RESEARCH

UDOT

Keeping Utah Moving

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

Report No. UT-26.02

ENHANCING ACCESSIBILITY FOR INDIVIDUALS WITH LIMITED MOBILITY: LEVERAGING AI AND CYCLING DATA FOR AN INCLUSIVE ACTIVE TRANSPORTATION NETWORK

Prepared For:

Utah Department of Transportation
Research & Innovation Division

Final Report
January 2026

DISCLAIMER

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

ACKNOWLEDGMENTS

This work was funded by the Utah Department of Transportation (UDOT) through the Research & Innovation Division. We thank staff across UDOT departments for guidance and helpful feedback. We are particularly grateful to Jared Bowling, Neda Kiani, and Kevin Nichol. We also appreciate the engagement of regional partners, including the Wasatch Front Regional Council, Mountainland Association of Governments, and Cache County. Any opinions, findings, and conclusions or recommendations expressed in this report are those of the authors and do not necessarily reflect the views of UDOT.

TECHNICAL REPORT ABSTRACT

1. Report No. UT-26.02	2. Government Accession No. N/A	3. Recipient's Catalog No. N/A
4. Title and Subtitle Enhancing Accessibility for Individuals With Limited Mobility: Leveraging AI and Cycling Data for an Inclusive Active Transportation Network		5. Report Date January 2026 6. Performing Organization Code N/A
7. Author(s) Madhu Mausam Thapa, Sanjay Luitel, Nikola Markovic, Abbas Rashidi, Robert Chamberlin		8. Performing Organization Report No. N/A
9. Performing Organization Name and Address The University of Utah Department of Civil and Environmental Engineering 201 Presidents Circle Salt Lake City, Utah 84112		10. Work Unit No. 5H094 80H 11. Contract or Grant No. 25-8243
12. Sponsoring Agency Name and Address Utah Department of Transportation 4501 South 2700 West P.O. Box 148410 Salt Lake City, UT 84114-8410		13. Type of Report & Period Covered Final Sept 2024 to Oct 2025 14. Sponsoring Agency Code UT24.404
15. Supplementary Notes Prepared in cooperation with the Utah Department of Transportation		

16. Abstract

This study introduces a low-cost, scalable framework for monitoring and assessing the condition of paved trail infrastructure to enhance accessibility for all users, particularly those with limited mobility. The research utilizes a custom-built "Data Bike" with GPS and accelerometers embedded with high-resolution cameras to collect multimodal data along Utah's trail networks. A simple and intuitive data processing pipeline was developed that uses accelerometer-derived "jerk" events to automatically identify potential surface anomalies, triggering the extraction of corresponding image frames. A deep-learning object detection model (YOLOv8) was fine-tuned on a custom dataset and augmented with public imagery to detect and classify ten surface defects and obstructions, including various cracks, potholes, and vegetation obstacles. The model achieved promising results, with detections geolocated to enable spatial analysis and hotspot identification. This data-driven approach provides transportation agencies an objective and efficient tool to move from reactive to proactive maintenance, allowing for better resource allocation and systematic improvements to trail safety and accessibility. The project lays the groundwork for an integrated asset management system, including an interactive dashboard for visualizing trail conditions and prioritizing repairs.

17. Key Words Active Transportation, Accessibility, Trail Maintenance, Machine Learning, Computer Vision, Defect Detection, Data Bike, YOLOv8		18. Distribution Statement Not restricted. Available through: UDOT Research Division 4501 South 2700 West P.O. Box 148410 Salt Lake City, UT 84114-8410 www.udot.utah.gov/go/research		23. Registrant's Seal N/A
19. Security Classification (of this report) Unclassified	20. Security Classification (of this page) Unclassified	21. No. of Pages 100	22. Price N/A	

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

AI	Artificial Intelligence
ARAN	Automatic Road Analyzer
CCI	Cycling Comfort Index
COCO	Common Objects in Context
CNN	Convolutional Neural Network
CSV	Comma-Separated Values
DCI	Dynamic Comfort Index
FCN	Fully Convolutional Network
GHSA	Governors Highway Safety Association
GIS	Geographic Information System
GRDD	Global Road Damage Detection
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPR	Ground-Penetrating Radar
IRI	International Roughness Index
IPB	Instrumented Probe Bicycle
LCMS	Laser Cracking Measurement System
LiDAR	Light Detection and Ranging
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
mAP50	Mean Average Precision at IoU threshold 0.50
mAP@0.5:0.95	Mean Average Precision averaged over IoU thresholds from 0.50 to 0.95
NMS	Non-Maximum Suppression
PVDF	Polyvinylidene Fluoride
RF	Random Forest

RDD	Road Damage Detection
SUP	Shared Use Paths
SVM	Support Vector Machine
SVR	Support Vector Regression
UDOT	Utah Department of Transportation
UTN	Utah Trail Network
YOLOv5	You Only Look Once version 5
YOLOv8	You Only Look Once version 8

EXECUTIVE SUMMARY

Maintaining safe and accessible active transportation networks is a critical goal for transportation agencies, yet traditional trail inspection methods are often labor intensive, subjective, and costly. This project addresses these challenges by developing a low-cost automated system for detecting and mapping trail surface defects to support proactive asset management and enhance accessibility for all users, including individuals with disabilities.

The core of this research is the "Data Bike," a custom-built bicycle equipped with a high-resolution camera, Global Positioning System (GPS), and an accelerometer. This platform was deployed across 34 miles of diverse trail segments in Salt Lake County, Utah, to collect video footage embedded with GPS and accelerometer data of real-world trail conditions.

A key contribution of this project is a data processing pipeline that uses accelerometer data to detect "jerk" events, which are sudden vibrations indicating potential surface anomalies and other surface conditions. These events automatically trigger the extraction of corresponding image frames, focusing the analysis on areas of interest and significantly improving efficiency and reducing computational cost. These images were used to build a comprehensive dataset, which was then used to train a You Only Look Once (YOLO) deep learning model supplemented with the Global Road Damage Detection (GRDD) dataset. The GRDD dataset is a publicly available benchmark designed to advance road-surface-condition detection research by providing real-world, annotated images for computer vision and deep learning applications. The model was trained to automatically identify and classify ten distinct categories of surface anomalies and conditions, such as potholes, various types of cracks, tactile pavements, and vegetation obstacles.

The system successfully identifies and geolocates trail defects, achieving an overall mean Average Precision (mAP50) of 0.621 and mAP@0.5:0.95 of 0.345 in its test phase. These results demonstrate the feasibility of using this technology for large-scale network monitoring. The geolocated dataset provides the foundation for creating hotspot maps and an interactive dashboard, equipping the Utah Department of Transportation (UDOT) with a data-driven tool to prioritize maintenance and improve the safety, usability, and overall condition of trails across Utah.

Beyond the technical demonstration, this project also highlights strategic implications for UDOT. The Data Bike system creates an opportunity to move from complaint-driven and sporadic inspections toward a proactive, predictive asset management framework. By integrating the outputs into enterprise Geographic Information System (GIS) and asset management systems, UDOT can make maintenance decisions based on measurable need, ensuring that resources are allocated efficiently to improve the condition of trails across Utah. A phased three-year action plan is proposed beginning with pilot deployment and staff training, followed by statewide data collection and database development, and culminating in predictive modeling and full integration into UDOT's planning and budgeting processes. This approach will allow UDOT to establish a centralized, living inventory of trail conditions, improve transparency in decision-making, and directly advance statewide goals for safety, usability, and sustainability in active transportation.

1.0 INTRODUCTION

1.1 Background

Trail infrastructure plays a pivotal role in enabling safe and convenient mobility across urban and suburban environments (Lindsey et al., 2008). Trails are more than just pathways; they are public spaces that facilitate social inclusion, provide access to transit systems, and contribute to environmental sustainability through active transportation. Well-maintained trails play a crucial role in facilitating routine physical activity and daily mobility across all age groups and ability levels. However, the usability of trails is frequently compromised by surface degradation such as cracks, heaving, settlement, vegetation obstacles, and foreign obstructions. These issues not only reduce walkability and visual appeal but also present serious safety hazards (Subedi et al., 2025).

This research examines how deficiencies in trail condition and design can limit the usability of active transportation networks, with implications for maintenance and asset management (Boschmann & Brady, 2013). The core of this issue lies in understanding and improving key factors such as connectivity of trails, which might be fragmented due to cracks, potholes, debris and obstacles, which significantly impact the usability of the facilities for everyone, particularly those with mobility challenges like the elderly and people with disabilities. Ensuring that facilities are accessible and safe for all users is not just a matter of convenience, but a fundamental question of safe access to outdoor spaces for all community members, which are vital for physical health, mental well-being, and healthy social relationships.

To tackle this, the project will utilize a camera-equipped Data Bike to systematically collect detailed data on these critical factors like surface defects like potholes, cracks, and other non-surface obstacles like vegetation obstacles and poles. Following data collection, the project will leverage the power of artificial intelligence (AI), specifically through computer vision techniques, to analyze the collected footage. This high-tech approach enables a comprehensive and nuanced analysis of facility conditions, highlighting the project's innovative blend of technology and social purpose in its effort to create a more inclusive active transportation network.

According to the Governors Highway Safety Association (GHSA), pedestrian fatalities in the United States surged by 77% between 2010 and 2021, with many of these incidents occurring

in areas lacking adequate pedestrian infrastructure or where trails were poorly maintained. Manual inspection remains the predominant strategy for identifying trail defects, but it is inherently limited by its labor intensity, subjectivity, and logistical inefficiency. At the same time, the cost of adopting high-precision alternatives such as Ground-Penetrating Radar (GPR), Light Detection and Ranging (LiDAR), or laser profilometry is prohibitive for many public agencies.

While extensive research has been conducted on the detection of surface conditions for vehicular pavements and sidewalks, a critical gap remains in methods and datasets tailored for trails. Trails differ significantly from roads in surface material, exposure to environmental elements, lighting conditions, and patterns of wear, making them especially challenging for traditional and even advanced detection models. Moreover, most image-based crack detection systems are trained on clean, controlled pavement datasets and fail to generalize to the heterogeneous and unstructured nature of trails.

This research gap is particularly pressing considering recent efforts by transportation agencies to invest in nonmotorized infrastructure. For example, UDOT is actively expanding the Utah Trail Network (UTN), a statewide initiative that aims to connect communities through a comprehensive system of paved trails for walking, biking, and other non-motorized modes. With UDOT's Trails Division overseeing planning, design, construction, and operations, there is a growing need for scalable, low-cost, and automated inspection frameworks to support this expanding infrastructure.

Researchers and transportation agencies have begun deploying mobile sensor platforms such as the Data Bike, a bicycle equipped with georeferenced cameras and inertial sensors, to address these challenges. This approach leverages accelerometer data to detect "jerk" events (sudden changes in acceleration), which may correspond to uneven surfaces or surface damage. These events can then be mapped to specific GPS coordinates and corresponding image frames, narrowing the inspection task to areas of concern. This study builds upon this advancement by integrating multimodal sensor fusion and computer vision to develop a robust, trail-adapted surface condition detection framework that enhances efficiency and inclusivity.

1.2 Problem Statement

The maintenance and monitoring of trail infrastructure present a substantial logistical and financial challenge for cities and transportation agencies. Manual inspections, though common, are not scalable or reliable when applied to large-scale trail networks. These inspections require significant human resources, are prone to error and inconsistency, and often fail to capture subtle or early-stage defects. Moreover, trail conditions vary widely across neighborhoods regarding material type, lighting, elevation, and environmental exposure.

Advanced inspection tools such as LiDAR, GPR, or vehicle-mounted laser systems are practical but cost-prohibitive for local municipalities. Image-based crack detection systems show promise but require significant amounts of training data, are sensitive to environmental noise, and often fail in complex urban settings, such as shaded trails or overgrown walkways. Most current models are trained on clean, vehicular pavement datasets and do not translate well to trail surfaces. Pavlidis and Liow (1990) highlight a need of robust datasets and models targeting trail crack detection, particularly in scenarios with complex textures, vegetation, or shadow artifacts.

This research introduces an innovative, multi-modal data collection and analysis pipeline to address these limitations. The proposed method will integrate accelerometer-based jerk detection with visual imagery collected from a Data Bike platform. Jerk events will serve as temporal anchors to extract image frames likely to correspond to defects. These image segments will then be passed through advanced object detection models, such as YOLOv8, to identify and classify defects, including transverse cracks, longitudinal cracks, vegetation obstacles, and surface separation.

This research is not merely technical; it also addresses broader issues of accessibility and equity in transportation infrastructure. This project will enable more proactive maintenance strategies by providing a scalable and replicable methodology for detecting trail surface conditions. It will empower local agencies to deploy resources where they are most needed and enhance accessibility for pedestrians of all abilities. Moreover, using low-cost hardware and open-source algorithms ensures that this solution can be scaled and adapted by cities with varying levels of technological readiness and funding.

1.3 Objectives

This project aims to support the Utah Department of Transportation (UDOT) in enhancing the accessibility and safety of paved trail infrastructure by developing a low-cost, scalable, and automated trail defect detection framework. The proposed method integrates video collected from a camera-equipped Data Bike featuring a high-resolution camera with GPS and accelerometer sensor embedded with advanced computer vision models to detect and classify trail surface anomalies. This approach seeks to provide UDOT and similar agencies with an efficient inspection alternative to traditional manual or high-cost survey methods. The resulting system will assist in prioritizing maintenance interventions to improve trail usability, particularly for individuals with mobility challenges. This study includes several key phases:

1. Extensive Data Collection

Deploy a Data Bike across selected segments of the Utah Trail Network to collect synchronized accelerometer readings (for jerk detection) and high-resolution imagery.

2. Jerk-Based Event Localization

Analyze accelerometer data to identify “jerk” events, i.e., sudden surface elevation or texture changes, and align them with corresponding GPS coordinates and image frames.

3. Dataset Development for Trail Surfaces

Create and curate a labeled dataset of trail-specific surface anomalies under various environmental and lighting conditions to support model training and benchmarking.

4. Defect Detection via Computer Vision

Apply deep learning-based object detection models (YOLOv8) to the jerk-triggered image segments to identify and classify defects such as cracks, vegetation obstacles, and surface separations.

5. Visualize the Defects Integrating Geospatial Information

Identify the geographic location of detected issues and integrate them into an operational, user-friendly dashboard.

Table 1-1: Project Objectives and Alignment

Objective	Data Source / Method	Expected Outcome	Alignment with UDOT Goals
Extensive Data Collection	Data Bike (cameras, accelerometers, GPS)	Synchronized multimodal data for selected trail segments	Efficient acquisition of comprehensive trail condition data
Jerk-Based Event Localization	Accelerometer data analysis; custom scripting	Identification of potential defect locations	Reduction of manual inspection effort and time
Dataset Development for Trail Surfaces	Manual annotation; quality control	Comprehensive, labeled dataset for model training	Improved model accuracy and generalizability
Defect Detection via Computer Vision	YOLOv8 model; custom dataset	Automated classification of diverse trail defects and surface conditions	Objective and consistent defect identification
Visualize the Defects and Surface Conditions	Dashboard	Visualize the defects and surface conditions on the map using the dashboard	Defect location records

1.4 Outline of Report

The chapters of this report are outlined as follows:

Chapter 1: Introduction presents the context and importance of well-maintained trail infrastructure, particularly for individuals with limited mobility. It outlines the limitations of

traditional inspection methods and introduces the project's objectives, including automated defect detection, data collection using a Data Bike, and equity-focused maintenance planning.

Chapter 2: Literature Review surveys existing work in pavement inspection, sensor-based cycling assessments, computer vision methods for crack detection, and machine learning approaches. It emphasizes the limitations of traditional and early machine learning techniques and supports the shift toward deep learning for complex, trail-specific conditions.

Chapter 3: Research Methods details the technical framework, including the hardware setup of the Data Bike, data synchronization strategies, and the development of a YOLOv8-based object detection model. It also explains how defects and surface conditions are linked to GPS coordinates and describes the creation of a labeled dataset tailored to trail surface anomalies.

Chapter 4: Results and Discussion presents the dataset characteristics, model performance metrics, and spatial analyses. It demonstrates how geolocated defect detection enables hotspot mapping and supports proactive, data-driven trail maintenance, with added benefits for user safety and accessibility.

Chapter 5: Strategic Implications and Action Plan for UDOT provides a roadmap for scaling the system across Utah's trail network. It outlines short-, medium-, and long-term implementation phases, cost and time estimates, workforce integration, and potential partnerships to support UDOT's long-term accessibility goals.

Chapter 6: Conclusions summarizes the project's key contributions and findings, including the effectiveness of the Data Bike system and the YOLOv8-based detection model. It discusses limitations such as data imbalance and infrastructure variability, and offers recommendations for future work in expanding coverage, improving model generalization, and integrating equity-driven decision-making into trail maintenance.

2.0 LITERATURE REVIEW

2.1 Challenges in Manual and Traditional Pavement Inspection

Trail infrastructure maintenance presents numerous logistical and financial challenges, particularly for municipalities with limited resources. Traditional manual inspection techniques remain the most common approach to evaluating pavement and trail surface conditions. However, such methods have several drawbacks that limit their scalability and reliability. Manual inspections are labor intensive and require substantial workforce and time, often necessitating full deployment of inspection teams. These inspections are typically carried out under variable environmental conditions and are prone to subjective biases depending on the inspector's judgment and fatigue. This inconsistency significantly affects the reproducibility and quality of defect detection over large trail networks (Chen et al., 2021; Matarneh et al., 2024).

Moreover, trail surfaces often feature a greater degree of heterogeneity compared to vehicular roads. Variations in lighting due to tree canopy, inconsistent surface materials such as asphalt, concrete, gravel, or brick, and environmental influences like fallen leaves or shadows all pose unique challenges to manual inspection (Bhat et al., 2020). These inconsistencies make it difficult to spot early-stage or subtle defects that may still significantly impact pedestrian safety and accessibility. Additionally, the costs associated with comprehensive manual surveys, especially in high-mileage trail networks, make them infeasible for frequent or proactive maintenance.

In an attempt to enhance precision and coverage, some municipalities have begun adopting advanced sensing technologies such as LiDAR, Ground Penetrating Radar (GPR), and vehicle-mounted laser systems (Hu et al., 2021; Ashraf et al., 2023). While effectively capturing detailed structural anomalies, these systems come at a high financial cost. The associated costs include the purchase of the technology itself and the expenses involved in training personnel to operate it and processing the high-volume data output. Consequently, their use is restricted mainly to larger metropolitan areas or high-priority infrastructure segments, excluding community trails and pedestrian pathways from frequent monitoring (Bayar & Bilir, 2019; Hassan et al., 2022; Cohen et al., 2020).

2.2 Application of Sensors and Data Bikes for Cycling Infrastructure Assessment

Several studies have employed smartphones and accelerometers mounted on bicycles to evaluate surface conditions. Niska et al. (2024) utilized smartphones on regular bicycles (open-frame for women and closed-frame for men) to measure irregularities in cycleways. The VTI app converted tri-axial accelerometer data into an Evenness Indicator, aligning with subjective comfort ratings. Similarly, Zang et al. (2018) mounted a smartphone on the bicycle handlebars to gather acceleration and GPS data. They computed the International Roughness Index (IRI) to assess road surface roughness and developed an algorithm to identify potholes and humps based on acceleration spikes. Gao et al. (2018) also used a shared bicycle with an accelerometer (HOBO Pendant G) and a GPS logger to monitor vibrations on urban roads, combining quantitative measures with subjective cyclist evaluations. Gogola (2020) investigated road vibrations using an e-bicycle equipped with a smartphone-mounted accelerometer and supplementary GPS tracking, focusing solely on surface roughness.

Instrumented Probe Bicycles (IPBs) have been used in multiple studies to collect more comprehensive data. Xie et al. (2018) developed two schemes: one using an IPB equipped with accelerometers, GPS watches, and GoPro cameras to collect detailed cycling data, and another using an Arduino Mega 2560 microcontroller for integrated GPS and accelerometer data. The cameras captured real-time pavement conditions, providing reference information. Similarly, Zhu and Zhu (2019) employed an IPB with a video camera, GPS, accelerometer, and gyroscope to introduce the Cycling Comfort Index (CCI). They used video processing and machine learning (XGBoost) to identify critical road features. Cafiso et al. (2021, 2022) also utilized IPBs fitted with GNSS, cameras, and other sensors. The 2022 study used an Automatic Road Analyzer (ARAN) system with a Laser Cracking Measurement System (LCMS) and cameras to detect road surface conditions, with camera data-enhancing vibration data calibration and distress classification. In 2021, the study synchronized GNSS data with video recordings to identify and evaluate safety-critical events on various road types.

Other research has explored different sensor configurations and advanced data processing methods. Bil et al. (2015) used three types of bicycles (racing, touring, and mountain bikes) equipped with accelerometers to assess cycling comfort through the Dynamic Comfort Index

(DCI). They applied the Kolmogorov–Smirnov test to compare DCI distributions from two accelerometers. Olieman et al. (2012) mounted wireless ProMove2 inertial sensors at key locations on road and mountain bikes and a GPS receiver to analyze the impact of road surface, tire pressure, and speed on cycling vibrations. Hölzel et al. (2012) used a racing bicycle with an accelerometer mounted under the seat to evaluate comfort over various surfaces, using a 500 Hz sampling rate to capture vibrations across different cycling speeds.

In addition to accelerometer usage, some studies developed smartphone-based applications for data collection. Ho and Ren (2024) employed an instrumented bike paired with an Arduino board and a mobile app to collect real-time vibration data, using the Long Short-Term Memory (LSTM) method to identify anomalies. Low and Krisp (2024) introduced a smartphone-based roughness-centric routing approach using accelerometers on mountain and city bikes. The Physics Toolbox Suite application was used for data storage and geotagging, with a roughness index developed from the vertical acceleration data. Wage et al. (2020) developed the RideVibes app to measure accelerations in three axes and GPS data, transforming it into a comfort index for assessing bike path roughness.

A few studies used alternative sensor technologies or methodologies. Rizelioğlu and Yazıcı (2023) developed a system using a mountain bike with a three-axis accelerometer and a GPS module mounted on the handlebars to measure vertical displacements and road roughness. They validated their findings with a laser profilometer. In a subsequent study, Rizelioğlu et al. (2024) employed polyvinylidene fluoride (PVDF) sensors mounted symmetrically on the front wheel to detect road roughness through tire-road interaction, with GPS for location data. They used the Support Vector Regression (SVR) algorithm to estimate IRI values. Neither study incorporated camera data, focusing solely on accelerometer and GPS measurements.

While most research focused on accelerometer and GPS data, some studies leveraged camera data for more detailed analyses. Lee et al. (2021) used a dual-acquisition system with a smartphone's camera and accelerometer. Mounted on a vehicle windshield, the camera captured high-resolution images, while the accelerometer recorded vibrations. A fully convolutional neural network (FCN) processed the photos to identify road anomalies, enhancing the accuracy of road condition assessments by combining visual and vibration data. Cafiso et al. (2022) also integrated

camera data using the ARAN system, providing a visual reference for distress classification that complemented vibration data to identify specific road defects.

These studies highlight the effectiveness of combining accelerometer data with other sensors, such as GPS and video cameras, to assess cycling infrastructure. Some research emphasized the integration of cameras to provide visual context, while others relied on accelerometer and GPS data alone. Multi-sensor systems, including microcontroller-based setups like Arduino and smartphone applications, show potential for large-scale analysis and improved understanding of cycling comfort.

2.3 Image-Based Crack Detection and Traditional Computer Vision

Computer vision has long been utilized to detect cracks and surface defects, starting with traditional image processing methods. These approaches initially relied on basic pixel-level operations such as grayscale conversion, histogram equalization, and thresholding to segment crack features from their background (Yi et al., 2019, 2020; Tang et al., 2019). Algorithms like the Canny edge detector and directional segmentation were employed to identify sharp gradients and edges corresponding to potential cracks (Ding & Goshtasby, 2001). While these techniques offered a quick and computationally inexpensive means of identifying surface anomalies, they lacked the sophistication necessary to handle complex visual environments typical of pedestrian trail systems (Han et al., 2021; Jie & Liu, 2012; Rong et al., 2014).

In trail contexts, the visual scene is often cluttered by overgrown vegetation, dappled lighting due to tree canopy, and irregular surface textures, resulting in significant noise for threshold-based methods. As such, early vision-based methods frequently generated false positives or failed to detect subtle cracks altogether. This lack of robustness has hindered their deployment in large-scale trail maintenance operations. Image-based methods also struggle with environmental factors like shadows, wet surfaces, and varying material reflectance, which interfere with pixel intensity-based decision rules (Quan et al., 2019; Li et al., 2020).

Subsequent refinements in traditional computer vision attempted to introduce contextual analysis into crack detection pipelines. Region-growing algorithms and neighborhood histogram analysis were implemented to group visually similar pixels and reduce false positives (Li et al.,

2020; Piironen et al., 1990; Zheng et al., 1990; Pavlidis & Liow, 1990; Shneier, 1983). Techniques such as morphological filtering and directional segmentation expanded improved detected cracks' continuity. However, these improvements were often narrowly tuned to specific pavement types or lighting conditions, limiting generalizability. Furthermore, in practical scenarios, these algorithms require preprocessing steps such as noise filtering and contrast enhancement, making real-time deployment challenging (Hu et al., 2021; Chakurkar et al., 2023).

The inadequacy of these methods in uncontrolled outdoor environments, like shaded pedestrian trails, underscores the need for more adaptable and intelligent detection systems. The inability of traditional computer vision to learn spatial hierarchies and context from the data limits its scalability and accuracy. This gap has increasingly directed attention toward learning-based techniques, particularly deep learning, which can capture complex features and contextual relationships through end-to-end training. Still, these traditional methods laid the foundation for subsequent innovations and remain useful for benchmark comparisons in assessing more advanced approaches.

2.4 Machine Learning Approaches and Their Limitations

The shift toward machine learning-based approaches marked a significant advancement in pavement defect detection, attempting to overcome the limitations of traditional image processing. Machine learning models utilize manually extracted features such as textures, gradients, and geometric descriptors to classify crack and non-crack regions. Popular algorithms in this domain include Support Vector Machines (SVM), Random Forests (RF), k-Nearest Neighbors (k-NN), and ensemble learning methods (Hoang & Nguyen, 2018; Zhang et al., 2019; Wang et al., 2019, Farhadmanesh et al., 2024). For instance, Cord and Chambon (2012) developed a supervised learning framework that emphasized the spatial texture characteristics of cracked pavements, applying linear and nonlinear filtering combined with AdaBoost classifiers to improve classification outcomes.

While these models outperformed early image thresholding methods, they heavily relied on handcrafted features, making them sensitive to environmental variations such as lighting changes, debris, and vegetation commonly found on trails. The extraction of robust features for

diverse trail conditions remains a challenging task. Models like CrackForest employed Random Structured Forests trained on grayscale image patches to achieve good results on simplified urban roads. Still, they failed to generalize across more complex, unstructured trail networks (Shi et al., 2016).

Another drawback of these models lies in their lack of transferability. Features designed for one pavement type may not be practical on another, necessitating constant re-engineering and tuning of the model pipeline. In most studies, datasets were limited in size and diversity, restricting the models' capacity to generalize across different surface materials or environmental settings. Additionally, training machine learning models typically involves extracting statistical summaries or texture descriptors from small image regions, which prevents the models from capturing higher-order spatial relationships essential for distinguishing between different types of defects (Jiang et al., 2025).

Despite these challenges, machine learning techniques laid necessary groundwork for more sophisticated crack detection systems. Their use highlighted the critical importance of robust feature extraction and variability in data inputs. These insights have helped shape subsequent research toward more flexible, adaptive models. The limitations in performance under complex environmental noise, difficulty in feature generalization, and high dependency on manual engineering eventually prompted the research community to pivot toward deep learning models capable of learning hierarchical representations directly from data.

2.5 Deep Learning-Based Defect Detection

Deep learning has revolutionized crack detection by automatically allowing systems to learn features from data through end-to-end training pipelines. Convolutional Neural Networks (CNNs) have demonstrated impressive capabilities in extracting multi-scale, hierarchical features essential for identifying complex crack patterns. CrackNet and its improved version, CrackNet-V, were among the early deep learning models that focused on pixel-level crack segmentation without pooling layers, maintaining resolution and spatial accuracy (Qu et al., 2022).

More recent architectures like DeepCrack, FCN, and U-Net have further improved detection by incorporating encoder-decoder structures. These models can fuse low-level spatial

and high-level semantic information, crucial for detecting thin, irregular, or obscured cracks often found on pedestrian trails. For example, DeepCrack uses multi-scale feature extraction and skip connections to enhance its sensitivity to fine-grained and large-scale features. Such models can generalize better to diverse environmental settings, including shadowed, overgrown, or rough trail conditions (Liu et al., 2019; Li et al., 2022).

Nonetheless, the performance of deep learning models is tightly linked to the quality and diversity of training data. Most public datasets used in training—like CRACK500 or CFD—are composed of clean, high-resolution vehicular road images, which do not accurately reflect the conditions of community trails. This mismatch in data distribution leads to poor performance when models are deployed in real-world trail environments. Furthermore, even though CNNs reduce the need for handcrafted features, they often require large, labeled datasets to train effectively, which poses an issue in domains like trail inspection where annotated data are scarce (Tang et al., 2024; Yang et al., 2020).

Despite these challenges, the flexibility of deep learning has allowed for innovations like multitask learning, attention mechanisms, and transfer learning, which have improved robustness in noisy or complex settings. Models such as CrackLens introduce transformer-based segmentation that excels in identifying cracks under vegetation or partial occlusion (Koh et al., 2024). These advancements mark a significant step forward in trail-specific defect detection and suggest that future systems must further tailor architecture and training protocols to accommodate the idiosyncrasies of non-vehicular surfaces.

2.6 Summary of Literature Analysis

The existing literature reveals substantial limitations in traditional trail inspection methods and early sensing technologies. Manual inspections are labor-intensive, inconsistent, and costly, making them unsuitable for proactive or large-scale trail maintenance. While some municipalities have experimented with high-resolution systems like LiDAR and GPR, their expense and operational complexity make them inaccessible for routine use, particularly on community trails. More recent approaches involving instrumented bicycles and accelerometer-based sensing have

shown promise in evaluating surface roughness and comfort. Yet, most studies rely either on vibration data alone or lack automated defect classification capabilities.

Deep learning has emerged as a powerful alternative for detecting surface cracks under varied and challenging trail conditions. Deep learning models like YOLO and U-Net offer robust feature extraction and higher generalization compared to traditional image processing or classical machine learning. However, most applications focus on vehicular roads and lack trail-specific datasets, limiting model performance on nonstandard, diverse trail surfaces. This gap underscores the need for adaptable, low-cost systems that can be deployed across wide trail networks with minimal technical overhead. This study directly addresses that need by combining jerk-based accelerometer signals with deep learning-based visual detection and GPS mapping, producing interpretable, GIS-ready outputs. For UDOT, this approach offers a scalable and cost-efficient solution to modernize trail condition monitoring, enabling proactive, equitable, and data-driven maintenance strategies that enhance accessibility and safety for all users.

3.0 RESEARCH METHODS

This section outlines the comprehensive methodology employed in this project, from assembling the custom Data Bike to the data processing pipeline and deep learning model implementation.

3.1 Equipment Setup and Data Collection

A custom-built Data Bike was deployed for multimodal data acquisition across various trail types and environmental conditions to facilitate detailed and cost-effective trail condition monitoring. The platform was designed to support agile, scalable surveying while maintaining operational safety and data consistency rides.

Hardware Configuration and Mounting. The Data Bike was constructed using a Magnum Voyager electric bicycle as the base frame with aluminum alloy frame. It was equipped with dual high-resolution GoPro cameras, one mounted at the front and another at the rear, allowing for bidirectional surface and environmental imaging (as shown in **Error! Reference source not found.**). Acc[illegible]vere collected from the GoPro embedded metadata

Figure 3-1: Data Bike setup

(tri-axial accelerometer) and recorded redundantly using a smartphone vibration-sensor application.

The total hardware setup, including a $1,500 bike and an additional $1,000 for other equipment such as GoPro cameras, amounted to approximately $2,500, making it a cost-efficient solution for large-scale trail inspection.

Pilot and Refinement. Pilot data collection was conducted on the 9-Line Trail to validate mounting stability, image clarity, and vibration sensitivity under field conditions. Insights from the pilot were used to refine camera angles, reinforce mounts, adjust sampling frequency, and tune vibration-sensitivity thresholds to improve detection reliability. For example, camera pitch was lowered by ~5° to better capture pavement texture, front suspension lockout was engaged to reduce damping and amplify vibration sensitivity, and jerk thresholds were tuned by comparing known smooth and rough segments (thresholds are detailed in Appendix A).

Full-Scale Data Collection. The primary data-collection campaign was conducted over four months (May–August), with additional supplemental rides completed in September–October cover additional trails. Surveys were conducted mainly on weekends and during daylight hours to capture variability in lighting conditions. Typical riding speed along survey segments was maintained between 10–15 mph to balance safety and data quality; this speed range ensured that jerk thresholds triggered reliably across similar surface conditions. Each survey required a two-person crew (rider and support/data monitor), with approximately 0.15–0.20 person-hours expended per trail mile for data collection, and 15 – 30 minutes pre-ride checks, and post-ride quality control per deployment. Data products included synchronized video footage, GPS traces, and accelerometer readings. A comprehensive data collection protocol was developed to standardize survey procedures. This included predefined routes, data capture intervals, camera orientation, sensor synchronization, and operator safety guidelines. Graduate students were thoroughly trained in Data Bike operation, including equipment handling, field troubleshooting, GPS path tracking, and adherence to safety measures.

Study Trails and Selection Rationale. Five trails were selected (

Figure ,

Table) to capture variation in surface material (asphalt vs. concrete), lighting and shading, environmental context (urban vs. suburban), and maintenance condition. Selection criteria included the presence of representative surface defects, surface conditions, variation in usage intensity, and stakeholder input. For instance, the 9-Line Trail was chosen for the pilot because it offered both asphalt and concrete surfaces, moderate user volumes, tactile pavement at crossings, and rail tracks crossing the path. This combination provided diverse contexts for calibrating the system.

Table 3-1 provides a comprehensive breakdown of the environmental characteristics captured at each location, demonstrating how site selection directly supported dataset variability objectives.

Table 3-1: Environmental Characteristics of Data Collection Sites

Trail Name	Surface Composition	Canopy Cover	Lighting Conditions	Collection Time	Urban Context
9 Line	90% concrete/asphalt	Sparse to partial; mixed trees	Clear/bright; some cloudy; partial shade with direct sun	8-10 AM	Suburban; mixed use
3600 S SUP	90% concrete/asphalt	None; open corridor	Direct sunlight; no shade	1-2 PM	Urban; heavily paved/formal
D&G Trail	Asphalt	Mixed: mostly open; small dense sections; sparse forest	Clear; some sections shaded with dense canopy	12-1 PM	Recreational; cycling; mixed use
Jordan River Parkway	90% asphalt/10% concrete	Sparse; some open sections; debris noted	Sun with shaded regions	3-4 PM	Recreational; mixed use
Porter Rockwell	Asphalt	Open corridor	Cloudy; overcast	2-3 PM	Recreational

The 9 Line Trail was chosen for the pilot because it offered both asphalt and concrete surfaces, moderate user volumes, tactile pavement at crossings, and rail tracks crossing the path. This combination provided diverse contexts for calibrating the system. The Jordan River Parkway

(90% asphalt/10% concrete) was selected primarily for its extensive asphalt sections with mature crack development and realistic debris accumulation. The 3600 S SUP was chosen specifically to capture concrete pavement conditions in a heavily formalized urban setting with direct sunlight exposure. The D&G Trail and Porter Rockwell Trail were included to represent fully asphalt recreational corridors under varying canopy and weather conditions (clear vs. overcast), ensuring the dataset would capture the full spectrum of trail environments encountered in practice.

This site selection strategy systematically captured environmental diversity across critical dimensions:

Lighting Condition Diversity.

- **Direct sunlight** (unobstructed sun exposure): 3600 S SUP (midday), portions of Porter Rockwell Trail, open sections of Jordan River Parkway, and D&G Trail representing approximately 40% of collection conditions
- **Partial shade** (mixed canopy with dappled light): 9 Line Trail, sparse canopy sections of Jordan River Parkway, and D&G Trail representing approximately 35% of collection conditions
- **Full shade** (dense canopy or structural shadows): Small dense sections of D&G Trail, heavily shaded portions of Jordan River Parkway representing approximately 15% of collection conditions
- **Overcast/diffuse lighting**: Porter Rockwell Trail representing approximately 10% of collection conditions

Surface Type Distribution.

- **Asphalt surfaces**: D&G Trail (100%), Porter Rockwell Trail (100%), Jordan River Parkway (90%), portions of 9 Line Trail representing approximately 70% of collection sites

- **Concrete surfaces**: 3600 S SUP (90%), portions of 9 Line Trail, portions of Jordan River Parkway (10%) representing approximately 30% of collection sites

This distribution reflects the predominance of asphalt in the regional trail network while ensuring adequate concrete representation for model generalization across pavement types.

Temporal and Contextual Variation.

- Collection spanned multiple months (May through October) to capture seasonal differences in sun angle and vegetation density

- Data was collected systematically across different times of day: morning (8-10 AM: 9 Line), midday (12-2 PM: D&G, 3600 S), and afternoon (2-4 PM: Porter Rockwell, Jordan River Parkway)

- Sites represented diverse land use contexts: urban sidewalk consisting of (3600 S SUP), suburban mixed-use (9 Line), and recreational natural corridors (Jordan River Parkway, Porter Rockwell, D&G)

Standard Operating Procedure (SOP). A structured protocol was established to standardize field operations and improve repeatability. Key elements included:

- **Route definition and coverage:** predefined trail segments were used to ensure consistent spatial coverage along with variance in trail conditions.
- **Pre-ride checks:** riders inspected mounts, battery levels, lens cleanliness, and storage capacity; GPS logging and camera synchronization (using the *GoPro Remote* app) were confirmed before departure.

- **Riding practice and speed control:** riders maintained the target speed range (10–15 mph) and followed a "ride-over" rule for surface anomalies to ensure that vibrations were captured where safe; avoidance maneuvers were used only when necessary for safety.
- **Safety practices:** riders adhered to traffic rules at crossings, communicated intentions to other trail users, and exercised conservative riding behavior in mixed-use environments.
- **Post-ride quality checks:** data files were reviewed for completeness, and any anomalies were documented.

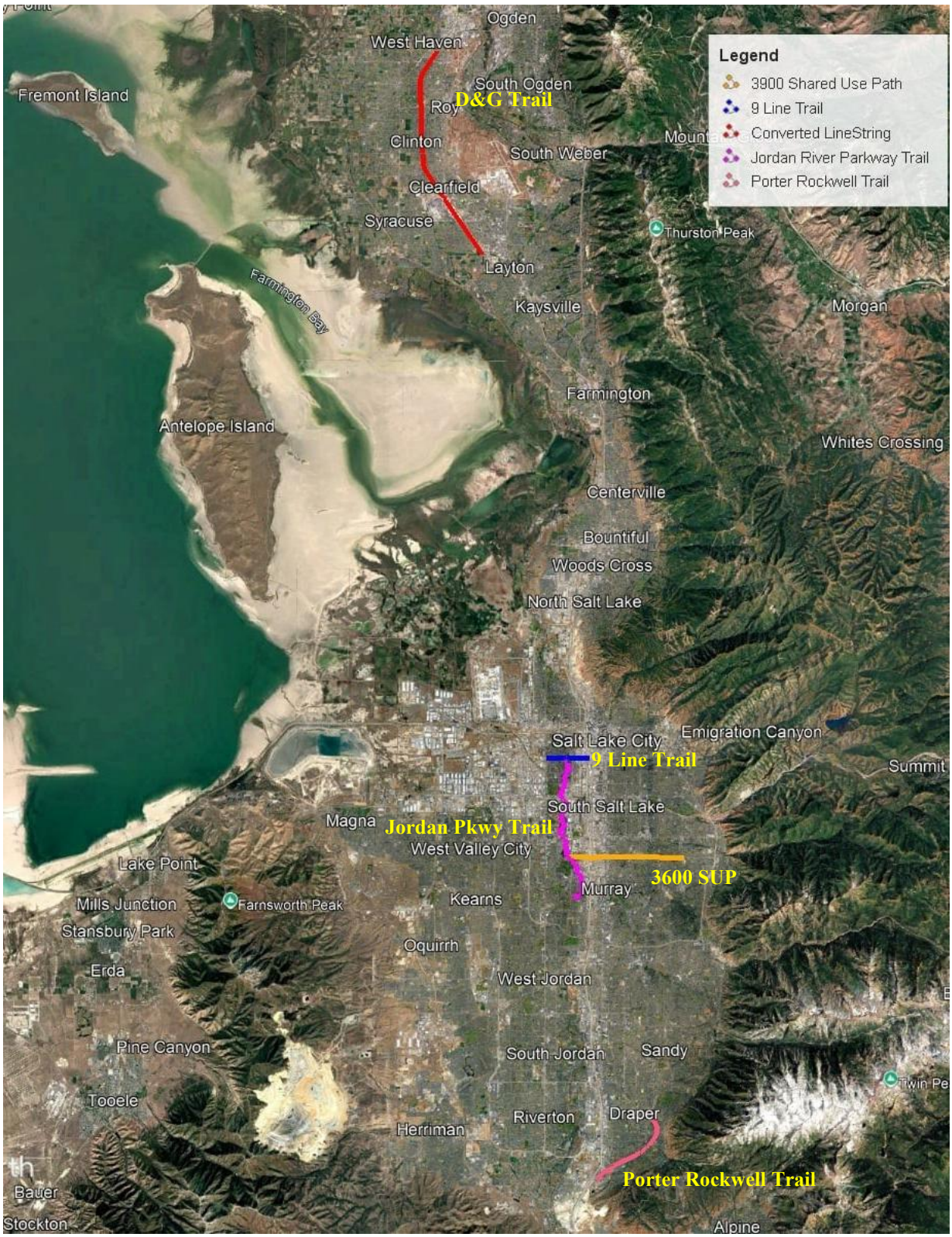


Figure 3-2: Data collection sites for this study

Table 3-2: List of Trails Included in Data Collection

Serial Number	Trail Name	Length (miles)
1	9 Line Trail	2
2	Aldo Tree to Jordan Park (3600S Shared Use Path (SUP))	6
3	Jordan River Parkway Trail	9
4	D&G Western Rail Trail	12
5	Porter Rockwell Trail	5

3.2 Data Extraction and Processing

To identify trail segments exhibiting potential surface irregularities or obstructions, accelerometer data collected from the Data Bike were analyzed using a multi-step signal processing pipeline. The objective was to detect high-jerk events, i.e., sudden changes in acceleration that often indicate rough or degraded surface conditions. These events were subsequently mapped to corresponding image frames from the GoPro footage for visual verification.

Timestamping and Synchronization. Accelerometer and GPS metadata embedded in the GoPro video files were used as the primary synchronized data stream. Recording was initiated using a multi-camera control/synchronization application (GoPro Remote) to align start times across devices, and embedded metadata timestamps were used to maintain alignment between vibration events, GPS locations, and video frames. When small offsets were observed among devices, they were corrected by aligning synchronization markers at the time of post processing.

Acceleration Magnitude and Smoothing. Tri-axial accelerometer signals were converted to a single magnitude time series. A zero-phase 4th-order Butterworth low-pass filter was applied to reduce high-frequency sensor noise prior to differentiation (see Appendix A). Threshold calibration rides were conducted over known smooth and rough segments to determine jerk thresholds that minimize false alarms while capturing relevant events.

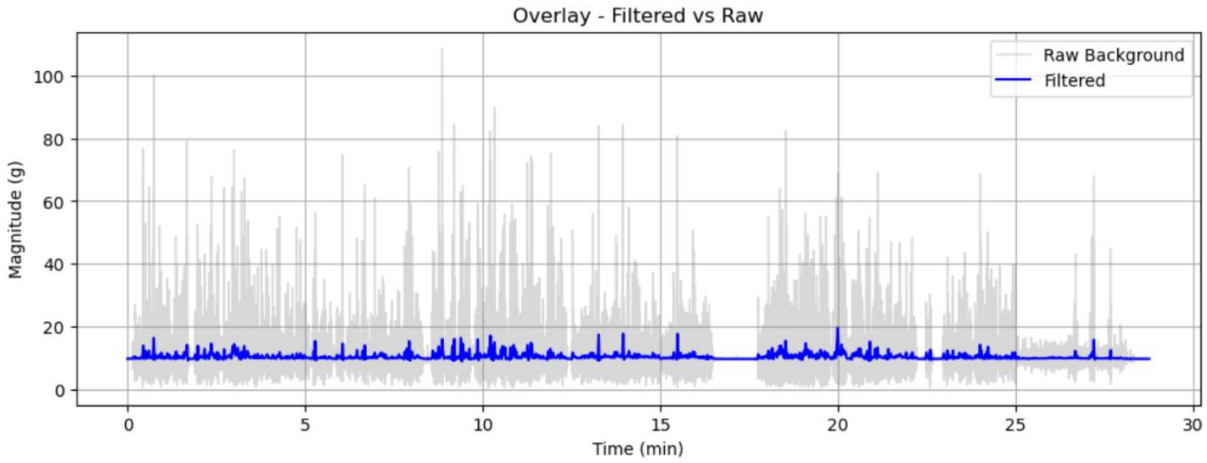


Figure 3-3: Raw and Filtered Magnitude of Acceleration

Jerk Computation and Thresholding. Jerk was computed as the derivative of the smoothed acceleration magnitude. The threshold for event detection was empirically determined from pilot rides and maintained constant during full-scale surveys; the threshold value and calibration methodology are documented in Appendix A. Because jerk magnitude depends on

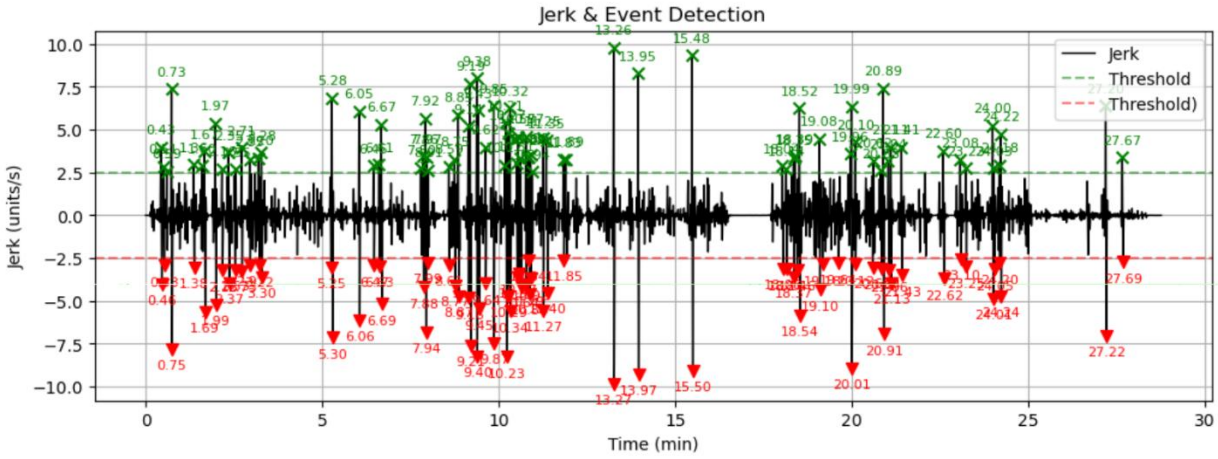


Figure 3-4: Detection of jerk based on threshold calibration

speed, maintaining a speed range (10–16 mph) was essential for reproducibility. GPS-derived speed was used to screen out near-zero-speed intervals (stops) and to interpret unusually large events that may correspond to emergency braking. We can see sample plot showing the jerks as shown in Figure 3-4.

Frame Extraction Offsets. Upon identifying high-jerk timestamps, the corresponding video frames were extracted from the synchronized GoPro footage using automated scripts. We

applied multiple temporal offsets during frame extraction to ensure accurate spatial alignment between jerk events and their causative events. Since the camera visually captures defects and surface conditions before the bike physically encounters them (and thus before vibrations are felt), we extracted frames at 0.25, 0.5, and 0.75-second intervals prior to each jerk timestamp. This multi-offset approach addresses the inherent delay between visual detection and physical response while providing images of defects and surface conditions at varying distances from the camera. The resulting three images per jerk event capture the defect from different perspectives and scales,

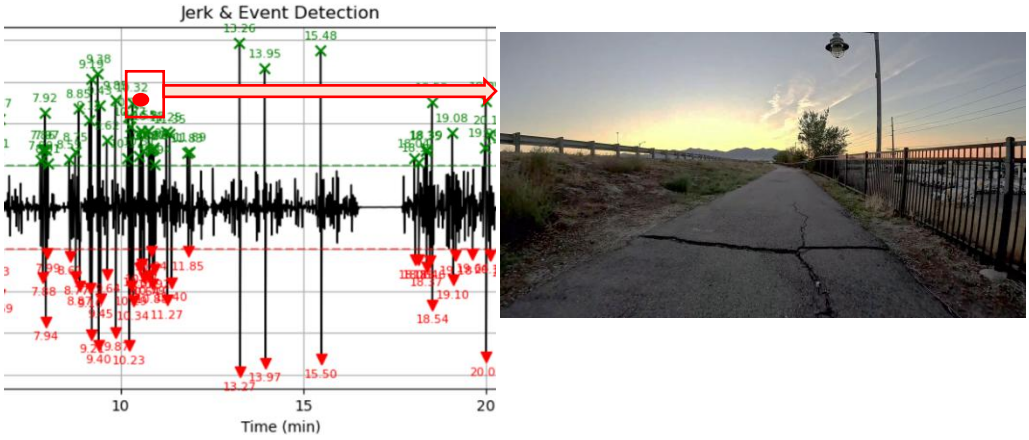


Figure 3-5: Extraction of crack image based on jerk timestamp

improving model robustness and ensuring that at least one frame accurately represents the defect that caused the corresponding vibration. The extracted frames were compiled into a dedicated repository for manual inspection and subsequent analysis using object detection models to classify surface defects and surface conditions (Figure 3-5).

Interpretation for Obstacles and Limitations. The jerk-triggered extraction targets surface defects and surface conditions that produce measurable vertical vibration. Obstacles that are avoided without inducing significant vibration may be under-sampled, although three-axis accelerometer data can capture lateral maneuvers (y- and z-axis) when the rider swerves around an obstacle. Vegetation obstacles or gradual narrowing that requires only slight steering may not trigger a jerk event; this limitation motivates future enhancements such as periodic frame sampling or continuous video-based scanning.

3.3 Computer Vision Model for Surface Condition Detection

This project utilized a deep-learning object-detection framework built on YOLOv8 (You Only Look Once, version 8) to automate identification of trail surface defects and conditions. Renowned for its balance of speed and accuracy, YOLOv8 offers an effective solution for detecting and classifying multiple object types within images. Its efficient architecture makes it well-suited for processing the image datasets collected from mobile platforms like the Data Bike, enabling systematic analysis of trail conditions at scale.

The methodology for surface defects and conditions detection comprises the following significant steps:

- **Model Initialization:** A pre-trained YOLOv8 object detection model was loaded as base architecture. This model, originally trained on the Common Objects in Context (COCO) dataset, a large-scale, publicly available benchmark used for object detection, segmentation, and image captioning tasks in computer vision research, was selected for its robust generalization capabilities and transfer learning potential.
- **Image Preprocessing:** Each jerk-triggered frame was preprocessed by resizing to the YOLOv8 input shape, applying standard intensity normalization, and converting to the expected tensor/image format. The experiments reported here use a 640×640 resolution; however, YOLOv8 supports multiple input sizes, allowing for adjustments to accommodate computational budget and detection performance.
- **Custom Model Training:** The base YOLOv8 model was fine-tuned using a custom-labeled dataset comprising images of trail defects and surface conditions captured during field surveys. Annotations included bounding boxes and class labels for each defect type. Data augmentation techniques such as flipping, brightness variation, and rotation were applied to enhance model generalization across varying trail and lighting conditions.
- **Taxonomy:** The classification scheme was designed to capture various trail conditions and obstructions. The model was trained to detect and classify the following ten surface defects and condition categories:
    - Pothole
    - Longitudinal Crack
    - Transverse Crack
    - Alligator Crack
    - Misalignment
    - Vegetation Obstacles
    - Pole Obstacle (fixed object near/in path)
    - Debris Obstacle (loose materials on surface)

- Tactile Pavement
- Rail Crossing

This taxonomy encompasses both structural defects (cracks, potholes) and contextual features (tactile pavement, rail crossings) that affect trail usability and accessibility.

- **Inference Phase:** During the inference phase, each preprocessed frame was processed through the trained YOLOv8 model. The model output included bounding box coordinates, class labels, and confidence scores for each detected object, enabling precise localization and classification of trail anomalies.
- **Post-Processing and Filtering:** Model outputs were filtered using a minimum confidence threshold (e.g., 0.40). In this study, a false positive refers to a detection reported by the model that does not correspond to an issue/condition in the image (i.e., a spurious detection). Non-maximum suppression (NMS) was applied to remove duplicate, overlapping bounding boxes that refer to the same defect. The confidence threshold was selected to strike a balance between spurious detections and missed issues.

The resulting outputs enabled precise spatial localization and classification of trail anomalies, supporting subsequent statistical analyses and geospatial visualization. The model's performance was systematically evaluated using standard object detection metrics on a held-out validation set. Key metrics included:

- Precision: The proportion of correct detections among all detections made by the model
- Recall: The proportion of actual defects and surface conditions successfully detected by the model
- Mean Average Precision (mAP): A comprehensive metric that summarizes precision across different confidence thresholds and object classes

These metrics guided iterative refinements to training data composition, augmentation strategies, and model hyperparameters, ensuring robust and reliable defect detection across diverse trail environments. The resulting detection outputs enabled precise spatial localization and classification of trail anomalies, supporting subsequent statistical analyses and geospatial visualization components of the project.

3.4 Domain Adaptation Considerations

A significant challenge in this project was the limited size of the locally collected trail dataset, which contained approximately 4,971 instances in the locally available dataset. To address this data scarcity, the project incorporated the GRDD dataset as supplementary training data.

However, this approach introduced domain adaptation considerations, as GRDD primarily comprises vehicular road imagery captured from roadways, which differ from trail environments in several aspects including path width, edge characteristics, and surrounding context.

Rationale for Cross-Domain Transfer. Despite these differences, these factors support the transferability of knowledge from GRDD to trail applications:

- *Shared Defect Morphology:* The visual appearance and structural characteristics of common pavement defects, including potholes, longitudinal cracks, transverse cracks, and alligator cracks, remain consistent regardless of whether they occur on roads or paved trails. These defects result from similar degradation mechanisms (thermal expansion, water infiltration, and load stress) and exhibit comparable visual patterns. Consequently, features learned from GRDD imagery are directly applicable to identifying these same defect types in trail contexts.

- *Surface Material Similarity:* All the trail segments surveyed in this study feature asphalt or concrete paving, materials that closely match those represented in the GRDD dataset. While edge environments differ, roads typically have curbs and shoulders, whereas trails may border grass, gravel, or natural vegetation. These edge differences do not substantially alter the visual characteristics of surface defects themselves, which remain the primary focus of detection.

Hybrid Training Strategy and Data Scarcity. To train the model, we employed a hybrid dataset strategy. We combined our locally collected trails data with the massive GRDD road dataset. This provided the model with ample examples of structural defects (cracks, potholes), which are morphologically similar across both domains. However, for trail-specific "contextual obstacles" (such as Pole Obstacles, Debris, and Vegetation), the training data was limited to the few instances captured during our data collection. Consequently, the model's current ability to detect these rare classes is constrained by data scarcity.

3.5 Geospatial Integration

Following the detection and classification of trail surface defects and conditions using the computer vision model, the next critical step involves associating each defect with its precise geospatial location. This geospatial integration enables spatially anchored assessments of trail conditions, facilitates targeted maintenance planning, and supports downstream visualizations and spatial analytics. This effort results in the development of an interactive, web-based dashboard designed to provide UDOT stakeholders with an intuitive and comprehensive view of trail conditions.

The integration process merges outputs from the deep learning model with synchronized GPS data collected concurrently via GoPro cameras mounted on the Data Bike, which is embedded within the prediction script. The key steps involved are as follows:

- **Mapping Detections to GPS Coordinates**: Each defect identified by the object detection model is first linked to the corresponding image frame. Using synchronized timestamps, the system matches each frame to the nearest GPS record, thus assigning latitude, longitude, and (if available) altitude information to each detected defect.
- **Timestamp Alignment:** Accurate synchronization between the image capture times and GPS logging is essential to ensure spatial fidelity. At typical survey speeds (10–15 mph), small timing mismatches between video/jerk events and GPS can translate into noticeable along-track shifts. We applied a fixed time offset to align the streams, testing candidate offsets of 0.25 s, 0.50 s, and 0.75 s (≈ 1–5 m at 10–15 mph) and selecting the value that best aligned visually identifiable landmarks/defects with their mapped locations along the GPS track, achieving sub-second synchronization.
- **Structured Output Generation:** The integrated data is then exported into a structured format, typically a CSV file (e.g., predictions_with_gps.csv), consolidating all relevant metadata. Each row in the dataset includes:
  - A unique identifier for the detection instance.
  - The defect category (e.g., pothole, vegetation_obstacle).
  - The model's confidence score for the classification.
  - Geographical coordinates (latitude and longitude).
  - The timestamp of detection.
  - The filename of the source image.
  - Bounding box coordinates (x, y, width, height) within the image.

This spatially enriched dataset provides a robust foundation for subsequent trail condition analysis, enabling heatmaps, cluster detection, and maintenance prioritization. Moreover, it

facilitates integration with interactive GIS dashboards and digital trail inventories, significantly enhancing the transparency and accessibility of inspection data for planners, engineers, and stakeholders.

3.6 Dashboard Development

A three-step processing pipeline generated a spatially enriched dataset of trail surface conditions. First, frames were extracted from bicycle-mounted video using synchronized accelerometer and GPS data to identify vibration events indicative of potential surface defects and surface conditions. Second, a pretrained YOLOv8 object detection model was applied to these frames to identify and geolocate defects, producing a structured output containing bounding boxes, classifications, confidence scores, and coordinates. Finally, we developed the dashboard to translate this dataset into actionable insights. This tool, built using Streamlit for its robust and easy-to-use visualization capabilities, provides a user-centric interface that enables planners, engineers, and maintenance crews to interactively explore defect locations, compare pre- and post-maintenance conditions, and prioritize interventions without requiring specialized data science expertise.

3.7 Dataset Development and Annotation

The performance of any object detection model is highly dependent on the quality and relevance of the training data. In this project, a significant effort was devoted to curating a specialized dataset that accurately captures the visual characteristics of trail defects and conditions across varied surface types and environmental conditions.

The dataset development and annotation process involved the following key steps:

- **Image Collection:** Raw video footage was collected using the Data Bike across the five selected trail segments documented in

- Table 3-2. As detailed in Section 3.2, the site selection intentionally captured diverse environmental contexts including varied lighting conditions (direct sun to overcast), surface materials (asphalt and concrete), canopy cover (open to dense shade), seasonal vegetation states, and realistic surface debris patterns as shown in Table 3-1. This approach to environmental diversity improved the dataset's variability and real-world applicability.
- **Preprocessing for Annotation:** From the collected footage, relevant image frames were extracted using timestamp-aligned jerk events (Section 3.2). These frames were then formatted and resized to prepare them for manual annotation.
- **Manual Annotation:** Expert annotators manually inspected each frame and drew bounding boxes around instances of defined defect categories. Each bounding box was labeled with a class name such as pothole, longitudinal_crack, or vegetation_obstacle. This process, though labor-intensive, is essential to provide high-quality supervision signals for model training. Because the model needed to be both trained and tested, all defect images were manually annotated. The annotated dataset presented in Appendix B, reflects the real-world prevalence of surface defects and surface conditions observed in the project. While deep learning models generally benefit from hundreds to thousands of labeled instances per category, leveraging transfer learning and targeted augmentation allowed the model to effectively learn from the real-world class distributions available in this study. This approach aligns with best practices when working with imbalanced datasets in surface defect detection.
- **Quality Control:** All annotated images underwent a multi-phase quality assurance process to ensure consistency and reliability. This included cross-validation by multiple reviewers, correction of mislabels, and refinement of bounding box boundaries to improve spatial precision.
- **Dataset Splitting**: The curated dataset was partitioned into three subsets and is presented in Appendix B:
    - **Training set** – used to optimize model weights.
    - **Validation set** – used to tune hyperparameters and evaluate model performance during training.
    - **Test set** – held out for final evaluation to ensure unbiased assessment of model generalization.

Dataset Robustness and Real-World Applicability

This tailored dataset captures the unique challenges and visual complexities of trail environments. The systematic site selection strategy documented in Table 3-1, combined with quantified environmental diversity across lighting conditions (from overcast to direct midday sun), surface types (asphalt-dominant to concrete-dominant), canopy cover (open corridors to dense shade), collection times (morning through late afternoon), and land use contexts (urban formal to

recreational natural), ensures that the training dataset reflects the range of conditions encountered in operational trail assessment scenarios. This documented environmental variability provides empirical support for the model's ability to generalize beyond the specific trails and conditions present during initial data collection, directly addressing the practical requirement that automated detection systems maintain reliable performance across the heterogeneous conditions characteristic of real-world trail networks. The diversity and annotation accuracy of this dataset significantly contribute to the YOLOv8 model's ability to detect and classify defects and surface conditions under varying field conditions, ensuring robustness in real-world deployment scenarios. Detailed class counts and distribution statistics are provided in Appendix B, and representative annotated examples illustrating the range of surface conditions and environmental contexts are shown in Appendix C.

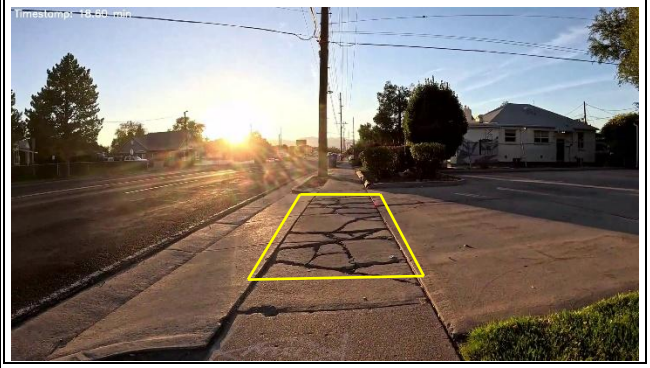
4.0 RESULTS AND DISCUSSION

4.1 Overview of Surface Deficiencies on Trails

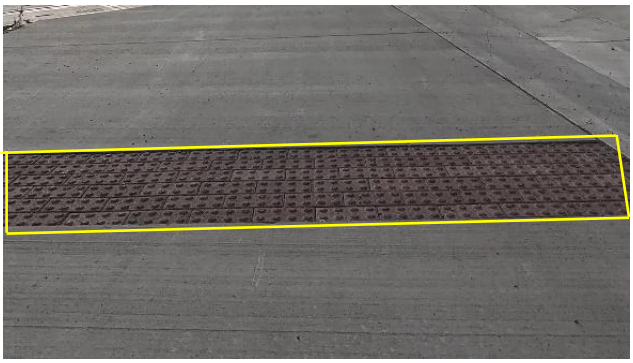
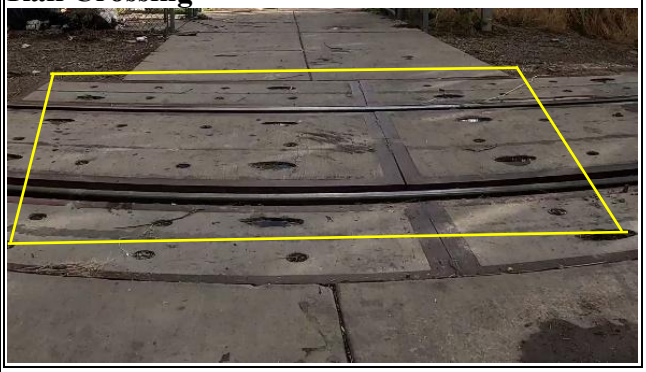
This study considers 10 types of trail surface conditions and contextual obstructions relevant to trail safety, accessibility, and maintenance planning. Surface defects (cracking, potholes, misalignments) can directly affect ride quality and safety. Contextual obstructions/features (e.g., vegetation obstacles, poles, tactile pavement, rail crossings) are included because they are operationally relevant for inspection and corridor management, even though they are not always "defects" requiring repair. Based on typical trail conditions and field observations, the list of these surface conditions and defect types, and a brief description of each is shown in Table 4-1.

Table 4-1: Defect/Surface Conditions Type

Surface Conditions Categories	Description
Pothole 	A bowl-shaped depression in the trail surface, typically caused by water infiltration and wear, poses a trip hazard or discomfort for pedestrians and jarring impact for cyclists and wheelchair users.
Longitudinal Crack 	A crack that runs parallel to the direction of the trail, often due to poor joint bonding or ground movement, can affect wheelchair or stroller stability. Moisture infiltration accelerates pavement deterioration.

Surface Conditions Categories	Description
Transverse Crack 	A crack running perpendicular to the trail path, commonly due to temperature changes or aging materials, leading to minor surface disruption. Particularly problematic for small-wheeled mobility devices, indicating a potential for future pothole formation.
Alligator Crack 	A series of interconnected cracks resembling alligator skin, typically indicating structural fatigue and requiring surface replacement. Rough surface impedes rolling mobility devices, high priority for maintenance.
Misalignment 	A vertical or horizontal shift between pavement sections, usually caused by tree roots or subsidence, creates a significant tripping hazard; impassable obstacle for wheelchairs when vertical offset exceeds 0.5 inches (ADA threshold); can occur in both flexible and rigid pavements.

Surface Conditions Categories	Description
Vegetation Obstacle 	Overgrown plants, bushes, or branches encroaching into the usable trail width below ADA minimum (60 inches). These conditions reduce effective width and sight distance creating safety concerns. Vegetation may be avoided without generating a strong vibration response; detection therefore relies on whether vegetation is visible in extracted frames and can be missed.
Pole Obstacle 	Fixed objects such as signposts, bollards, or utility poles placed close to or within the usable trail width. These may pose collision risks and constrain lateral clearance. Because riders typically steer around poles, the jerk-triggered system may under-sample them; they are retained as contextual features rather than defects.
Debris Obstacle 	Loose gravel, branches, excessive leaf accumulation, trash, or other foreign materials on the trail surface can cause slipping, falling, or impede movement along with rolling hazard for small wheels.

Surface Conditions Categories	Description
Tactile Pavement 	Textured surface features used at crossings or transitions to provide cues. This category is included as a contextual feature; its condition becomes relevant when the feature is damaged, misaligned, or creates a localized hazard.
Rail Crossing 	Locations where a trail intersects rail infrastructure. Condition issues become relevant when the crossing creates unevenness, gaps, or other hazards. Rail crossings are treated as contextual features, and only damaged or hazardous crossings warrant maintenance action.

Detection Methodology Considerations. The accelerometer-based jerk detection system (Section 3.2) is optimized for identifying surface defects and surface conditions that produce measurable vertical vibration, specifically potholes, cracks, misalignments, and rail crossings. Obstacles that are visually avoided without inducing significant vibration (vegetation obstacles, pole obstacles) may be under sampled in the automatically extracted frame set. While the accelerometer captures three-axis data that can detect lateral maneuvers when riders swerve around obstacles, vegetation obstacles, or gradual width reduction that requires only slight steering adjustments may not trigger jerk thresholds.

This limitation is acknowledged in the dataset composition (Section 4.2) where obstacle classes show lower instance counts, and motivates future enhancements such as periodic frame sampling independent of jerk events or continuous video-based scanning using temporal object tracking (Section 6.2).

4.2 Trail Surface Conditions and Defect Dataset Overview

Dataset Development and Composition. Using the Data Bike system described in Section 3.1, we collected video data across five distinct trails totaling 34 miles, capturing diverse environmental conditions documented in Table 3-1. Video frames were extracted using the jerk-triggered methodology (Section 3.2) and manually annotated into the ten predefined categories. The local dataset yielded 4,971 instances across all ten classes. While this represents comprehensive coverage of trail-specific features and environmental diversity, the instance counts for several critical defect categories were insufficient to train a robust deep learning model independently. For example, potholes, a safety-critical category, had only 30 instances, while alligator cracks had 80 instances.

Strategic Integration of External Data. To address this data scarcity while preserving trail-specific learning, we implemented a two-stage transfer learning strategy. Stage 1 involved augmenting our local dataset with the GRDD dataset, but only for the four structural defect classes that exhibit consistent visual morphology across both road and trail contexts: potholes, longitudinal cracks, transverse cracks, and alligator cracks. Table 4-2 summarizes the composition of our total dataset:

Table 4-2: Dataset Composition

Class	Local Data Bike	GRDD Dataset	Total
Pothole	30	6544	6574
Longitudinal Crack	517	26220	26737
Transverse Crack	2858	12923	15781
Alligator Crack	80	10630	10710
Misalignment	615	0	615
Vegetation Obstacle	147	0	147
Pole Obstacle	196	0	196
Debris Obstacle	78	0	78
Tactile Pavement	346	0	346
Rail Crossing	104	0	104
Overall Total	4971	56317	61288

Rationale for Cross-Domain Transfer:

To address this data scarcity, we augmented our local trail dataset with the GRDD dataset, but only for the four structural defect classes that exhibit consistent visual morphology across both road and trail contexts: potholes, longitudinal cracks, transverse cracks, and alligator cracks. This strategic decision was based on three key factors:

1. **Shared Defect Morphology:** The four structural defect types supplemented from GRDD exhibit fundamentally similar visual characteristics, whether they appear on vehicular

roadways or paved trails. These defects result from identical degradation mechanisms, including thermal expansion, water infiltration, and load stress, and display comparable visual patterns. Surface material similarity further supports this transferability: all surveyed trail segments feature asphalt or concrete paving, the exact materials represented in GRDD. While edge environments differ (roads have curbs and shoulders; trails border grass or vegetation), these contextual differences do not substantially alter the visual characteristics of the surface defects themselves.

2. **Surface Material Similarity**: All surveyed trail segments feature asphalt or concrete paving, the exact materials represented in GRDD (which spans road imagery from Japan, India, Czech Republic, Norway, and the United States). While edge environments differ (roads typically have curbs and shoulders; trails border grass, gravel, or natural vegetation), these contextual differences do not substantially alter the visual characteristics of the surface defects themselves, which remain the primary detection target. The model learns crack patterns and pothole geometry from GRDD's extensive examples, then applies this knowledge to trail surfaces during inference.

3. **Exclusive Trail-Specific Coverage**: Six of our ten classes--tactile pavement, rail crossings, vegetation obstacles, pole obstacles, debris obstacles, and misalignments--are present only in the local dataset, with 1486 annotations representing 30% of local data. These categories capture features entirely absent from vehicular road imagery, ensuring the model learns trail-specific contexts that GRDD cannot provide. This preserves the unique character of trail environments while leveraging external data for universal defect types.

Validation-Focused Local Data Strategy. The local dataset served dual purposes: (1) providing exclusive training examples for trail-specific features, and (2) functioning as validation and test sets to rigorously assess how well GRDD-learned features transfer to actual trail environments. The environmental diversity documented in Table 3-1, spanning lighting conditions (direct sun to overcast), surface materials (asphalt-dominant to concrete-dominant), canopy cover (open to dense shade), collection times (morning 8 AM through afternoon 4 PM), and land use contexts (urban formal sidewalk to recreational natural corridors), ensures that even our limited

local annotations expose the model to the full spectrum of real-world trail conditions during validation.

Our validation results (mAP@0.5 of 0.621, Section 4.5) confirm successful cross-domain transfer, demonstrating that structural defect representations learned from roads generalize effectively to trails despite differences in path width, edge characteristics, and surrounding vegetation context. This validates the strategic decision to leverage GRDD for scaling training data while preserving trail-specific learning through local examples and fine-tuning.

Final Dataset Distribution and Characteristics. As summarized in Table 4-2, the final dataset comprises 61,288 labeled instances across ten classes, formed by combining 4,971 locally annotated Data Bike instances with 56,317 instances from the GRDD dataset. External data were incorporated exclusively for four structural defect categories, potholes, longitudinal cracks, transverse cracks, and alligator cracks, while the remaining six classes are represented solely by local trail imagery.

Structural defects dominate the dataset numerically. Longitudinal cracks (26,737 instances) and transverse cracks (15,781 instances) account for the largest shares, followed by alligator cracks (10,710) and potholes (6,574). These counts reflect both the prevalence of cracking-related deterioration mechanisms and the targeted augmentation strategy using GRDD. In contrast, trail-specific categories—including misalignment (615), tactile pavement (346), vegetation obstacles (147), pole obstacles (196), debris obstacles (78), and rail crossings (104)—are intentionally limited to locally collected data, preserving the contextual uniqueness of trail environments.

For model development, the combined dataset was partitioned into GRDD + Local Dataset for stage 1 that is split into training and validation splits at the image level, and remaining local-only dataset is used for Fine tuning at stage two which has train, test and val splits. Detailed split-level counts are provided in Appendix B.

Dataset Quality and Spatial Characteristics. The correlogram from the labeled dataset illustrates the distribution and pairwise relationships of normalized bounding box attributes (x-center, y-center, width, and height). Most bounding boxes are concentrated near the center and

upper region of the images, as shown by the peak in y-center values around 0.6–0.8. The distributions of width and height are heavily skewed toward smaller values, indicating that most objects detected are relatively small. The triangular patterns in the scatter plots reflect the natural geometric constraint that boxes must remain within image boundaries. These insights confirm consistent and valid annotations and suggest that the model should be optimized to detect minor and centrally located defects, potentially benefiting from finer anchor boxes and higher-resolution input images.

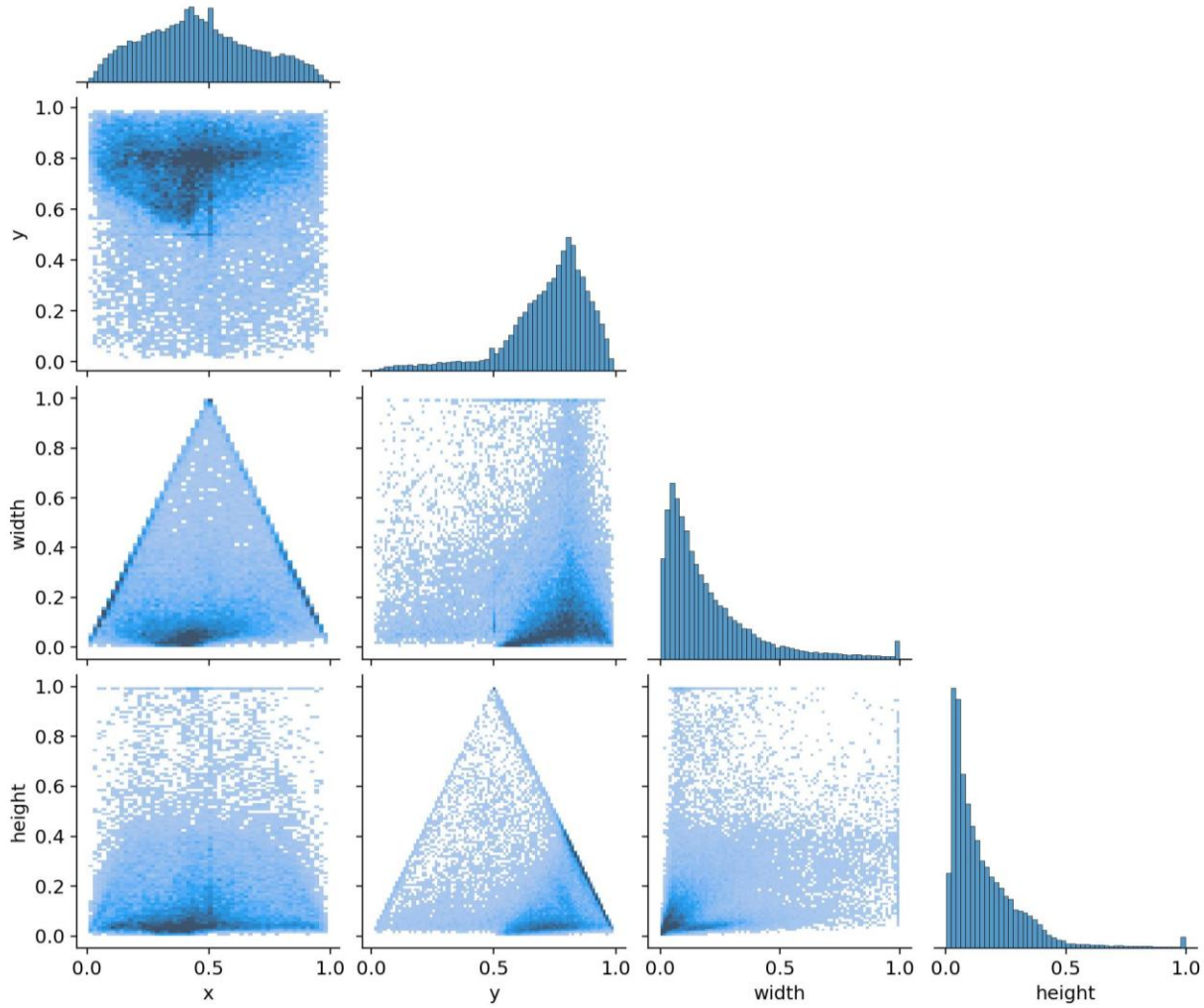


Figure 4-1: Correlogram showing distribution and pairwise relationships of normalized bounding box attribute of the labeled dataset

4.3 Spatial Distribution and Hotspot Identification Through Dashboard

The primary outcome of this research is the ability to translate discrete defect data points into a coherent trail-level understanding of trail conditions. This is achieved through the Dashboard, an interactive tool designed to visualize the geospatially tagged data collected by the Data Bike (accessible at https://databike-dashboard-demo.streamlit.app/). Through interactive visualization and filtering of georeferenced inspection results, the dashboard provides a clear, data-supported overview of defect and surface condition distribution across the analyzed trail network, enabling an understanding of spatial patterns in trail conditions.

The dashboard's spatial visualization capabilities are critical for identifying hotspot areas with a high concentration of defects that demand attention. Users can immediately observe distinct patterns by rendering each defect as an interactive point, color-coded by type. For example, dashboard analysis revealed a significant cluster of pavement cracking, specifically transverse cracks, along the Jordan River Parkway segment near high-traffic access points. In contrast, vegetation-related issues were more prevalent in less-trafficked, heavily shaded areas.

The operational utility of the dashboard lies in how it empowers planners and maintenance managers to interact with this data. Users can dynamically filter the map to display specific defect types or focus on a particular trail segment. By clicking on any point, they can bring up the associated source image and metadata to visually verify the issue's severity. This capability moves analysis beyond subjective assessments and allows for a direct, evidence-based evaluation of trail conditions from a desktop environment.

The identification of these hotspots is a critical finding. It provides objective evidence that maintenance needs are not uniformly distributed but concentrated in specific areas, likely influenced by user volume, environmental conditions, and pavement age. This insight enables a fundamental shift in maintenance strategy from a reactive, complaint-based model to a proactive, geographically targeted approach. Using the dashboard to locate and verify these hotspots, the agency can ground its maintenance planning in robust data, ensuring that resources are allocated efficiently and directed where they will have the most significant impact.

4.4 Impact on Accessibility and User Experience

The insights derived from the Data Bike project have significant implications for improving trail accessibility and overall user experience. The system's ability to reliably identify and locate barriers such as transverse crack and misalignments allows for their systematic maintenance. This enables proactive maintenance before issues escalate, ensuring a consistently high-quality and inclusive trail network that supports public safety and accessibility.

4.5 Model Performance Evaluation

This section evaluates the performance of the YOLOv8 object detection model trained to identify surface defects and contextual features in trail imagery. Performance is assessed using standard object-detection metrics that jointly capture classification accuracy and localization quality. The results are interpreted both statistically and operationally, with emphasis on suitability for assisted inspection workflows rather than fully autonomous decision-making.

Training Duration, Epochs, and Model Selection. Model development followed a two-stage training strategy, consisting of an initial foundation training phase followed by a subsequent fine-tuning phase on local trail imagery.

In the first stage, model training was conducted for up to 108 epochs, where an epoch denotes one complete pass through the training dataset. As is typical in deep learning, early epochs primarily learn low-level visual features (edges, textures), while later epochs refine higher-level, class-specific representations. Beyond a certain point, additional epochs often yield diminishing performance gains and may increase the risk of overfitting to the training data.

To mitigate this risk, early stopping was enabled with a patience of 20 epochs, meaning training was terminated if validation performance did not improve for 20 consecutive epochs. Under this criterion, training stopped at epoch 88, which corresponds to the checkpoint with the highest validation mAP. This checkpoint represents the best balance between learning capacity and generalization and was retained as the Stage 1 foundation model.

Following this initial training, a fine-tuning phase was performed using locally collected trail imagery. Fine-tuning was initialized from the Stage 1 checkpoint and conducted for a smaller

number of epochs using a reduced learning rate. This strategy allows the model to adapt to trail-specific visual characteristics, such as surface texture, camera viewpoint, lighting variability, and edge vegetation, while preserving the robust feature representations learned from the larger training corpus.

Importantly, fine-tuning does not replace the original training but refines it. The foundation model provides broad defect recognition capability, while the fine-tuning phase emphasizes domain adaptation and improves sensitivity to locally relevant defect patterns and contextual features. All final evaluation results reported in this section correspond to the fine-tuned model, selected based on validation performance during the fine-tuning stage.

Performance Metrics and Acceptability Criteria. The Model's performance is reported using the following commonly accepted object-detection metrics:

- **Precision**, measuring the proportion of detected instances that are correctly classified (i.e., resistance to false positives).
- **Recall**, measuring the proportion of ground-truth defects that are successfully detected (i.e., sensitivity).
- **mAP@0.5**, the mean Average Precision computed at an Intersection-over-Union (IoU) threshold of 0.5, which is widely used as a benchmark for "reasonable" localization accuracy in infrastructure inspection applications.
- **mAP@0.5:0.95**, a stricter metric averaging performance across IoU thresholds from 0.50 to 0.95, providing insight into bounding-box precision.

The final fine-tuned model achieved an overall **mAP@0.5 of 0.621** and **mAP@0.5:0.95 of 0.345** on the test set. In the context of visual infrastructure inspection, where approximate localization is sufficient to guide field crews to candidate locations, an mAP@0.5 above 0.60 is generally regarded as operationally acceptable for assisted inspection and prioritization (Arya et

al. (2021), Li et al. (2024)). The lower mAP@0.5:0.95 reflects the expected difficulty of achieving near-pixel-perfect localization for irregular and visually diffuse defects such as cracks.

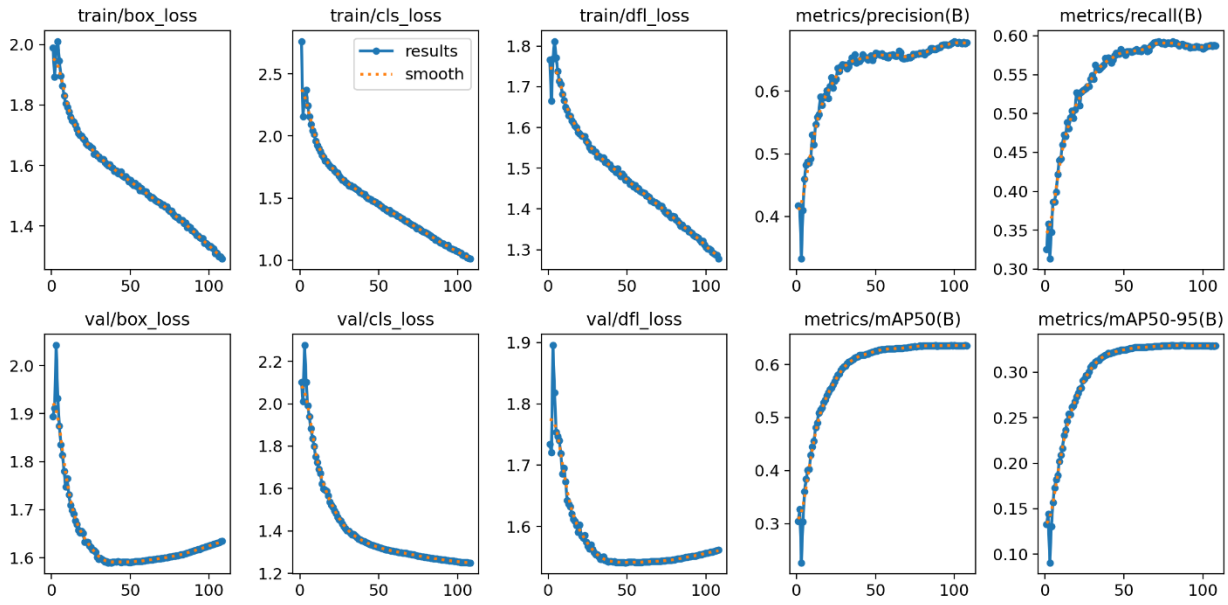


Figure 4-2: Training curves for model metrics and loss

Class-Level Performance and Implications of Class Imbalance. The Precision–Recall (PR) curves illustrate the trade-off between precision and recall for each defect class as the detection confidence threshold varies. These curves provide insight into how reliably each defect type can be detected across operating points. Among the evaluated classes, alligator cracking demonstrates the strongest and most stable precision–recall balance, achieving a high area under the curve relative to other structural defects. This reflects the distinctive interconnected crack patterns associated with alligator cracking, which are visually easier for the model to distinguish from background textures.

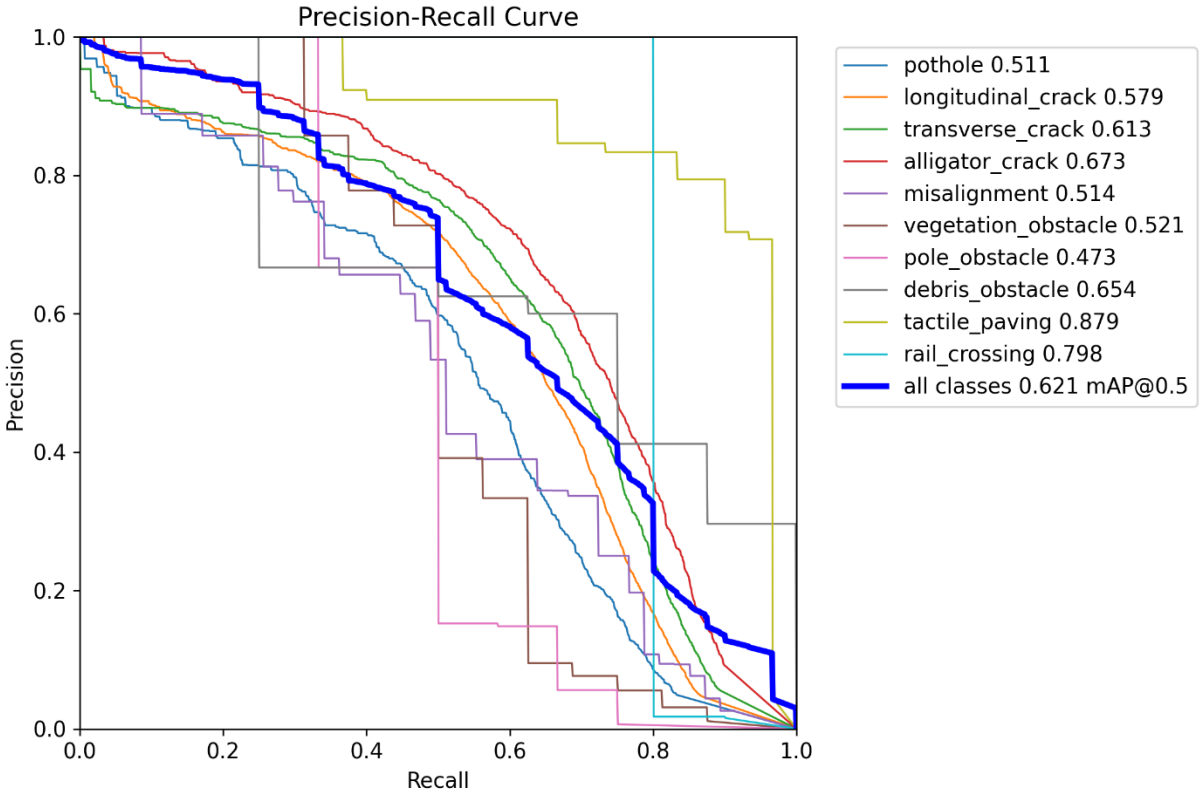


Figure 4-3: Precision-Recall curves for YOLOv8 model on various defect classes

Table 4-3 summarizes the class-specific performance of the fine-tuned YOLOv8 model across major structural defects and trail-specific features using precision, recall, mAP@0.5, and mAP@0.5:0.95 as evaluation metrics. Results are reported on a held-out test set.

Table 4-3: Classification Performance Metrics for Ten Categories

Class Name	Precision	Recall	mAP50	mAP 0.5:0.95	Literature Benchmark (Typical Range)
Pothole	0.608	0.494	0.511	0.224	mAP@0.5: ~0.60–0.65; mAP@0.5:0.95: ~0.30–0.40
Longitudinal Crack	0.657	0.547	0.609	0.319	
Transverse Crack	0.688	0.574	0.613	0.305	
Alligator Crack	0.625	0.672	0.673	0.356	
Misalignment	0.581	0.472	0.514	0.214	
Tactile Pavement	0.786	0.856	0.879	0.468	
Pole Obstacle	0.315	0.500	0.473	0.243	
Vegetation Obstacle	0.441	0.500	0.521	0.360	
Rail Crossing	0.798	0.800	0.798	0.601	
Debris Obstacle	0.551	0.314	0.654	0.360	
OVERALL	**0.605**	**0.573**	**0.621**	**0.345**	

Among the structural surface defects, alligator cracks achieved the highest overall performance, with a precision of 0.625, recall of 0.672, mAP@0.5 of 0.673, and mAP@0.5:0.95 of 0.356. These values indicate that the model detects this defect type accurately and consistently across moderate and stricter localization criteria. Transverse cracks and longitudinal cracks also demonstrate reliable detection performance, with mAP@0.5 values of 0.613 and 0.509, respectively. Some confusion between these two classes persists, which is expected given their similar linear appearance and shared visual features under varying lighting and pavement conditions.

In contrast, potholes exhibit the weakest performance among the structural defect categories, with a recall of 0.494 and an mAP@0.5:0.95 of 0.224. This indicates that while detected potholes are reasonably well classified when found (precision 0.608), a substantial fraction of potholes are missed. This outcome is consistent with the limited number of original pothole training examples and the high visual variability of potholes in terms of size, depth, surface material, and illumination. Oversampling during fine-tuning improves numerical balance but cannot fully compensate for insufficient visual diversity.

Trail-specific and contextual classes show a wider range of outcomes. Tactile pavement achieves excellent performance (mAP@0.5 = 0.879, recall = 0.856), reflecting both its distinctive texture pattern and consistent visual appearance. Rail crossings also perform strongly (mAP@0.5 = 0.798), despite a small number of validation instances, due to their highly structured and repeatable geometry. Other obstacle-related classes—such as vegetation and pole obstacles—show more variable results, largely attributable to limited validation samples and substantial appearance variability.

Overall, the model achieves an aggregate mAP@0.5 of 0.621 and mAP@0.5:0.95 of 0.345, indicating moderate to strong detection performance across the full set of evaluated classes. These values align closely with reported results in recent literature using YOLOv5 and YOLOv8-based architectures for road and pavement defect detection, where typical mAP@0.5 values range from approximately 0.60 to 0.65, and mAP@0.5:0.95 values range from 0.30 to 0.40. For example, RDD-YOLO reported an mAP@0.5 of 0.625 and mAP@0.5:0.95 of 0.364 on the RDD2022

dataset, while SEA-YOLOv8 achieved approximately 0.632 mAP@0.5 under similar evaluation protocols.

The mAP@0.5 metric evaluates performance when a detected object overlaps the ground truth by at least 50 percent, representing the proportion of correctly localized defects under a moderate matching criterion. The mAP@0.5:0.95 metric is stricter, averaging performance over multiple overlap thresholds (from 50 to 95 percent) and therefore rewarding models that localize defects precisely. A perfect model would achieve a value of 1.0 for all these metrics, but this is rarely attainable for real-world trail imagery due to illumination changes, surface texture variability, and small or subtle defect features.

Recent studies using advanced deep learning models such as YOLOv5 and YOLOv8 have shown strong performance in detecting cracks and defects on roads and trails. These models can correctly identify about 60–70% of defects when evaluated with a standard accuracy measure (called mAP@0.5), and around 30–40% when tested under stricter conditions (mAP@0.5:0.95), which is considered a strong result for human-supervised workflow. For example, one model called RDD-YOLO, tested on a large dataset named RDD2022, reached 62.5% accuracy on the easier test and 36.4% on the stricter one. Another model, SEA-YOLOv8, achieved about 63.2% accuracy under the same easier test.

In comparison, the Data Bike model's results (0.621 mAP@0.5 and 0.345 mAP@0.5:0.95) fall squarely within this benchmark range, confirming that the model performs at a strong, practically useful level given the dataset size and class imbalance.

Interpreted operationally, an mAP@0.5 $\approx$ 0.62 indicates that the model correctly detects and reasonably localizes approximately two-thirds of visible defects and surface conditions on average, which is consistent with current research-grade detection systems used for infrastructure inspection. The stricter mAP@0.5:0.95 $\approx$ 0.35 reflects the inherent difficulty of precisely delineating irregular and low-contrast defects, such as cracks and shallow surface failures, under real-world trail conditions that include illumination changes, surface texture variation, and partial occlusion.

From a practical standpoint, for UDOT and similar agencies, an mAP@0.5 above 0.6 can be considered a strong model suitable for assisting or partially automating visual inspections. Values below 0.4 typically indicate a need for more training data or model refinement. Given the current results, the Data Bike model demonstrates strong to moderate performance, successfully identifying many major trail defects, particularly cracking patterns and misalignments, while occasionally missing rare or subtle features, such as thin cracks and isolated obstacles. Continued collection of balanced trail imagery and additional fine-tuning are expected to further improve detection performance, aiming for the 0.6-0.7 mAP range, thereby enhancing the reliability of automated trail condition monitoring and supporting proactive maintenance planning.

The observed variation in class-level performance highlights the impact of class imbalance in the training data. Frequently occurring defects, such as cracks, benefit from extensive visual exposure and yield stable performance, while rare classes exhibit greater uncertainty and sensitivity to individual samples. This suggests that future improvements will be driven less by additional training iterations and more by targeted data collection for underrepresented defect types. As such, the current model is best suited for assisted inspection and prioritization, where automated detections guide human review rather than replace it.

Training Runtime and Computational Context. Model training was conducted using a single consumer-grade GPU (NVIDIA GeForce RTX 4060 Laptop GPU with 8 GB VRAM), demonstrating that the proposed workflow does not require specialized or dedicated high-performance computing resources. For both training and inference, all images were resized to a unified model input resolution of 640 × 640 pixels. Locally collected trail imagery was downscaled from its native resolution (3840 × 2160 pixels), while images from the Global Road Damage Detection (GRDD) dataset were upscaled from their native resolution (512 × 512 pixels). This unified preprocessing ensured consistent feature scaling across datasets while prioritizing spatial detail relevant to the target deployment environment.

The foundation training stage completed in approximately 29 hours, with early stopping triggered after 107 epochs based on validation performance. The subsequent fine-tuning stage using locally collected trail imagery required approximately 8 hours. These runtimes reflect one-time or infrequent model development activities rather than recurring operational requirements.

Inference runtime was evaluated separately to assess feasibility for routine deployment. Under this configuration, the deployed YOLOv8 model achieved an average inference time of approximately 22–29 ms per image, with preprocessing and postprocessing contributing an additional 2–6 ms per image. This corresponds to an effective throughput on the order of 30–40 images per second during GPU execution. Together, these computational characteristics indicate that model inference imposes a relatively small overhead compared to data handling, quality control, and downstream review of flagged locations, supporting the use of the system as a practical tool for routine trail condition screening and assisted inspection workflows. The details can be found in Appendix E.

4.6 Model Output Interpretation

The model output table (Table 4-4) provides a structured sample of surface defect detections aligned with high-jerk events recorded by the Data Bike. Each row includes the frame filename, jerk and GPS timestamps, geolocation (latitude and longitude), class ID of the detected defect, confidence score, and bounding box coordinates within the image. This detailed output enables precise spatial identification of cracks and other anomalies. The confidence score helps filter high-certainty detections, while the class ID distinguishes between defect types. Together, these attributes support targeted review and classification of trail surface conditions.

Beyond detection, this output can be transformed into GIS-compatible layers such as shapefiles or GeoJSON. These georeferenced points or bounding boxes can be mapped onto trail networks for spatial analysis, including defect clustering, severity mapping, and maintenance prioritization. Planners can identify high-risk segments and allocate resources more effectively by filtering detections by type, location, or confidence level. This integration bridges field data with actionable planning, laying the groundwork for a proactive and data-driven trail asset management strategy.

Table 4-4: Sample Output of Inference from Model

frame_filename	event_ index	jerk_ time_s	gps_ time_s	latitude	longitude	Class _id	confidence	x1	y1	x2	y2
event_001_1.jpg	1	56.89748	56.90899	40.74715	-111.922	2	0.563073	2646.217	1536.067	3840	1814.805
event_001_2.jpg	1	56.89748	56.90899	40.74715	-111.922	2	0.561456	2330.956	1716.539	3839.424	2159.722
event_001_3.jpg	1	56.89748	56.90899	40.74715	-111.922	2	0.597425	2575.29	1365.34	3836.41	1577.205
event_001_4.jpg	1	56.89748	56.90899	40.74715	-111.922	2	0.417843	2314.375	1517.054	3831.494	1838.529
event_001_5.jpg	1	56.89748	56.90899	40.74715	-111.922	2	0.538614	2523.721	1287.851	3722.908	1447.031
event_001_6.jpg	1	56.89748	56.90899	40.74715	-111.922	2	0.49459	1630.223	1697.83	3840	1967.201
event_002_1.jpg	2	135.3277	135.3084	40.74531	-111.921	2	0.520054	427.2671	1581.002	2877.534	1932.22
event_002_2.jpg	2	135.3277	135.3084	40.74531	-111.921	2	0.694593	753.8535	1413.8	2748.005	1702.779
event_004_1.jpg	4	204.658	204.6139	40.74406	-111.919	2	0.417939	794.8125	1607.165	2712.264	1756.751
event_005_1.jpg	5	304.1218	304.1107	40.74168	-111.92	2	0.330672	698.7416	1496.878	2271.515	1874.907

4.7 Dashboard Interpretation

To facilitate exploration of defect detection outputs, an interactive dashboard was deployed and made publicly accessible. Users can access the live demo at https://databike-dashboard-demo.streamlit.app/. The interface allows uploading pre- and post-maintenance GPS route files and corresponding defect CSVs, selecting relevant columns, and specifying the image directory. The tool then generates side-by-side maps, annotated defect images, and location-specific defect summaries, enabling straightforward evaluation of maintenance impacts on trail conditions. A README file and a demonstration video are provided (linked here: https://github.com/hereasmadhu/data_bike) for more detailed usage instructions to guide users through dashboard access and features, offering additional insight into its functionality and applications.

The dashboard's core is an interactive map that serves as the primary user interface. Location data from the structured CSV file is ingested and plotted as individual points, each representing a detected trail defect. This integration allows for a one-glance overview of the entire trail network. Key features of the dashboard include:

- **Interactive Map Display:** Geospatial points for all detected defects and surface conditions are displayed on a high-resolution map. Each point is color-coded by defect category to provide an immediate visual summary of issue types.

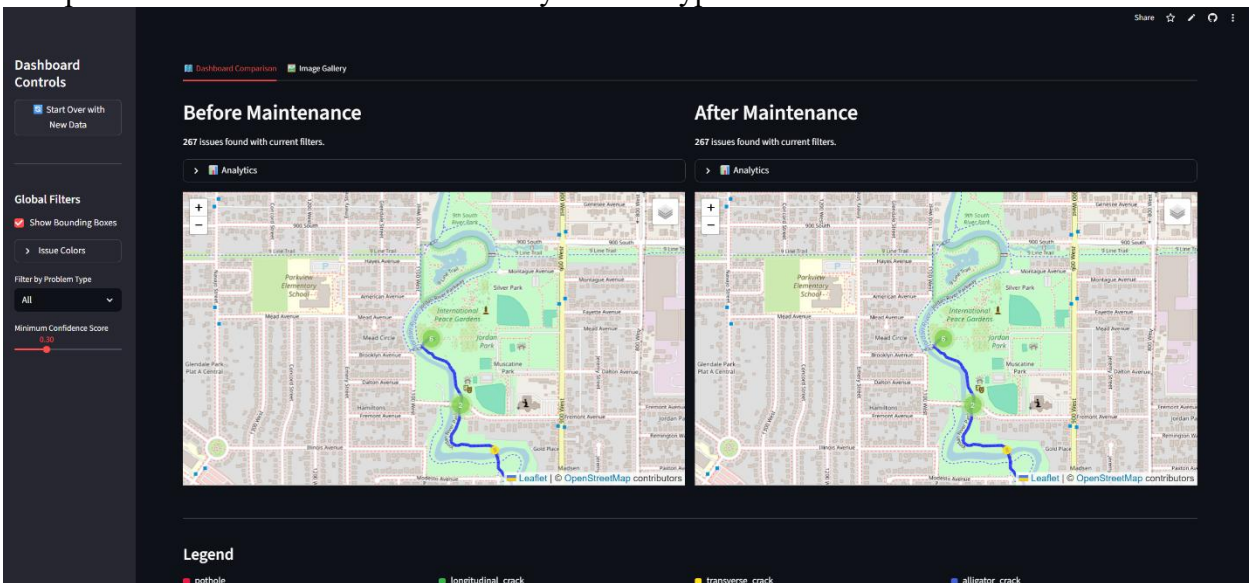


Figure 4-4: Interactive Map with Hotspots Identified

- **On-Demand Detailed Information:** Clicking on any point on the map opens a new window that displays the complete metadata for that specific defect, including the confidence score, timestamp, and, most importantly, the source image captured by the Data Bike. This feature enables instant visual verification of the defect.

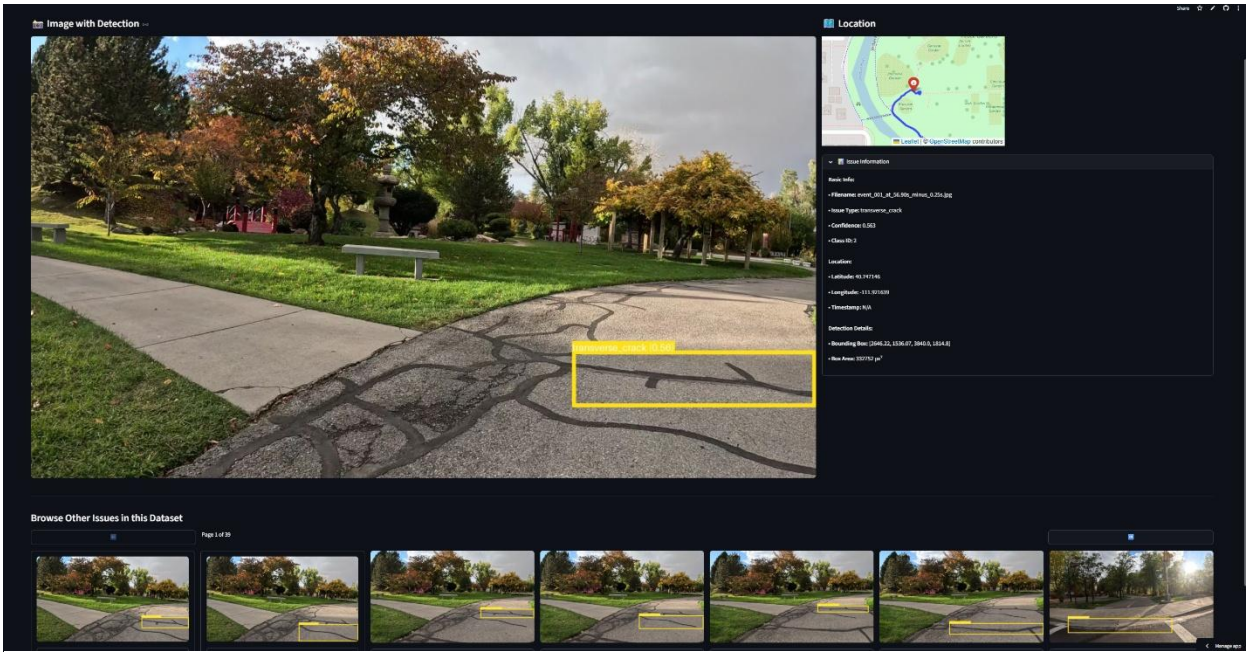


Figure 4-5: Detailed view of defect

- **Dynamic Filtering and Queries:** The dashboard has interactive filters that allow users to query the data dynamically. Stakeholders can filter the view by surface condition type and confidence score range. For example, maintenance

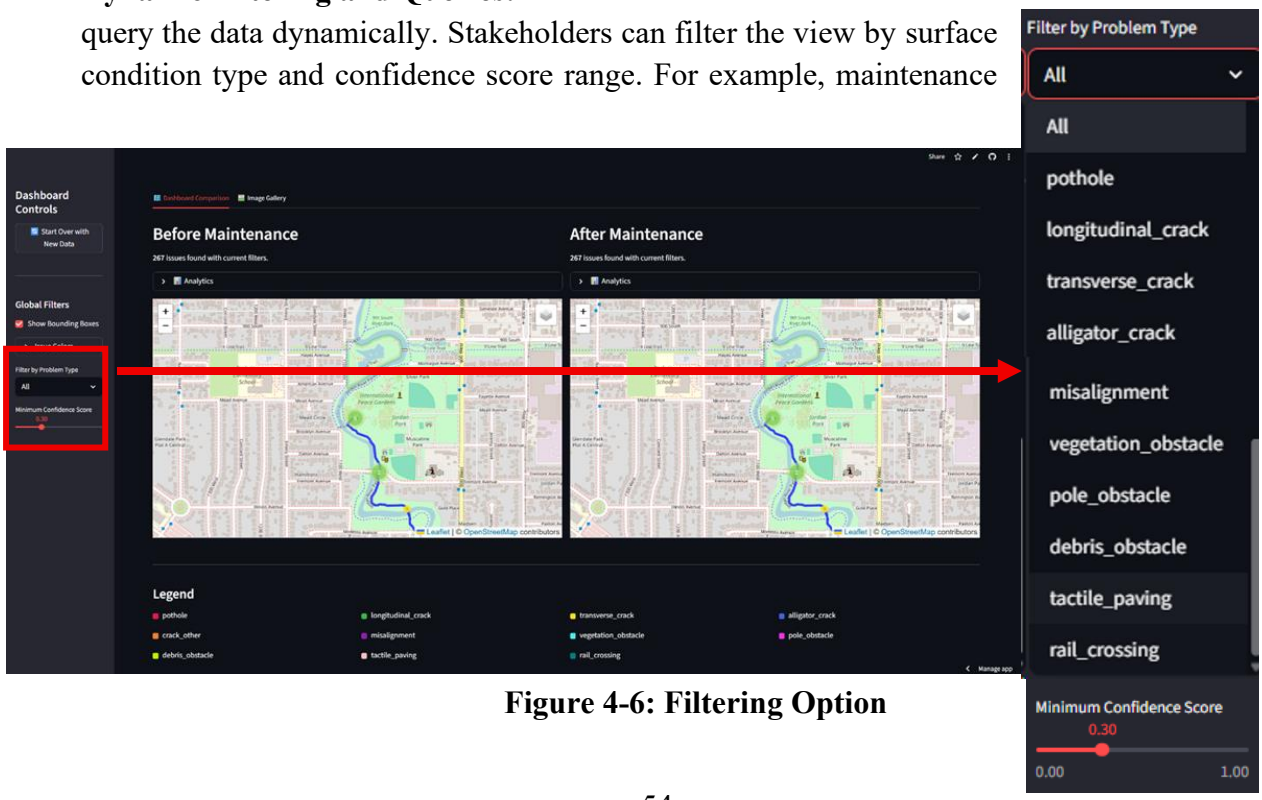


Figure 4-6: Filtering Option

teams can generate a map showing only the transverse cracks detected in the trail.

- **Summary Analytics and Charts:** The dashboard includes summary statistics and charts that update in real-time as the map is filtered. These widgets provide an aggregated overview, displaying metrics such as the total number of defects and surface conditions, the distribution of defect categories, and a bar chart illustrating defect counts by trail

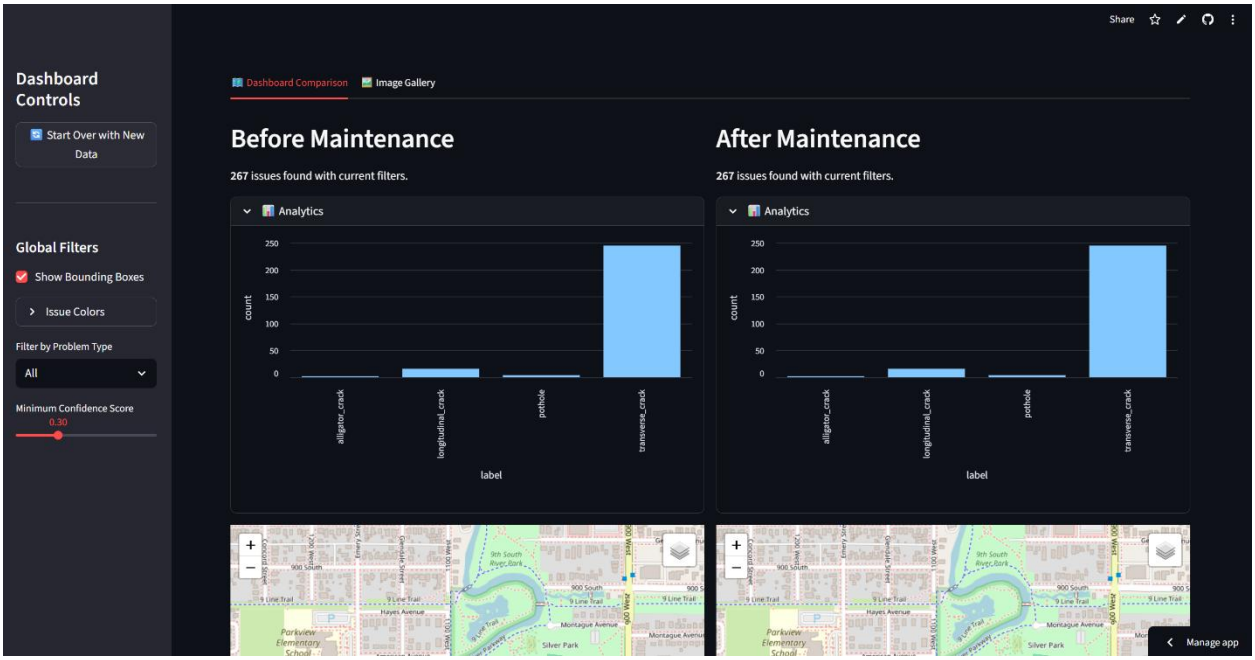


Figure 4-7: Analytics Summary Chart

section.

By integrating geospatial data directly into this intuitive dashboard, we provide a holistic and easily accessible tool for monitoring trail health. This system enhances the transparency and accessibility of inspection data, empowering UDOT to move from a reactive to a proactive and data-driven maintenance strategy.

5.0 STRATEGIC IMPLICATIONS AND ACTION PLAN FOR UDOT

The successful demonstration of the low-cost, automated trail defect system presents a strategic turning point for UDOT's maintenance of active transportation infrastructure. What began as a research prototype has evolved into a viable framework capable of transforming how UDOT maintains, prioritizes, and invests in its trail network. The system moves trail management beyond the constraints of sporadic manual inspections and citizen complaints, providing instead a continuous, objective, and data-driven assessment of infrastructure conditions. This shift enhances efficiency and accountability, creating opportunities to align maintenance operations with UDOT's broader policy goals of safety, equity, and sustainability.

The following subsections outline the strategic implications of this transition and provide a clear action plan for UDOT to operationalize the findings. UDOT can leverage this research to build a proactive and predictive asset management framework that ensures Utah's trails remain safe, equitable, and resilient by combining immediate pilot testing, structured data collection, database development, workforce training, and phased system integration.

5.1 Actionable Recommendations for Maintenance and Capital Improvements

The most direct application of this system lies in reshaping how UDOT prioritizes maintenance and capital improvement decisions. The outputs of the Data Bike, detailed geospatial records of defect type, location, and frequency, allow maintenance planning to be guided by measurable infrastructure needs rather than ad hoc requests. This ensures that limited resources are allocated to the most critical repairs first, maximizing return on investment and extending the service life of trail assets. Over time, the accumulation of longitudinal data will also allow UDOT to track the rate of deterioration, identify corridors most at risk, and evaluate the effectiveness of past treatments. Ultimately, this creates a feedback loop where real-world evidence continuously informs capital planning, enabling UDOT to allocate funds strategically and transparently.

5.2 A Framework for Implementation and System Integration

Operationalizing the system requires a structured approach that embeds the technology into UDOT's existing workflows. The first step is to integrate the project outputs, CSV files of defect

locations, GPS coordinates, and GIS-compatible shapefiles, into UDOT's enterprise GIS and asset management platforms, like UPLAN currently being used by UDOT. Once integrated, these datasets can be viewed alongside other infrastructure records, allowing decision-makers to visualize trail conditions in the same framework used for roads, bridges, and other assets.

A periodic re-survey program can be established on a two-year cycle to ensure the database remains current and valuable for monitoring infrastructure health over time. A biennial program offers a practical balance between (i) the pace at which many trail surface conditions meaningfully evolve, (ii) the resource requirements of recurring corridor coverage, and (iii) the value of maintaining comparable observations over time for trend analysis. This interval is intended as a baseline recommendation rather than a fixed rule: higher-priority corridors (e.g., high-use commuter paths, segments with recurring issues, or corridors scheduled for construction) can be surveyed more frequently, while lower-priority segments can be updated on a longer cycle as resources allow. As repeated observations accumulate, this will allow UDOT to develop predictive models of pavement degradation, providing foresight into future maintenance needs. Staff training will make the system accessible and usable across divisions. Training modules should cover the technical aspects of dashboard navigation and the interpretation of results for practical maintenance decisions. To accelerate system-wide adoption, resources such as a user guide, dashboard, and demonstration video (linked here: https://github.com/hereasmadhu/data_bike) can serve as foundational tools for workforce capacity building.

Finally, UDOT can consider expanding data collection beyond the pilot corridors. This will build a truly comprehensive statewide inventory of trail conditions while also enhancing the robustness of the computer vision model by exposing it to a wider range of diverse surfaces and environmental conditions. Partnerships with local municipalities, advocacy groups, and universities can reduce costs and foster a collaborative ecosystem for data sharing and model refinement.

5.3 Strategic Outcomes and Policy Implications

Creating a centralized digital trail inventory of trail surface conditions represents one of the most valuable strategic outcomes for UDOT. When updated through periodic and consistent

data collection, this inventory can function as a condition-screening layer that complements existing asset management systems and can be integrated with current GIS and maintenance management platforms. Rather than replacing established systems, the inventory provides a standardized and repeatable source of spatially referenced condition information that supports network-level assessment of trail infrastructure. As the dataset is refreshed over time, it will enable longitudinal analysis, allowing UDOT to track deterioration rates, compare the performance of different repair strategies, and develop predictive models that forecast future conditions. Significantly, such a database also enhances transparency by offering a straightforward, data-backed narrative of infrastructure needs that can be communicated to policymakers and the public to justify funding decisions.

Beyond efficiency and planning considerations, the system has implications for consistent and comprehensive infrastructure management. By relying on systematic data collection rather than primarily on complaint-driven inputs, the approach reduces the likelihood that maintenance attention is disproportionately directed toward locations with higher reporting activity. As a result, trail segments serving areas with lower visibility or fewer reported issues are assessed using the same criteria as all other corridors. In this way, the system supports statewide goals for consistent service quality and broader access to safe, well-maintained active transportation facilities.

5.4 A Phased Three-Year Action Plan

To translate these strategic implications into tangible results, a phased action plan is recommended, designed to systematically build from a focused pilot to a fully integrated, statewide management framework:

Phase 1 – Pilot and Integration (Months 1–12): Deploy the Data Bike on several high-priority trail segments (100-150 miles) to validate data collection and processing workflows, train operational staff, and refine technical integration with UDOT's existing asset management systems. The focus will be on establishing a reliable operational baseline and building internal capacity. This pilot is estimated to require approximately 80 total person-hours per 100 miles: 32 for field data collection and 48 hours for data extraction, processing, model execution, and finding generation.

Phase 2 – Network Expansion and Database Development (Months 13–24): Scale data collection to cover the majority of UDOT's managed trails miles. Formalize and populate a centralized, cloud-hosted trail condition database, and expand data-sharing partnerships. A parallel key initiative will be the development and calibration of an automated defect severity scoring system to enable objective, data-driven maintenance prioritization.

Phase 3 – Predictive Management and Statewide Integration (Months 25–36+): Establish a recurring survey cycle for full network coverage. Leverage accumulated longitudinal data to build predictive deterioration models for forecasting and lifecycle planning. Fully embed the condition scoring system and data outputs into UDOT's formal capital budgeting and long-range planning processes, transitioning trail management to a proactive, data-driven framework.

Resources and Strategies. Staffing needs are projected to scale from 1.0 FTE in Phase 1 to 1.5 FTEs for full operations during Phase 3. While an initial investment in technology and model training is required, the Data Bike system offers superior efficiency and objectivity over manual inspections. It provides comprehensive, repeatable, and quantifiable network-level data at a speed and consistency manual methods cannot match. The annual cost for cloud hosting, dashboard maintenance, and model refinement is estimated to be modest (20-30 person-hours), delivering a strong return through optimized lifecycle management, defensible budgeting, and the prevention of costly rehabilitation through timely, data-informed maintenance.

6.0 CONCLUSIONS

6.1 Summary of Key Findings and Contributions

This study successfully demonstrates the feasibility of a low-cost, scalable automated trail defect detection system using a custom-built Data Bike. By integrating accelerometer data with computer vision, the project has established a cost-effective and efficient workflow for monitoring the condition of active transportation networks. The core achievement of this research effort is the development of a complete data pipeline from collection with a camera-equipped bicycle to the automated identification and geospatial mapping of surface conditions and defects using a fine-tuned YOLOv8 model.

The research confirmed that accelerometer-based "jerk" detection is an effective method for pre-screening and locating potential surface anomalies, significantly reducing the amount of data requiring manual or computational review. Though still undergoing refinement, the resulting computer vision model shows promising performance in identifying critical defects like cracks and potholes, achieving an overall mAP50 of 0.621. The model performed exceptionally well in detecting different cracks, a key indicator of pavement structural problems.

A key challenge identified was the scarcity of annotated, trail-specific training data. This was mitigated by augmenting our field-collected imagery with a larger, public road damage dataset, improving the model's foundational learning. However, the results also underscore the need for continued data collection across a wider variety of Utah's trails to enhance the model's accuracy and ability to generalize to unique local conditions.

A key outcome of this research is the development of an interactive dashboard that visualizes geolocated defects and surface conditions on an intuitive map interface. This tool enables stakeholders to explore trail conditions, verify defect images, filter results by type or severity, and generate real-time summary statistics. By providing a transparent and user-friendly platform, the dashboard bridges the technical outputs of the study with the practical needs of UDOT staff, planners, and decision-makers.

Most importantly, the project has also outlined a clear action plan for UDOT to translate these research findings into practice. The plan emphasizes a phased approach beginning with pilot deployment and staff training, expanding to statewide data collection and centralized database development, and ultimately achieving predictive, data-driven maintenance planning. The Data Bike system, GIS-based dashboard, and phased action plan provide UDOT with an immediately actionable framework to move from reactive, complaint-driven maintenance toward proactive, objective, and equitable asset management.

This project has laid a strong foundation for a data-driven asset management framework for a trail network in Utah. The methodology developed here has the potential to transform trail maintenance from a reactive, often subjective process into a proactive, objective, and comprehensive strategy. By enabling agencies to identify and prioritize repairs systematically, this work directly contributes to creating safer, more accessible, and more enjoyable active transportation infrastructure for all users, especially those with limited mobility.

6.2 Limitations and Avenues for Future Research

While this project has successfully demonstrated a structured framework for trail-condition data collection and automated screening, it is essential to acknowledge key limitations and identify practical avenues for strengthening the system. The model's performance is influenced by data coverage and environmental variability. In particular, extreme lighting (e.g., harsh glare, deep shadows) and weather-related conditions (e.g., wet surfaces, snow cover) can reduce visual contrast and affect detection reliability. In addition, the model's accuracy for less frequent classes remains more variable than for commonly observed cracking patterns. Underrepresented categories, including certain obstacle types and rare surface conditions, would benefit from additional labeled examples collected across a wider range of corridors, surface materials, and seasonal conditions.

A second limitation is that the current pipeline prioritizes frames using vibration-based triggering, which improves efficiency but can introduce sampling bias. Some visually apparent conditions may not consistently generate strong jerk signals (e.g., thin cracks, gradual surface wear, or low-profile misalignments), particularly when riding speed or rider behavior varies.

Future work should evaluate hybrid sampling strategies that combine jerk-triggered extraction with periodic or stratified frame sampling to reduce the likelihood that relevant conditions will be missed.

A key area for future enhancement is the development of a quantitative indicator for defect severity. The current system classifies defect type and provides approximate location, but it does not estimate severity (e.g., crack width, pothole depth, or extent). Future research can pursue two complementary paths to address this:

- **Correlating Jerk Magnitude with Severity:** A targeted validation study could examine whether jerk amplitude and event duration correlate with defect dimensions measured in the field. If consistent relationships are observed, jerk statistics could serve as a simple sensor-based proxy for severity, supporting more informative prioritization.
- **Advancing the Vision Model Toward Extent Estimation:** The current YOLOv8 detection framework can be extended to more detailed outputs. Future work should evaluate segmentation-based approaches that estimate defect extent more directly, enabling computation of geometric measures such as crack length, affected area, and patch size. These quantities would support more repeatable severity screening and improve alignment with maintenance decision criteria.

Finally, the full value of this system is expected to emerge through longitudinal use rather than single-pass inventories. Repeated surveys on a consistent schedule would allow UDOT to track defect and condition progression, evaluate before/after changes following maintenance treatments, and support corridor-level trend analysis. Establishing these capabilities will require consistent field protocols (e.g., equipment setup, speed ranges, and seasonal coverage) and standardized postprocessing to ensure that results are comparable across time.

7.0 REFERENCES

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., & Sekimoto, Y. (2021). RDD2020: An annotated image dataset for automatic road damage detection using deep learning. Data in Brief, 36, 107133.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., & Sekimoto, Y. (2022). RDD2022: A multi-national image dataset for automatic Road Damage Detection. arXiv preprint arXiv:2209.08538.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2021). Deep learning-based road damage detection and classification for multiple countries. Automation in Construction, 132, 103935.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Omata, H., Kashiyama, T., & Sekimoto, Y. (2020). Global road damage detection: State-of-the-art solutions. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 5533–5539). IEEE.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Omata, H., Kashiyama, T., & Sekimoto, Y. (2022). Crowdsensing-based Road Damage Detection Challenge (CRDDC'2022). In 2022 IEEE International Conference on Big Data (Big Data) (pp. 6378–6386). IEEE.

Ashraf, A., et al. (2023). Machine learning-based pavement crack detection, classification, and characterization: A review. Bulletin of Electrical Engineering and Informatics, 12(6), 3601–3619. https://doi.org/10.11591/eei.v12i6.5345

Bayar, G., & Bilir, T. (2019). A novel study for the estimation of crack propagation in concrete using machine learning algorithms. Construction and Building Materials, 215, 670–685. https://doi.org/10.1016/j.conbuildmat.2019.04.227

Bhat, S., et al. (2020). A survey on road crack detection techniques. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE) (pp. 1–6). IEEE. https://doi.org/10.1109/ic-etite47903.2020.67

Bil, M., Andrasik, R., & Kubecek, J. (2015). How comfortable are your cycling tracks? A new method for objective bicycle vibration measurement. Transportation Research Part C: Emerging Technologies, 56, 415–425. https://doi.org/10.1016/j.trc.2015.05.007

Boschmann, E. E., & Brady, S. A. (2013). Travel behaviors, sustainable mobility, and transit-oriented developments: A travel counts analysis of older adults in the Denver, Colorado metropolitan area. Journal of Transport Geography, 33, 1–11. https://doi.org/10.1016/j.jtrangeo.2013.09.001

Cafiso, S., Di Graziano, A., et al. (2022). Urban road pavements monitoring and assessment using bike and e-scooter as probe vehicles. Case Studies in Construction Materials, 16, e00889. https://doi.org/10.1016/j.cscm.2022.e00889

Cafiso, S., Pappalardo, G., & Stamatiadis, N. (2021). Observed risk and user perception of road infrastructure safety assessment for cycling mobility. Infrastructures, 6(11), 154. https://doi.org/10.3390/infrastructures6110154

Chakurkar, P. S., et al. (2023). Data-driven approach for AI-based crack detection: Techniques, challenges, and future scope. Frontiers in Sustainable Cities, 5. https://doi.org/10.3389/frsc.2023.1253627

Chen, Y., et al. (2021). Surface defect detection methods for industrial products: A review. Applied Sciences, 11(16), 7657. https://doi.org/10.3390/app11167657

Clifton, K. J., Livi Smith, A. D., & Rodriguez, D. (2007). The development and testing of an audit for the pedestrian environment. Landscape and Urban Planning, 80(1–2), 95–110. https://doi.org/10.1016/j.landurbplan.2006.06.008

Cohen, R., Fernie, G., & Roshan Fekr, A. (2020). A vision-based approach for sidewalk and walkway trip hazards assessment. International Journal of Environmental Research and Public Health, 17(22), 8438. https://doi.org/10.3390/ijerph17228438

Cord, A., & Chambon, S. (2012). Automatic road defect detection by textural pattern recognition based on AdaBoost. Computer-Aided Civil and Infrastructure Engineering, 27(4), 244–259. https://doi.org/10.1111/j.1467-8667.2011.00736.x

Ding, L., & Goshtasby, A. (2001). On the Canny edge detector. Pattern Recognition, 34(3), 721–725. https://doi.org/10.1016/s0031-3203(00)00023-6

Farhadmanesh, M., Rashidi, A., Subedi, A. K., & Marković, N. (2024). A computer vision-based standalone system for automated operational data collection at non-towered airports. IEEE Access: Practical Innovations, Open Solutions, 12, 95245–95264. doi:10.1109/access.2024.3425572

Gao, J., et al. (2018). Evaluating the cycling comfort on urban roads based on cyclists' perception of vibration. Journal of Cleaner Production, 192, 531–541. https://doi.org/10.1016/j.jclepro.2018.04.275

Gogola, M. (2020). Analysing the vibration of bicycles on various road surfaces in the city of Žilina. The Archives of Automotive Engineering – Archiwum Motoryzacji, 88(2), 77–97. https://doi.org/10.14669/am.vol88.art6

Governors Highway Safety Association. (2025, March). Pedestrian traffic fatalities by state: 2021 preliminary data (January–December). PDF report.

Han, H., et al. (2021). An advanced Otsu method integrated with edge detection and decision tree for crack detection in highway transportation infrastructure. Advances in Materials Science and Engineering, 2021(1). https://doi.org/10.1155/2021/9205509

Hassan, S., et al. (2022). Detecting patches on road pavement images acquired with 3D laser sensors using object detection and deep learning. In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. SCITEPRESS. https://doi.org/10.5220/0010830000003124

Ho, C.-H., & Ren, K. (2024). Vibration data mining and machine learning for anomaly detection of cycling trails using instrumented bike. In 2024 9th International Conference on Big Data Analytics (ICBDA) (pp. 123–127). IEEE. https://doi.org/10.1109/icbda61153.2024.10607239

Hoang, N.-D., & Nguyen, Q.-L. (2018). A novel method for asphalt pavement crack classification based on image processing and machine learning. Engineering with Computers, 35(2), 487–498. https://doi.org/10.1007/s00366-018-0611-9

Hölzel, C., Höchtl, F., & Senner, V. (2012). Cycling comfort on different road surfaces. Procedia Engineering, 34, 479–484. https://doi.org/10.1016/j.proeng.2012.04.082

Hu, G. X., et al. (2021). Pavement crack detection method based on deep learning models. Wireless Communications and Mobile Computing, 2021(1). https://doi.org/10.1155/2021/5573590

Jiang, J., et al. (2025). Investigation of cracking behavior in asphalt pavement using digital image processing technology. Frontiers in Built Environment, 11. https://doi.org/10.3389/fbuil.2025.1580379

Jie, G., & Liu, N. (2012). An improved adaptive threshold Canny edge detection algorithm. In 2012 International Conference on Computer Science and Electronics Engineering (pp. 164–168). IEEE. https://doi.org/10.1109/iccsee.2012.154

Koh, C. Y., Ali, M., & Hendawi, A. (2024). CrackLens: Automated sidewalk crack detection and segmentation. IEEE Transactions on Artificial Intelligence, 5(11), 5418–5430. https://doi.org/10.1109/tai.2024.3435608

Lee, T., Chun, C., & Ryu, S.-K. (2021). Detection of road-surface anomalies using a smartphone camera and accelerometer. Sensors, 21(2), 561. https://doi.org/10.3390/s21020561

Li, G., et al. (2020). Automatic crack recognition for concrete bridges using a fully convolutional neural network and naive Bayes data fusion based on a visual detection system. Measurement Science and Technology, 31(7), 075403. https://doi.org/10.1088/1361-6501/ab79c8

Li, Y., et al. (2022). A novel approach for UAV image crack detection. Sensors, 22(9), 3305. https://doi.org/10.3390/s22093305

Li, Y., Yin, C., Lei, Y., Zhang, J., & Yan, Y. (2024). RDD-YOLO: Road Damage Detection Algorithm Based on Improved You Only Look Once Version 8. *Applied Sciences*, *14*(8), 3360. https://doi.org/10.3390/app14083360

Lindsey, G., et al. (2008). Urban greenways, trail characteristics and trail use: Implications for design. Journal of Urban Design, 13(1), 53–79. https://doi.org/10.1080/13574800701804033

Liu, Y., et al. (2019). DeepCrack: A deep hierarchical feature learning architecture for crack segmentation. Neurocomputing, 338, 139–153. https://doi.org/10.1016/j.neucom.2019.01.036

Low, P. S., & Krisp, J. M. (2024). Smoothing the ride: A surface roughness-centric approach to bicycle routing. AGILE: GIScience Series, 5, 1–6. https://doi.org/10.5194/agile-giss-5-39-2024

Matarneh, S., et al. (2024). Evaluation and optimisation of pre-trained CNN models for asphalt pavement crack detection and classification. Automation in Construction, 160, 105297. https://doi.org/10.1016/j.autcon.2024.105297

Niska, A., et al. (2024). Determination of riding comfort on cycleways using a smartphone application. Journal of Traffic and Transportation Engineering (English Edition), 11(4), 747–760. https://doi.org/10.1016/j.jtte.2023.05.010

Olieman, M., Marin-Perianu, R., & Marin-Perianu, M. (2012). Measurement of dynamic comfort in cycling using wireless acceleration sensors. Procedia Engineering, 34, 568–573. https://doi.org/10.1016/j.proeng.2012.04.097

Pavlidis, T., & Liow, Y.-T. (1990). Integrating region growing and edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(3), 225–233. https://doi.org/10.1109/34.49050

Piironen, T., et al. (1990). Automated visual inspection of rolled metal surfaces. Machine Vision and Applications, 3(4), 247–254. https://doi.org/10.1007/bf01211850

Qu, Z., Li, Y., & Zhou, Q. (2022). CrackT-net: A method of convolutional neural network and transformer for crack segmentation. Journal of Electronic Imaging, 31(02). https://doi.org/10.1117/1.jei.31.2.023040

Quan, Y., et al. (2019). The method of the road surface crack detection by the improved Otsu threshold. In 2019 IEEE International Conference on Mechatronics and Automation (ICMA) (pp. 1615–1620). IEEE. https://doi.org/10.1109/icma.2019.8816422

Rizelioğlu, M., et al. (2024). Using a bike as a probe vehicle: Experimental study to determine road roughness with piezoelectric sensors. Journal of Infrastructure Systems, 30(3). https://doi.org/10.1061/jitse4.iseng-2442

Rizelioğlu, M., & Yazıcı, M. (2023). New approach to determining the roughness of bicycle roads. Transportation Research Record: Journal of the Transportation Research Board, 2678(1), 781–793. https://doi.org/10.1177/03611981231172753

Rong, W., et al. (2014). An improved Canny edge detection algorithm. In 2014 IEEE International Conference on Mechatronics and Automation (pp. 577–582). IEEE. https://doi.org/10.1109/icma.2014.6885761

Shi, Y., et al. (2016). Automatic road crack detection using random structured forests. IEEE Transactions on Intelligent Transportation Systems, 17(12), 3434–3445. https://doi.org/10.1109/tits.2016.2552248

Shneier, M. (1983). Using pyramids to define local thresholds for blob detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(3), 345–349. https://doi.org/10.1109/tpami.1983.4767397

Subedi, A. K., Rashidi, A., & Marković, N. (2025). Assessing roadside safety with computer vision: FHWA ratings as the key predictor of rural Road Departure crashes and severity. Journal of Advanced Transportation, 2025(1). doi:10.1155/atr/5559576

Tang, J., et al. (2024). Enhancing road crack detection accuracy with BsS-YOLO: Optimizing feature fusion and attention mechanisms. https://doi.org/10.48550/ARXIV.2412.10902

Tang, P., et al. (2019). Deep FisherNet for image classification. IEEE Transactions on Neural Networks and Learning Systems, 30(7), 2244–2250. https://doi.org/10.1109/tnnls.2018.2874657

Wage, O., et al. (2020). Ride vibrations: Towards comfort-based bicycle navigation. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLIII-B4-2020, 367–373. https://doi.org/10.5194/isprs-archives-xliii-b4-2020-367-2020

Wang, L., Zhuang, L., & Zhang, Z. (2019). Automatic detection of rail surface cracks with a superpixel-based data-driven framework. Journal of Computing in Civil Engineering, 33(1). https://doi.org/10.1061/(asce)cp.1943-5487.0000799

Xie, N., et al. (2019). Measurement of dynamic vibration in cycling using portable terminal measurement system. IET Intelligent Transport Systems, 13(3), 469–474. https://doi.org/10.1049/iet-its.2018.5181

Yang, F., et al. (2020). Feature pyramid and hierarchical boosting network for pavement crack detection. IEEE Transactions on Intelligent Transportation Systems, 21(4), 1525–1535. https://doi.org/10.1109/tits.2019.2910595

Yi, Y., Chen, Y., et al. (2020). Joint feature representation and classification via adaptive graph semi-supervised nonnegative matrix factorization. Signal Processing: Image Communication, 89, 115984. https://doi.org/10.1016/j.image.2020.115984

Yi, Y., Wang, J., et al. (2019). Joint graph optimization and projection learning for dimensionality reduction. Pattern Recognition, 92, 258–273. https://doi.org/10.1016/j.patcog.2019.03.024

Zhao Y, Shi B, Duan X, Zhu W, Ren L, et al. (2025) Research on road surface damage detection based on SEA-YOLO v8. PLOS ONE 20(6): e0324439. https://doi.org/10.1371/journal.pone.0324439

Zang, K., et al. (2018). Assessing and mapping of road surface roughness based on GPS and accelerometer sensors on bicycle-mounted smartphones. Sensors, 18(3), 914. https://doi.org/10.3390/s18030914

Zhang, Y., et al. (2019). A novel ensemble method for k-nearest neighbor. Pattern Recognition, 85, 13–25. https://doi.org/10.1016/j.patcog.2018.08.003

Zheng, N., Duan, X., & Lin, W. (1990). Computer vision system for inspecting defects in welding seams based on X-ray pictures. In A. G. Tescher (Ed.), Applications of Digital Image Processing XII (Vol. 1153, p. 494). SPIE. https://doi.org/10.1117/12.962351

Zhu, S., & Zhu, F. (2019). Cycling comfort evaluation with instrumented probe bicycle. Transportation Research Part A: Policy and Practice, 129, 217–231. https://doi.org/10.1016/j.tra.2019.08.009

8.0 APPENDICES

APPENDIX A. Signal Processing and Event Detection Details

This appendix documents the signal processing workflow used to identify vibration-based events from Data Bike surveys and align them with video and GPS data.

A.1. Accelerometer Data and Preprocessing

Tri-axial accelerometer data (a_x, a_y, a_z) were extracted from embedded GoPro metadata and combined into an acceleration magnitude:

$$a(t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2}$$

This formulation allows vibration events to be detected regardless of the axis along which they occur and is commonly used in mobile sensing applications for infrastructure condition screening.

A.2. Signal Smoothing (Butterworth)

To reduce high-frequency sensor noise before differentiation, $a(t)$was smoothed using a 4th-order Butterworth low-pass filter. The sampling frequency $f_s$was estimated from the accelerometer timestamps as $f_s \approx 1/\text{mean}(\Delta t)$. The filter was implemented in digital form using a normalized cutoff $f_c/(0.5f_s)$ and applied using zero-phase forward–backward filtering to avoid phase distortion.

A.3. Jerk Computation

Jerk was computed as the first derivative of the smooth acceleration magnitude with respect to time:

$$j(t) = \frac{d}{dt}a(t)$$

Discrete differentiation was applied using the sampling interval inferred from the accelerometer timestamp stream. Generally, for our camera and settings, we use a frequency of 200 Hz. The resulting jerk signal highlights abrupt changes in acceleration that typically

correspond to traversing surface defects such as cracks, potholes, or vertical misalignments, along with surface conditions like the presence of tactile pavement.

A.4. Threshold Calibration and Event Detection

High-jerk events were identified using a fixed threshold applied to the jerk magnitude. Threshold values were calibrated using pilot rides conducted over known smooth and rough trail segments. Calibration focused on minimizing false positives from routine riding behavior while retaining sensitivity to meaningful surface irregularities.

Once calibrated, the threshold was held constant during full-scale data collection to ensure consistency across riders, corridors, and survey periods.

Table A- 1: Example Jerk Threshold Calibration Rides

Trail Segment	Surface Type	Typical Speed (mph)	Observed Jerk Range	Selected Threshold
Pilot Segment A	Asphalt	15	Low–Moderate	2.5 m/s^3
Pilot Segment B	Concrete	14	Moderate	2.5 m/s^3
Pilot Segment C	Asphalt	12	Moderate–High	2.5 m/s^3

A.5. Speed Screening and Quality Control

Because vibration magnitude is influenced by riding speed, GPS-derived speed was used to identify and exclude near-zero-speed intervals (e.g., prolonged stops).

APPENDIX B. Dataset Summary

This appendix summarizes the composition and partitioning of the datasets used for model training, validation, and testing.

B.1 Data Sources

Two primary datasets were used:

1. **Local Trail Dataset**
   Images extracted from Data Bike surveys conducted on Utah trails and manually annotated into ten classes.
2. **Global Road Damage Detection (GRDD) Dataset**
   A publicly available dataset containing labeled roadway surface defects, used to supplement training for visually similar defect classes (cracking and potholes).

B.2 Local Fine-Tuning Dataset

The local dataset was split into training, validation, and test subsets to support fine-tuning and evaluation. Table B-1 summarizes the class distribution for each split.

Table B- 1: Local Dataset Class Distribution

Class	Train	Val	Test
Pothole	3	2	2
Longitudinal crack	44	11	11
Transverse crack	869	221	221
Alligator crack	15	8	8
Misalignment	174	47	47
Vegetation obstacle	31	16	13
Pole obstacle	43	12	13
Debris obstacle	13	8	5
Tactile pavement	93	30	32
Rail crossing	21	10	10

These distributions reflect observed field conditions, with transverse cracking dominating the dataset and several defect and obstacle categories remaining relatively rare.

B.3 Combined Training Dataset (GRDD + Local)

To address limited local sample sizes for visually complex defect types, the training dataset was augmented with GRDD imagery. Table B-2 summarizes the composition of the combined training and validation datasets.

Table B- 2: Combined (Local + GRDD) Dataset Class Distribution

Class	Train – GRDD	Train – Local	Val – GRDD	Val – Local
Pothole	4908	17	1636	6
Longitudinal crack	19665	334	1636	117
Transverse crack	9692	1160	3231	387
Alligator crack	7973	37	2658	12
Misalignment	0	260	0	87
Vegetation obstacle	0	65	0	22
Pole obstacle	0	96	0	32
Debris obstacle	0	39	0	13
Tactile pavement	0	143	0	48
Rail crossing	0	47	0	16

Cracking and pothole classes were dominated by GRDD imagery, while categories such as misalignment, tactile pavement, and obstacle classes relied primarily on local data. This composition reflects both data availability and visual relevance.

APPENDIX C. Representative Annotation Examples

This appendix documents representative examples of annotated images used for model training and evaluation. The examples illustrate annotation consistency, bounding-box placement, and the visual diversity present across defect and obstacle categories.

Figure C- 1: Longitudinal Crack

Figure C- 2: Transverse Crack

Figure C- 3: Alligator Crack

Figure C- 4: Pothole

Figure C- 5: Misalignment

Figure C- 6: Vegetation

Figure C- 7: Debris

Figure C- 8: Pole

Figure C- 9: Tactile Pavement

Figure C- 10: Rail Crossing

APPENDIX D. Dashboard

This appendix describes the dashboard developed as part of the project and provides guidance to support long-term reference.

D.1 Dashboard Overview

An interactive dashboard was developed to visualize geolocated defect detections and support exploratory analysis. The dashboard enables users to view detected conditions spatially, filter by defect category and confidence, and inspect associated images for verification.

D.2 Required Inputs

The dashboard accepts the following inputs:

- CSV file containing defect detections with latitude, longitude, class label, confidence score, timestamp, and image filename
- GPS route files (CSV or GPX format)
- Directory containing extracted image frames

D.3 Core Functionalities

Key functionalities include:

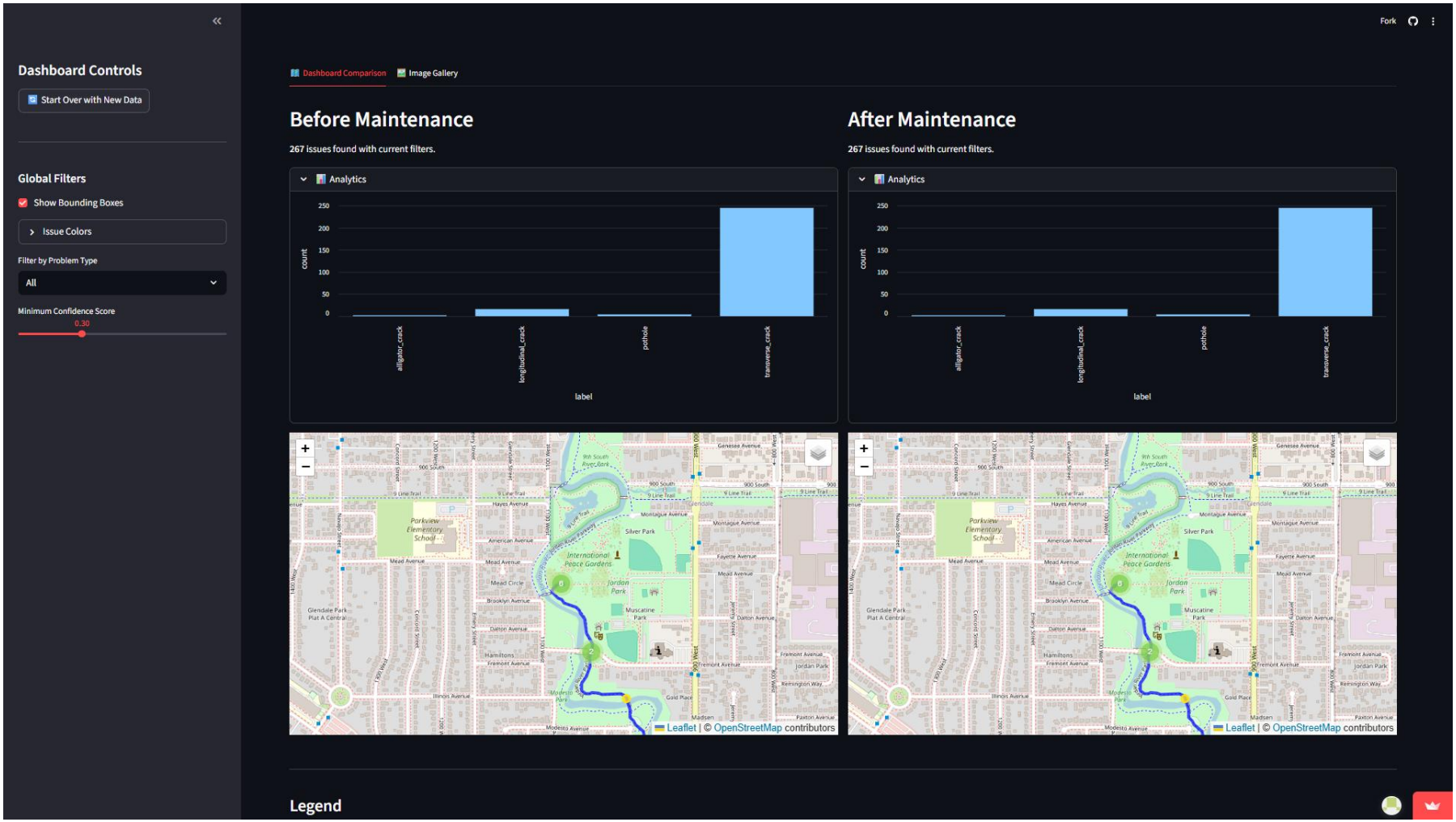
- Map-based visualization of surface condition with locations
- Sidebar filters for surface conditions type, confidence threshold
- On-click display of images and metadata for verification

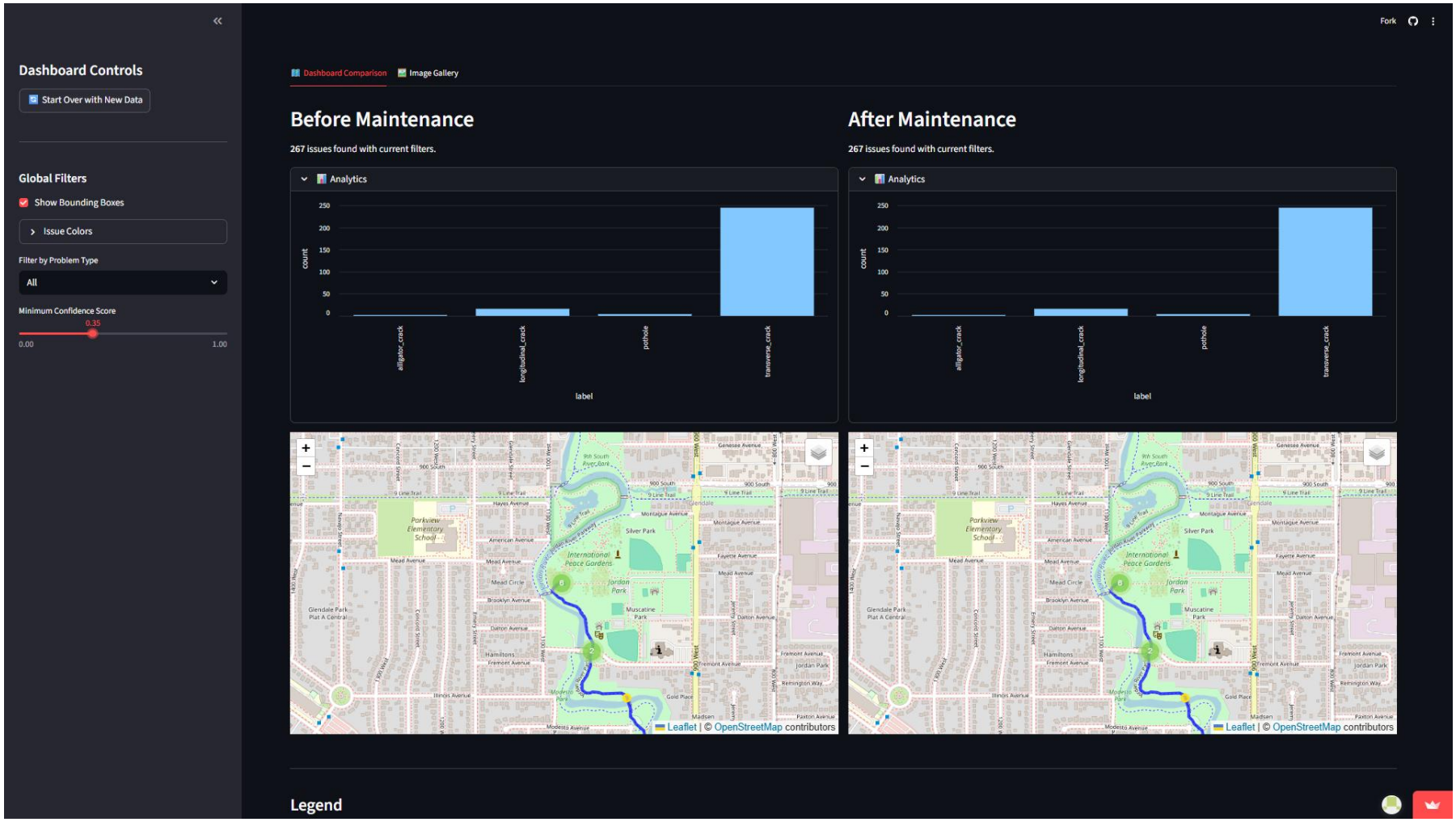
D.4 Archival Considerations

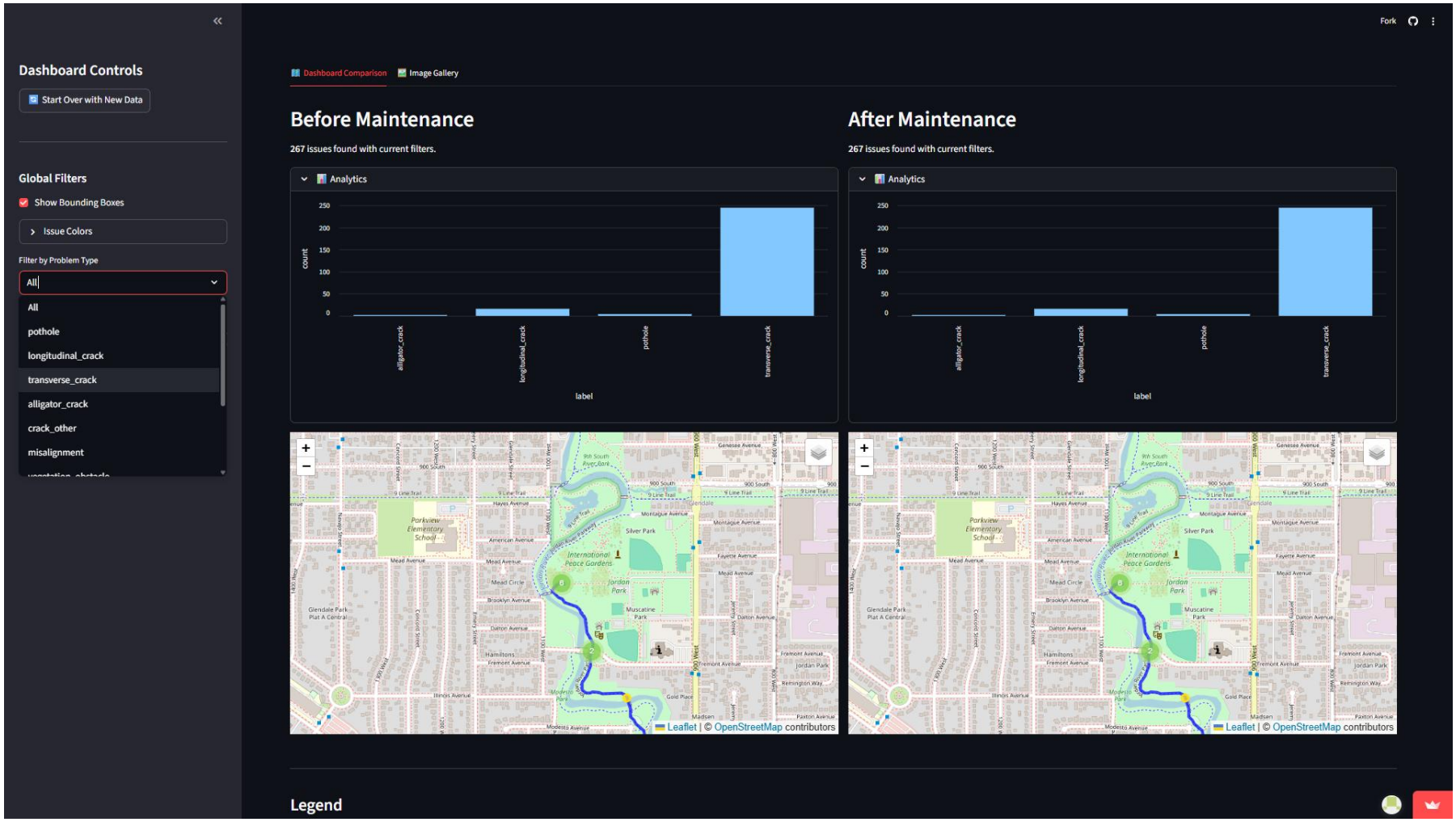
Because web-based dashboards may evolve over time, static screenshots of the dashboard interface (including filters and map views) should be archived with project documentation. This ensures long-term reference value even if the live deployment changes.

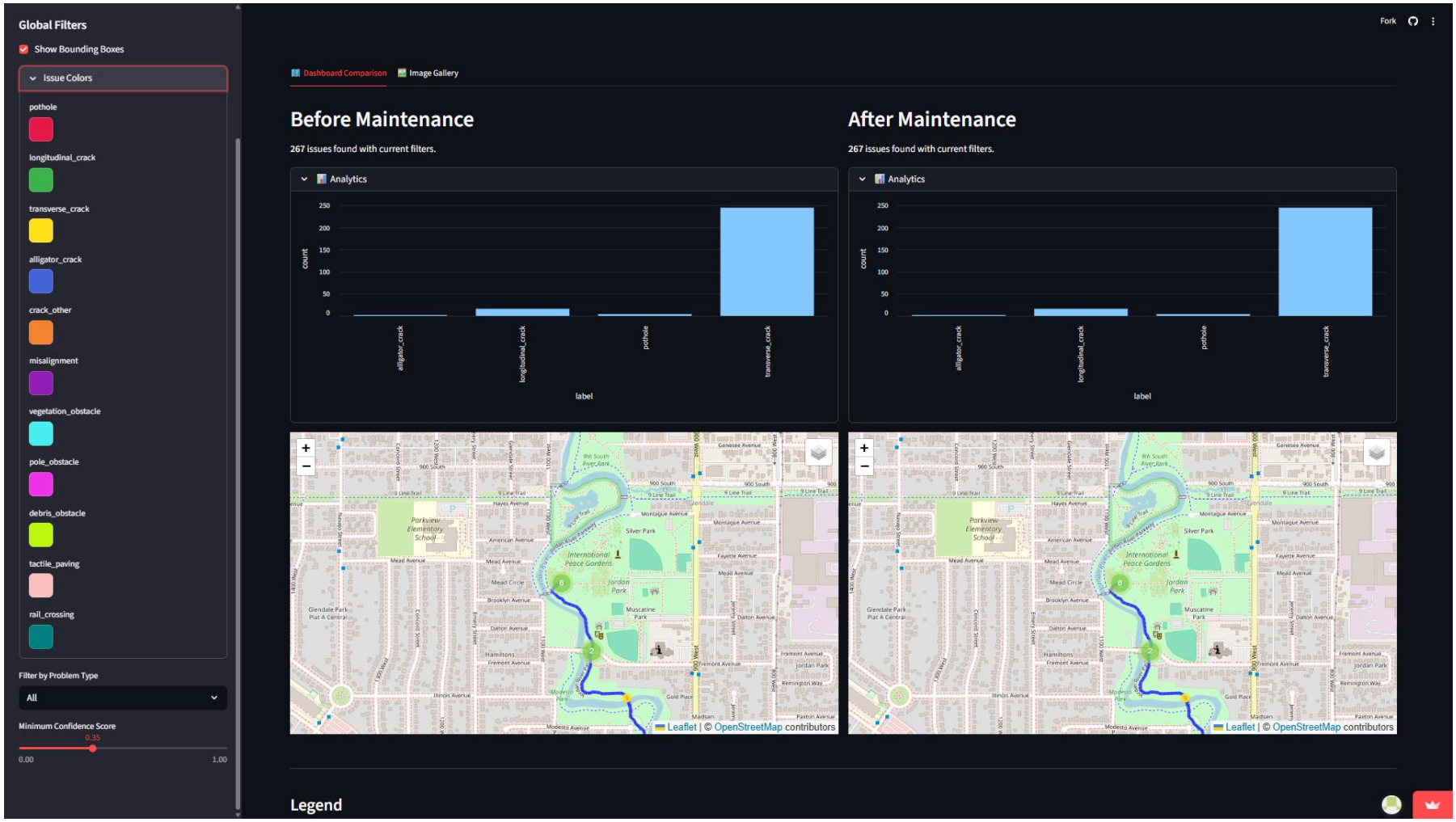

Fork
Data Upload & Config
Upload data, select columns, specify paths, and then click 'Run Comparison'.
1. Before Maintenance Data
Upload Route CSV (Before)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
Jordan_front_HERO13 Black-GPS...
2.1MB
Upload Predictions CSV (Before)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
Jordan_Holm_predictions.csv
39.2KB
2. After Maintenance Data
Upload Route CSV (After)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
Jordan_front_HERO13 Black-GPS...
2.1MB
Upload Predictions CSV (After)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
Jordan_Holm_predictions.csv
39.2KB

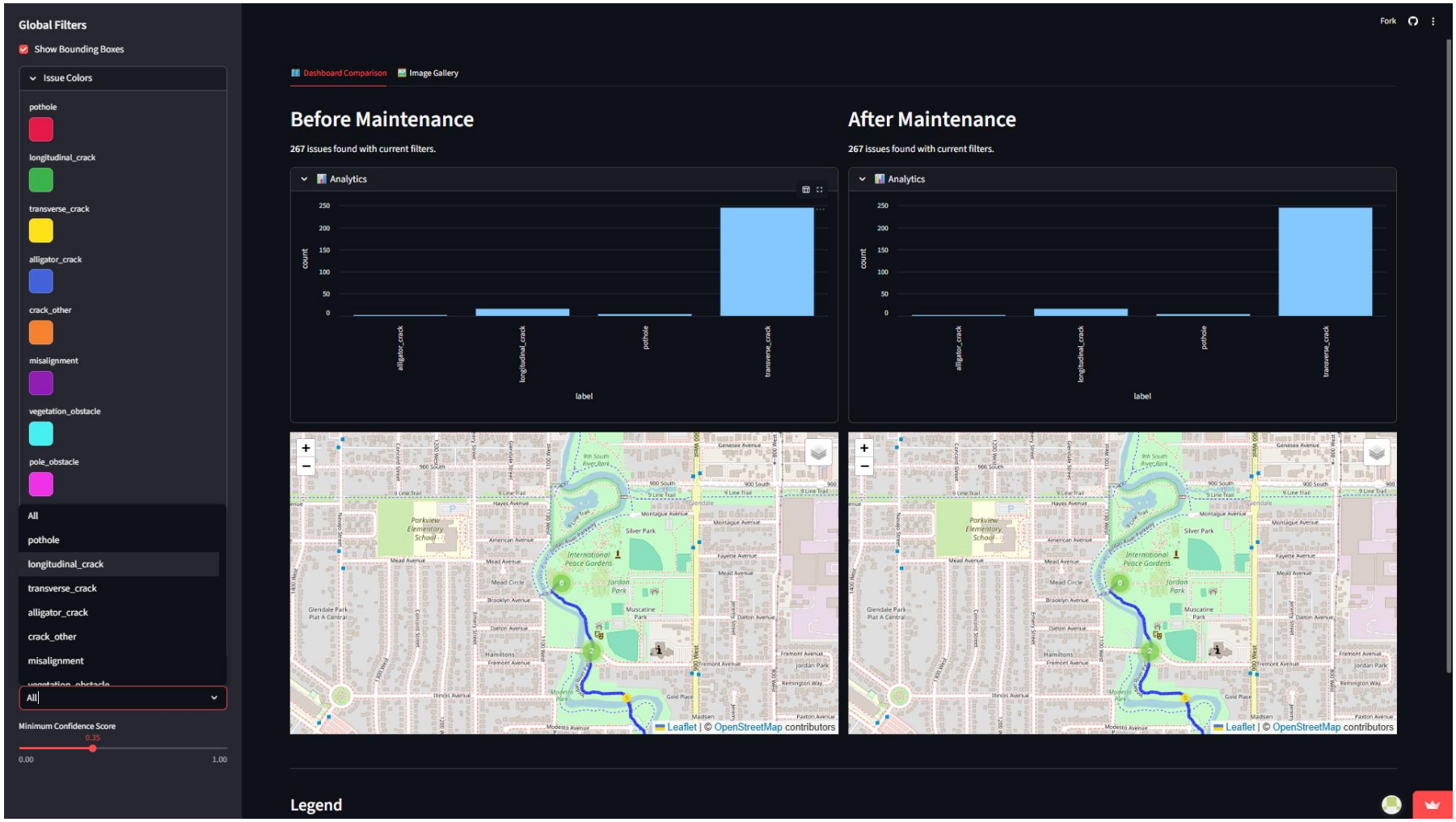

Jordan_front_HERO13 Black-GPS...
2.1MB
Upload Predictions CSV (After)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
Jordan_Holm_predictions.csv
39.2KB
3. Column & Path Configuration
'Before' Settings
Latitude Column (Before)
GPS (Lat.) [deg]
Longitude Column (Before)
GPS (Long.) [deg]
Confidence Column (Before)
confidence
Image Directory Path (Before)
Jordan_Holm/Jordan_Holm_event_frames
'After' Settings
Latitude Column (After)
GPS (Lat.) [deg]
Longitude Column (After)
GPS (Long.) [deg]
Confidence Column (After)
confidence
Image Directory Path (After)
Jordan_Holm/Jordan_Holm_event_frames
Press Enter to apply
Run Comparison
Fork

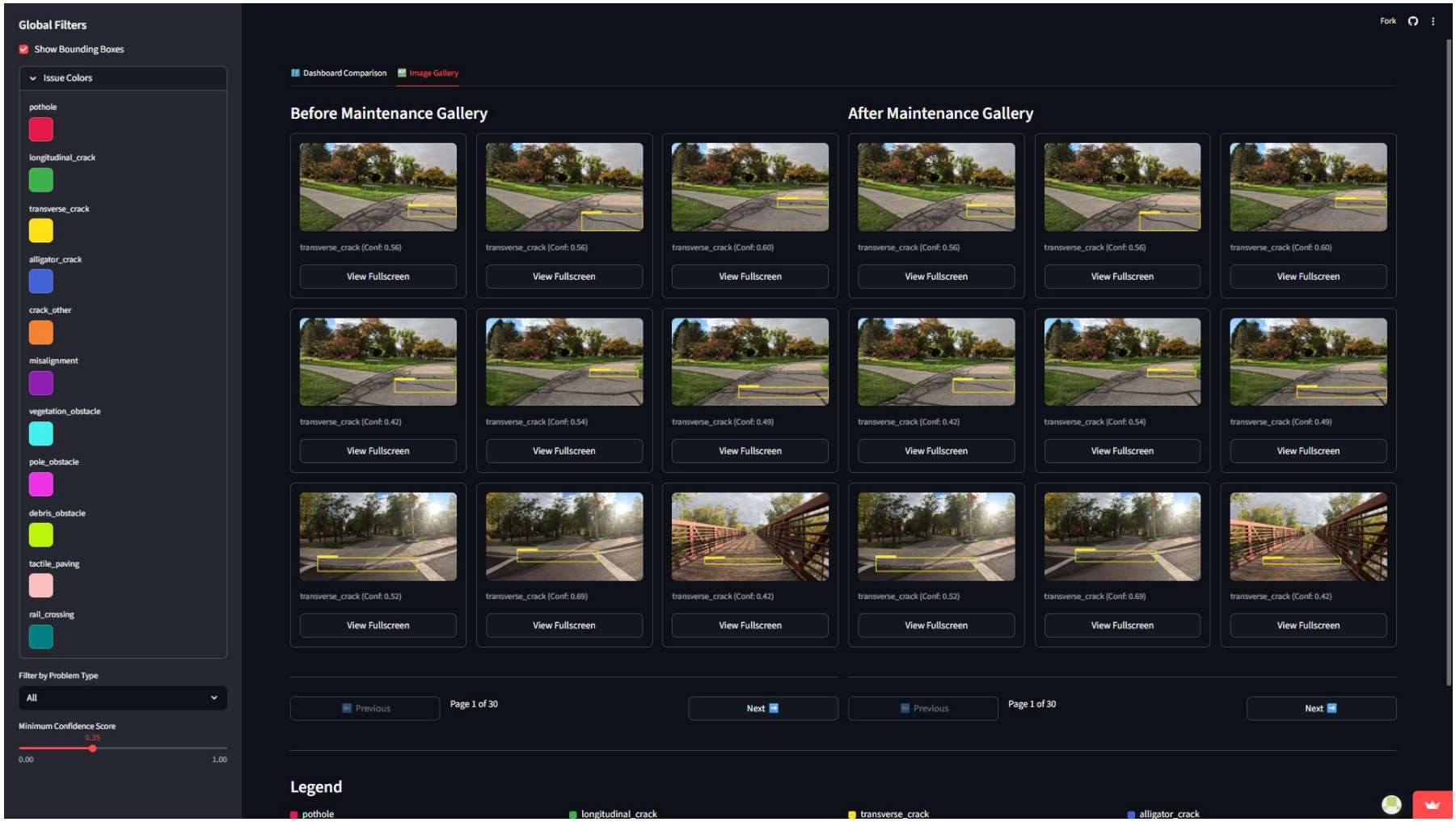

Dashboard Controls
Start Over with New Data
Global Filters
Show Bounding Boxes
Issue Colors
Filter by Problem Type
All
Minimum Confidence Score
0.30
Fork
Dashboard Comparison
Image Gallery
Before Maintenance
267 issues found with current filters.
After Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
Leaflet | © OpenStreetMap contributors
Legend

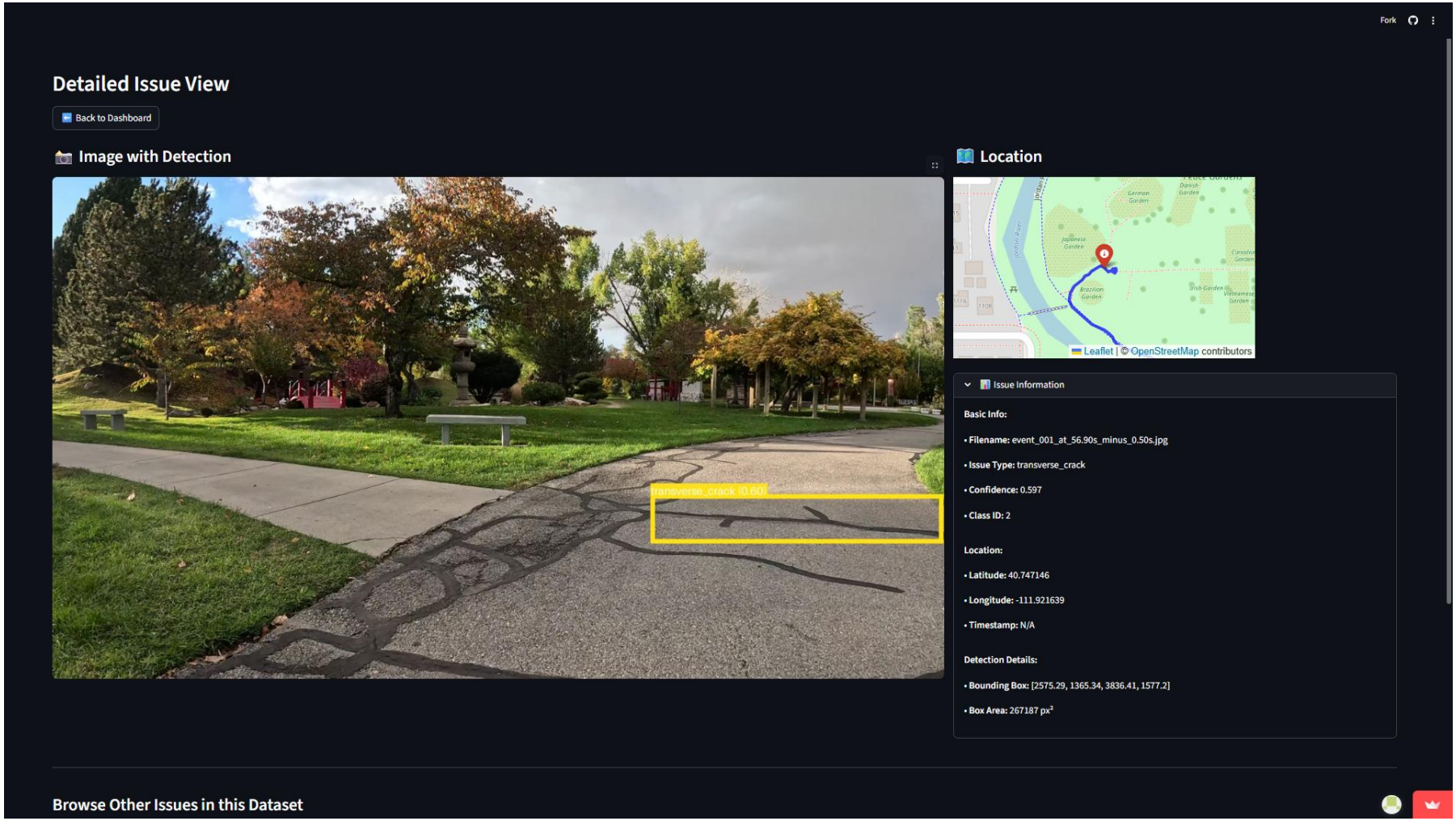

Dashboard Controls
Start Over with New Data
Global Filters
Show Bounding Boxes
Issue Colors
Filter by Problem Type
All
Minimum Confidence Score
0.35
0.00
1.00
Fork
Dashboard Comparison
Image Gallery
Before Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
After Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
Leaflet | © OpenStreetMap contributors
Leaflet | © OpenStreetMap contributors
Legend

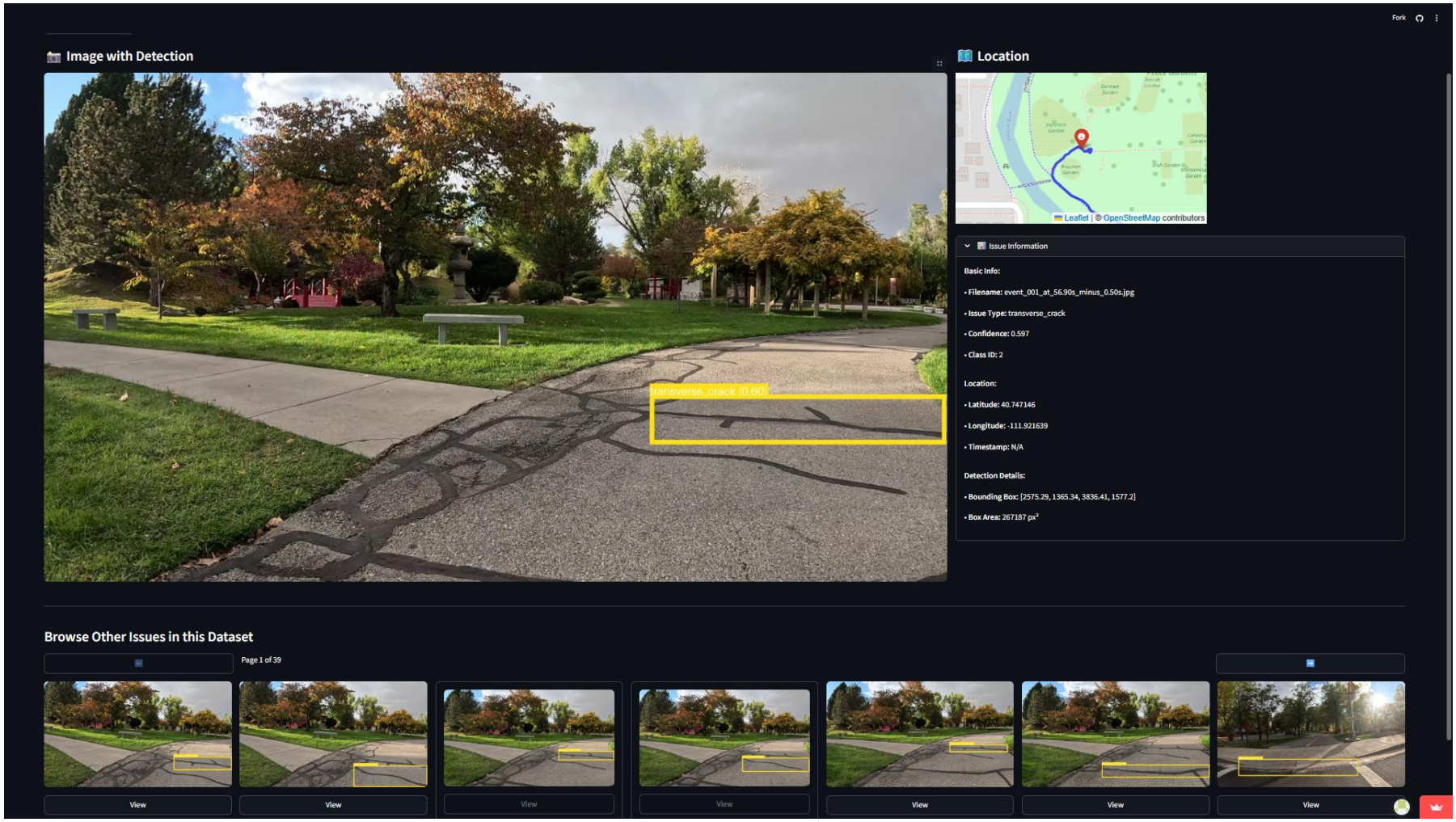

Dashboard Controls
Start Over with New Data
Global Filters
Show Bounding Boxes
Issue Colors
Filter by Problem Type
All
All
pothole
longitudinal_crack
transverse_crack
alligator_crack
crack_other
misalignment
Fork
Dashboard Comparison
Image Gallery
Before Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
After Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
Leaflet | © OpenStreetMap contributors
Leaflet | © OpenStreetMap contributors
Legend

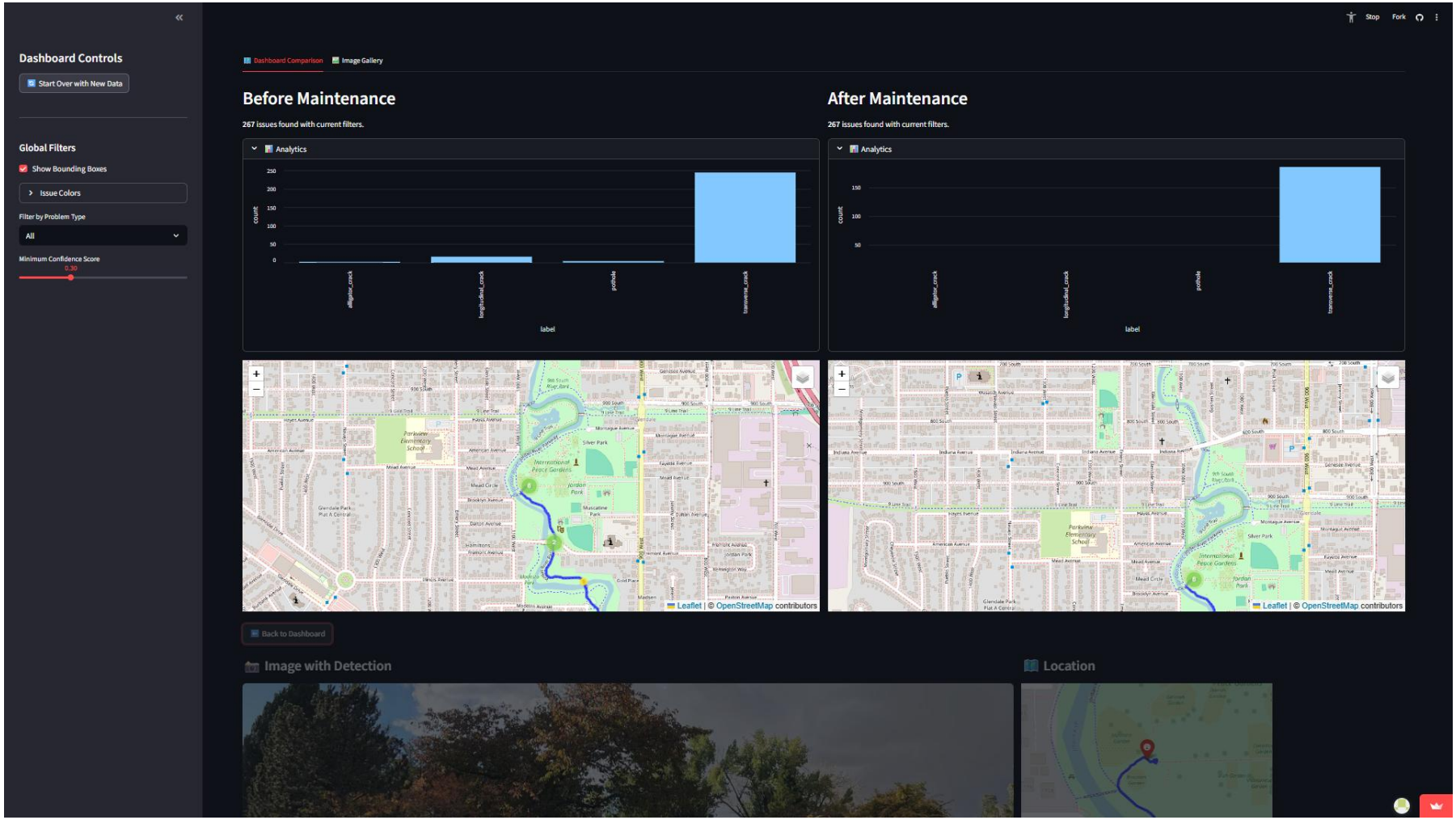

Global Filters
Show Bounding Boxes
Issue Colors
pothole
longitudinal_crack
transverse_crack
alligator_crack
crack_other
misalignment
vegetation_obstacle
pole_obstacle
debris_obstacle
tactile_paving
rail_crossing
Filter by Problem Type
All
Minimum Confidence Score
0.35
0.00
1.00
Fork
Dashboard Comparison
Image Gallery
Before Maintenance
267 issues found with current filters.
After Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
Leaflet | © OpenStreetMap contributors
Legend


Global Filters
Show Bounding Boxes
Issue Colors
pothole
longitudinal_crack
transverse_crack
alligator_crack
crack_other
misalignment
vegetation_obstacle
pole_obstacle
All
pothole
longitudinal_crack
transverse_crack
alligator_crack
crack_other
misalignment
All
Minimum Confidence Score
0.35
0.00
1.00
Fork
Dashboard Comparison
Image Gallery
Before Maintenance
267 issues found with current filters.
After Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
Leaflet | © OpenStreetMap contributors
Legend


Global Filters
Show Bounding Boxes
Issue Colors
pothole
longitudinal_crack
transverse_crack
alligator_crack
crack_other
misalignment
vegetation_obstacle
pole_obstacle
debris_obstacle
tactile_paving
rail_crossing
Filter by Problem Type
All
Minimum Confidence Score
0.35
0.00
1.00
Fork
Dashboard Comparison
Image Gallery
Before Maintenance Gallery
After Maintenance Gallery
transverse_crack (Conf: 0.56)
transverse_crack (Conf: 0.56)
transverse_crack (Conf: 0.60)
transverse_crack (Conf: 0.56)
transverse_crack (Conf: 0.56)
transverse_crack (Conf: 0.60)
transverse_crack (Conf: 0.42)
transverse_crack (Conf: 0.54)
transverse_crack (Conf: 0.49)
transverse_crack (Conf: 0.42)
transverse_crack (Conf: 0.54)
transverse_crack (Conf: 0.49)
transverse_crack (Conf: 0.52)
transverse_crack (Conf: 0.69)
transverse_crack (Conf: 0.42)
transverse_crack (Conf: 0.52)
transverse_crack (Conf: 0.69)
transverse_crack (Conf: 0.42)
View Fullscreen
Previous
Page 1 of 30
Next
Previous
Page 1 of 30
Next
Legend
pothole
longitudinal_crack
transverse_crack
alligator_crack


Fork
Detailed Issue View
Back to Dashboard
Image with Detection
transverse_crack (0.60)
Location
Leaflet | © OpenStreetMap contributors
Issue Information
Basic Info:
• Filename: event_001_at_56.90s_minus_0.50s.jpg
• Issue Type: transverse_crack
• Confidence: 0.597
• Class ID: 2
Location:
• Latitude: 40.747146
• Longitude: -111.921639
• Timestamp: N/A
Detection Details:
• Bounding Box: [2575.29, 1365.34, 3836.41, 1577.2]
• Box Area: 267187 px²
Browse Other Issues in this Dataset


Fork
Image with Detection
transverse_crack 0.60
Location
Leaflet | © OpenStreetMap contributors
Issue Information
Basic Info:
• Filename: event_001_at_56.90s_minus_0.50s.jpg
• Issue Type: transverse_crack
• Confidence: 0.597
• Class ID: 2
Location:
• Latitude: 40.747146
• Longitude: -111.921639
• Timestamp: N/A
Detection Details:
• Bounding Box: [2575.29, 1365.34, 3836.41, 1577.2]
• Box Area: 267187 px²
Browse Other Issues in this Dataset
Page 1 of 39
View
View
View
View
View
View
View


Dashboard Controls
Start Over with New Data
Global Filters
Show Bounding Boxes
Issue Colors
Filter by Problem Type
All
Minimum Confidence Score
0.30
Stop
Fork
Dashboard Comparison
Image Gallery
Before Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
After Maintenance
267 issues found with current filters.
Analytics
count
label
alligator_crack
longitudinal_crack
pothole
transverse_crack
Leaflet | © OpenStreetMap contributors
Leaflet | © OpenStreetMap contributors
Back to Dashboard
Image with Detection
Location

APPENDIX E. Computational Cost and Runtime Considerations

This appendix documents the computational resources and runtime characteristics associated with training and deploying the computer vision model used in this study. The intent is to provide transparency regarding computational requirements while clarifying the distinction between one-time model development costs and recurring operational costs.

E.1 Hardware Configuration

All model training and inference experiments were conducted using a single consumer-grade graphics processing unit (GPU):

- **GPU:** NVIDIA GeForce RTX 4060 Laptop GPU
- **Memory:** 8 GB VRAM
- **Execution environment:** Single-GPU, local workstation
- **Model architecture:** YOLOv8 object detection model

No specialized high-performance computing infrastructure or multi-GPU configuration was required for any stage of the workflow.

E.2 Image Resolution and Preprocessing

Raw images collected during Data Bike surveys were recorded at a native resolution of 3840 × 2160 pixels. For both training and inference, all images were resized internally to a unified model input resolution of 640 × 640 pixels. Images drawn from the Global Road Damage Detection (GRDD) dataset were upscaled from their native resolution **(**512 × 512 pixels) to the same input size. This unified preprocessing ensured consistent feature scaling across datasets while prioritizing spatial detail relevant to trail surface conditions.

E.3 Model Training Runtime

Model development consisted of two stages:

1. Foundation training, which leveraged a combination of GRDD imagery and local trail data to learn general visual representations of surface defects.

2. Fine-tuning, which focused on locally collected trail imagery to improve performance on trail-specific defect and obstacle classes.

The foundation training stage completed in approximately 29 hours, with early stopping triggered after 88 epochs based on validation performance. The fine-tuning stage required approximately 8 hours of additional runtime. These activities represent one-time or infrequent model development efforts and are not expected to recur during routine operational use.

E.4 Inference Runtime and Throughput

Inference runtime was evaluated to assess feasibility for routine, corridor-scale screening. Under the hardware and preprocessing configuration described above, the deployed model achieved the following average per-image runtimes:

- Preprocessing: approximately 2–6 ms per image
- Model inference: approximately 22–29 ms per image
- Postprocessing: approximately 1–3 ms per image

This corresponds to an effective throughput on the order of 30–40 images per second during GPU execution. At this rate, several thousand extracted frames can be processed within minutes, supporting efficient batch analysis following each survey.

E.5 Operational Cost Considerations

From an operational perspective, computational costs are dominated by model inference, not training. Inference requires substantially fewer resources than model development and can be performed using either in-house hardware or short-duration cloud-based computing, depending on agency IT practices. When expressed on a per-survey basis, the marginal computational cost of inference is small relative to labor associated with data collection, quality control, and downstream review of flagged locations.

Model retraining or fine-tuning may be conducted periodically as new labeled data become available or as the system is expanded to new environments. Such activities should be treated as periodic system enhancement efforts rather than recurring operational expenses.

E.6 Interpretation and Limitations

The computational characteristics documented here reflect the specific hardware and software configuration used in this study. Actual runtimes may vary depending on hardware availability, batching strategies, and storage performance. However, the results demonstrate that the proposed workflow can be executed using readily available computing resources and does not impose significant computational barriers to deployment.