Report No. UT-23.14

DASHCAM-ENABLED DEEP LEARNING APPLICATIONS FOR AIRPORT RUNWAY PAVEMENT DISTRESS DETECTION

Prepared For:

Utah Department of Transportation
Research & Innovation Division

Final Report
July 2023

RESEARCH

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

DISCLAIMER

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

ACKNOWLEDGMENTS

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

- Robert Chamberlin (UDOT Consultant)
- Paul Damron
- Craig Ide
- Matthew Maass
- Nick Holt
- Austin Baysinger

1. Report No. UT-23.14	2. Government Accession No. N/A	3. Recipient's Catalog No. N/A
4. Title and Subtitle DASHCAM-ENABLED DEEP LEARNING APPLICATIONS FOR AIRPORT RUNWAY PAVEMENT DISTRESS DETECTION		5. Report Date July 2023
		6. Performing Organization Code
7. Author(s) Arman Malekloo, Xiaoyue Cathy Liu, David Sacharny		8. Performing Organization Report No.
9. Performing Organization Name and Address University of Utah Department of Civil and Environmental Engineering 110 Central Campus Drive, Suite 2000 Salt Lake City, UT 84112		10. Work Unit No. 5H088 80H
		11. Contract or Grant No. 23-8193
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 Aug. 2022 to July 2023
		14. Sponsoring Agency Code 22.501

15. Supplementary Notes

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

16. Abstract

Pavement distress detection plays a vital role in ensuring the safety and longevity of runway infrastructure. This project presents a comprehensive approach to automate distress detection and geolocation on runway pavement using state-of-the-art deep learning techniques. A Faster R-CNN model is trained to accurately identify and classify various distress types, including longitudinal and transverse cracking, weathering, rutting, and depression. The developed model is deployed on a dataset of high-resolution dashcam images captured along the runway, allowing for real-time detection of distresses. Geolocation techniques are employed to accurately map the distresses onto the runway pavement in real-world coordinates. The system implementation and deployment are discussed, emphasizing the importance of a seamless integration into existing infrastructure. The developed distress detection system offers significant benefits to the Utah Department of Transportation (UDOT) by enabling proactive maintenance planning, optimizing resource allocation, and enhancing runway management capabilities. Future potential for advanced distress analysis, integration with other data sources, and continuous model improvement are also explored. The project showcases the potential of low-cost dashcam solutions combined with deep learning for efficient and cost-effective runway distress detection and management.

17. Key Words Pavement distress detection, runway infrastructure, automation, deep learning techniques, low-cost dashcam solutions		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 77	22. Price $50,000	

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

UNIT CONVERSION FACTORS

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2000 lb)	0.907	megagrams (or "metric ton")	Mg (or "t")
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or "t")	megagrams (or "metric ton")	1.103	short tons (2000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m^2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	0.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for the International System of Units. (Adapted from FHWA report template, Revised March 2003)

LIST OF ACRONYMS

AUC	Area Under the Curve
AI	Artificial Intelligence
AP	Average Precision
CNN	Convolutional Neural Network
CVAT	Computer Vision Annotation Tool
DCNN	Deep Convolutional Neural Network
FAA	Federal Aviation Administration
IoU	Intersection over Union
LBP	Local Binary Pattern
LTPP	Long-Term Pavement Performance
mAP	Mean Average Precision
NMS	Non-Maximum Suppression
PAVER	Pavement Evaluation
PASER	Pavement Surface Evaluation and Rating
PCI	Pavement Condition Index
PID	Pavement Image Dataset
PMP	Project Management Professional
PR	Precision and Recall
RDD	Road Damage Detection
ROI	Region of Interest
RANSAC	Random Sample Consensus
RPN	Region Proposal Network
SVM	Support Vector Machine
UDOT	Utah Department of Transportation

EXECUTIVE SUMMARY

This project aims to address the need for efficient and cost-effective pavement distress detection on airport runways. By leveraging a dashcam as a low-cost solution and implementing the Faster R-CNN model, the system achieved promising results in automated distress identification and localization. The use of computer vision techniques allows for the detection of various distress types, including longitudinal and transverse cracking, rutting, depression, and weathering.

The application of the Faster R-CNN model demonstrated its effectiveness in accurately identifying distress areas on airport runways. The system's ability to detect and localize distresses provides airport authorities and maintenance teams with valuable insights for proactive maintenance planning. By automating the distress detection process, the system offers significant advantages over manual inspection methods, including increased efficiency, objectivity, and the ability to cover larger areas in a shorter time. Additionally, the utilization of a dashcam as a low-cost solution makes the system accessible and scalable for airports with limited budgets, allowing them to enhance runway maintenance practices without substantial financial investments.

While the project achieved promising results, limitations were identified, such as the need for additional training data, labeling accuracy improvements, and challenges in accurately detecting certain distress types. These limitations present opportunities for future research and refinement of the system. Overall, this project contributes to the advancement of automated pavement distress detection, offering a practical and cost-effective approach for maintaining the safety and quality of airport runways.

1.0 INTRODUCTION

1.1 Problem Statement

Pavement distress appears in many civil infrastructures, from roads to bridges and buildings. The deterioration process would be exacerbated if distresses were not treated and maintained properly and timely. The goal of pavement maintenance is to provide a safe and operable pavement system for a feasible cost. Poor pavement condition is a safety risk and therefore is of top priority to related agencies. Roadway pavements and airport pavements share similar characteristics; however, airport pavements, including taxiways, aprons, and especially runways, are far more critical for ensuring safe aircraft operations, and failure can have severe consequences. While all parts of the airport pavements require regular maintenance and inspection, different regulations and standard specifications require airport runway pavements to be more regulated. Federal Aviation Administration (FAA) standard specifications require higher ratings for the Pavement Condition Index (PCI) and the Pavement Surface Evaluation and Rating (PASER). Current FAA Advisory Circular 150/5380-7B recommends using PCI to assess airfield pavement conditions for maintenance and rehabilitation planning (FAA, 2014). PCI is a widely used index for assessing the overall condition of a pavement, including both structural and functional aspects. The structural performance of pavements is critical to ensuring the safety of aircraft during takeoff and landing. Structural distresses, such as cracking and rutting, can cause significant damage to aircraft and endanger the lives of passengers and crew. In contrast, functional performance measures the surface condition of a pavement and its ability to provide a smooth and safe surface for aircraft operations.

A comprehensive pavement maintenance program (PMP) for airports considers several key elements to ensure effective management and upkeep of the pavement infrastructure. These elements include:

1) Detailed Pavement Inventory: A thorough and well-documented pavement inventory is established, providing comprehensive information about the airport's pavement areas, including runways, taxiways, aprons, and associated surfaces.
2) Consistent Inspection Schedule: A regular and consistent inspection schedule is implemented to assess the condition of the pavement. This involves systematic and periodic inspections

conducted by trained personnel to identify and document any distresses or abnormalities. The inspection frequency may vary depending on factors such as pavement age, traffic volume, and environmental conditions.

3) Distress Recording and Documentation: The PMP includes a robust system for recording and documenting the identified distresses during inspections. Distresses such as cracking, rutting, potholes, and other abnormalities are accurately documented, including their severity, extent, and location. This data serves as a baseline for evaluating the pavement's condition, tracking changes over time, and prioritizing maintenance and repair activities.
4) Information Retrieval System: An efficient and well-established system is implemented to organize and retrieve information related to the pavement inventory, inspection reports, distress records, maintenance activities, and historical data. This allows for quick access to relevant information, facilitates data analysis and decision-making, and enables effective communication between stakeholders involved in the PMP.

The first three items of a PMP described above could incur major costs to the agencies. Specifically, resource allocation and manual inspection of distresses requires trained personnel, equipment, and time. Moreover, it becomes challenging to maintain a consistent inspection schedule without disrupting normal airport operations. The dimensions of runways and airports' heavy air traffic make it extremely difficult for agencies to ensure a timely and detailed examination (Morian et al., 2020). Therefore, many agencies showed their interest in utilizing automated and/or semi-real-time distress detection.

Essentially, a ***data acquisition*** system, a ***distress detection*** algorithm, and a **decision *support*** framework are required to transition from a manual to an automated distress detection system. Data acquisition provides two- or three-dimensional data (2D or 3D). Depending on the device used, the input data is one of imagery, radar, thermal, and/or electric signal data types. Out of the above-mentioned data types, 2D and 3D images using various image-capturing technologies are getting more and more popular and widely used in many recent automated distress detection applications (Mohan & Poobal, 2018). Using the collected input data and a properly chosen distress detection algorithm, it is then possible to identify and classify the distress's type, severity, and extent. The output of the distress detection algorithm then provides the necessary inputs for the decision support framework, which offers optimal treatment strategies, including allocating resources through a decision-making system or generating reports for review. Although a fullfledged automated pavement distress detection system would comprise all three components, this project primarily focuses on data acquisition systems and distress detection algorithms. A promising solution in detection algorithms can be leveraged by incorporating artificial intelligence (AI), or more precisely, computer vision and deep learning algorithms. A plethora of algorithms and application-specific detection methods have demonstrated the potential for pavement distress detection in terms of effectiveness and robustness (Hsieh & Tsai, 2020).

This project introduces an innovative and cost-effective runway pavement distress detection solution using AI and dash cameras. By harnessing the power of deep learning, we have developed a state-of-the-art model capable of identifying and classifying various types of distresses on airport runways. Our approach leverages location-specific frame extraction techniques, ensuring comprehensive coverage of designated runway segments. This enables us to gather invaluable observations without causing any disruption to airport operations. By adopting this low-cost methodology, airports worldwide can swiftly detect and monitor runway conditions, leading to enhanced safety and improved maintenance practices. This transformative solution not only revolutionizes the way runway distresses are detected but also showcases the potential of AI and dash cameras in streamlining infrastructure management. Embracing this technology opens up new possibilities for efficient and cost-effective runway monitoring, laying the foundation for smarter and more sustainable airport operations.

1.2 Objectives

The objectives of this project are twofold. The primary objective of this project is to develop and implement an AI-assisted, cost-effective solution for the detection and classification of airport runway pavement distresses using dash cameras. By leveraging advanced computer vision and deep learning techniques, the project aims to deliver an automated distress detection system that can accurately and efficiently identify and record different types of pavement distresses, including longitudinal and transverse cracking, rutting, depression, and weathering. The system will utilize a unique frame extraction algorithm to collect a diverse dataset from various angles and times of day, thereby ensuring a comprehensive range of conditions are represented. This data will be used to train a robust deep learning model that is capable of detecting even obscured distresses and classifying them with high precision. The second objective of this project is to visualize the identified distresses onto a GIS map for reporting purposes, where a performance rating similar to PCI is developed to quickly inform airport staff of the locations of potentially unsafe portions of the runway. That being said, the GIS map can also be updated as new images are recorded and can help both to generate better performance ratings and also to validate the location of the identified distresses.

This study is of great importance. First, research on runway pavement damage using state-of-the-art technologies is minimal. The proposed dashcams system can be mounted inside airport maintenance vehicles and capture high-quality images. Moreover, recent dashcam technologies can automatically transfer data to the cloud for easier access and management. This image-capturing method might be more beneficial than others, such as drones or other dedicated systems that are typically used now in different airports, which have inherent operational (e.g., temporary airport closure) and cost issues. Second, runway closures due to pavement maintenance or repair can cause significant delays and cancellations, resulting in loss of revenue for airlines and inconvenience for passengers. Early detection and repair of pavement distress using deep learning applications can reduce the need for lengthy runway closures and improve airport efficiency. Third, there has not been any dedicated public runway pavement dataset. Generating a new dataset for airport pavement distress detection using deep learning techniques is a significant contribution to the field. The creation of this new dataset can provide a benchmark for future research in this area and can aid in the development of more accurate and reliable deep learning models.

The use of dash cameras augmented with deep learning models for airport pavement distress detection enables airport authorities to make data-driven decisions. The model can provide near-real-time analysis of the pavement conditions, allowing authorities to prioritize maintenance and repair activities based on the severity of the distresses detected. The successful implementation of such technology has the potential to significantly improve the safety and efficiency of airport operations while reducing costs and environmental impacts.

1.3 Outline of Report

The rest of the report is structured as follows. Section 2.0 summarizes the literature on the recent computer vision advancement in distress detection and the current practices and limitations of runway pavement distress detection. Section 3.0 illustrates the procedure implemented for distress detection using the Faster RCNN model and the geolocation measurements for the identified distresses. Section 4.0 describes the data sources used for this study, and Sections 5.0 and 6.0 present the results and findings. Section 7.0 presents a detailed roadmap for the implementation and deployment of the distress detection system, encompassing essential steps and considerations throughout the process. Lastly, Section 8.0 concludes the study and outlines recommendations for future research.

2.0 LITERATURE REVIEW

There are various sources for obtaining pavement images. The widely varied features in these images, associated with pavement distress, present a challenge in detection and classification. Additionally, different camera views (wide-view and top-down) produce different image datasets, each with a distinct set of unique functions that might serve a specific purpose. That being said, recent advancements in computer vision and deep learning have resulted in a more robust process of detecting distresses in pavements, which is less affected by vagaries in data collection. A distress detection algorithm based on these new technologies can detect and, in some cases, quantify the severity of the distresses. The following section will discuss the state-of-the-art distress detection algorithms used in the literature.

2.1 Distress Detection Methods

The image processing techniques for distress detection in computer vision are divided into five families of approaches (Chambon & Moliard, 2011): 1) histogram, 2) morphology, 3) learning, 4) filtering, and 5) model. The current standard distress detections methods which use one or a combination of the algorithms stated above are CrackTree (Zou et al., 2012), CrackIT (Oliveira & Correia, 2014), FFA (Nguyen et al., 2011), MPS (Amhaz et al., 2014), and, CrackForest (Shi et al., 2016). These algorithms are usually used as benchmarks to compare against any newly developed methods. Of those five families of techniques, the learning approaches are the most popular. In particular, machine learning has been widely adopted in many fields, including pavement distress detection. Given sufficient data, machine learning algorithms can provide outstanding performances in many tasks. Chen et al. (2022) proposed a new pavement distress detection algorithm - local binary patterns (LBP), with support vector machine (SVM) - a computer vision technique that is augmented with a machine learning technique. With 150 images (500 × 500 pixels), feature extraction was done using LBP, a texture description operator in the gray range, and coupled with PCA to reduce the dimensionality of high-dimensional data. Later, the SVM was used to categorize the distress types.

The main problem with machine learning techniques is their shallow learning approach, i.e., they require a predefined, manually programmed feature extraction stage (Hsieh & Tsai, 2020). Often, images contain complex information that can be affected by illumination and environment, which limits the performance of traditional machine learning methods. More recently, deep learning, a subset of machine learning techniques, has emerged to remedy these drawbacks. Convolutional Neural Networks (CNN) are the backbone of many deep learning applications. Deep Convolutional Neural Networks (DCCN) have been shown to outperform edge detection algorithms used in the traditional distress detection approaches (Dorafshan et al., 2018). With the recent improvement in these networks, the problems of object classification, detection, and segmentation have gained much attention. In classification problems, the goal is to determine the presence of distress in an image and classify its type. Object detection problems, on the other hand, draw bounding boxes around the areas of identified distress, with some methods generating the distress type. Finally, segmentation is a pixel-wise prediction of distress, i.e., each pixel is classified as either crack or no crack.

The first attempt at using deep learning in pavement distress detection was carried out by Zhang et al. (2016). Using a low-cost smartphone, the authors used a DCNN to classify square image patches sampled from 500 pavement images (3264 × 2448 pixels). With a sampling strategy, the authors generated 640,000 samples as the training set, 160,000 samples as the validation set for 5-fold cross-validation when training the network, and 200,000 samples as the testing samples from the 500 images – each sample is a 3-channel (RGB) 99 × 99 pixel image. Using an Nvidia QuadroK220 GPU, they achieved the minimum error during training in 20 epochs. (Pauly et al., 2017) extended the previous approach to demonstrate that deeper DCNNs are better at performing the same task since a 4-layer network could only be considered a shallow network compared to the current networks used for other applications. It is shown that accurate results are highly dependent on the network's depth (Urban et al., 2017). Using an Nvidia Quadro M4000 GPU, the authors observed increased performance for deeper networks.

Transfer learning is a recent technique developed to improve deep learning performance. The low-level convolutional layers in DCNNs could learn more generic features with strong transferability, and higher-level layers learn more specific knowledge (S. J. Pan & Yang, 2010). This improves the learning ability of the model significantly and can greatly benefit the deep learning models with relatively small datasets. (Gopalakrishnan et al., 2017) utilized transfer learning and a single-layer neural network classifier trained on ImageNet pre-trained VGG-16 DCNN to classify whether an image has distress. They used a subset of pavement distress images (2048 × 3072 pixels) from the publicly available FHWA's Long-Term Pavement Performance

(LTPP) program. In total, 760 training samples, 84 validation samples, and 212 testing samples were used to evaluate the performance of pre-trained DCNNs used in the study. More recently, newer approaches include the use of DCNNs followed by PCA (X. Wang & Hu, 2017), Fully Connected Networks (R. Fan et al., 2019; Z. Fan et al., 2018), Fully Convolutional Networks (FCNs) (Dung & Anh, 2019; X. Yang et al., 2018), U-Net (Cheng et al., 2018), and Residual Networks (Bang et al., 2018).

The efficacy of deep learning-based distress detection algorithms is highly dependent on the quality and quantity of input data. 2D or 3D images used for training deep learning models can be captured from a plethora of image-capturing devices. A critical aspect of these images is the camera views. If the training images' camera views are not identical to the target images' camera views, the deep learning model will not perform as intended after deployment. The following section delves more into the camera views and discusses the two types of image datasets used in pavement distress detection.

2.2 Top-Down vs. Wide-View Images

Most datasets used in pavement detection with deep learning resort to top-down images. Low-cost smartphones, either handheld or vehicle-mounted, and versatile drones have demonstrated capabilities in capturing these high-quality images. For example, Zhang et al. (2018) used a single line-scan industry camera to capture 800 (2,000 × 4,000 pixels) pavement images on a moving vehicle. They used a DCNN transfer learning technique to detect and separate distress regions from sealed cracks. After 8,000 epochs, the model achieved its best performance and accurately distinguished cracks from sealed cracks.

Smartphones are light, inexpensive, and easy to deploy in the field. However, they have limited capabilities in extreme weather conditions, and the image quality may not be as consistent as dedicated and robust image-capturing systems built for capturing pavement images. Furthermore, drones are well-matched for aerial reconnaissance and have a larger field of view.

Y. Pan et al. (2018) used multispectral pavement images acquired by drones to detect asphalt pavement damage, including cracks and potholes. They used a suite of machine learning and deep learning approaches and concluded that drones offer a powerful tool for monitoring pavement conditions. In a similar study by Zhu et al. (2022), a drone with a high-resolution camera was used. The study employed three state-of-the-art object detection models, Faster R-CNN, YOLOv3, and YOLOv4, to train the dataset. A total of 300 images were captured (7952 × 5304 pixels). These images were further divided into 77 blocks, each 512 × 512 pixels, where 3,151 pavement images with different types of distresses were identified. Transfer learning was also used to accelerate the training process in the first few convolutional layers. The results showed that the prediction performance of YOLOv3 was the best, and it performs more robustly under different environments, including shadows, trees, and pavement markings.

Using wide-view images taken from smartphones or dashboard cameras (dashcams) is a very low-cost approach and can be used readily by municipalities and road agencies. Such an approach generally contains many images with features of non-interest (such as shadows, cars, sidewalks, and surfaces with no distresses) and in various environmental conditions (cloudy, rainy, etc.) While this problem was intractable until recently, the latest deep learning methods have been able to solve this to a high degree of accuracy. Furthermore, in order to make the training easier, Some (2016) suggested that data should be collected during cloudy days or use additional lighting to avoid shadows and increase the image quality. Another common problem is unbalanced datasets, where there are many more training images without cracks than those with cracks which leads to improper training. One preprocessing step suggested by the author was to use a classification algorithm to filter images with no pavement distress to create a more balanced dataset for training.

Maeda et al. (2018) built a mobile app that used an efficient deep-learning architecture to find cracks in road images. They mounted a smartphone with the app on the dashboard of a moving car, which captured wide-view images once per second and recognized and classified into eight different categories. Furthermore, they released the complete dataset of 9,053 images with 15,435 annotations. This dataset was used to kick off the IEEE BigData Road Damage Detection (RDD) Challenge 2018 and has led to considerable work in this area.

The RDD2020 (Arya et al., 2021) dataset, released as part of the IEEE RDD Challenge 2020, is a larger version of this dataset, with additional images from Japan, the Czech Republic, and India. The RDD2020 dataset contained a training set of 21,041 images, and the test set contained 5,295 images. Strong approaches to solving this problem include single-stage methods like SSD, Yolov1, YoloV2 (Mandal et al., 2018), and two-stage methods like FasterRCNN.

Mandal et al. (2020) analyzed the performance of three networks, YOLO, CenterNet, and EfficientDet, on this dataset. Similar to previous studies, transfer learning was also used to speed up training and improve performance. The best overall model was found to be YOLOv4, with an F1-score of 0.58, beating others by a margin of 0.1. Maeda et al. (2021) and Zhang et al. (2020) proposed the use of Generative Adversarial Networks (GANs) and various tricks to augment this dataset to get better results.

Cao et al. (2020) used 9,493 dashcam images sourced from multiple databases and evaluated them on eight deep-learning-based road damages. The Inception V2 model provided the best balance of road-damage-detection accuracy and image processing time. The study found that some distress types are underrepresented in image databases and that additional data with diversified distress types could help increase the detection accuracy of the models significantly.

Another wide-view dataset is introduced by Ma et al. (2017). The authors matched the Google Street View images with ratings of government inspectors found in public databases to create a dataset of wide-view images.

Majidifard et al. (2020) introduced The Pavement Image Dataset (PID). The PID dataset includes 7,237 wide-view and top-down images of pavement with annotated bounding boxes. The intent was to use the wide-view images to classify distress and the top-down-view images to calculate the distress densities. They showed that deep learning could be used to create a consistent measure for scoring pavement distress.

Finally, a recent work by Zhang (2022) has introduced a new dataset of wide-view images called CRDD, consisting of 4,319 images with 21,552 annotations. It aims to address some of the shortcomings of the previous datasets, such as using dashboard-mounted professional cameras, allowing for consistent image quality and camera angles.

2.3 Runway Pavement Distress Detections

The accuracy of the distress detection rate should be relatively high, which might require an extensive dataset. However, the majority of the existing open-source pavement images datasets, for example, Crack500 (F. Yang et al., 2019), RDD2020 (Arya et al., 2021), and others introduced earlier, are collected from roadways and primarily from flexible pavements (i.e., asphalt). Airport pavements, however, use a mix of flexible and rigid pavement (i.e., Portland cement concrete) in different parts of the airport, such as taxiways, aprons, and runways. Moreover, the roadway pavement distress types differ significantly from runway pavements (W. Wang et al., 2019).

Therefore, a suite of diverse sets of pavement images would be required for an acceptable level of distress detection in airport pavement applications.

In airports, highly sophisticated non-destructive testing machines or robots are typically used for pavement condition monitoring. For example, Li et al. (2021) used the airport pavement detection robot in 13 airports to capture 3,485 images with distress (1800 × 900 pixels). The deep learning model used in the paper is an encoder-decoder that adopts the convolution network VGG-19 and U-Net. Their results found that their model performs better than U-Net, a pixel-level classifier that already performs well for image segmentation. Recently, airports have turned to newer technologies, such as drones. In a recent study by Jiang et al. (2020), pavement distress segmentation based on drone-captured images was studied. In addition to the captured images, the Crack500 dataset was also used to augment the training set. The study used the U-Net model and outperformed the FPHBN model.

As described earlier, using dashcams as an image source has not been explored. This paves the way for a more streamlined and cost-effective pavement damage detection implementation using dashcams that can be mounted on airport maintenance vehicles. Since the cost of deploying dashcams to capture runway pavement images is low, they can be a source of up-to-date information about the runway, compared to a drone-based system which is usually deployed a few times a year. Another benefit of dashcams is that since different images of the same pavement in different times and conditions are captured, it would allow for the distress detection model to gain more confidence about its results.

3.0 METHODOLOGY

As noted in Section 2.0, the unprecedented improvements in deep learning have shown significant advantages in computer vision tasks. The plethora of algorithms indicates such strides toward data-driven decision-making platforms. The steps in which a successful model can achieve high performance in detecting distress lie in ensuring that such steps are clearly defined and validated. A typical architecture of an image processing-based method for pavement distress detection comprises the following elements:

1) Image acquisition (data collection)
2) Preprocessing (filtering, segmenting, and labeling)
3) Detection and classification (model training)
4) Model testing

More specifically, the implementation and deployment of a dash camera-enabled distress detection system using the state-of-the-art Faster R-CNN deep learning model involves several key elements. Each of these elements is described in detail in the following sections. Additionally, a visual representation of the implementation steps is provided in the form of a flowchart shown in Figure 3.1.

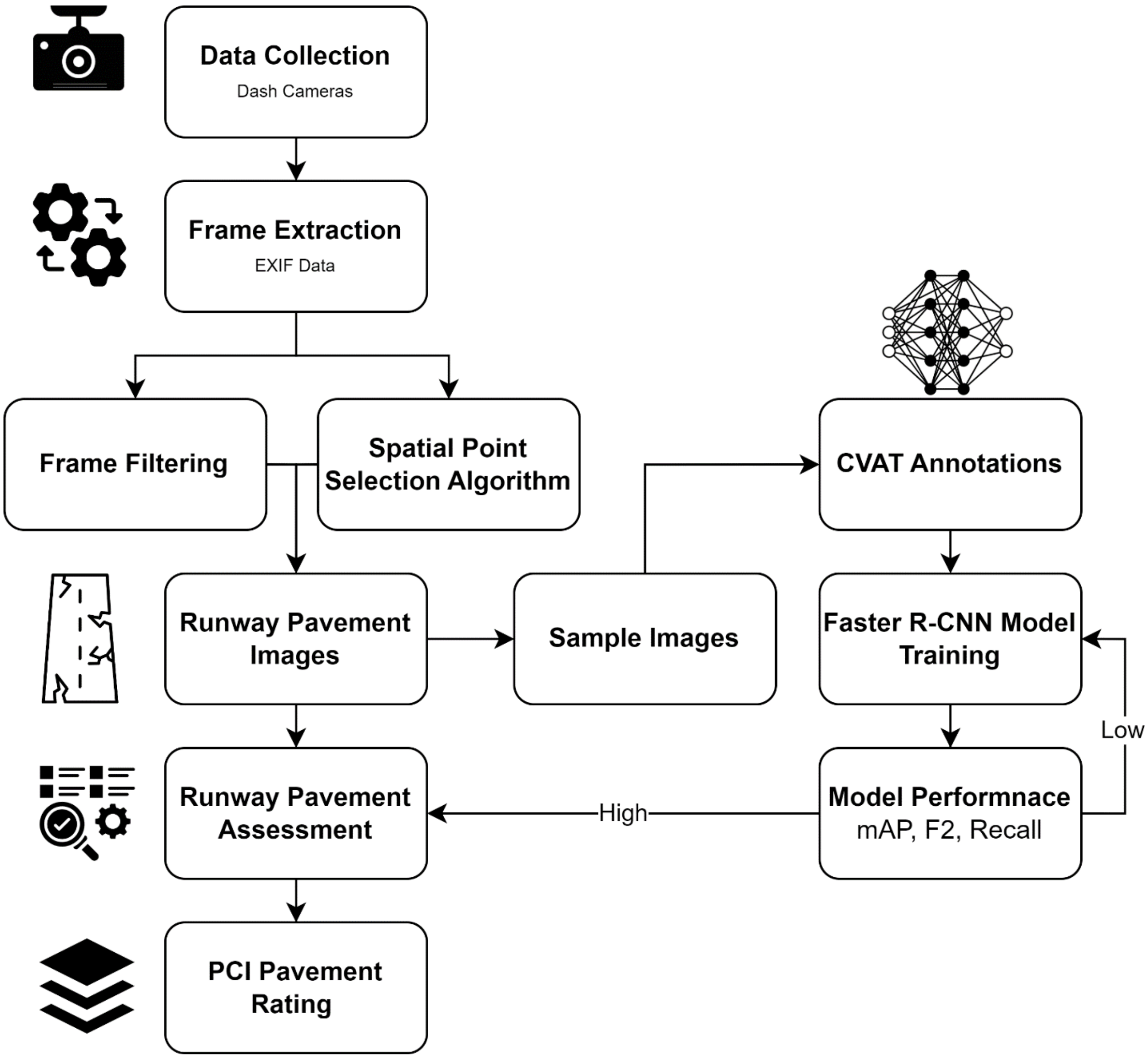


Figure 3.1: Dashcam-Enabled Distress Detection System

3.1 Image Acquisition

Collecting high-quality images is the first and foremost element in developing a decent distress detection model. As described earlier, 2D and 3D images from different sources can provide the necessary inputs for dataset collection. Image sources can be categorized into 1) visible light images from cameras, 2) laser images that are mainly used to reconstruct 3D point clouds of distresses, 3) radar images used for distresses with a depth, and 4) infrared images used for the measurement of distress depth for special occasions. This project utilizes a high-quality, low-cost dash camera that can be easily mounted on the windshield of the airport maintenance vehicles without obstructing the driver’s view and, more importantly, without disrupting the airport's normal operation.

This project incorporated a NEXAR dashcam (model number: NBDVR622GW) (*NEW 622GW Dash Cam | Nextbase*, n.d.). It is a high-quality 8-megapixel lens capable of capturing 4K (3840 x 2160 pixels) images at 30 frames per second (FPS) and a GPS sampling rate of 10 Hz. The dashcam is powered via a 12V cigarette lighter and turns on and records as soon as the vehicle starts up and stops when the vehicle is turned off.

Data collection was performed at two airports in the State of Utah: Spanish Fork Municipal Airport and Provo Airport. Dashcams were installed on October 18, 2022, at Provo Airport, and on October 20, 2022, at Spanish Fork Municipal Airport, respectively. The recording period lasted until November 1, 2022, at both locations. Special attention was given to ensure that the dashcams were securely fixed and remained stable throughout the vehicle's movement. Figure 3.2 shows the precise placement of the dashcam on the windshield of the maintenance vehicle. With an onboard 64 GB memory, the dashcam was able to record the footage across the entire data collection period. During the data collection, vehicle drivers were not given any specific instructions regarding the route they should follow. This approach allowed us to maintain a standard inspection procedure across both airports.

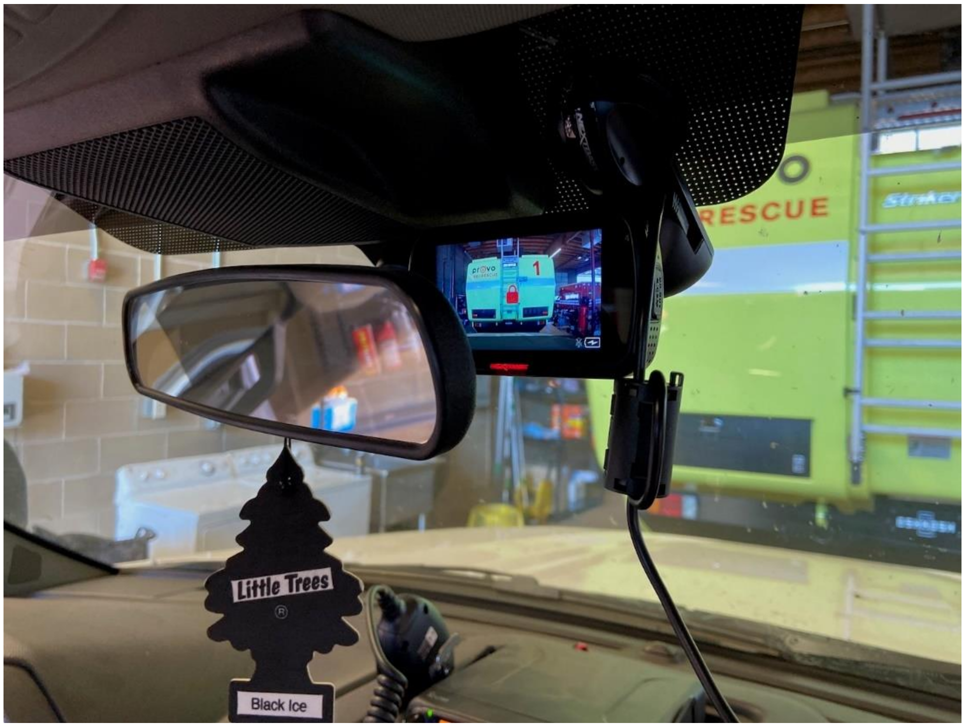


Figure 3.2: Mounted Dash Camera Inside the Airport Maintenance Vehicle

3.2 Preprocessing

In order to extract image frames for sections of the airport that are of interest to this project, the geotagged metadata embedded in the video frames is used. The EXIF (Exchangeable Image File Format) file stores important information about the frames and is extracted from the video using the ExifTool command line interface program (*ExifTool by Phil Harvey*, n.d.) using Python. The following items are extracted and stored:

1) DateTimeOriginal
2) GPSLatitude
3) GPSLongitude
4) GPSSpeed
5) GPSTrack
6) SampleDuration

where *GPSTrack* is the GPS heading from the true North direction, and *SampleDuration* is the number of seconds since the beginning of the recording, and the rest are self-explanatory. The frame number is then simply calculated by multiplying *SampleDuration* by FPS. Finally, the OpenCV python library is used to extract the frames and attach the corresponding EXIF tags to each frame. Given the large file sizes of the 4K video footage obtained, this process took several days, even with multiprocessing.

The ASTM D5340-11 testing standard requires subdividing asphalt pavement into a 5,000 contiguous ft^2 area (+ 2,000 ft^2 if the total area of the runway section is not evenly divisible by 5,000). For example, the runway section shown in Figure 3.3 is divided into 10 sample units, each containing 5,000 ft^2.

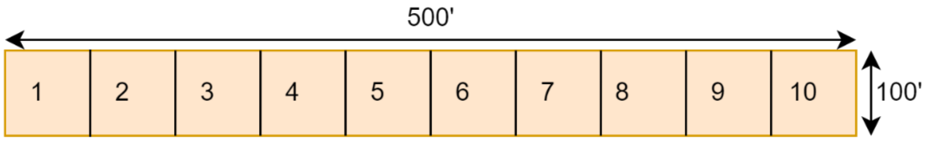


Figure 3.3: Subdivided Sections into Sample Units

Consequently, the airport responsible staff would partition the airport into equal segments. For example, Figure 3.4 illustrates the subdivided Spanish Fork Airport. Since the airport is not evenly divisible, each segment has a total area of 6,500 ft^2 (50 ft ×130 ft)

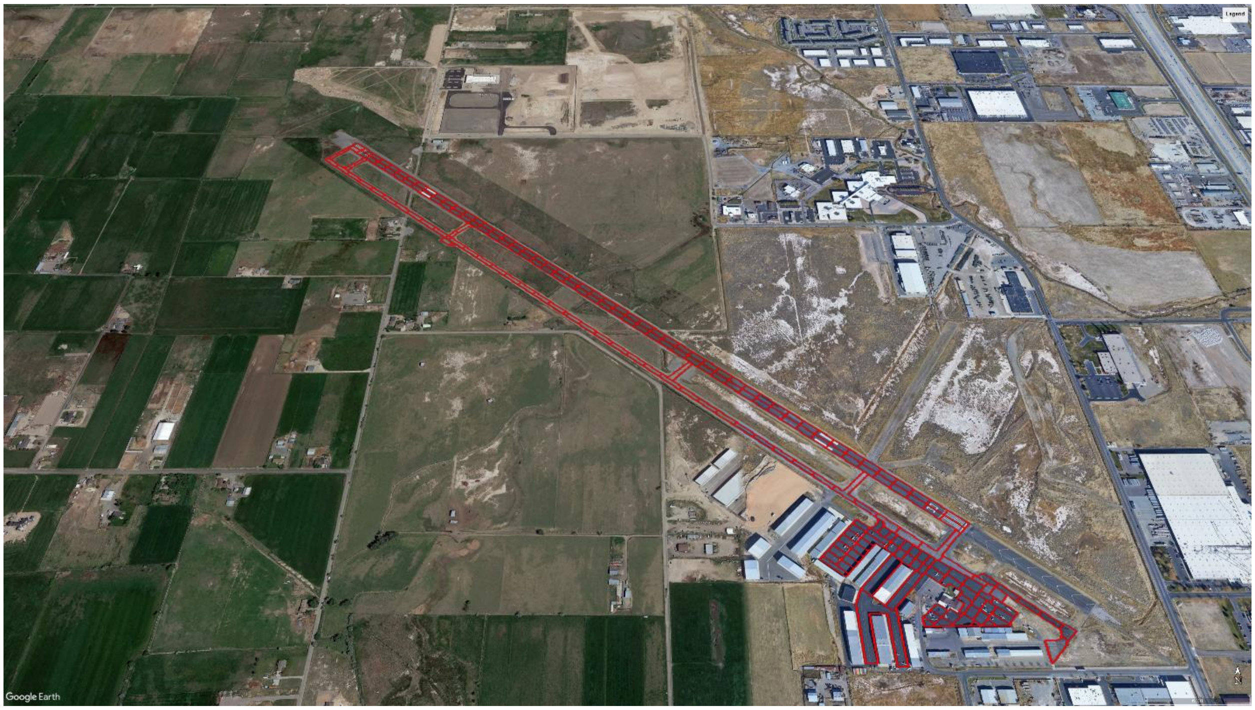
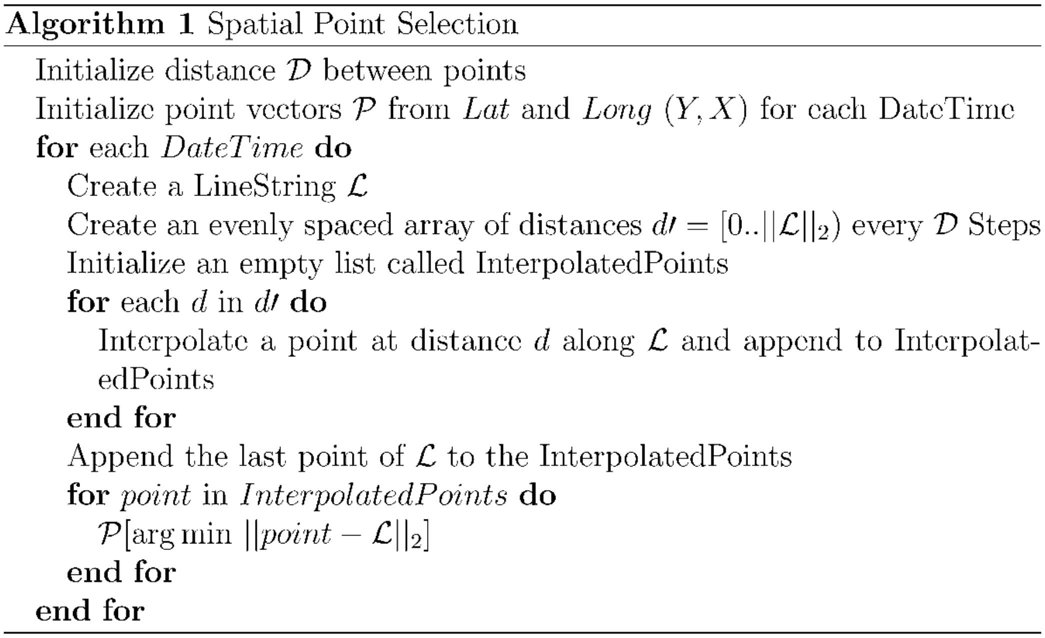


Figure 3.4: Spanish Fork Airport Subdivided Segments

In order to select images that are inside any of the segments, it is necessary to use the previously extracted geolocation information. Therefore, geospatial analysis is carried out using the extracted GPSLatitude, GPSLongitude, and airport segment geometry (in .KML format). Only those frames that are within each segment are selected.

There were several occasions where the vehicle did not move, and no useful images were captured. To remedy this and using the extracted *GPSSpeed*, only those frames where the vehicle speed was more than 10 kph (6 mph) were selected. Finally, to ensure satisfactory lighting conditions and using the extracted *DateTimeOriginal*, the final selection was applied to those frames where the time of recording was between 8 AM and 4 PM.

To ensure we capture a diverse range of footage, we developed an algorithm that takes into account the frame rate of the dashcam (30 frames per second) and the varying speed of the vehicle. This algorithm aims to record non-overlapping samples from different times of the day within each segment. By implementing this algorithm, as illustrated in Figure 3.5, we aim to obtain a comprehensive and well-rounded dataset such that selected consecutive frames contain the same distress from multiple angles.

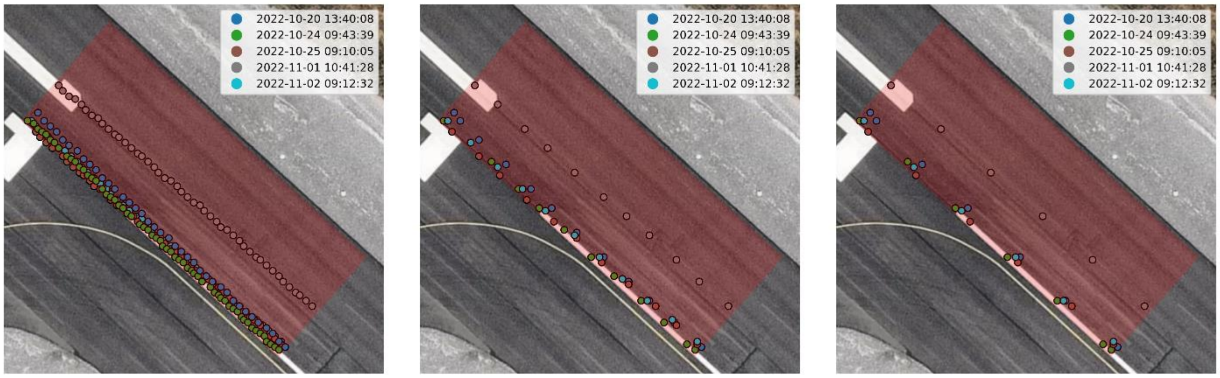


```
Algorithm 1 Spatial Point Selection
Initialize distance $\mathcal{D}$ between points
Initialize point vectors $\mathcal{P}$ from $Lat$ and $Long$ $(Y, X)$ for each DateTime
for each $DateTime$ do
    Create a LineString $\mathcal{L}$
    Create an evenly spaced array of distances $d' = [0..||\mathcal{L}||_2)$ every $\mathcal{D}$ Steps
    Initialize an empty list called InterpolatedPoints
    for each $d$ in $d'$ do
        Interpolate a point at distance $d$ along $\mathcal{L}$ and append to InterpolatedPoints
    end for
    Append the last point of $\mathcal{L}$ to the InterpolatedPoints
    for $point$ in $InterpolatedPoints$ do
        $\mathcal{P}[\arg\min ||point - \mathcal{L}||_2]$
    end for
end for
```

Figure 3.5: Spatial Point Selection Algorithm

Here is an example to illustrate how the algorithm works. In Figure 3.6(a), we can see all the GPS sample points extracted from each frame for a specific runway segment on different dates. Figure 3.6 (b) and (c) depict the sample points that are approximately 5 and 10 meters apart, respectively. As demonstrated, the algorithm offers flexibility in selecting points with a desired distance between them. It is important to note that choosing a larger distance between the points may result in a smaller number of images for subsequent processes. However, this approach can avoid duplicated recordings of the same distress from multiple frames. The general objective is to have as many frames as possible that contain the same distress from multiple angles.

Two main factors influence the approximation of the desired distance between sample points. First, there are instances where no GPS signals can be recorded, resulting in missing data points. This can occur due to signal loss or other factors that disrupt GPS reception. As a result, the actual distances between sample points may deviate from the desired distance. Second, the vehicle's speed can fluctuate significantly throughout the segment, and this affects the frequency at which GPS sample points are obtained. For example, when the vehicle is traveling at higher speeds, the time interval between consecutive GPS samples increases, resulting in fewer data points within the segment. In this project, a distance of 5 meters between sampling points was selected to ensure a balance between capturing sufficient data and minimizing the impact of the aforementioned factors.



(a) All points (b) 5 m (c) 10 m

Figure 3.6: Sampled GPS points in a segment where (a) shows all the points (i.e., no sampling), (b) at least 5 meters difference between each sampled point, and (c) at least 10 meters difference between each sampled point.

3.3 Runway Pavement Distresses

Data annotation can be carried out after extracting sample points on the pavement. Data annotation is the process of labeling or annotating data with specific information or metadata to make it more understandable and useful for deep learning algorithms or human analysis. It plays a crucial role in training deep learning models for automated pavement distress detection and analysis. The labeled data serves as a training set that enables the algorithms to learn patterns and characteristics of different distress types, ultimately allowing them to classify and detect distress automatically in new, unseen pavement images.

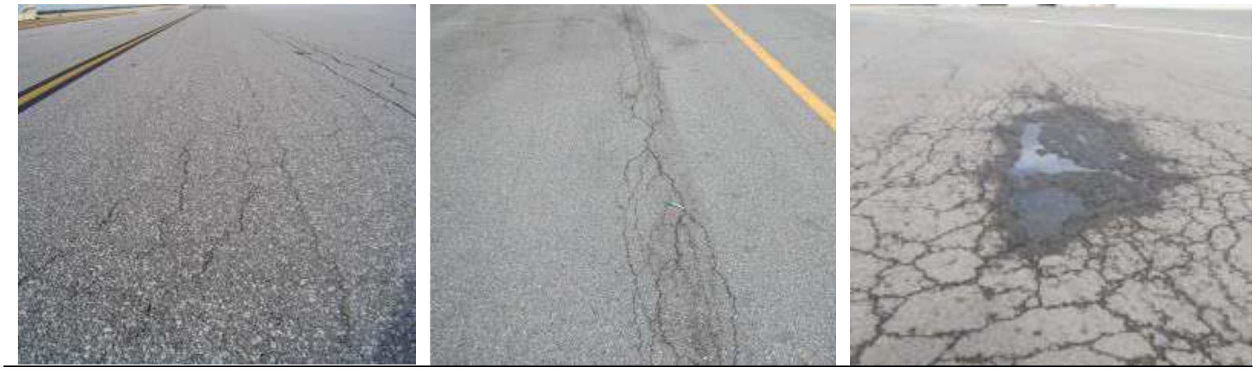
To facilitate the data annotation process for this project, we utilized the open-source Computer Vision Annotation Tool (CVAT)[1]. Using CVAT, bounding boxes were created around the identified distress areas within the pavement images. These bounding boxes serve as visual markers that encapsulate the specific distress for further analysis. Each bounding box is then associated with a corresponding distress label, indicating the type of distress it represents. Once the pavement images have been labeled with distress annotations, they are utilized to create training and testing datasets for further analysis and model development. In this project, the distress types were defined based on established standards such as the ASTM D5340-20 and the

[1] https://www.cvat.ai/

Pavement Maintenance Management System (PAVER)[2]. The defined distress types serve as a standardized classification system that enables consistent identification and categorization of pavement distress. Some of the most common distress types, as per the aforementioned standards, are outlined below:

a) Alligator or fatigue cracking

This is mainly caused by repeated loading in traffic areas.

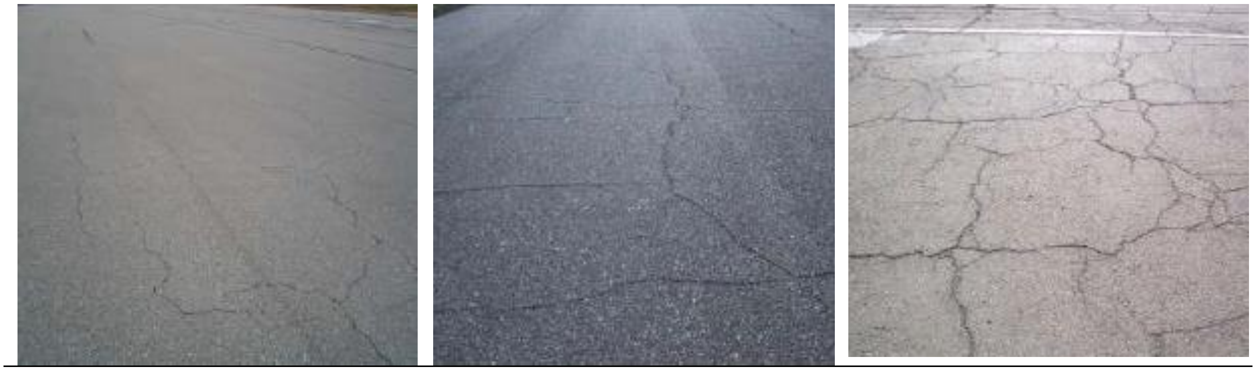


(a) (b) (a)

Figure 3.7: Alligator cracking (a) low severity, (b) medium severity, and (c) high severity.

b) Block cracking

The leading cause of this crack is the weather, and it can occur in large proportions of pavements and sometimes in traffic areas. This crack type is different from alligator cracks. Alligator cracks form smaller, multi-sided pieces with sharp angles. Also, alligator cracks are caused by repeated loadings.



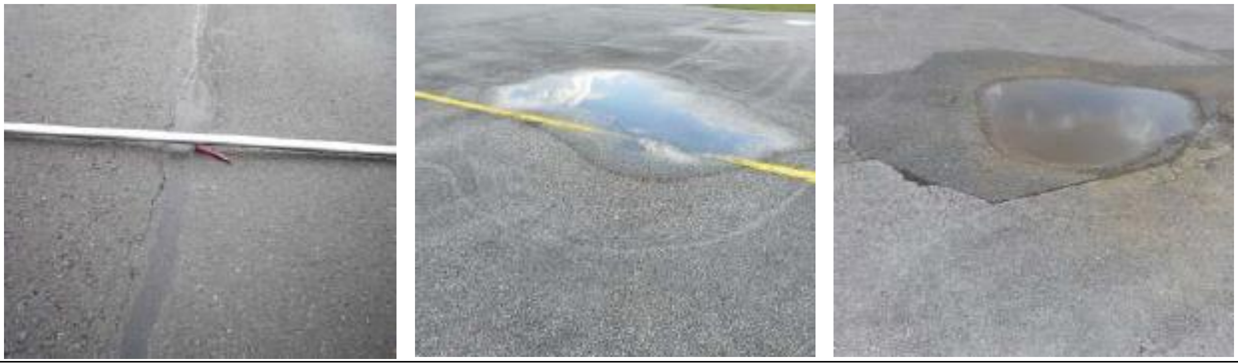
(a) (b) (a)

Figure 3.8: Block cracking (a) low severity, (b) medium severity, and (c) high severity.

[2] https://transportation.erdc.dren.mil/paver/Index.htm#

c) Depression

This distress type is caused by the settlement of the foundation or built-up from construction. Light depressions are not noticeable until after a rain. They can also be located without rain because of stains created by ponding water.

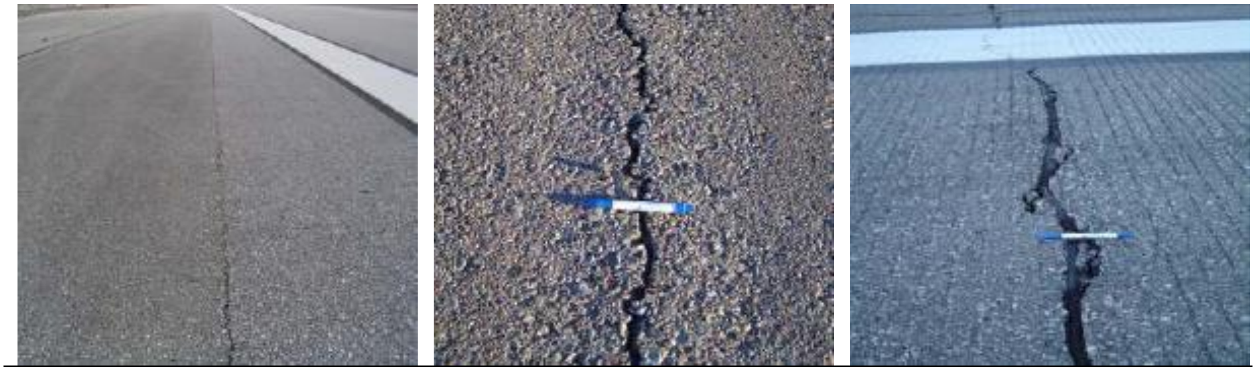


(a) (b) (a)

Figure 3.9: Depression (a) low severity, (b) medium severity, and (c) high severity.

d) Longitudinal and transverse cracking

This is perhaps the most common type of crack on pavements that can happen anywhere. It is caused by either 1) poorly constructed paving lane joint, 2) shrinkage of the asphalt concrete surface due to low temperatures or hardening of the asphalt, or 3) reflective crack caused by cracks beneath the surface course.

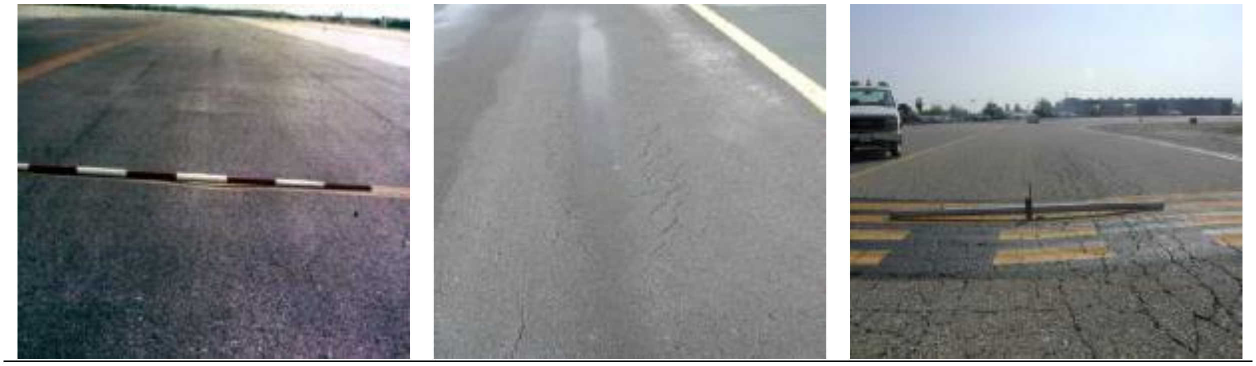


(a) (b) (a)

Figure 3.10: Longitudinal and transverse cracking (a) low severity, (b) medium severity, and (c) high severity.

e) Rutting

It is surface depression in the wheel path caused by consolidation or lateral movement of the materials due to traffic loads.

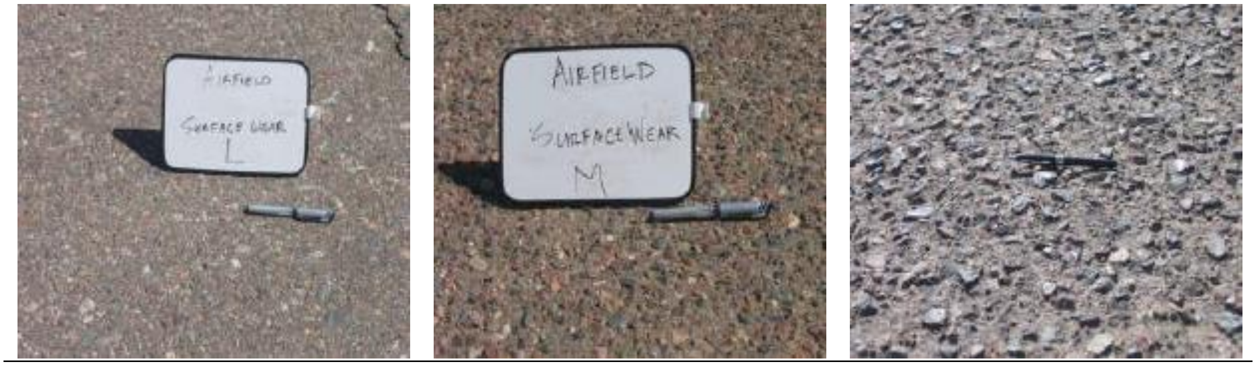


(a) (b) (a)

Figure 3.11: Rutting (a) low severity, (b) medium severity, and (c) high severity.

f) Weathering

Weathering is defined as the wearing away of the asphalt binder and fine aggregate matrix from the pavement surface.



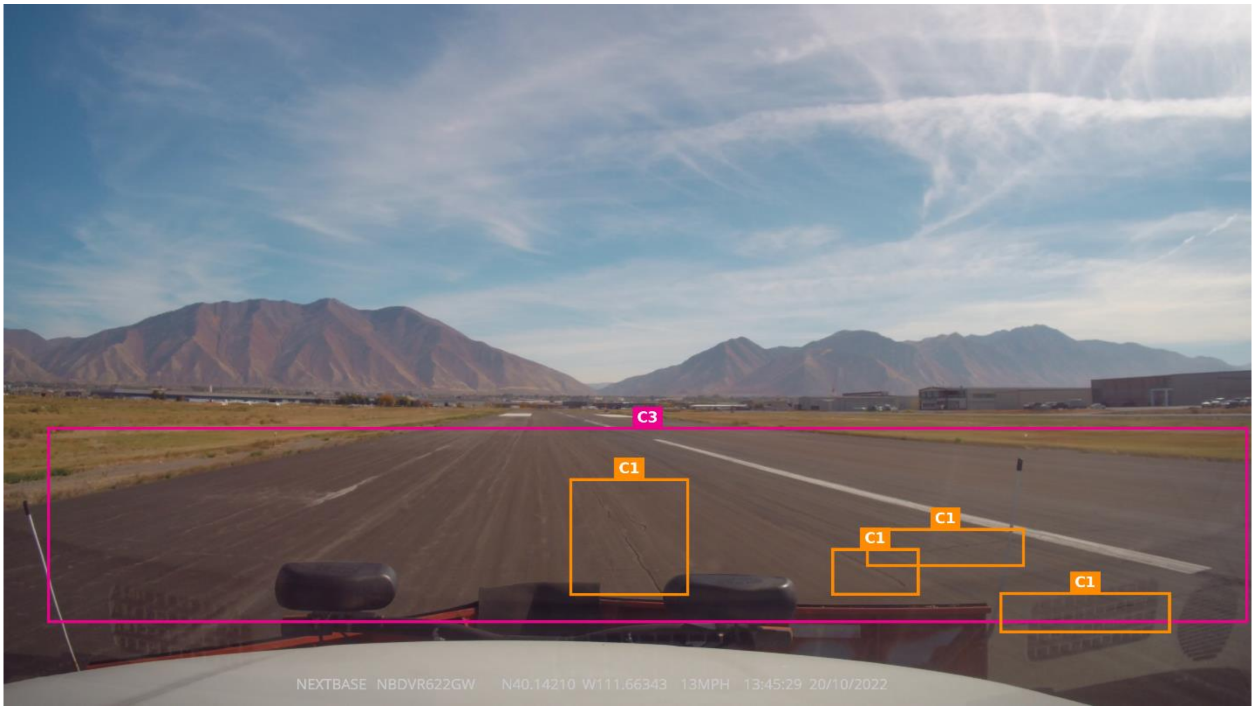
(a) (b) (a)

Figure 3.12: Rutting (a) low severity, (b) medium severity, and (c) high severity.

3.4 CVAT Annotations

With the distress types identified, the annotations process can begin. We validated several distress types from the imagery based on the inputs from the Utah Department of Transportation (UDOT) expert. Due to the nature of dashcam images being wide-view, the severity of the distress areas cannot be identified by visually examining the images. Therefore, this project will only focus on detecting the distress types. The rest of the images are annotated accordingly with a baseline reference of distresses by the UDOT expert. The annotators followed the guidelines and instructions provided by the UDOT expert to accurately label and annotate the distress areas in the pavement images.

The focus of the annotation process was to capture and label distress types effectively while acknowledging the limitations in assessing distress severity based solely on visual examination. This allows the project to gather valuable data regarding the occurrence and distribution of different distress types, which can subsequently be utilized for further analysis and decision-making in pavement maintenance and management. An example of the annotated images showcasing distress areas is depicted in Figure 3.12.


C3
C1
C1
C1
C1
NEXTBASE NBDVR622GW N40.14210 W111.66343 13MPH 13:45:29 20/10/2022

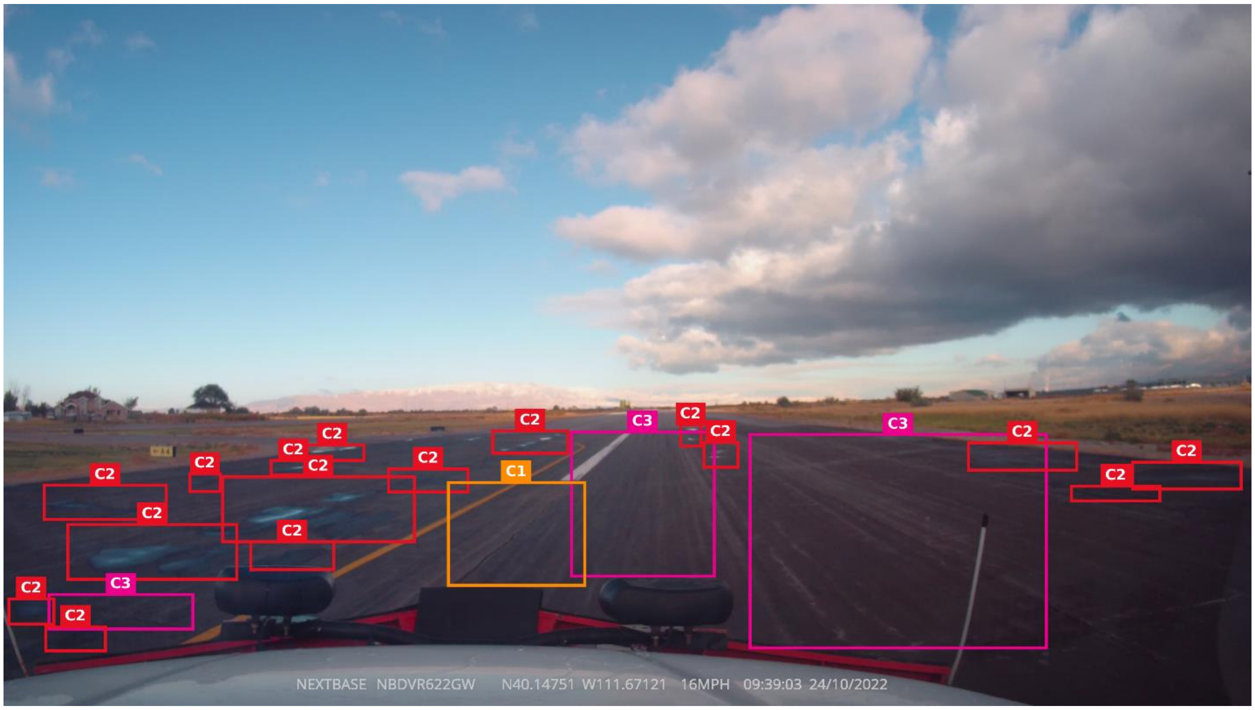

C2
C2
C3
C2
C3
C2
C2
C2
C2
C2
C2
C1
C2
C2
C2
C2
C2
C2
C3
C2
NEXTBASE NBDVR622GW N40.14751 W111.67121 16MPH 09:39:03 24/10/2022

Figure 3.13: Annotated distress types on a segment of the runway. Note: To reduce cluttering, C1 refers to longitudinal and transverse cracking, C2 refers to rutting/depression, and C3 refers to weathering.

3.5 Faster R-CNN Object Detection and Classification

The Faster R-CNN (Region-Based Convolutional Neural Network) is a powerful deep learning model commonly used for object detection. It consists of two major stages: *region proposal generation* and *object classification and localization*.

The Region Proposal Network (RPN) is employed in the first stage. RPN is a deep neural network that takes an input image and generates potential bounding box proposals for objects within the image. Bounding box proposals refer to the predicted regions or areas in an image that potentially contain objects of interest. To accomplish this, the RPN slides a small window known as an anchor across the image and predicts whether each anchor includes an object or not. The RPN uses various anchor boxes with different sizes and aspect ratios to handle objects of varying scales and shapes. These predicted proposals serve as candidate regions of interest for further analysis.

The Faster R-CNN model's second stage involves classifying and localizing objects within the proposed regions. The model utilizes the region proposals generated by the RPN and applies a

Region of Interest (RoI) pooling layer. The RoI pooling layer extracts fixed-size feature maps from each proposal, enabling the model to focus on relevant details within the proposed regions. For example, if the RoI pooling layer is set to output a fixed-size feature map of 7x7 spatial dimensions, each proposed region will be divided into a 7x7 grid of sub-regions. The RoI pooling layer then performs a max-pooling operation within each sub-region. The pooling layer with max-pooling operation downsamples the input feature maps, reducing their spatial dimensions while retaining the most prominent or significant features.

The extracted feature maps are then passed through fully connected layers, followed by two output layers: one for classifying the object category and the other for regressing the bounding box coordinates. The classification output layer assigns a class label to each proposed region, indicating the object's category. The regression output layer refines the predicted bounding box coordinates to better align with the actual object boundaries. The Faster-RCNN model architecture is shown in Figure 3.14.

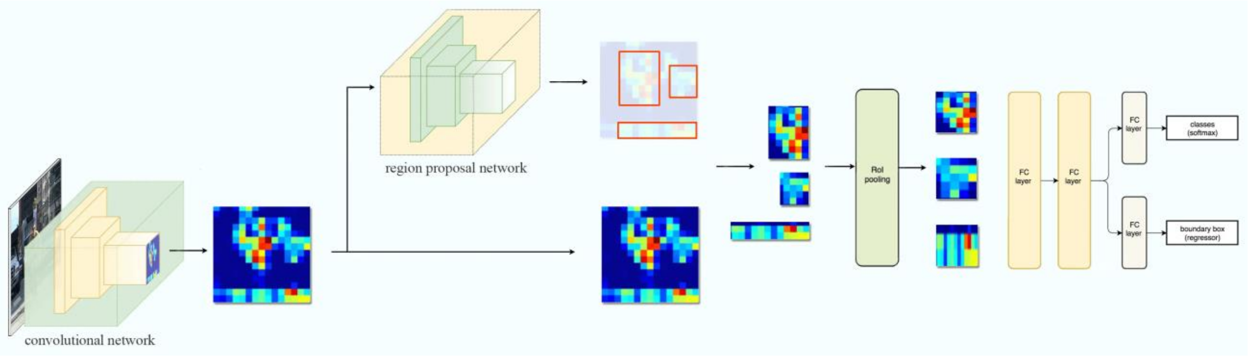


Figure 3.14: Faster R-CNN Architecture

The Faster R-CNN pipeline can be broken down into the following steps:

1) **Input image preprocessin**g: The input image is resized and normalized to ensure consistent processing.
2) **Feature extraction:** A deep convolutional neural network (usually pre-trained on ImageNet) is used to extract features from the input image. Popular choices for the backbone include VGG-16, ResNet-50, or ResNet-101. The backbone network performs a series of convolutions, non-linear activations (e.g., ReLU), pooling operations, and other operations to progressively extract hierarchical and abstract features from the input image. The features capture different levels of visual information, ranging from low-level details such as edges, textures, and color

to higher-level semantic concepts such as abstract representation of the presence of linear structures or the arrangement of distresses within the pavement.

3) **RPN:** The RPN takes the feature maps from the backbone network and generates region proposals using a sliding window approach. It outputs two values for each anchor box (a predefined set of bounding box shapes and sizes): an objectness score and the coordinates of the proposed bounding box.
4) **Non-Maximum Suppression (NMS):** To reduce the number of region proposals, NMS is applied to the RPN output. This step removes overlapping proposals with lower objectness scores, retaining only the most likely bounding boxes.
5) **RoI Pooling:** The feature maps corresponding to the retained region proposals are pooled to fixed-size feature maps using RoI pooling, allowing for variable input sizes.
6) **Classification and bounding box regression**: The pooled feature maps are passed through fully connected layers and split into two branches. One branch outputs the class probabilities using a softmax layer, while the other branch outputs the refined bounding box coordinates using a linear regression layer.
7) **Final predictions:** The class probabilities and refined bounding box coordinates are combined to produce the final object detection results.

3.6 Model Testing

To assess the effectiveness of the Faster R-CNN model, a range of performance metrics and evaluation techniques are employed. These measures offer valuable insights into the model's accuracy, robustness, and efficiency in detecting objects. Various widely adopted performance metrics and evaluation techniques are utilized in this study to gauge the model's performance in object detection tasks.

1) Precision & recall

Precision measures how well the model can find true positives (TP) from all positive predictions (TP+FP). Recall measures how well the model can find all the positives. Accordingly, their mathematical definitions are:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

where TP is true positive, FP is false positive, and FN is false negative.

2) F-beta score

The F-beta score measures the binary classification model's performance, combining precision and recall into a single metric. It is a weighted harmonic mean of precision and recall, where the weight of recall is determined by a parameter β, which controls the importance of recall relative to precision. It is defined as:

$$f_\beta = (1 + \beta^2) \times \frac{preceision \times recall}{(\beta^2 \times precision) + recall}$$

The β parameter controls the weight of recall relative to precision, with larger values of beta giving more weight to recall. In general, there are three most common values for the beta parameter:

- F0.5 score ($\beta = 0.5$) makes a precision value more important than a recall one. In other words, it focuses more on minimizing FP than minimizing FN;
- F1 score ($\beta = 1$) is the true harmonic mean of precision and recall;
- F2 score ($\beta = 2$) makes a recall value more important than a precision one. In other words, it focuses more on minimizing FN than minimizing FP.

Generally, in distress detection tasks, recall is typically considered more important than precision, as missing distress can have more severe consequences than incorrectly detecting a non-distress area as a distress.

3) Intersection Over Union (IoU)

IoU is a commonly used evaluation metric in object detection and segmentation tasks. IoU measures the overlap between the predicted bounding box or mask and the ground truth bounding box or mask for a given object, as shown in Figure 3.14. The IoU score is computed as the ratio of the intersection of the predicted and ground truth areas to the union of those areas. In other words, it measures how much the predicted and ground truth areas overlap relative to their total area.

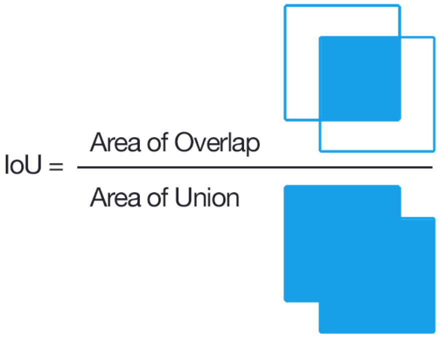


Figure 3.15: Graphical Representation of IoU

The IoU score ranges between 0 and 1, where a score of 1 indicates perfect overlap between the predicted and ground truth regions, while a score of 0 indicates no overlap at all. A threshold value is typically set for the IoU score to determine whether a prediction is considered a TP or an FP. The threshold value may vary depending on the specific application and tolerance level for FP and FN.

A TP occurs when the IoU score between a predicted bounding box and its corresponding ground truth bounding box is above a specified threshold. A high IoU score indicates that the predicted bounding box aligns well with the ground truth, denoting a correct detection.

An FP happens when the IoU score between a predicted bounding box and any ground truth bounding box is below the specified threshold. This means that the predicted bounding box does not sufficiently overlap with any ground truth bounding box, denoting an incorrect detection.

An FN is defined as a ground truth bounding box that does not have a corresponding predicted bounding box with an IoU score above the specified threshold. It means that the model failed to detect and predict the presence of an object that exists in the ground truth, resulting in a false negative.

For example, as demonstrated in Figure 3.16, by setting the threshold at 0.5, we can determine the TP, FP and FN based on the IoU score between predicted bounding box and ground truth bounding box.

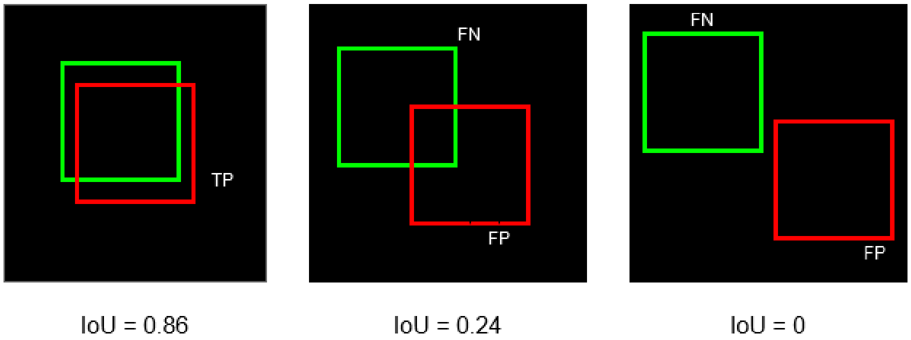


Figure 3.16: IoU thresholds for predicting TP and FP. The green bounding box is the ground truth, and the red bounding box is the predicated bounding box.

4) Precision-Recall (PR) curve

The precision-recall (PR) curve is a commonly used metric for evaluating the performance of machine learning models, particularly in cases where the dataset is imbalanced, or the cost of FP and FN is uneven. The PR curve plots the relationship between precision and recall at various confidence score thresholds. The confidence score serves as a threshold for determining whether a predicted bounding box is considered a TP or an FP based on the level of overlap with the ground truth bounding box (i.e., IoU score). Visualizing the PR curve can help us understand the trade-off between precision and recall at different confidence score thresholds. Depending on the desired balance between precision and recall, it can also help identify the optimal threshold for a given application.

5) Mean Average Precision (mAP)

Mean Average Precision (mAP) is a commonly used evaluation metric in object detection tasks. It is used to measure the average precision of a model across multiple object classes, taking into account both the precision and recall of the model.

The mAP is calculated by first computing the Average Precision (AP) for each object class. The AP for a given class is computed by first ranking the predicted bounding boxes for that class by their confidence score and then computing the precision and recall at each ranked position. The precision-recall curve for the class is then computed by plotting the precision-recall pairs at each ranked position, and the area under the curve (AUC) is computed using the all-point interpolation method according to the Pascal VOC competition. The AP for the class is then equal to the AUC of the precision-recall curve. The mAP is then calculated by taking the average of the AP values across all object classes. The mAP score ranges between 0 and 1, where a score of 1 indicates perfect performance, while a score of 0 indicates no correct detections.

3.7 Distress Location Measurements

Geolocating the distresses identified by the model predictions is crucial for decision-makers to gain a comprehensive understanding of distress densities and to identify critical pavement segments. However, accurately determining the location of distresses on the runway pavement in real-world coordinates requires expensive hardware such as stereo cameras or depth sensors. These specialized devices provide depth information, enabling precise distance measurements and scene understanding that consider spatial relationships between objects.

Although dashcams provide valuable visual information, they do not inherently capture depth information. Luckily, there are techniques available to overcome this limitation. Two computer vision techniques based on Homography are implemented to associate the identified distresses from the model predictions in pixel coordinates (obtained from image analysis) with their corresponding real-world coordinates. Homography involves warping the perspective of the image to a bird's-eye view, enabling the measurement of physically known objects within the image. By applying these techniques, the distresses detected in the images can be accurately geolocated, providing valuable spatial information for further analysis and decision-making.

In this project, we have experimented with two different approaches to establish a mapping between pixel coordinates and real-world coordinates in both x- and y-directions. The two approaches both use the white centerline marking as a reference. However, in the first approach, we utilize the known physical dimensions of the white centerline marking on the runway, while the second approach leverages the GPS information captured by the dashcam.

The known physical dimension of the white centerline marking offers a practical and visually identifiable reference point for geolocation which can establish a relationship between pixel coordinates and real-world distances. This approach can be useful in situations where GPS data may not be available or reliable. However, it relies on the visibility and consistency of the centerline markings, which can vary depending on environmental conditions and maintenance practices. On the contrary, the GPS-based approach provides a direct and objective source of location information. By relying on the GPS coordinates recorded by the dashcam, we can obtain consistent mapping values. This approach, though, is dependent on the availability of reliable GPS signals. In cases where the GPS signal is weak or disrupted, the accuracy of the mapping values may be compromised.

Before determining the mapping values, we first need to detect the white centerline. To achieve this, vanishing points must be determined first. The estimation of vanishing points in the image is accomplished using the RANSAC (Random Sample Consensus) algorithm, which is widely used for robust estimation in computer vision and geometric modeling tasks. Vanishing points represent the points in an image where parallel lines appear to converge. The RANSAC algorithm follows an iterative process to estimate the vanishing point from a set of candidate lines. It randomly selects a subset of lines and fits a model to these lines. This model is then evaluated against the remaining lines, and the number of inliers (lines that agree with the model) is counted. The process is repeated multiple times, and the model with the highest number of inliers is considered the best estimate for the vanishing point. We utilized the algorithm developed by Chaudhury et al. (2014) for vanishing point estimation.

Once the vanishing point is identified (an example is shown in Figure 3.17), the trapezoid edges can be calculated to encompass the arbitrary position of the runway, as shown in Figure 3.18. The corners of the trapezoid serve as source points, while the destination points are defined as the four corners of the desired new warped image. In this project, we transformed and warped the images to a dimension of 600 x 700 pixels. By performing this perspective transform, images undergo a distortion that aligns the white centerline markings with a more top-down view. This transformation facilitates the detection of the centerline markings using various image processing techniques such as image thresholding. Once detected, a rectangular bounding box representing the white centerline marking is drawn on the image.

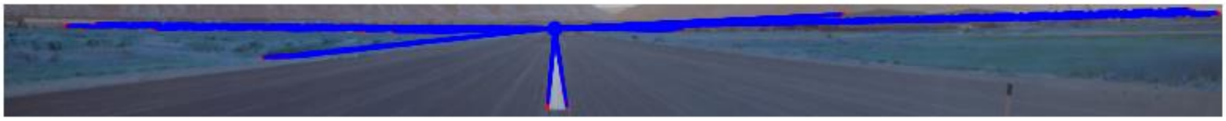


Figure 3.17: Vanishing Point Estimation

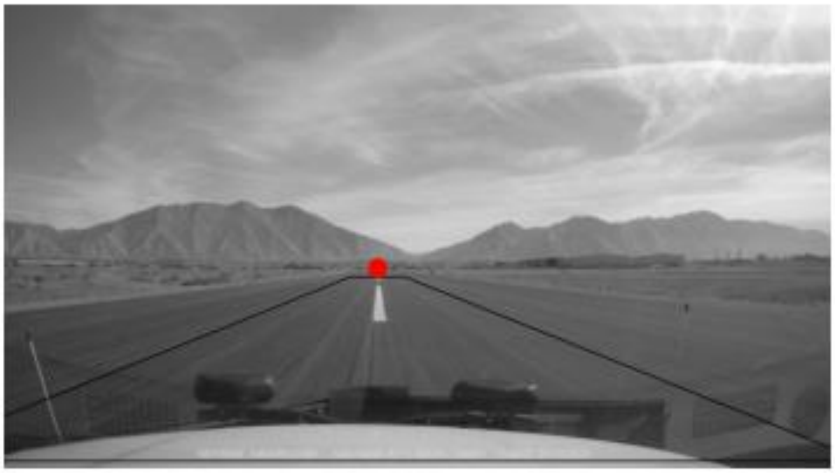


Figure 3.18: Trapezoid Edges Covering Portions of a Runway Segment

3.7.1 Y-Direction Mapping Value

The first approach to estimate a mapping value in the y-direction utilizes multiple random white centerline markings and their known physical dimensions. According to the Advisory Circular 150/5300-13, white centerline markings on runways serving small airplanes with approach speeds of less than 50 knots should have a length of 120 feet and a width of 36 inches. By leveraging this information and analyzing multiple random frames that contain a clear view of the white centerline marking, a mapping value can be established. Specifically, we measure the length of the white centerline marking in pixels in the y-direction and find the corresponding ratio in real-world coordinates in feet. Using 40 random images with a clear view of at least one white centerline marking, the median ratio of pixel to real-world coordinate is about 1.8:1. Due to the random iterative process of RANSAC, each image was tested 10 times and the median of the results was taken.

With the GPS information available, we can rely on a more accurate source of location input. Therefore, in the second approach, once the transformed images are obtained, the next step is to determine the number of pixel shifts between consecutive frames (e.g., Figure 3.19). This information, along with the approximate distance covered in the real-world coordinates during the two frames obtained from GPS points, can be utilized to establish a mapping value. To estimate the distance traveled between two consecutive frames, we converted the default WGS84 coordinate reference system to the corresponding UTM zone (i.e., 12 N for Utah). We then used the center of the rectangular bounding box representing the white centerline marking and measured the amount of pixel shift between two consecutive frames in the y-direction. Using multiple consecutive frames, we determined the ratio of pixel to real-world coordinate to be approximately 0.4:1.

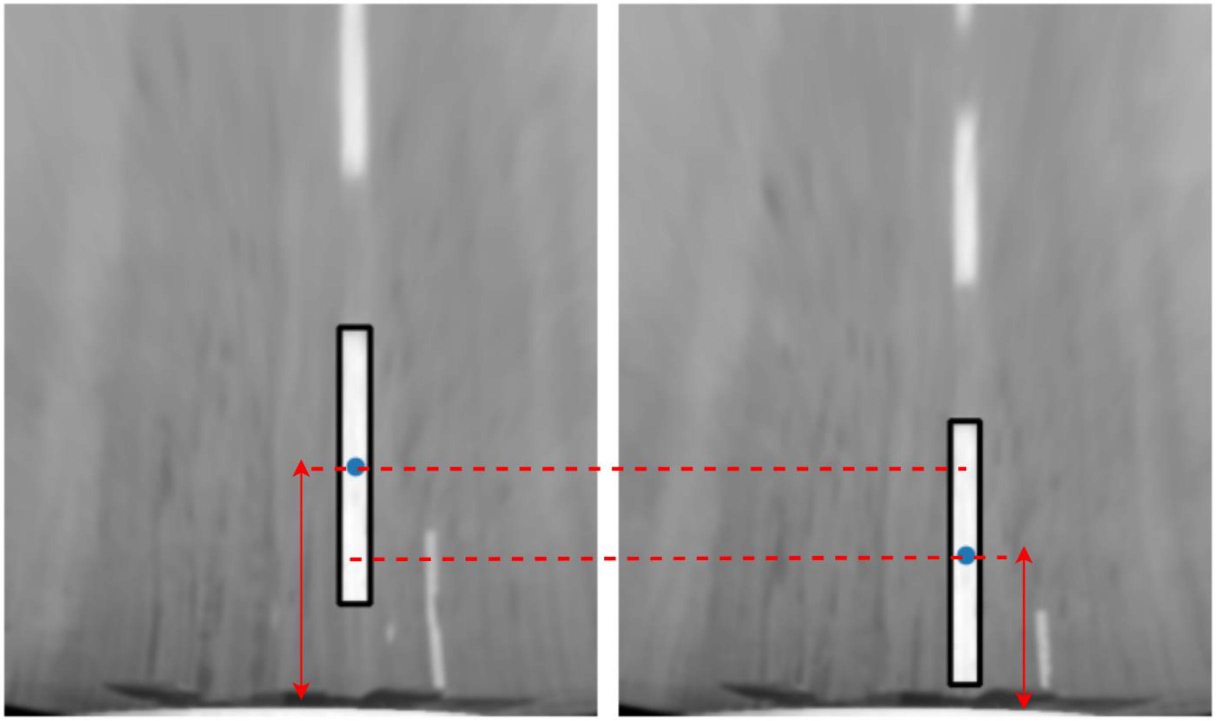


Figure 3.19: Warped perspective of two consecutive frames showcasing the pixel shift of the center of the white centerline marking.

These two approaches offer distinct advantages and cater to different scenarios. The first approach relies on visually identifiable centerline markings and can provide reasonably accurate geolocation results when GPS data is unavailable. It serves as a practical solution for situations where visual markers can be relied upon.

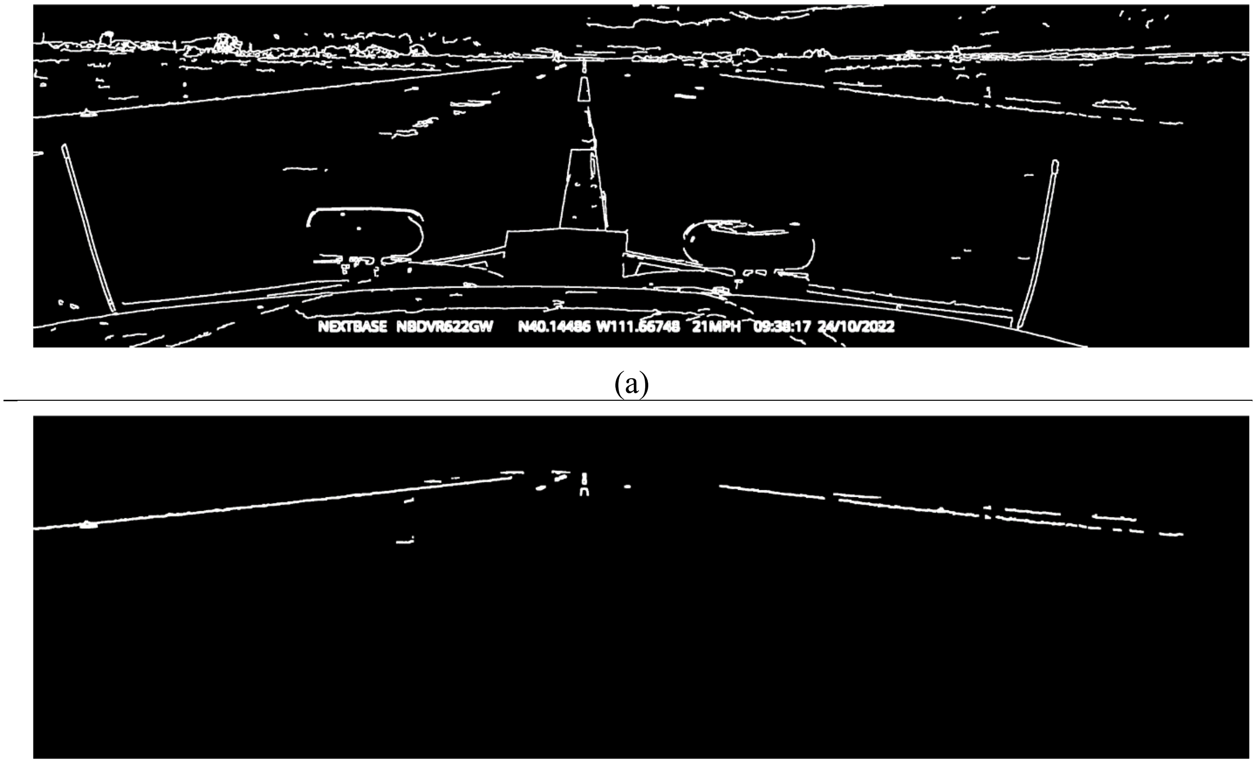
On the other hand, the second approach, leveraging reliable GPS information from the dashcam, offers a more accurate and robust source of location input. It eliminates the dependency on centerline markings and directly correlates pixel shifts between frames to real-world distances. This approach is particularly advantageous when continuous GPS data is available, allowing for accurate mapping and analysis over time. Moreover, the second approach demonstrates flexibility in adapting to various runway scenarios, even those lacking well-defined or visible markings. Given its higher accuracy, continuous monitoring capabilities, and adaptability, the second approach is the preferred choice for estimating the mapping value in this project.

3.7.2 X-Direction Mapping Value

To find a mapping value in the x-direction, an alternative approach is employed by leveraging another known dimension in the runway: the width of the runway. Without the need to determine the vanishing point, this approach involves the following steps:

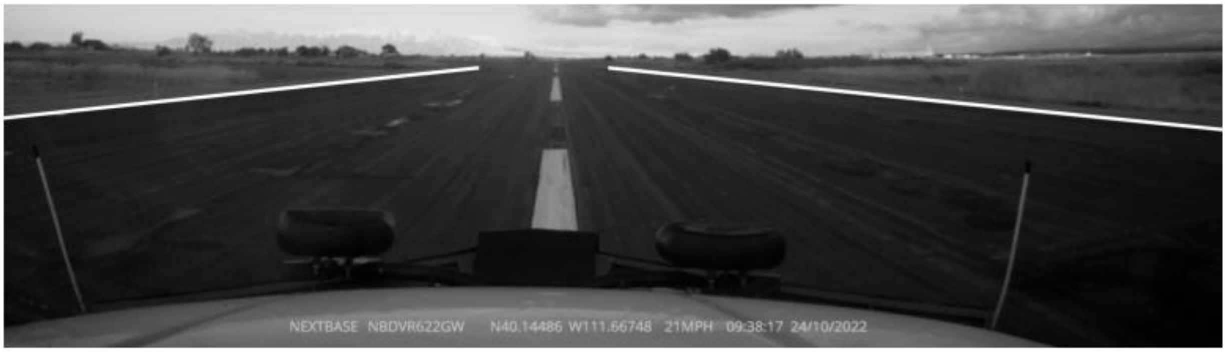
1) **Runway Edge Detection:** Canny edge detection is applied to the images, preceded by a Gaussian blur to smooth out the edges. This process enhances the visibility of the runway edges.
2) **ROI Masking:** A region of interest is defined to focus only on the area where the runway edges are expected. This helps filter out irrelevant edges and reduces computational complexity.
3) **Extraction of Runway Edges:** The defined ROI acts as a mask, allowing the extraction of lines that correspond to the runway edges. These lines represent the boundaries of the runway.
4) **Line Representation of Edges:** The Probabilistic Hough Line Transform is employed to construct a line representation of the extracted runway edges. This transform identifies the most prominent lines representing the edges, taking into account line segments with sufficient support.

Figure 3.20 shows the steps implemented above.


NEXTBASE NBDVR622GW N40.14486 W111.66748 21MPH 09:38:17 24/10/2022

(a)

(b)


NEXTBASE NBDVR622GW N40.14486 W111.66748 21MPH 09:38:17 24/10/2022

(c)

Figure 3.20: Estimation of the runway edges where (a) shows the result of the Canny Edge detection, (b) shows the ROI masking applied to the edges, and (c) shows the line representation of the extracted runway edges by using probabilistic Hough Line Transform.

The extracted lines representing the runway edges are used as the source points for warping the image's perspective, while the destination points are kept the same as before for a 600 x 700 pixel warped image. Therefore, the warped image is aligned with the runway width allowing a bird's-eye view perspective of the runway, as depicted in Figure 3.21.

According to the runway design standard specified in the Advisory Circular 150/5300-13 for airplane design group III, the runway width is 100 ft. This known information is utilized to establish the corresponding mapping value between pixels and real-world measurements in the x-direction. With the runway width of 100 ft and the transformed image showing the runway from a top-down perspective, the mapping value can be determined. This mapping value enables the conversion of pixel coordinates in the x-direction to their corresponding real-world distances. After testing multiple images, the ratio of pixel to real-world coordinate in the x-direction is measured to be about 0.16:1.

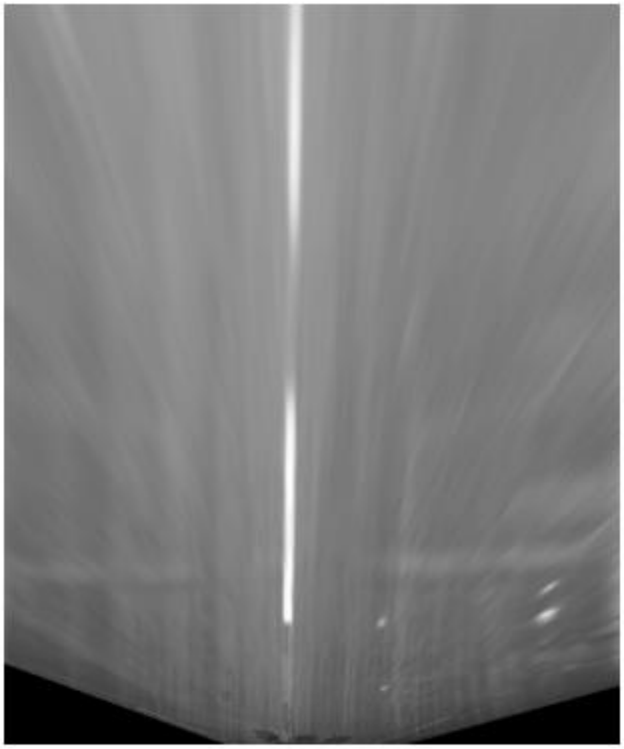


Figure 3.21: Bird's Eye View of the Entire Width of a Runway Segment

By incorporating both the mapping value in the y-direction and the mapping value in the x-direction, a comprehensive mapping between pixel coordinates and real-world measurements is established, facilitating precise geolocation and analysis of the distresses on the runway.

4.0 DATA DESCRIPTION

We collected videos using a NEXAR dashcam, as described in Section 3.1. Two airports in Utah, namely Spanish Fork, and Provo, were selected to obtain a substantial amount of video footage. These airports allowed us to gather multiple hours of footage capturing the desired scenarios. Video frames were extracted from the recorded videos for further processing and analysis. To ensure accuracy and consistency, sample points within each predefined segment were identified. We used a 5 meters distance between the sample points following the procedure described in Section 3.2. These sample points serve as the basis for training and validating our model, enabling us to detect and analyze pavement distress accurately. However, it's important to note that the extraction process of frames from the large-sized 4K videos was a time-consuming task. Due to the high-resolution nature of the videos, the extraction process required several days to complete. This was necessary to ensure that all frames were extracted correctly and available for subsequent analysis.

After conducting our initial investigation, we discovered that the footage obtained from Provo airport did not meet our reliability criteria. It was observed that the vehicle did not traverse the main portions of the airport, and the majority of the footage captured was from outside of the airport area. Due to these limitations, we made the decision to exclude the Provo airport dataset from our analysis.

The Spanish Fork airport comprises a single runway and is organized into 100 segments, as illustrated in Figure 3.3. Each segment encompasses an area of 6,500 square feet, measuring 50 feet by 130 feet. The segments were carefully examined through meticulous surveying, and a subset of them was chosen for detailed evaluation. In this study, 377 images from 67 sections were selected and labeled using the CVAT tool, encompassing four distinct types of distress: longitudinal and transverse cracking, weathering, rutting, and depression, as outlined in Table 4.1. Note that some distress types, such as rutting and depression, exhibited similarities that made differentiation challenging. To address this, we combined rutting, and depression distresses into one group. Despite these challenges, our dataset remains a valuable resource for the development and evaluation of pavement distress detection models.

While the dataset encompasses a variety of pavement distress types, it is important to acknowledge that certain distress types may be underrepresented or difficult to discern. Nonetheless, the labeled images generated through the CVAT tool serve as a significant benchmark for future studies in this field. Researchers can utilize this dataset to evaluate the performance of their pavement distress detection models and potentially improve upon the existing methodologies by providing a diverse range of distress types, albeit with certain complexities. Our dataset offers researchers a realistic representation of the challenges faced in pavement distress detection.

Table 4.1: Number of Distresses Identified

Distress Type	Size
Longitudinal and Transverse Cracking	1,516
Weathering	624
Rutting	344
Depression	250

During this project, we encountered a challenge in the limited availability of a dataset for training our pavement distress detection model. To address this challenge, we employed the technique of image augmentation to expand our dataset and enhance the model's ability to recognize and generalize to new data. Image augmentation involves applying various transformations to existing images, thereby creating new training samples with different variations of the same pavement distresses. This technique allows us to generate a more diverse and representative training dataset, ultimately improving the model's performance.

Our data augmentation pipeline encompassed several transformations, including random image cropping, random bounding box cropping, horizontal flipping, random brightness changes, and image enhancement techniques. By randomly cropping images and bounding boxes, we introduced variations in the spatial location of the distresses within the images. Flipping operation further augmented the dataset by simulating different orientations and perspectives. Adding random brightness and image enhancement techniques also helped introduce variations in pixel values, making the model more robust to noise and improving its generalization capabilities. Additionally, we employed the strategy of cropping the images at the horizon, retaining only the bottom 50% of the image. This image size reduction improved training performance and aided in reducing memory consumption, enabling more efficient training.

To ensure the quality and consistency of our training data, we implemented additional preprocessing steps within our data augmentation pipeline. This included data normalization and standardization, ensuring the images had a consistent scale and brightness level. By standardizing the images, we eliminated potential biases introduced by variations in illumination or color distribution, thereby enhancing the model's accuracy.

5.0 RESULTS AND ANALYSIS

The training and testing of the developed model were conducted in a high-performance server equipped with a dedicated GPU. The computing environment specifications are shown in Table 5.1.

Table 5.1: Model Training Environment Specification

Environment	Configuration
OS	Ubuntu 22.04.1 LTS
CPU	AMD EPYC 7H12 64-Core Processor
GPU	NVIDIA A100 80GB
Memory	1.5TB
Programming language	Python 3.10.6
Deep learning framework	PyTorch 2.0
CUDA	Version 11.8

Furthermore, the inputs to the model and the tuned hyperparameters are shown in Table 5.2.

Table 5.2: Model Parameters

Parameter	Setting
Image size	3840 × 2160 pixel
Weight initializer	COCO
Optimizer	Adam
Learning scheduler	1cycle learning rate policy
Learning rate	1.33E-4
Epochs	500
Batch size	32

A total of 377 annotated 4K images captured by the dashcam were used as input to our model. We split the images into 80% training and 20% testing datasets. Table 5.3 shows the number of distresses in each dataset.

Table 5.3: Training and Testing Dataset Distress Distributions

Dataset	Longitudinal and Transverse Cracking	Depression/Rutting	Weathering	Total	Ratio
Train	1187	493	499	2179	0.80
Test	329	101	125	555	0.20
Total	1516	594	624	2734	1

Each image in the training dataset was augmented according to the procedure laid out in Section 4.0 with a certain probability. The testing dataset was left untouched. Figure 5.1 illustrates how different variations of an original image through the augmentation pipeline generates new images that are used during training. The original images underwent a series of transformations with varying probabilities. These transformations include pixel-level or spatial-level changes. In cases where spatial transformations are applied, such as cropping a random portion of the image, the original annotations are also adjusted automatically to maintain consistency. This ensures that the annotations align with the transformed images and accurately represent the corresponding distresses.

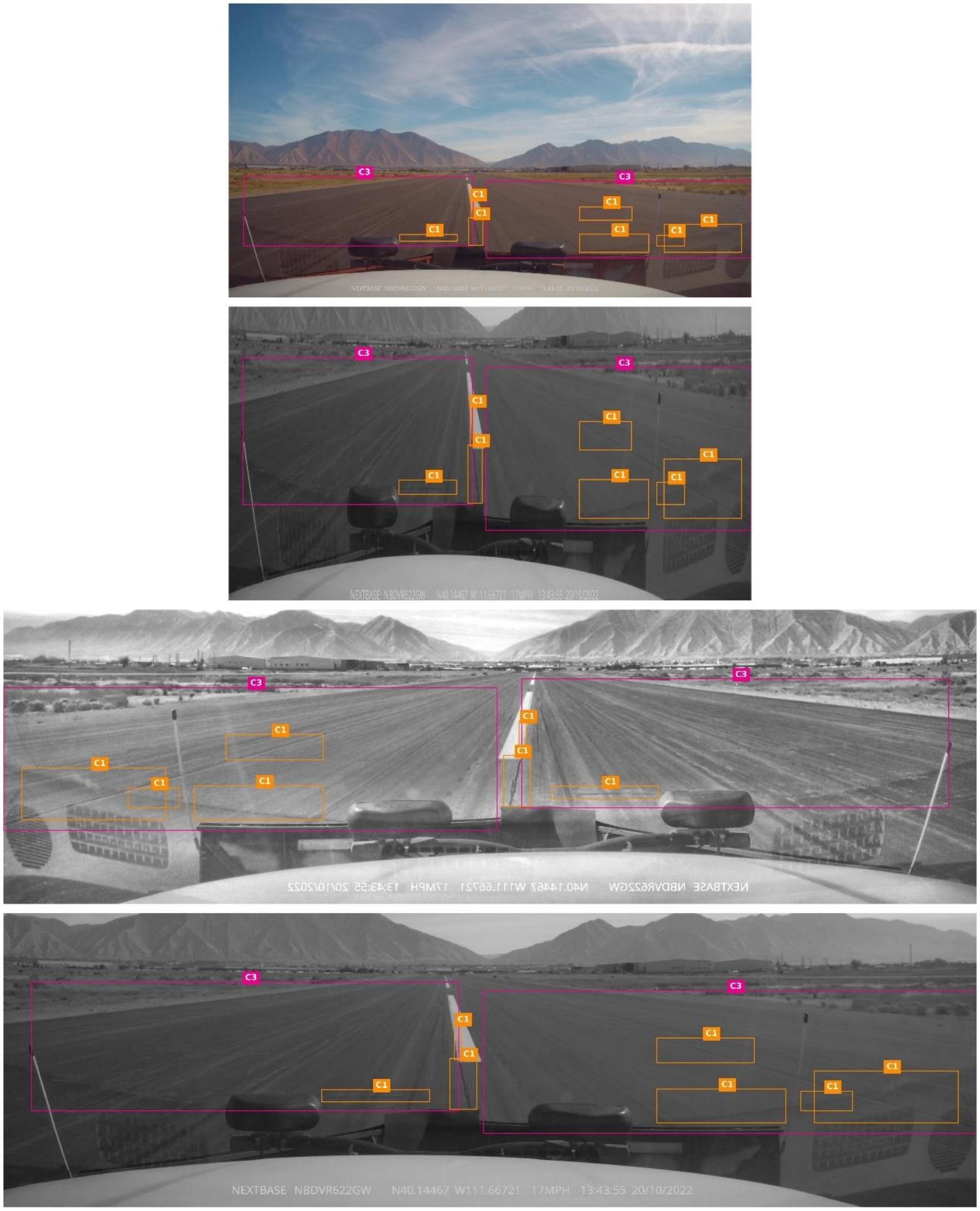

C3
C3
C1
C1
C1
C1
C1
C1
C1
NEXTBASE NBDVR622GW N40.14467 W111.66721 17MPH 13:43:55 20/10/2022

Figure 5.1: An example of three randomized augmentation pipeline on the original image. Note: To reduce cluttering, C1 refers to longitudinal and transverse cracking, C2 refers to rutting/depression, and C3 refers to weathering.

Utilizing pre-trained models for transfer learning is a valuable technique in pavement distress detection. In our study, we initialized the weights with a pre-trained Faster R-CNN model that had been trained on the COCO dataset for object detection. This initialization allowed our model to benefit from the learned features and knowledge captured by the pre-trained model on a large-scale dataset. By leveraging these pre-trained weights, the model can extract meaningful and generalizable features from the pavement distress images, enhancing its ability to recognize and classify distresses accurately. The pre-trained Faster R-CNN model used a ResNet-50 backbone, which is a deep convolutional neural network with 50 layers, known for its excellent performance in image classification tasks.

Certain adjustments were necessary to adopt the pre-trained Faster R-CNN model, which was originally trained for object detection on the COCO dataset (containing 80 classes) to the pavement distress detection task. Therefore, the head of the pre-trained model, which represents the final layers responsible for producing the desired output prediction, needed to be replaced. In the Faster R-CNN model, the last layers consist of two fully connected layers with input and output dimensions of 1,024, along with a box regressor and a SoftMax layer (see Figure 3.15). The goal was to generate predictions specific to the pavement distress detection task based on the learned features from the feature extraction layers. Thus, the classification layer in the new head had an input dimension of 1,024 and an output dimension of 4, representing the three distress types plus the background. On the other hand, the bounding box regression layer had an input dimension of 1,024 and an output dimension of 16, accounting for the four classes multiplied by the four coordinate values. These adjustments enabled the pre-trained model to detect and classify pavement distresses in the target task effectively. The learning scheduler is set to "1cycle learning rate policy." The *1cycle learning rate policy* is a technique that adjusts the learning rate during training, cycling between lower and higher learning rates within each epoch. This approach is designed to accelerate convergence and improve generalization. The learning rate is set to 1.33E-4, and the model is trained for 500 epochs with a batch size of 32 (i.e., training examples).

The model was trained on the computing environment, as detailed in Table 5.1. The total time taken for the model to finish training and testing was 11 hours (approximately 80 seconds per epoch). During the training phase, we closely monitored the training loss to evaluate the model performance and learning capability. Figure 5.2 shows the trend of training loss over 500 epochs. From the graph, we observe a consistent decrease in the training loss as the number of epochs increases. This downward trend indicates that the model is effective in learning and adapting to the characteristics of various pavement distresses. It is worth noting that the training loss has not yet reached a plateau or flattened out completely. This suggests that there is still room for improvement and fine-tuning of the model.

However, it is important to note that solely relying on the training loss as an evaluation metric can be misleading. While a decreasing training loss indicates that the model fits well to the training data, it does not guarantee good performance on new, unseen data. Overfitting is a common concern where the model becomes too specialized to the training data and fails to generalize well to unseen examples.

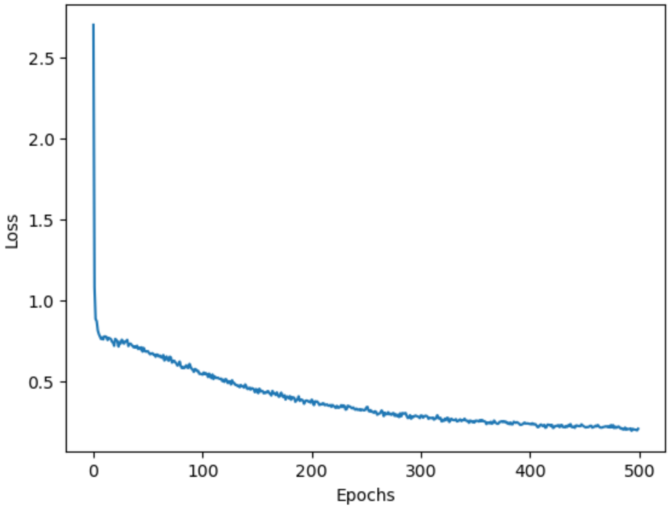


Figure 5.2: Training Loss Over 500 Epochs

The final model is tested on the testing dataset. Here, the model might predict multiple bounding boxes for distress in the image with an associated confidence score. The score is a combination of the predicted object classification score and the regression score for the bounding boxes. We ignore any prediction score less than 0.5 to filter out bad predictions. We also apply NMS on the overlapping predicted bounding boxes and only use the predictions with the highest score. To assess the model's performance, we utilized an IoU threshold of 0.5, which required a minimum overlap of 50% between the predicted bounding box and the ground truth bounding box to discern a TP. The precision results are depicted in Figure 5.3. It becomes evident that the model achieved higher precision, specifically for weathering. These distresses were consistently predicted with a higher degree of precision. It is important to note that the model struggled to effectively learn and predict depression/rutting distresses, as well as longitudinal and transverse cracking. This discrepancy in performance suggests that our model may possess a higher proficiency in detecting certain types of pavement distress while encountering challenges in accurately identifying and classifying other types.

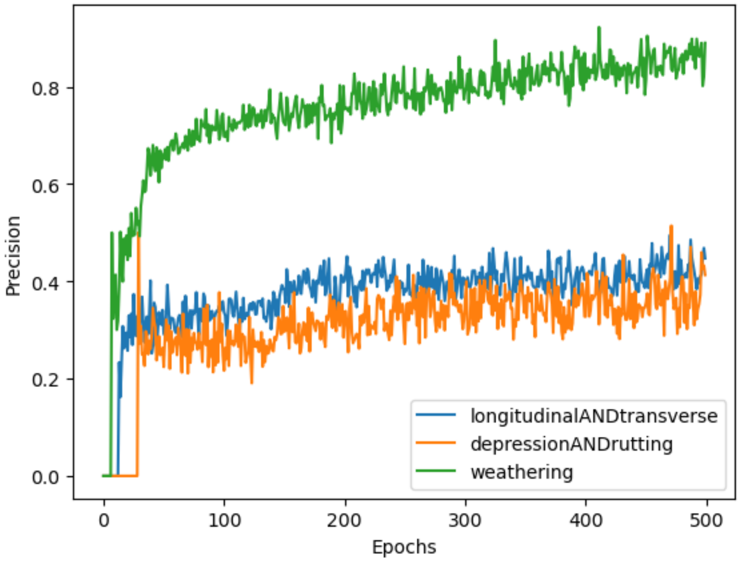


Figure 5.3: Precision Performance of the Model in Identifying Distress Types

Using the same IoU threshold of 0.5, we also evaluated the F2 score. F2 score measures the balance between precision and recall by considering the beta value of 2. By placing more weight on recall, the F2 score prioritizes the detection of distress instances while still accounting for the precision of the model's predictions. This metric is especially valuable in distress detection applications, as it highlights the significance of capturing as many TP distresses as possible to prevent potential pavement deterioration and associated risks. Figure 5.4 illustrates the F2 score of the model. Similar to the precision results, the result revealed a similar pattern for different types of pavement distresses. The F2 score indicated that the model achieved higher scores for detecting weathering, while the performance for depression/rutting and longitudinal and transverse cracking was relatively low.

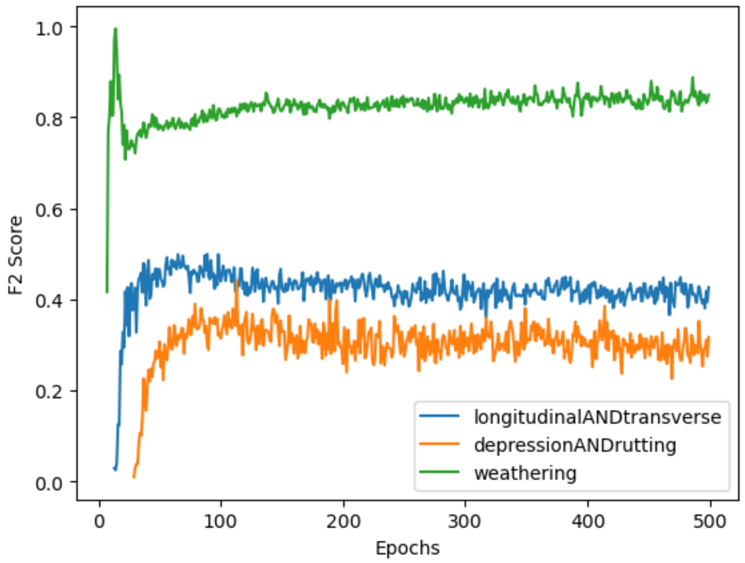


Figure 5.4: F2 Score Performance of the Model in Identifying Distress Types

Using precision or F2 score alone as the evaluation metric may not provide a comprehensive understanding of the model's performance in pavement distress detection. Precision alone does not consider the model's ability to capture all actual positive instances or the trade-off with FN. Meanwhile, the F2 score is specifically tailored to address the importance of recall in distress detection tasks and prioritize the identification of TP distresses. We can analyze the model's performance in distress detection more comprehensively, considering the precision-recall relationship across different IoU thresholds.

We provide a toy example here to explain the precision-recall relationship further. Consider the detection results shown in Figure 5.5, which contains 12 detection (red) and 9 ground truths (green). A detection is TP if the IoU is greater than 50%. For multiple detections of the same object, a detection with the highest confidence score is labeled TP, and all other detections are marked as FPs, provided that the detection has an IoU greater than 50%. The TP and FP results are tabulated in Table 5.4. Table 5.4 is sorted by confidence score in descending order. The precision and recall are calculated based on the CumTP and CumFP as discussed in Section 3.6, and the precision-recall curve can be drawn.

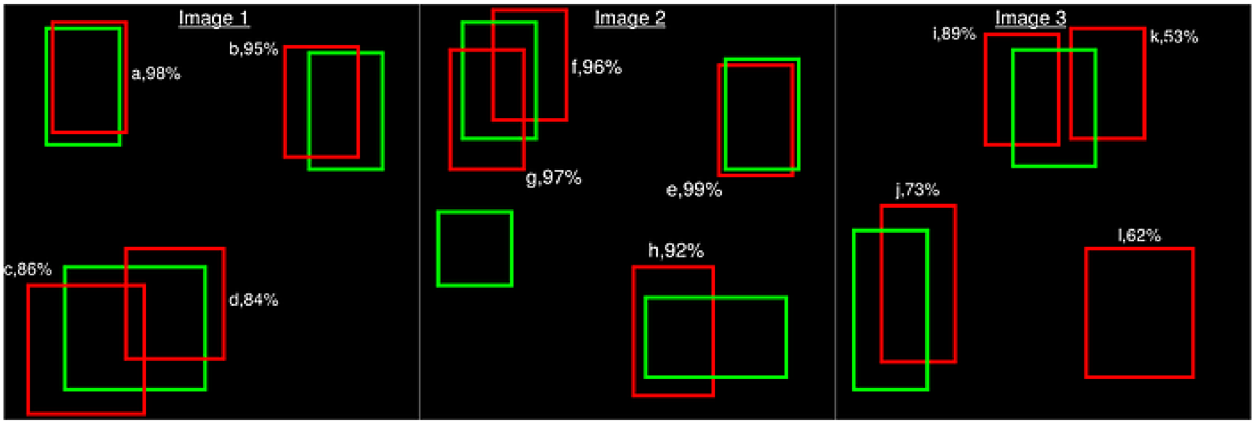


Figure 5.5: A model detecting objects of the same class and the confidence score.

Table 5.4: A toy example for explaining precision and recall. IoU threshold is set as 50%.

Detection	Confidence Score	IoU	TP	FP	CumTP	CumFP	Precision	Recall
e	99	97	1	0	1	0	1.00	0.11
a	98	92	1	0	2	0	1.00	0.22
g	97	87	1	0	3	0	1.00	0.33
f	96	82	0	1	3	1	0.75	0.33
b	95	73	1	0	4	1	0.80	0.44
h	92	48	0	1	4	2	0.67	0.44
i	89	66	1	0	5	2	0.71	0.56
c	86	47	0	1	5	3	0.63	0.56
d	84	42	0	1	5	4	0.56	0.56
j	73	54	1	0	6	4	0.60	0.67
l	62	45	0	1	6	5	0.55	0.67
k	53	38	0	1	6	6	0.50	0.67

In the example provided in Table 5.4, we can observe the precision and recall values at different stages of the detection process. Initially, when considering the first detection, we have one TP and no FP, resulting in a precision of 100% (1 TP / 1 detection) and a recall of 11% (1 TP / 9 ground truths). As we progress through the detections, additional FPs are introduced. This leads to a decrease in precision because the number of FP increases while the number of TP remains the same or increases at a slower rate.

Increasing the IoU threshold can lead to a decrease in the number of TP and an increase in FN, resulting in lower recall values. However, it may also result in higher precision since the detections are more likely to be accurate when there is a higher degree of overlap. On the contrary, decreasing the IoU threshold can make a detection to be considered TP, as it requires a lower degree of overlap between the predicted and ground truth bounding boxes. This can increase the number of TP and potentially improve recall. However, it may also lead to more FN, which can lower precision.

Analyzing the precision-recall trade-off at different IoU thresholds helps determine the optimal threshold for a specific application. Figure 5.6 provides insights into the trade-off between precision and recall at IoU threshold of 0.5. Upon analyzing the curve, we observed a rapid decrease in precision as recall increases for depression/rutting and longitudinal and transverse cracking. This trend implies that the model's performance may be limited in accurately identifying and classifying pavement distresses. What this means is that as the model tries to increase its recall (i.e., minimize FN), it starts to incorrectly classify more negative instances as positive, thus increasing FP and reducing precision. In other words, the model may have difficulty distinguishing between true pavement distresses and other objects or background noise, resulting in incorrect classifications. In general, an optimal precision-recall curve would exhibit high precision values for a wide range of recall values.

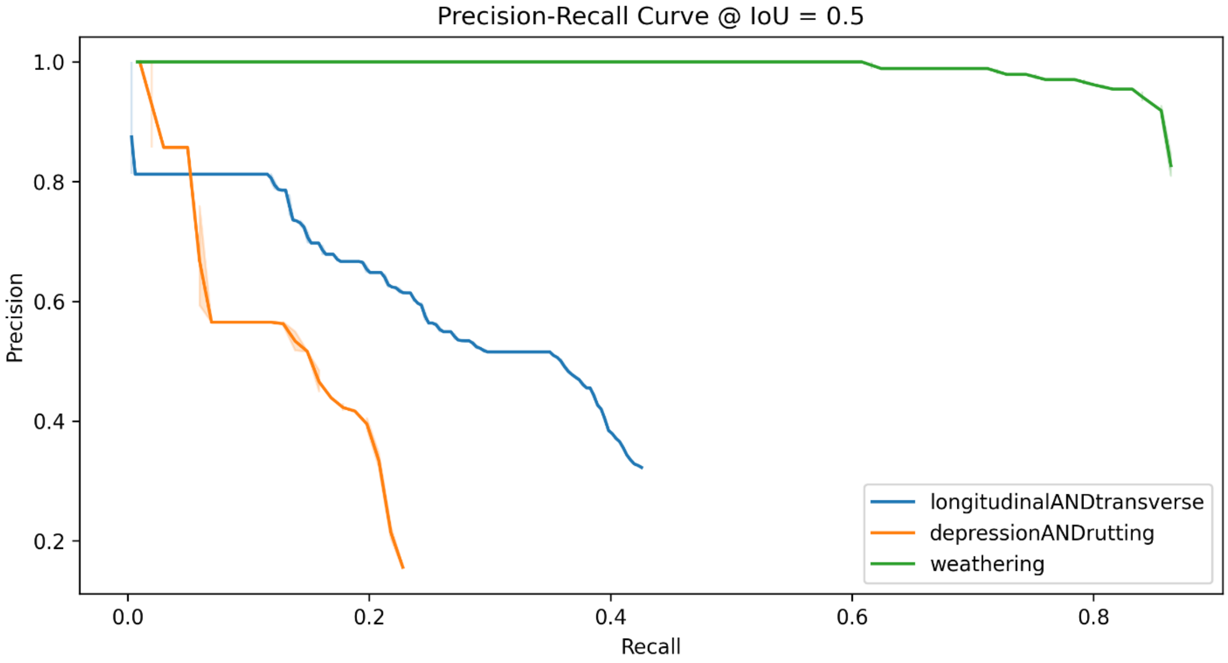


Figure 5.6: Trade-Off Between Precision and Recall at Cut of IoU Threshold of 0.5

We also computed mAP, which serves as an aggregate metric for evaluating the model's performance across multiple categories. The mAP provides a comprehensive assessment by considering both precision and recall at different IoU thresholds. In our evaluation, we again focused on IoU threshold of 0.5. The mAP scores presented in Table 5.5 reveal the model's performance for different types of pavement distresses.

The evaluation metrics used so far at the IoU threshold of 0.5 resulted in relatively low scores. These low scores indicate that there is a significant number of FN (missed distress detections) and FP (incorrectly identified non-distress regions or types) in the model's predictions. The implication of these low scores is that the model may not be reliable or effective enough in detecting and accurately delineating distress in the pavement. This could be attributed to factors such as limitations in insufficient training data or challenges in distinguishing distresses from similar features or noise in the images.

Table 5.5: Mean Average Precision at IoU of 0.5

Distress Types	**mAP @ IoU 0.5**
Longitudinal and Transverse Cracking	0.267071
Rutting/Depression	0.124120
Weathering	0.849097

To capture distress instances that may have only partial overlap with the ground truth, such as smaller distresses or distresses that are partially occluded, the IoU threshold can be relaxed. By reducing the IoU threshold, we allow for more lenient criteria for considering a predicted bounding box as a correct detection. Figure 5.7 visually illustrates the impact of varying IoU thresholds on the mAP metric. We can observe that as the IoU threshold increases, the mAP scores decrease for all types of distresses. Higher IoU thresholds require a stricter overlap between the predicted bounding boxes and the ground truth.

The mAP values at different IoU thresholds provide insights into the model's performance regarding the precision and recall trade-off. Lower IoU thresholds, such as 0.1, result in higher mAP values, indicating that the model captures a larger number of TP instances but with a higher number of FP. On the other hand, higher IoU thresholds, such as 0.7 and above, show decreased mAP values, indicating that the model becomes more selective in its predictions, resulting in a lower number of FP but potentially missing some TP instances. This suggests a trade-off between capturing more instances with lower precision and capturing fewer instances with higher precision as the IoU threshold increases.

Considering the specific requirements and trade-offs of our pavement distress detection application, we recommend using an IoU threshold of 0.3. This threshold strikes a balance between capturing distresses with a reasonable level of overlap and maintaining a satisfactory precision-recall balance. The precision-recall curve for an IoU threshold of 0.3 is shown in Figure 5.8, along with its mAP scores listed in Table 5.6.

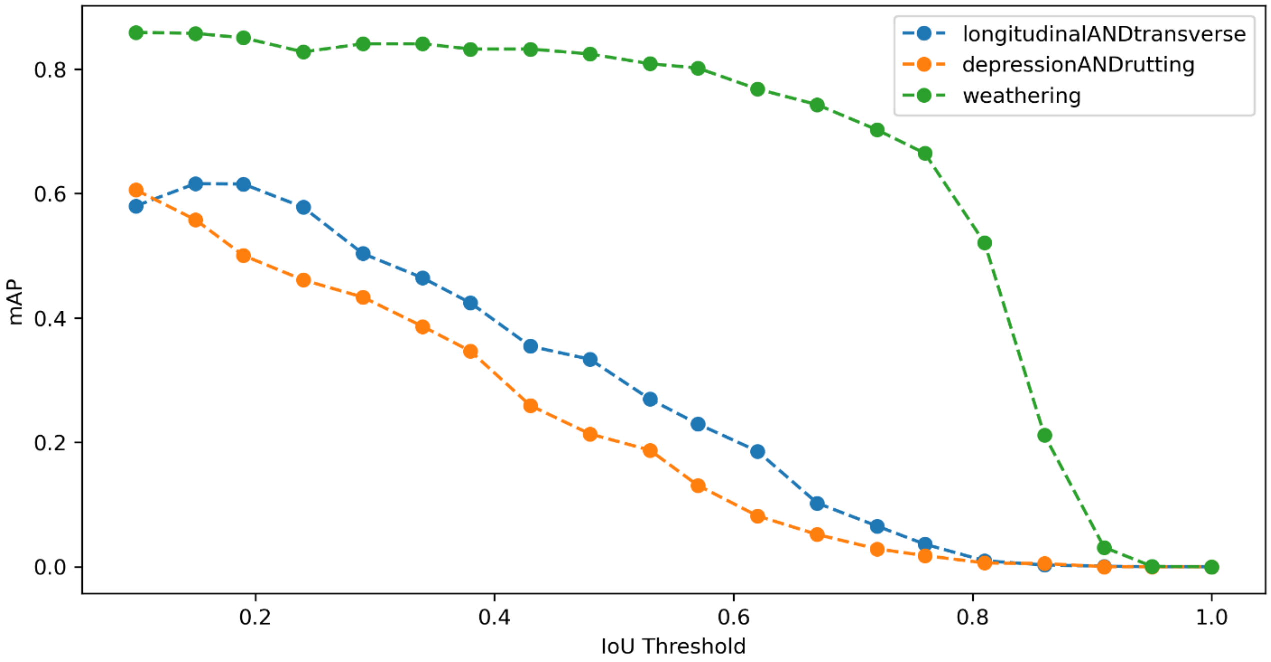


Figure 5.7: The Effect of Different IoU Thresholds on mAP

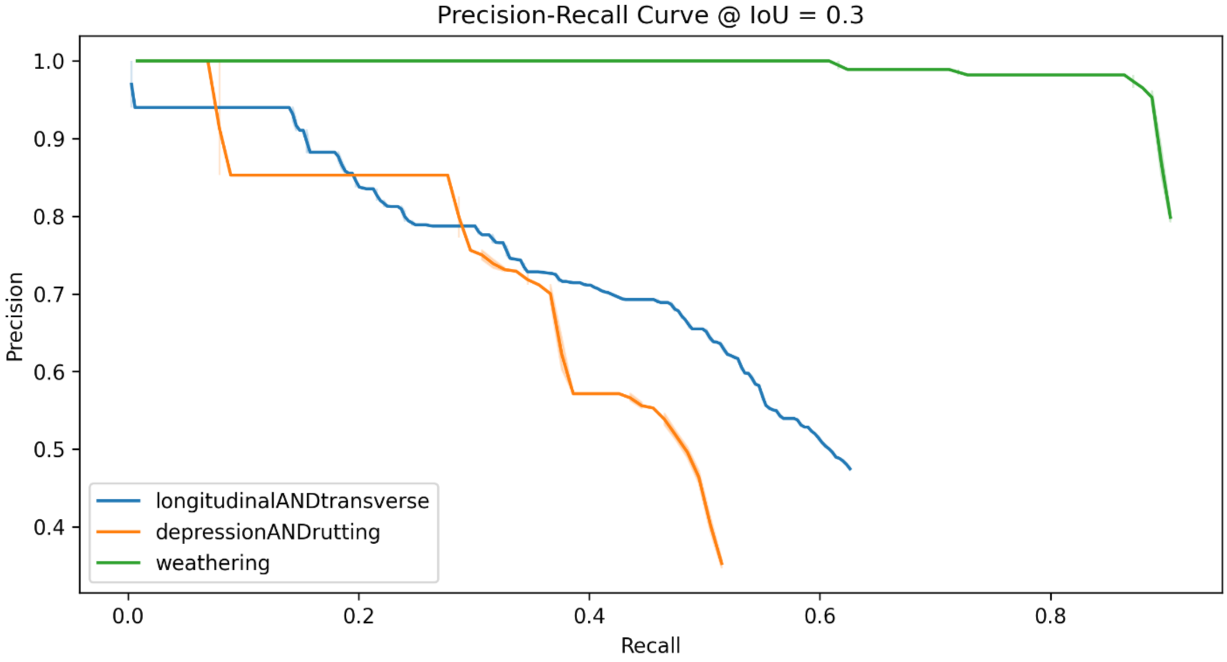


Figure 5.8: The trade-off between precision and recall at the cut of the IoU threshold of 0.3.

Table 5.6: Mean Average Precision at IoU of 0.3

Distress Types	**mAP @ IoU 0.3**
Longitudinal and Transverse Cracking	0.503336
Rutting/Depression	0.433312
Weathering	0.840619

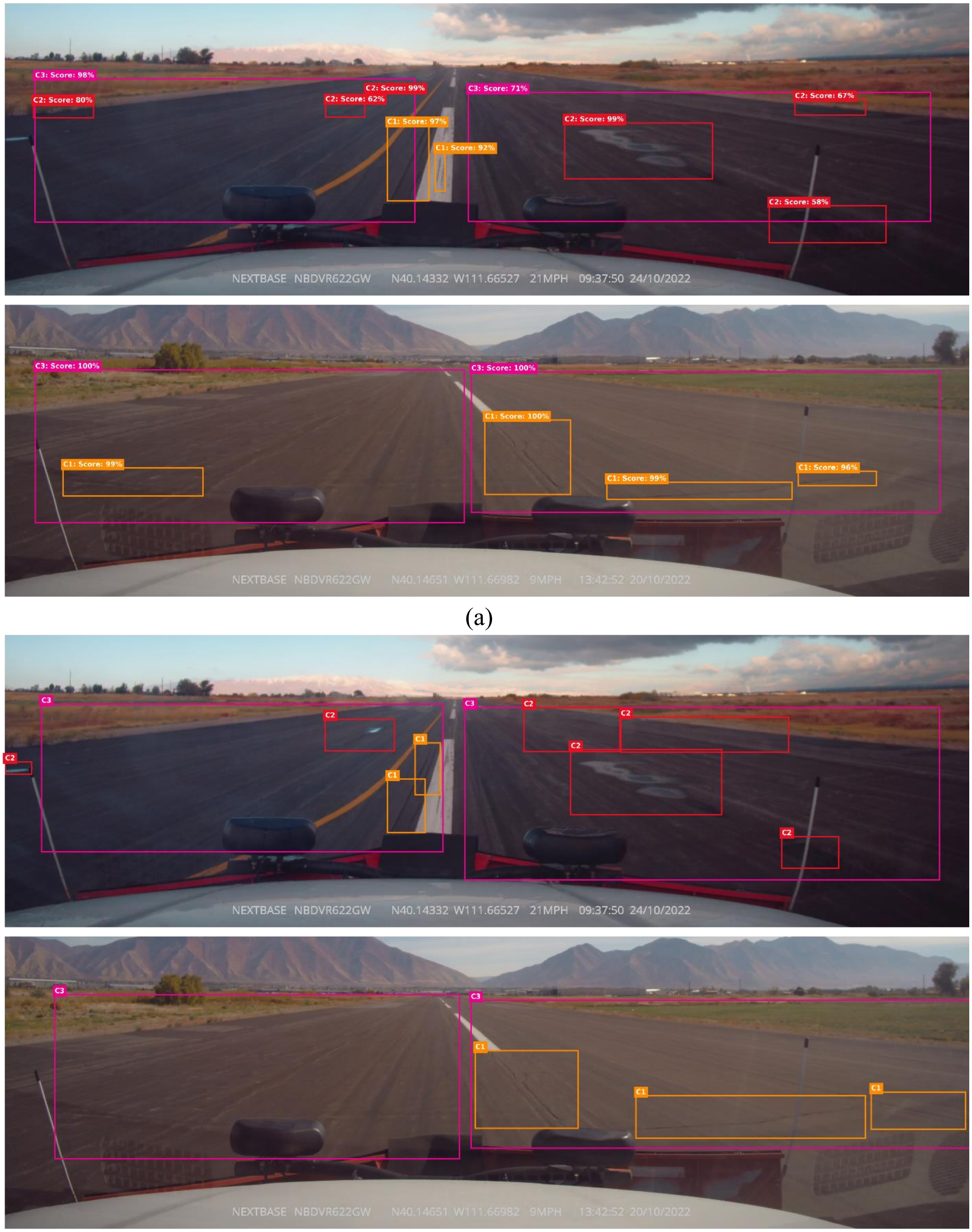
Achieving an optimal balance between precision and recall is essential in pavement distress detection, and selecting the appropriate IoU threshold plays a vital role in optimizing overall model performance. It is worth noting that the choice of the optimal IoU threshold may vary depending on the specific requirements and constraints of the application. Factors such as the importance of minimizing FP, the tolerance for missing some distresses, and the desired precision-recall trade-off should be taken into consideration when selecting the IoU threshold.

Figure 5.9 visually compares the predicted distresses generated by our model (Figure 5.9a) and the ground truth data (Figure 5.9b). By comparing the two figures, we can assess the performance of the model in accurately identifying distresses. A close alignment between the predicted distress and the ground truth indicates a higher level of accuracy and reliability in the model's predictions.

The prediction scores associated with each predicted distress in Figure 5.9 (a) represent the model's confidence level or probability assigned to each prediction. Higher prediction scores indicate a stronger belief in the model's accuracy for particular distress, while lower scores suggest a lower confidence level in the prediction.

This visual comparison allows for a qualitative assessment of the model’s performance and can provide insights into its ability to detect and localize distresses accurately. In the evaluation of our model's performance, we identified several areas where it failed to accurately predict the distresses. It may be due to the complexity and variability of the pavement distress patterns, as well as the presence of noise or occlusions in the input data. The model may struggle to capture subtle variations in distress patterns, leading to misclassifications or missed detections.

The following reasons can explain the model’s inability to predict the distress areas, 1) insufficient or unrepresentative training data, 2) model limitations, 3) unoptimized data preprocessing and augmentation, 4) distress type distribution imbalance, and 5) incorrect or missed annotations.


C3: Score: 98%
C2: Score: 80%
C2: Score: 99%
C2: Score: 62%
C1: Score: 97%
C1: Score: 92%
C3: Score: 71%
C2: Score: 67%
C2: Score: 99%
C2: Score: 58%
NEXTBASE NBDVR622GW N40.14332 W111.66527 21MPH 09:37:50 24/10/2022
C3: Score: 100%
C1: Score: 99%
C3: Score: 100%
C1: Score: 100%
C1: Score: 99%
C1: Score: 96%
NEXTBASE NBDVR622GW N40.14651 W111.66982 9MPH 13:42:52 20/10/2022
C3
C2
C2
C1
C1
C3
C2
C2
C2
C2
NEXTBASE NBDVR622GW N40.14332 W111.66527 21MPH 09:37:50 24/10/2022
C3
C3
C1
C1
C1
NEXTBASE NBDVR622GW N40.14651 W111.66982 9MPH 13:42:52 20/10/2022

(a)

(b)

Figure 5.9: Comparison of (a) predicted distresses from the model and (b) the ground truth data. Note: To reduce cluttering, C1 refers to longitudinal and transverse cracking, C2 refers to rutting/depression, and C3 refers to weathering.

5.1 Map Visualization and Pavement Condition Rating

By mapping the detected distresses, we can identify the specific locations where pavement issues are prevalent. This information helps transportation agencies and road maintenance teams to prioritize segments that require immediate attention and allocate resources effectively for repairs and maintenance activities.

Using the methods described in Section 3.7, we calculated the mapping values to establish a conversion from pixel coordinates to real-world coordinates in both x and y directions. By knowing the GPS position of the camera and its heading, along with the calculated mapping values, we were able to approximate the location of distresses on the pavement. We assumed that the center of the predicted bounding boxes from the model represents the approximate location of the distress. This approach provides a convenient and practical way to identify the specific areas on the pavement that require attention and maintenance. While efforts were made to ensure accuracy, there may still be some inherent limitations and uncertainties associated with this estimation process.

To ensure a comprehensive evaluation of the deep learning model's performance, we utilized a subset of 377 randomly selected images for training and testing purposes. However, during the actual deployment phase, the model was applied to the entire dataset, which consisted of all the frames obtained during the data collection process after preprocessing (see Section 3.2). To geolocate and visualize the distresses on the runway pavement, a GIS visualization was generated. This visualization, represented in Figure 5.10, provides a comprehensive view of the distresses detected across the entire runway (using the Faster R-CNN) that is based on the entire dataset.

Distresses within a segment may appear multiple times and exhibit slight variations in their location. The vehicle may pass a segment multiple times during the day, and slight changes in factors such as vehicle speed, position, and angle can contribute to variations in the captured frames. As a result, some images may show similar or overlapping distresses within a segment. While efforts were made to merge nearby points and eliminate duplications, all instances were included in the final model's predictions due to the inherent challenges in location measurement approximation and frame variability. In practice, these duplications should not significantly impact the overall analysis, as they only result in scaled values.

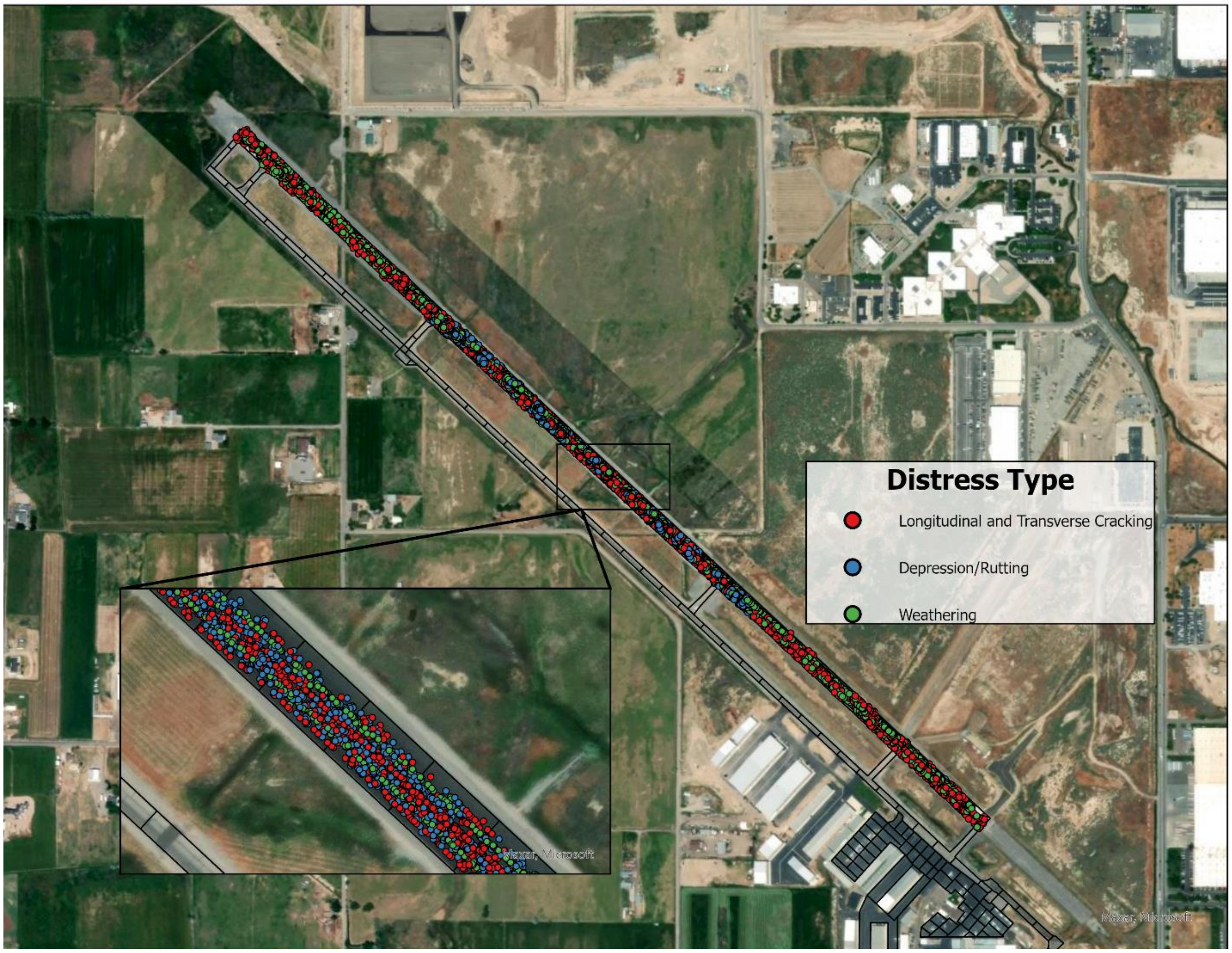
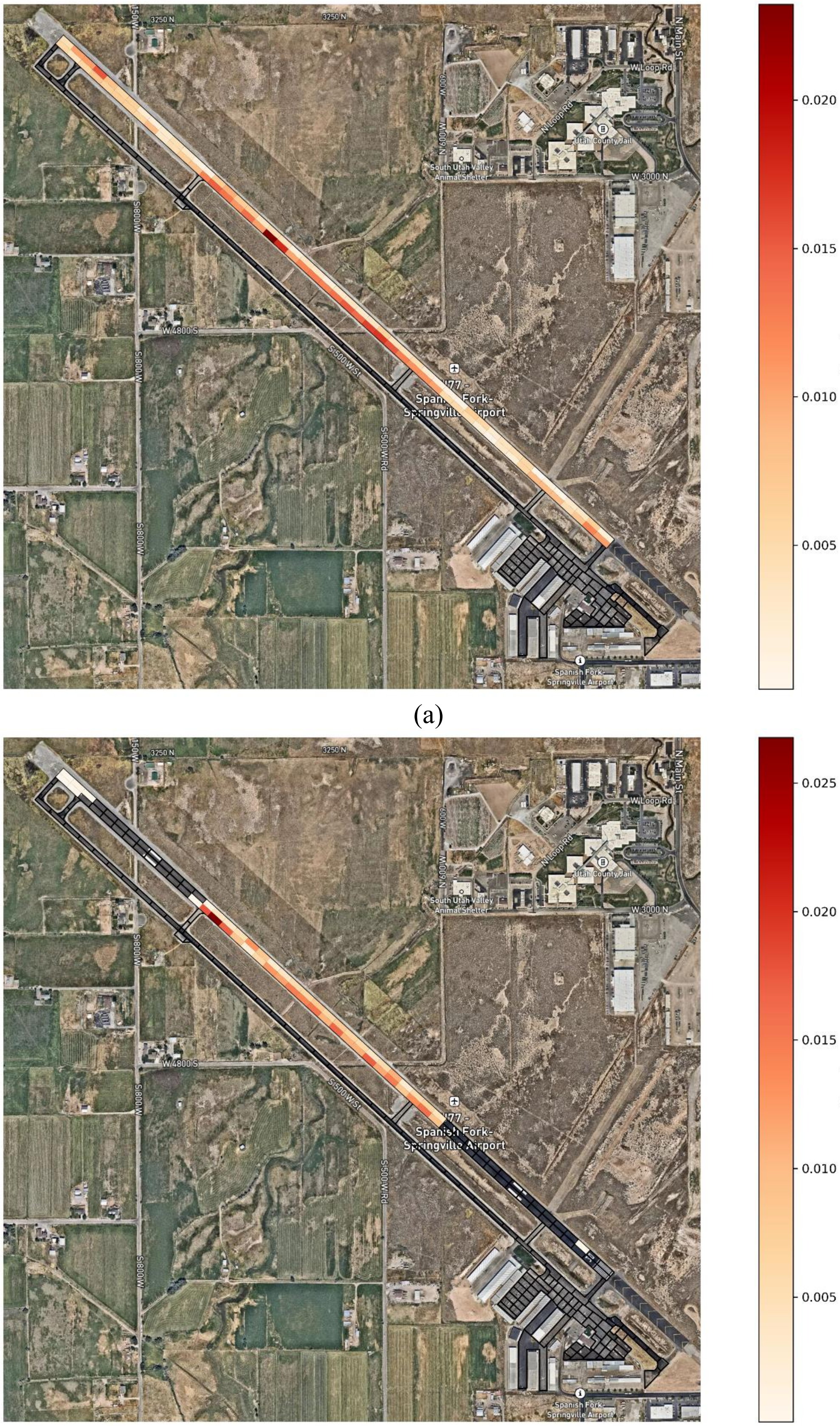


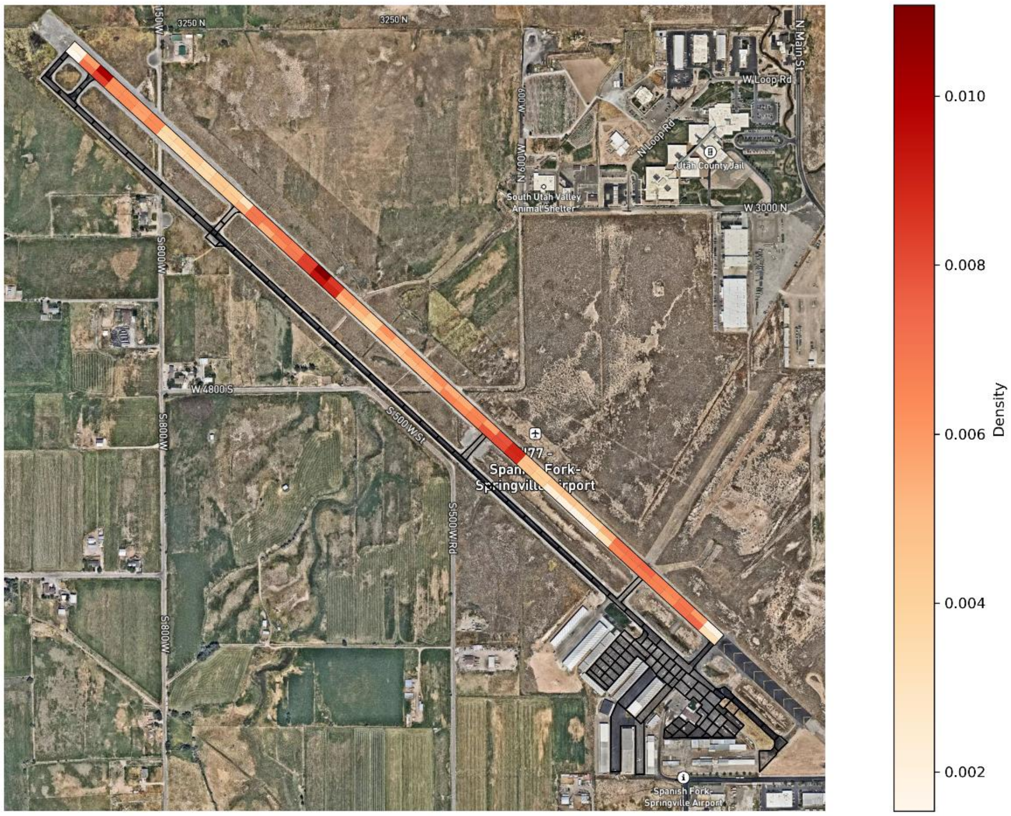
Figure 5.10: GIS Visualization of the Distresses Located on the Runway Pavement

To provide a rating similar to the PCI, we calculate the density of the detected distresses within each segment. The results are shown in Figure 5.11. The distress density can be calculated by dividing the total number of distresses within a segment by the area of the segment. The resulting value represents the number of distresses per unit area. The color-coded map allows for easy interpretation, with darker shades indicating higher distress densities and lighter shades representing lower densities.


3250 N
3250 N
150W
N Main St
W Loop Rd
N Loop Rd
Utah County Jail
W 3000 N
N 600 W
South Utah Valley Animal Shelter
S 800 W
W 4800 S
S 500 W St
S 500 W Rd
U77 -
Spanish Fork-Springville Airport
Spanish Fork-Springville Airport
Density
0.005
0.010
0.015
0.020
0.025

(a)

(b)



(c)

Figure 5.11: Density of detected distresses where (a) is longitudinal and transverse cracking, (b) rutting/depression, (c) weathering within pavement segments.

6.0 DISCUSSION OF RESULTS

The evaluation of the model's performance in pavement distress detection reveals several important findings. As analyzed through the precision-recall curve, the relationship offers insights into the model's ability to capture TP instances while controlling for FN and FP. By considering different IoU thresholds, we can observe the trade-off between precision and recall.

In our evaluation, an IoU threshold of 0.3 was recommended for pavement distress detection, striking a balance between capturing distresses with a reasonable level of overlap and maintaining a satisfactory precision-recall balance. The mAP scores at this threshold indicated relatively improved performance compared to higher IoU thresholds. However, it is crucial to consider the specific needs and constraints of the application when selecting the optimal IoU threshold.

The visual comparison between the predicted distresses, and ground truth data provided qualitative insights into the model's accuracy and reliability. The complexity and variability of pavement distress patterns, along with the presence of noise or occlusions, posed challenges for the model in accurately capturing subtle variations in distress patterns. Several factors can contribute to the limitations observed in the model's ability to accurately predict distress areas:

1) Limited training data, particularly for distress types like rutting and depression that are mostly visible during rainy periods.
2) Imbalance in the distribution of distress types in the training and testing datasets.
3) Presence of wrong labeling or inconsistencies in labeling distresses.
4) Difficulty in discerning distresses on the pavement, especially when zoomed out or captured from a wide-view dashcam perspective.
5) Weathering distress exhibiting exceptionally larger bounding boxes, potentially biasing the model's predictions toward higher precision for this distress type.
6) Limitations of the feature descriptor used for extracting low-level details, such as edges and colors, due to image quality and wide-view orientation of the dashcam.
7) Challenges posed by the presence of skid marks on the pavement, which may resemble longitudinal and transverse cracks and result in FP predictions.

The geolocation of detected distresses provided valuable information for transportation agencies and road maintenance teams to prioritize maintenance activities effectively. By mapping the distresses onto the pavement surface, specific locations requiring attention could be identified.

Although efforts were made to ensure accuracy, limitations, and uncertainties associated with the estimation process should be acknowledged.

In terms of actionability, the findings suggest that automated pavement distress detection can provide valuable support to pavement management. Automated methods offer the advantages of processing large amounts of data efficiently and consistently, identifying distress patterns across the entire pavement network, and detecting distresses that may not be immediately visible to the naked eye. The quantitative data on distress density obtained through automated detection allows for more objective decision-making, monitoring distress progression, and assessing the effectiveness of maintenance strategies.

Utilizing the distress detection model can be a cost-effective solution in the long run. It reduces the reliance on manual inspections, which can be labor-intensive and time-consuming. By promptly identifying distresses and addressing them in their early stages, the model helps prevent further deterioration of the pavement, potentially reducing repair costs and extending the lifespan of the runway.

However, it is important to recognize that the expertise and judgment from experienced engineers are still crucial in interpreting the detected distresses and making informed decisions. Manual inspection can provide additional contextual information, nuanced analysis, and considerations that may not be captured solely through automated detection. Ultimately, rather than replacing the role of an experienced engineer, the model complements their expertise. By providing a visual representation of distress areas and their density, the model assists engineers in identifying and prioritizing segments that require closer inspection and intervention. It aids engineers in focusing their efforts on targeted areas, optimizing their time and resources.

Therefore, a combination of automated distress detection and manual inspection by experienced engineers can offer a comprehensive approach to pavement management, leveraging the efficiency and consistency of automated methods while benefiting from human expertise.

7.0 SYSTEM IMPLEMENTATION AND DEPLOYMENT

The implementation and deployment of the distress detection model and geolocation techniques in real-world applications are essential for effective airport runway maintenance and management. This section delves into the practical aspects of integrating the developed model and geolocation techniques into the existing infrastructure, highlighting the necessary steps, considerations, and potential future developments. By emphasizing the importance of a seamless implementation and deployment strategy, the aim is to enable accurate detection and geolocation of distresses on the runway pavement. This comprehensive approach ensures that the distress detection system becomes a valuable tool for UDOT, facilitating proactive maintenance planning, optimizing resource allocation, and ultimately enhancing the overall condition and safety of the runway infrastructure.

7.1 Implementation Steps

7.1.1 Data Acquisition and Processing

First, data collection is performed by installing a dashcam or similar recording device on a vehicle that traverses the runway. The vehicle captures high-resolution images at regular intervals, ensuring comprehensive coverage of the pavement surface. This dataset serves as the foundation for training the deep learning model.

To effectively utilize the high-resolution and high FPS capabilities of the dashcam, a precise frame extraction technique is employed to extract a diverse set of images from the runway. This technique ensures that the extracted frames do not overlap and provide different views of the distresses present on the pavement. The frame extraction process, along with the spatial point selection algorithm, is fully automated, allowing for efficient and seamless data collection and processing.

The data annotation step is critical in creating a reliable training dataset for the pavement distress detection system. To ensure the dataset is representative and covers various scenarios, a subset of images from the entire dataset is selected. It is preferable to include images from different times of the day to capture variations in lighting conditions. Each selected image is then annotated by drawing precise bounding boxes around the distress areas, indicating their exact location and type. This annotation process ensures that the model can learn to detect and classify distresses accurately.

To maintain consistency and labeling quality across the dataset, it is important to annotate each distress in sequential images as long as it remains clearly visible to the annotator. This ensures that the annotations accurately reflect the temporal and spatial continuity of the distresses. Furthermore, a validation check should be conducted by UDOT experts to verify the identified distresses and ensure their validity. This step helps to maintain the reliability and accuracy of the training dataset.

7.1.2 Model Training and Performance Evaluations

Model training plays a crucial role in developing an accurate and robust pavement distress detection system. The chosen deep learning model, such as the Faster R-CNN, is trained using the annotated dataset obtained through the data annotation process. This model training phase involves optimizing the model's parameters to enable it to accurately detect and classify distresses on the runway pavement. The training dataset, consisting of annotated images with bounding boxes around distress areas, serves as the basis for the model to learn the spatial and visual characteristics of different distress types. By repeatedly exposing the model to this labeled data and adjusting its parameters, the model gradually learns to recognize patterns and features specific to each distress type.

To ensure the model’s performance and prevent overfitting, techniques such as data augmentation are applied during training. Data augmentation involves applying various transformations to the input images, such as rotation, scaling, and flipping, to increase the diversity and variability of the training data. This helps the model generalize well to unseen distress patterns and improves its overall performance.

Following model training, evaluation is conducted to assess its performance and accuracy. Performance metrics such as precision, recall, F2 score, and mAP are used to measure the model's ability to detect and classify distresses. These metrics provide insights into the model's accuracy and robustness in identifying pavement distresses.

Once the model evaluation meets the required performance standards, the trained model can be deployed on the entire dataset to predict distresses across the runway pavement. This deployment involves applying the trained model to all the images captured during the data collection process.

However, if the model evaluation indicates room for improvement or falls short of the desired performance metrics, further training and optimization steps are necessary. This entails refining the model by adjusting hyperparameters, fine-tuning the training process, or exploring different deep learning architectures. Additionally, if the training dataset is deemed insufficient or lacking diversity, the inclusion of additional annotated images can be considered to enhance the model's ability to accurately detect and classify distresses.

The decision to deploy the model or continue training and optimization depends on the evaluation results and the specific requirements of UDOT. It is crucial to ensure that the model meets the desired performance criteria and aligns with the agency's objectives. By carefully evaluating and refining the model, UDOT can develop a reliable and effective distress detection system that aids in pavement management and maintenance decision-making processes.

7.1.3 Geolocation and GIS Visualizations

After the model evaluation process confirms that the distress detection model meets the required performance standards of the agency, the geolocation of the detected distresses becomes a crucial step in providing accurate spatial information for decision-making. Geolocating the distresses identified by the model predictions is a critical aspect to accurately determine their positions on the runway pavement. However, dashcams used for data collection do not inherently provide depth information, which is necessary for precise geolocation. To overcome this limitation, computer vision techniques based on Homography can be employed to establish a mapping between pixel coordinates and real-world coordinates.

The pixel mapping process involves establishing a conversion between the pixel coordinates and real-world coordinates in both the X (horizontal) and Y (vertical) directions. For both directions, first, the Homography technique is used to warp the perspective of the image to a bird's-eye view, enabling the measurement of physically known objects within the image. For the Y-direction mapping, by correlating the GPS information captured by the dashcam with the shift in the white centerline pixels coordinates in two sequential images, a conversion factor can be calculated to determine the real-world distance in meters corresponding to each pixel in the Y-direction. To establish the pixel mapping for the X-direction, a similar approach can be adopted by corresponding the width of the runway in pixels to its known real-world measurement.

To ensure accurate geolocation of distresses, it is important to apply the pixel mapping process for every individual dashcam. If different dashcams are employed, mapping values specific to each dashcam should be calculated beforehand. This is necessary because different dashcams may have variations in their field of view, image resolution, and distortion characteristics, which can impact the accuracy of geolocation. However, if the same dashcam is consistently used throughout the airport, additional images can be seamlessly integrated into the geolocation process without any issues. The previously established pixel mapping values for that specific dashcam can be directly applied to the new images. This allows for efficient and reliable geolocation of distresses, ensuring spatial accuracy and consistency across the dataset.

It is crucial to emphasize that the pixel mapping step is a manual process and requires careful validation. It is recommended to involve experts from UDOT or qualified personnel to verify the accuracy and reliability of the geolocation results. This validation step ensures that the distresses are accurately placed in their corresponding real-world locations, providing reliable spatial information for decision-making and further analysis.

Once the mapping values are determined and distresses are accurately geolocated, a GIS visualization can be utilized to provide a comprehensive assessment of the pavement condition. The center of the detected distresses bounding boxes from the model can be used as an approximation to locate the distress locations on the runway pavement. However, it is important to note that the geolocation process may have inherent inaccuracies due to factors such as the resolution of the dashcam images, the quality of the mapping values, and the limitations of the pixel mapping techniques used. These factors can introduce uncertainties in the precise positioning of the distresses on the pavement. Despite these limitations, GIS visualization enables decision-makers to gain valuable insights into the spatial distribution and density of distresses across the runway. By visually representing the distresses on a map, patterns and trends can be identified, and critical areas requiring maintenance or repair can be highlighted. This information assists transportation agencies, road maintenance teams, and stakeholders in prioritizing resources, planning interventions, and implementing effective pavement management strategies.

7.2 Deployment Steps

The deployment of the pavement distress detection system offers significant benefits to UDOT in terms of runway maintenance and management. Once the system has been developed and thoroughly tested, it can be deployed for regular use at the Spanish Fork airport and potentially expanded to other airports within UDOT's jurisdiction.

To deploy the system, the trained model and associated software components need to be integrated into UDOT's infrastructure. This may involve setting up a dedicated server or computing environment equipped with the necessary hardware and software requirements, such as GPUs, deep learning frameworks, and data storage capabilities. The integration process ensures smooth data flow, model execution, and visualization of distress information within UDOT's existing runway maintenance systems.

Once deployed, the system provides UDOT with near-real-time and accurate information about the condition of the runway pavement. This enables proactive maintenance planning and timely interventions to address identified distress areas. By leveraging the power of deep learning and computer vision, UDOT can achieve a higher level of efficiency and accuracy in detecting and monitoring pavement distresses.

Furthermore, the pavement distress detection system can be further enhanced and expanded in the future. Some potential avenues for future development include:

1) **Advanced distress analysis:** The system can be extended to perform in-depth analysis of distress patterns, severity assessment, and prediction of pavement deterioration. By analyzing the historical data and capturing temporal changes in distress patterns, the system can provide insights into the progression of distresses and aid in proactive maintenance planning.
2) **Automated reporting and alerts:** The system can be equipped with automated reporting and alert functionalities. This would enable automatic generation of maintenance reports, highlighting distress areas, recommended repair actions, and priority levels. Real-time alerts can also be triggered based on predefined thresholds for distress density or severity, allowing immediate attention to critical areas.
3) **Continuous model improvement:** As the system is deployed and more data is collected, UDOT can continue to refine and improve the deep learning model. This includes retraining the model with new data, addressing any shortcomings or limitations identified during

deployment, and leveraging advancements in deep learning techniques to enhance the model's performance.

In summary, the future potential of the pavement distress detection system lies in the adoption of low-cost dashcam solutions for continuous monitoring and assessment of runway surfaces. This approach can enhance UDOT's runway management capabilities, improve maintenance planning, and enable proactive decision-making based on real-time distress data. By embracing emerging technologies and exploring innovative solutions, UDOT can stay at the forefront of runway maintenance practices and ensure the safety and longevity of its airport infrastructure.

8.0 CONCLUSION

Pavement distresses on runways pose significant challenges to the safety, durability, and overall performance of airport infrastructure. Distresses such as longitudinal and transverse cracking, rutting, depression, and weathering not only affect the visual appearance of the runway but also have critical implications for aircraft operation. Runway maintenance and management require a comprehensive understanding of distress patterns, their severity, and spatial distribution. Traditionally, this has relied on manual inspections performed by experienced engineers or technicians, which can be time-consuming, subjective, and prone to human error. The integration of advanced computer vision and deep learning techniques into the assessment process brings numerous advantages.

This project has successfully developed and implemented a low-cost solution for pavement distress detection using a dashcam and advanced computer vision techniques. The utilization of a dashcam as a low-cost solution offers several advantages. It eliminates the need for specialized and expensive equipment, making it accessible to a wider range of airports and organizations with budget constraints. The affordability and ease of installation of dashcams make them a practical choice for runway maintenance, allowing for efficient and cost-effective distress detection.

Our pavement distress detection model has demonstrated the potential to revolutionize the way we assess and monitor runway conditions. Through rigorous evaluation and analysis of the developed model utilizing Faster R-CNN deep learning architecture, we have gained valuable insights into the model's performance, its limitations, and the practical implications of its implementation. The results of our research highlight the effectiveness of the model in identifying distress areas and quantifying their density. The AI-assisted distress detection system offers several advantages over traditional manual inspections, including enhanced speed, consistency, and objectivity.

Overall, this project represents a promising step towards leveraging AI and computer vision technologies for effective runway maintenance and management. By combining the expertise of engineers with the capabilities of automated distress detection models, we can enhance the accuracy, efficiency, and cost-effectiveness of runway inspections. Further research and development in this field have the potential to revolutionize infrastructure management practices, leading to safer and more sustainable transportation systems.

REFERENCES

Amhaz, R., Chambon, S., Idier, J., & Baltazart, V. (2014). A new minimal path selection algorithm for automatic crack detection on pavement images. *2014 IEEE International Conference on Image Processing (ICIP)*, 788–792. https://doi.org/10.1109/ICIP.2014.7025158

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. https://doi.org/10.1016/j.dib.2021.107133

Bang, S., Park, Â. S., Kim, Â. H., & Kim, Â. Y.-S. Y. andÂ H. (2018). A Deep Residual Network with Transfer Learning for Pixel-level Road Crack Detection. *ISARC Proceedings*, 753–756.

Cao, M.-T., Tran, Q.-V., Nguyen, N.-M., & Chang, K.-T. (2020). Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources. *Advanced Engineering Informatics*, *46*, 101182. https://doi.org/10.1016/j.aei.2020.101182

Chambon, S., & Moliard, J.-M. (2011). Automatic Road Pavement Assessment with Image Processing: Review and Comparison. *International Journal of Geophysics*, *2011*, 1–20. https://doi.org/10.1155/2011/989354

Chaudhury, K., DiVerdi, S., & Ioffe, S. (2014). Auto-rectification of user photos. *2014 IEEE International Conference on Image Processing (ICIP)*, 3479–3483. https://doi.org/10.1109/ICIP.2014.7025706

Chen, C., Seo, H., Jun, C. H., & Zhao, Y. (2022). Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM. *International Journal of Pavement Engineering*, *23*(9), 3274–3283. https://doi.org/10.1080/10298436.2021.1888092

Cheng, J., Xiong, W., Chen, W., Gu, Y., & Li, Y. (2018). Pixel-level Crack Detection using U-Net. *TENCON 2018 - 2018 IEEE Region 10 Conference*, 0462–0466. https://doi.org/10.1109/TENCON.2018.8650059

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, *186*, 1031–1045. https://doi.org/10.1016/j.conbuildmat.2018.08.011

Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, *99*, 52–58. https://doi.org/10.1016/j.autcon.2018.11.028

ExifTool by Phil Harvey. (n.d.). Retrieved April 12, 2023, from https://exiftool.org/

FAA. (2014). *AC 150/5380-7B: Airport Pavement Management Program (PMP)*. FAA Washington, DC, USA.

Fan, R., Bocus, M. J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S., & Liu, M. (2019). Road Crack Detection Using Deep Convolutional Neural Network and Adaptive Thresholding. *2019 IEEE Intelligent Vehicles Symposium (IV)*, 474–479. https://doi.org/10.1109/IVS.2019.8814000

Fan, Z., Wu, Y., Lu, J., & Li, W. (2018). *Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network* (arXiv:1802.02208). arXiv. https://doi.org/10.48550/arXiv.1802.02208

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., & Agrawal, A. (2017). Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, *157*, 322–330. https://doi.org/10.1016/j.conbuildmat.2017.09.110

Hsieh, Y.-A., & Tsai, Y. J. (2020). Machine Learning for Crack Detection: Review and Model Performance Comparison. *Journal of Computing in Civil Engineering*, *34*(5), 04020038. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000918

Jiang, L., Xie, Y., & Ren, T. (2020). *A Deep Neural Networks Approach for Pixel-Level Runway Pavement Crack Segmentation Using Drone-Captured Images* (arXiv:2001.03257). arXiv. http://arxiv.org/abs/2001.03257

Li, H., Jing, P., Huang, R., & Gui, Z. (2021). Algorithm for Crack Segmentation of Airport Runway Pavement under Complex Background based on Encoder and Decoder. *2021 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1706–1711. https://doi.org/10.1109/ROBIO54168.2021.9739462

Ma, K., Hoai, M., & Samaras, D. (2017). Large-scale Continual Road Inspection: Visual Infrastructure Assessment in the Wild. *Procedings of the British Machine Vision Conference 2017*, 151. https://doi.org/10.5244/C.31.151

Maeda, H., Kashiyama, T., Sekimoto, Y., Seto, T., & Omata, H. (2021). Generative adversarial network for road damage detection. *Computer-Aided Civil and Infrastructure Engineering*, *36*(1), 47–60. https://doi.org/10.1111/mice.12561

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road Damage Detection and Classification Using Deep Neural Networks with Smartphone Images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1127–1141. https://doi.org/10.1111/mice.12387

Majidifard, H., Jin, P., Adu-Gyamfi, Y., & Buttlar, W. G. (2020). Pavement Image Datasets: A New Benchmark Dataset to Classify and Densify Pavement Distresses. *Transportation Research Record*, *2674*(2), 328–339. https://doi.org/10.1177/0361198120907283

Mandal, V., Mussah, A. R., & Adu-Gyamfi, Y. (2020). Deep Learning Frameworks for Pavement Distress Classification: A Comparative Analysis. *2020 IEEE International Conference on Big Data (Big Data)*, 5577–5583. https://doi.org/10.1109/BigData50022.2020.9378047

Mandal, V., Uong, L., & Adu-Gyamfi, Y. (2018). Automated Road Crack Detection Using Deep Convolutional Neural Networks. *2018 IEEE International Conference on Big Data (Big Data)*, 5212–5215. https://doi.org/10.1109/BigData.2018.8622327

Mohan, A., & Poobal, S. (2018). Crack detection using image processing: A critical review and analysis. *Alexandria Engineering Journal*, *57*(2), 787–798. https://doi.org/10.1016/j.aej.2017.01.020

Morian, D. A., Frith, D., Stoffels, S. M., Jahangirnejad, S., & United States. Federal Highway Administration. Office of Technical Services. (2020). *Developing Guidelines for Cracking Assessment for Use in Vendor Selection Process for Pavement Crack Data Collection/Analysis Systems and/or Services* (FHWA-RC-20-0005). https://rosap.ntl.bts.gov/view/dot/55791

NEW 622GW Dash Cam | Nextbase. (n.d.). Nextbase - United States. Retrieved April 12, 2023, from https://nextbase.com/dash-cams/622gw-dash-cam/

Nguyen, T. S., Begot, S., Duculty, F., & Avila, M. (2011). Free-form anisotropy: A new method for crack detection on pavement surface images. *2011 18th IEEE International Conference on Image Processing*, 1069–1072. https://doi.org/10.1109/ICIP.2011.6115610

Oliveira, H., & Correia, P. L. (2014). CrackIT — An image processing toolbox for crack detection and characterization. *2014 IEEE International Conference on Image Processing (ICIP)*, 798–802. https://doi.org/10.1109/ICIP.2014.7025160

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Pan, Y., Zhang, X., Cervone, G., & Yang, L. (2018). Detection of Asphalt Pavement Potholes and Cracks Based on the Unmanned Aerial Vehicle Multispectral Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(10), 3701–3712. https://doi.org/10.1109/JSTARS.2018.2865528

Pauly, L., Peel, H., Luo, S., Hogg, D., & Fuentes, R. (2017, July 1). *Deeper Networks for Pavement Crack Detection*. 34th International Symposium on Automation and Robotics in Construction, Taipei, Taiwan. https://doi.org/10.22260/ISARC2017/0066

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (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

Some, L. (2016). *Automatic image-based road crack detection methods*. http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-189245

Urban, G., Geras, K. J., Kahou, S. E., Aslan, O., Wang, S., Caruana, R., Mohamed, A., Philipose, M., & Richardson, M. (2017). *Do Deep Convolutional Nets Really Need to be Deep and Convolutional?* (arXiv:1603.05691). arXiv. https://doi.org/10.48550/arXiv.1603.05691

Wang, W., Wang, M., Li, H., Zhao, H., Wang, K., He, C., Wang, J., Zheng, S., & Chen, J. (2019). Pavement crack image acquisition methods and crack extraction algorithms: A review. *Journal of Traffic and Transportation Engineering (English Edition)*, *6*(6), 535–556. https://doi.org/10.1016/j.jtte.2019.10.001

Wang, X., & Hu, Z. (2017). Grid-based pavement crack analysis using deep learning. *2017 4th International Conference on Transportation Information and Safety (ICTIS)*, 917–924. https://doi.org/10.1109/ICTIS.2017.8047878

Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., & Ling, H. (2019). *Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection* (arXiv:1901.06340). arXiv. http://arxiv.org/abs/1901.06340

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1090–1109. https://doi.org/10.1111/mice.12412

Zhang, H., Wu, Z., Qiu, Y., Zhai, X., Wang, Z., Xu, P., Liu, Z., Li, X., & Jiang, N. (2022). A New Road Damage Detection Baseline with Attention Learning. *Applied Sciences*, *12*(15), 7594. https://doi.org/10.3390/app12157594

Zhang, K., Cheng, H. D., & Zhang, B. (2018). Unified Approach to Pavement Crack and Sealed Crack Detection Using Preclassification Based on Transfer Learning. *Journal of Computing in Civil Engineering*, *32*(2), 04018001. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000736

Zhang, L., Yang, F., Daniel Zhang, Y., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. *2016 IEEE International Conference on Image Processing (ICIP)*, 3708–3712. https://doi.org/10.1109/ICIP.2016.7533052

Zhang, X., Xia, X., Li, N., Lin, M., Song, J., & Ding, N. (2020). Exploring the Tricks for Road Damage Detection with A One-Stage Detector. *2020 IEEE International Conference on Big Data (Big Data)*, 5616–5621. https://doi.org/10.1109/BigData50022.2020.9377923

Zhu, J., Zhong, J., Ma, T., Huang, X., Zhang, W., & Zhou, Y. (2022). Pavement distress detection using convolutional neural networks with images captured via UAV. *Automation in Construction*, *133*, 103991. https://doi.org/10.1016/j.autcon.2021.103991

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, *33*(3), 227–238. https://doi.org/10.1016/j.patrec.2011.11.004