Michigan Department of Transportation

Enhance Bridge Image Attribution Through Automated Post Image Processing

FINAL REPORT – FEBRUARY 2026

WESTERN MICHIGAN UNIVERSITY
Civil and Construction Engineering
Center of Excellence for
Structural Durability

1. Report No. SPR-1771	2. Government Accession No. N/A	3. Recipient's Catalog No. N/A
4. Title and Subtitle Enhance Bridge Image Attribution Through Automated Post Image Processing		**5. Report Date** 02/28/2026 **6. Performing Organization Code** N/A
7. Author(s) Upul Attanayake, Ph.D., P.E., Naveen Battu, B.Sc., Pang-jo Chun, Ph.D., and Rajith Weerasinghe, B.Sc., M.Sc.		**8. Performing Org. Report No.** N/A
9. Performing Organization Name and Address Western Michigan University College of Engineering and Applied Sciences 1903 West Michigan Avenue Kalamazoo, Michigan 49008-5316		**10. Work Unit No.** N/A **11. Contract or Grant No.** Contract 2022-0434z7
12. Sponsoring Agency Name and Address Michigan Department of Transportation Research Administration 8885 Ricks Road P.O. Box 30049 Lansing, Michigan 48909		**13. Type of Report & Period Covered** Final Report 09/01/2024 – 02/28/2026 **14. Sponsoring Agency Code** N/A

15. Supplementary Notes
Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration. MDOT research reports are available at www.michigan.gov/mdotresearch.

16. Abstract
Highway agencies and consultants capture thousands of images during biennial, scoping, and request for action (RFA) inspections. Most of these files are stored with limited descriptions or inconsistent naming formats. As a result, a rich data source is underutilized. Understanding the value of this rich data source for asset management, MDOT initiated this project to explore how computer vision and artificial intelligence (AI) can automatically organize, label, and analyze bridge inspection images, turning unstructured images into structured data that can directly support the asset management program. The comprehensive review of state-of-the-art literature and practice showed that the commercially available tools are limited to roadside asset management. None of those tools is capable of performing semantic segmentation to detect bridge components. The foundation models, such as OpenAI's GPT-5.1 and Google's Gemini 3 Pro, are capable of delivering descriptive answers to given prompts. However, these models are not error-proof; hallucinations, non-determinism, and the reproducibility of results remain major issues when using them for highly specific and complex tasks such as bridge image analysis. The capabilities of the Integrated Bridge Analysis System, a comprehensive model for performing semantic segmentation and damage detection of bridge components, were demonstrated. Recommendations from this project include integrating the demonstrated model into the structural inspection program as a standalone application and using it to batch-process a large volume of images in the inspection database, converting them into a rich data source.

17. Key Words	18. Distribution Statement
Bridge, Computer Vision, Defect Detection, Feature Identification, Image Analysis, Inspection, Object Detection, Segmentation.	No restrictions. This document is available to the public through the Michigan Department of Transportation.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	96	N/A

Enhance Bridge Image Attribution Through Automated Post Image Processing

Final Report
(2024 - 2026)

Project Manager: Jacob Armour, P.E.

Submitted to:

Michigan Department of Transportation

Submitted by:

Upul Attanayake, Ph.D., P.E.
Professor and Director
Center of Excellence for Structural Durability
(269) 276 – 3217
upul.attanayake@wmich.edu

Naveen Battu, B.Sc.
Graduate Research Assistant
Center of Excellence for Structural Durability
(269) 276 – 3210
naveen.battu@wmich.edu

Pang-jo Chun, Ph.D.
Project Professor
Institute of Engineering Innovation,
The University of Tokyo
+81-3-5841-0442
chun@g.ecc.u-tokyo.ac.jp

Rajith Weerasinghe, M.Sc.
Consultant
(248) 425 – 0855
rajith95@gmail.com

Western Michigan University
Civil and Construction Engineering
Center of Excellence for Structural Durability

DISCLAIMER

"This material is based upon work supported by the Federal Highway Administration under SPR OR24-008. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the Federal Highway Administration."

If you require assistance accessing this information or require it in an alternative format, contact the Michigan Department of Transportation's (MDOT) Americans with Disabilities Act (ADA) coordinator at Michigan.gov/MDOT-ADA.

ACKNOWLEDGEMENTS

This project is funded by the Michigan Department of Transportation (MDOT). The authors would like to acknowledge the support and effort of Mr. Jacob Armour for initiating this research and managing it to have a successful conclusion. The authors also wish to acknowledge the continued assistance of the Research Advisory Panel (RAP) members in advancing this study. Special thanks are due to Richard Kathrens, Structural Service Group Director at Spicer Group, Inc., for his guidance throughout the project, helping us understand the MDOT MiBRIDGE system and develop implementation recommendations.

The use of the large language models (Claude.ai and Grammarly) was limited to editorial and summarization tasks. All technical analyses, data interpretations, and conclusions are solely the work of the authors and have been independently verified.

EXECUTIVE SUMMARY

Highway agencies and consultants capture thousands of images during biennial, scoping, and request for action (RFA) inspections. The Michigan Department of Transportation (MDOT) stores inspection images in MiBRIDGE — MDOT's bridge and culvert management system — with limited descriptions and inconsistent naming conventions, leaving a rich data source largely untapped. MDOT initiated this research project to explore how computer vision and artificial intelligence (AI) can automatically organize, label, and analyze bridge inspection images — transforming unstructured image data into structured information that directly supports MDOT's asset management program. The project was designed to:

1. Develop requirements for a post image processing tool.
2. Assess the feasibility of extracting data from inspection images and integrating it into the MiBRIDGE system.
3. Identify tools capable of recognizing major bridge components, load posting signs, traffic control devices, and duplicate images.
4. Evaluate the flexibility of identified tools for use with other asset types.

State-of-the-Art and Practice Review

A comprehensive review of current literature and commercial practice revealed that existing commercial tools — such as Blyncsy and Fugro — are limited to roadside and pavement asset management. None are capable of performing semantic segmentation to detect bridge-specific components from inspection imagery.

Only four to five research groups worldwide have developed purpose-built deep learning tools for bridge component detection, typically by fine-tuning architectures pretrained on large image datasets. Reviewed models achieved promising performance metrics (80%+ accuracy for most parameters), though direct comparisons remain difficult due to inconsistent reporting standards across studies.

Two foundation generative AI models — OpenAI's GPT-5.1 and Google's Gemini 3 Pro — were also evaluated. While both demonstrated strong descriptive capability, they are not suitable for primary bridge component detection due to hallucinations, non-determinism, and an inability to accurately generate spatial bounding boxes around identified components.

Integrated Bridge Analysis System

A set of AI models compiled, trained, and validated by Prof. Pang-jo Chun at the University of Tokyo was identified as the most capable one for bridge image analysis. Its capabilities were enhanced during this project and demonstrated through a purpose-built web portal — the Integrated Bridge Analysis System. The system combines two analysis modules.

Semantic Segmentation Module

Built on the SegFormer B5 model and trained on over 10,000 images, this module performs pixel-level classification of bridge components, including abutments, bearings, concrete and steel girders, decks, piers, joints, drainage systems, and utilities. Per-class accuracy (IoU) ranges from 0.82 to 0.93, with overall model accuracy of 92.1% and mean IoU of 70%. Outputs include segmentation masks, component statistics, and area analysis.

Damage Detection Module

Powered by the YOLO11x algorithm and trained on over 10,000 images, this module detects and classifies 18 damage and deterioration types — including cracks, corrosion, spalling, delamination, joint damage, and rebar corrosion — with bounding boxes and confidence scores to support maintenance planning and risk assessment.

Additional Capabilities

The system also detects traffic control devices, identifies and reads load posting signs using optical character recognition (OCR) at over 90% accuracy, and identifies duplicate or reused images using perceptual hashing techniques — supporting database cleanup and quality control.

Conclusions

The study reached the following conclusions:

1. No commercially available tool can currently detect bridge components from inspection images using semantic segmentation.
2. Generative AI foundation models (GPT-5.1, Gemini) are not reliable for precise bridge component detection due to hallucinations and the inability to produce accurate spatial outputs.

3. The Integrated Bridge Analysis System is the most advanced available tool for bridge component segmentation and damage detection.
4. Image quality is a critical limiting factor; poor exposure conditions and lack of detail reduce AI model accuracy.
5. A human-in-the-loop (HITL) approach remains essential for validating AI/ML outputs in bridge inspection workflows.

Recommendations

1. Hands-On Training: Use project deliverables — including training manuals and annotated datasets — to build MDOT engineers' experience with pretrained models and domain-specific fine-tuning.
2. Pilot Projects: Launch targeted pilot projects with clear objectives to evaluate AI/ML integration into inspection workflows, using the requirements and image storage guidance provided in the report appendices.
3. AASHTOWare BrM Integration: As MDOT migrates MiBRIDGE data to AASHTOWare BrM, deploy the Integrated Bridge Analysis System as a stand-alone batch processing tool for MiBRIDGE images, enabling seamless future integration into MDOT business practices.
4. Manual Updates: Revise MDOT inspection manuals and develop detailed bridge/structural component guides to support AI/ML integration and establish image quality standards for inspectors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

1.1 OVERVIEW

Departments of Transportation (DOTs) recognize the transformative potential of artificial intelligence (AI) to improve the safety, efficiency, and reliability of transportation systems. While many DOTs are still exploring the capabilities, limitations, and challenges of incorporating AI into their workflow, a few DOTs have developed AI governance documents, AI risk management frameworks, and strategic plans. In January 2023, the National Institute of Standards and Technology (NIST) released an AI Risk Management Framework (RMF) in response to President Biden's Executive Order (EO) 14110 on Safe, Secure, and Trustworthy Artificial Intelligence (NIST 2024). As listed in the NIST website as one of the use cases of RMF, the City of San Jose demonstrated how it updated its AI governance by including (i) a formal citywide AI policy to establish key aspects of AI governance, (ii) education for department staff, (iii) formal mechanisms to solicit feedback from system end-users, and (iv) comprehensive evaluation procedures for AI systems in the field. In December 2024, the Texas DOT published its AI strategic plan for fiscal years 2025 – 2027, outlining strategic priorities, use cases, best practices, and recommendations for AI integration over the next 3 years. While there are different classifications of AI, the TxDOT plan lists five application-based types: Generative AI, Machine Learning, Computer Vision, Natural Language Processing, and Robotic AI (TxDOT 2024). The TxDOT strategic plan lists 217 potential AI use cases across 30 topics. For example, AI-assisted bridge inspection and deterioration prediction, predictive maintenance planning, and intelligent photo metadata generation are among the 217 possible AI use cases.

At present, the Michigan Department of Transportation (MDOT) has an AI governance document. MDOT is implementing pilot projects to assess the capabilities, limitations, and risks of incorporating AI into workflows, and to develop an AI implementation roadmap. While AI types have been implemented sporadically on a case-by-case basis, the current emphasis is on identifying application-based AI types for asset management. MDOT alone captures thousands of images during biennial, scoping, and request-for-action (RFA) inspections. Most of these files are stored with limited descriptions or inconsistent naming formats. As a result, a rich data source is underutilized. The data hidden in these structural inspection pictures is vital for implementing AI-assisted inspections, deterioration prediction, and predictive maintenance planning.

Understanding the value of this rich data source for asset management, MDOT initiated this project with the following objectives to explore how computer vision and AI can automatically organize, label, and analyze bridge inspection images, turning unstructured images into structured data that can directly support MDOT's asset management program.

1.2 OBJECTIVES AND TASKS

The project objectives are to:

1. Develop requirements for a post image processing tool.
2. Identify the feasibility for pulling the data captured from the images into the MDOT Bridge Management System and identify the method for getting the extracted data from the tool and into the MDOT Bridge Management System.
3. Identify multiple tools that can recognize at a minimum the following items.
   a) major bridge components within an image
   b) load posting signs, including load posting values
   c) traffic control devices
   d) duplicate images.
4. Evaluate tools' flexibility to be agnostic to other asset types.
5. Produce a final report and other deliverables.

1.3 REPORT ORGANIZATION

This report is organized into six chapters.

Chapter 1 introduces the research project and outlines its objectives and the organization of the report.

Chapter 2 presents an overview of MDOT's current image data management process, a comprehensive review of computer vision models for bridge component and damage detection, available techniques for identifying duplicate images, AI based asset management experience, AI based asset management data collection tools or service providers, and a review of the performance of Generative AI models.

Chapter 3 describes the capabilities of a comprehensive tool that combines two analysis modules: Semantic Segmentation and Damage Detection. The semantic segmentation analysis module was developed using the SegFormer B5 model to perform pixel-level classification of bridge components. This module produces segmentation masks, component statistics, and area analysis to support the development and maintenance of structural component inventories. The damage detection module was developed using YOLO11x to localize and classify damage in bridge components. The tool's output includes bounding boxes, damage types, and confidence scores to support maintenance planning and risk assessment.

Chapter 4 presents a summary, conclusions, and recommendations.

Chapter 5 presents the list of references cited in this report.

Appendix A presents the prompts used to analyze a set of inspection images using OpenAI's GPT-5.1 model and the resulting outputs.

Appendix B presents the prompts used to analyze a set of inspection images using Google's Gemini model and the resulting outputs.

Appendix C presents the requirements for a high accuracy automated image attribution enhancement tool.

Appendix D presents the recommended image storage options for computer vision programs developed for bridge component and defect detection.

2 STATE-OF-THE-ART AND PRACTICE REVIEW

2.1 OVERVIEW

Due to the lack of clear guides or roadmaps for exploring and adopting AI/ML in state DOTs, the NCHRP Project 23-16, "Implementing and Leveraging Machine Learning at State Departments of Transportation," started on March 24, 2022. The primary objective of this project was to advance the understanding and use of ML tools and techniques at state DOTs and other transportation agencies. The project was concluded in 2024 and included a guide and report as deliverables (TRB 2025; Townsend et al., 2024; Cetin et al., 2024). According to Cetin et al. (2024), one of the top ten ML application areas is asset management and infrastructure. The problems being solved by the application of ML techniques include (i) pavement crack detection, (ii) defect detection for railway tracks, (iii) roadway asset inventory, (iv) preventive maintenance decisions and scheduling, (v) structural health monitoring, and (vi) traffic sign and pavement marking detection (Cetin et al. 2024). As discussed later in this chapter, the number of publications and implementation cases on the use of ML techniques for structural component and damage detection has increased significantly over the last couple of years. This is primarily due to improvements in neural network architectures over the last decade, the emergence of deep learning (DL) frameworks like TensorFlow, PyTorch, and Keras, and advances in hardware (GPUs – Graphics Processing Units and TPUs -Tensor Processing Units).

Current asset management practices using ML techniques are limited to pavement and roadside asset management. Even though 35 of 50 U.S. State DOTs having programs using unmanned aerial systems (UASs) as of March 2018, with 20 of the 35 using UASs for daily operations, it is unclear how many of those agencies are using AI to analyze the collected data and if the drones are used for inspection of pavements, bridges, and other transportation infrastructure (Cetin et al., 2024). Therefore, Cetin et al. (2024) administered a web-based survey from July to September 2022 to examine the current use of ML methods and applications by highway agencies and their plans to adopt ML in the near future. Of the 43 survey responses from 29 states, 42 were from state DOTs and one from a county transportation authority. By September 2022, 9 respondents had used the applications for less than 1 year, 3 for 3-4 years, and 2 for 1-2 years. The data to date show that only a very few agencies have had more than 6 years of experience with AI/ML applications. Further, the applications are primarily limited to pavement and roadside asset inventory development and condition assessment.

This chapter describes (i) image data management in MiBRIDGE, the MDOT's bridge and culvert management system, (ii) computer vision models for bridge component and damage detection, (iii) performance of computer vision models, (iv) available techniques to identify duplicate images, (v) AI-based asset management experience and available commercial tools, and (vi) Generative AI model performance when detecting bridge components and defects.

2.2 MIBRIDGE IMAGE DATA MANAGEMENT

MiBRIDGE is the MDOT's bridge and culvert management system that allows:

- Bridge owners, engineers, inspectors, consultants, and managers to view and enter information for bridge and culvert assets across the State of Michigan
- The ability to view assets on maps based on selected criteria
- Retrieval of structure information and standardized reports, including network summaries, bridge condition reports, bridge inspection reports, inspection schedules, scour critical structures, load rating needs, work recommendations, and custom reports (MiBRIDGE 2025).

To develop computer vision tools or Agentic AI models to harness data from structural inspection images and organize them within the MDOT bridge and culvert management system (MiBRIDGE), it was necessary to understand the current process and formats used to upload and store inspection images.

A Qualified Team Leader (QTL) is responsible for uploading inspection images and supporting documents following the submission of structure inspection reports. As shown in Figure 2-1, QTL assigns a file name, category, span, and description to each uploaded image. The file name represents a specific element, damage, or a view shown in the image. The category and span are provided with pull-down menus to select predefined labels. Since there is no predefined text for the description, the content provided by the QTL results in inconsistent formatting, terminology, etc., making it challenging to use such data in Chatbots or Agentic AI models.

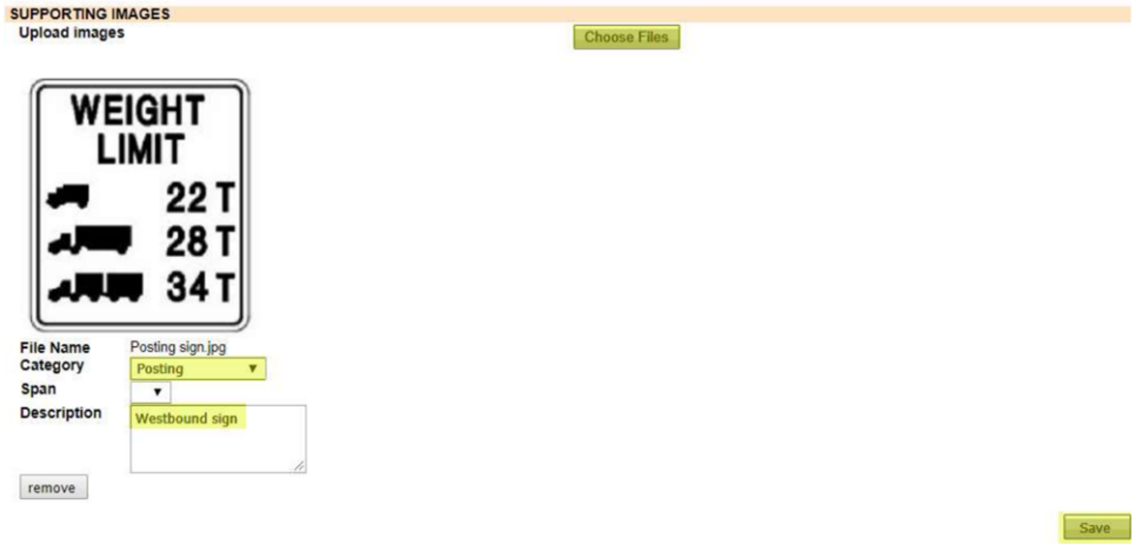


Figure 2-1. MiBRIDGE image upload window showing available options and required data.

Inspectors take photographs of bridges and structures showing general views, elements, conditions, and other features required for condition assessment and load rating. These images may also include additional features, such as traffic control devices, signs, and load posting signs. As shown in Figure 2-2, inspectors annotate certain pictures to highlight span numbers, beam numbers, specific elements, and conditions. These images are often stored temporarily in a cloud service, such as OneDrive, before being processed at the office. At that point, inspectors rename the files to match the image content, thereby serving as a simple form of annotation. Figure 2-3 shows the MiBRIDGE user interface with uploaded images. The uploaded images are stored in Bentley Systems ProjectWise. During image upload, each image is assigned a unique 15-character GUID (Globally Unique Identifier) and appended to the file name defined by the inspector.

Figure 2-2. Inspection picture with annotations.

Figure 2-3. Inspection images in MiBRIDGE, the bridge and culvert management system.

2.3 COMPUTER VISION MODELS FOR BRIDGE COMPONENT AND DAMAGE DETECTION

In recent years, the advent of deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized image recognition, as evidenced by promising results reported in the literature (Ma et al. 2020; Li et al. 2022). CNN-based methods can autonomously learn hierarchical representations of image features, empowering them to discern intricate patterns and variations within bridge structures (Cao et al. 2021). Techniques such as semantic segmentation, object detection, and instance segmentation have proven effective at extracting bridge features from images and point clouds, often surpassing the performance of traditional methods, especially in scenarios with complex backgrounds and occlusions. Moreover, previous studies, such as that by Dan et al. (2021), have underscored the significance of bridge detection from Synthetic Aperture Radar (SAR) images, given their strategic importance and practical applications. This area of research has witnessed the development of novel deep learning-based networks aimed at improving bridge detection from images (Kim et al. 2023).

Researchers have developed a variety of computer vision algorithms to automatically detect major bridge components and damage types. Table 2-1 summarizes recent studies on the recognition and detection of defects and bridge components. The "*Technique/Model*" column highlights the main architectural approach used in each study, and whether the authors used pretrained models, pretrained models with additional modifications, or new models developed by the authors, without relying on pretrained backbones or previously published networks. "*Pretrained models*" refer to using existing architectures, typically initialized with weights from large-scale datasets like ImageNet, and then fine-tuning them on domain-specific data. "*Pretrained + Modified*" refers to models that start from a known architecture and incorporate structural enhancements or task-specific modules to boost performance. Cha et al. (2017) and Wang et al. (2023) exemplify the "*Built by Authors*" category, as they created fully custom deep CNNs from scratch without using any pretrained weights. However, such work is limited to crack detection and characterization. All the other studies focused on fine-tuning or expanding pretrained architectures, reflecting a broader trend toward improving accuracy, reducing false detections, and increasing computational efficiency in bridge component and defect recognition tasks. The "*Defects and Bridge Components*" column lists basically the model capabilities. The "*Dataset Source/Size*" column shows the datasets employed for training, fine-tuning, and validation in each publication, excluding testing datasets used for evaluation.

As shown in Table 2-1, only a very few studies focused on identifying bridge components using images. Among these groups, Chun et al. (2022) used the largest dataset and identified the most components (also referred to as elements) and distress and deterioration types. Chun et al. (2022) used photographs taken during inspections of 3,118 bridges managed by the Kanto Regional Development Bureau of the Ministry of Land, Infrastructure, Transport, and Tourism of Japan. The generated dataset included 422,290 images and 504,923 explanatory texts. As shown in Figure 2-4, 38 different bridge elements and 27 distress and deterioration types were identified and presented in the text generated by their model. Please note that the bridge element and condition classification, as well as the terminology, reflect the Bridge Periodic Inspection Guideline published by the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) of Japan. The use of such models requires updating the annotated training and validation dataset to reflect the terminology in agency-specific manuals and guides, such as the Michigan Bridge Element Inspection Manual (MiBEIM 2015).

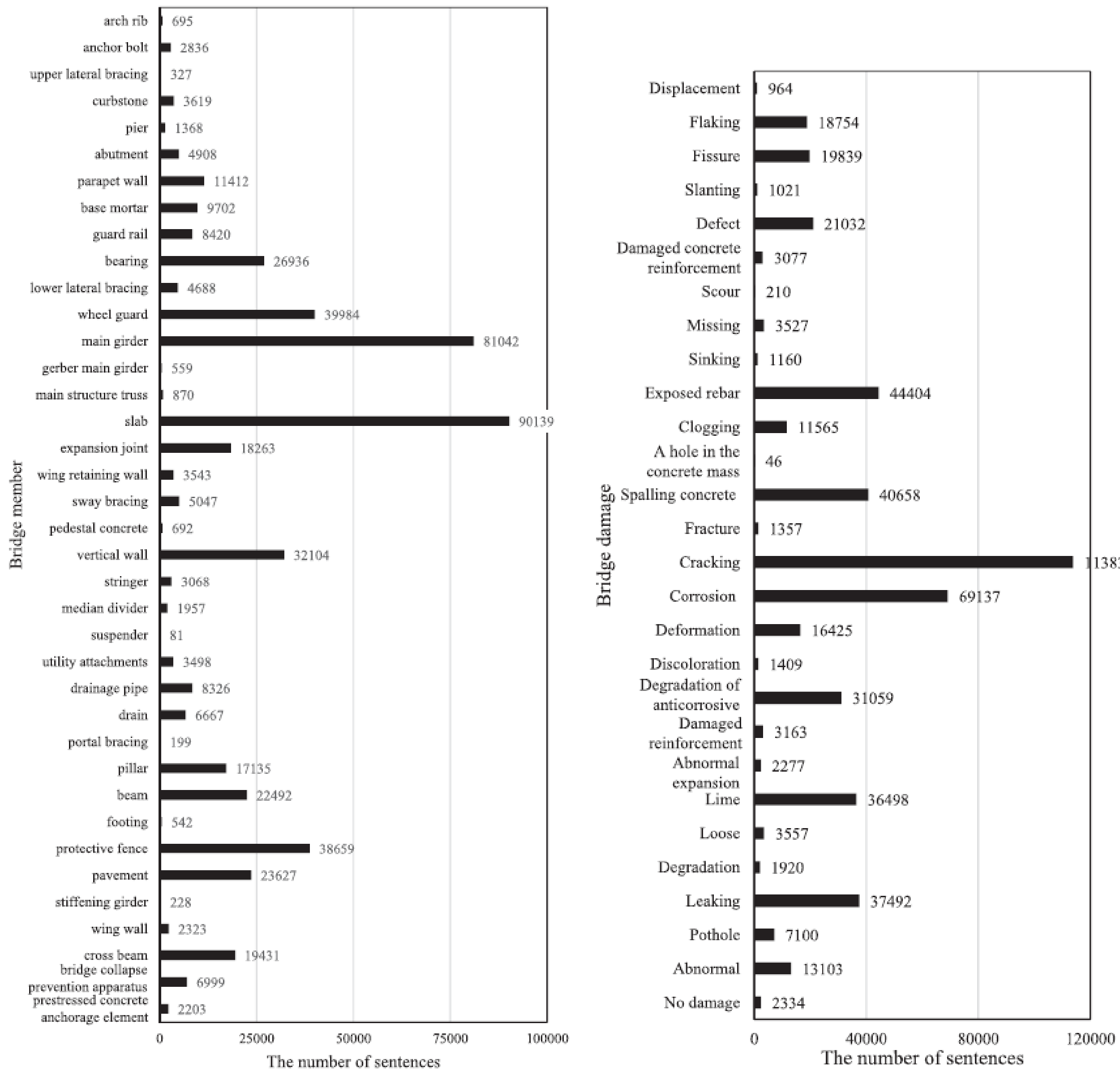
Chun et al. (2022) developed a deep learning model that generates sentences describing the damage condition of bridge elements identified from images using an image captioning method. This CNN based model generates highly accurate descriptive sentences after incorporating an attention mechanism and adds these sentences to the images as attributes. The integration of attention mechanisms is necessary when the model is expected to generate multiple statements from images with various levels of distress and deterioration (Zhang and Yuan 2022). This advanced model, as demonstrated by Chun et al. (2022), employs image-oriented intelligent methods for damage identification, focusing on surface damage classification, identification, and segmentation using deep learning CNNs. Various CNN architectures, such as LeNet-5, AlexNet, VGG16, ResNet50, and others, have been explored for tasks such as surface damage classification and identification, highlighting the versatility of these methods (Zakaria et al., 2022; Han and Yang, 2022). These techniques help identify specific features in bridge inspection images by leveraging transfer learning (TL) from pre-trained models on datasets such as ImageNet, thereby improving classification accuracy, especially when training data are limited. As listed in Table 2-1, several pretrained models are available to achieve the same objectives. However, the success of implementing such models depends on the size and diversity of the dataset used for TL, the specific purpose, and the implementation team's experience with such models.

Table 2-1. Deep Learning Techniques, Defects and Bridge Components, Datasets, and Model Classification in Recent Literature

Technique/Model	Defects and Bridge Components	Dataset Source/Size	Reference
YOLOv7 *[Pretrained]*	Cracks, spalling, delamination	11,000 images	Shafei and Odeh, 2025
SAM + U-Net Decoder *[Pretrained + modified]*	Background, abutment, bearing, decks, piers/bents, primary and secondary members	1000 bridge inspection images from WSDOT and ODOT	Wang et al., 2025
YOLOv8s *[Pretrained]*	Concrete cracks, spalling, exposed bars, corrosion stains, and efflorescence	CONBRID-YOLOv8 dataset (831 images)	Saseethar and Narkhede, 2024
DeepLabv3+ ResNet-101 *[Pretrained]*	Deck, beam, girder, pier, abutment, bearing, slab, railing, joint, culvert, waterway, and primary and secondary members	1,142 training and 127 test images	Wang and El-Gohary, 2024
Swin Transformer models + U-Net networks *[Pretrained + modified]*	Cracks	9603 training and 1695 test images	Lu et al., 2024
ProtoNet (damage detection); GoogleNet (classification) *[Pretrained + modified]*	Cracks (patch-level damage detection)	4055 crack images; 2014 background images	Gao et al., 2023
YOLOv5s-M *[Pretrained + modified]*	Longitudinal crack, transverse crack, and alligator crack	1740 training images; 580 validation images; 580 test images	Ren et al., 2023
Convolutional Neural Network (CNN) ***[Built by Authors]***	Cracks	DeepCrack Dataset: 300 training images; 237 test images CrackForest Dataset (CFD): 100 training images; 18 test images CrackTree 260 Dataset: 200 training images; 60 test images	Wang et al., 2023
CNN-based Image Captioning Model: Inception-v3 encoder + GRU decoder *[Pretrained + modified]*	38 bridge components and 27 distress and deterioration types	422,290 bridge inspection images; 504,923 explanatory texts	Chun et al., 2022
ResNet-50 (classification); YOLOv3 (detection); Mask R-CNN (segmentation) *[Pretrained]*	Piers, towers, decks, girders	1200 image (ResNet-50); 300 images (YOLOv3); 800 images (Mask R-CNN)	Yu and Nishio, 2022
SegCrack model *[Pretrained + modified]*	Cracks	1971 training images; 216 validation images; 548 test images	Wang and Su 2022
RUC-Net *[Modified]*	Cracks	CFD: 82 training images; 36 test images Crack500: 1896 training images; 348 validation images; 1123 test images. DeepCrack: 300 training images; 237 test images.	Yu et al., 2022

Table 2-1. Deep Learning Techniques, Defects and Bridge Components, Datasets, and Model Classification in Recent Literature (Contd.)

Technique/Model	Defects and Bridge Components	Dataset Source/Size	Reference
YOLOv5s + UNet with EfficientNet-B0 encoder *[Pretrained]*	Concrete cracks and spalling on bridge components (deck, soffit, girders, piers)	1,600 training images; 180 testing images	Zakaria et al., 2022
HRNet-W32 *[Pretrained]*	Bearing, Bracing, Deck, Floor Beam, Girder, Substructure, Background, and Corrosion	130 training images; 15 test images	Zhang et al., 2022
YOLOv4 *[Pretrained + modified]*	Concrete surface cracks	7000 training images; 1000 validation images; 2000 test images	Yao et al., 2021
AlexNet (Classification); Faster R-CNN (Recognition); GoogLeNet (Crack detection) *[Pretrained]*	Bridge components (tower, deck) + crack detection	3832 bridge images (AlexNet); 600 bridge images (Faster R-CNN); 1455 concrete images cropped into 60,000 images (GoogLeNet)	Zhao et al., 2018
U-Net *[Modified]*	Road surface cracks	CFD:78~79 training images; 39~40 test images. AigleRN Dataset: 25~26 training images; 12~13 test images.	Cheng et al.,2018
Multi-Scale CNN + Scene Understanding Classifier *[Pretrained]*	Columns, beams, slabs, other structural and non-structural	1652 bridge images; 3403 general images; 6842 urban images	Narazaki et al., 2017a
Multi-scale CNN + SLIC + CRF optimization *[Pretrained]*	Columns, beams, slabs, other structural, other non-structural	1652 bridge images; 3403 general images; 6842 urban images	Narazaki et al., 2017b
VGG and ResNet *[Pretrained]*	Concrete bridge deck cracks	12,000 crack and 19,500 background images	Li et al., 2022
CNN architecture (8-layer network) ***[Built by Authors]***	Concrete cracks	332 images (277 for training and 55 for validation). The training images were cropped into 40,000 images.	Cha et al., 2017



(a) Bridge elements identified from the images

(b) Distress and deterioration types

Figure 2-4. Bridge elements and deterioration/distress types identified by Chun et al. (2022).

2.3.1 Performance of Computer Vision Models

The most common performance matrices used for the assessment of computer vision programs used for component and defect detection include (i) precision, average precision (AP), recall, F1-score, and IoU (intersection over union) at the class level and (ii) the mean precision (mPrecision), mean average precision (mAP), mean recall (mRecall), mean F1-score (mF1), and mean IoU (mIoU) at the task level (Zhang et al., 2022; Wang and El-Gohary, 2024). However, the literature reports the values of these performance matrices without clearly stating whether they are mean values. Table 2-2 and Table 2-3 present the performance documented in literature for various models and datasets, as well as the models, bridge components, defects, and datasets used in such studies. The *"Model/Method"* column lists the specific deep-learning architectures evaluated in each study, along with whether they were pretrained, modified, or built entirely by the authors. The *"Dataset"* column describes the size and source of the training, validation, and test sets used to train and evaluate each model. The performance columns list several important evaluation metrics used to assess model accuracy. Finally, the *"Reference"* column cites the original studies from which each model configuration and its corresponding results were obtained. The values provided in the tables are encouraging, as they indicate a promise of assigning 80% or more to most parameters.

The programs calculate the performance matrices. However, evaluating the accuracy of the results requires implementing other methods in a human-in-the-loop (HITL) setup. Since a quantitative assessment of the accuracy of the results is required for the study conducted by Chun et al. (2022), the Bilingual Evaluation Understudy (BLEU) score was used as the evaluation index for the generated sentences. This method evaluates the similarity in sentence structure between a sentence generated by algorithms and one written by an expert in the field. Since the BLEU score does not assess the meaning of sentences, Chun et al. (2022) included an experienced bridge inspector in the process to evaluate the accuracy of the sentences generated by the model. To assess the performance of the program, the sentences generated by the model were classified as *completely correct*, *partially correct*, or *incorrect*. Through this evaluation process, Chun et al. (2022) confirmed that the program achieved 93.3% accuracy in generating sentences that were completely correct (69.3%) and partially correct (24%). As per Chun et al. (2022), the reasons for generating partially correct and incorrect sentences are the lack of details in images or poor exposure conditions. Hence, it is essential to establish an image acceptance criterion and to educate inspectors on the requirements for capturing high-quality images. Also, developing firm conclusions on the performance of various models based on the values presented in Table 2-2 and Table 2-3 are challenging primarily because of not knowing the number of components, such as girders, bearings, etc., in each image. For example, the image dataset used by Yu and Nishio (2022) included 295 images with girders and 222 images with piers, totaling 576 girders and 1110 piers, respectively. Further, the performance parameters and limits need to be carefully selected based on the objectives and refined with a diverse set of images.

Table 2-3 presents the performance of various computer vision models in detecting specific bridge components and defects. Wang et al. (2025) and Wang and El-Gohary (2024) adopted the superstructure classification framework in Hartle et al. (2002), which classifies bridge superstructure components as *primary* and *secondary*. The primary members are those that directly support vehicular live loads and form the load-bearing system of a bridge superstructure, such as girders, floor beams, trusses, and pin-and-hanger links. The secondary members do not typically resist traffic loads directly; instead, they provide stability, stiffness, and connectivity within the structural system. These components include diaphragms, cross- or X-bracing, lateral bracing, and various assembly elements such as through-bolts, pin caps, nuts, and cotter pins. The work by Wang et al. (2025) and Wang and El-Gohary (2024) underscores the importance of developing an agency-specific bridge component inventory, a roadside asset inventory, and color coding for the effective implementation of AI/ML applications.

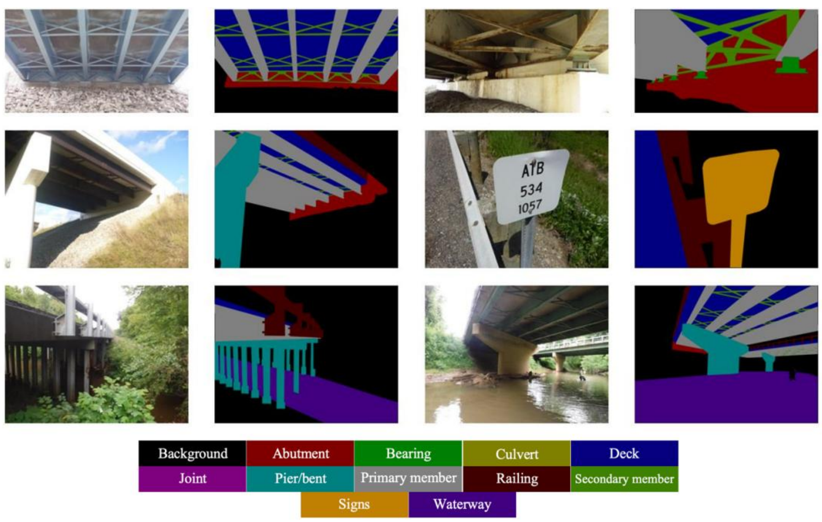


Figure 2-5. Bridge component classification used by Wang and El-Gohary (2024).

Figure 2-6. Bridge component classification used by Wang et al. (2025).

Table 2-2. Performance of Computer Vision Models When Detecting Bridge Components and Defects

Model/Method	Dataset	mIoU	mPrecision	mRecall	mF1	mAP	Reference
SAM + U-Net Decoder *[Pretrained + modified]*	Component image dataset: 900 training images; 100 test images	49.7%	85.5%	57.2%	66.2%	-	Wang et al., 2025
DeepLabv3+ ResNet-101 *[Pretrained]*	Component image dataset: 1,142 training images; 127 test images	70.4%	86.8%	78.2%	81.4%	-	Wang and El-Gohary, 2024
HRNet-W32 *[Pretrained]*	Component and corrosion image dataset: 130 training images; 15 test images	82.4 ~ 84.8 %	89.9 ~ 91.5 %	89.8~91.8 %	-	-	Zhang et al., 2022
YOLOv4 *[Pretrained + modified]*	Crack image dataset:7000 training images; 1000 validation images; and 2000 test images	-	-	-	-	94.09%	Yao et al., 2021
Model/Method	**Dataset**	**IoU**	**Precision**	**Recall**	**F1**		**Reference**
Swin Transformer models + U-Net networks *[Pretrained + modified]*	Crack images amalgamated from 12 crack datasets: 9603 training images; 1695 test images	-	96.95%	94.83%	95.61%		Lu et al., 2024
YOLOv5s-M *[Pretrained + modified]*	Pavement condition dataset (crack, Pothole, patches):1740 training images; 580 validation images; and 580 test images	-	78.2%	72.1%	75.0%		Ren et al., 2023
Convolutional Neural Network (CNN) ***[Built by Authors]***	DeepCrack Dataset: 300 training images; 237 test images	-	89.31%	82.33%	85.68%		Wang et al., 2023
	CrackForest Dataset (CFD): 100 training images; 18 test images	-	78.47%	81.29%	79.85%		
	CrackTree 260 Dataset: 200 training images; 60 test images.	-	80.55%	70.76%	75.34%		
RUC-Net *[Modified]*	CFD: 82 training images; 36 test images	58.63%	71.25%	76.80%	73.92%		Yu et al., 2022
	Crack500: 1896 training images; 348 validation images; 1123 test images	57.36%	69.88%	76.19%	72.90%		
	DeepCrack: 300 training images; 237 test images.	73.33%	88.33%	81.20%	84.61%		
SegCrack model *[Pretrained + modified]*	Crack image dataset:1971 training images; 216 validation images; 548 test images	92.63%	96.66%	95.46%	96.05%		Wang and Su, 2022

Table 2-3. Performance of Various Computer Vision Models When Detecting Specific Bridge Components or Defects

Components/ Defects	Model/Method	Dataset	IoU/mIoU	Precision/ mPrecision	Recall/ mRecall	F1/mF1	AP/mAP	Reference
Abutment	SAM + U-Net Decoder *[Pretrained + modified]*	900 training images; 100 testing images	30.9%	89.6%	33.6%	48.9%	-	Wang et al., 2025
	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	81.3%	94.1%	85.8%	89.7%	-	Wang and El-Gohary, 2024
Bearing	SAM + U-Net Decoder *[Pretrained + modified]*	900 training images; 100 testing images	68.0%	87.2%	75.4%	80.9%	-	Wang et al., 2025
	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	38.8%	80.8%	42.8%	55.9%	-	Wang and El-Gohary, 2024
	HRNet-W32 *[Pretrained]*	130 training images; 15 testing images	82.4 ~ 84.8 %*	89.9 ~ 91.5 %*	89.8~91.8 %*	-	-	Zhang et al., 2022
	Mask RCNN *[Pretrained]* (Bounding box)	305 training images; 86 validation images; 51 testing images	-	95.45%	95.45%	-	95.08%	Yu and Nishio, 2022
	Mask RCNN *[Pretrained]* (Mask)	305 training images; 86 validation images; 51 testing images	-	95.45%	95.45%	-	95.08%	
Bracing	HRNet-W32 *[Pretrained]*	130 training images; 15 testing images	82.4 ~ 84.8 %*	89.9 ~ 91.5 %*	89.8~91.8 %*	-	-	Zhang et al., 2022
Deck	SAM + U-Net Decoder *[Pretrained + modified]*	900 training images; 100 testing images	40.3%	75.7%	48.7%	59.3%	-	Wang et al., 2025
	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	83.9%	89.9%	92.7%	91.2%	-	Wang and El-Gohary, 2024
	HRNet-W32 *[Pretrained]*	130 training images; 15 testing images	82.4 ~ 84.8 %*	89.9 ~ 91.5 %*	89.8~91.8 %*	-	-	Zhang et al., 2022
Floor Beam	HRNet-W32 *[Pretrained]*	130 training images; 15 testing images	82.4 ~ 84.8 %*	89.9 ~ 91.5 %*	89.8~91.8 %*	-	-	Zhang et al., 2022

** indicates the reported values for mIoU, mPrecision, mRecall, mF1-score, and mAP*

Table 2-3. Performance of Various Computer Vision Models When Detecting Specific Bridge Components or Defects (Contd.)

Components/ Defects	Model/Method	Dataset	IoU/mIoU	Precision/ mPrecision	Recall/ mRecall	F1/mF1	AP/mAP	Reference
Girder	HRNet-W32 *[Pretrained]*	130 training images; 15 testing images	82.4 ~ 84.8 %*	89.9 ~ 91.5 %*	89.8~91.8 %*	-	-	Zhang et al., 2022
	YOLO v3 *[Pretrained]*	221 training images; 59 validation images; 15 testing images	-	93.55%	72.50%	-	79.51%	Yu and Nishio, 2022
	Mask RCNN *[Pretrained]* (Bounding box)	286 training images; 82 validation images; 33 testing images	-	87.85%	85.45%		86.51%	
	Mask RCNN *[Pretrained]* (Mask)	286 training images; 82 validation images; 33 testing images	-	82.24%	80.00%	-	79.26%	
Joint	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	50.9%	82.5%	57.1%	67.5%	-	Wang and El-Gohary, 2024
Pier/Bent	SAM + U-Net Decoder *[Pretrained + modified]*	900 training images; 100 testing images	33.7%	90.1%	39.7%	55.1%	-	Wang et al., 2025
	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	63.3%	78.7%	76.4%	77.5%	-	Wang and El-Gohary, 2024
	YOLO v3 *[Pretrained]*	172 training images; 40 validation images; 10 testing images	-	82.64%	54.95%	-	71.62%	Yu and Nishio, 2022
Railing	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	77.7%	86.3%	88.7%	87.5%	-	Wang and El-Gohary, 2024
Substructure	HRNet-W32 *[Pretrained]*	130 training images; 15 testing images	82.4 ~ 84.8 %*	89.9 ~ 91.5 %*	89.8~91.8 %*	-	-	Zhang et al., 2022
Primary member	SAM + U-Net Decoder *[Pretrained + modified]*	900 training images; 100 testing images	55.2%	86.9%	63.2%	73.2%	-	Wang et al., 2025
	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	90.9%	93.6%	97.0%	95.3%	-	Wang and El-Gohary, 2024

** indicates the reported values for mIoU, mPrecision, mRecall, mF1-score, and mAP*

Table 2-3. Performance of Various Computer Vision Models When Detecting Specific Bridge Components or Defects (Contd.)

Components/ Defects	Model/Method	Dataset	IoU/mIoU	Precision/ mPrecision	Recall/ mRecall	F1/mF1	AP/mAP	Reference
Secondary member	SAM + U-Net Decoder *[Pretrained + modified]*	900 training images; 100 testing images	43.9%	91.4%	45.5%	60.8%	-	Wang et al., 2025
	DeepLabv3+ ResNet-101 *[Pretrained]*	1,142 training images; 127 testing images	63.1%	77.3%	77.4%	77.4%	-	Wang and El-Gohary, 2024
Corrosion	HRNet-W32 *[Pretrained]*	130 training images; 15 testing images	82.4 ~ 84.8 %*	89.9 ~ 91.5 %*	89.8~91.8 %*	-	-	Zhang et al., 2022
Cracks	YOLOv4 *[Pretrained + modified]*	7k training images;1k validation images;2k testing images	-	-	-	-	94.09%*	Yao et al., 2021

** indicates the reported values for mIoU, mPrecision, mRecall, mF1-score, and mAP*

2.4 IDENTIFYING DUPLICATE IMAGES

Identifying identical images helps clean image databases and detect image reuse by inspectors when uploading images to asset management systems. Two images are identical when the Hamming distance is zero (Samanta and Jain, 2021). As needed, a threshold can be set for the Hamming distance to identify 100% identical or nearly identical images.

A hashing function is a mathematical process that transforms input data of any size, such as text, files, or images, into a fixed-length output called a hash value or digest. It functions like a digital fingerprint: the same input always produces the same hash, but even a tiny change in the input yields a completely different output. These codes act like a digital fingerprint of the image. Depending on the method, some hashes are very strict, detecting only exact binary duplicates, while others tolerate minor modifications like resizing, cropping, or color adjustments. More advanced AI-based hashing techniques can even identify semantic similarity, recognizing that two different photos may still depict the same object. The table below summarizes different hashing techniques and highlights the image variations they can detect, ranging from exact duplicates to images that are visually or semantically similar.

Table 2-4. Overview of Hashing Techniques and Their Capabilities

Category	Hashing Technique	Capabilities	References
Cryptographic Hashing	MD5	Detects exact copies only. Not tolerant to visual changes.	MD5: Rani et al., 2020 SHA 256: Modke et al., 2025
	SHA Family (SHA-1, SHA-256, SHA-3)		
	Bloom Filter Hashing	Works on binary data; useful for checking if an exact image exists in a set	Breidenbach et al., 2021
Similarity-Preserving	Locality-Sensitive Hashing (LSH)	Detects similar images by mapping feature vectors to a common space.	Kulis and Grauman, 2009
Perceptual / Image Hashing	Average Hash (aHash)	Detects similar or slightly resized/rotated images. Sensitive to major lighting or color changes.	Ofcom, 2022
	Difference Hash (dHash)	Detects near-duplicate images, tolerant to scaling, rotation, and slight color adjustments.	
	Perceptual Hash (pHash)	Detects visually similar images even with color adjustments, brightness/contrast edits, or minor cropping. Stronger than aHash/dHash.	Samanta and Jain, 2021
AI / Learning-to-Hash	Supervised Deep Hashing	Recognizes semantic similarity even with significant variations.	Liong et al., 2015
	Unsupervised Deep Hashing	Learns to group similar-looking images without labels.	Liong et al., 2015
	Pairwise/Triplet Hashing	Recognizes fine-grained similarity.	Deng et al., 2019

2.5 AI-BASED ASSET MANAGEMENT

2.5.1 Nebraska DOT Experience

The Nebraska Department of Transportation (NDOT) has completed two proof-of-concept projects using the existing image data collected by the forward-facing camera onboard the pavement profiler. The outcome of the first project was an inventory of classified guardrails and guardrail attenuators, along with their locations for the entire state highway system. The outcome of the second project is an inventory of marked pedestrian crossings and their locations.

About 1,500 images containing guardrails and guardrail attenuators were annotated and used to train a CNN model for object detection (i.e., detecting guardrails). After multiple rounds of model training and evaluation on the validation dataset, the guardrail detection models achieved accuracies of 97% and 85% for detecting the presence of guardrails and classifying them into the three types (i.e., crash cushion, Type 1, or Type 2 attenuator), respectively. The images classified as Type 1 were further analyzed to identify the five endcap types with an accuracy of approximately 95%. The CNN model analyzed roughly 2.5 million images and created a statewide inventory. The results from the object detection and classification models were used to create a GIS layer to locate relevant guardrail information as points and line features on a map (Cetin et al., 2024).

The second project implemented by NDOT, as a proof-of-concept, targeted identifying the locations of all statewide marked pedestrian crossings on the urban highway system using roadway imagery data. Cetin et al. (2024) did not report the number of images used for training and validation of the model, nor the accuracy achieved by the trained model. As a result of this study, NDOT could identify and inventory a total of 1,303 marked crosswalks on the state's urban roadway network. Due to the small number of images per crosswalk marking type (e.g., standard, zebra), the type was manually assigned after reviewing each image. Although the model's accuracy is not stated in Cetin et al. (2024), the example image in the report shows lower confidence levels due to poor image quality.

NDOT did not have an official evaluation plan to assess the level of accuracy of the ML models. However, they were satisfied with the outcome, with each project costing $25k. NDOT acknowledged the value of initiating low-cost and low-risk applications with clear objectives.

They discourage implementing complex ML applications with large project budgets, e.g., inventorying every asset and spending millions of dollars. Further, they are working with universities to develop image databases with the necessary annotations, with a limited budget of about $50k per case.

2.5.2 Caltrans Experience

Among many other initiatives and projects, the litter abatement program and asset condition monitoring and maintenance program are relevant to the work described in this report. Even though Caltrans is successfully working with external companies and consultants, it values developing inhouse capabilities to avoid potential issues arising from a lack of transparency in commercial solutions (Cetin et al. 2024).

Caltrans works with PathWeb (https://pathweb.pathwayservices.com/rip/) to monitor roadway assets such as median barriers, guardrails, signage, boxes, etc., and to use ML for object detection and condition assessment. As discussed in Section 2.5.3.1 of this report, Blyncsy, which was acquired by Bentley Systems in 2023, provides similar services (https://blyncsy.com/). PathWeb also offers a similar solution for pavement condition assessment.

According to Cetin et al. (2024), Caltrans initiated a pilot study with Payver for streamlining litter assessment and data collection by using dashboard imagery on selected roadways on a limited basis. As discussed in Section 2.5.3.1 of this report, Blyncsy and Payver are integrated systems offered by Bentley Systems (https://blyncsy.com/) and provide the capabilities to integrate the identified assets and other objects into ESRI platforms for georeferencing.

As indicated several times in this report, the accuracy of the results depends on the quality and clarity of the images, unless a large number of annotated images are used for training. Another factor contributing to accuracy is the team's experience in developing, fine-tuning, and implementing the algorithms. Caltrans also acknowledges the value of having a human-in-the-loop for evaluating the accuracy of vision-based methods. Caltrans's experience with the On-land Visual Trash Assessment (OVTA) pilot study showed that the detection model was correct 50% of the time in detecting actual trash. The classification of trash level (low, moderate, or high) was only 70% accurate when compared with the manual rating (Cetin et al. 2024).

2.5.3 Commercial Tools or Service Providers

One of the project objectives was to identify commercial tools capable of recognizing key major bridge components, load-posting signs, and traffic control devices. While there are several commercial tools for pavement condition and roadside asset detection and condition assessment, none have shown the capability to identify bridge components.

2.5.3.1 Blyncsy

Blyncsy, founded in 2014 by Mark Pittman in Salt Lake City, has been used by several DOTs and other agencies to identify roadside assets and evaluate their condition. One of the key features is its ability to use crowdsourced dashcam imagery to collect large volumes of data for AI/ML analysis. After Bentley Systems acquired it in 2023, its integration capabilities have been improved. For example, the analysis output can be integrated into ESRI platforms for georeferencing.

Table 2-5 lists several case studies published on Blyncsy's website (https://blyncsy.com/projects/). These case studies reflect the platform's work with various transportation agencies across the United States. Table 2-6 summarizes findings from this review of these case studies. The check mark (✓) indicates detection capability reported in the corresponding agency's case study.

Table 2-5. Agencies Using Blyncsy for Data Collection and Analysis

Agency/Project Highlight
Port Authority of New York and New Jersey explores significant cost savings with Blyncsy
How Plano discovered almost $500,000 in potential savings using Blyncsy's automated AI-powered platform
Alaska DOT launching weekly inspections thanks to automated crowd-sourced imagery and AI analysis from Blyncsy
Hawaii DOT switched to Blyncsy to automate roadway condition assessment and damage detection
New York City DOT launches AI for real-time roadway management and proactive maintenance

Table 2-6. Blyncsy's Detection Capabilities Utilized by Various Transportation Agencies.

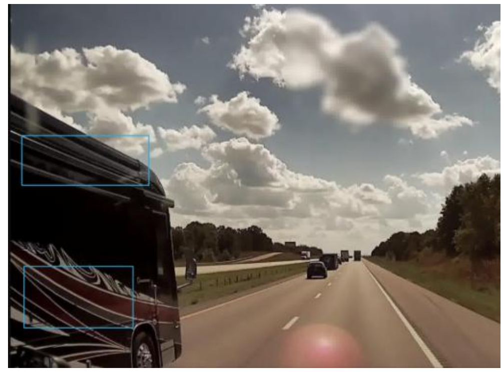
Solutions by Use Case	Port Authority Of New York And New Jersey	City of Plano	Alaska DOT	Hawaii DOT	New York City DOT
Active Transportation	✓	✓			
Barrel, Cone, and Barricade Detection				✓	
Cable Barrier Damage					
Cracking and Potholes	✓		✓	✓	
Debris				✓	
Debris Size Estimation					
Guardrail Detection and Damage			✓	✓	
Malfunctioning Street Light Detection		✓	✓		
Missing Signs	✓	✓		✓	
MUTCD Sign Inventory	✓	✓	✓		
Paintline Degradation	✓	✓			
PASER Analysis on Roadways	✓	✓		✓	✓
Quality of Road Striping	✓	✓			
Sign Damage	✓	✓			
Street Light Detection			✓		
Striping Retroreflectivity	✓	✓	✓	✓	
Vegetation Encroachment		✓		✓	

While Blyncsy provides a wide range of roadway asset detection features, a review of their publicly available U.S. Roadways Asset Map showed instances of incorrect or inconsistent predictions. These mistakes highlight the current limitations of AI-based systems in complex, real-world environments, especially when dealing with occlusions, overlapping infrastructure, or changing lighting conditions. The publicly available data can be accessed from the following link: https://blyncsy.com/explore/united-states-interstate-highway-map/#explore

Table 2-7 presents examples of incorrect predictions identified during our review of the data. These include misclassifications of objects, inaccurate bounding boxes, and missed detections of critical roadway features.

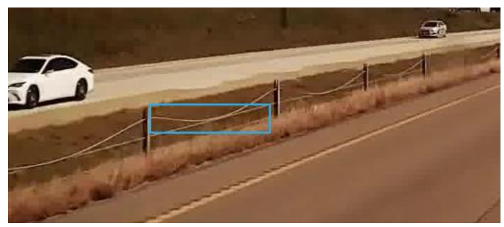
Table 2-7. Examples of Mispredictions by Blyncsy (Last accessed: 12/2/2025)

Index: 47091
Labels: guardrail
Cardinal direction: SW
X: -92.61855788
Y: 37.7073906



Index: 47248
Labels: guardrail, guardrail
Cardinal direction: NE
X: -92.60100636
Y: 37.727651612

Index: 47305
Labels: guardrail
Cardinal direction: NE
X: -92.53523283
Y: 37.75903224



Index: 46562
Labels: guardrail
Cardinal direction: SW
X: -92.230864124
Y: 37.799386466

Index: 36874
Labels: guardrail
Cardinal direction: NE
X: -91.4434286
Y: 38.05601557

Index: 37165
Labels: guardrail
Cardinal direction: SW
X: -91.36471084
Y: 38.0957815

The Alaska DOT&PF has been one of the early adopters of Blyncsy's platform, which initiated a pilot program leveraging crowd-sourced imagery and weekly automated inspections to collect roadside asset data along routes not covered by Fugro, the primary vendor**.** To better understand Blyncsy's performance in a deployed setting, a short survey and follow-up discussion were conducted.

The Alaska DOT&PF is currently assessing Blyncsy as a supplementary asset data collection tool alongside Fugro. Although the original plan called for weekly data deliveries, the agency shifted to a one-time delivery model to better align with internal workflows. Data collection continues through dashcams installed in DOT&PF vehicles, complemented by Blyncsy-installed cameras in selected regions. Alaska-specific sidewalk imagery and route network data were also provided to retrain Blyncsy's models.

Data outputs are provided through WMS (Web Mapping Services) and WFS (Web Feature Services), including attributes such as RouteIDs and Milepoints, with data layers representing sidewalks, guardrails, crosswalks, and PASER ratings. The current agreement is a one-time purchase with the first year of camera use offered at no cost; future years might involve leasing or licensing options. Privacy and security are still under review, and the agency continues to assess how Blyncsy's more frequent data collection could complement Fugro's annual road surveys.

During initial deployments, DOT&PF reported several limitations in Blyncsy's outputs. Traffic signs were mapped in ArcGIS at the location of the capturing vehicle rather than at the sign itself, leading to spatial inaccuracies, as shown in Figure 2-7. In addition, duplicate detections occurred when vehicles traveling in opposite directions captured the same signs. Blyncsy addressed this concern by deploying staff to refine duplicate filtering; however, the issue of detecting multiple signs mounted on a single post remains unresolved. These findings illustrate both the progress achieved through iterative improvements and the challenges still present in adapting Blyncsy's platform to complex roadway environments.

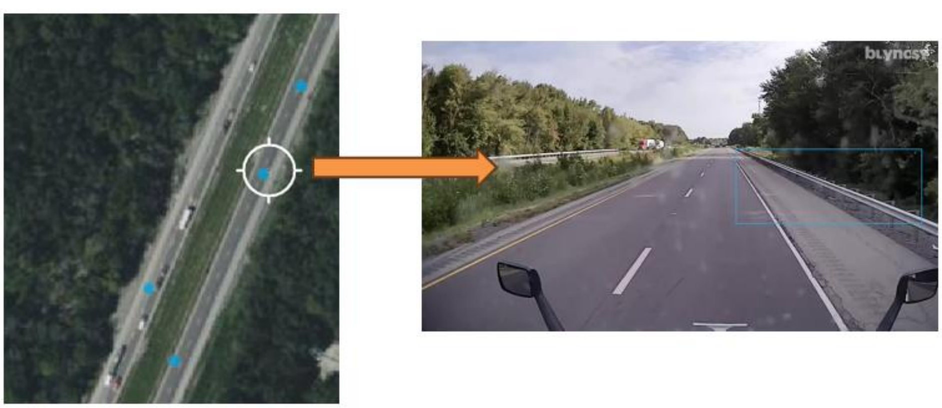


Figure 2-7. Spatial offset between the detected sign location and the actual roadside position.

According to Cetin et al. (2024), NDOT plans to identify fixed objects along the roadway and estimate their proximity to the travel lanes for safety assessment and crash risk modeling. They are exploring applicable ML methods to document such objects in image data, so that the density of potentially hazardous roadside objects per mile can be quantified across the highway network. To achieve the objectives, NDOT needs to consider incorporating additional technology to improve georeferencing accuracy.

2.5.3.2 Fugro

Fugro provides AI/ML integrated asset management solutions using data collected from vehicle-mounted sensors and cameras. Its primary data collection system, the Automatic Road Analyzer (ARAN), shown in Figure 2-8, collects high-quality pavement and roadside images, along with GPS and motion data, to create a detailed visual record of transportation assets.

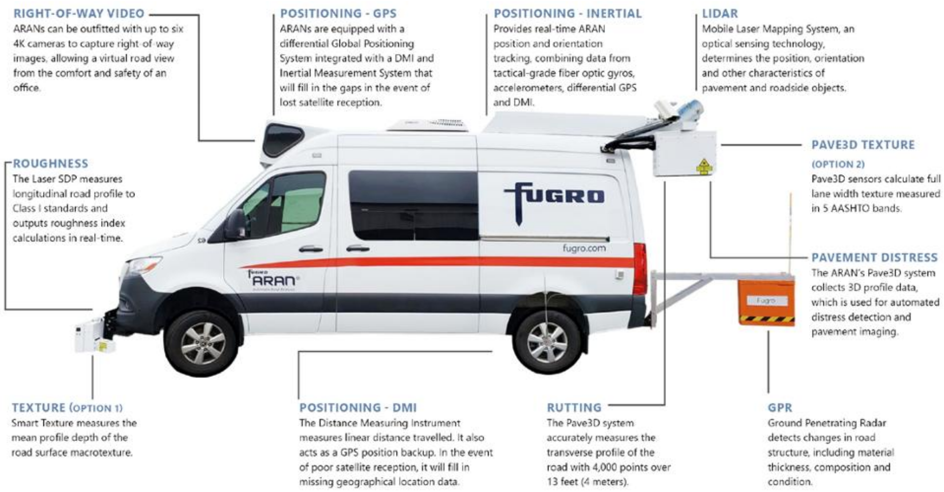


Figure 2-8. Fugro's ARAN (Automatic Road Analyzer) (Fugro 2025).

According to a proposal submitted to the North Central Texas Council of Governments on February 24, 2023, for pavement analysis and related services, Fugro claims it can develop databases for 22 right-of-way assets (TXShare 2025). Fugro provides multiple software programs for data processing and visualization. Vision software integrates key modules for data upload, georeferencing and segmentation, video and sensor data quality analysis, and pavement distress analysis, providing a complete data analysis workflow. The WiseCrax program can perform classification and severity rating, continuous full-lane or zone rating, zone detection, and crack maps for quality control of distress ratings. According to Connelly-Taylor and Annovi (2019), the WiseCrax detection pipeline with the LeNet-5 CNN architecture achieved an average precision of 90.9%, a recall of 99.9%, and an F1 score of 94.8% for automated crack detection. Although many highway agencies widely use Fugro pavement and roadside asset detection and condition evaluation, the available literature suggests that it cannot detect bridge components captured in digital images.

2.6 BRIDGE COMPONENT AND DAMAGE DETECTION USING GENERATIVE AI MODELS

Applications built with large language models (LLMs), such as OpenAI's ChatGPT and Google's Gemini, with billions of parameters, have been trained on vast amounts of data, enabling many tasks to be performed using natural language queries without extensive coding expertise. In an era when enthusiasm for using available Generative AI tools and their derivatives is high, evaluating their capabilities takes precedence over other tools under development or in use. Appendix A and Appendix B present the prompts and the outcomes when using foundation models such as OpenAI's GPT-5.1 and Google's Gemini for image analysis. The Microsoft Copilot was not evaluated since it is built on GPT-5.0. The evaluation process included a prompt requesting a list of visible elements and defects. As described with examples, these models are highly capable of delivering descriptive answers to given prompts. These models are not error-proof; hallucinations, non-determinism, and the reproducibility of results remain issues for highly specific and complex tasks such as bridge image analysis. Therefore, explainability and/or interpretability are critical to their practical implementation. According to Townsend et al. (2024), "Interpretability is about the clarity of a model or the degree to which a human can understand the cause of a decision and the inner workings of the model." Since the interpretability of these foundation models is challenging, the best approach is to explore methods to make their outputs explainable. One such approach is to prompt these models to provide visual indicators of their focus when making decisions. Therefore, additional prompts were provided requesting bounding boxes and/or labels for the identified elements and defects. As LLMs, these foundation models failed to accurately provide bounding boxes and/or labels when used as generative AI tools. Therefore, these foundation models cannot be used with confidence to achieve the primary objectives of this project, detecting bridge components. These models demonstrate mastery in explaining the most commonly available images, such as load-posting signs. Google's Gemini 3 Pro models provide comparatively better results than GPT-5.1 when identifying bridge components and defects.

Since GPT-5.1 and Gemini are Generative AI models, they can identify components, defects, and other items visible in the image by converting the details of the image to tokenized patches of information. However, they do not have the necessary tools to accurately delineate boxes around the identified components, defects, etc., using pixel mapping. Since the model does not reject the prompt, its output leads to hallucinations. An agentic AI workflow equipped with a graphical tool could help an LLM (Generative AI) generate the necessary output with bounding boxes. Furthermore, to minimize hallucinations, the LLM needs to be grounded using a Retrieval-Augmented Generation (RAG) application, with a manual that contains all the required bridge components, damage details, other relevant information, and agency-specific technical terms. These LLMs have a session memory and a long-term memory. With the long-term memory, the output is further refined. However, if the model is repeatedly given many incorrect or irrelevant prompts, its outputs can be swayed. This also contributes to the non-deterministic nature of the LLM. Another limitation is the loss of long-term memory when the LLM is updated. Several memory management tools could be integrated with the LLM within an Agentic AI workflow to address this issue. Regardless of the workflow changes, updates, training, and grounding implemented with the LLMs, accuracy and reproducibility are unlikely to match those of a purpose-built tool, such as the one discussed in Chapter 3.

3 INTEGRATED BRIDGE ANALYSIS SYSTEM

3.1 OVERVIEW

After conducting a comprehensive review of state-of-the-art and practice, it was evident that the tool developed by Prof. Pang-jo Chun at the University of Tokyo, Japan, is the only one currently capable of identifying bridge components and conditions. The capabilities of the tool were enhanced during this project to achieve the objectives. A web portal was developed during this project to demonstrate the tool's capabilities. Figure 3-1 shows the user interface. As noted in the documentation tab, this tool combines two analysis modules: Semantic Segmentation and Damage Detection. The semantic segmentation analysis module was developed using the SegFormer B5 model, pre-trained on the Cityscapes dataset at 1024×1024 resolution, and fine-tuned for bridge infrastructure via transfer learning. Training incorporates mixed-precision computation, gradient accumulation, and augmentation strategies to enhance robustness across varying field conditions. One aspect of using semantic segmentation is to perform pixel-level classification of bridge components. As an output, this module produces segmentation masks, component statistics, and area analysis to support the development and maintenance of structural component inventories. The damage detection module was developed using YOLO11x, a state-of-the-art object detection algorithm, to localize and classify damage in bridge components. The output of the combined semantic segmentation and damage detection modules includes bounding boxes, damage types, and confidence scores to support maintenance planning and risk assessment. To efficiently implement this tool on an agency server, Python 3.8+ is required, along with a CUDA-capable GPU (e.g., a GeForce RTX 5060 or higher), 8GB+ RAM, and a modern web browser (Chrome, Firefox, or Edge). Other possibilities include hosting the program in a cloud platform.

Please note that the terminology used for annotations might differ from that used in the U.S., as this program is primarily developed for a Japanese customer.

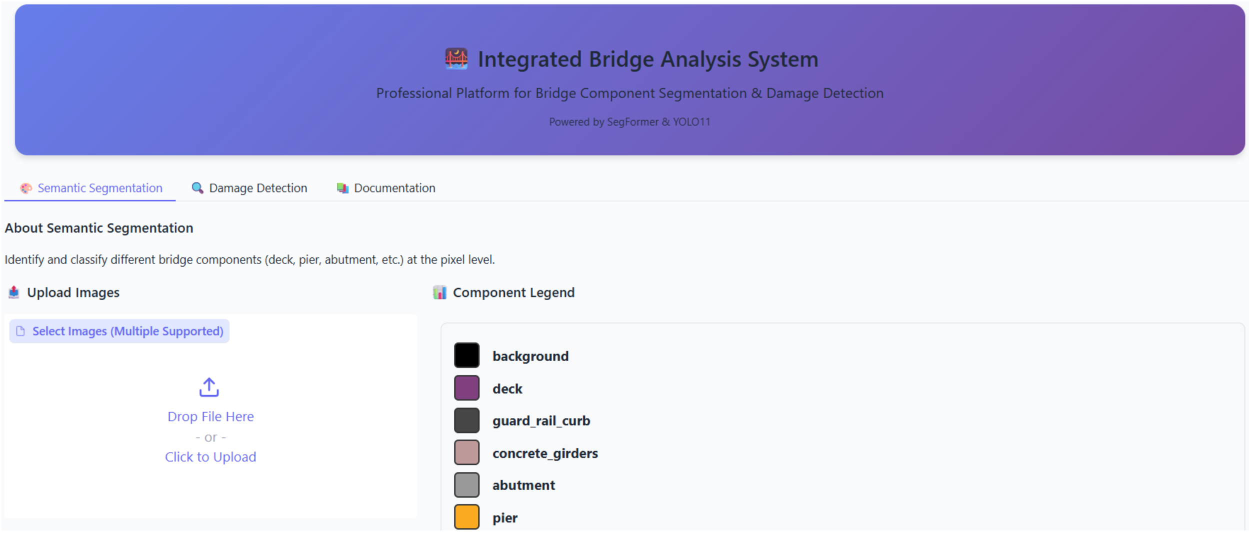


Figure 3-1. User interface of the Integrated Bridge Analysis System tool.

3.2 COMPONENT SEGMENTATION

The model has been trained with more than 10,000 images and is capable of identifying the following structural and bridge management elements:

- Abutment, including backwalls and wingwalls
- Bearings
- Concrete girders, including diaphragms
- Curb/Sidewalk/Guard rail
- Deck
- Drainage system, including the drain inlet and the drain pipe
- Joint/Expansion joint
- Pavement (deck top surface)
- Pier, including pier cap, column, wall, and footing
- Steel girders, including cross frames, diaphragms, and stiffeners
- Other utilities, including conduits and pipes.

Pixel-level segmentation used in training datasets enables precise boundary delineation and accurate area quantification for maintenance planning. Per-class IoU, Intersection over Union, which is used to evaluate object detection performance by comparing the ground-truth bounding box to the predicted bounding box, ranges from 0.82 (guard rails) to 0.93 (bridge deck), with higher scores for geometrically regular components. Figure 3-2 shows a set of annotated images used for training and validation of the program.

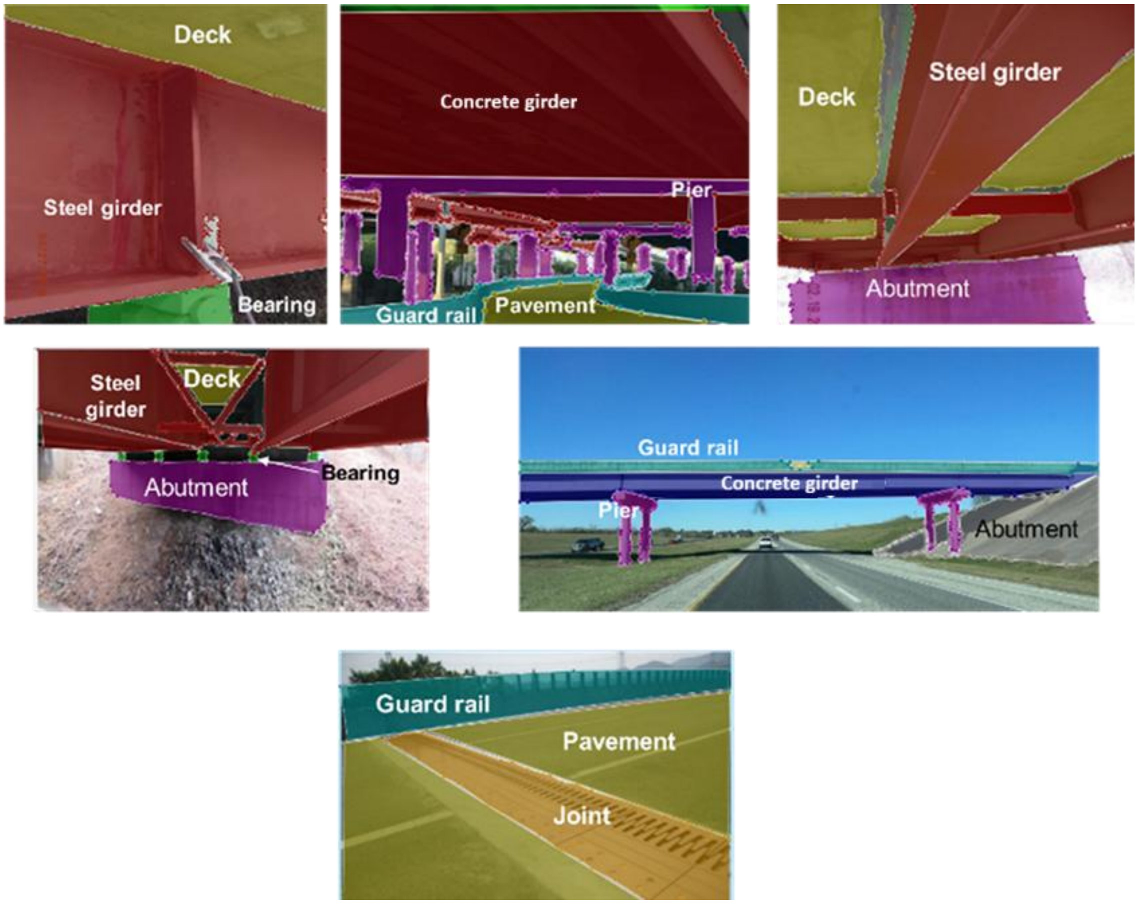


Figure 3-2. A few examples of annotated images used for component segmentation.

For example, the image shown in Figure 3-3 was submitted to the Integrated Bridge Analysis web portal, as shown in Figure 3-4. Figure 3-5 shows the segmentation results: the original image, the segmentation mask, and the segmentation mask overlaid on the original image. Figure 3-6 shows a summary of the segmentation results, including the average processing time and occurrence rate, the average percentage, and the total pixels for each component. The occurrence rate represents the presence of the identified component in the image. The average percentage represents the proportion of pixels belonging to the identified component relative to the total number of pixels in the image. The total pixels listed under each component represent the image pixels that belong to the identified component. Figure 3-7 presents the other output formats, including a *.CSV file. The program can produce output in other formats, such as JSON or TXT. Figure 3-7(d) presents an error map and performance results, an accuracy of 92.1% and the mean IoU of 70%. Figure 3-8 shows the segmentation results for a bridge with hammerhead piers, as another example.

Figure 3-3. Bridge image (H(32).jpg) used for semantic segmentation.

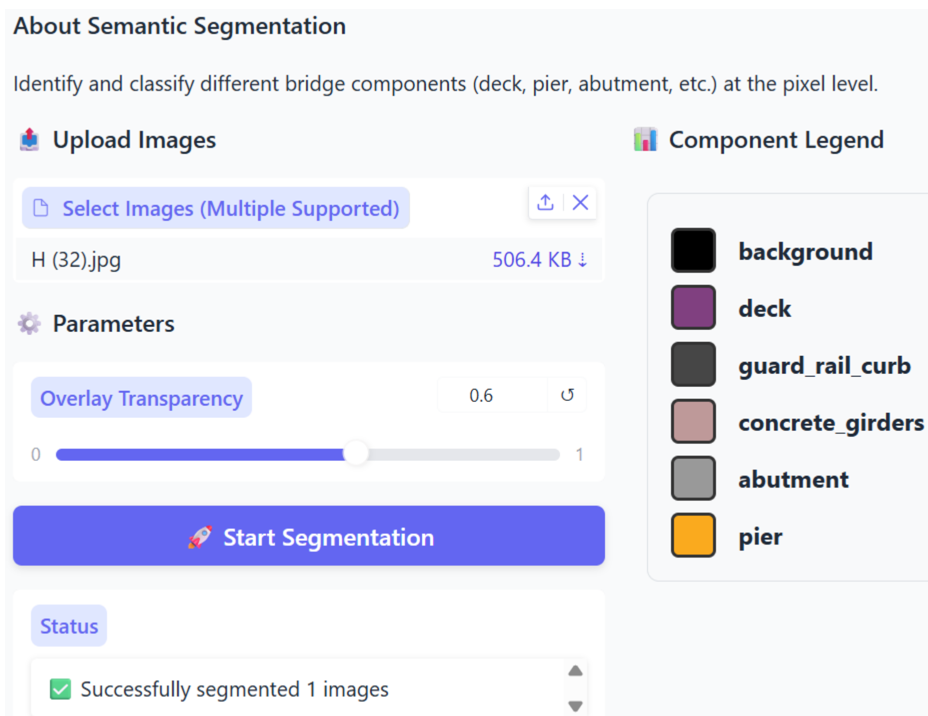
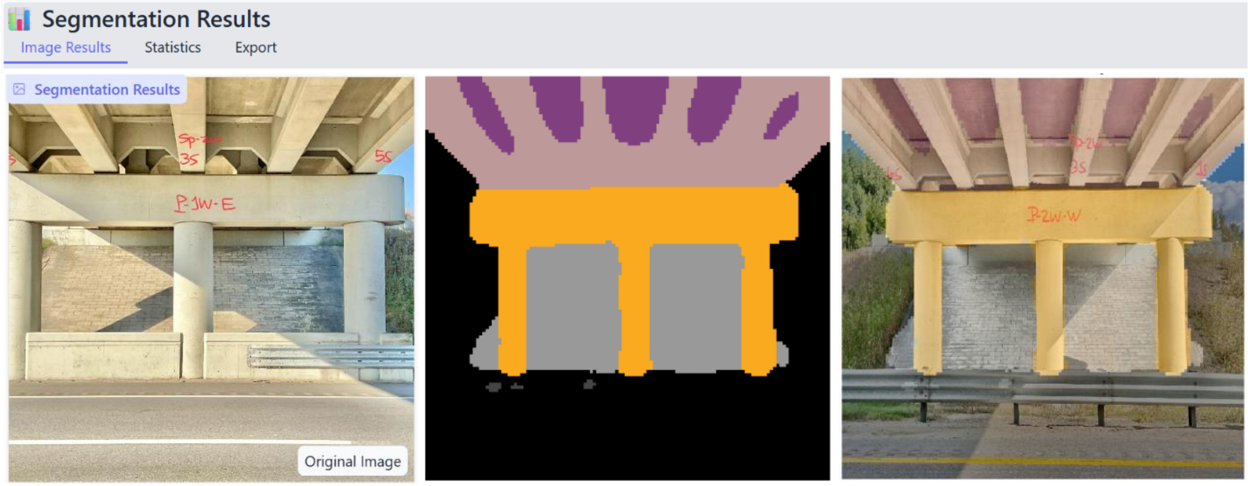
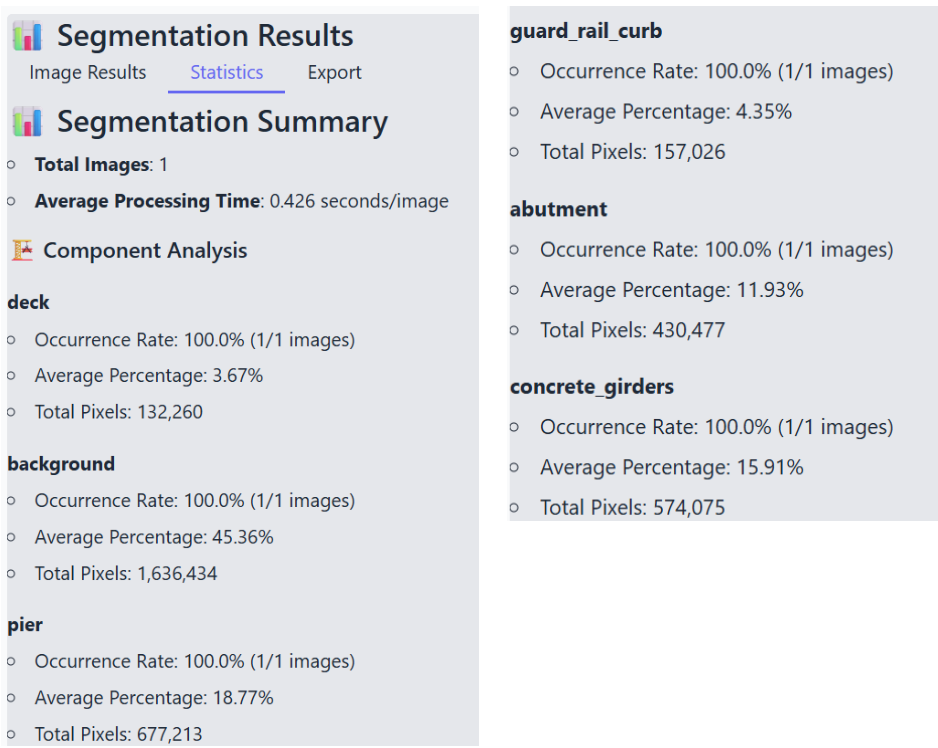


Figure 3-4. Integrated Bridge Analysis web portal with uploaded image H(32).jpg.



(a) Original image (b) Segmentation results (c) Overly

Figure 3-5. Segmentation results.



Segmentation Results

Image Results Statistics Export

Segmentation Summary

- **Total Images**: 1
- **Average Processing Time**: 0.426 seconds/image

Component Analysis

deck

- Occurrence Rate: 100.0% (1/1 images)
- Average Percentage: 3.67%
- Total Pixels: 132,260

background

- Occurrence Rate: 100.0% (1/1 images)
- Average Percentage: 45.36%
- Total Pixels: 1,636,434

pier

- Occurrence Rate: 100.0% (1/1 images)
- Average Percentage: 18.77%
- Total Pixels: 677,213

guard_rail_curb

- Occurrence Rate: 100.0% (1/1 images)
- Average Percentage: 4.35%
- Total Pixels: 157,026

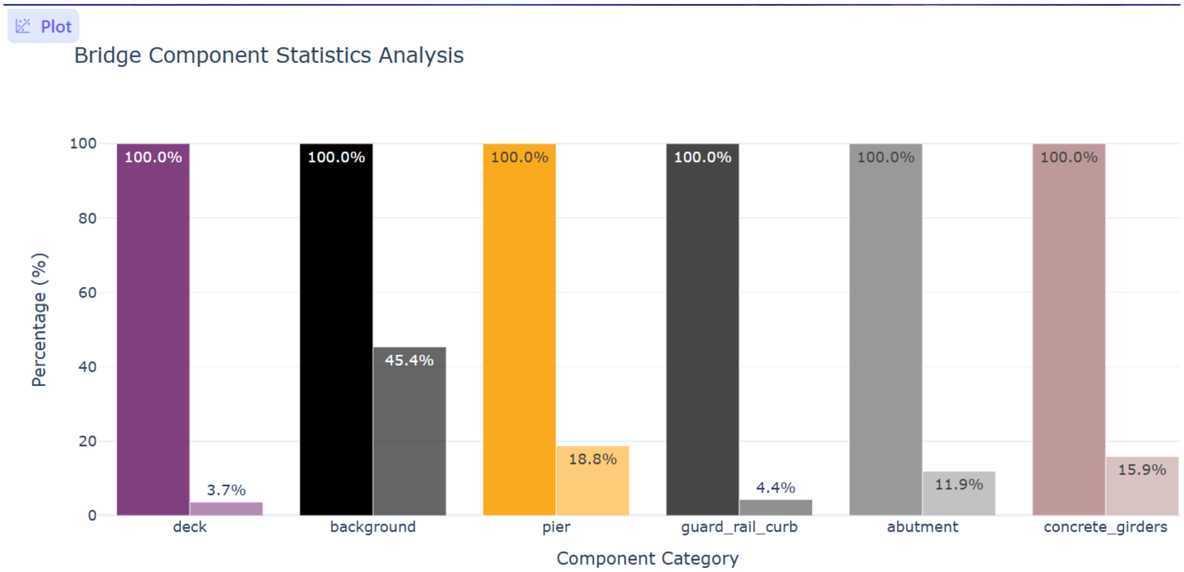
abutment

- Occurrence Rate: 100.0% (1/1 images)
- Average Percentage: 11.93%
- Total Pixels: 430,477

concrete_girders

- Occurrence Rate: 100.0% (1/1 images)
- Average Percentage: 15.91%
- Total Pixels: 574,075

Figure 3-6. Summary of segmentation results.

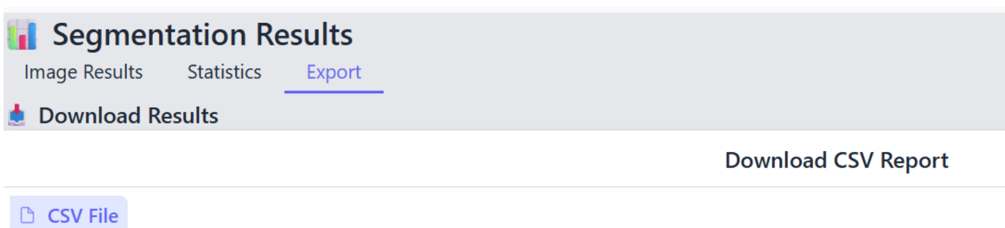


(a) Graphical presentation of output statistics

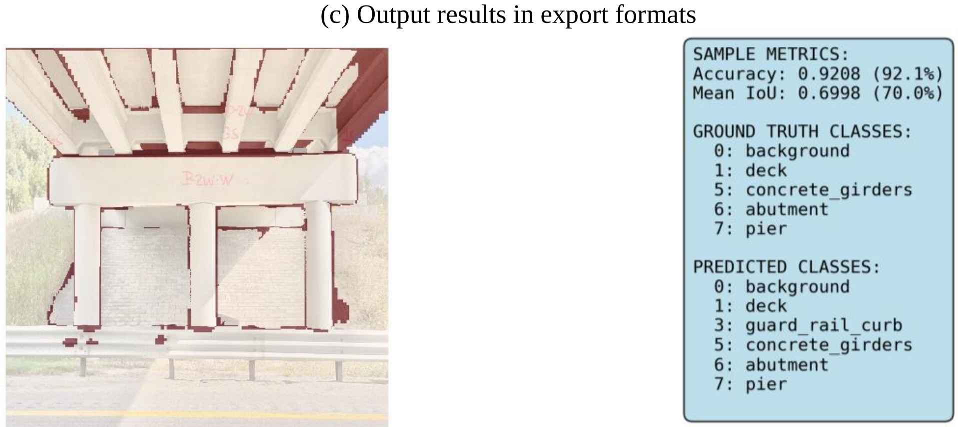
Data Preview

No.	Filename	Time (s)	background (%)	deck (%)	guard_rail_curb (%)	concrete_girders (%)	abutment (…	pier (%)
1	H 32.jpg	0.426	45.36	3.67	4.35	15.91	11.93	18.77

(b) Output statistics in tabular format



(c) Output results in export formats



(d) Error map and performance matrix

Figure 3-7. Output formats of semantic segmentation results.

Figure 3-8. Segmentation results of a bridge with a hammerhead pier.

3.2.1 Damage Detection

The model has been trained with more than 10,000 images and is capable of identifying the following damage types and conditions:

- Bearing damage
- Bearing mortar damage
- Concrete deformation/defect
- Corrosion
- Crack
- Deformation
- Delamination
- Floating
- Guardrail and barrier damage
- Joint damage

- Lime (efflorescence)
- Paint degradation
- Pavement crack
- Poor drainage
- Rebar corrosion
- Sediment accumulation
- Spall/pothole
- Water accumulation/leakage

The transformer's self-attention mechanism excels at tracing continuous crack patterns across large surfaces. Beyond detection, pixel-level segmentation enables quantification of damaged areas, mapping spatial distributions, and tracking the temporal progression of damage across inspection cycles.

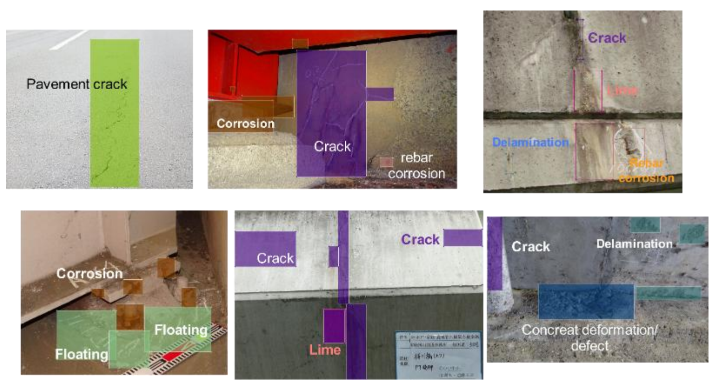


Figure 3-9. A few examples of annotated images used for damage detection.

For example, the image shown in Figure 3-10 was submitted to the Integrated Bridge Analysis web portal, as shown in Figure 3-11, for damage detection. The interface allows setting the confidence threshold for damage detection (Figure 3-12), with higher thresholds omitting hard-to-detect conditions due to image clarity. Figure 3-13 shows the damage detection results, including bounding boxes and confidence levels. For example, three cracks were identified with confidence levels ranging from 0.73 to 0.80, which depends on the quality of the image used for analysis. Figure 3-14 shows the output formats for damage detection results. Figure 3-15 shows rebar corrosion detection with bounding boxes and confidence levels ranging from 0.26 to 0.92. Figure 3-16 shows detected damages in a steel beam bridge.

Figure 3-10. Image 10.jpg submitted for damage detection.

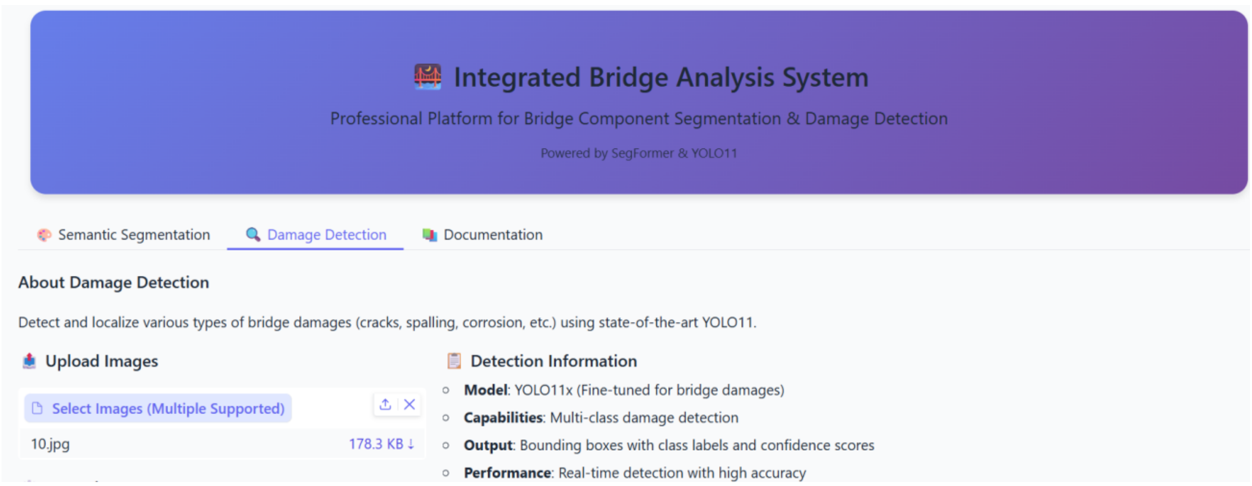


Figure 3-11. Integrated Bridge Analysis web portal with uploaded image 10.jpg.

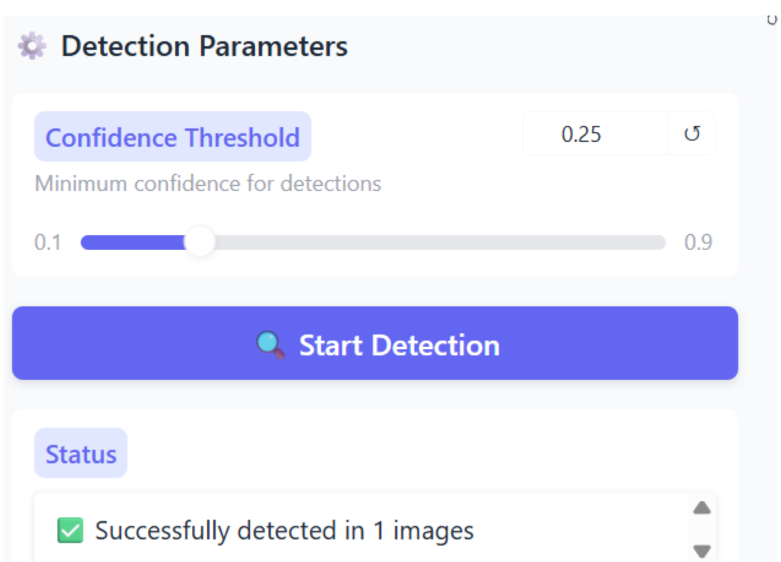


Figure 3-12. An option for setting up a confidence threshold for damage detection.

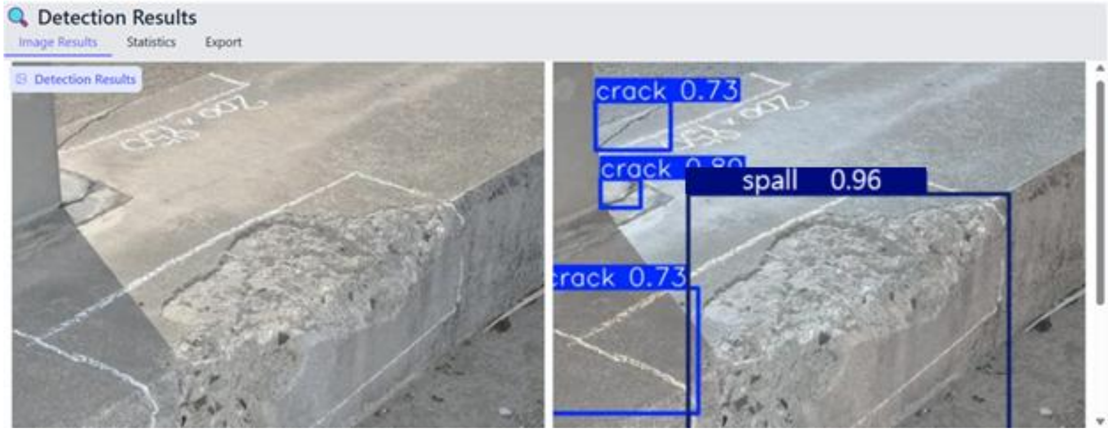
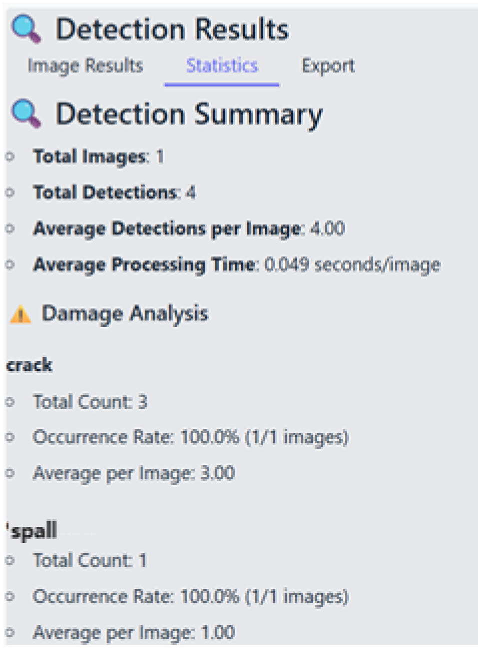
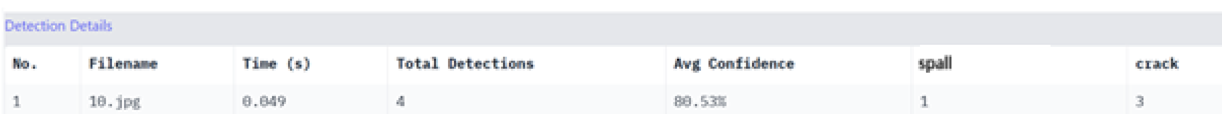


Figure 3-13. Damage detection results: damages shown in image 10.jpg.



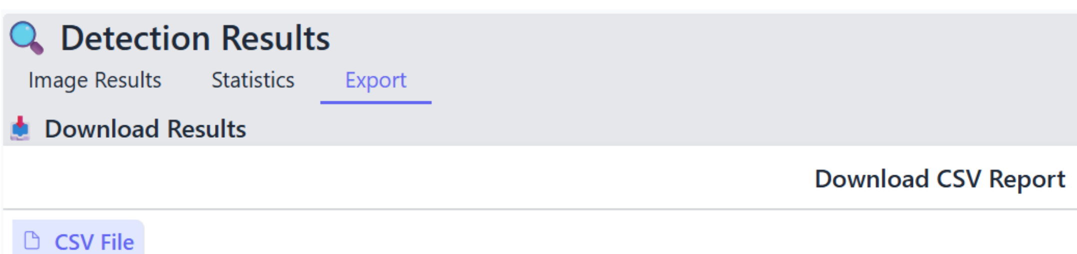
(a) Summary of detection results



Detection Details

No.	Filename	Time (s)	Total Detections	Avg Confidence	spall	crack
1	10.jpg	0.049	4	80.53%	1	3

(b) Output statistics in tabular format



(c) Output results in export formats

Figure 3-14. Output formats of damage detection results.

(a) Original image (b) Bounding boxes with confidence levels

Figure 3-15. Rebar corrosion detection.

Figure 3-16. Detection of damage in a steel beam bridge.

3.2.2 Other Capabilities

Beyond structural elements, combining YOLO v11-x object detection with segmentation models enables automated recognition of auxiliary components such as traffic signs and regulatory markers (Figure 3-17 to Figure 3-19). The pipeline first localizes sign regions and then applies the SAM 2 model to extract precise masks. Subsequently, optical character recognition (OCR) is employed to extract critical textual information (e.g., speed limits, weight restrictions). This integrated approach achieves over 90% accuracy in sign text recognition, providing comprehensive datasets for asset management. Figure 3-20 shows the detection of identical images. Two images are identical when the Hamming distance is zero. A threshold for the Hamming distance can be set to identify images as identical or nearly identical. This is a very useful option for cleaning up image databases by removing duplicates, as well as for detecting image reuse by inspectors when uploading images to asset management systems.

Figure 3-17. Detection of traffic signs and extracting textual information.

Figure 3-18. Detection of traffic control devices.

Image Name	Vehicle Type	Associated Text
East Advance Posting Sign.jpg	Load posting signs	N/A
East Advance Posting Sign.jpg	Vehicle 1	29t
East Advance Posting Sign.jpg	Vehicle 2	31T
East Advance Posting Sign.jpg	Vehicle 3	34T

Figure 3-19. Detecting load posting signs and extracting vehicle types and load limits.

(a) Image (3).JPG (b) Image (4).JPG

image1	image2	hamming_distance
Image (3).JPG	Image (4).JPG	0

(a) Hashing Results

Figure 3-20. Detection of identical images.

4 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

4.1 SUMMARY AND CONCLUSIONS

At present, the Michigan Department of Transportation (MDOT) is implementing pilot projects to assess the capabilities, limitations, and risks of incorporating artificial intelligence (AI) into its workflows. MDOT alone captures thousands of images during biennial, scoping, and request-for-action (RFA) inspections. Most of these files are stored with limited descriptions or inconsistent naming formats. As a result, a rich data source is underutilized. Understanding the value of this rich data source for asset management, MDOT initiated this project to explore how computer vision and AI can automatically organize, label, and analyze bridge inspection images, turning unstructured images into structured data that can directly support MDOT's asset management program.

The comprehensive review of state-of-the-art literature and practice showed that the commercially available tools are limited to roadside asset data collection and condition evaluation for asset management. None of those tools is capable of performing semantic segmentation to detect bridge components.

Only four to five research groups have attempted to develop purpose-built tools using existing architectures, typically initialized with weights from large-scale datasets such as ImageNet, and then fine-tuned on domain-specific data to detect and label bridge components.

The capabilities of two foundation models, OpenAI's GPT-5.1 and Google's Gemini 3 Pro, were evaluated for detecting bridge components and defects. The evaluation process included a prompt requesting a list of visible elements and defects. These models are highly capable of delivering descriptive answers to given prompts. However, these models are not error-proof; hallucinations, non-determinism, and the reproducibility of results remain major issues when using them for highly specific and complex tasks such as bridge image analysis.

The capabilities of the most comprehensive model available for performing semantic segmentation and damage detection of bridge components, the Integrated Bridge Analysis System, were demonstrated. The tool can analyze images as individual files or batches. It can present the results in various formats, such as overlays on the original images, error maps, and tables or charts of output statistics. Output can be in multiple file formats, including *.CSV and *.json.

The Integrated Bridge Analysis System can identify traffic control devices, traffic control signs, load posting signs, the content of these signs, and duplicate images.

The major drawback of using AI/ML tools for image analysis is the lack of details in images or poor exposure conditions. This requires establishing an image acceptance criterion and educating inspectors about the requirements for capturing quality images.

4.2 RECOMMENDATIONS

1) The content of the report demonstrates the challenges of implementing AI/ML tools in projects with highly specific and complex tasks, such as roadside asset detection, classification, and bridge image analysis. Therefore, it is recommended to use the training manual and datasets submitted as deliverables for this project to provide MDOT engineers with opportunities to gain hands-on experience with pretrained models and fine-tune them on domain-specific data for specific applications.

2) It is recommended to initiate pilot projects with clear objectives to understand the strengths and limitations of technology concerning the specific application, and evaluate the benefits and challenges of implementing AI/ML into the typical workflow. MDOT can use the content of the documents in *Appendix C: Requirements for a High Accuracy Automated Image Attribution Enhancement Tool* and *Appendix D: Recommended Image Storage Options for Computer Vision Programs Developed for Bridge Component and Defect Detection* to structure such projects.

3) MDOT is in the process of migrating MiBRIDGE data into AASHTOWare BrM. It is recommended to implement the Integrated Bridge Analysis System as a stand-alone program, with additional training, validation, and testing to batch-process MiBRIDGE images. This will help in understanding the process for seamless integration of the program output into the MDOT business practice. Further, the program will be trained, validated, and tested through this process to be available to MDOT inspectors and consultants as a stand-alone program to preprocess images before uploading them to MDOT servers.

4) It is recommended to revise MDOT inspection manuals and develop detailed bridge/structural component manuals to support the integration of AI/ML into the asset management workflow.

5 REFERENCES

AASHTO. (2025). AASHTO Manual for Bridge Element Inspection- 2025 Interim Revision. 2nd Edition, American Association of State Highway and Transportation Officials, 555 12th Street NW, Suite 1000, Washington, DC, 20004.

Cao, J. 2021. "Research on crack detection of bridge deck based on computer vision." *IOP Conference Series: Earth and Environmental Science*, 768 (1): 012161. https://doi.org/10.1088/1755-1315/768/1/012161.

Cetin, M., Ishak, S., Samach, M., Townsend, H., and Ozbay, K. (2024). *Implementing and Leveraging Machine Learning at State Departments of Transportation*, NCHRP Web-Only Document 404, the National Cooperative Highway Research Program (NCHRP), 500 Fifth Street, NW, Keck 360, Washington, DC 20001.

Cha, Y.-J., Choi, W., and Büyüköztürk, O. (2017). "Deep learning-based crack damage detection using convolutional neural networks." *Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361–378.

Chen, L., T. Weng, X. Jin, Z. Pan, Z. Yuan, X. Xing, and P. Zhang. 2020. "A New Deep Learning Network for Automatic Bridge Detection from SAR Images Based on Balanced and Attention Mechanism." *Remote Sensing*, 12 (3): 441. https://doi.org/10.3390/rs12030441.

Cheng, J., Xiong, W., Chen, W., Gu, Y., and Li, Y. (2018). *"Pixel-Level Crack Detection Using U-Net."* In Proceedings of TENCON 2018 – IEEE Region 10 Conference, Jeju, Korea, Oct. 28–31, 2018, 462–466.

Chun, P.-J., Yamane, T., and Maemura, Y. (2022). A deep learning-based image captioning method to automatically generate comprehensive explanations of bridge damage. Comput Aided Civ Inf, 37, 1387–1401. https://doi.org/10.1111/mice.12793

Connelly-Taylor, M., and Annovi, A. (2019). "Pavement Distress Detection Using Advanced Machine Learning Methods with Intensity and Depth Data," Pavement Evaluation 2019, Virginia Tech, Roanoke, Virginia.

Dan, D., and Q. Dan. 2021. "Automatic recognition of surface cracks in bridges based on 2D-APES and mobile machine vision." *Measurement*, 168: 108429. https://doi.org/10.1016/j.measurement.2020.108429.

Fugro (2025). 726ad3662657481a9f056f2518859a7c.pdf (Last accessed: Dec 03, 2025)

Gao, Y., Li, H., and Fu, W. (2023). "Few-shot learning for image-based bridge damage detection." *Engineering Applications of Artificial Intelligence*, 126, 107078.

https://doi.org/10.1016/j.engappai.2023.107078

Gwon, G.-H., J. H. Lee, I. Kim, and H. Jung. 2023. "CNN-Based image quality classification considering quality degradation in bridge inspection using an unmanned aerial vehicle." *IEEE Access*, 11: 22096–22113. https://doi.org/10.1109/access.2023.3238204.

Han, P., and X. Yang. 2022. "Deep Learning-Enabled Automatic Detection of Bridges for Promoting Transportation Surveillance under Different Imaging Conditions." *Journal of Advanced Transportation*, 2022: 1–16. https://doi.org/10.1155/2022/6932040.

Hartle, R. A., Ryan, T. W., Mann, E., Danovich, L. J., Sosko, W. B., and Bouscher, J. W. (2002). *Bridge Inspector's Reference Manual: Volume 1 and Volume 2*, the United States Department of Transportation. https://rosap.ntl.bts.gov/view/dot/54492.

Jo, B.-W., Y. S. Lee, J. H. Jo, and R. M. A. Khan. 2018. "Computer Vision-Based bridge displacement measurements using Rotation-Invariant Image Processing technique." *Sustainability*, 10 (6): 1785. https://doi.org/10.3390/su10061785.

Kim, H.-J., Y. Narazaki, and B. F. Spencer. 2023. "Automated bridge component recognition using close-range images from unmanned aerial vehicles." *Engineering Structures*, 274: 115184. https://doi.org/10.1016/j.engstruct.2022.115184.

Kim, H.-J., J. Yoon, and S. Sim. 2020. "Automated bridge component recognition from point clouds using deep learning." *Structural Control & Health Monitoring*, 27 (9). https://doi.org/10.1002/stc.2591.

Li, B., H. Guo, Z. Wang, and M. Li. 2022. "Automatic crack classification and segmentation on concrete bridge images using convolutional neural networks and hybrid image processing." *Intelligent Transportation Infrastructure*, 1. https://doi.org/10.1093/iti/liac016.

Li, B., Yang, F., Zhou, H., Lv, Z., and Liu, Y. (2022). "Automatic crack classification and segmentation on concrete bridge images using CNNs and hybrid image processing." *Construction and Building Materials*, 328, 126904. https://doi.org/10.1016/j.conbuildmat.2022.126904

Lu, W., Qian, M., Xia, Y., Lu, Y., Shen, J., Fu, Q., and Lu, Y. (2024). *"Crack_PSTU: Crack Detection Based on the U-Net Framework Combined with Swin Transformer."* Structures, 62, 106241. https://doi.org/10.1016/j.istruc.2024.106241

Ma, C., K.-J. Zhong, and P. Huang. 2020. "Application analysis of bridge support safety detection recognition and deep learning image processing technology." *CICTP 2020*. https://doi.org/10.1061/9780784483053.157.

MiBRIDGE (2025). Michigan Bridge Element Inspection Manual (MiBEIM), Michigan Department of Transportation, 425 W. Ottawa St., Lansing, Michigan 48909. https://www.michigan.gov/mdot/programs/bridges-and-structures/mibridge (Last accessed: 11/20/2025)

MLIT (2019). Bridge periodic inspection guideline, published by the Ministry of Land, Infrastructure, Transport and Tourism of Japan. https://www.mlit.go.jp/road/sisaku/yobohozen/tenken/yobo3_1_6.pdf (in Japanese)

Narazaki, Y., V. Hoskere, T. Hoang, Y. Fujino, A. Sakurai, and B. F. Spencer. 2019. "Vision-based automated bridge component recognition with high-level scene consistency." *Computer-Aided Civil and Infrastructure Engineering*, 35 (5): 465–482. https://doi.org/10.1111/mice.12505.

Narazaki, Y., Hoskere, V., Hoang, T. A., and Spencer, B. F. Jr. (2017). "Automated vision-based bridge component extraction using multiscale convolutional neural networks." *Proc., 3rd Huixian Int. Forum on Earthquake Engineering for Young Researchers*, Univ. of Illinois at Urbana-Champaign, Urbana, IL.

Narazaki, Y., Hoskere, V., Hoang, T. A., and Spencer, B. F. (2017). "Vision-based automated bridge component recognition integrated with high-level scene understanding." *Proc., 13th Int. Workshop on Advanced Smart Materials and Smart Structures Technology*, Univ. of Tokyo, Japan.

NIST (2024). Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, NIST Trustworthy and Responsible AI, NIST AI 600-1, National Institute of Standards and Technology, U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.600-1 : https://airc.nist.gov/airmf-resources/usecases/

Okazaki, Y., S. Okazaki, S. Asamoto, and C. Pang Jo. 2020. "Applicability of machine learning to a crack model in concrete bridges." *Computer-Aided Civil and Infrastructure Engineering*, 35 (8): 775–792. https://doi.org/10.1111/mice.12532.

Qin, R. 2020. *A Training Framework of Robotic Operation and Image Analysis for Decision-Making in Bridge Inspection and Preservation*. *http://works.bepress.com/genda-chen/517/*.

Ren, M., Liu, Y., Li, J., Ye, R., and Wang, P. (2023). *"YOLOv5s-M: An Improved Deep Learning Framework for Pavement Damage Detection from Street-View Images."* International Journal of Applied Earth Observation and Geoinformation, 120, 103335.

RiP (2024). Computer Vision Tools for Bridge Inspections and Reporting, the Transportation Research Board, Research in Progress (RiP), https://rip.trb.org/View/2190087 (last accessed: 03/22/2024)

Saseethar, S. and Narkhede, D. (2024). "Distress identification in concrete bridge inspection using computer vision and YOLOv8 framework." *Proc., ICSECT 2024*, Pune, India.

Shafei, B., and Odeh, A. (2025). "Automated assessment of defects in bridge structures: Final research report." Iowa Department of Transportation, Ames, IA.

Townsend, H., Samach, M., Cetin, M., Ishak, S., Bare, K., and Ozbay, K. (2024). *Implementing Machine Learning at State Departments of Transportation: A Guide*, NCHRP Research Report 1122, the National Cooperative Highway Research Program (NCHRP), 500 Fifth Street, NW, Keck 360, Washington, DC 20001.

TRB (2025). apps.trb.org/cmsfeed/TRBNetProjectDisplay.asp?ProjectID=5001 (Last accessed on 11/30/2025).

TxDOT (2024). Artificial Intelligence Strategic Plan, Fiscal Years 2025-2027, Texas Department of Transportation. https://www.txdot.gov/content/dam/docs/division/str/ai-strategic-plan-09-20-2024.pdf

TXShare (2025). https://txshare.org/getattachment/6f1326ff-3a52-42d0-802a-7c35fb242867/Fugro-Proposal-for-RFP-2022-063.pdf?lang=en-US (Last accessed: December 03, 2025)

Wang, J., Liu, H., Chen, Z., and Zhang, Y. (2023). *"Automated Pavement Crack Segmentation Based on Improved Deep Learning Framework."* Automation in Construction, 153, 104116.

Wang, S., Huang, Y., and El-Gohary, N. (2025). "SAM-based segmentation of multi-class bridge components from diverse real-scene inspection images." *Proc. 23rd CIB World Building Congress*, Purdue University, West Lafayette, USA.

Wang, S. and El-Gohary, N. (2024). "Semantic segmentation of bridge components from various real-scene inspection images." *Proc., 2024 ASCE CI & Construction Research Congress (CRC)*, Des Moines, IA, Mar. 20–23, 2024.

Wang, W., and Su, C. (2022). *"Automatic Concrete Crack Segmentation Model Based on Transformer."* Automation in Construction, 139, 104275. https://doi.org/10.1016/j.autcon.2022.104275

Yao, G., Sun, Y., Wong, M., and Lv, X. (2021). *"A Real-Time Detection Method for Concrete Surface Cracks Based on Improved YOLOv4."* Symmetry, 13(9), 1716. https://doi.org/10.3390/sym13091716

Yu, G., Dong, J., Wang, Y., and Zhou, X. (2023). *"RUC-Net: A Residual-Unet-Based Convolutional Neural Network for Pixel-Level Pavement Crack Segmentation."* Sensors, 23(1), 53. https://doi.org/10.3390/s23010053

Yu, W., and Nishio, M. (2022). "Multilevel structural components detection and segmentation toward computer vision-based bridge inspection." *Sensors*, 22(9), 3502.

Zakaria, M., Karaaslan, E., and Catbas, F. N. (2022). "Advanced bridge visual inspection using real-time machine learning in edge devices." *Advances in Bridge Engineering*, 3(27). https://doi.org/10.1186/s43251-022-00073-y

Zhang, C., Karim, M. M., and Qin, R. (2022). "A multitask deep learning model for parsing bridge elements and segmenting defect in bridge inspection images." *Transportation Research Record*, 2677(7), 693–704.

Zhang, Y., and K. Yuen. 2022. "Review of artificial intelligence-based bridge damage detection." *Advances in Mechanical Engineering*, 14 (9): 168781322211227. https://doi.org/10.1177/16878132221122770.

Zhang, X., B. E. Wogen, Z. Chu, S. J. Dyke, R. W. Poston, T. J. Hacker, X. Liu, L. Iturburu, P. Baah, and J. Hunter. 2022. "Machine-Supported bridge inspection image documentation using artificial intelligence." *Transportation Research Record*, 2677 (5): 720–736. https://doi.org/10.1177/03611981221135803.

Zhao, X., Li, S., Su, H., Zhou, L., and Loh, K. J. (2018). "Image-Based Comprehensive Maintenance and Inspection Method for Bridges Using Deep Learning." *Proc., ASME 2018 Conf. on Smart Materials, Adaptive Structures and Intelligent Systems (SMASIS 2018)*, San Antonio, TX.

APPENDIX A

Legend: Text highlighted in cyan represents the prompt
The text in red is commentary on the results.

Evaluation of GPT-5.1 (November 24, 2025)

Case 1:

Prompt: Identify all visible elements and defects in this image using MDOT/FHWA terminology. Consider only the elements and defects that are clearly visible. If unclear, label as 'Not clearly visible.' Provide a list of visible elements and defects.

☑ Visible Elements

1. Concrete Parapet / Bridge Rail
   - A cast-in-place concrete parapet or curb-mounted rail is present.
   - Surface finish appears weathered.
2. Bridge Deck (Edge / Fascia)
   - The concrete surface immediately behind the parapet is visible.
   - Appears to be part of the deck or sidewalk slab.
3. Pavement / Shoulder Surface
   - Concrete pavement or shoulder surface adjacent to the parapet.
4. Chalk/Field Markings
   - White field markings outlining the deteriorated repair area.
   - Measurement annotation (“200”) visible.

   Comment: Dimensions of the delaminated area are marked in the image. Due to a lack of clarity, the tool failed to recognize the other dimension.
5. Structural Post / Rail Support
   - A vertical post (likely part of the bridge railing system) is visible at the top left.
6. Debris
   - Small aggregated debris and a piece of wood lying on the pavement.

   Comment: Interestingly, ChatGPT-5.1 could detect many more details than a specially trained program for detecting the elements and defects in the trained dataset.

❌ Elements Not Clearly Visible

These components may exist, but are NOT visible enough for identification:

- Reinforcing steel (rebar) → **Not clearly visible**
- Joint sealant or expansion joint features → **Not clearly visible**
- Drainage features → **Not clearly visible**
- Barrier anchorage details → **Not clearly visible**

  Comment: This demonstrates GPT-5.1's knowledge about the typical details of this structure, but this is a hallucination and not useful for the specific purpose.

⚠ Visible Defects (Using MDOT/FHWA Terminology)

1. Concrete Spalling
   - Major spall on the parapet/deck edge.
   - Coarse aggregate fully exposed.
   - Shape and boundaries marked with field chalk.
2. Delamination (Likely but Not Confirmable)
   - Delamination often accompanies spalling, but cannot be confirmed from the image → Not clearly visible
3. Surface Scaling / Abrasion
   - The damaged area shows loss of mortar and paste, consistent with scaling or abrasion along with the spall.
4. Section Loss (Concrete)
   - Significant loss of concrete section where the spall occurred.
5. Loose or Deteriorated Concrete
   - Fragments and deteriorated concrete texture are visible inside the spalled zone.
6. Cracking
   - No clear cracks are identifiable outside the spalled region → Not clearly visible

   Comment: ChatGPT-5.1 fails to identify cracking.

📋 Summary List (Ready for Inspection Report)

Visible Elements

1. Concrete parapet/bridge rail
2. Deck/sidewalk concrete surface at parapet base
3. Pavement/shoulder surface
4. Chalk field markings
5. Rail post/support
6. Surface debris

Visible Defects

1. Concrete Spalling – major, deep, aggregate-exposed
2. Concrete Section Loss in parapet edge
3. Surface Scaling/Abrasion within damaged region
4. Loose/Deteriorated Concrete present inside spall

Defects Not Clearly Visible

1. Delamination
2. Cracking (outside spall)
3. Exposed reinforcement
4. Joint deterioration
5. Anchorage damage

FHWA ELEMENT-LEVEL CONDITION STATE ASSESSMENT

(Using AASHTO Pontis/BrM element terminology and MDOT practice)

Element 331 – Reinforced Concrete Bridge Rail (Parapet)

Comment: ChatGPT-5.1 presented an incorrect element. This picture does not show a reinforced concrete bridge rail - element 331.

Condition State Assignment (Based on Visible Area Only)

- **Condition State 1 (Good):**
  None visible in the photographed region.
- **Condition State 2 (Fair/Minor Defects):**
  None visible.
- **Condition State 3 (Poor/Moderate Deterioration):**
  - Scaling and abrasion are present at the damaged area.
  - Surface mortar loss.

    (These are present but overshadowed by more severe defects.)
- **Condition State 4 (Severe/Advanced Deterioration):**

  ✔ **Assigned**
  - Large, deep spall with coarse aggregate fully exposed.
  - Concrete section loss affecting parapet geometry.
  - Potential structural consequence depending on depth.
  - No reinforcement is visible, but presence cannot be ruled out.

→ Therefore: The visible portion of Element 331 is in *Condition State 4***.**

REPORT-READY NARRATIVE (INSPECTION-QUALITY TEXT)

Observed Deficiencies:

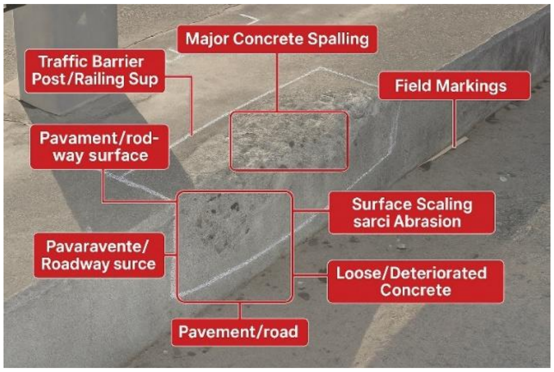
A significant concrete spall is present on the exterior face of the concrete parapet/bridge rail. The deteriorated area exhibits substantial section loss with coarse aggregate fully exposed, consistent with advanced material breakdown. Field markings outline the planned repair boundaries. The spall appears deep enough to have removed the cover concrete, although no reinforcing steel is visible at this angle. Minor scaling and abrasion are present within the deteriorated zone. *No active cracking, delamination, exposed rebar, or joint defects are clearly visible in the provided image*. Adjacent deck/sidewalk surfaces appear intact. Debris is present at the base of the parapet. The defect severity is consistent with FHWA Condition State 4 for *Element 331 (Reinforced Concrete Bridge Rail)*. Repair is recommended to prevent further degradation.

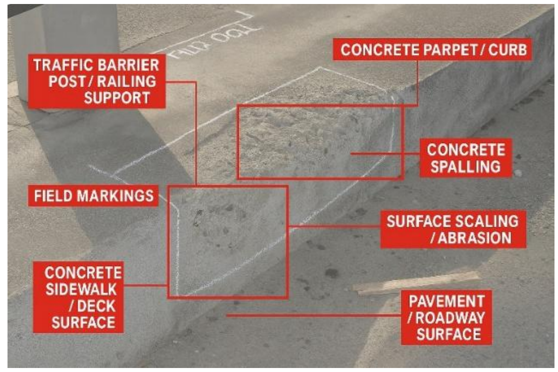
Comment: ChatGPT-5.1 presented an incorrect element and conditions. This picture does not show a reinforced concrete bridge rail - element 331. Delamination cannot be identified using digital images. Furthermore, cracking is visible in the picture, yet ChatGPT-5.1 failed to identify it.

****

Comment: Even though the previous prompt yielded seemingly accurate output, there is no clear evidence to confirm its accuracy. Therefore, the following prompt was given. The output contains significant inaccuracies and misspellings.

Prompt: provide a labeled version of the image marking each visible element and defect.





Labeled Features Shown in the Image

- Concrete parapet/curb
- Concrete spalling (major)
- Surface scaling/abrasion
- Concrete sidewalk/deck surface
- Traffic barrier post/railing support
- Field markings (repair boundary/chalk lines)
- Pavement/roadway surface
- Loose/deteriorated concrete inside the spall zone

Case 2:

Prompt: Identify all visible elements and defects in this image using MDOT/FHWA terminology. Consider only the elements and defects that are clearly visible. If unclear, label as 'Not clearly visible.' Provide a list of visible elements and defects.

☑ VISIBLE ELEMENTS

Superstructure Elements

1. Prestressed Concrete I-Beams (PCI Girders)
   - Multiple girders visible.
   - Bottom flanges, webs, and lower portions clearly visible.

2. Diaphragms/Cross-frames
   - Prestressed concrete *intermediate diaphragms* visible between girders.

   Comment: ChatGPT-5.1 could not discriminate between intermediate and end diaphragms. The picture includes end diaphragms, but they are recognized as intermediate diaphragms.

3. Deck Underside (Concrete Slab Underside)
   - Visible from below; appears uniform in color.

Substructure Elements

1. Pier Cap/Pier Cap Beam
   - Reinforced concrete cap supporting girders.
2. Pier Columns (Round RC Columns)
   - Two circular reinforced-concrete columns supporting pier cap.
3. Pier Footing/Pedestal Block (Visible Portion)
   - Rectangular concrete *pedestals* supporting columns.

   Comment: ChatGPT-5.1 incorrectly identified the crash wall as a pedestal.
4. Retaining/Slope Protection System
   - Concrete block slope paving behind the pier.
5. Guardrail/Approach Traffic Barrier
   - Steel W-beam guardrail attached to concrete parapet/sleeper base.

Roadway/Adjacent Site Elements

1. Roadway Surface (Asphalt Pavement)
2. Roadway Shoulder
3. Embankment/Vegetation
4. Sun Shadow Pattern (Not an element but relevant for visibility)

✕ ELEMENTS NOT CLEARLY VISIBLE

- Bearings → Not clearly visible (obscured by pier cap/beam).
  Comment: ChatGPT-5.1 could not recognize them even though they were slightly visible under beam 3S.
- Reinforcing steel → Not clearly visible.
- Deck joints → Not clearly visible.
- Abutments (full height) → Not clearly visible.
  Comment: ChatGPT-5.1 could not recognize the stub abutment. It seems like the program is only looking for full height abutments.
- Pier foundation → Not clearly visible.
- Beam ends/end diaphragms → Not fully visible, only interior bays seen.
  Comment: ChatGPT-5.1 could not discriminate between intermediate and end diaphragms. The picture includes end diaphragms, but they are recognized as intermediate diaphragms.

⚠ VISIBLE DEFECTS (MDOT/FHWA Terminology)

Superstructure Defects

Beam Surface Staining

- Minor surface discoloration on girder webs/flanges.
- No cracking or spalls clearly visible.

Beam Efflorescence

- Slight whitening on some girder edges—possible efflorescence, but not strongly evident → Not clearly visible.

Spalling Marking (Field Annotation "Sp-350")

- The red text indicates an inspector-marked spall location, but the spall itself is NOT visually identifiable in the photo → Not clearly visible.
  Comment: ChatGPT-5.1 misinterpreted this. The red text shows "SP-2W" and "3S". This is meant for the 2nd west span and the 3rd beam from the south. This marking has nothing to do with concrete spall.

Substructure Defects

Column Surface Abrasion / Minor Staining

- Light surface scratches/streaks on the right column.

Surface Wear on Pier Cap

- Minor staining and dust accumulation; no cracks or spalls visible.

Concrete Block Slope Protection – Uniform / No Defects Visible

- No missing blocks, no displacement, no vegetation growth through blocks.

Traffic Barrier / Guardrail Defects

Guardrail Minor Surface Corrosion

- Light discoloration consistent with early-stage corrosion.
  Comment: ChatGPT-5.1 misinterpreted this. There is no corrosion on the guardrail. The shadow might have been interpreted as corrosion.

Guardrail Alignment

- Appears straight; no impact damage visible.

❌ DEFECTS NOT CLEARLY VISIBLE

- Concrete cracking → Not clearly visible.
- Concrete spalling (actual physical spall, separate from annotation) → Not clearly visible.
- Delamination → Not clearly visible.
- Scaling → Not clearly visible.
- Exposed rebar → Not visible.
- Pier settlement / rotation → Not clearly visible.
- Drainage issues → Not clearly visible.
- Beam end deterioration → Not clearly visible.

Comment: This is another example of hallucination. It generates a seemingly correct list. For example, it lists delamination, which is not possible to detect from a typical digital image.

📋 FINAL SUMMARY (COPY-READY FOR INSPECTION REPORT)

Visible Elements

- PCI girders
- Intermediate diaphragms
- Concrete slab underside
- Concrete pier cap
- *Two RC round columns* *(Comment: Pier has three columns.)*
- *Concrete pedestals/pier bases*
- Block slope paving
- Guardrail with concrete seat
- Asphalt pavement
- Embankment

Visible Defects

- Minor staining on girders
- Minor abrasion / streaking on right column
- Light corrosion on guardrail
- Surface dusting/staining on pier cap
- Inspector markings (Sp-350, 1S, 5S, etc.), but no actual spalls visible

Defects Not Clearly Visible

- Cracking
- Spalling (physical)
- Delamination
- Exposed rebar
- Concrete scaling
- Bearing condition
- Beam end deterioration

If you'd like, I can also provide:

☑ **FHWA condition states** for each visible element
☑ A **narrative inspection paragraph**
☑ A **defect diagram overlay**
Just let me know!

Prompt: Yes, please

ChatGPT said:
Thought for 13s
Here's everything for this bent based on what's **clearly visible in the photo**.

---

1) FHWA / MDOT ELEMENT-LEVEL CONDITION STATE ASSESSMENT

(Limited to visible faces in the photo)

Element 104 – Prestressed Concrete Girders

(Incorrect. Element 109 is in MiBEIM and the AASHTO Manual for Bridge Element Inspection.)

- **Observed condition:**
  - Uniform concrete color.
  - No clearly visible cracking, spalling, delamination, or section loss.
  - Very light surface staining/dust only.
- **Condition State Assignment:**
  - **CS1 (Good):** 100% of visible girder area.
  - CS2–CS4: 0% (no structural or durability distress visible).

Element 12 – Reinforced Concrete Deck (Underside)

(Element 12 is listed in the AASHTO Manual for Bridge Element Inspection for the reinforced concrete deck. According to MiBEIM, the element number should be in the 800 series depending on the type of reinforcement used in the deck. For example, MiBEIM Element 811 represents the reinforced concrete deck bottom surface.)

- **Observed condition:**
  - Underside of deck appears sound and uniform.
  - No visible leakage, efflorescence, rust staining, cracking, or spalls.
- **Condition State Assignment:**
  - **CS1 (Good):** 100% of visible deck underside.
  - CS2–CS4: 0%.

Element 234 – Reinforced Concrete Pier Cap

- **Observed condition:**
  - Surface generally smooth with uniform color.
  - No visible cracking, spalling, delamination, or exposed reinforcement.
  - Minor dirt/staining only.
- **Condition State Assignment:**
  - **CS1 (Good):** 100% of visible cap area.
  - CS2–CS4: 0%.

Element 215 – Reinforced Concrete Columns

(Incorrect. Element 205 is in MiBEIM and the AASTO Manual for Bridge Element Inspection. Element 215 is a reinforced concrete abutment.)

- **Observed condition:**
  - Two round RC columns.
  - No visible structural cracking, spalling, delamination, or section loss.
  - Right column has faint surface marks/abrasion and light staining.
- **Condition State Assignment:**
  - **CS1 (Good):** 100% of visible column surface (abrasion/stains treated as minor cosmetic only).
  - CS2–CS4: 0%.

Element 806 – Slope Protection (Concrete Block / Paving)

(Incorrect. No specific element has been assigned for slope protection. Element 806 in MiBEIM is a reinforced concrete slab with stainless steel bars.)

- **Observed condition:**
  - Concrete block slope paving intact.
  - No missing or displaced blocks, significant vegetation, or settlement visible.
- **Condition State Assignment:**
  - **CS1 (Good):** 100% of visible slope protection.
  - CS2–CS4: 0%.

Element 333 – Metal Bridge Rail / Guardrail (with Concrete Seat)

(Incorrect. Element 330 is in MiBEIM and the AASTO Manual for Bridge Element Inspection. Element 333 is other bridge railing.)

- **Observed condition:**
  - W-beam guardrail appears straight and aligned.
  - No impact deformation evident.
  - Slight discoloration/light surface corrosion visible.
- **Condition State Assignment:**
  - **CS1 (Good):** ~90–100% (depending on agency practice for very light corrosion).
  - **CS2 (Fair – Minor corrosion):** 0–10% (localized light surface corrosion only).
  - CS3–CS4: 0%.

---

2) REPORT-READY NARRATIVE (INSPECTION PARAGRAPH)

Example language you can paste into an inspection report:

The inspected intermediate pier consists of prestressed concrete I-girders supported on a reinforced concrete pier cap and two reinforced concrete round columns founded on concrete pedestals. Concrete block slope paving is present behind the pier, and a steel W-beam guardrail is mounted on a concrete seat in front of the slope paving. The visible portions of the prestressed concrete girders, deck underside, pier cap, columns, and slope paving are in good condition with no observable structural cracking, spalling, delamination, or exposed reinforcement. Minor, uniform surface staining is present on several concrete elements, and the right column shows faint surface marks/abrasion that appear cosmetic only. The guardrail lies in proper alignment with no apparent impact damage; slight surface discoloration is consistent with early-stage corrosion. Based solely on the visible surfaces in the photograph, all primary structural elements (girders, deck underside, cap, and columns) are assigned to Condition State 1, with the guardrail in Condition State 1 with localized areas that may be considered Condition State 2 for minor corrosion per MDOT/FHWA guidance. No maintenance needs are indicated from the visible defects at this location.

---

3) TEXT-BASED DAMAGE MAPPING DIAGRAM (ELEVATION VIEW)

You can use this as a guide for a CAD overlay or sketch.

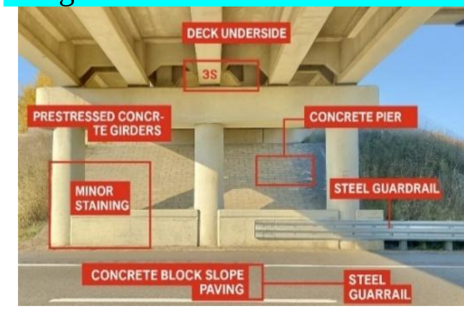
```
              SUPERSTRUCTURE – ELEVATION (PIER VIEW)

        [G1]      [G2]      [G3]      [G4]      [G5]
      -------------------------------------------------  Deck underside
      |                                               |
      |   Minor surface staining only                |
      -------------------------------------------------
                 REINFORCED CONCRETE PIER CAP
      [CS1 – no cracks/spalls; light surface staining]

      O                   O
      |                   |
      |                   |    Right column:
      |                   |    - faint surface marks/abrasion
      |                   |    - otherwise sound
   [Column Left – CS1]  [Column Right – CS1]

  +------------------+  +------------------------------+
  |  Concrete base   |  | Concrete base + Guardrail  |
  |  (CS1)           |  | (W-beam, minor surface     |
  |                  |  | corrosion – CS1/CS2)       |
  +------------------+  +------------------------------+

           CONCRETE BLOCK SLOPE PAVING (BEHIND PIER)
    - Blocks continuous, no visible displacement or loss
    - No significant vegetation or settlement (CS1)

Legend:
  CS1 – Good condition, no observable structural distress.
  CS2 – Localized minor surface corrosion on guardrail only.
  No spalls, delamination, or exposed rebar visible.
```

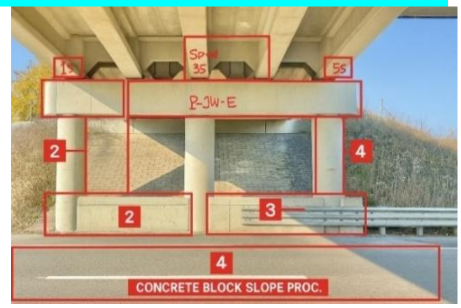
If you'd like, I can next turn this into:

- A **tabular element-level summary** (ready for BrM or Excel), or
- A **standard MDOT bridge-inspection form paragraph block** tailored to your district's format.

****

Prompt: Identify all visible elements and defects in this image using MDOT/FHWA terminology. Consider only the elements and defects that are clearly visible. If unclear, label as 'Not clearly visible. Return the same image with red bounding boxes drawn around each recognized element and/or defect. Provide a list of visible elements and defects.





Comment: Labels and bounding boxes are all over. The numbers provided in the figure are not explained.

Below is the full response including:

- ☑ The **labeled image** (generated above) with red bounding boxes and element/defect tags
- ☑ A complete **MDOT/FHWA-terminology list of visible elements**
- ☑ A complete **MDOT/FHWA-terminology list of visible defect**

Only features **clearly visible** in the photograph are listed. Anything uncertain is labeled **"Not clearly visible."**

☑ VISIBLE ELEMENTS (MDOT/FHWA TERMINOLOGY)

Superstructure Elements

1. Prestressed Concrete I-Beams / Girders
2. Intermediate Diaphragms (Concrete)
3. Deck Underside / Soffit

Substructure Elements

1. Reinforced Concrete Pier Cap / Cap Beam
2. Reinforced Concrete Columns (Round)
3. Concrete Pedestals / Column Footings (Visible Portions)
4. Concrete Block Slope Protection (Paving Blocks)

Roadside / Site Elements

8. Steel Guardrail (W-Beam)
9. Concrete Guardrail Seat / End Block
10. Roadway Surface (Asphalt Pavement)
11. Vegetated Embankment

⚠ VISIBLE DEFECTS (MDOT/FHWA TERMINOLOGY)

Concrete Defects

Minor Surface Staining (Columns / Cap)

- Light discoloration visible on pier column and cap.

Minor Abrasion / Surface Wear

- Subtle marks on the right column.

No visible cracking → Not clearly visible

No spalling visible on pier, cap, girders → Not clearly visible
(Note: "Sp-01" text is field annotation, not visible physical spalling.)

Steel Defects

Light Surface Corrosion on Guardrail
- Uniform discoloration.

No impact damage to guardrail → Not clearly visible

❌ NOT CLEARLY VISIBLE

- Bearing assemblies → Not clearly visible
- Beam end deterioration → Not clearly visible
- Concrete cracking in superstructure → Not clearly visible
- Delamination → Not clearly visible
- Spalling on superstructure → Not clearly visible
- Joint seals / drainage → Not clearly visible

🖼 LABELED IMAGE

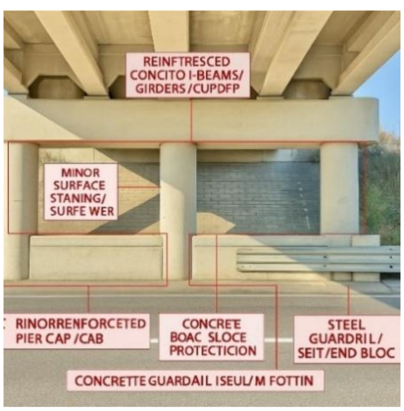
A labeled version of the image—with red bounding boxes and MDOT-style labels for each identified element and defect—has been generated above (see last generated image).
If you'd like, I can also:

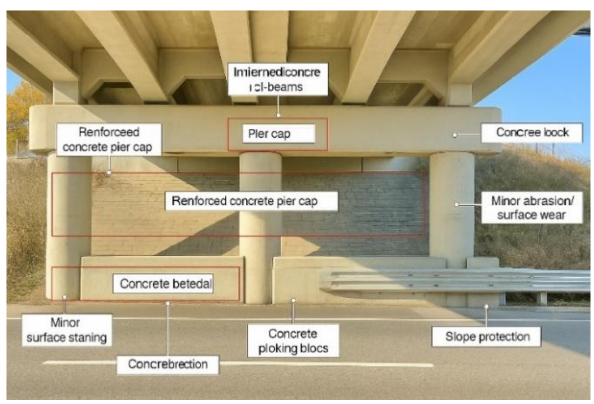
☑ Produce a **cleaner engineering-style markup** (white boxes, arrows, tags)
☑ Add **condition states (FHWA BrM)** for each element
☑ Provide a **narrative inspection paragraph** for a formal report

Just tell me!

****

Prompt: Produce a cleaner engineering-style markup





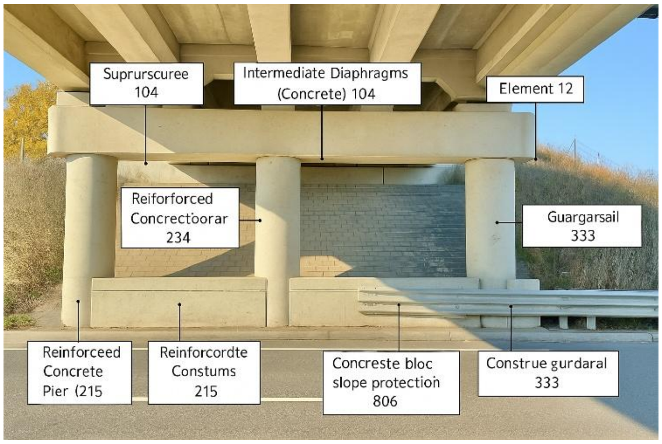
Comment: Labels and bounding boxes are all over with a lot of misspellings.

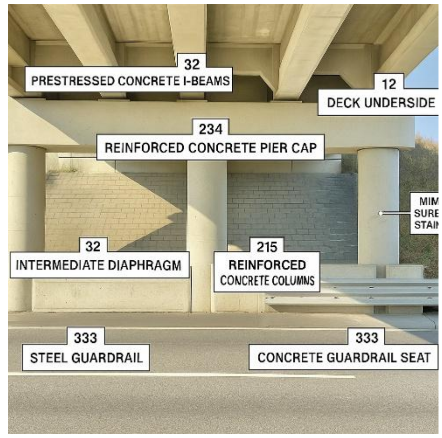
Your **clean engineering-style markup** has been generated above — the last image includes:

✔ Clean white labels
✔ Straight leader lines
✔ Clear bounding boxes
✔ Professional engineering-style annotation format

****

Prompt: Please produce a version with standardized MDOT element numbers





Your **MDOT element-numbered engineering markup** has been generated above.
The last image includes standard MDOT / AASHTO BrM element numbers, professionally labeled on each structural component:

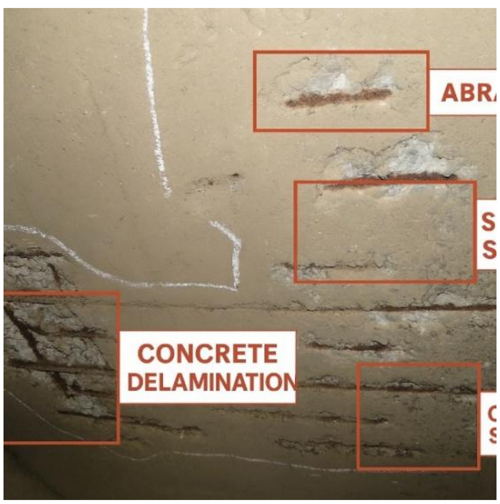
- ✔ 104 – Prestressed Concrete I-Beams
  (Incorrect. Element 109 is in MiBEIM and the AASHTO Manual for Bridge Element Inspection.)
- ✔ 104 – Intermediate Diaphragms (Concrete)
  (Incorrect. No element is defined for diaphragms.)
- ✔ 12 – Concrete Deck Underside
  (Element 12 in the AASHTO Manual for Bridge Element Inspection is reinforced concrete deck. According to MiBEIM, Element 811 is reinforced concrete deck bottom surface.)
- ✔ 234 – Reinforced Concrete Pier Cap
- ✔ 215 – Reinforced Concrete Columns (Round)
  (Incorrect. Element 205 in MiBEIM and the AASTO Manual for Bridge Element Inspection is a reinforced concrete column. Element 215 is a reinforced concrete abutment.)
- ✔ 215 – Concrete Pedestals / Footings (Visible Portions)
  (Incorrect. Element 220 in MiBEIM and the AASTO Manual for Bridge Element Inspection is the reinforced concrete pile cap/footing. Element 215 is a reinforced concrete abutment.)
- ✔ 806 – Concrete Block Slope Protection
  (Incorrect. No specific element has been assigned for slope protection. Element 806 in MiBEIM is a reinforced concrete slab with stainless steel bars.)
- ✔ 333 – Steel Guardrail
  (Incorrect. Element 330 in MiBEIM and the AASTO Manual for Bridge Element Inspection is a steel guardrail. Element 333 is assigned to other bridge railing.)
- ✔ 333 – Concrete Guardrail Seat / End Block
  (Incorrect. Element 333 in MiBEIM and AASTO Manual for Bridge Element Inspection is assigned to other bridge railing.)

The markup follows a clean engineering style with white text boxes and leader lines.

Comment: Incorrect element numbers with a lot of misspellings.

Case 3:

Prompt: Identify all visible defects in this image using MDOT/FHWA terminology. Consider only the clearly visible defects. If unclear, label as 'Not clearly visible.' Return the same image with red bounding boxes drawn around each recognized defect. Provide a list of visible elements and defects.



Below is a clear, inspection-grade response listing **all visible elements** and **all clearly visible defects** in the provided image using **MDOT/FHWA terminology**, along with the **image marked with red bounding boxes** (generated above).

Comment: Concrete delamination is indicated in the image with bounding boxes. However, the list of defects provided on the next page lists these as spalls. Further, the original image is altered and presented with the results.

☑ VISIBLE ELEMENTS

Because the image shows only a *small portion of a concrete structural member*, the only visible element is:

1. Reinforced Concrete Structural Member (likely deck underside or slab/soffit)

   No other structural components are visible.

⚠ VISIBLE DEFECTS (MDOT/FHWA TERMINOLOGY)

All defects listed below are clearly visible in the photo.

1. Concrete Spalling
    - Multiple spalls exposing reinforcement.
    - Loss of concrete section around rebar.
    - Shallow and deep spalls present.
2. Exposed Reinforcing Steel
    - Several reinforcement bars are exposed.
    - Visible due to loss of concrete cover.
3. Corrosion of Reinforcement (Rusting Rebar)
    - Rust staining and corrosion products visible on exposed bars.
    - Indicates active corrosion and likely cause of spalling.
4. Delamination (Surface Separation)
    - Several areas show delaminated concrete around spalled zones.
    - Chalk markings around the delaminated regions.
5. Concrete Cracking
    - Cracks visible adjacent to spalled areas.
    - Includes both longitudinal and transverse cracking relative to rebar.
6. Loose / Deteriorated Concrete
    - Multiple locations where concrete is fractured, loose, or flaking.

❌ DEFECTS NOT CLEARLY VISIBLE

- Efflorescence → **Not clearly visible**
- Leakage / staining → **Not clearly visible**
- Scaling outside spall zones → **Not clearly visible**
- Structural deformation (sagging/deflection) → **Not clearly visible**

🖼 LABELED IMAGE WITH RED BOUNDING BOXES

A version of the image with clearly marked defect bounding boxes has been generated above (see the most recent image produced). It includes bounding boxes around:

- Major spalls
- Exposed/corroded rebar
- Delaminated regions
- Cracked/deteriorated concrete

If you would like a **clean engineering-style markup** with labels (e.g., “Spall 1,” “CS4,” “Exposed Rebar,” etc.), I can generate that next.

If you want, I can also produce:

☑ FHWA **condition state assignment (CS1–CS4)** for this member

☑ A **formal inspection narrative** (MDOT or FHWA format)

☑ A **repair recommendation summary** (patching, removal, epoxy coating, etc.)

Just let me know

****

Case 4:

Prompt: Identify and summarize the content in this image.

The image shows a **roadway weight-limit sign** posted on a metal pole in a wooded area. The sign displays **truck weight restrictions** for different vehicle configurations.

Content of the Sign

- **Header:** *WEIGHT LIMIT*
- **Weight limits by vehicle type:**
  1. **Single-unit truck: 42 tons**
  2. **Tractor + single trailer: 63 tons**
  3. **Tractor + double trailer / multi-unit combination: 77 tons**

Additional Observations

- The sign is slightly tilted.
- There appears to be **tape or paper** wrapped around the top of the pole.
- Dense green foliage surrounds the sign.

If you'd like, I can also:

- Interpret whether this is a **load-posted bridge sign** (likely).
- Provide an **engineering summary for a bridge inspection report**.
- Create a **bounding-box markup** labeling the sign elements.

Comment: The above results show GPT's ability to detect and describe the features of commonly available images.

APPENDIX B

Legend: Text highlighted in cyan represents the prompt
The text in red is commentary on the results.

Evaluation of Gemini 3 pro (November 21, 2025)

Case 1:

Prompt: Identify all visible elements and defects in this image using MDOT/FHWA terminology. Consider only the elements and defects that are clearly visible. If unclear, label as 'Not clearly visible.' Provide a list of visible elements and defects.

Based on the visual evidence in the image, here are the identified elements and defects using standard MDOT (Michigan Department of Transportation) and FHWA (Federal Highway Administration) bridge inspection terminology:

Visible Elements

- **Reinforced Concrete Bridge Railing (Parapet/Curb):** The primary structural element shown is the concrete curb or parapet section of a bridge barrier.
- **Metal Railing Post:** A portion of a gray metal post (likely steel or aluminum) anchored into the concrete parapet is visible on the left.
- **Deck/Wearing Surface:** The <u>*asphalt*</u> or concrete surface adjacent to the curb, representing the roadway or shoulder of the bridge deck.
- **Inspection Markings:** White chalk or paint markings delineating the defective area, including dimensions *(200×950)*, indicating a recent inspection or preparation for repair.

Comment: Except for including the $ sign with the dimensions and hallucinating about the deck/earing surface due to the lack of clarity in the image, the rest of the information is accurate. Even though the visible element is listed as Reinforced Concrete Bridge Railing (Parapet/Curb), the description notes that it may be a concrete curb. This level of performance indicates that Gemini Pro outperforms ChatGPT-5.1.

Visible Defects

- **Spall (Spalling):** There is a large, deep spall on the vertical face and corner of the concrete parapet. This is a depression resulting from the detachment of a significant portion of the surface concrete.
- **Exposed Aggregate:** The spall is deep enough to reveal the coarse aggregate within the concrete matrix.
- **Cracking:**
  - There is visible cracking on the top surface of the parapet, specifically to the left of the spall near the metal post.
  - Minor cracking is visible surrounding the spalled area.

  Comment: It identified only one crack, as shown in the labeled image below. Even though it states that minor cracking is visible around the spalled area, the labeled figure does not show it. However, Gemini Pro is better than ChatGPT-5.1 for detecting cracks.
- **Delamination (Implied):** The white rectangular outline drawn by the inspector typically indicates a "sounded" area of delamination (hollow concrete) that extends beyond the currently visible spall, marking the boundaries for concrete removal.

Comment: The above information is accurate. Gemini Pro is better than ChatGPT-5.1.

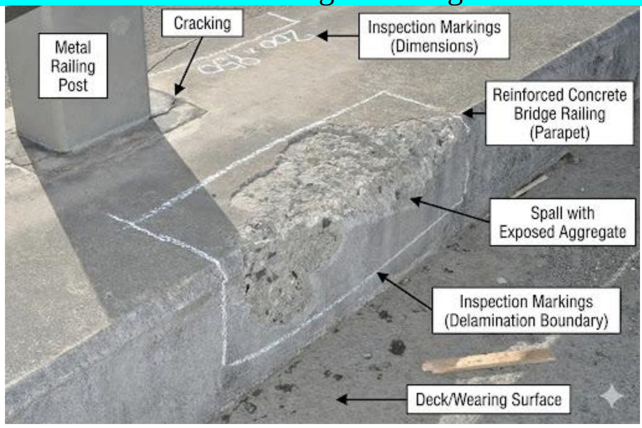
Not Clearly Visible

- **Exposed Reinforcing Steel (Rebar):** While the spall is deep, distinct corroded steel bars are not definitively visible in the image resolution; only aggregate is clearly confirmed.
- **Efflorescence:** No clear signs of white mineral deposits are visible in this specific angle.

Comment: This is a hallucination, but it provided a reasonable answer for the prompt.

Would you like me to draft a specific repair recommendation or quantity calculation based on the dimensions written in the image?

Prompt: provide a labeled version of the image marking each visible element and defect.



Comment: The labeled image is mostly accurate. Even though it states that minor cracking is visible around the spalled area, the labeled figure does not show it. Further, it failed to identify the wide crack at the left edge of the curb. However, Gemini Pro is better than ChatGPT-5.1 at detecting damage and cracks and providing labels.

Case 2:

Prompt: Identify all visible elements and defects in this image using MDOT/FHWA terminology. Consider only the elements and defects that are clearly visible. If unclear, label as 'Not clearly visible.' Provide a list of visible elements and defects.

Based on the visual evidence in the image, which depicts a bridge pier and superstructure viewed from the roadway below, here are the identified elements and defects using standard MDOT (Michigan Department of Transportation) and FHWA (Federal Highway Administration) bridge inspection terminology.

Visible Elements

Substructure:

- **Reinforced Concrete Pier Cap:** The horizontal member resting on the columns that supports the superstructure. It bears the red inspection marking "P-1W-E".
- **Reinforced Concrete Columns:** Three circular vertical supports transferring the load from the cap to the foundation.
- **Pier Protection Wall (Crash Wall):** The solid concrete wall connecting the base of the columns, designed to protect the pier from vehicle impact.
- **Slope Paving:** The concrete block surfacing on the embankment behind the pier, used for erosion control.

Superstructure:

- **Prestressed Concrete I-Beams (Stringers):** The longitudinal beams supporting the deck, marked with "1S", "3S", and "5S".

  Comment: This failed to identify the other two beams (see the labeled figure shown later in this document).
- **Concrete Diaphragms:** The lateral concrete braces connecting the beams to distribute loads and provide stability.
- **Bearings (Elastomeric Pads):** Visible as dark rectangular pads between the bottom of the beams and the top of the pier cap.

Roadway/Safety Features:

- **W-Beam Guardrail:** A metal safety rail attached to the face of the crash wall on the right side.
- **Roadway Pavement:** The asphalt surface in the foreground.

** markings:**

- **Inspection/Inventory Markings:** Red spray paint markings used for identifying elements (e.g., "P-1W-E" likely indicating Pier 1 West, East Face; "3S" indicating Stringer 3).

Visible Defects / Conditions

The structure appears to be in **Very Good to Excellent Condition**, with almost no structural defects visible.

- **Minor Surface Abrasion/Scuff:** There is a very small, dark scuff mark on the rightmost column (mid-height), likely from minor impact or construction activity.
- **Formwork Lines (Not a defect):** Vertical lines visible on the columns are seams from the casting forms, not structural cracks.

  Comment: This is not correct. Formwork lines are not visible in the image.
- **Inspection Markings:** While essential for inspection, the red paint is technically a surface alteration (though not a structural defect).

Not Clearly Visible

- **Deck Underside:** The bottom surface of the bridge deck is obscured by the beams and diaphragms.

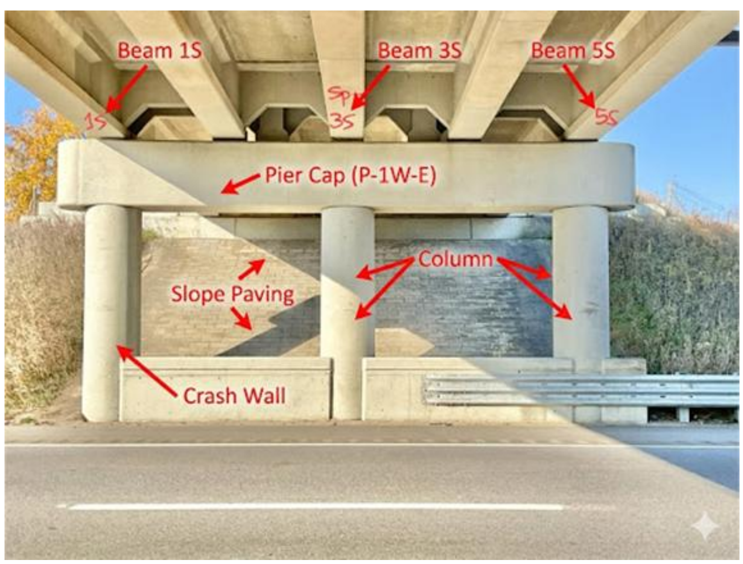
  Comment: This is not correct. The deck underside is clearly visible.
- **Foundation/Footings:** Buried below the crash wall and roadway level.
- **Bearing Conditions:** While the pads are visible, their specific condition (bulging, tearing) cannot be assessed from this distance.
- **Top of Deck:** Not visible from this angle.

  Comment: There are certainly hallucinations, but not as severe as what we observed from ChatGPT-5.1.

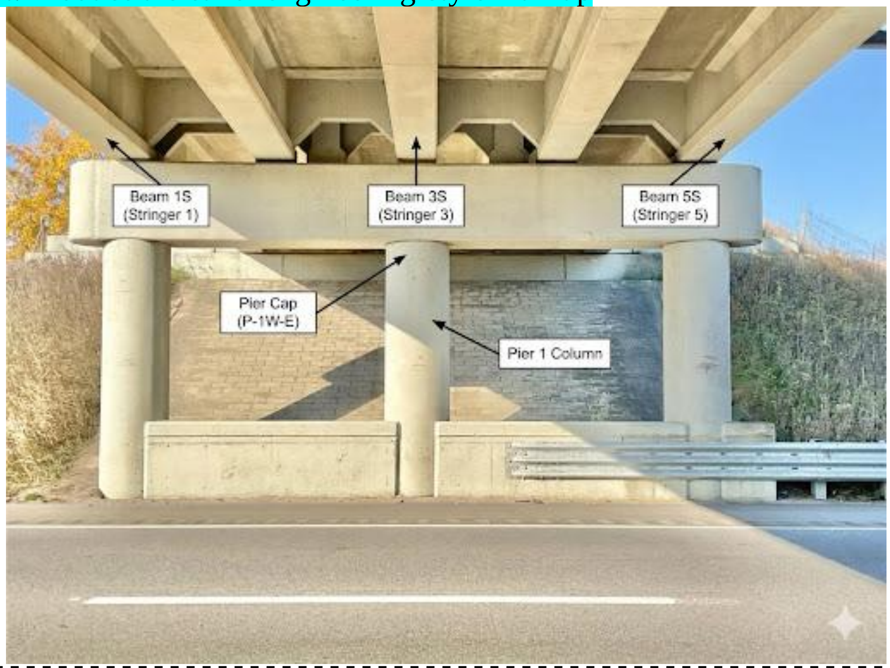
****

Would you like me to generate a labeled version of this image pointing out the specific beam numbering and structural components?

Prompt: yes

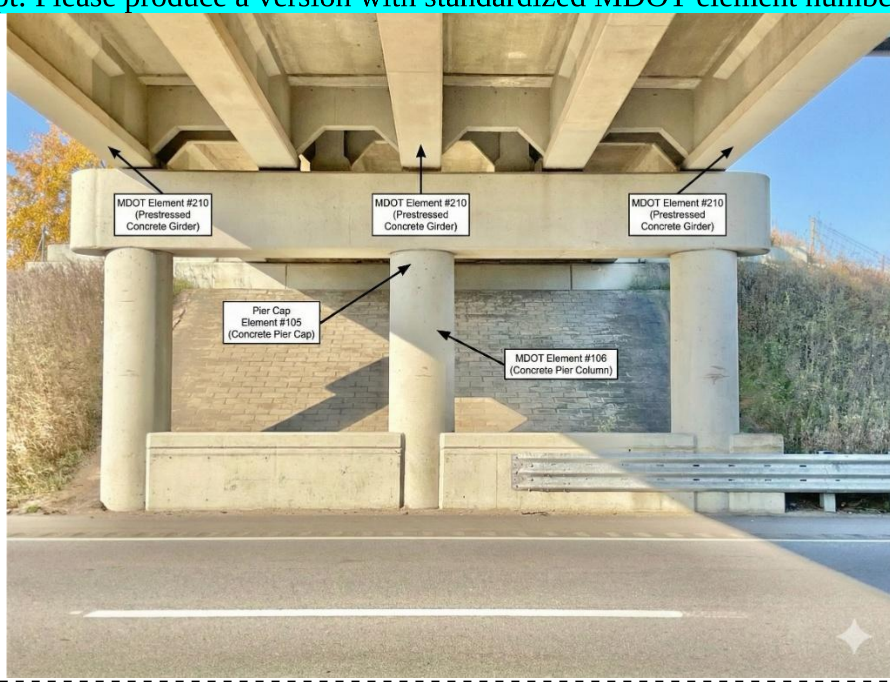


Prompt: Produce a cleaner engineering-style markup



Comment: As shown in the two figures above, the results are not convincing. Only the beams with labels were identified. Two interior beams were not identified. Label positions are incorrect. Some of the components listed in the results are not labeled.

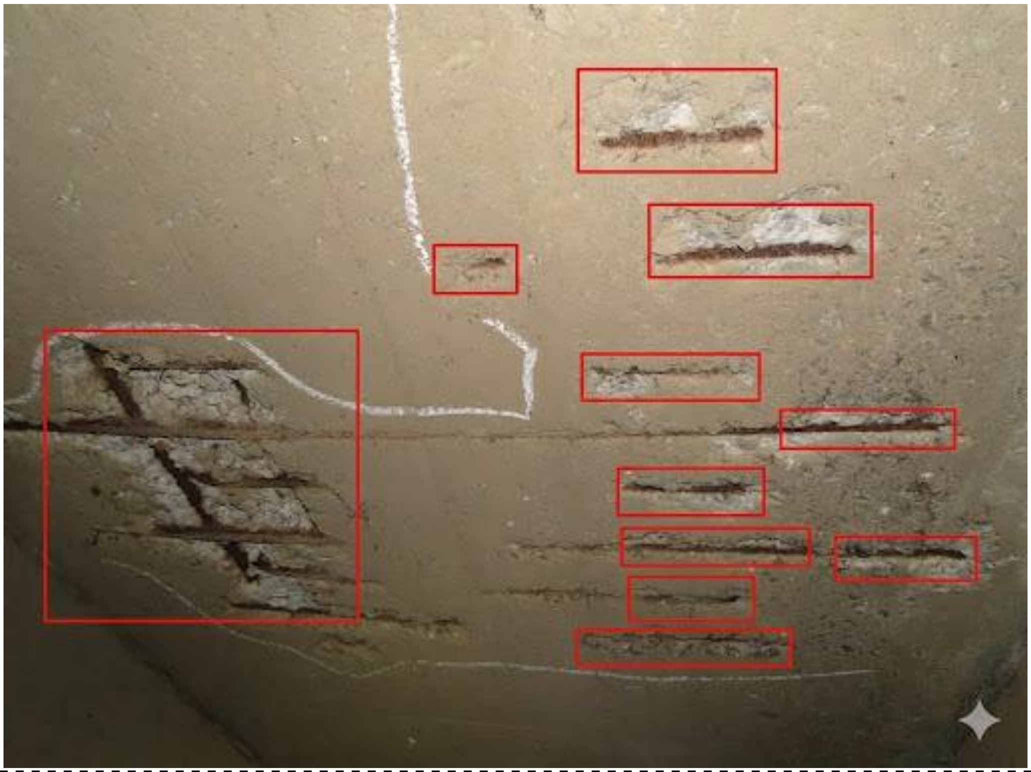
Prompt: Please produce a version with standardized MDOT element numbers



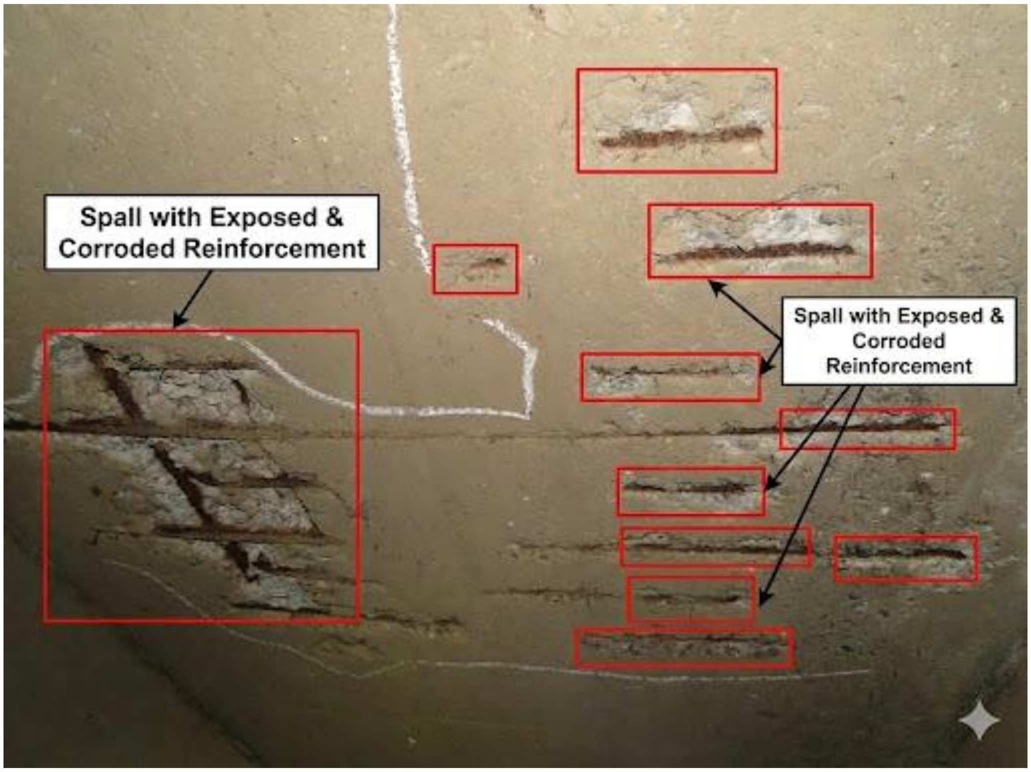
Comment: As shown above, incorrect element numbers are provided. Several of the components listed at the beginning as visible are not labelled in this figure.

Case 3:

Prompt: Identify all visible defects in this image using MDOT/FHWA terminology. Consider only the clearly visible defects. If unclear, label as 'Not clearly visible.' Return the same image with red bounding boxes drawn around each recognized defect. Provide a list of visible elements and defects.



Prompt: provide clear labels to the recognized defects



Comment: Gemini Pro seems much better than ChatGPT-5.1 for damage detection and providing bounding boxes with labels.

Case 4:

Prompt: Identify and summarize the content in this image.

Based on the image provided, here is a summary of the content and its engineering context:

Image Content Summary

- **Sign Type:** Regulatory Weight Limit Sign (Load Posting).
- **Text Header:** "WEIGHT LIMIT"
- **Restrictions Listed:**
    1. **Single Unit Truck (Box Truck symbol):** 42 T (Tons)
    2. **Combination Truck (Semi-trailer symbol):** 63 T (Tons)
    3. **Double Combination Truck (*<u>Double-bottom symbol</u>*):** 77 T (Tons)

Visual Elements

- **Mounting:** The sign is mounted on a metal U-channel post.
- **Condition:** The sign face appears clean and reflective.
- **Surroundings:** Dense vegetation/trees in the background.
- **Additional Marker:** There appears to be a faded orange flag attached to the top of the post, which is typically used to alert drivers to a **newly installed** or **recently changed** sign (e.g., a recent down-posting of a bridge).

Engineering Context (MDOT/FHWA)

This sign is used to post load restrictions for a bridge that cannot safely carry the full legal loads for all vehicle configurations. The three silhouettes correspond to standard Michigan legal vehicle classes (Single Unit, Combination, and *<u>Double Bottom</u>*).

Would you like me to explain how these specific tonnage limits are calculated based on operating ratings?

Comment: Gemini Pro and ChatGPT-5.1 performances are similar when analyzing load posting signs. There are a couple of words that do not match the correct technical terms. They are underlined with italics for reference. These types of mistakes can be easily fixed by using Agentic AI with a Retrieval-Augmented Generation (RAG) application.

Evaluation of Thinking with 3 Pro (November 24, 2025)

Case 1:

Prompt: Identify all visible elements and defects in this image using MDOT/FHWA terminology. Consider only the elements and defects that are clearly visible. If unclear, label as 'Not clearly visible.' Provide a list of visible elements and defects.

Based on the provided image, here is the identification of visible elements and defects using standard MDOT (Michigan Department of Transportation) and FHWA (Federal Highway Administration) bridge inspection terminology.

Visible Bridge Elements

These elements are categorized by their structural function.

Substructure (Supporting Elements)

- **Pier Cap (Reinforced Concrete):** The large horizontal beam resting on the columns that supports the superstructure. Identified by the red marking "P-1W-E" (likely denoting Pier 1 West, East Face).
- **Pier Columns (Reinforced Concrete):** The three circular vertical supports transferring load from the cap to the foundation.
- **Crash Wall / Pier Protection:** The solid concrete wall connecting the bases of the columns. This is designed to protect the pier columns from vehicle impact.
- **Elastomeric Bearing Pads:** The dark rectangular pads visible between the bottom of the beams and the top of the pier cap.

Superstructure (Span Elements)

- **Prestressed Concrete I-Beams (Girders):** The main horizontal support members running perpendicular to the pier. These appear to be AASHTO or PCI type I-beams.
- **Concrete Diaphragms:** The vertical concrete braces located between the beams (visible near the bearing points) to provide lateral stability.
- **Deck Underside (Soffit):** The bottom surface of the concrete roadway deck, visible between the top flanges of the I-beams.

Site & Safety Features

- **Slope Protection:** The concrete block paving (often called block mat or cellular concrete mat) covering the embankment under the bridge to prevent erosion.
- **Guardrail (W-Beam):** The galvanized steel safety barrier in the foreground, attached to the crash wall end.
- **Inspection Markings:** Red spray-painted identifiers for the pier ("P-1W-E") and beam locations ("1S", "3S", "5S", "Sp-2W").

Comment: Results are better than what was received from Gemini 3 Pro.

Visible Defects / Conditions

The structure appears to be in generally **Good** to **Satisfactory** condition based on this visual overview. Significant structural distress is not apparent at this resolution, but the following conditions are visible:

Concrete Elements (Pier Cap, Columns, Crash Wall)

- **Inspection Markings (Aesthetic):** The red spray paint, while functional for inspection, is technically a surface coating/graffiti, though not a structural defect.
- **Minor Discoloration/Staining:** There are very faint vertical streaks on the columns and crash wall, likely due to minor water runoff or weathering.
- **Formwork Lines:** Vertical lines are visible on the pier cap (especially near the right column). These appear to be casting lines from the construction forms rather than structural cracks, but close-up sounding would be required to confirm.

Comment: Interesting results. Gemini 3 Pro stated, "Vertical lines visible on the columns are seams from the casting forms, not structural cracks." In contrast, the Thinking with 3 Pro model mentions the vertical lines on the pier cap. These models are providing contradicting results.

Slope Protection

- **Vegetation Encroachment:** There is dry vegetation growing over the edges of the block slope paving on the far left and right sides.
- **Minor Settlement/Unevenness:** The transition between the slope paving and the pier columns appears tight, but minor settling is common in these areas (not clearly definable as a defect here).

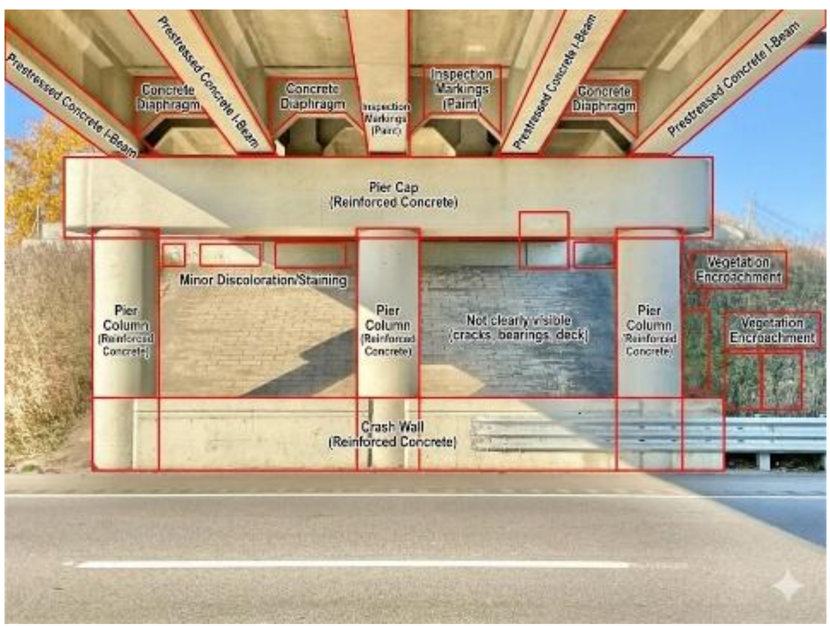
Not Clearly Visible (Limitations)

- **Hairline Cracking:** Standard shrinkage cracks (<0.012 inches) cannot be seen at this distance.
- **Bearing Condition:** The physical condition of the elastomeric pads (bulging, tearing) is not visible from this angle.
- **Deck Top Surface:** The roadway surface condition is obscured.

Comment: Slope protection related information and the list provided under Not Clearly Visible are hallucinations.

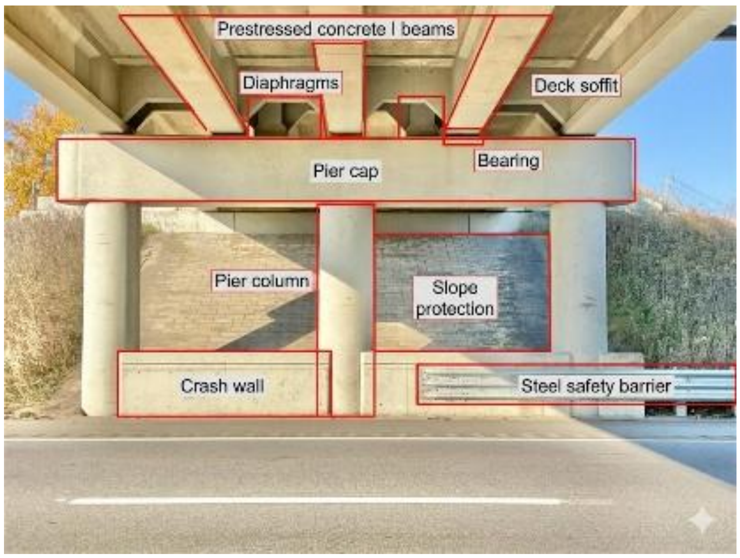
Would you like me to explain the likely decoding of the specific red inspection markings (e.g., what "Sp-2W" specifically indicates in inspection reporting)?

Prompt: Identify all visible elements and defects in this image using MDOT/FHWA terminology. Consider only the elements and defects that are clearly visible. If unclear, label as 'Not clearly visible.' Return the same image with red bounding boxes drawn around each recognized element and/or defect. Provide a list of visible elements and defects.



Prompt: Identify the following components and provide labels with bounding boxes:

Prestressed concrete I beams
Diaphragms
Pier cap
Pier column
Crash wall
Steel safety barrier
Deck soffit
Abutment
Bearing
Slope protection
Wingwall



Prompt: Identify the given element separately. Provide bounding boxes for each element. If not identified, provide reasons.

Prestressed concrete I beams
Diaphragms
Pier cap
Pier column
Crash wall
Steel safety barrier
Deck soffit
Abutment
Bearing
Slope protection
Wingwall

Based on the image provided, here is the identification of the specific components you requested.

Identified Elements

- **Prestressed concrete I-beams: Identified.** These are the five longitudinal girders supporting the deck, labeled 1S, 3S, 5S, etc.
- **Diaphragms: Identified.** These are the vertical concrete cross-members located between the I-beams, resting on the pier cap. They provide lateral stability.
- **Pier cap: Identified.** The large horizontal concrete beam (labeled P-1W-E) that rests on the columns and supports the I-beams.
- **Pier column: Identified.** The three vertical circular concrete supports transferring loads from the cap to the foundation.
- **Crash wall: Identified.** The solid concrete wall connecting the lower portions of the columns. This structural element protects the pier from vehicle impact.
- **Steel safety barrier: Identified.** The galvanized steel W-beam guardrail seen in the foreground, attached to the end of the crash wall.
- **Deck soffit: Identified.** The underside surface of the concrete bridge deck, visible between the top flanges of the I-beams.
- **Abutment: Identified.** Visible in the background at the top of the slope. This is the retaining wall/support structure at the end of the bridge span.
- **Bearing: Identified.** The rectangular elastomeric pads positioned between the bottom of the I-beams and the top of the pier cap.
- **Slope protection: Identified.** The concrete block paving (block mat) covering the earthen embankment in the background to prevent erosion.

Not Clearly Visible

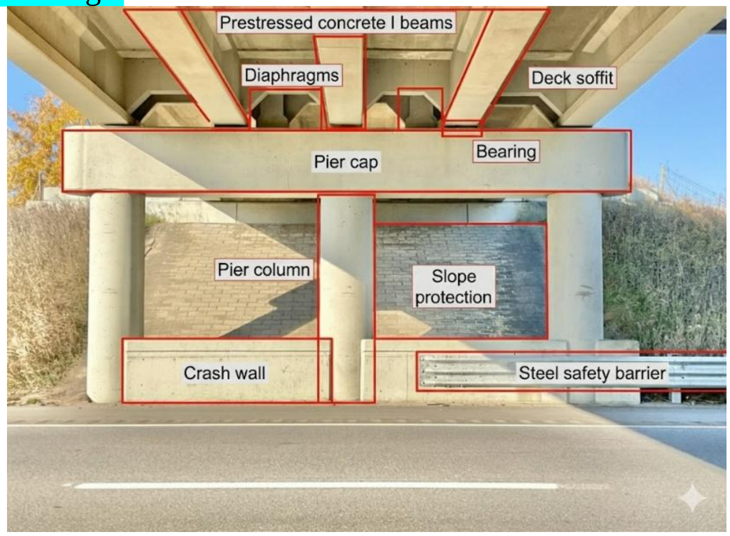
- **Wingwall: Not clearly visible.** Wingwalls are the retaining walls extending from the abutment to hold back the side earth. While the abutment face is visible in the background, the specific angled wingwalls are obscured by the perspective and the slope/vegetation at the far edges of the frame.

Comment: Even though the bounding boxes and labels are not accurate, the descriptions provided for the listed elements are correct. Furthermore, it produces output in JSON format. This can be used to verify the accuracy of the interpretations.

JSON


```
[
  {"box_2d": [0, 65, 223, 321], "label": "Prestressed concrete I beam"},
  {"box_2d": [0, 362, 219, 492], "label": "Prestressed concrete I beam"},
  {"box_2d": [0, 592, 220, 760], "label": "Prestressed concrete I beam"},
  {"box_2d": [0, 772, 218, 992], "label": "Prestressed concrete I beam"},
  {"box_2d": [169, 332, 240, 427], "label": "Diaphragm"},
  {"box_2d": [171, 670, 244, 763], "label": "Diaphragm"},
  {"box_2d": [169, 507, 244, 597], "label": "Diaphragm"},
  {"box_2d": [234, 72, 374, 857], "label": "Pier cap"},
```

Prompt: We don't need the JSON file. Provide bounding boxes for the identified elements on the given image.



Comment: The descriptions provided in text format are promising since they contain a few simple mistakes. However, the labels and bounding boxes are not accurate; hence, these tools need to be improved using Agentic AI.

APPENDIX C

Requirements for A High Accuracy Automated Image Attribution Enhancement Tool

1. Purpose and Scope

The [State] Department of Transportation (hereinafter "the Department") is seeking proposals from qualified vendors to provide a Computer Vision Software Platform capable of performing automated image and/or video analysis for tasks such as object detection, classification, segmentation, and feature extraction.

The system shall recognize major bridge components, traffic control devices, and load posting signs. The system shall extract information from load posting signs, speed limit signs, and bridge clearance signs, and shall detect and identify (tag) structural defects (damages). Through these functions, the system shall support condition documentation and assessment workflows.

The Department conducts regular bridge inspections in accordance with the Michigan Bridge Element Inspection Manual (MiBEIM) and the Michigan Structure Inspection Manual (MiSIM). The increasing volume of image and video data from drone and ground-based inspections has created a need for advanced software capable of automated analysis. This project seeks to implement an intelligent computer vision platform to enhance efficiency and consistency in bridge inspection and condition documentation.

2. Functional Requirements

The software must:

- ☑ Automatically identify and label (tag) bridge components (e.g., deck, girders, bearings, piers, abutments, joints) using MDOT standard terminology.
- ☑ Automatically identify and label (tag) specific features and defects (e.g., cracks, corrosion, spalling, etc.) using MDOT standard terminology.
- ☑ Automatically identify and label (tag) traffic control devices, load posting signs, and speed limit signs, and extract their content.
- ☑ Automatically identify and label (tag) duplicate or near-duplicate images.
- ☑ Support core computer vision tasks:
    - Object detection and recognition (static and/or dynamic).
    - Image segmentation (semantic or instance-based) (for components and defects).
- ☑ Allow training and/or fine-tuning of machine learning models using local or custom datasets to improve model weights.
- ☑ Handle multiple input formats: still images, and (as a future extension) video files.
- ☑ Provide tools for visualization (segmentation overlays, labels).
- ☑ Enable result export to common formats (CSV, JSON, etc.) to support use in asset management and GIS systems.
- ☑ Support batch (offline) or on-demand processing of imagery and video from multiple sources, including UAVs and mobile systems.
- ☑ Store metadata associated with inspection imagery (location, time, bridge ID, etc.).
- ☑ Provide automated and manual review capabilities through a graphical user interface.

☑ Include API access for advanced users (negotiable).

3. Minimum Technical Requirements

The proposed system must meet or exceed the following minimum requirements:

Requirement Category	Minimum Requirement
Architecture	Deployable as a cloud-based (e.g., AWS) or on-premise (desktop application) environment; supports GPU acceleration. Compatible with Windows, Linux, and/or containerized environments (e.g., Docker). (Kubernetes is not required.)
Model Performance	Machine learning models capable of identifying bridge components with ≥ 80% precision and recall under standard imaging conditions.
Model and Algorithm Transparency	Allow inspection of model metadata, accuracy metrics, and version control. The system should provide explainability features for supported models. For instance, gradient-based class activation maps (Grad-CAM or equivalent). Must allow retraining or replacing models without complete reinstallation.
Data Handling	Supports datasets with more than 5,000 images per project; maintains metadata and version tracking for models.
Integration	Provides data export functionality or API (REST/gRPC) for integration with existing DOT databases, GIS systems, etc.
Performance	Capable of processing high-resolution imagery (e.g., 4K). Based on modern model architectures (e.g., YOLO, Segformer), enabling high-speed processing of approx. 2-3 seconds per image on standard GPU hardware.
Security	Complies with NIST SP 800-53 or equivalent; includes encryption in transit and at rest, role-based access control, and audit logs.
Usability	Intuitive graphical interface; supports user annotations, quality review, and web browser access.
Accessibility	WCAG 2.1 Level AA compliant interface.
Documentation	Vendor must provide user manuals, API documentation, and training materials.

4. Vendor Qualifications

- Minimum of **_________** years of prior deployments of computer vision or AI-based inspection systems in civil infrastructure or transportation domains.
- Demonstrated experience integrating software with GIS and/or asset management platforms.
- Ability to provide ongoing technical support, updates, and maintenance.

5. Security and Data Ownership

- All imagery, video, and analytical outputs shall remain the sole property of the Department.
- The vendor shall not store or reuse Department data without express written consent.
- All data transmissions must be encrypted; data hosted externally must comply with state and federal data protection regulations.
- Any cloud hosting environment must meet FedRAMP Moderate or equivalent security certification.

6. Deliverables

The successful vendor shall provide:

- Fully functional computer vision software system
- System integration documentation and API specifications
- Training for Department personnel (minimum of two sessions)
- User manuals and administrator documentation
- Product roadmap, AI roadmap, and scalability information
- Technical support and maintenance agreement
- Accuracy validation report (including precision, recall, and F1-score for bridge component and defect detection).

  (i) Training Performance Report
    - A comprehensive training performance matrix report that includes all relevant performance metrics (e.g., accuracy, precision, recall, loss curves, etc.), presented clearly in both tabular and graphical formats.
    - The report should demonstrate the model’s learning progress and validation performance.

  (ii) Prediction Results and Accuracy Evaluation
    - The vendor shall process a set of bridge inspection images provided by MDOT and deliver:
    - Dataset Variability: MDOT will provide three distinct datasets representing varying lighting conditions, structural configurations, and image quality levels.
    - The model's performance shall be evaluated across all three datasets.
    - Prediction Outputs: Annotated images with the predicted bridge components clearly marked, along with corresponding text files (e.g., CSV, JSON, or TXT) containing the coordinates and labels of each identified component.
    - Accuracy Assessment: MDOT will compare these outputs against existing annotations to calculate prediction accuracy.
    - Minimum Performance Threshold: The vendor's model must achieve an overall accuracy of at least 80% across these datasets to meet MDOT's quality expectations.

7. Implementation Plan

The proposal shall include a detailed implementation plan, including:

- Project timeline with key milestones.
- Pilot testing phase for performance validation.
- User acceptance testing (UAT) plan.
- Data migration or integration strategy.
- Training and knowledge transfer schedule.

APPENDIX D

Recommended Image Storage Options for Computer Vision Programs Developed for Bridge Component and Defect Detection

The optimal file format differs for each process, such as on-site capture, analysis by AI models, and integration with GIS systems. For AI model training or large-scale data processing, image formats, such as HDF5, NPY, and Parquet, are required. However, for applications limited to component/defect segmentation, sign recognition, character recognition, and GIS integration, practically feasible formats are used. This document describes the **primary file formats** suitable for the application discussed in the report and provides clarifications for their selection.

Primary File Formats for Application Support

Input (Read): JPEG (.jpg)

- **Use Case:** Standard field inspection images captured by drones or ground-based cameras.
- **Rationale:** Versatility, small file size, and fast processing speeds make it suitable for standard analysis.
- **Concern:** Lossy compression can introduce artifacts, though standard quality is generally acceptable.

Output (Write): JPEG (.jpg)

- **Concern:** Due to its lossy compression, JPEG is *not suitable* for analysis outputs. Compression can blur mask edges and cause a loss of precise area information. *However, if the client strongly requests JPEG output, it is technically possible to provide it, provided this reduction in quality is understood and accepted.*

Input (Read): PNG (.png)

- **Use Case: Reading existing segmentation masks, annotations, or other lossless images** provided by an agency or third-party systems.
- **Rationale:** As a lossless format, it ensures that high-fidelity data (like existing masks) is read without any degradation, which is crucial for analysis or comparison.
- **Concern:** File sizes are larger than JPEG.

Output (Write): PNG (.png)

- **Use Case:** Segmentation masks generated by the AI model or annotation overlays (detection results).
- **Rationale:**
  - **Lossless Compression:** This is a mandatory requirement for accurately saving pixel-level segmentation masks (e.g., crack areas).
  - **Transparency Support:** Ideal for creating overlay images to be superimposed on the original image.

Input (Read): GeoTIFF (.tif)

- **Use Case:** Inspection images provided by an agency with embedded location data (geographic coordinates).
- **Rationale:** Essential for GIS integration and asset management. The underlying TIFF format is high-fidelity.
- **Concern:** File sizes tend to be larger than JPEG.

Other Considerations

- **Additional Input Format Support:**
  - While the primary formats (JPEG, GeoTIFF, PNG) are recommended, it is technically possible to support reading other basic uncompressed raster formats, such as **BMP (.bmp)**, if required. However, active use of BMP is not recommended as it results in very large files.
- **Video Files (MP4, AVI, etc.):**
  - Based on the defined scope (processing every few frames), it is recommended that the application process **still images (JPEG sequences)** extracted from videos, rather than reading video files directly.
- **AI Model Training Data (HDF5, NPY, etc.):**
  - These are formats used during the AI model development and training phase. The deliverable application does not need to read or write these formats.

Conclusions

The bridge inspection application demonstrated during this project supports the reading and writing of the following file formats:

- **Read (Input):**
  1. **JPEG:** For general inspection images.
  2. **GeoTIFF:** For GIS integration and location-aware images.
  3. **PNG:** For reading existing masks or lossless data.
- **Write (Output):**
  1. **PNG:** (Recommended) For segmentation masks and analysis results.
- **(Optional):**
  1. **BMP:** Support for reading legacy formats as needed.
  2. **JPEG (Output):** Possible upon strong client request, acknowledging quality trade-offs.
  3. **WEBP:** For web dashboards or cloud storage.