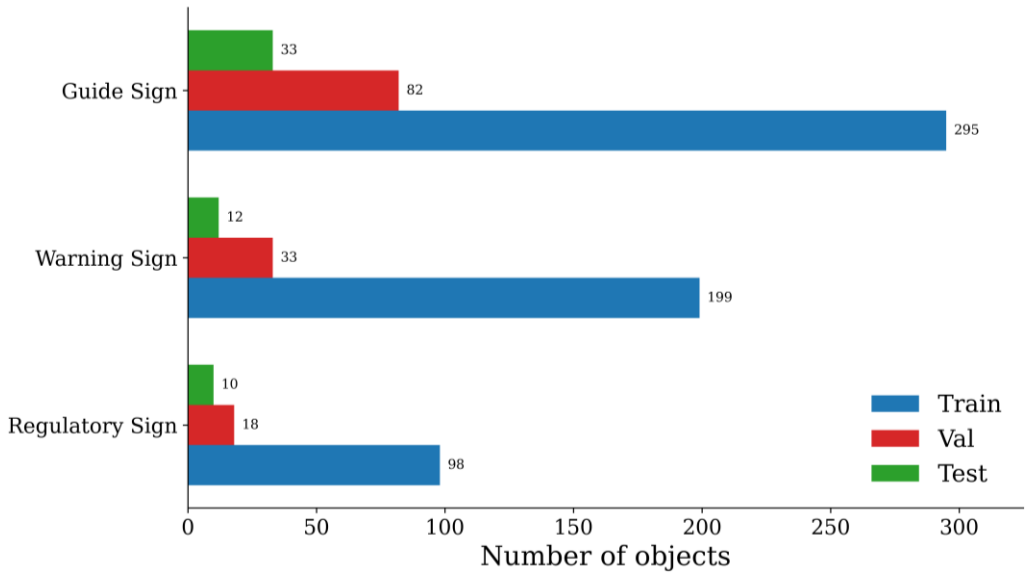

Figure 4.1: Distribution of training, validation, and test samples across traffic sign classes

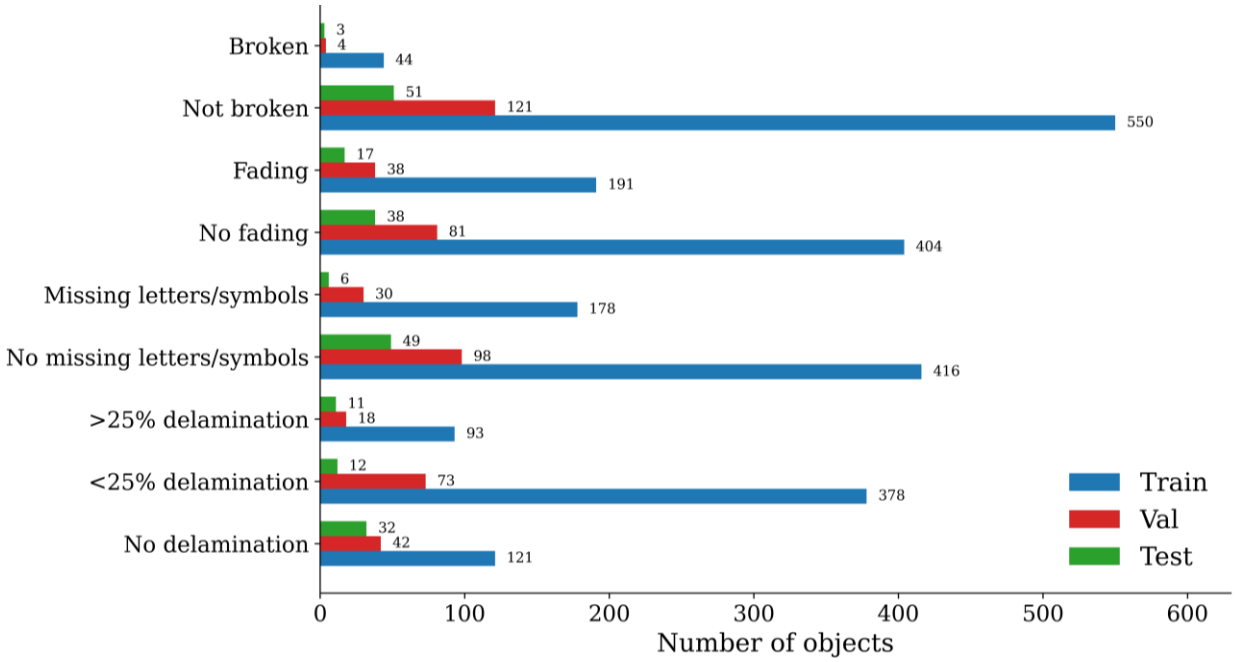

Figure 4.2: Distribution of training, validation and test samples across traffic sign condition labels

Class imbalance is a common challenge in CV tasks. When a minority class is strongly under-represented, the models tend to bias their predictions toward the majority class and fail to learn reliable decision boundaries for the rare but important cases. This limitation is consequential, especially in safety-critical settings such as defect classification, where missed classification can undermine the system's functionality. In this dataset, imbalance is observed across both sign types and defect labels; however, their impacts are more significant for certain defect labels.

There are several reasons that a moderate imbalance in sign-type classes poses less risk to model performance. First, sign-type classification primarily relies on distinctive color and geometric shape defined in traffic control standards (Greenhalgh & Mirmehdi, 2012); MUTCD, 2023). Regulatory signs typically use white or red backgrounds with black or white legends and standard shapes such as rectangles and octagons (e.g., STOP); warning signs generally use yellow backgrounds with black legends and diamond or rectangular shapes; and guide signs primarily use green, blue, or brown rectangular panels. Second, modern object detectors such as YOLO11 (Ultralytics, 2024) and RT-DETR (Zhao et al., 2024) build on backbones pre-trained on large-scale datasets like Microsoft Common Objects in Context (MS-COCO), which include a

wide variety of objects with diverse shapes, colors, and at least one traffic-related class (e.g., stop sign). This pretraining helps the networks learn general visual features (e.g., edges, color patterns, and shapes) that help distinguish objects.

Within the defect classification tasks, the label distribution exhibits severe class imbalance, with comparatively only a few positive defect instances. For example, in the training set, “no fading” instances occur approximately twice as frequently as “fading” instances, even though identifying fading is a primary objective of this study. Beyond the imbalance in counts, the defect task is inherently more critical because the defect labels depend on subtle texture- and intensity-based cues on the sign face, rather than the color and geometric cues that govern sign-type classification. Consequently, the training process must expose the model to sufficient examples from each defect subclass to enable learning discriminative patterns and effectively distinguishing between “no defect” and “defect” conditions, rather than biasing toward the majority “no defect” prediction.

4.2.1 Strategies to Address Data Imbalance

Severe class imbalance across defect labels can bias the model predictions toward the majority “no defect” classes. To address this issue, the study applies two strategies: (1) offline data augmentation, and (2) class weighting during training. The following sections describe each strategy in detail and examine its impact on defect classification performance.

4.2.1.1 Data Augmentation

Data augmentation is a common strategy in deep learning that generates new training examples by applying controlled transformations to existing images, such as rotation, scaling, translation, cropping, and color jitter (Shorten & Khoshgoftaar, 2019). Augmentation increases the effective size of the training set and exposes the model to a broader range of appearance variations that resemble real-world conditions, which improves model robustness and generalization.

Augmentation techniques usually follow two main strategies: online and offline (Wagner et al., 2023). Online augmentation applies random transformations to each image on-the-fly

during training, with the augmented images existing only in memory for that training step and not occupying extra disk space, whereas offline augmentation generates transformed images and their corresponding labels in advance and saves them to disk as a new expanded training set.

In this study, both strategies are implemented. Offline augmentation targets severely underrepresented defect classes by generating numerous additional examples of rare defects, thereby improving class balance and reducing bias toward the dominant "no defect" label. Online augmentation is applied on-the-fly during training, but is intentionally mild (e.g., rotation up to 1° and translation up to 0.01) to introduce small viewpoint and alignment variability without affecting the appearance of the defect. These augmentation strategies are applied only to the training set, while the validation and test sets are left unaltered, to ensure that performance on these sets reflects how well the model generalizes rather than how well it performs on the augmented data. Figure 4.3 illustrates a simple example of offline augmentation using a small rotation of the original roadway image.

(a) **(b)**

Figure 4.3: Example of rotation-based augmentation: (a) original image; (b) image after a slight rotation

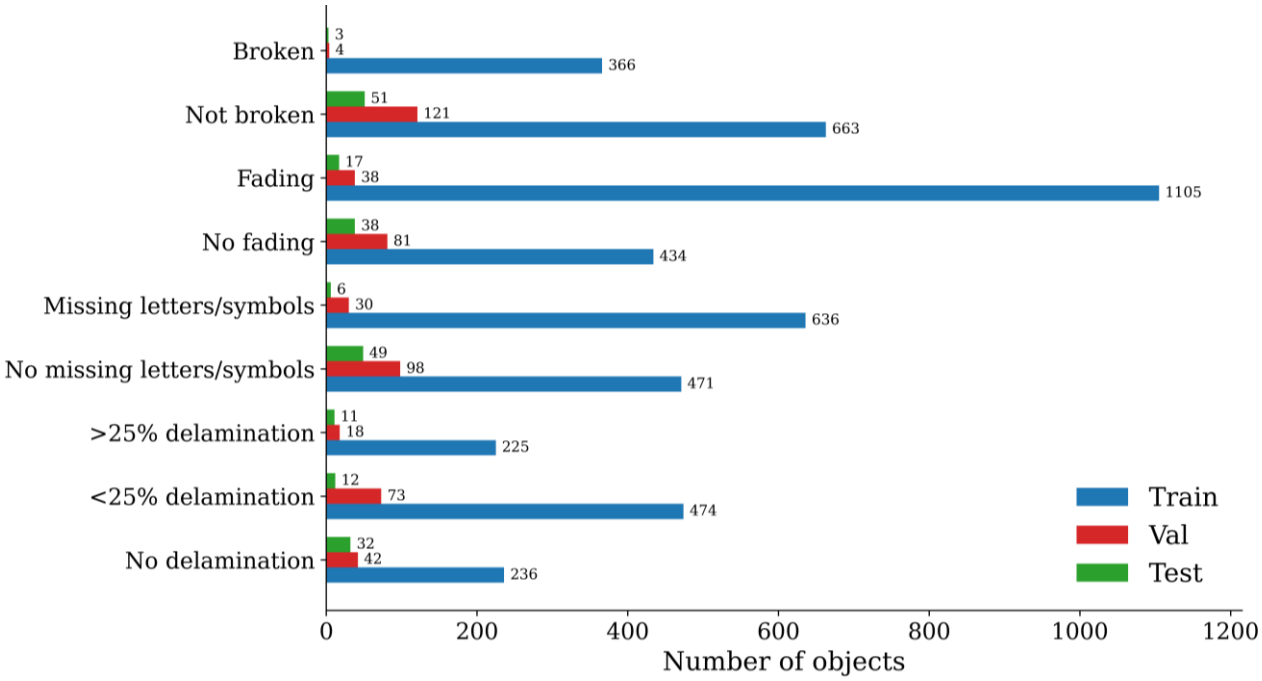

Figure 4.4: Distribution of training, validation, and test samples across traffic sign condition labels after targeted offline augmentations

Figure 4.4 shows the updated distribution of defect labels after applying targeted offline augmentation to the training set. Augmentation substantially increased the number of defect samples. For example, broken instances in the training set increased from 44 to 366 (8.3x) which shifted the broken/not broken ratio from approximately 1:12.5 to 1:1.8. Similarly, positive samples for fading increased from 191 to 1105 (5.8x), and missing letters/symbols increased from 178 to 636 (3.6x), while the corresponding "no defects" counts changed only modestly. For delamination, the rare " >25% delamination" class increased from 93 to 225 (2.4x). Despite these gains, delamination severity remained the most imbalanced task because "<25% delamination" remained the dominant class, and "no delamination" remained comparatively limited, thereby constraining severity classification. This limitation arises because augmentation is applied at the image level, and many roadway images contain both defective and non-defective signs. When the pipeline increased the number of samples for a rare defect, it often increased the corresponding "no defect" instances from the same images at the same time. As a result, the dataset still retains moderate majority–minority differences even after augmentation. To further mitigate this effect during training, the study applies a second strategy based on class weighting, which is described in detail in the next section.

4.2.1.2 Class Weight

Class weights provide a simple and effective way to address class imbalance by assigning higher weights to minority classes in the loss function, which helps penalize the errors on minority class than on majority class (He & Garcia, 2009). One common scheme uses inverse class frequency to calculate those weights (Abhinav, 2023). For a classification problem with n classes, and a total of N samples, the weights for class i can be computed as:

$$w_{i=} N / (n * N_i) \quad (1)$$

where N_i is the number of samples in class i. Using these weights, the weighted binary cross-entropy (BCE) loss can be calculated as follows:

$$\mathcal{L}(y, p) = -\sum_{i=0}^{n-1}(w_i\, y_i\, \log(p_i) + (1 - y_i)\, \log(1 - p_i)) \quad (2)$$

where $y_i \in [0,1]$ is the ground-truth label and $p \in (0,1)$ is the predicted probability of the defect class. When the positive (defect) class is rare and w_i is large, the loss penalizes misclassified defect samples more heavily, which encourages the model to learn better decision boundaries for the minority class instead of collapsing to the majority "no defect" prediction.

4.3 Methodology

There are different types of CV tasks as illustrated in Figure 4.5. Image classification assigns a single label to an object or scene in an image. Classification with localization assigns a label and also predicts a bounding box that specifies the object's position. Object detection extends this idea to images that contain multiple objects by predicting a label and a bounding box for each instance in the scene (Liu et al., 2020).

This study uses an object detection framework to locate traffic signs and assign defect labels to each sign. Figure 4.6 illustrates the overall workflow for the proposed approach. This framework uses a shared dataset (described in Section 3.2) to train five individual detection models instead of using one multi-task model to reduce negative transfer between tasks and to allow task-specific loss weighing, which enables clear attribution of performance to each defect

task. Each model is task-specific, where one model focuses on sign type, and four models focus on individual defects: (i) missing letters/symbols, (ii) delamination, (iii) fading, and (iv) broken signs. Each model uses the same preprocessed imagery described in Section 3.2, where the high-resolution roadway images are divided into three tiles and then resized to the model input size. This approach helps preserve details around the sign while keeping computational cost manageable.

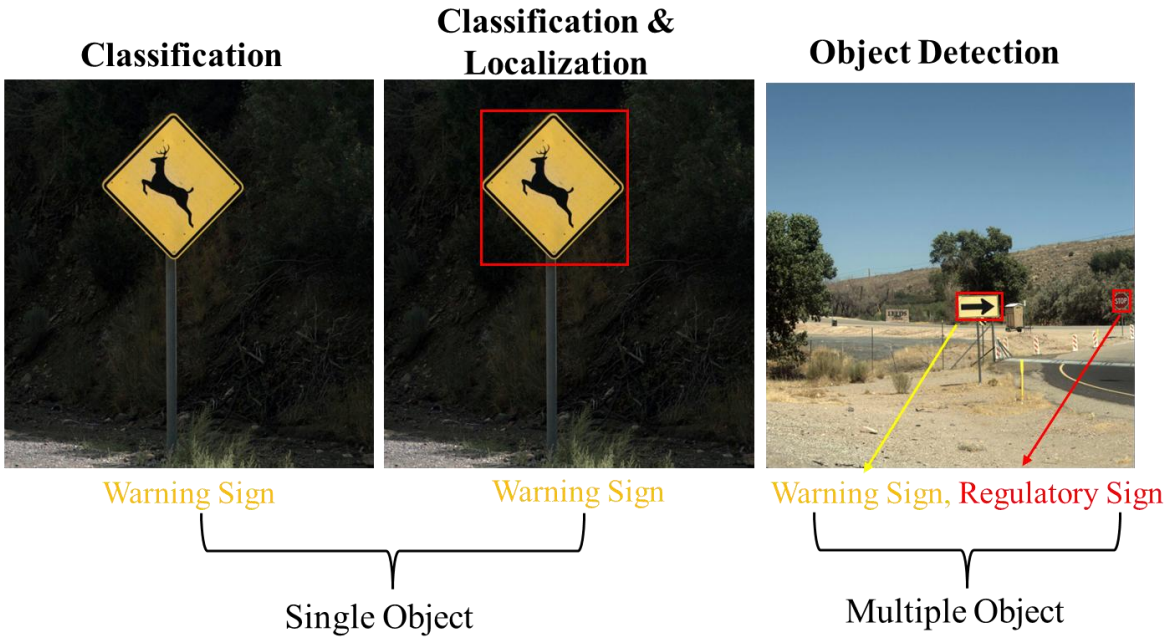

Figure 4.5: Examples of classification, classification with localization, and object detection for traffic signs.

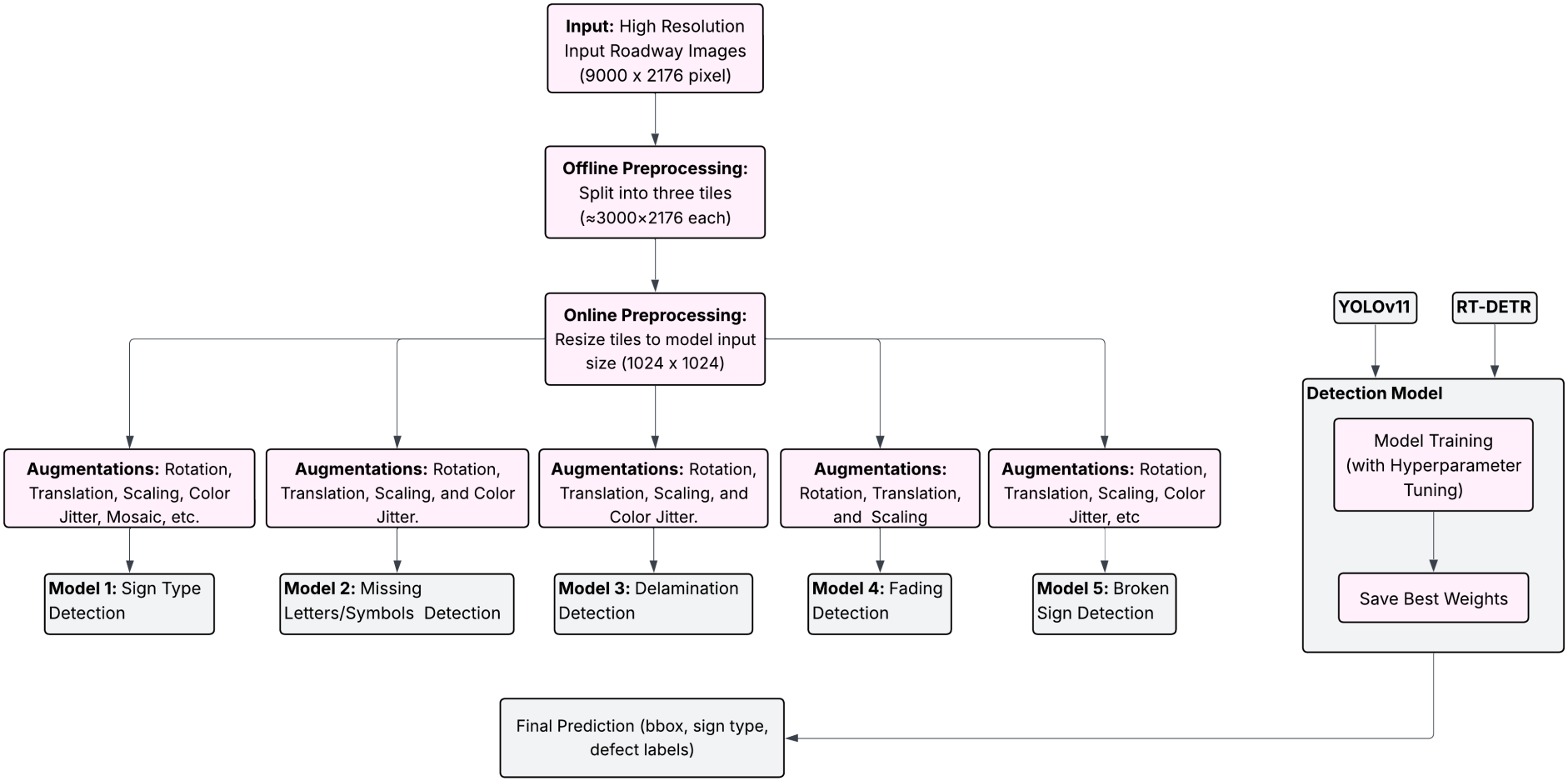

Figure 4.6: Overview of the proposed traffic sign detection and defect-classification workflow.

As shown in Figure 4.6, the framework used specific augmentation for each model. The sign-type model applied geometric and photometric transformations such as rotation, translation, scaling, and color jitter to improve robustness to viewpoint and illumination changes, whereas the defect models applied more selective transformations. For example, in fading, the pipeline avoided strong color or brightness changes that could mislead the model into learning these cues, since fading directly relates to a loss of color intensity. Delamination, missing letters/symbols, and broken signs used geometric augmentation and moderate color jitters to increase variability without erasing defect patterns.

This study used YOLO11 for traffic sign type detection and classification, and trained and compared two detector families, YOLO11 and RT-DETR, for traffic sign defect detection. Table 4.1 summarizes their roles, key architectural features, and suitability for this application. The following sections describe the YOLO and RT-DETR architectures and explain why these two models are appropriate choices for this study.

Table 4.1: YOLO11 and RT-DETR Overview

Aspect	YOLO11 (CNN-based)	RT-DETR (Transformer-based)
Detector type	One-stage dense CNN detector	End-to-end transformer detector
Design goal	Balance between accuracy and inference speed for real-time detection	Real-time DETR-style detection without NMS using set-based prediction
Feature handling	Backbone and neck extract and fuse multi-scale features from full images	Efficient hybrid encoder (AIFI + CCFM) keeps multi-scale reasoning with lower computation
Small / distant targets	Multi-scale feature maps help detect small and distant traffic signs in highway imagery	Query selection and multi-scale features focus on the most informative sign regions
Model scaling / deployment	Multiple model sizes (Nano–X) support both edge devices and server-side deployment	Adjustable number of decoder layers lets users trade accuracy for speed without retraining
Role in this study	Baseline CNN detector for traffic sign detection and defect labeling	Modern transformer detector used to test potential gains for defect detection

4.3.1 YOLO

The original YOLO architecture considers object detection as a single regression problem by directly mapping image pixels to bounding boxes and class probabilities. A single neural network processes the entire image in a single forward pass and directly predicts multiple bounding boxes and class scores, which enables real-time performance with reasonable accuracy (Redmon et al., 2016b). YOLO divides the input image into an S×S grid, where S denotes the number of grid cells along each spatial dimension (i.e., the image is partitioned into S cells in width and S cells in height). Each grid cell predicts a set of bounding boxes, an objectness score for each box, and class probabilities. The detector then multiplies the objectness score and class probabilities to obtain confidence scores, then applies non-maximum suppression (NMS) to keep the highest-confidence boxes and remove duplicates. This prediction workflow by YOLO is illustrated in Figure 4.7.

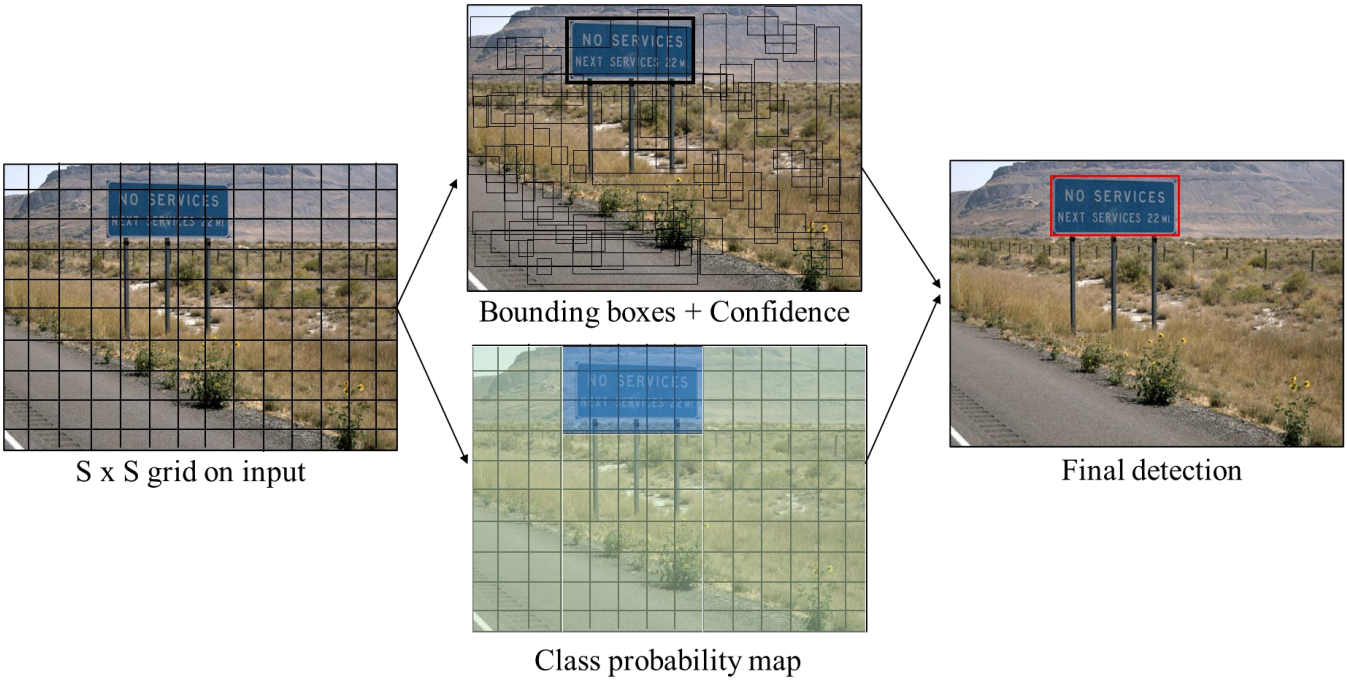

Figure 4.7: Simplified prediction workflow in YOLO

This study uses YOLO11 (Ultralytics, 2024), the latest YOLO variant, as the primary CNN-based model (Lecun et al., 1998). YOLO11 maintains the standard three-part structure of backbone, neck, and head. The backbone extracts visual features; the neck combines multi-scale features to improve detection on small and distant traffic signs; and the head predicts bounding boxes and class labels at multiple scales. The YOLO11 architecture incorporates C3k2 bottleneck blocks, a Spatial Pyramid Pooling-Fast (SPPF) module for multi-scale context aggregation, and a Cross Stage Partial with Spatial Attention (C2PSA) attention module to support feature extraction across scales and improve spatial relationship modeling in the learned feature maps. Lighter variants (e.g., YOLO11n and YOLO11s) are suitable for real-time deployment due to lower inference latency and fewer parameters, whereas larger variants (e.g., YOLO11l and YOLO11x) typically achieve higher detection accuracy at increased computational cost, making them more appropriate for server-side inference (Ultralytics, 2024). YOLO11 achieved a higher mean average precision (mAP) than earlier YOLO versions on the COCO dataset, which makes it a suitable choice for large-scale traffic sign detection. Unlike anchor-based YOLO variants, YOLO11 uses an anchor-free detection head that predicts the center of the object and the dimensions of the bounding box directly from feature maps instead of relying on pre-defined anchor boxes. The architecture of YOLO11 is shown in Figure 4.8.

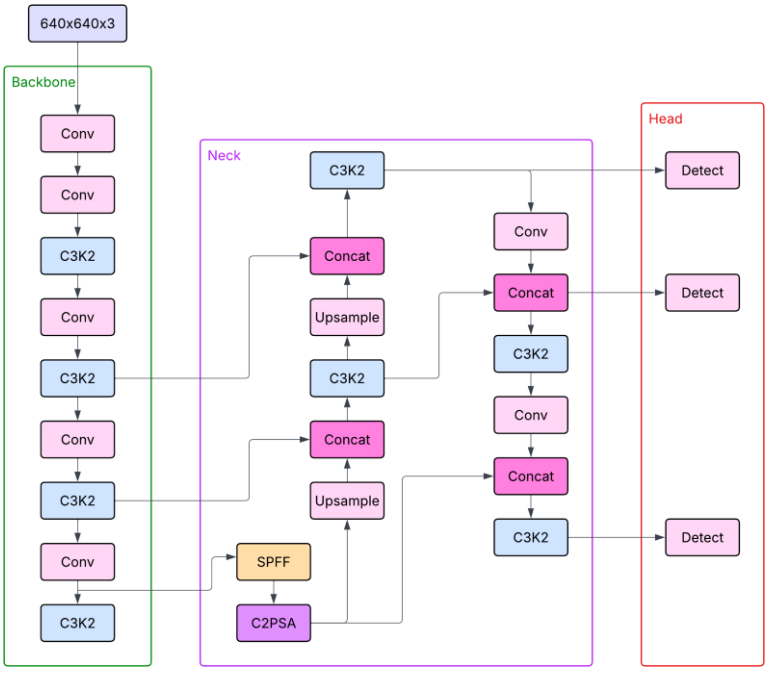

Figure 4.8: YOLO11 network architecture

4.3.2 RT-DETR

RT-DETR is a transformer-based object detector that aims for a real-time, end-to-end detection without NMS. Zhao et al. (2024) introduced RT-DETR, which uses a CNN backbone to extract multi-scale feature maps at intermediate stages. The encoder includes an Attention-based Intra-Scale Feature Interaction (AIFI) module, which applies self-attention within a feature scale and is implemented on the high-level S5 feature map. It also includes a CNN-based Cross-Scale Feature Fusion (CCFF) module, which aggregates information across backbone scales (e.g., S3-S5) using convolutional fusion blocks to produce the enhanced multi-scale feature. Then, an uncertainty-minimal query selection module assigns a score to the encoder features, which helps retain only the most reliable ones as object queries. Next, through an iterative process, a lightweight transformer decoder refines these queries and through auxiliary prediction heads, it outputs object categories and bounding boxes in a single forward pass. Furthermore, RT-DETR supports flexible speed tuning without retraining and eliminates the inconvenience caused by NMS thresholds. On the COCO benchmark, RT-DETR achieves a competitive speed–accuracy trade-off and, under the evaluation settings, it outperforms previously reported YOLO-based baselines in both average precision and inference speed (Zhao et al., 2024c). Because benchmark rankings do not necessarily transfer to traffic-sign defect data, this study evaluates YOLO11 and RT-DETR empirically on the proposed dataset (Chapter 5). The architecture of RT-DETR has been presented in Figure 4.9.

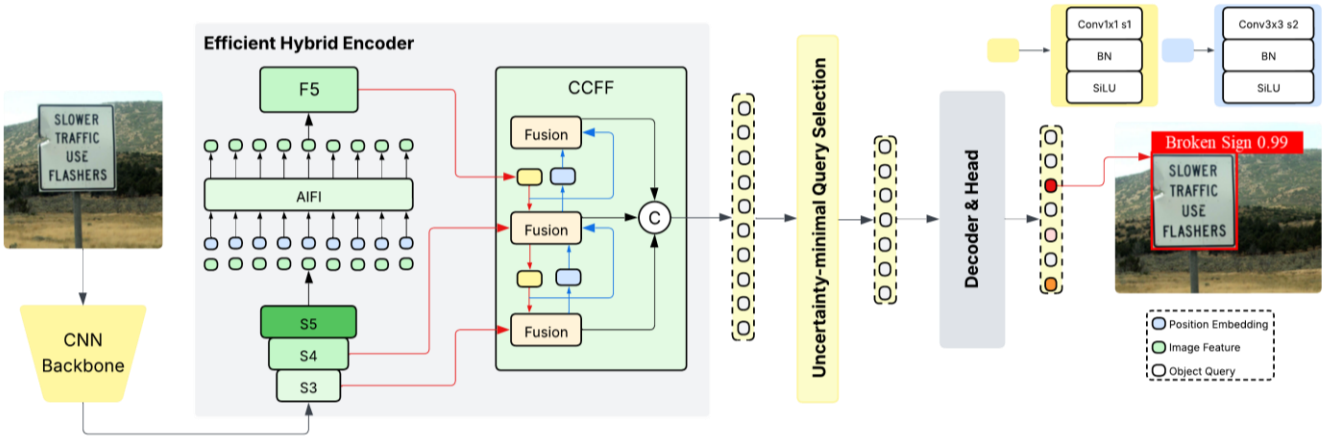

Figure 4.9: RT-DETR network architecture

4.3.3 Hyperparameter Tuning

Hyperparameters govern how the model learns from data and are specified prior to training; unlike model parameters, they are not updated during optimization. Typical hyperparameters in neural networks include the learning rate, weight decay, class weights, and loss-component weights, all of which can significantly affect model performance. The two goals of hyperparameter tuning are (1) to improve generalization and training stability on an imbalanced dataset, and (2) to reduce the bias of the majority (no-defect) classes.

The tuning process followed a two-stage approach. In the first stage, a coarse grid search was performed over a broad range of values for the most influential hyperparameters to identify promising regions of the search space. In the second stage, a refined search was conducted within narrow intervals centered on the best-performing candidates identified in the first-stage validation results. Key tuned hyperparameters included class weights to emphasize rare defect classes, the initial (lr0) and final learning rates (lrf) to control the pace of optimization, the classification-loss weight (cls) to prioritize defect-class accuracy, and weight decay to regularize the model and limit overfitting.

4.4 Evaluation Metrics

The model's performance was evaluated using multiple metrics, including precision, recall, and F1-score. F1-score summarizes the precision-recall trade-off and is particularly well-suited to class imbalance. Because precision, recall, and F1-score are computed from the counts of correct and incorrect predictions, the evaluation is defined in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). TP denotes cases where the model correctly identifies a defect on a traffic sign, whereas TN corresponds to signs correctly classified as having no defect (i.e., good condition). FP occur when the model incorrectly predicts a defect on a non-defective sign, and FN occur when an existing defect is missed and the sign is incorrectly classified as non-defective. This can be summarized in a confusion matrix, which tabulates counts of predictions by ground-truth and predicted classes to show where misclassifications occur. These outcomes are illustrated in Figure 4.10 through the confusion matrix layout and representative prediction example.

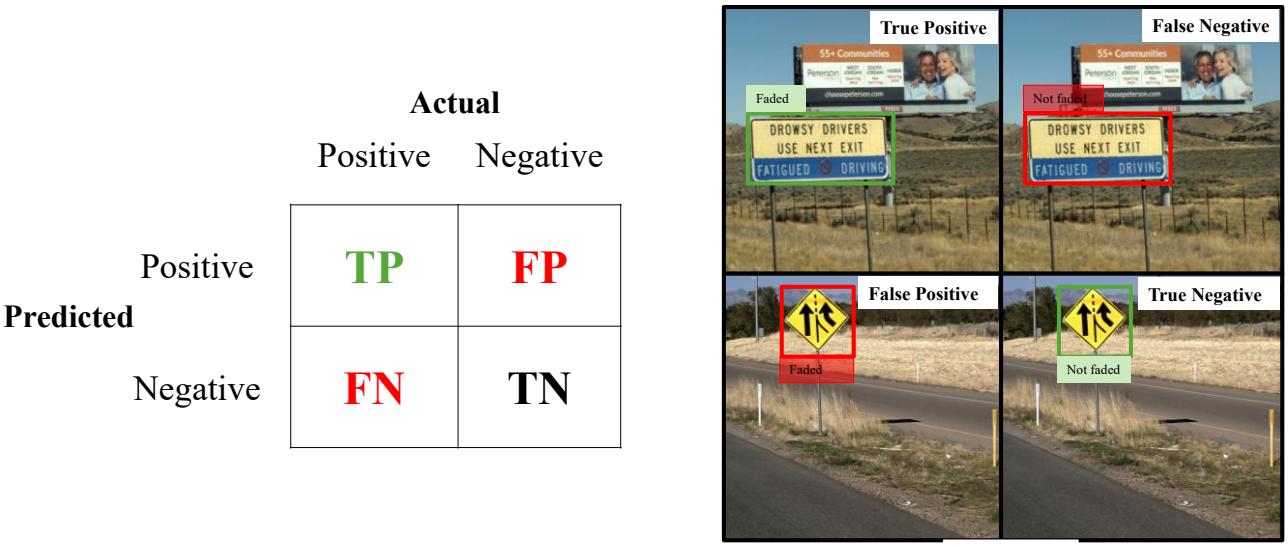

Figure 4.10: (a) Confusion-matrix layout and (b) example outcomes (TP, FP, TN, FN) for the fading defect

Precision measures the proportion of predicted positive instances that are correctly classified. It reflects the accuracy of the model's positive predictions and is defined as:

$$Precision = TP/(TP + FP) \quad (3)$$

Recall quantifies the model's ability to correctly identify all actual positive instances. It measures completeness in detection and is expressed as:

$$Recall = TP/(TP + FN) \quad (4)$$

The F1-score is the harmonic mean of precision and recall, balancing both measures into a single performance metric. The F1-score is calculated as:

$$F1 = 2 * (Precision * Recall) / (Precision + Recall)) \quad (5)$$

An F1-score closer to 1 indicates strong overall detection performance, reflecting both high precision (few false positives) and high recall (few false negatives).

4.5 Summary

This chapter described the end-to-end methodology for training and evaluating CV models for traffic-sign condition assessment under highly imbalanced defect labels. It first characterized the imbalance across defect classes and explained why rare and subtle defects create instability in training and unreliable estimates in evaluation when the dataset contains few positive samples. The chapter then described two mitigation strategies used in this study. The first strategy applied targeted offline augmentation to increase the representation of under-sampled defect classes in the training set while keeping the validation and test sets unchanged. The second strategy applied class weighting during training to reduce the tendency of models to favor the dominant "no defect" labels and to emphasize learning from minority defect examples.

The chapter then introduced the comparative modeling framework built around YOLO11 and RT-DETR. It outlined the training pipeline and summarized the main hyperparameter settings and tuning choices used to support stable optimization and fair model comparison. Rather than training a single multi-task network, the framework trained five independent models for the condition tasks, which enabled task-specific sampling and loss weighting and reduced interference across defects with different label structure and prevalence. Finally, the chapter defined the evaluation protocol and reporting format. It described precision, recall, and F1-score as the primary metrics and summarized model behavior through normalized confusion matrices to highlight both overall performance and class-specific strengths and limitations.

5.0 Results

5.1 Overview

This section compares the detection and classification results for both YOLO11 and RT-DETR for sign type and individual defect tasks using precision, recall, and F1-score. The comparison examines these models in terms of evaluation metrics, failure modes, and inferences to identify which model better supports traffic sign condition assessment.

5.2 Sign Type

Sign-type detection and classification were performed using only YOLO11. Figures 5.1 summarize the model's performance on the validation and test sets, respectively, using normalized confusion matrices and class-wise precision, recall, and F1-score as bar plots. On the test set, the model achieved an F1-score of 84% for both regulatory and guide signs, whereas the warning signs had the lowest F1-score of 76%. The lower F1 score for the warning signs is primarily driven by lower recall (i.e., 67%) while precision remains high (i.e., 89%) and is comparable to that of the regulatory and guide signs. Figure 5.1 shows that the model frequently missed the warning sign and predicted as background, indicating that the dominant error mode for this class is missed detection rather than confusion with other sign types.

The test set was intentionally selected to include challenging roadway scenes, such as frames containing multiple nearby or overlapping signs, to assess the model's performance under conditions expected in real deployments. In these crowded cases, detectors often struggle because substantial overlapped instances share similar features, and NMS can suppress correct boxes along with duplicates (Chu et al., 2020). Figure 5.3 presents representative predictions for the sign type task, including correctly detected and classified signs, as well as cases with missed or overlapping boxes.

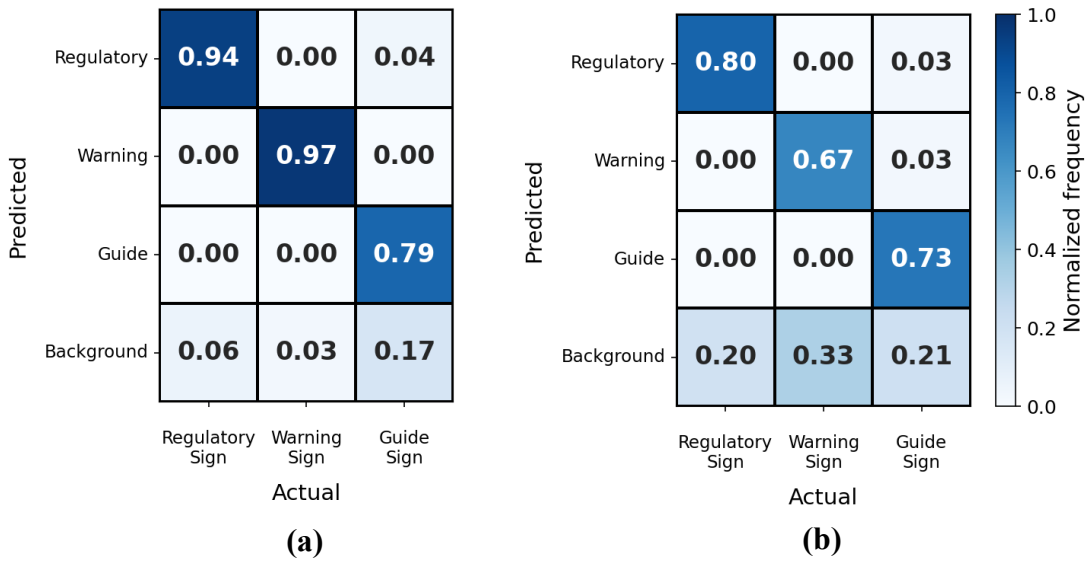

Figure 5.1: Normalized confusion matrix (a) on validation set (b) on test set for sign type

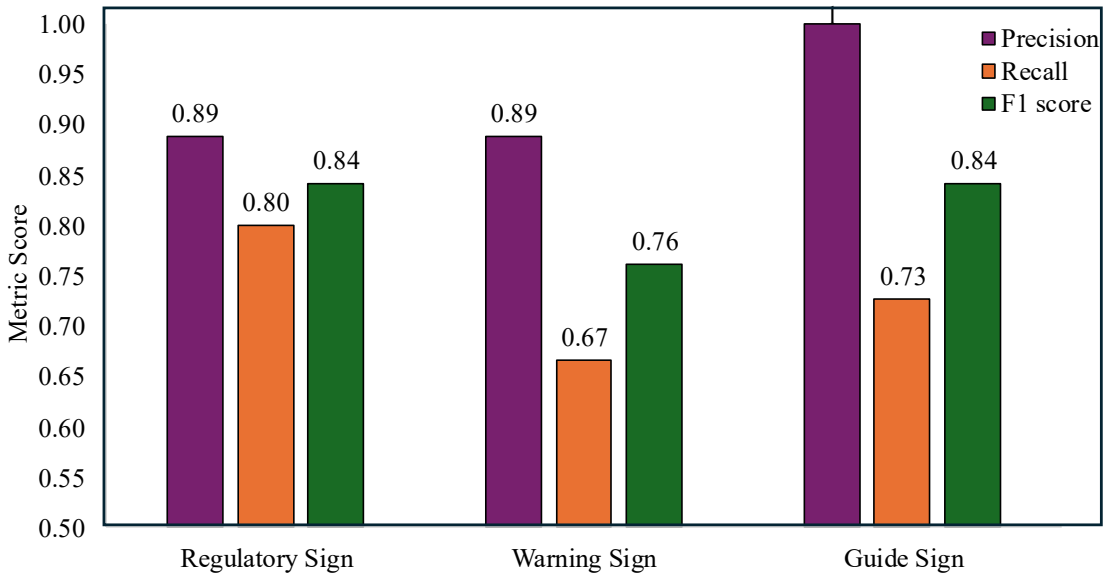

Figure 5.2: Precision, Recall, and F1-score for sign-type classification on the test set

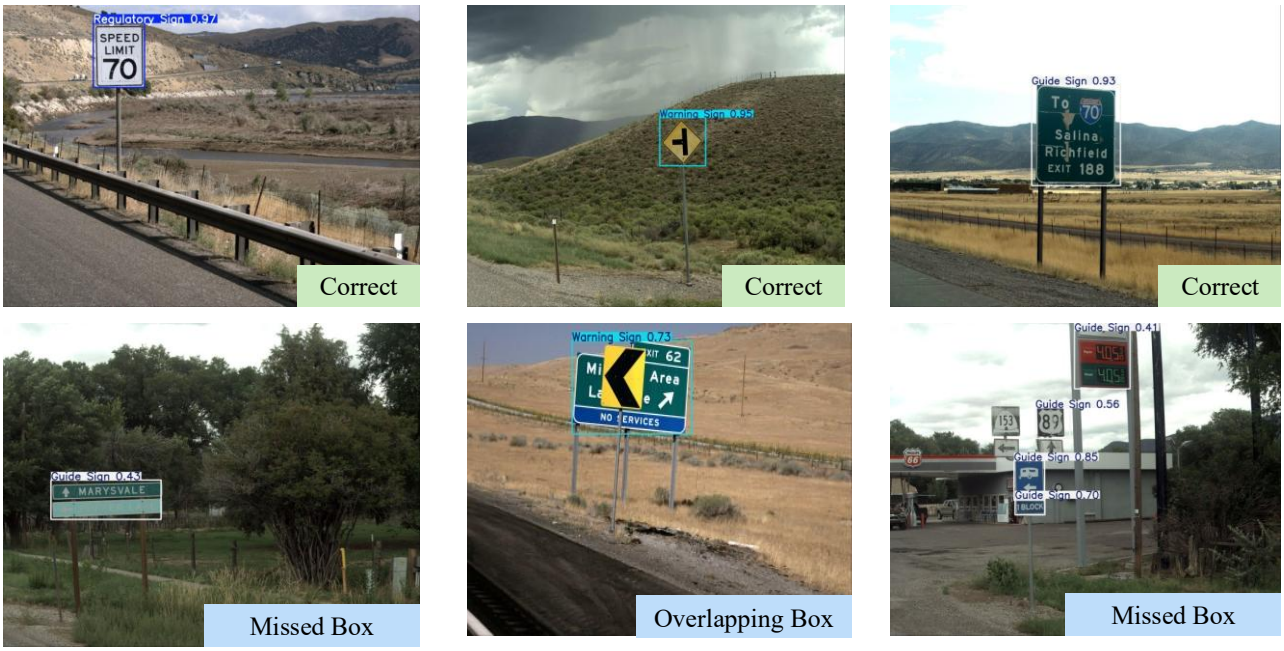

Figure 5.3: Representative YOLO sign-type inferences

5.3 Fading

A comparative evaluation of YOLO11 and RT-DETR was conducted for fading defect detection. On the validation set, both models achieved identical F1-scores of 75% (Figure 5.4); however, the test-set results show a clear generalization gap for RT-DETR relative to YOLO11. On the test set, YOLO11 outperformed RT-DETR across all three metrics, achieving a precision of 64%, a recall of 70%, and an F1-score of 67%, compared with 50%/50%/50% for RT-DETR (Figure 5.5). The normalized confusion matrices further indicate that RT-DETR correctly identifies 50% of faded signs and predicts the remaining signs as “no fading,” thereby reducing the true-positive rate for the fading class (Figure 5.4). In contrast, YOLO11 shows a higher proportion of correctly identified faded instances, indicating stronger sensitivity to fading cues under the test conditions. Representative qualitative examples in Figure 5.6 illustrate typical detections and failure modes for both models on the fading task.

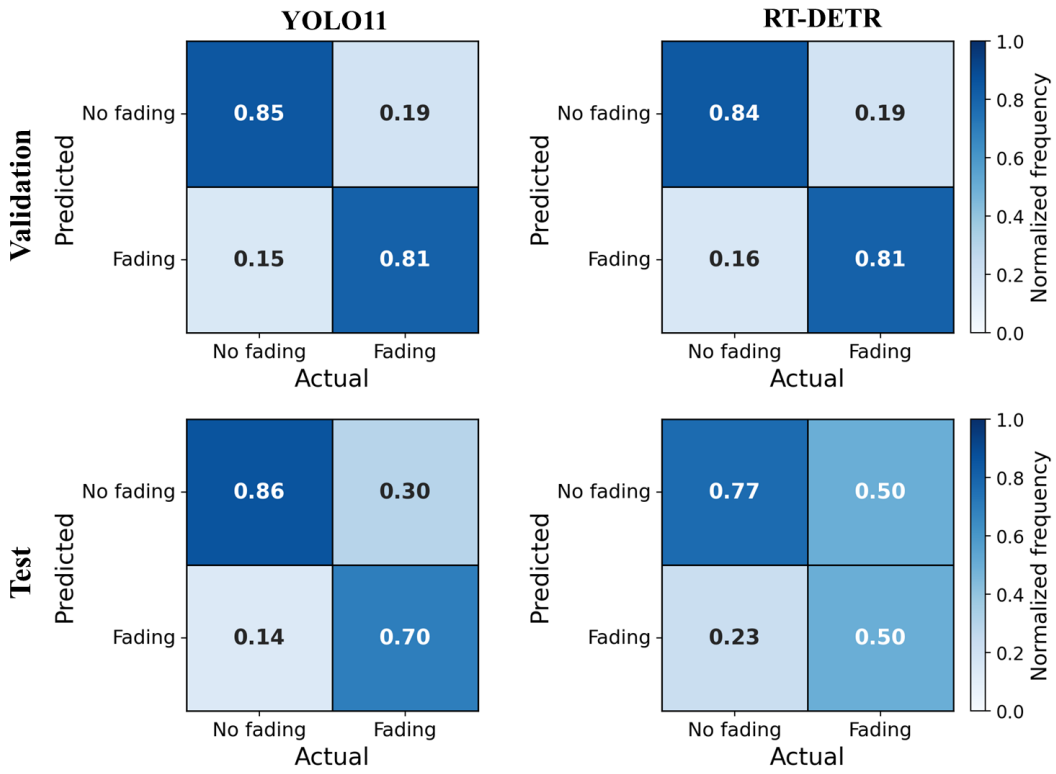

Figure 5.4: Normalized confusion matrix on validation and test set for fading with YOLO RT-DETR

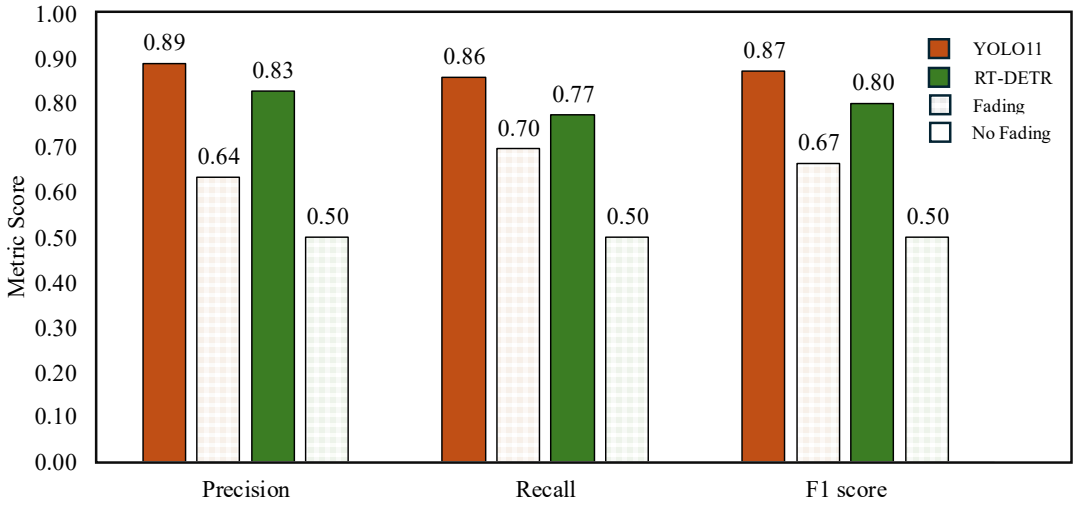

Figure 5.5: Precision, Recall, and F1-score for fading vs. no fading defect on the test set (YOLO11 vs RT-DETR)

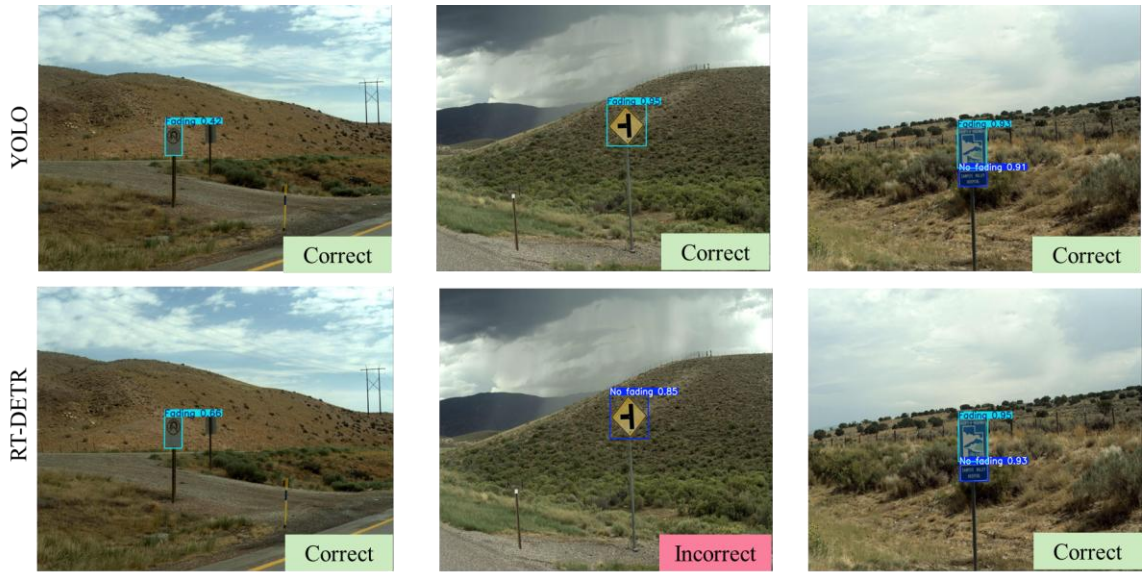

Figure 5.6: Representative inferences from YOLO11 and RT-DETR for fading

Several factors may reduce the reliability of fading detection in this dataset. One source of error is the labeling strategy, which has introduced ambiguity, as cases involving localized loss of letters or symbols due to peeling were annotated as “fading” (Figure 5.16) which differs from uniform color loss across the sign face. Such localized fading effect occupies a small fraction of the sign area as compared to the undamaged portion of the object’s surface, which led to accuracy issues (Spencer et al., 2019). Consequently, these rare, fine-scale defects challenge both models and are likely to reduce performance on the fading class.

5.4 Delamination

Delamination was categorized into three classes based on the estimated affected area: no delamination, delamination affecting less than 25% of the sign face (hereafter, <25% delamination), and delamination affecting more than 25% of the sign face (hereafter, >25% delamination). Both YOLO11 and RT-DETR were trained and evaluated for this defect. The delamination results are summarized using normalized confusion matrices (Figure 5.7), class-wise precision, recall, and F1-score as bar plots (Figure 5.8), and representative inferences illustrating typical detections and failures (Figure 5.10).

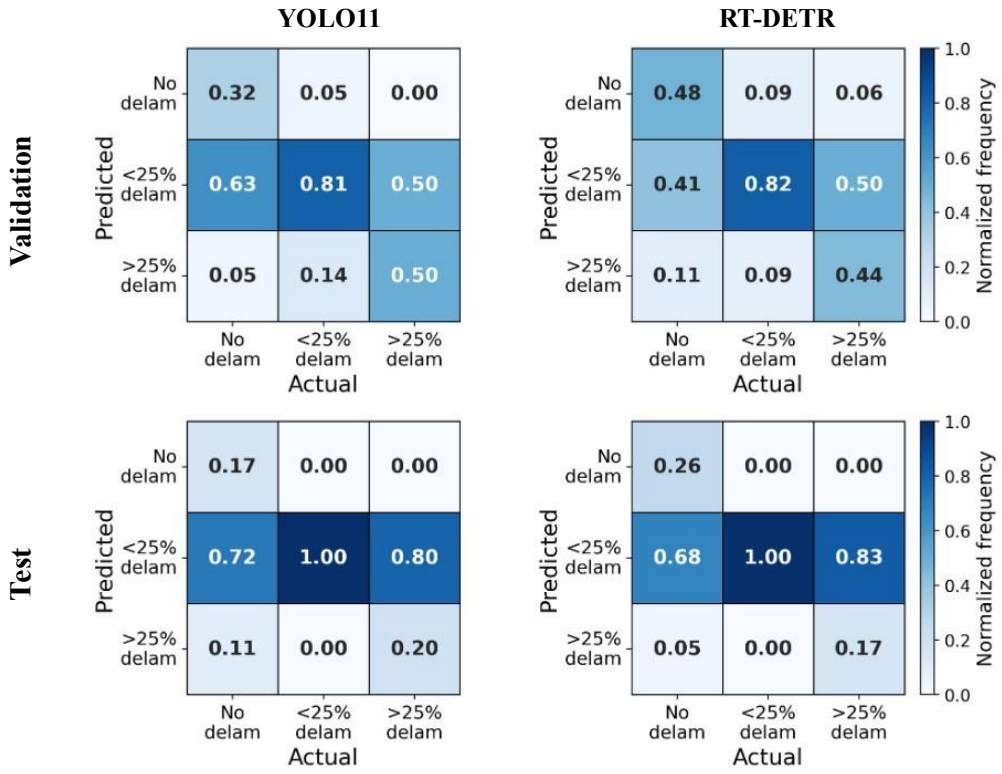

Figure 5.7: Normalized confusion matrix on validation and test set for delamination with YOLO and RT-DETR

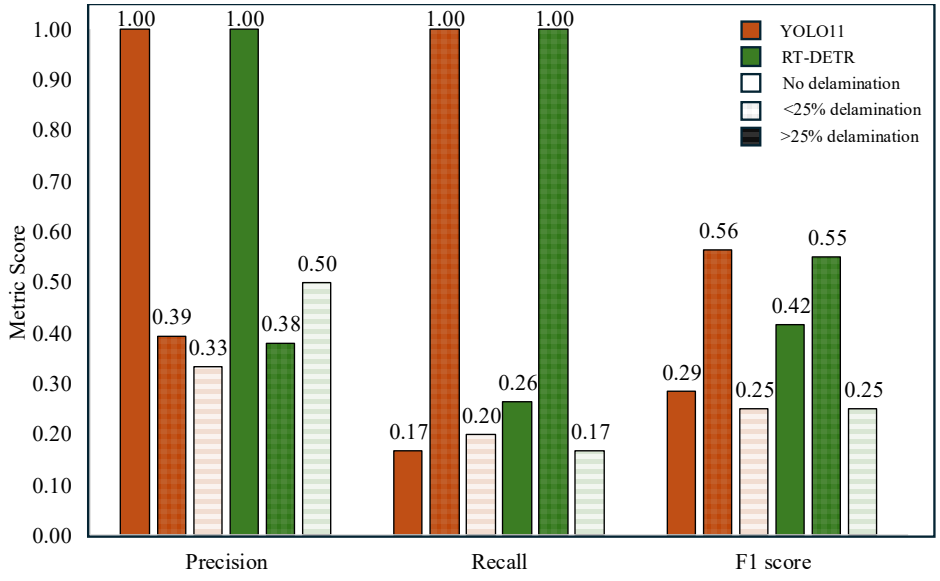

Figure 5.8: Precision, Recall, and F1-score for three-class delamination defect on the test set (YOLO11 vs RT-DETR)

On the test set, both models exhibit a strong bias toward the <25% delamination category. The recall for this class is 100% for both the detectors, indicating that all true <25% instances are identified; however, precision falls around 39% for YOLO11 and 38% for RT-DETR. This observation suggests that many signs from the other two categories are incorrectly mapped to the <25% class. Consistent with this, the confusion matrices indicate substantial cross-confusion between no delamination and >25% delamination, with a significant fraction of both classes predicted as <25% delamination.

These experiments indicate poor severity discrimination for delamination. Both detectors converge to the dominant class (<25%) and cannot effectively distinguish between the three categories. The severity of this defect shows a fine texture that may appear nuanced and distant, and it is visually similar on either side of the 25% threshold. The imbalanced label distribution and weak visual contrast across this boundary likely led the models to generalize to a coarse "some delamination" feature rather than to reliably distinguish severity.

As a diagnostic experiment to assess whether data scarcity prevented reliable severity separation, the two positive subclasses (<25% and >25%) were merged into a single delamination class, while retaining the no-delamination class. This relabeling converts the original three-class severity problem into a binary detection task (delamination vs. no delamination). After merging, the delamination class contains more instances than no delamination, which can bias predictions toward delamination. Nevertheless, performance improves relative to the three-class setting because the models are required to detect the presence of delamination rather than distinguish between visually similar severity levels near the 25% threshold. On the test set, YOLO11 achieved a precision of 53%, a recall of 94%, and F1-score of 68%, while the RT-DETR achieved 52% / 100% / 68% (Figure 5.9). These results indicate that, given the current data volume, both models support binary delamination detection more reliably and consistently than severity-level classification.

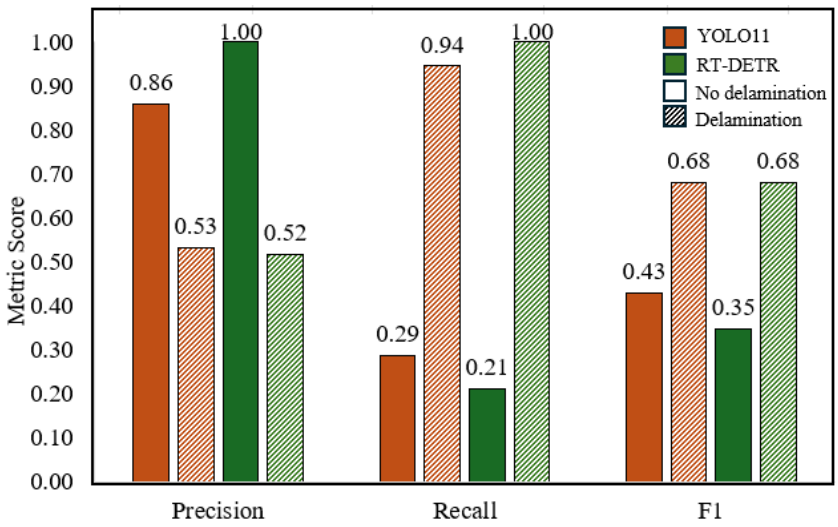

Figure 5.9: Precision, Recall, and F1-score for two-class delamination defect on the test set (YOLO11 vs. RT-DETR)

Figure 5.10: Representative inferences from YOLO11 and RT-DETR for delamination

5.5 Missing letters/symbols

The missing letters/symbols condition was formulated as a binary classification problem with two labels: no missing letters/symbols and missing letters/symbols. The normalized confusion matrices for the validation and test set (Figure 5.11) indicate that models produce a high number of missed detections for missing letters/symbols. On the test set, YOLO11 achieved

a precision of 25%, a recall of 17%, and an F1-score of 20% for the missing class, whereas RT-DETR achieved 18% / 33% / 24% (Figure 5.12). Relative to YOLO11, RT-DETR recovers a fraction of missing cases (higher recall) at the expense of false positives (lower precision).

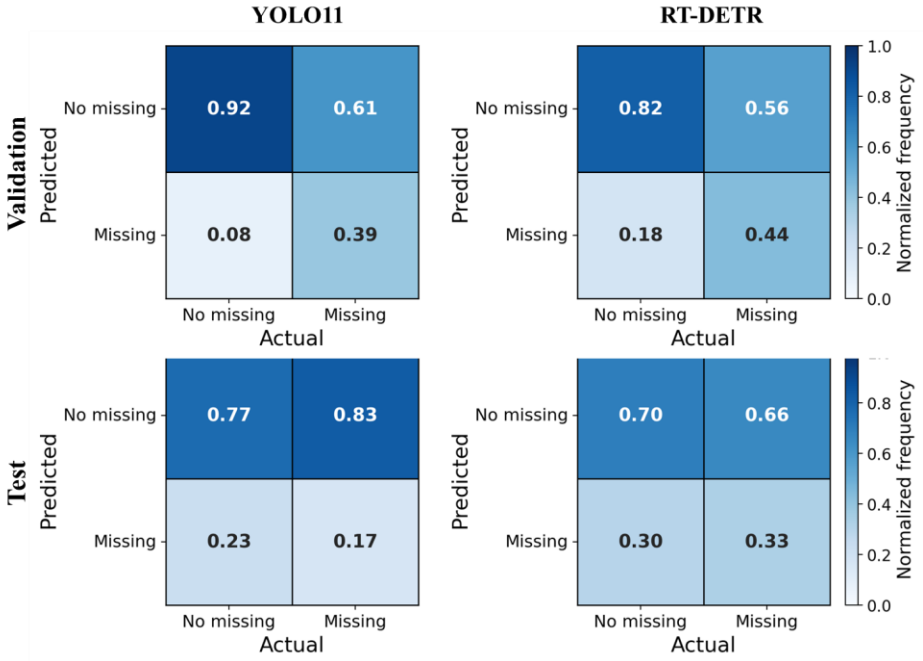

Figure 5.11: Normalized confusion matrix on validation and test set for missing letters/symbols with YOLO and RT-DETR

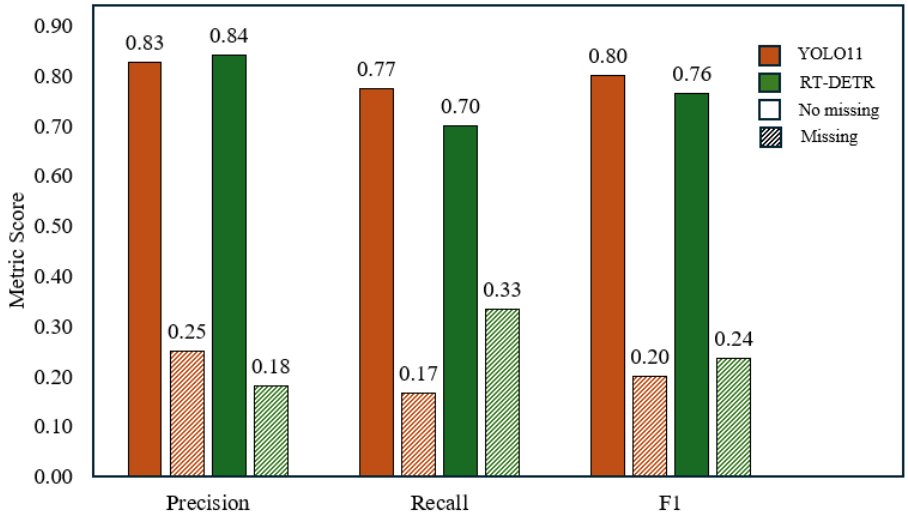

Figure 5.12: Precision, Recall, and F1-score for missing letters/symbols on the test set (YOLO11 vs. RT-DETR)

Both models exhibit limited performance on the missing letters/symbols defect, which is characterized by weak and fine-grained visual cues. In many instances, these cues often overlap with delamination, which produces a visually similar peeling texture on the legend or sign face. The labeling scheme treats missing letters/symbols as a separate class, but several "no-missing" samples still contain background delamination; therefore, these instances function as noisy negative examples during training. This overlap increases false positives and missed detections, contributing to the reduced precision-recall performance observed for YOLO11 and RT-DETR for this defect.

Representative predictions for the missing letters/symbols defect are shown in Figure 5.13. The example shows that the models correctly identify missing letters when the defect is visually clear and isolated, but they struggle when the missing content overlaps with background delamination, leading to incorrect defect classifications.

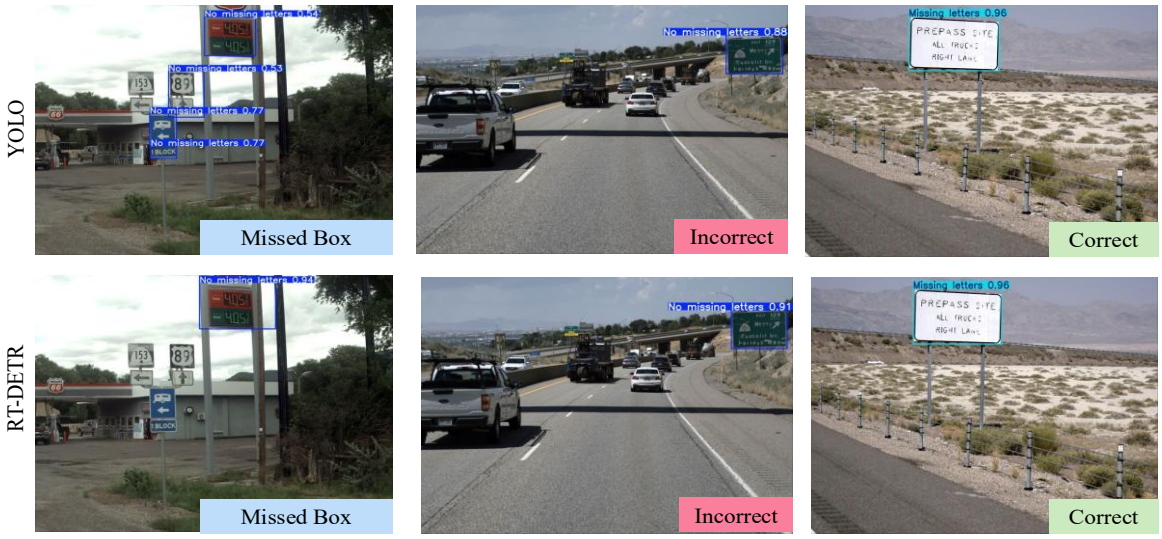

Figure 5.13: Representative inferences from YOLO11 and RT-DETR for missing letters/symbols

5.6 Broken

Broken sign detection was formulated as a binary problem with two classes: not broken and broken. YOLO11 and RT-DETR were trained using the same data and evaluation protocol described earlier in this chapter. Performance is reported using normalized confusion matrices for the validation and test sets (Figure 5.14). The validation results show limited sensitivity to broken signs, but this behavior does not transfer to the test set, as both detectors classify all of

the “broken” instances as “not broken,” indicating that the dominant performance issue of the CV models is the missed detection of the broken class (i.e., zero recall for broken). This observation underscores that broken sign detection remains unreliable under the current dataset.

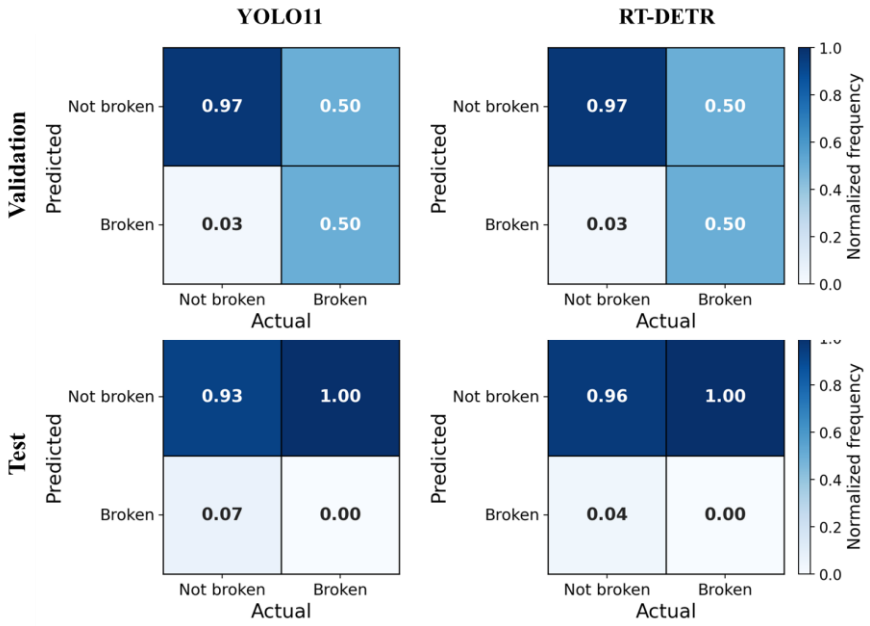

Figure 5.14: Normalized confusion matrix on validation and test set for broken signs with YOLO and RT-DETR

Both YOLO11 and RT-DETR exhibit limited reliability in detecting the broken sign defect. Although the training set contains a moderate number of broken samples (e.g., 366 broken versus 660 not broken), the validation and test sets include only a few broken signs (4 and 3, respectively). With these tiny sample sizes, performance estimates for the broken class are highly sensitive to individual samples. Thus, the models tend to default to the majority “not broken” label, resulting in missed detections of broken signs. In addition, many broken signs exhibit only a small missing corner or edge rather than a large missing region, which often erases the cue indicating damage after resizing the input image for training. Thus, the results suggest that the primary bottleneck is the combination of limited positive examples and weak, fine-scale visual evidence at the current image scale, rather than a model-specific architectural limitation. Representative predictions for this defect are shown in Figure 5.15, illustrating the dominant failure mode of missed broken cases.

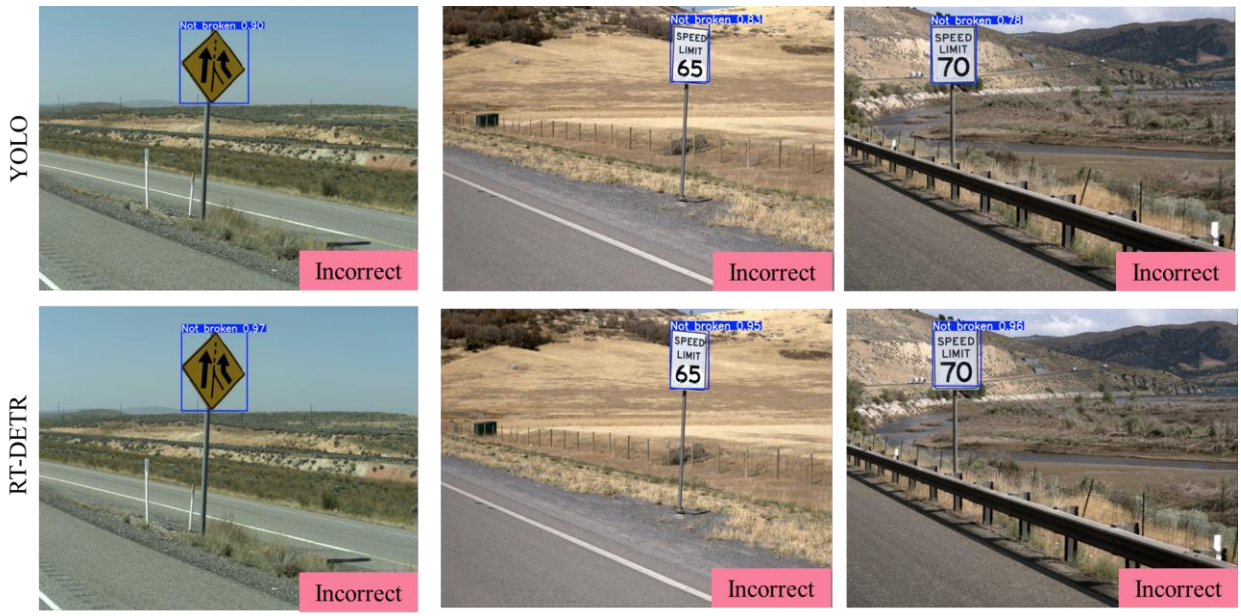

Figure 5.15: Representative inferences from YOLO11 and RT-DETR for broken signs

5.7 Challenges

There were several challenges that limited the models' performance on the defect classification task for traffic signs. First, most of the signs exhibited multiple types of defects simultaneously. For example, delamination and missing letters often co-occur, and they both have a similar texture. This overlap reduced separability between classes: both YOLO11 and RT-DETR tended to learn a generic "peeling" appearance, yielding comparatively stronger performance for delamination but weaker performance for missing letters. The missing letters task was particularly affected when both delamination and missing letters co-occurred. Delamination added noisy background patterns, but the missing-letters label targeted only the character loss. Consequently, when peeling textures were present in the background, the models frequently misattributed the visual cues and misclassified the defect because the peeling texture dominated the extracted features (Figures 17 and 18).

Second, the definition and naming convention for the defect were susceptible to inconsistencies. The study did not depend on precise pixel measures to distinguish between "<25%" and ">25% delamination." This provided vague boundaries for the models to differentiate between the classes. For fading, the study also included images with peeling or

partially peeled letters, which produced a fading-like appearance (Figure 5.16), broadening the visual variability within the fading class and increasing intra-class heterogeneity, thereby making the fading decision boundary more challenging to learn.

Third, data limitations introduced additional challenges. The frequency of the "broken sign" class was very low, and noticeable damage usually involved subtle nuances around the edges. Owing to the need to improve the count of "broken sign," the sign with <10% damage was also marked as "broken sign," thereby weakening the significance of the class and pushing their F1 scores toward zero. Deep learning models, such as YOLO11 and RT-DETR, require ample, diverse, and high-quality datasets, but existing datasets did not meet these requirements, especially for rare defects. Additional failure modes were observed in dense roadway scenes containing multiple nearby signs. In such cases, both detectors sometimes missed instances or produced merged/overlapping boxes that represent multiple signs as a single detection. Collectively, these limitations created a challenging evaluation setting that constrained defect-level reliability for both YOLO11 and RT-DETR under the current data and labeling conditions.

Figure 5.16: Fading effect due to peeling of letters

Figure 5.17: Co-existing delamination and missing letters defect

Figure 5.18: Delamination is present, but no missing letters

Figure 5.19: Minor vs. apparent visible breakage

5.8 Summary

This chapter presented the labeled dataset and the modeling workflow used to detect and classify traffic sign conditions. It described the distribution of sign types and condition classes and highlighted the severe underrepresentation of several defect conditions, which likely limited the model's ability to learn defect-specific cues and contributed to weaker performance on the rare class. The chapter also explained the strategies used to address class imbalance, including offline data augmentation and class weights, as well as key hyperparameter tuning for both YOLO11 and RT-DETR. In addition, the chapter outlined the YOLO11 and RT-DETR architectures and documented their training and evaluation procedures. Performance was reported using precision, recall, F1-score, and confusion matrices for both detection and condition classification. In summary, YOLO11 performed better, especially for coarse defect screening (binary fading and binary delamination), whereas both models struggled with severity-level delamination and missing letters/symbols. Broken sign results were not reliable because the

test split contained very few positive broken samples. Finally, the chapter summarized some of the significant challenges observed in this study, which included defect overlap between delamination and missing letters, subjective severity labeling for delamination (e.g., <25% vs. >25%), overlapping signs in dense/crowded scenarios, and the practical limitations of training high-capacity models with very few defect instances and visually sparse damage patterns.

6.0 CONCLUSIONS

6.1 Summary

This study developed a CV training and evaluation framework to automate traffic sign detection and condition assessment using high-resolution roadway imagery collected along Utah highways. It established a labeled dataset according to the MUTCD guidelines that included three different sign types: (i) regulatory, (ii) warning, and (iii) guide signs, and four defect conditions: (i) fading, (ii) delamination, (iii) missing letters/symbols, and (iv) broken signs. The pipeline preprocessed the images, trained and tested YOLO11 and RT-DETR using identical training/validation/test splits, and evaluated performance using per-class precision, recall, F1-score, normalized confusion matrices, and qualitative inference examples.

6.2 Findings

YOLO11 achieved higher test performance, with an F1-score of 84% for regulatory and guide signs, and 76% for warning signs. The lower F1-score for warning was driven by a recall of 67%, despite a precision of 89%, comparable to that of regulatory and guide signs, indicating that warning signs were missed more often than they were falsely detected. For fading, YOLO11 performed better on the test set, achieving 64% precision and 70% recall, yielding an F1 score of 67%, whereas RT-DETR achieved 50% precision, 50% recall, and 50% F1 score. This indicates that YOLO11 identified faded signs more reliably than RT-DETR, with fewer false positives and fewer missed faded signs on the test set. Three-class delamination severity (no delamination, <25%, >25%) reflected residual imbalance effects, where both models achieved 100% recall for the dominant "<25% delamination" class, but precision was only 39% for YOLO11 and 38% for RT-DETR, indicating over-prediction of that class; for ">25% delamination," recall dropped to 20% for YOLO11 and 17% for RT-DETR, indicating frequent misses. Reformulating delamination as a binary task improved performance for both models, with a delamination-class F1-score of 68%. YOLO11 achieved 94% recall and 53% precision, whereas RT-DETR achieved 100% recall and 52% precision, suggesting that most delamination instances were captured, but false positives remained. Missing letters/symbols remained challenging for both

models, with 25% precision and 17% recall, yielding an F1 score of 20% for YOLO11, whereas RT-DETR achieved 18% precision and 33% recall, yielding a slightly higher F1 score of 24% compared with YOLO11. The performance is consistent with texture overlap between missing letters/symbols and delamination, which contributed to confusion between these defect classes. Precision and Recall for broken-sign were 0%, indicating unstable results due to the very few positive test samples, and therefore should not be interpreted as an estimate of the model's capability.

Overall, YOLO11 served as a reliable baseline for sign-type and fading detection, whereas RT-DETR achieved comparable results on binary delamination and slightly better performance on missing letters. In addition, the main limitations appear to be driven less by architectural differences and more by data characteristics. Overlapping and coexisting defects, subjective severity thresholds, rare and subtle damages, and dense multi-sign scenes cap the performance of these models on these defects. Although offline augmentation and class weighting reduced class imbalance, these measures could not fully compensate for the limited availability of high-quality positive examples with unambiguous defect cues, which likely constrained generalization for the rare and subtle defect classes. This study highlighted these limits and established a practical benchmark for future work focused on better data, sharper definitions, and more specialized models for fine-grained assessment of traffic signs.

6.3 Limitations and Challenges

This study faced several limitations and challenges. First, it evaluated only two modern detectors, YOLO11 and RT-DETR, so the results did not represent the full landscape of strong CNN- and transformer-based models. Another model might perform better under the same conditions, so these results do not rule out stronger alternatives.

Second, both models relied exclusively on imagery from highway-scale collections, so these were not targeted close-up captures. Many signs appeared small in the frame, and therefore, those defects looked subtle or unclear. Hence, this imaging setup limited what the models could learn, especially for fine-grained defects.

Third, the dataset remained relatively small and highly imbalanced. Rare defects, especially broken signs, appeared too infrequently and with slight variations. The models never got the opportunity to see enough diverse examples to learn that class reliably. Future work should prioritize targeted data collection and defect-focused sampling instead of relying solely on large volumes of general highway imagery.

Fourth, the defect labels depended on a single set of visual judgments. Subjective thresholds, such as "<25%" versus ">25%" delamination, reduced label reliability. Future work should use a multi-person engineering committee for labeling, clearly define thresholds, and regularly review labels to maintain a consistent ground truth.

Finally, the study focused only on traffic signs. It did not include other critical transportation assets such as barriers, guardrails, pavement markings, or signals. Thus, a true network-level condition monitoring system will need consistent datasets, defect definitions, and evaluation standards across multiple asset types, not just traffic signs.

7.0 RECOMMENDATIONS AND IMPLEMENTATION

7.1 Recommendations

The results obtained from this study indicate that the proposed CV models can support traffic sign maintenance as decision-making tools, especially for tasks with adequate and consistent labels. In particular, the sign-type and fading/delamination models showed the most reliable performance for network-level functions such as traffic-sign-type detection and classification, and coarse defect screening (e.g., fading vs. no fading and delamination vs. no delamination).

To improve long-term consistency and model learnability, the study recommends that labels be provided by a committee of engineers or inspectors, which helps reduce variability in the labels across engineers over time. In addition, future data development should emphasize targeted collection for rare and visually subtle defects while reflecting operational deployment conditions, with multiple close-range views of broken signs and severely delaminated signs, rather than relying exclusively on highway-scale imagery in which defect evidence often remained small or ambiguous.

Finally, the study encourages open and reproducible research practices by publishing the dataset, annotation rules, and model-training code under an open-source license in a public repository. This will help in the extension of this study by other agencies and researchers. Over time, this framework can also expand to additional roadside assets, including barriers, guardrails, pavement markings, and signals.

7.2 Implementation Plan

The immediate implementation plan focuses on organizing the data, annotation rules, and training code into a well-documented online repository to support reproducibility and maintenance. With access to standard deep-learning compute (e.g., a modern GPU workstation), users can apply the shared data and code to new roadway imagery to generate an end-to-end

pipeline for assessing the condition of traffic signs. Open release of the code and data also enables independent verification, benchmarking, and community-driven improvements.

The next step forward involves building a simple web-based interface around the models, which lets users upload images of traffic signs and receive output that includes detection and classification of these signs, along with associated defect labels and confidence scores. Another application would be to link the CV outputs to the existing asset inventory, so that the output can be associated with a sign ID, location, and maintenance record, and support queries by route, defect type, and condition.

With this structure in place, the agencies can move from a research prototype to an operational, data-driven condition assessment system for traffic signs and, later, progress toward additional roadside assets.

REFERENCES

Abdel-Salam, R., Mostafa, R., & Abdel-Gawad, A. H. (2022). RIECNN: Real-time image enhanced CNN for traffic sign recognition. *Neural Computing and Applications*, *34*(8), 6085–6096. https://doi.org/10.1007/s00521-021-06762-5

Ahmed, S., Kamal, U., & Hasan, Md. K. (2022). DFR-TSD: A Deep Learning Based Framework for Robust Traffic Sign Detection Under Challenging Weather Conditions. *IEEE Transactions on Intelligent Transportation Systems*, *23*(6), 5150–5162. https://doi.org/10.1109/TITS.2020.3048878

Balali, V., & Golparvar-Fard, M. (2016). Evaluation of Multiclass Traffic Sign Detection and Classification Methods for U.S. Roadway Asset Inventory Management. *Journal of Computing in Civil Engineering*, *30*(2), 04015022. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000491

Chu, X., Zheng, A., Zhang, X., & Sun, J. (2020). Detection in Crowded Scenes: One Proposal, Multiple Predictions. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12211–12220. https://doi.org/10.1109/CVPR42600.2020.01223

Dewi, C., Chen, R.-C., Liu, Y.-T., Jiang, X., & Hartomo, K. D. (2021). Yolo V4 for Advanced Traffic Sign Recognition With Synthetic Training Data Generated by Various GAN. *IEEE Access*, *9*, 97228–97242. https://doi.org/10.1109/ACCESS.2021.3094201

Ettalibi, A., Elouadi, A., & Mansour, A. (2024). AI and Computer Vision-based Real-time Quality Control: A Review of Industrial Applications. *Procedia Computer Science, 14th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 13th International Conference on Current and Future Trends of Information and*

Communication Technologies in Healthcare (EUSPN/ICTH 2023), *231*, 212–220. https://doi.org/10.1016/j.procs.2023.12.195

Federal Highway Administration. (2023). *Manual on Uniform Traffic Control Devices for Streets and Highways* (11th ed.). U.S. Department of Transportation.

Feng, D., & Feng, M. Q. (2018). Computer vision for SHM of civil infrastructure: From dynamic response measurement to damage detection – A review. *Engineering Structures*, *156*, 105–117. https://doi.org/10.1016/j.engstruct.2017.11.018

Fredj, H. B., Chabbah, A., Baili, J., Faiedh, H., & Souani, C. (2023). An efficient implementation of traffic signs recognition system using CNN. *Microprocessors and Microsystems*, *98*, 104791. https://doi.org/10.1016/j.micpro.2023.104791

Greenhalgh, J., & Mirmehdi, M. (2012). Real-Time Detection and Recognition of Road Traffic Signs. *IEEE Transactions on Intelligent Transportation Systems*, *13*(4), 1498–1506. https://doi.org/10.1109/TITS.2012.2208909

Habibi Aghdam, H., Jahani Heravi, E., & Puig, D. (2016). A practical approach for detection and classification of traffic signs using Convolutional Neural Networks. *Robotics and Autonomous Systems*, *84*, 97–112. https://doi.org/10.1016/j.robot.2016.07.003

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hoskere, V., Narazaki, Y., Hoang, T., & Jr, B. S. (2018). *Vision-based Structural Inspection using Multiscale Deep Convolutional Neural Networks* (arXiv:1805.01055). arXiv. https://doi.org/10.48550/arXiv.1805.01055

Khajwal, A. B., Cheng, C.-S., & Noshadravan, A. (2023). Post-disaster damage classification based on deep multi-view image fusion. *Computer-Aided Civil and Infrastructure Engineering*, *38*(4), 528–544. https://doi.org/10.1111/mice.12890

Khalilikhah, M., & Heaslip, K. (2016). The effects of damage on sign visibility: An assist in traffic sign replacement. *Journal of Traffic and Transportation Engineering (English Edition)*, *3*(6), 571–581. https://doi.org/10.1016/j.jtte.2016.03.009

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. *Advanced Engineering Informatics, Infrastructure Computer Vision*, *29*(2), 196–210. https://doi.org/10.1016/j.aei.2015.01.008

Koyuncu, M., & Amado, S. (2008). Effects of stimulus type, duration and location on priming of road signs: Implications for driving. *Transportation Research Part F: Traffic Psychology and Behaviour*, *11*(2), 108–125. https://doi.org/10.1016/j.trf.2007.08.005

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Lin, Y.-K., Tu, C.-Y., Kurosawa, L., Liu, J.-H., Wang, Y.-Z., & Roy, D. (2024). Applications of Computer Vision in Transportation Systems: A Systematic Literature Review. *SHS Web of Conferences*, *194*, 01004. https://doi.org/10.1051/shsconf/202419401004

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep Learning for Generic Object Detection: A Survey. *International Journal of Computer Vision*, *128*(2), 261–318. https://doi.org/10.1007/s11263-019-01247-4

Manzari, O. N., Boudesh, A., & Shokouhi, S. B. (2022). Pyramid Transformer for Traffic Sign Detection. *2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)*, 112–116. https://doi.org/10.1109/ICCKE57176.2022.9960090

Qian, Y. J., & Wang, B. (2024). TSDet: A new method for traffic sign detection based on YOLOv5-SwinT. *IET Image Processing*, *18*(4), 875–885. https://doi.org/10.1049/ipr2.12991

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016a). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Seo, J., Duque, L., & Wacker, J. (2018). Drone-enabled bridge inspection methodology and application. *Automation in Construction*, *94*, 112–126. https://doi.org/10.1016/j.autcon.2018.06.006

Seo, J., Han, S., Lee, S., & Kim, H. (2015). Computer vision techniques for construction safety and health monitoring. *Advanced Engineering Informatics*, *29*(2), 239–251. https://doi.org/10.1016/j.aei.2015.02.001

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Soleimani-Babakamali, M. H., Askari, M., Heravi, M. A., Sisman, R., Attarchian, N., Askan, A., Soleimani, R., & Taciroglu, E. (2023). *Deep Ensemble Learning for Rapid Large-Scale Post-Earthquake Damage Assessment—Application to 2023 Kahramanmaraş Earthquake Sequence*. Research Square. https://doi.org/10.21203/rs.3.rs-3375920/v1

Spencer, B. F., Hoskere, V., & Narazaki, Y. (2019a). Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring. *Engineering*, *5*(2), 199–222. https://doi.org/10.1016/j.eng.2018.11.030

Wagner, F., Eltner, A., & Maas, H.-G. (2023). River water segmentation in surveillance camera images: A comparative study of offline and online augmentation using 32 CNNs. *International Journal of Applied Earth Observation and Geoinformation*, *119*, 103305. https://doi.org/10.1016/j.jag.2023.103305

Yang, J., Wilde, A., Menzel, K., Sheikh, M. Z., & Kuznetsov, B. (2023). Computer Vision for Construction Progress Monitoring: A Real-Time Object Detection Approach. In L. M. Camarinha-Matos, X. Boucher, & A. Ortiz (Eds.), *Collaborative Networks in Digitalization and Society 5.0* (pp. 660–672). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-42622-3_47

Zhang, L., Yang, K., Han, Y., Li, J., Wei, W., Tan, H., Yu, P., Zhang, K., & Yang, X. (2025). TSD-DETR: A lightweight real-time detection transformer of traffic sign detection for long-range perception of autonomous driving. *Engineering Applications of Artificial Intelligence*, *139*, 109536. https://doi.org/10.1016/j.engappai.2024.109536

Zhang, X., Wang, Q., Fang, H., & Ying, G. (2025). Automatic settlement assessment of urban road from 3D terrestrial laser scan data. *Journal of Infrastructure Intelligence and Resilience*, *4*(1), 100142. https://doi.org/10.1016/j.iintel.2025.100142

Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024a). *DETRs Beat YOLOs on Real-time Object Detection*. 16965–16974. https://openaccess.thecvf.com/content/CVPR2024/html/Zhao_DETRs_Beat_YOLOs_on_Real-time_Object_Detection_CVPR_2024_paper.html

Zhou, K., Wang, Z., Ni, Y.-Q., Zhang, Y., & Tang, J. (2023). Unmanned aerial vehicle-based computer vision for structural vibration measurement and condition assessment: A concise survey. *Journal of Infrastructure Intelligence and Resilience*, *2*(2), 100031. https://doi.org/10.1016/j.iintel.2023.100031

Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., & Liu, W. (2016). Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing*, *214*, 758–766. https://doi.org/10.1016/j.neucom.2016.07.009