Evaluating Primary and Secondary Roadway Pavement Conditions using Deep Learning

NCDOT Project 2023-01
FHWA/NC/2023-01
December 2024

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

NC STATE UNIVERSITY

Don Chen, Ph.D.
Wenwu Tang, Ph.D.
University of North Carolina at Charlotte

Christopher Vaughan
North Carolina State University

STATE OF NORTH CAROLINA DEPARTMENT OF TRANSPORTATION

RESEARCH & DEVELOPMENT

Evaluating Primary and Secondary Roadway Pavement Conditions using Deep Learning

Final Report
(Report No. RP 2023-01)

To North Carolina Department of Transportation
(Research Project No. RP 2023-01)

Submitted by

Don Chen, Ph.D.
Professor, Dept. of Engineering Technology and Construction Management
University of North Carolina at Charlotte, Charlotte, NC 28223
Phone: (704) 687-5036; E-mail: dchen9@charlotte.edu

Wenwu Tang, Ph.D.
Executive Director, Center for Applied Geographic Information Science
Professor, Department of Geography and Earth Sciences
University of North Carolina at Charlotte, Charlotte, NC 28223
Phone: (704) 687-5988; E-mail: WenwuTang@charlotte.edu

Christopher Vaughan
Research Associate, Institute for Transportation Research and Education (ITRE)
North Carolina State University, Raleigh, NC 27695
Office: (919) 515-8036; Email: clvaugha@ncsu.edu

Yaying Shi, Zach Slocum, Tamim Adnan, Yanfang Su
Graduate Research Assistant
University of North Carolina at Charlotte, Charlotte, NC 28223

Savannah Wright, Mitchell Brooks, Daniel Coble
Graduate Research Assistant
North Carolina State University, Raleigh, NC 27695

Department of Engineering Technology and Construction Management
University of North Carolina at Charlotte
Charlotte, NC

December 2024

1. Report No. ***FHWA/NC/2023-01***	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle Evaluating Primary and Secondary Roadway Pavement Conditions using Deep Learning	5. Report Date December 2024
	6. Performing Organization Code
7. Author(s) Don Chen, Wenwu Tang, Chris Vaughan, Yaying Shi, Zach Slocum, Tamim Adnan, Yanfang Su, Savannah Wright, Mitchell Brooks, Daniel Coble	8. Performing Organization Report No.
9. Performing Organization Name and Address Smith 274 Dept. of Engineering Technology and Construction Management University of North Carolina at Charlotte Charlotte, NC 28223-0001	10. Work Unit No. (TRAIS)
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address North Carolina Department of Transportation Research and Analysis Group 1 South Wilmington Street Raleigh, North Carolina 27601	13. Type of Report and Period Covered Final Report Final Report August 2022 – December 2024
	14. Sponsoring Agency Code RP 2023-01

Supplementary Notes:

16. Abstract

This research project aimed to identify a cost-effective method for collecting distress data on secondary roadways and to develop efficient deep learning models for classifying and quantifying pavement cracks on both primary and secondary roadways in North Carolina. The project utilized high-resolution images provided by the NCDOT as well as images captured using GoPro cameras. The findings revealed that GoPro cameras, when mounted on the rear of a vehicle, driven at speeds below 20 mph, and used in fair weather conditions, offer a low-cost solution for data collection. For high-accuracy image annotation, Fiji (ImageJ) with its Trainable Weka Segmentation (TWS) plugin proved to be highly effective.

For crack classification on primary roadways, both ResNet and Vision Transformer models demonstrated exceptional performance, achieving a 97% accuracy in identifying crack types (longitudinal, transverse, and alligator cracks). Similarly, for secondary roadways, ResNet and Vision Transformer models are recommended, achieving an 85% accuracy using images extracted from GoPro videos. In terms of crack segmentation, U-Net and DeepSegmentor models are recommended for primary roadways, with Dice Coefficients of 97%. For secondary roadways, the DeepSegmentor model is recommended due to its superior performance in handling complex crack patterns. For crack quantification, the use of pixel-level segmentation with real-world calibration ensured precise measurements of crack length, width, and area.

Once implemented, the outcomes of this research have the potential to significantly enhance the current NCDOT Pavement Management System (PMS). This will enable more frequent and cost-effective monitoring of roadway conditions, improve the accuracy of pavement performance predictions, and assist engineers in maintaining roadways with enhanced performance, extended lifespan, and reduced maintenance requirements.

17. Key Words Deep learning, PMS, Longitudinal/Transverse/ Alligator crack, classification, segmentation, Image annotation, GoPro, ResNet, Vision Transformer, DeepSegmentor, U-Net, Fiji, Photogrammetry	18. Distribution Statement

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 70	22. Price

DISCLAIMER

The contents of this report reflect the views of the authors and not necessarily the views of the University. The authors are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of either the North Carolina Department of Transportation or the Federal Highway Administration at the time of publication. This report does not constitute a standard, specification, or regulation.

ACKNOWLEDGMENTS

This research was sponsored by the North Carolina Department of Transportation. The authors gratefully acknowledge the support and assistance from the following individuals:

Research Project Steering and Implementation Committee: Camille Coombes (Chair), Mustan Kadibhai (Research Engineer), Clark Morrison, Shihai Zhang, Joseph Barbour, Todd Whittington, Matthew Hilderbran, and Andrew Wargo.

The authors also would like to thank the following UNC Charlotte and ITRE/NC State students for their participation in this research project :

- UNC Charlotte students: Yaying Shi, Zach Slocum, Tamim Adnan, Tianyang Chen, and Yanfang Su.
- ITRE/NC State students: Savannah Wright, Mitchell Brooks, and Daniel Coble.

EXECUTIVE SUMMARY

This research project was conducted to identify a cost-effective method for collecting distress data on secondary roadways, classify crack types, and quantify crack sizes using deep learning. To achieve these goals, both high-resolution images provided by NCDOT and images collected with GoPro cameras mounted on vehicles were collected and annotated at a pixel level, creating databases for the development of deep learning models.

Key Findings of this research project are:

- GoPro cameras can be used as a low-cost method for collecting pavement distress data for secondary roadways. The optimal setup involves mounting the camera on the rear of the vehicle, maintaining a speed of 20 mph or less, and avoiding data collection during poor weather conditions. Video clips were found to be more suitable than still images for data collection.
- Fiji (ImageJ) is an effective tool for image annotation. The Trainable Weka Segmentation (TWS) plugin within Fiji allows for high-accuracy, pixel-level annotation of cracks.
- For primary roadways, either a ResNet or a Vision Transformer model is suitable for classifying cracks (longitudinal, transverse, or alligator). These models achieved an accuracy of 97% and 98%, respectively.
- Similarly, for secondary roadways, either a ResNet or a Vision Transformer model is recommended for classifying cracks. Using images extracted from GoPro videos, these models achieved an accuracy of 85%.
- For segmenting cracks in primary roadways, either a U-Net or DeepSegmentor model is recommended. Both models showed satisfactory results with a Dice Coefficient score of 97%.
- For segmenting cracks in secondary roadways, a DeepSegmentor model is recommended. This model performed better at capturing complex crack patterns and subtle cracks.
- For crack quantification, the use of pixel-level segmentation with real-world calibration ensured precise measurements of crack length, width, and area.
- The deep learning models developed in this project were found to be efficient in processing pavement distress images. Training times were short, and image processing times were extremely quick.

Recommendations for Future Research are:

- Annotating more crack images to enhance the training data.
- Addressing limitations of current models.
- Applying the proposed methods to non-cracking distress.
- Testing the proposed GoPro method on selected secondary routes in North Carolina.

Once implemented, the outcomes of this research have the potential to significantly enhance the current NCDOT Pavement Management System (PMS). This will enable more frequent and cost-effective monitoring of roadway conditions, improve the accuracy of pavement performance predictions, and assist engineers in maintaining roadways with enhanced performance, extended lifespan, and reduced maintenance requirements.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION AND OBJECTIVES

1.1 Background

Pavements deteriorate over time, and as the single most expensive asset that a state owns, they need to be strategically maintained. According to the 2021 American Society of Civil Engineers (ASCE) Infrastructure Report Card, America's overall road condition is rated as D, revealing that "over 40% of the system is now in poor or mediocre condition" [1]. Our nation's roads are chronically underfunded, with a budget deficit of $786 billion in 2021, which is a significant increase from $420 billion in 2017. The North Carolina Department of Transportation (NCDOT) manages the second largest state-maintained highway network in the United States, and its roadway network "functions at a high level of efficiency and safety." Sufficient funding, however, is required for the NCDOT to sustain this level of quality. In 2013, roads in North Carolina "require[d] a $23.04 billion investment to meet all performance targets for the existing infrastructure; an additional $61.05 billion is estimated to be accrued over the next 30 years to meet target levels of service" [1]. These funding needs will likely become greater in 2024. A pavement management system (PMS) that allows roadway conditions to be monitored frequently in a cost-effective manner, predicts more accurate pavement performance, and assists engineers in maintaining roadways that have improved performance with longer life and reduced maintenance is desirable.

1.2 Research Needs and Significance

In the past several decades, state Departments of Transportation (DOTs) have adopted optimal strategies for maintaining pavements through Pavement Management Systems (PMS). Today, all 50 states utilize some form of a PMS to manage their roadway networks. These PMSs, however, have the following limitations:

- Manual Date Collection for Secondary Roadways. Distress data for secondary roadways is still primarily collected using a manual ("windshield") method, as the use of automated asset collection vehicles is very expensive and generally only cost-effective for primary roadways. This manual method is not only time-consuming but also burdensome for agency staff, particularly when they have limited time to perform these tasks. This highlights the need for a more efficient and cost-effective method to collecting distress data on secondary roadways.
- Outdated Image Processing Techniques. Traditional computer vision methods are still used for image processing when more advanced technologies are available. Despite rapid advancements in data processing and visualization technologies, the workflow of a traditional PMS remains largely unchanged, affecting their accuracy and overall effectiveness. Therefore, there is a need to explore cutting-edge technologies, such as deep learning, to assess their potential for efficiently classifying and quantifying pavement cracks.

To address these limitations, this research project aims to achieve three goals: (1) to identify a cost-effective distress data collection method for secondary roadways, (2) to classify crack types using deep learning, and (3) to quantify crack sizes using deep learning.

1.3 Research Objectives

To achieve the above-mentioned goals of this research project, the following objectives are proposed:

- To determine adequate camera specifications for a hood-mounted, low-cost image collection method for evaluating secondary roadway pavement distress.
- To collect pavement surface images on samples of secondary roadways.
- To annotate images collected from samples of primary and secondary roadways and develop deep-learning training databases.
- To develop deep learning algorithms for evaluating primary and secondary roadway pavement conditions, specifically, providing answers to the following three research questions:
    1. If a roadway surface image contains a crack (Yes/No)?
    2. What type of crack is it (Longitudinal/Transverse/Alligator)?
    3. What is the size (Width/Length/Area) of this crack?

1.4 Report Organization

An introduction to the research project, research needs and objectives are presented in Chapter 1. A comprehensive literature review is provided in Chapter 2. The data collection and preparation process are described in Chapter 3. Chapter 4 focuses on data analysis and results. Chapter 5 provides conclusions drawn from this research and recommendations for future research.

Appendix A includes lessons learned from the GoPro image annotation process.

CHAPTER 2 LITERATURE REVIEW

An extensive literature review was conducted to synthesize historical advancements, current practices, and emerging innovations in pavement management and distress analysis, establishing a foundation for identifying research gaps and justifying the research methodology proposed in this research project.

2.1 Pavement Date Collection using GoPro Cameras

To identify a cost-effective method for collecting pavement performance data, prior studies conducted on the applications of portable cameras, such as GoPro, were reviewed to gain insights. In 2017, Coenen and Golroo [2] successfully identified cracks in the urban streets of Bengaluru, India, with an accuracy of 80% using pavement data collected by a GoPro camera mounted on the rear of a vehicle. In a similar study, Mei and Gül [3] collected video footage of pavement surfaces across various interstates in Edmonton, Canada, using a GoPro Hero7 Black camera mounted on the rear of a vehicle. Still images extracted from 3 hours of video were analyzed using a ConnCrack deep learning neural network. In 2020, Leduc and Assaf [4] collected still images of pavements using a GoPro camera with a 60-degree shooting angle. These images were subsequently processed using a Convolutional Neural Network (CNN). In another study (Mahlberg et al., 2021), two GoPro Hero pavement conditions.

2.2 GoPro Image Processing

Most of the work conducted recently regarding GoPro (or other action camera) data extraction deals primarily with the metadata recorded on the GoPro or splicing captured still images together. Specifically, researchers have used GoPros for photogrammetry to view or recreate an environment after image capture [5] [6]. This is possible because GoPro GPS data and sensors such as the accelerometer and gyroscope have become more accurate. Software packages have even been developed and are being made available to the public via subscription (Telemetry Overlay and Telemetry Extractor [7]). While this topic is a portion of this research project, one key component that seems to be less studied is how to glean information captured in the GoPro image itself. However, piecing together the available research in photogrammetry from GoPro cameras, computer vision, and machine learning points to the possibility of utilizing not only GoPro metadata but also the GoPro-recorded imagery.

2.3 Image Annotation Methods

Image annotation is a crucial step for supervised machine learning. Two widely used methods are bounding box annotation [8] and pixel-level annotation [9]. The bounding box method involves placing a bounding box around a crack and another around a non-cracking area. Although this method is less time-consuming, it only provides approximate locations of cracks, resulting in low accuracy. The pixel-level annotation method requires a trained operator to draw a mask on top of cracks at the pixel-level resolution. This method is very time-consuming but provides the best accuracy for training machine learning models. Many studies [10] [11] [12] have used Fiji [13] to produce high-accuracy image annotations. Fiji is an open-source image processing package that is widely used for image annotation tasks in biological image analysis and scientific image processing. Its extensive plugin ecosystem makes Fiji a state-of-the-art tool for image annotation. One of Fiji's machine learning-based annotation tool, the Trainable Weka Segmentation (TWS) plugin uses a limited number of manual annotations to train a classifier and can segment the remaining image automatically.

2.4 Photogrammetry

Photogrammetry is the science of obtaining reliable information about physical objects through image measuring and interpreting processes. It remains a cornerstone of modern surveying and mapping practices. Aerial photogrammetry, utilizing UAVs or aircraft, efficiently generates high-resolution orthomosaics, Digital Elevation Models (DEMs), and topographic maps. In archaeology, it aids in the 3D recording and reconstruction of artifacts and excavation sites. In architecture and engineering, photogrammetry is employed for building information modeling (BIM), structural analysis, and the documentation of heritage sites [14]. Environmental monitoring benefits from photogrammetry's ability. These geospatial products find applications in urban planning, infrastructure development, and natural resource management [15] [16] [17] [18] [19] [20] [21] [22].

Despite the significant advancements, photogrammetry still faces challenges. Image quality, camera movement, and atmospheric conditions can affect the accuracy of 3D reconstructions [23]. Computational resources remain a concern, particularly for large datasets [24]. Future research needs to be conducted to focus on developing robust algorithms to address these challenges and exploring the integration of photogrammetry with other technologies, such as artificial intelligence and machine learning.

2.5 Applications of Computer Vision in Image Processing

Traditional image processing using computer vision applications in pavement distress detection often involves manual filtering to enhance distress visibility to technicians. This approach was time-consuming, as the survey vehicle produced a large quantity of images even within a short survey roadway section. To address these limitations, researchers explored various alternative techniques, including filter-based methods, segmentation and thresholding. While these methods are more efficient than manual image inspection, they are still time-consuming and not very accurate [25].

2.6 Machine Learning and Deep Learning in Image Recognition

With advancements in technology and computational power, machine learning and deep learning techniques have become increasingly prominent in image recognition. Machine learning, as defined by Murphy [26], is “a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty.” The core objective of machine learning is to empower computers to autonomously learn patterns, eliminating the need for direct human intervention in the prediction process. Machine learning algorithms are broadly categorized into two primary types: supervised and unsupervised learning. Supervised learning involves training computer programs on labeled examples, enabling them to apply learned knowledge to new, unseen data. In contrast, unsupervised learning is employed when labeled data is unavailable. In such cases, the computer program analyzes the dataset to identify hidden patterns and structures within the unlabeled data.

Essentially, machine learning can be described as an algorithm that predicts outcomes based on previously learned data, often referred to as training data [27]. Machine learning has been utilized across diverse fields, including agriculture, supply chain management, and cancer detection in medical imagery when large quantities of data are available [28].

While machine learning can generate accurate predictions based on previously learned data, it has its limitations when handling large and complex datasets. Consequently, deep learning was developed to deal with these datasets. Today, deep learning algorithms can effectively classify various forms of complex data, including speech, natural language, and images [28].

In deep learning, neural networks consist of multiple hidden layers and a substantial number of neurons are developed; these are known as deep neural networks [29]. Deep learning enables computers to automatically discover patterns for detection or classification. For example, in a deep learning network, the input layer includes images that are represented in the form of an array of pixel values. The first hidden layer typically discovers the presence or absence of edges of objects within the images. The second layer generally discovers a pattern in particular arrangements near the edge of the objects. The third layer integrates these patterns into larger combinations to detect familiar objects. This process of discovering key patterns or arranging them in a particular order is not designed by human engineers; rather, the deep learning algorithm autonomously performs this task, and this is the reason it has become a powerful tool across various domains [28].

However, deep learning models require substantial amounts of labeled data to achieve highly accurate prediction results. The MNIST dataset, one of the first data sets that were created for deep learning experimentation, comprises 60,000 labeled training images of handwritten digits in black and white, with an image size of 28x28 pixels [29]. Some deep learning models trained using the MNIDT dataset can recognize the digits with an accuracy of up to 98.13%. Another example is IBM's facial recognition dataset, containing one million labeled images. Their deep learning model has an impressive accuracy of 99.6% in identifying light-skinned male subjects. Despite its advancements in image recognition, deep learning has limitations in accurately classifying all image types, and it always requires substantial amounts of labeled images for high-accuracy predictions.

While deep learning algorithms have demonstrated high accuracy in image recognition in many fields, it is challenging to use them for pavement distress detection as it requires pixel-level accuracy, which is compromised due to the unavailability of accurately labeled data sets required to train the deep learning algorithm [31]. One of the first datasets created for deep learning applications in pavement image analysis was the Cracktree dataset. Created at Temple University, it contained 500 images of size 99x99 pixels with manual annotation [32]. These images were small, allowing the cracks on the road surface to be easily identified as each pixel contained more information about the road crack and road surface. However, these images are not suitable for an automated pavement surveying system because the images were taken by a mobile phone. In 2017, a study utilized a DSLR camera to collect 500 images, which were subsequently manually labeled [33]. Two more datasets, CrackNet and CrackNet II, containing 2,000 and 3,000 images, respectively, were created recently. Images included in these two datasets were manually labeled by multiple teams over one year, resulting in deep learning algorithms achieving accuracy ranging from 87.63% to 90.13% [34] [35].

In a previous study [36], video log images were processed using the Statistical Color Model (SCM) that includes an Artificial Neural Network to detect traffic signs. Recently, the integration of video and image processing, using cameras mounted on vehicles moving at speed, has generated significant research interest in advancing automated data collection processes.

Shen [37] developed a method to detect transverse cracks, longitudinal cracks and crack sections by processing video images collected from pavements. Hadjidemetriou, et al. [38] developed an algorithm incorporating built-in MATLAB features to process pavement surface videos captured by rear-view parking cameras, a common hardware in modern vehicles. The algorithm processed the videos using image entropy systems to create random gray-scale images by removing shadows from the video frames. Machine learning algorithms were then used to identify inconsistencies such as cracks or surface deformations and classify the pavements into healthy or distressed categories.

2.7 Convolutional Neural Networks (CNN) for Image Classification

Convolutional Neural Networks (CNN) are one of the most widely used architectures in deep learning, particularly for image processing tasks [39][40]. As a foundational technology, CNNs have been a significant breakthrough in both machine learning and image processing domains. CNNs are a class of deep learning models that use supervised learning to perform tasks such as image classification, its architecture revolutionized the field by demonstrating a highly effective approach to building deep learning models.

CNNs have proven highly effective in image classification tasks. Typically, they utilize two fully connected layers, followed by a softmax layer to predict image classes. The principle behind CNNs is straightforward: the network uses filters to extract feature maps from the original image through convolution operations, essentially matrix multiplications. These extracted feature maps are then reduced in size by pooling layers, followed by a backpropagation algorithm to update the filter weights and refine the learning process.

Several notable CNN architectures have contributed to the progression in this field. AlexNet, with its seven layers, was the first attempt to extend CNN depth by adding pooling layers [41]. VGG further improved this approach by replacing large convolution kernels with smaller 3x3 kernels, resulting in a deeper network with 19 layers [42]. However, VGG encountered challenges such as overfitting and gradient vanishing. GoogleNet increased the network depth to 22 layers by introducing the Inception module to manage computational complexity [43]. ResNet further advanced the field by incorporating residual blocks, allowing for the development of networks with 52 layers and beyond, effectively mitigating the vanishing gradient problem [44].

2.8 CNN Adaptation for Image Segmentation

While CNNs were initially designed for image classification, where a CNN model predicts a single class label for the entire group of images, their applicability has been successfully extended to segmentation tasks, where pixel-wise classification is required. Image segmentation aims to assign a class label to each pixel with an image group, effectively partitioning the image group into meaningful regions.

To adapt CNNs for segmentation, researchers have employed an encoder-decoder architecture, wherein the encoder extracts features from the input image, and the decoder reconstructs a pixel-wise prediction map of the same image. For example, Fully Convolutional Networks (FCNs) adapted classification networks such as VGG and ResNet for dense predictions by replacing the fully connected layers with convolutional layers [45]. In FCNs, the first five layers retain the convolutional architecture of traditional CNNs, while the fully connected layers are replaced by convolutional layers with kernel sizes of 7x7, 1x1, and 1x1, respectively. The decoder in FCNs performs up-sampling using deconvolution layers to ensure that the dimensions of the output prediction maps match those of the input images. This crucial step enables FCNs to generate pixel-wise predictions. A softmax layer is then applied to each pixel, facilitating pixel-wise classification and making FCNs highly effective for tasks such as semantic segmentation. By effectively adapting existing classification networks, FCNs have established a strong foundation for the development of future segmentation architectures.

2.9 Advances in Deep Networks for Segmentation

In addition to FCNs, more advanced models have emerged to accomplish segmentation tasks. U-Net, a prominent encoder-decoder architecture, further improved upon FCNs by introducing skip connections between the encoder and decoder, allowing for better recovery of spatial information lost during down-sampling in the encoder [46]. Similarly, DeepLab introduced atrous convolutions and Conditional Random Fields (CRFs) to improve segmentation accuracy, particularly in tasks that demand precise boundary detection [47].

For medical image segmentation, FCNs and U-Net have become the de facto standard. These models, with their ability to predict each pixel's class, are particularly suitable for tasks such as organ delineation, tumor detection, and lesion segmentation. FCNs and other convolution-based models have consistently achieved state-of-the-art performance on benchmark datasets specifically designed for medical image segmentation.

CHAPTER 3 DATA COLLECTION AND PREPARATION

This chapter outlines the systematic approach adopted to acquire two types of pavement condition data: high-resolution images provided by NCDOT and GoPro images collected by ITRE, as well as the methodologies for processing and annotating these datasets.

3.1 High-Resolution Images

3.1.1 Data Source

High-resolution pavement surface images used in this study were collected by a vendor using data collection vehicles. A total of 71,090 top-down images collected in 2021 from asphalt pavements in Division 5 were provided to the researchers by NCDOT. One sample is shown in Figure 1 below. These high-resolution images have a dimension of 1,028 x 2,011 pixels. For longitudinal cracks, one hundred images were identified and annotated at a pixel-level accuracy. The same was performed for transverse cracks and alligator cracks. The goal was to annotate a total of 300 images, 100 of each crack type, and use them to build the training, validation, and testing datasets for the deep learning process.

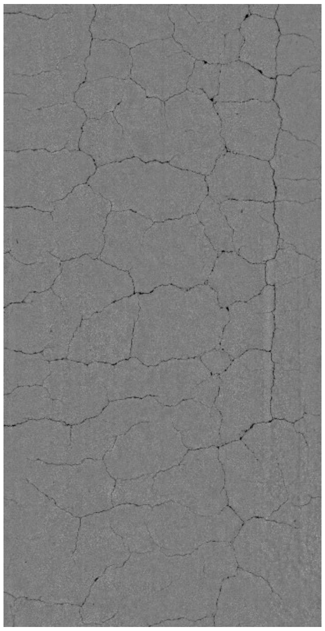


Figure 1. High Resolution Images Provided by NCDOT

3.1.2 Image Annotation

One of the main objectives of this research is to develop deep learning algorithms for evaluating primary and secondary roadway pavement conditions, specifically, providing answers to the following three research questions:

1. If a roadway surface image contains a crack (Yes/No)?
2. What type of crack is it (Longitudinal/Transverse/Alligator)?
3. What is the size (Width/Length/Area) of this crack?

To accurately answer the third research question above, a high-precision pixel-level annotation method is essential. Due to its low accuracy, the bounding box annotation method was not considered for this task. Instead, the Trainable Weka Segmentation (TWS) plugin, one of Fiji's machine learning-based annotation tools was selected for pixel-level annotation. This choice was mainly driven by two reasons: (1) TWS effectively addresses the time-consuming issue. It uses a small number of manual annotations to train a classifier, which can be saved and reused to automatically segment the remaining image, significantly reducing the overall annotation effort. (2) TWS is capable of high-accuracy annotation. As a pixel classifier, TWS excels in boundary detection, semantic segmentation, object detection and localization.

In this research project, image annotation of high-resolution images using Fiji's TWS plugin involves several steps, as described below.

Step 1. Load an image into Fiji.

Step 2. Launch the TWS plugin.

Step 3. Create two new classes. One is for cracking, and the other one is for background (non-cracking regions of the pavement surface).

Step 4. Define these two classes. As shown in Figure 2, red lines were drawn within the cracks (Class #1), and green rectangles in the background (Class #2).

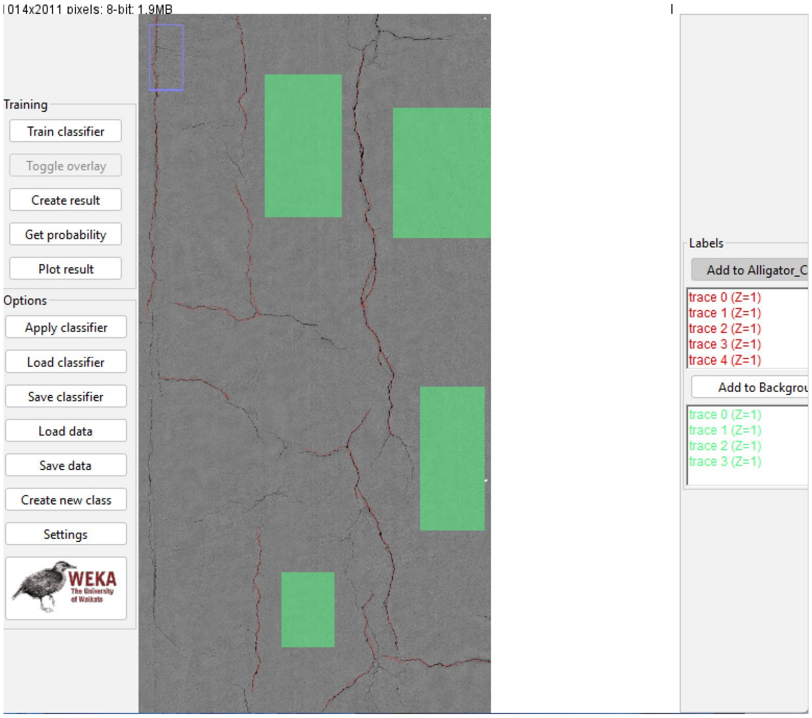


Figure 2. TWS Plugin in Fiji

Step 5. Train the classifier and annotate images. Essentially, this step is to train a learning scheme (the classifier) using the lines and rectangles (Class #1 and Class #2) defined in Step #4 as training data and then apply this classifier to annotate the remaining images in the dataset. The performance of the classifier can be iteratively improved over time by correcting or adding lines and/or rectangles (as shown in Step #4) to subsequent images in the dataset.

Step 6. Save the probability map. The output of the annotation process, the probability map, can be saved as a PNG binary image (Figure 3) The annotation of this image is complete, and the binary image serves as the ground truth for subsequent deep-learning analysis.

Step 7. Save the classifier.

Step 8. Apply the classifier to new images. The saved classifier can be applied to annotate the newly loaded images. For large image datasets, Fiji's batch processing functionality can be used to streamline the annotation process, provided sufficient computational resources are available.

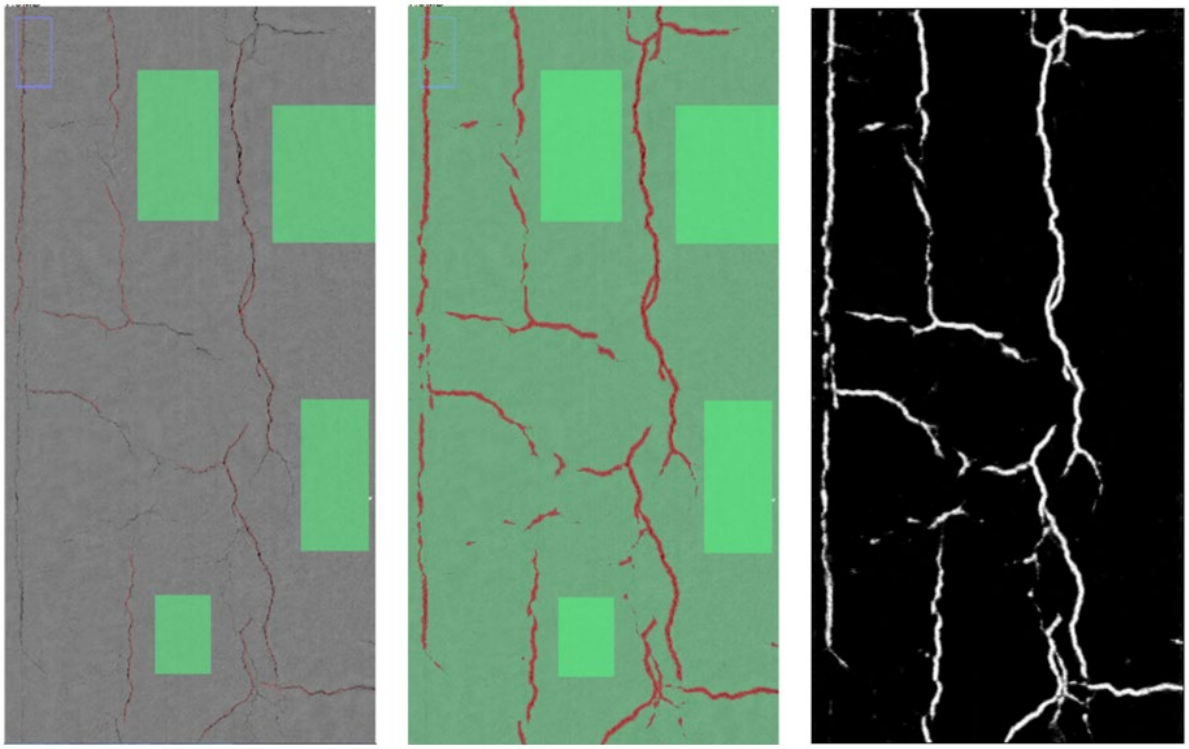


Figure 3. TWS Annotation (Left: Original Image; Middle: Probability Map; Right: Binary Image)

As shown in Figure 3, the predicted cracks in the binary image appear wider than the actual cracks in the original image. This issue can affect the accuracy of crack width quantification, highlighting a need for a post-image-annotation solution. To address this issue, morphological filters were employed.

Morphological filters are a set of image processing techniques used to remove noise and fill small gaps within an image. Key morphological operations include:

- Erosion: Shrinks the boundaries of objects in an image.
- Dilation: Expands the boundaries of objects in an image.
- Opening: Combines erosion followed by dilation, effectively removing small objects and smoothing object boundaries.

- Closing: Combines dilation followed by erosion, filling small holes within objects and smoothing object contours.

The Morphological Filters plugin and its Opening operation in Fiji were used to enhance visibility and analysis of cracks, and the result was satisfactory (Figure 4).

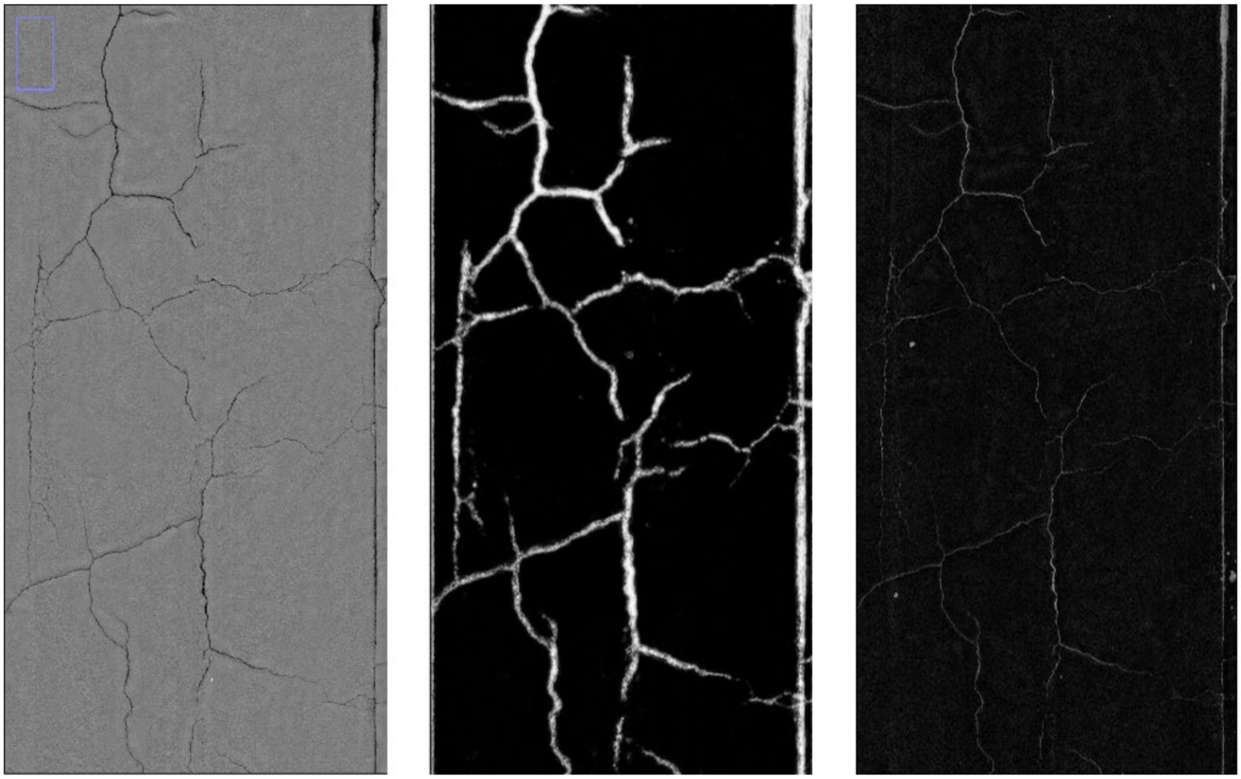


Figure 4. Improved Annotation (Left: Original Image; Middle: TWS Annotation; Right: Annotation after Morphological Filters were Applied)

3.2 GoPro Images

3.2.1 Camera Specifications

Based on findings of the literature review, a GoPro camera was selected as the primary tool for the cost-effective distress data collection method for secondary roadways in North Carolina, due to its cost-effectiveness, portability, and adaptability to dynamic roadway conditions.

Six site tests were conducted by UNCC researchers on February 5, 2022, March 3, 2022, August 24, 2022, September 7, 2022, September 21, 2022, and October 10, 2022, to determine the optimal specifications for GoPro camera applications. Results indicated that cracks are clearly visible when the data collection vehicle was driven at a low speed (~ 20 mph) under good lighting conditions. Furthermore, the analysis indicated that GoPro video clips were found to be more suitable for this research project compared to still images. Video clips offer continuous coverage of the roadway surface, eliminating the need to determine appropriate intervals for capturing still images. Additionally, the use of video clips significantly minimized the impact of the crack double-counting issue, improving the accuracy and reliability of the data collection process.

The camera specifications recommended by UNC Charlotte researchers are:

- Camera selection: GoPro 10 or GoPro 11
- Camera mounting location: rear mounting on the tailgate (Figure 5)
- Camera mounting height: 44” above the ground (Figure 5)
- Camera shooting angle: 40°
- Video shooting mode: Standard or Wide
- Driving speed: 20 mph
- Weather Restrictions: “Pavement images are not collected during rain, snow, or under other conditions contributing to poor pavement visibility” [48].

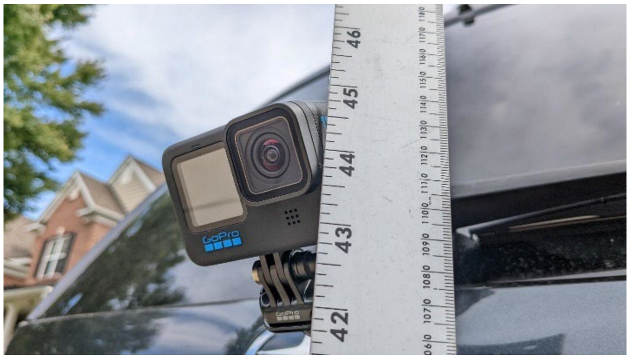


Figure 5. Recommended Setup of a GoPro Camera

3.2.2 Data Collection

To prepare for field data collection, the ITRE/NCSU research team used information provided by the NCDOT that included pavement condition ratings. This information was filtered to show street segments from the Raleigh, NC area for easy access from North Carolina State University. A research team member scoured the data for a wide range of pavement conditions that were near one another. As such, street segments in Cary, NC were chosen that ranged from very poor to excellent pavement conditions, as shown in Figure 6 below. This allowed the research team to capture varying pavement conditions for the crack annotation portion of the project. After identifying the street segments of interest, the field data collection followed.

The field data collection for this project occurred over two days. The research team from UNCC had previously conducted some testing of the appropriate installation height and angle for the GoPro camera, and provided this information to the NCSU team members. The field research team attached a GoPro camera to the rear window of a minivan and positioned the camera lens 44 inches above the pavement at an angle of 40 degrees. As often as was safe, the driver maintained a speed of 20 miles per hour, as prescribed by the UNCC team members. This was feasible between controlled intersections due to strobe lights and printed vehicle magnets attached to the minivan throughout the field data collection process.

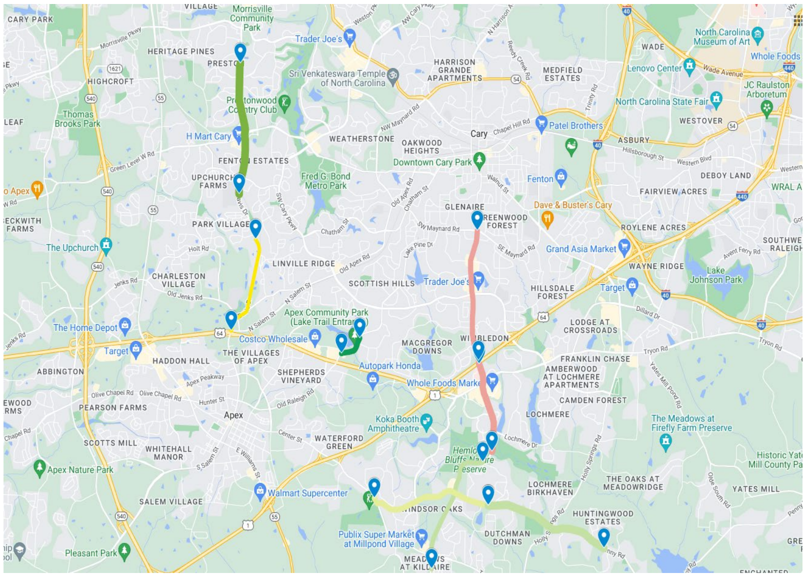


Figure 6. Map of Study Segments

3.2.3 Image Distortion Correction

3.2.3.1 Image Distortion Correction using Hugin

The image annotation process for GoPro images is different from that of for high-resolution images provided by NCDOT. Before annotating cracking in images, the GoPro images must be corrected for distortion that is introduced because of the lens type of the GoPro camera, which skews the image, particularly toward its extent and creates a slight fisheye effect – a rounding effect on the extent of the image. Likewise, as with any camera lens or human sight in general, locations farther away from the camera appear closer to one another and stretched in the line-of-sight axis. The latter distortions are more important to remove than the fisheye effect if the fisheye effect is subtle, as in the images captured for this project. This is especially true because the area of interest is closer to the center of the image and not to the image extent, where the fisheye effect is more pronounced. An example of an image that has not undergone distortion correction is below in Figure 7. For reference, the blue line in the middle of the lane is approximately 15 feet long and extends from the camera to the furthest point included in the ultimate area of interest for this image. This line is shown in the following images for continuity.

As can be seen in Figure 7, which is an original image captured by the research team's GoPro and used for this project, there is a fisheye effect along with typical visual compression as objects are further away, highlighted by lines overlaying the image. The red lines show how the lane lines appear closer together the further they recede from the camera, which is typical visual compression for how the human brain processes imagery. Likewise, note the short green lines between the red lines and the lane markings, which demonstrate the slight fisheye effect caused by the GoPro camera lens. Both effects should be accounted for and as far as is possible, corrected in the image. However, before conducting distortion corrected, raw images like the one in Figure 7 can be cropped to remove obvious areas of disinterest (i.e., the vehicle following the research van, the concrete curbs/medians, or even portions of the pavement containing debris), as seen below in Figure 8, where the following vehicle is omitted.

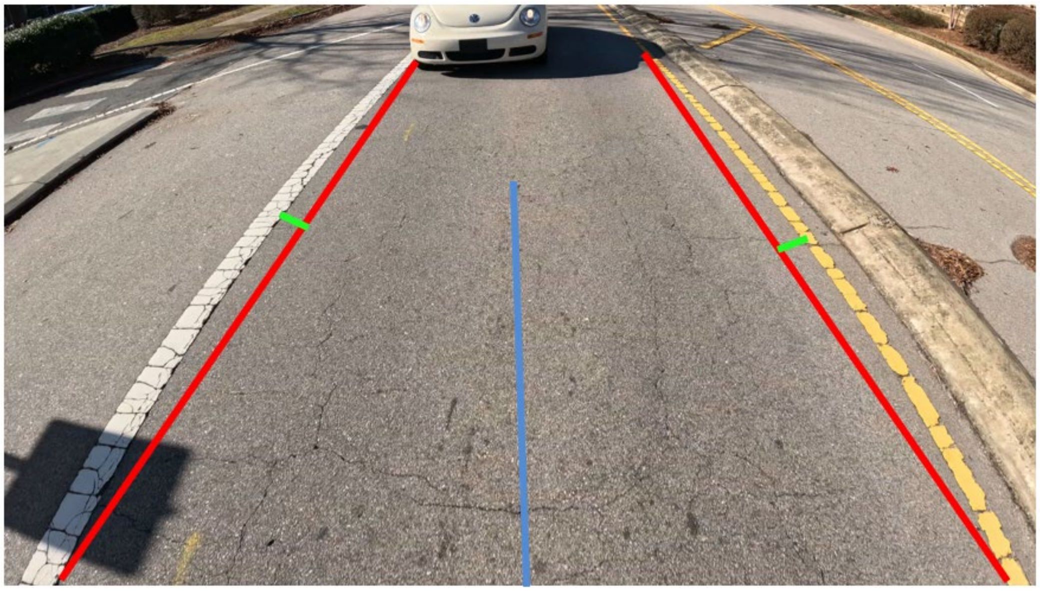


Figure 7. Raw Image with no Distortion Correction

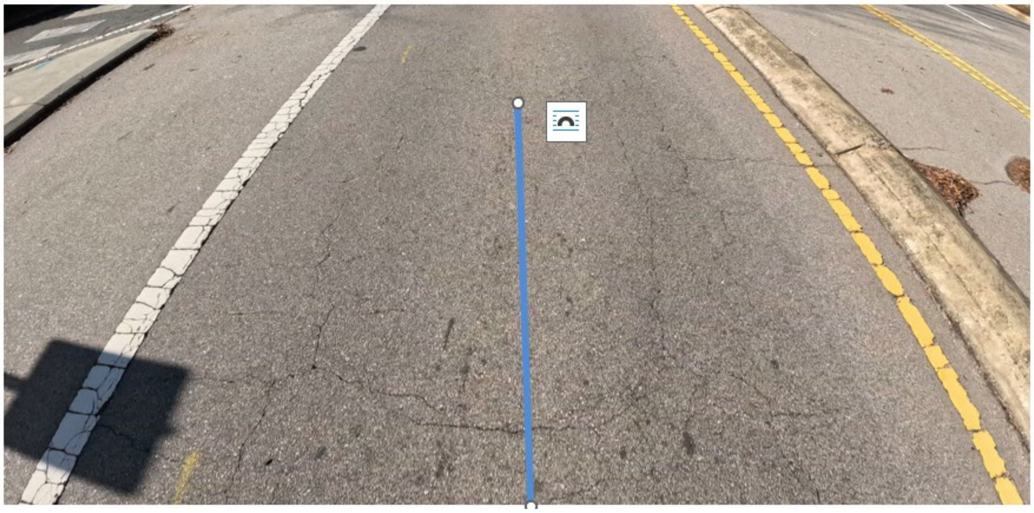


Figure 8. Cropped Raw Image

Researchers at UNCC recommended utilizing the open-source software titled Hugin for distortion correction. This software is free to use and effective in mostly or completely removing these two types of distortion. A distortion-corrected image before the second cropping is shown below in Figure 9. After correcting distortion to the extent possible, the image is cropped again to leave only the area of interest. As with before, this removes curbs, medians, debris, adjacent vehicles, and sometimes even lane markings. As stated previously, the blue line in the middle of the lane shows the original area of interest referenced from the first image until now. Again, this line is approximately 15 feet long on the image, showing just how much area can be included in a single image even after distortion correction and cropping, even from a camera less than four feet above the ground.

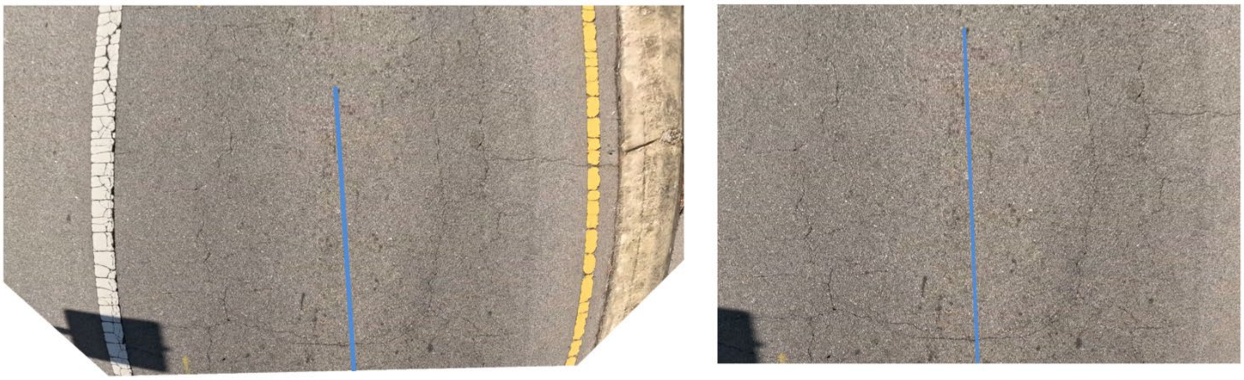


Figure 9. Distortion-Corrected Image (Left: After the First Cropping; Right: After the Second Cropping)

3.2.3.2 Image Distortion Correction using Photogrammetry

Photogrammetry is another technology that was used in this research project to correct image distortion. Photogrammetry retrieves reliable spatial information about physical objects and their surroundings by interpreting photographs. It can be used for but not limited to 3D modeling and 2D mapping. Georeferencing and perspective correction are two essential steps integrated into photogrammetry. Therefore, photogrammetry is one potential solution for preparing qualified images for crack detection on secondary roads. To implement professional photogrammetry, Pix4DMapper software [49], a leading photogrammetry software for drone mapping was employed. One of the most critical requirements for qualified georeferencing and correction via photogrammetry is the overlapping rate, which is the overlapping among images or frames in our case. A formula was derived based on the corresponding variables in Equation 1 to retrieve a specific overlapping rate.

$$FPS = \frac{v}{(Tan(\beta+\alpha)-Tan(\beta-\alpha))*h*(1-r)} \qquad (1)$$

In this equation, FPS is the required Frames Per Second to keep the overlapping rate at a given speed (v) and configurations of the camera, which include half of the Field of View (FOV), α; camera orientation concerning the perpendicular line to level ground, β; camera height to level ground, h; and expected overlapping rate, r.

The principles are demonstrated in Figure 10.

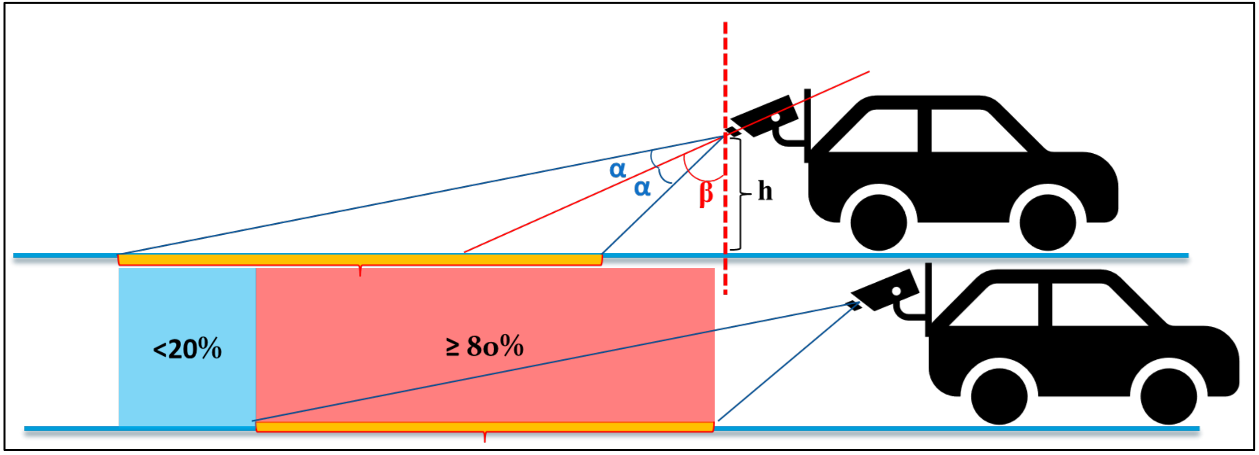
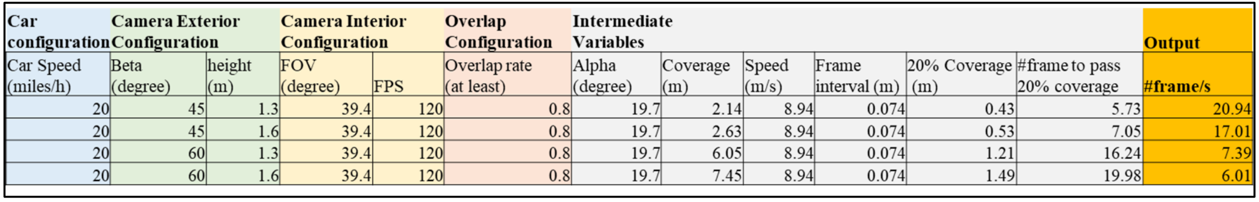


Figure 10. Factors to Ensure a Particular Overlapping Rate

In Figure 10, the orange bolded line is the area captured by the photo. The red and blue areas represent the overlapping part and not overlapping parts with rates on them between two consecutive frames.

Beyond the derived formula, the calculation of the expected FPS was further automated by implementing this formula in Excel. As shown in Table 1, the inputs are categorized into four groups: car configuration, camera exterior configuration, camera interior configuration, and overlap configuration.

Table 1. Estimation of Expected Frame per Second (FPS) Based on Different Variables



Car configuration	**Camera Exterior Configuration**		**Camera Interior Configuration**		**Overlap Configuration**	**Intermediate Variables**						**Output**
Car Speed (miles/h)	Beta (degree)	height (m)	FOV (degree)	FPS	Overlap rate (at least)	Alpha (degree)	Coverage (m)	Speed (m/s)	Frame interval (m)	20% Coverage (m)	#frame to pass 20% coverage	**#frame/s**
20	45	1.3	39.4	120	0.8	19.7	2.14	8.94	0.074	0.43	5.73	20.94
20	45	1.6	39.4	120	0.8	19.7	2.63	8.94	0.074	0.53	7.05	17.01
20	60	1.3	39.4	120	0.8	19.7	6.05	8.94	0.074	1.21	16.24	7.39
20	60	1.6	39.4	120	0.8	19.7	7.45	8.94	0.074	1.49	19.98	6.01

In addition, several experiments were conducted using the frames extracted from the GoPro video, and 100 frames were used for a preliminary experiment. The results are demonstrated in Figure 11, where the cracks and a manhole in the middle of the road can be clearly seen. However, the manhole on the road's edge seems distorted. This is because the GoPro camera is set to shoot the area with a focus on the road, resulting in less overlapping of imagery on the edge of the road.

For photogrammetry, the perspective correction seems well conducted, as the round shape of the manhole in the middle of the road is well preserved. Therefore, it can be inferred that the shape of the cracks, especially those in the middle of the road, should be also in good shape. However, georeferencing problems still exist, as suggested in Figure 12. During the imagery collection, one camera was set up on the back of the vehicle and drove along one road lane. Therefore, the trajectory of several consecutive frames formed a straight line, which caused uncertainty of the z-axis as shown in Figure 12C. The surface of the road should face upward but the results show that there is a big slope from one side to the other of the road (Figure 12C). This problem can be mitigated by using multiple cameras mounted on different locations of the vehicle to inform the algorithms in photogrammetry about a correct z-axis.

In Figure 12, Figure 12A is the reconstructed trajectory of the camera when taking the imagery. Figure 12B is a zoom-in view to show the imagery and the trajectory of the camera. Figure 12C is a front view to show the incorrect z-axis. The bright background indicates upward, and the dark background refers to downward in terms of gravity.

Another experiment was conducted at a parking lot to test the performance of photogrammetry in terms of georeferencing and perspective correction. Figure 13A shows the GPS locations (red dots) of the camera when collecting corresponding photos. The extracted frames are shown in Figure 13B. Figure 13C shows differences between the GPS trajectory (blue dot) and the calibrated camera locations (green dot) through photogrammetry. Figure 13D delineates the orthomosaics photo of the collected imagery.

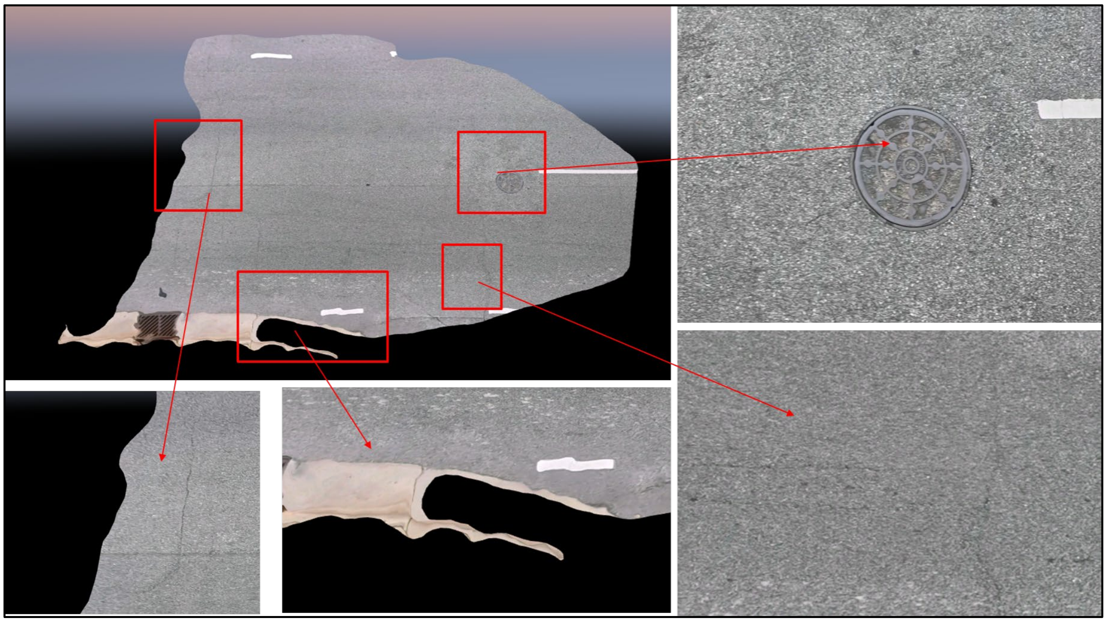


Figure 11. The 3D Mesh Reconstructed by the Selected Frames from the GoPro Video

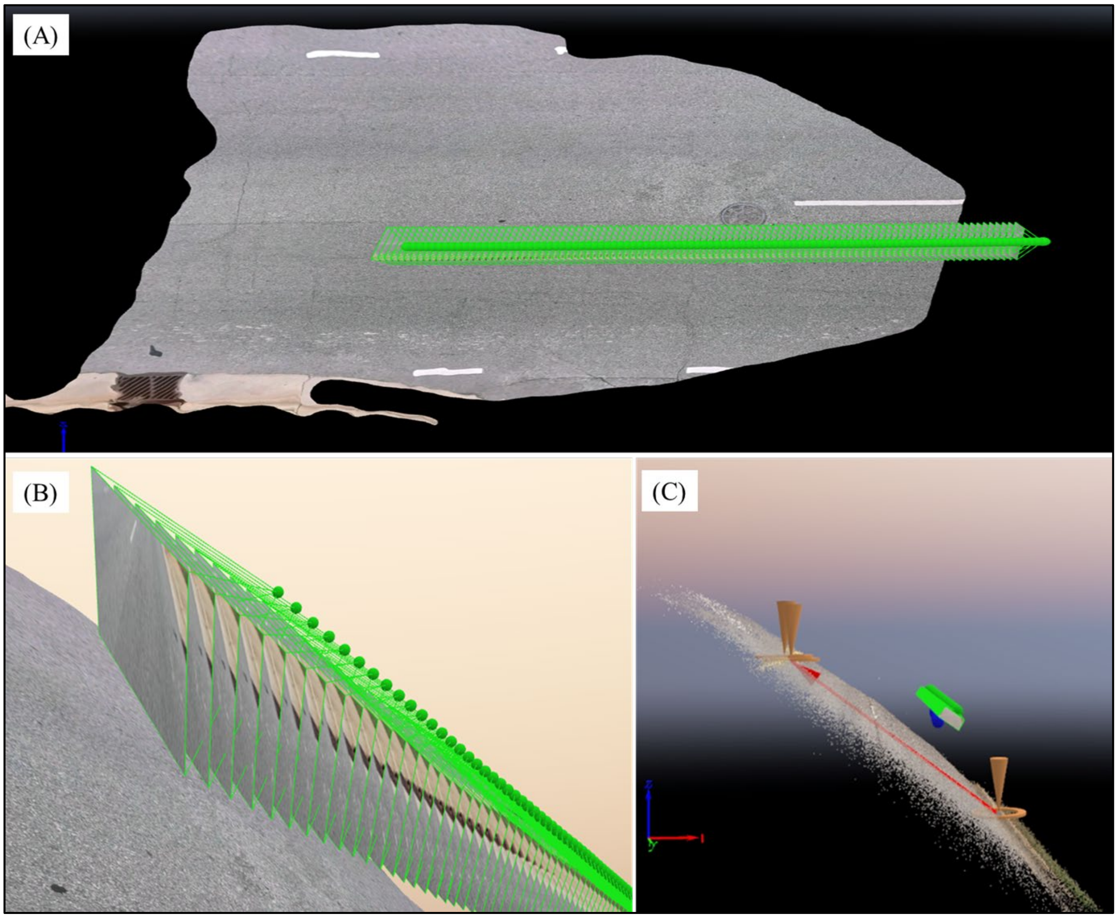


Figure 12. Challenges Encountered during the Photogrammetry Process

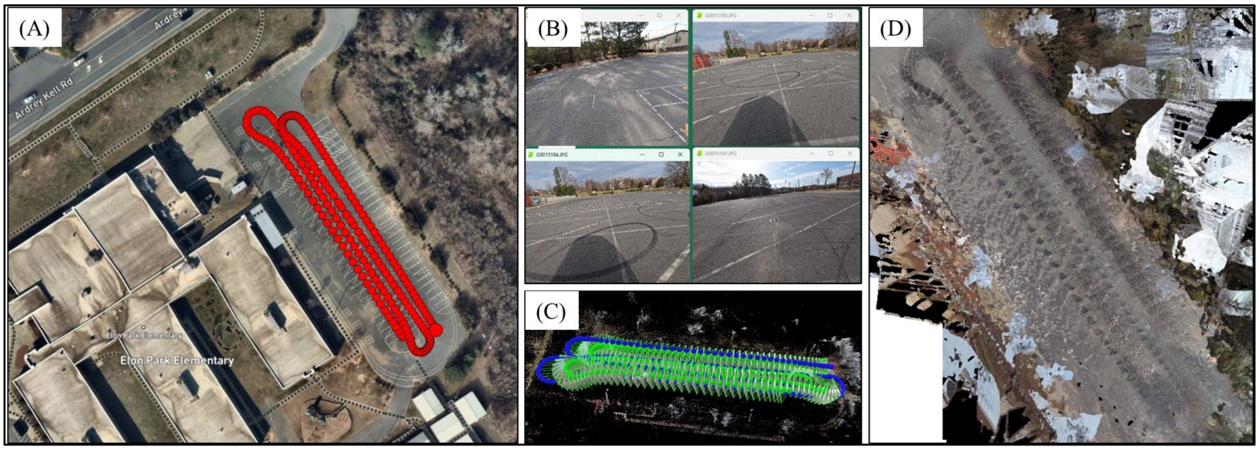


Figure 13. The Photogrammetry Process

Photogrammetry is a widely used approach for surveying and mapping based on photos. In this research project, photogrammetry is needed if the precision of the crack width quantified by the proposed method matters. Based on the experiments so far, control points are required since it is not reliable to simply rely on the low-precision location sensor from GoPro. It works well to collect GPS control points in the experiment, but it can be a challenge for the entire primary road network in terms of labor and cost.

3.2.4 Image Annotation

Upon completion of distortion correction (Section 3.2.3.1), the image is now ready for annotation. The annotation process is the same as the steps described in Section 3.1.2, and the research team at ITRE also established a goal of 100 annotated images per crack type (longitudinal, transverse, and alligator).

Upon completion of the first annotation, the researchers trained the classifier through an executable command. This took as little as 1-2 minutes later in the annotation stage, and as many as 10-15 minutes at the beginning of the annotation process. The resulting probability maps were often very "noisy" through the first few trainings and were gradually improved through each successive training. Noise, in this case, is an area of the image that should have been identified by the classifier as background (i.e., containing no cracks), but instead included small "cracks". This noise could be due to obvious objects such as shadows, pavement markings, and stains on the pavement, or simply from background pixels being similar enough in color to a pixel included in a crack annotation. A sample image alongside a corresponding noisy probability map is shown in Figure 14, with portions of the noisy area identified in both images (red rectangles).

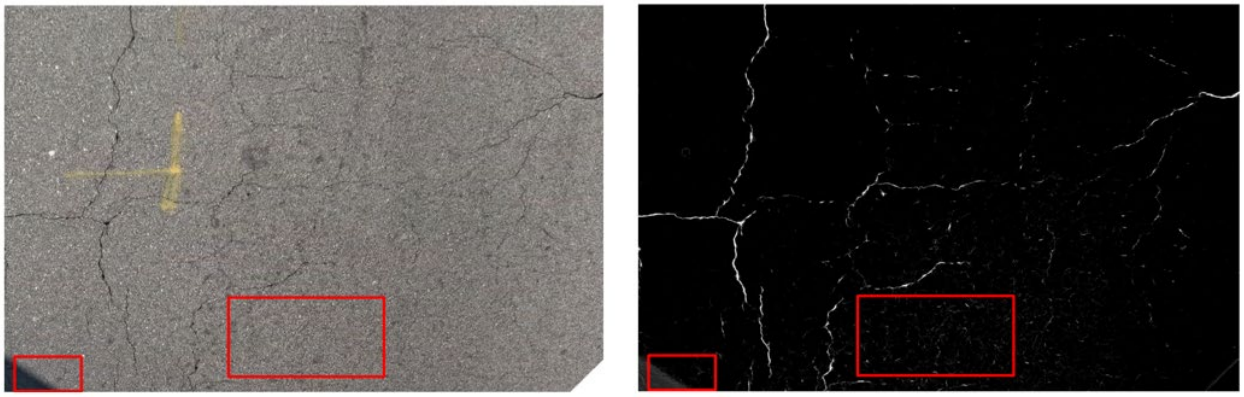


Figure 14. Noisy Probability Map of a Sample Image

This issue was addressed by iteratively training the classifier with additional information to make it "smarter." A sample image is shown in Figure 15, and the corresponding images annotated using a classifier that has been trained multiple times are included in Figure 16.

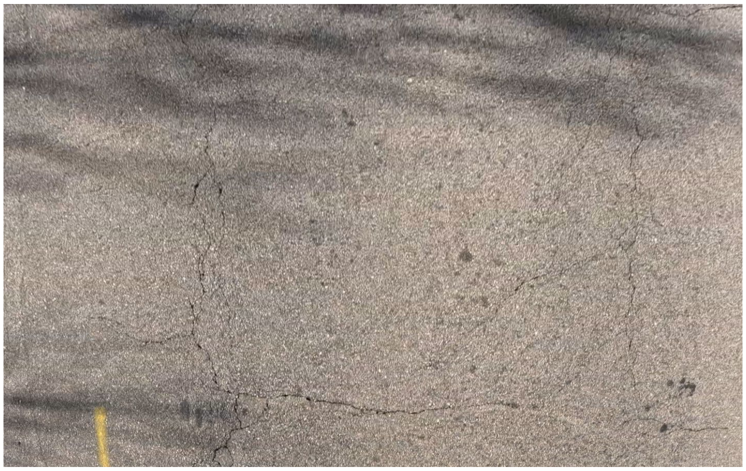
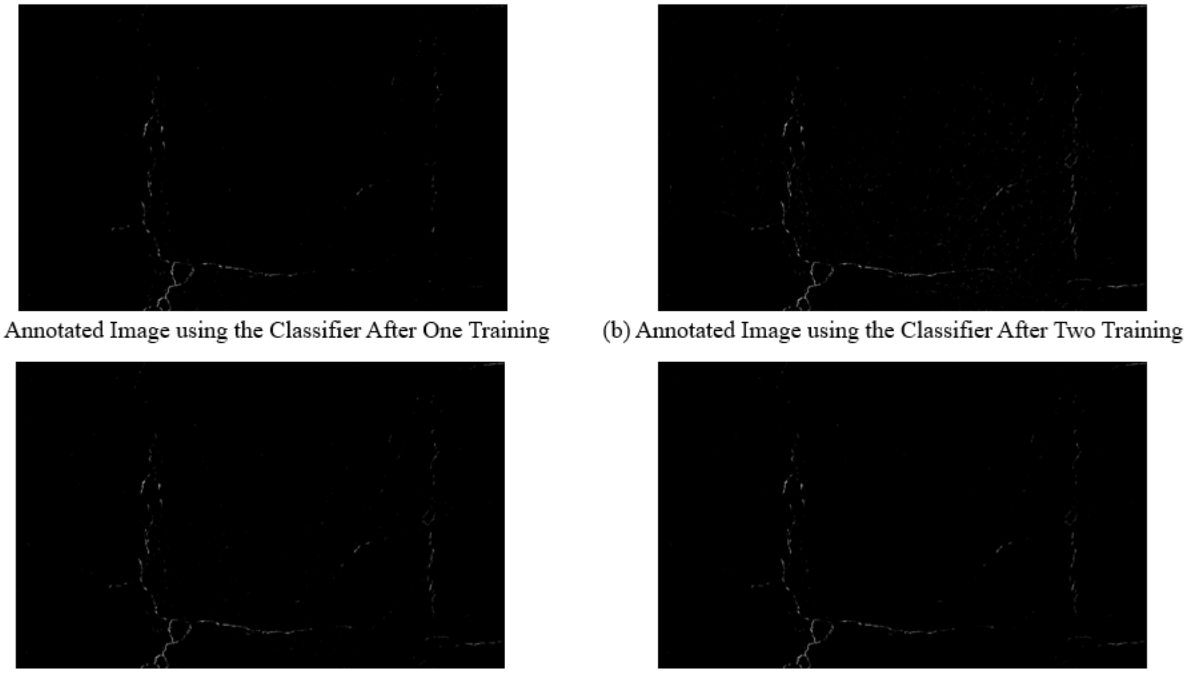


Figure 15. A Sample Image for Annotation



(a) Annotated Image using the Classifier After One Training

(b) Annotated Image using the Classifier After Two Trainings

(c) Annotated Image using the Classifier After Three Trainings

(d) Annotated Image using the Classifier After Four Trainings

Figure 16. Annotated Images using a Classifier After Multiple Trainings

Lessons learned from annotating GoPro images are included in Appendix A.

CHAPTER 4 DATA ANALYSIS AND RESULTS

This chapter presents a comprehensive analysis of high-resolution images and GoPro images, with a focus on crack classification, crack segmentation, and crack quantification. The corresponding analysis results are provided at the end of each section.

The various approaches employed for each task are summarized in Figure 17. The approach that achieved the highest performance was identified and is recommended for future research efforts.

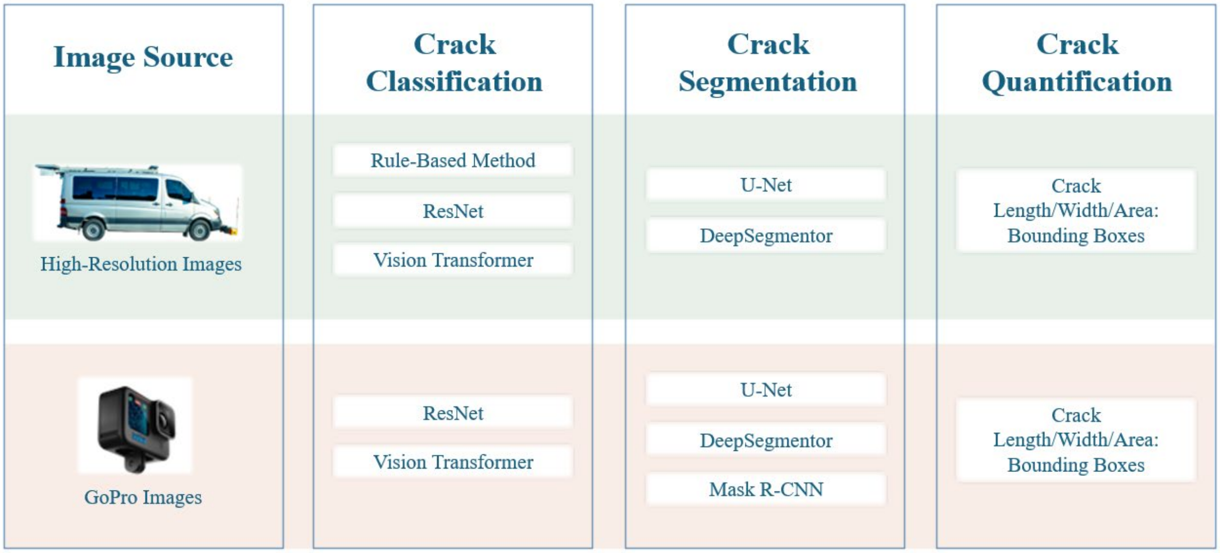


Figure 17. Image Analysis Approaches

4.1 Analysis of High-Resolution Images

The analysis of high-resolution images provided by NCDOT involves four core tasks, each designated to address specific aspects of pavement distress evaluation:

- **Crack Classification**: This task focused on identifying and categorizing different crack types (e.g., longitudinal, transverse, alligator) using traditional rule-based methods and deep learning models, such as Vision Transformers and ResNet. The method that provided the most accurate results was identified.
- **Crack Segmentation**: This task involved generating detailed delineation of cracks using advanced deep learning architectures such as U-Net and DeepSegmentor. The resulting segmentations can be used to quantify the size of cracks.
- **Crack Quantification**: This task involved calculating the size, such as the length, the width, or the area, of cracks. The results can be used to determine the severity of cracks.
- **Performance Evaluation**: This task provided evaluation metrics to assess the effectiveness of both classification and segmentation methods. Metrics such as accuracy, precision, recall, F1-score, and Intersection over Union (IoU) were employed to ensure robust and reliable model performance:

- **Accuracy**: Measures the overall correctness of the model by calculating the ratio of correctly predicted crack types to the total number of samples. It provides a general indication of the model's performance.
- **Precision**: Assesses the model's ability to avoid false positives by measuring the proportion of true positive predictions among all positive predictions. High precision indicates that the model is reliable in its positive classifications.
- **Recall**: Evaluates the model's ability to identify all actual positive cases, focusing on minimizing false negatives. High recall ensures that the model captures the majority of relevant crack instances.
- **F1-Score:** Represents the harmonic mean of precision and recall, providing a balanced metric that is particularly useful for imbalanced datasets.
- **Intersection over Union (IoU):** Measures the overlap between predicted and ground truth regions, commonly used in segmentation tasks. It quantifies the accuracy of pixel-level predictions, with higher IoU values indicating better alignment between predicted and actual crack boundaries.

The following sections detail the steps involved in each task. At the end of each section, the analysis results and corresponding performance metrics are presented to evaluate the effectiveness of the deep learning models.

4.1.1 Crack Classification

One of the main research objectives is to use the deep learning model to classify crack types – specifically longitudinal, transverse, and alligator – in roadway surface images. To achieve this objective, three approaches were explored:

1. **The Rule-Based Method:** This method relies on predefined criteria to classify cracks based on their geometric properties. While this method provides a straightforward and interpretable approach to crack classification, oftentimes its effectiveness is limited when handling complex or noisy crack patterns.
2. **ResNet:** This method excels in hierarchical feature extraction, enabling robust and efficient crack classification. Its residual learning framework mitigates the vanishing gradient problem, allowing for the training of deeper networks with superior performance.
3. **Vision Transformer:** Vision Transformers (ViTs) use self-attention mechanisms to capture global features within high-resolution images. Unlike CNNs, which focus on local features, ViTs analyze the entire image at once, making them particularly effective for identifying complex patterns and long-range dependencies in pavement distress data.

Data Source

For the crack classification task, the high-resolution roadway surface images provided by NCDOT were manually categorized into three datasets based on crack type: Longitudinal, Transverse, and Alligator. The Longitudinal dataset includes 100 annotated images featuring longitudinal cracks, the Transverse dataset includes annotated 101 images featuring transverse cracks, and the Alligator dataset includes annotated 94 images featuring alligator cracks. These three datasets are carefully created to ensure a balanced representation of the three crack types, facilitating robust and unbiased model training and evaluation.

To develop reliable classification models, each dataset was divided into training and testing subsets using an 70:30 split ratio:

- **Longitudinal**: 70 images for training, 30 for testing
- **Transverse**: 71 images for training, 30 for testing
- **Alligator**: 63 images for training, 32 for testing

Several preprocessing techniques were applied to optimize the images for deep learning workflows. Due to the high-resolution of the original images (up to 1045x2011 pixels), all images were resized to a standardized dimension of 384x384 pixels. This resizing strategy effectively balances the preservation of crucial visual details with the need to conserve GPU memory for efficient training. Additionally, pixel intensity values were normalized to ensure consistency across the dataset and enhance the model's ability to generalize across diverse images. These preprocessing steps collectively prepare the datasets for deep learning training, enabling the models to learn effectively from high-resolution imagery while maintaining computational efficiency.

(1) Rule-Based Method

The rule-based method was developed using OpenCV's automatic contour detection function. A contour is defined as a curve that connects all continuous points along the boundary of a crack, with similar intensity values. Essentially, it represents the outline of a crack. This rule-based approach relies on three rules derived from the geometric properties of these detected contours to classify cracks into three categories.

The method involves the following three steps:

1. Contour Detection. All contours within the high-resolution image are extracted using OpenCV's findContours function. This step identifies the boundaries of cracks based on pixel intensity variations.
2. Contour Property Analysis. For each extracted contour, the length and width of its bounding box are calculated. These geometric properties serve as the basis for crack classification.
3. Classification Rules:
    - **Longitudinal Cracks:** If the length of the bounding box is significantly greater than its width, the crack is classified as Longitudinal.
    - **Transverse Cracks:** If the width of the bounding box exceeds its length, the crack is classified as Transverse.
    - **Alligator Cracks:** If two or more bounding boxes intersect at an angle between 70 and 90 degrees, the crack is classified as Alligator.

Analysis Results

As shown in Table 2, the rule-based method achieved an overall accuracy of 89%. While this method is highly effective in classifying alligator and longitudinal cracks, its performance in classifying transverse cracks is notably weaker, with 33 misclassified transverse crack images and an accuracy of only 67%.

A key factor contributing to this misclassification is the presence of longitudinal cracks within the same images as transverse cracks. This co-occurrence often results in bounding boxes that are significantly larger than the actual transverse cracks, leading to false classifications. Apparently, noisy crack patterns can limit the effectiveness of this method in such scenarios.

Table 2. Crack Classification: Rule-Based Method Results

Crack Type	**Total Number of Images**	**Correct Prediction**	**Wrong Prediction**	**Accuracy**
Alligator	94	94	0	100%
Longitudinal	99	99	0	100%
Transverse	101	68	33	67%
Total	294	261	33	89%

(2) ResNet

In this research project, ResNet, or Residual Network, was selected for crack classification due to its ability to effectively address the vanishing gradient problem commonly encountered in training deep neural networks, and its capacity to efficiently learn hierarchical feature representations from high-resolution images.

- **Model Design**: ResNet utilizes skip connections, which allows gradients to flow directly across layers during backpropagation. This innovative design mitigates the vanishing gradient problem, allowing for the training of deeper networks without significant loss of accuracy. For this research project, the vanilla ResNet architecture was employed with its default settings, and pre-trained weights from ImageNet were leveraged to accelerate the training process and harness the benefits from transfer learning. The architecture of a ResNet model is illustrated in Figure 18.

  To adapt ResNet for this specific crack classification task, the earlier layers of the network were frozen. These frozen layers output 512 features, preserving the pre-trained knowledge from ImageNet. A Fully Connected (FC) layer was added on top of the frozen layers to predict the three crack types: Longitudinal, Transverse, and Alligator. This approach enhances computational efficiency by focusing the training process on fine-tuning the final layers specifically for this classification task.

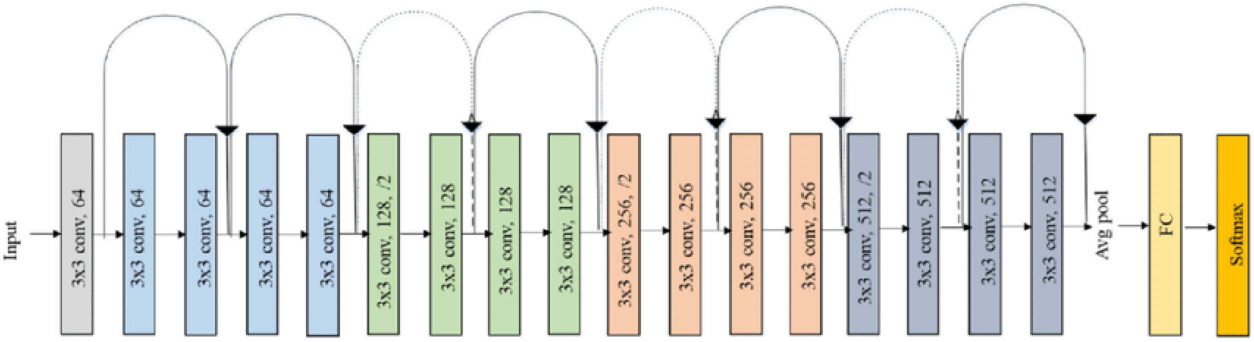


Figure 18. Vanilla ResNet Architecture [26]

- **Training Process**: The resized and normalized images were fed into the ResNet model, which progressively extracts features such as crack orientation, shape, and intersection patterns to classify the cracks.
- **Training Hyperparameters**: The training hyperparameters used for training the ResNet model are included in Table 3. These parameters were carefully selected to optimize model performance and ensure efficient convergence during training.

Table 3. Hyperparameter for the ResNet Model

Hyperparameter	**Value**	**Description**
Epochs	200	Number of complete passes over the training dataset
Batch Size	32	Number of samples processed in one forward/backward pass
Initial Learning Rate (LR)	0.0002	Learning rate at the start of training (epoch 0)
Optimizer	Adam/SGD	Optimization algorithm used for training
Amsgrad	Ture	Enables the Amsgrad optimizer for improved stability
Weight Decay	1.00E-05	Regularization parameter to prevent overfitting by penalizing large weights
Validation Dataset Split	0.2	Proportion of the dataset reserved for validation
Validation Epochs Every n	20	Validation is conducted every n training epochs

Analysis Results

As shown in Table 4, the ResNet method achieves an accuracy of 97% after 200 epochs. This approach demonstrates exceptional effectiveness in classifying all three crack types: alligator, longitudinal, and transverse. The high accuracy indicates the model's robustness and its ability to generalize across diverse crack patterns, making it a reliable solution for pavement distress classification.

Table 4. Crack Classification: ResNet Results

Epoch	**Pre-train**	**Optimizer**	**SGD**	**Decay**	**Accuracy**	**F-1**	**Recall**	**Precision**
80	ImageNet	Adam	FALSE	1.00E-05	0.94	0.94	0.94	0.95
100	ImageNet	Adam	FALSE	1.00E-05	0.95	0.95	0.95	0.96
200	ImageNet	Adam	FALSE	1.00E-05	0.97	0.97	0.97	0.97
200	N/A	Adam	FALSE	1.00E-05	0.80	0.80	0.80	0.88

(3) Vision Transformer

Vision Transformer (ViT) represents a cutting-edge approach to image classification, utilizing self-attention mechanisms rather than traditional convolutional layers. This architecture is particularly well-suited for high-resolution images, as it can capture global contextual information and long-range dependencies. Furthermore, ViT is particularly effective in handling complex patterns, such as intersecting cracks and irregular geometries, which often times pose challenges for traditional convolutional models. Its ability to learn global features makes it a powerful tool for distinguishing subtle variations in road crack types.

- **Model Design:** The Vision Transformer (ViT) processes an input image by dividing it into fixed-size patches, treating each patch as a token-like word in natural language processing. These tokens are then embedded and processed using multi-head self-attention mechanisms, allowing the model to capture long-range dependencies and understand relationships across the entire image.

  In this research project, the vanilla Vision Transformer architecture was utilized with its default settings, and pre-trained weights from ImageNet were used to expedite the training process and leverage the benefits of transfer learning. The architecture of a ViT model is illustrated in Figure 19.

  To adapt the ViT for this specific task, the earlier layers of the network were frozen, which includes the multi-head attention outputs. A classification head was added to the frozen layers to enable prediction across the three crack classes: Longitudinal, Transverse, and Alligator. This approach optimizes computational resources while focusing fine-tuning efforts on the final layers, ensuring accurate and efficient classification.

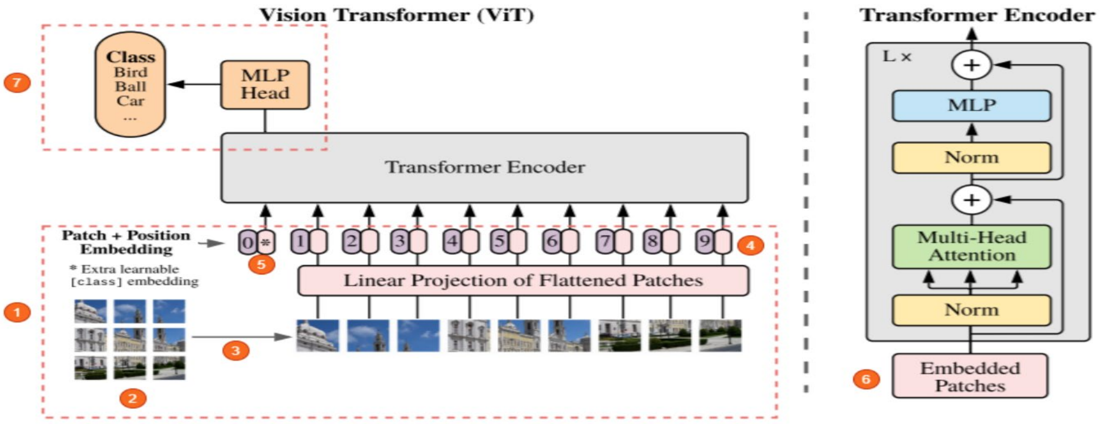


Figure 19. Network Structure of Vision Transformer [41]

- **Training Process:** The preprocessed 384x384 images were split into smaller patches, and positional embeddings were added to preserve spatial information. These embeddings were then processed through the transformer layers to classify the crack types.
- **Training Hyperparameters:** The hyperparameters used for training the Vision Transformer are included in Table 5.

Table 5. Hyperparameter for the ViT Model

Hyperparameter	Value	Description
Epochs	200	Number of complete passes over the training dataset

Batch Size	32	Number of samples processed in one forward/backward pass
Initial Learning Rate (LR)	0.0002	Learning rate at the start of training (epoch 0)
Optimizer	Adam/SGD	Optimization algorithm used for training
Amsgrad	Ture	Enables the Amsgrad optimizer for improved stability
Weight Decay	1.00E-05	Regularization parameter to prevent overfitting by penalizing large weights
Validation Dataset Split	0.2	Proportion of the dataset reserved for validation
Validation Epochs Every n	20	Validation is conducted every n training epochs

Analysis Results

As shown in Table 6, the Vision Transformer method achieves an accuracy of 98% after 80 epochs. Similar to the ResNet method, this approach demonstrates exceptional effectiveness in classifying all three crack types: alligator, longitudinal, and transverse. The high accuracy underscores the model's robustness and its ability to generalize across diverse crack patterns, making it a reliable solution for pavement distress classification.

Table 6. Crack Classification: ViT Results

Epoch	Pre-trained Model	Optimizer	SGD	Decay	Accuracy	F1	Recall	Precision
40	ImageNet	Adam	FALSE	1.00E-05	0.96	0.96	0.96	0.96
60	ImageNet	Adam	FALSE	1.00E-05	0.97	0.97	0.97	0.97
80	ImageNet	Adam	FALSE	1.00E-05	0.98	0.98	0.98	0.98
100	ImageNet	Adam	FALSE	1.00E-05	0.93	0.93	0.93	0.94
120	ImageNet	Adam	FALSE	1.00E-05	0.90	0.90	0.90	0.92

4.1.2 Crack Segmentation using U-Net and DeepSegmentor

Image segmentation plays a crucial role in road crack analysis by providing pixel-level delineation of crack regions. Unlike classification, which assigns a single label to an image, segmentation offers detailed insights into the location, shape, and extent of cracks. This information is vital for accurately assessing the severity of road damage. In this research project, two state-of-the-art deep learning models for segmentation were explored: U-Net and DeepSegmentor. Both models are well-suited for high-resolution imagery and enable precise segmentation of road cracks.

Data Source

Accurate segmentation of road cracks requires a large, high-quality dataset with well-annotated ground truth. However, the annotated dataset described in Section 3.3 lacks a sufficient number of comprehensive annotations suitable for segmentation tasks. To address this limitation, publicly available road crack datasets were used.

The selected dataset comprises 11,298 high-resolution images, each accompanied by detailed pixel-level annotations, making it an ideal data source for training and evaluating segmentation models. This Crack Segmentation Dataset is publicly accessible online at https://www.kaggle.com/datasets/lakshaymiddha/crack-segmentation-dataset. The distribution of the dataset, illustrated in Figure 20, demonstrates its diversity across various crack types, ensuring robust model training and evaluation.

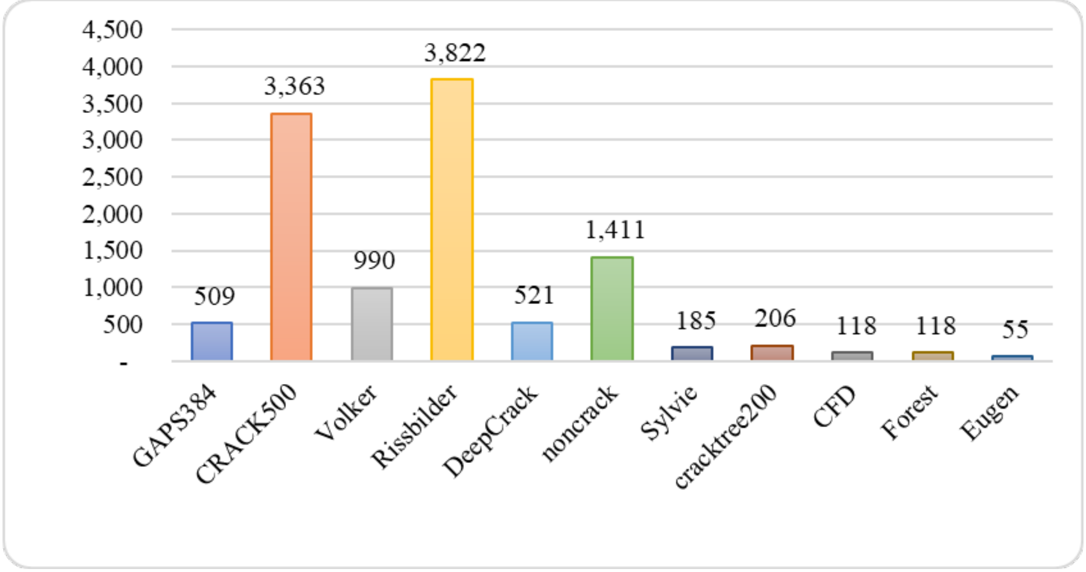


Figure 20. External Data Sources for Crack Segmentation

To ensure robust model evaluation and generalizability, these datasets were divided into training and testing subsets with a 70:30 ratio, resulting in a well-balanced training set for learning key features and a reliable testing set for performance validation.

Given the variability and complexity of road crack patterns, a series of preprocessing and augmentation techniques were applied to prepare the datasets for deep learning workflows:

- **Random Shift**: Introduced minor adjustments to account for positional variations during data collection.
- **Random Rotation**: Applied angular rotations to enhance the model invariance to crack orientation.
- **Random Clipping**: Simulated partial occlusions or incomplete observations to improve the model's robustness.
- **Normalization**: Standardized pixel intensity values across images to ensure consistent input data.
- **Resizing**: Scaled images to a uniform size of 512x512 pixels, balancing memory efficiency with the preservation of critical crack details.

These preprocessing steps ensure that the dataset is diverse, representative, and optimized for deep learning models such as U-Net and DeepSegmentor.

(1) U-Net

U-Net is a widely used encoder-decoder architecture initially designed for biomedical image segmentation, and its flexibility makes it suitable for road crack analysis. Its skip connections help preserve spatial details, making it effective for segmenting thin and irregular cracks.

- **Model Design:** The U-Net architecture is a widely recognized model for image segmentation, characterized by its encoder-decoder structure. The encoder extracts high-level features through a series of convolutional layers and pooling operations, progressively reducing spatial dimensions while capturing critical semantic information. The decoder then reconstructs the spatial resolution by upsampling the encoded features and utilizing skip connections to combine high-resolution spatial information from the encoder with semantic features from deeper layers.

  For road crack segmentation, a customized U-Net architecture tailored to the dataset and requirements was implemented. While the overall structure remains similar to the original U-Net, various optimizations were introduced to enhance its performance on high-resolution road imagery. These adjustments improve the model's ability to detect fine crack details and irregular geometries. The detailed architecture of the U-Net implementation in this research project is illustrated in Figure 21.

- **Training Process:** The resized, normalized and augmented images were fed into the U-Net model for crack segmentation.
- **Training Hyperparameters:** The specific training hyperparameters used in this research project are included in Table 7.

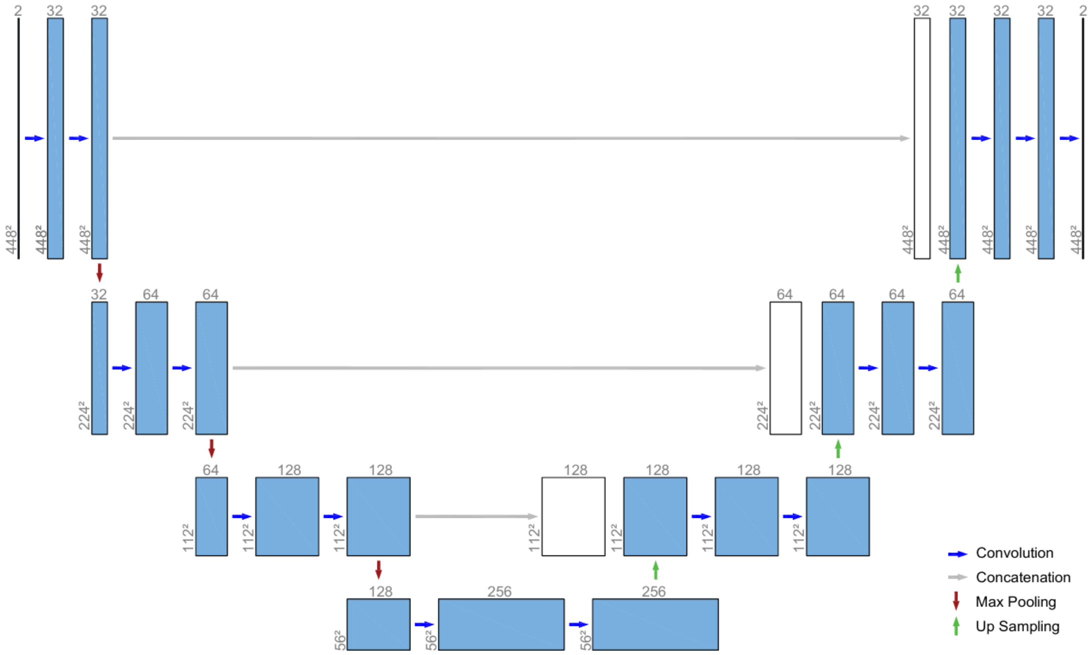


Figure 21. Network Structure of U-Net

Table 7. Hyperparameter for the U-Net Model

Hyperparameter	Value	Description
Epochs	200	Number of complete passes over the training dataset
Batch Size	16	Number of samples processed in one forward/backward pass
Initial Learning Rate (LR)	0.001	Learning rate at the start of training (epoch 0)
Optimizer	Adam/SGD	Optimization algorithm used for training
Amsgrad	Ture	Enables the Amsgrad optimizer for improved stability
Weight Decay	1.00E-05	Regularization parameter to prevent overfitting by penalizing large weights
Validation Dataset Split	0.2	Proportion of the dataset reserved for validation
Validation Epochs Every n	50	Validation is conducted every n training epochs

For the crack segmentation task, the Dice Coefficient was used to evaluate the performance of the developed models. Similar to the Intersection over Union (IoU), the Dice Coefficient places more weight to the overlap, making it particularly suitable for evaluating tasks such as road crack segmentation.

It is important to note that the developed model was evaluated using external datasets, as its prediction quality exceeded that of the ground truth annotations.

Analysis Results

The Dice Coefficient achieves 97% on the testing dataset, indicating the U-Net model is highly effective in segmenting all three types of cracks: alligator, longitudinal, and transverse.

(2) DeepSegmentor

DeepSegmentor is a specialized segmentation model designed for infrastructure applications. It excels at handling complex and noisy datasets, making it highly effective for accurately segmenting cracks with irregular patterns.

- **Model Design:** The DeepSegmentor model is specifically designed for segmentation tasks, incorporating advanced attention mechanisms that enhance its focus on crack regions while effectively reducing background noise. These attention mechanisms allow the model to prioritize relevant features, making it particularly effective at handling complex and noisy road crack imagery.

  In this research project, the default DeepSegmentor settings were utilized for both training and inference without any modifications. This ensures that the model's architecture and parameters remain consistent with its proven design, optimized for high-resolution road crack segmentation tasks. The architecture of a DeepSegmentor network is illustrated in Figure 22.

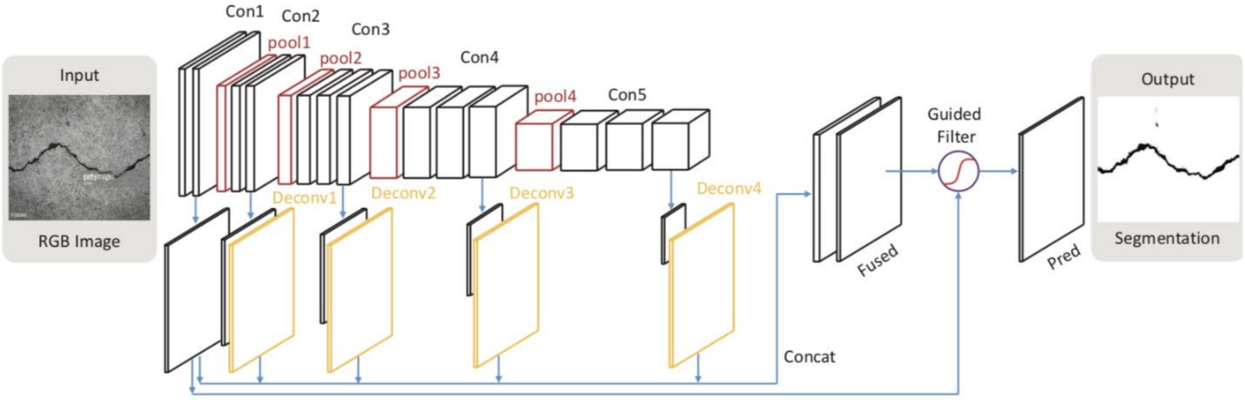


Figure 22. Network Structure of DeepSegmentor [42]

- **Training Process**: The resized, normalized and augmented images were fed into the DeepSegmentor model for crack segmentation.
- **Training Hyperparameter:** The specific training hyperparameters used in this research project are included in Table 8.

Table 8. Hyperparameter for the DeepSegmentor Model

Hyperparameter	**Value**	**Description**
Epochs	1000	Number of complete passes over the training dataset
Batch Size	1	Number of samples processed in one forward/backward pass
Initial Learning Rate (LR)	0.0001	Learning rate at the start of training (epoch 0)
Optimizer	Adam/SGD	Optimization algorithm used for training
Amsgrad	Ture	Enables the Amsgrad optimizer for improved stability
Weight Decay	1.00E-05	Regularization parameter to prevent overfitting by penalizing large weights
Validation Dataset Split	0.2	Proportion of the dataset reserved for validation
Validation Epochs Every n	100	Validation is conducted every n training epochs

Again, the Dice Coefficient was used to evaluate the performance of the developed models. Additionally, it is important to note that the developed model was evaluated using external datasets, as its prediction quality exceeded that of the ground truth annotations.

Analysis Results

The Dice Coefficient achieves 97% on the testing dataset, again, indicating the DeepSegmentor model is also highly effective in segmenting all three types of cracks: alligator, longitudinal, and transverse.

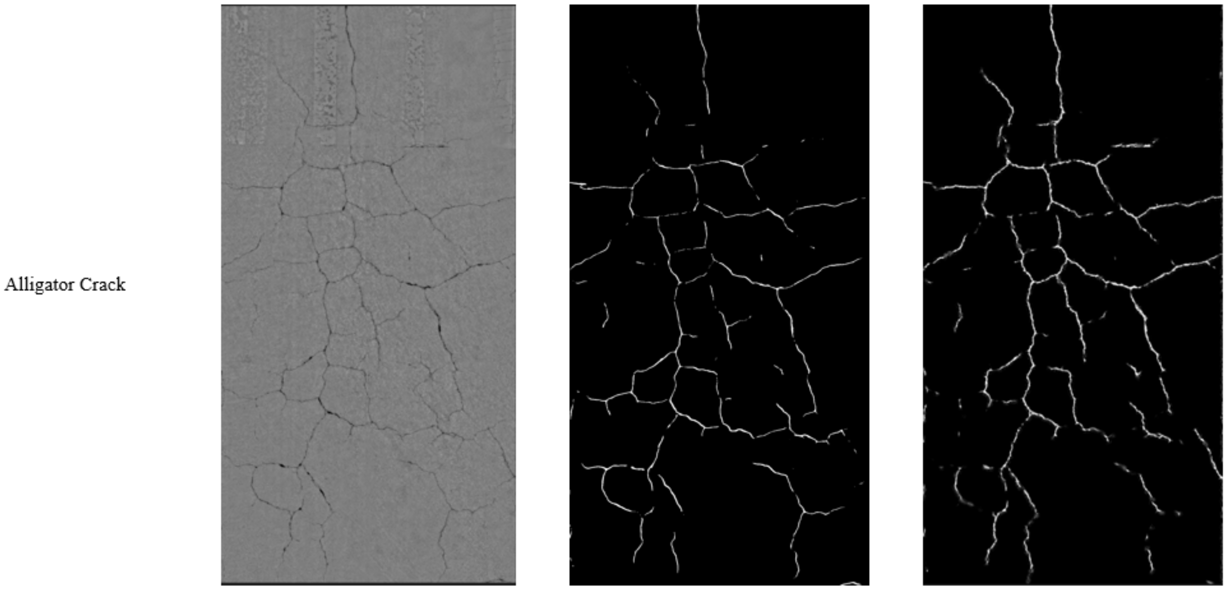
Comparison

For this crack segmentation task, the U-Net model required less training time and performed well on relatively simple crack patterns, but its ability to segment overlapping or faint cracks was limited. In contrast, the DeepSegmentor model, with its attention mechanisms, achieved higher segmentation accuracy on complex and noisy images, but it required more computational resources.

Visual Analysis

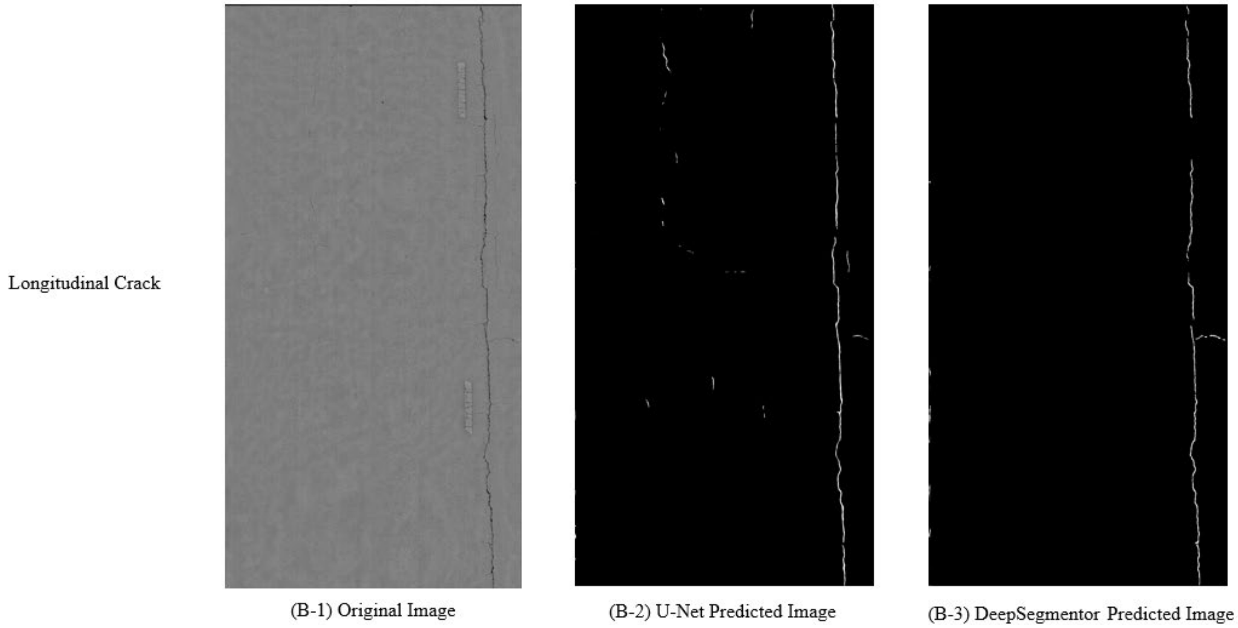
A visual analysis was conducted on the segmentation results for three types of cracks within images provided by NCDOT, and the results show that both deep learning models have achieved highly satisfactory performance (Figure 23, Figure 24, Figure 25).

Pavement markings are clearly visible in both Figure 23 and Figure 24. The developed U-Net and DeepSegmentor models effectively excluded these markings, providing precise predictions of only the cracks.



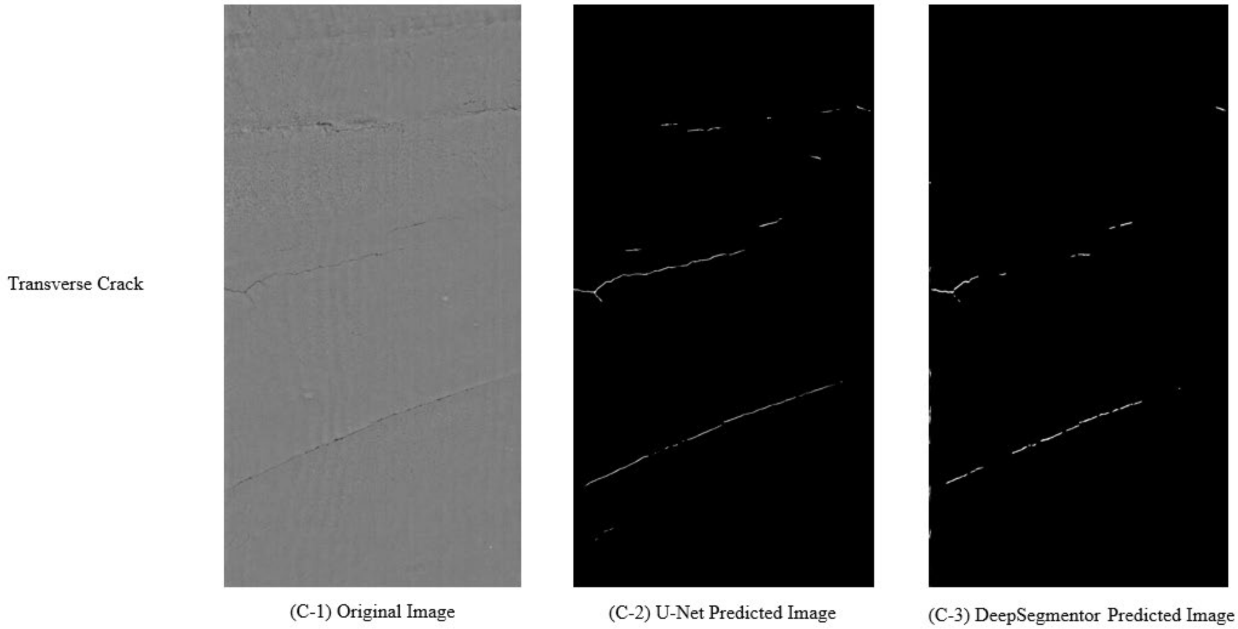
(A-1) Original Image (A-2) U-Net Predicted Image (A-3) DeepSegmentor Predicted Image

Figure 23. Visual Analysis - Alligator Crack Segmentation



(B-1) Original Image (B-2) U-Net Predicted Image (B-3) DeepSegmentor Predicted Image

Figure 24. Visual Analysis - Longitudinal Crack Segmentation



(C-1) Original Image (C-2) U-Net Predicted Image (C-3) DeepSegmentor Predicted Image

Figure 25. Visual Analysis - Transverse Crack Segmentation

4.1.3 Crack Quantification

In this task, the size of cracks, i.e., length, width, and area, was quantified. To achieve this goal, type-specific analysis tailored to the unique characteristics of each crack type was performed. The quantification process utilized segmentation results and bounding boxes to ensure consistent and accurate measurements. The actual dimension of a manhole was used to calibrate the quantification results.

- **Crack Length for Longitudinal Cracks and Transverse Cracks**: After segmentation, all pixels corresponding to longitudinal cracks and transverse cracks were identified. A series of small bounding boxes were generated along the crack's path, and the total crack length was calculated by summing the lengths of all detected segments. The pixel-based length was converted to real-world measurement using a scaling factor derived from the known dimensions of a manhole visible in the images.
- **Crack Width for Longitudinal Cracks and Transverse Cracks**: For longitudinal cracks and transverse cracks, bounding boxes were created perpendicular to the crack's path, with a step size of 10 pixels. The segmentation results within these bounding boxes were analyzed to calculate the crack's minimum and maximum width. Similar to the crack length, the pixel-based width was converted to a real-world measurement using the same scaling factor.
- **Crack Area for Alligator Cracks**: Bounding boxes were generated around the segmented regions of alligator cracks to capture their overall shape. The maximum x and y coordinates of the segmented pixels were used to define the boundaries. The area of the alligator cracks was calculated based on these maximum coordinates, and the pixel-based area was converted to a real-world measurement using the same scaling factor.

The use of pixel-level segmentation with real-world calibration ensures precise measurements of crack length, width, and area. These metrics offer valuable insights into the severity and extent of each type of crack, supporting effective road maintenance planning. Screenshots of animations of the crack quantification process are shown in Figure 26.

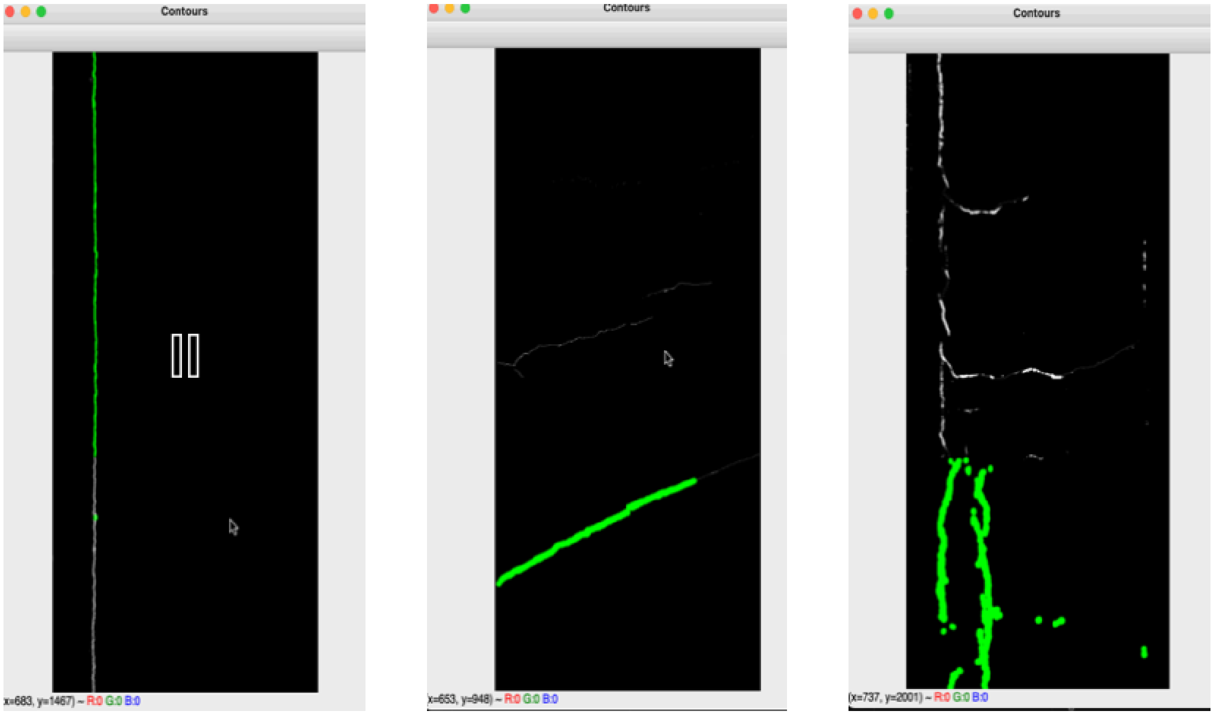

Contours
x=683, y=1467) ~ R:0 G:0 B:0
Contours
x=653, y=948) ~ R:0 G:0 B:0
Contours
(x=737, y=2001) ~ R:0 G:0 B:0

Figure 26. Screenshots of Animations of Crack Quantification on Three Types of Cracks

4.2 Analysis of GoPro Images

4.2.1 Crack Classification

For crack classification, the same deep learning models as described in Section 4.1.1, ResNet and Vision Transformer, were utilized. The rule-based method was not used due to the significantly larger size of the processed GoPro images, which made it unsuitable for the fixed geometric heuristics employed in that approach.

GoPro images, captured in high resolution and large scales, present unique challenges and opportunities for road crack classification. Due to the worse condition of the selected roadway section, a substantial portion of the cracks detected in these images were classified as alligator cracks. This imbalance highlighted the need for additional preprocessing steps to optimize the classification process.

To address this, the GoPro images were split into smaller patches, ensuring manageable input sizes for the deep-learning models. Each patch was then classified into one of three categories: Longitudinal, Transverse, or Alligator cracks. This patch-based approach not only reduced computational complexity but also enhanced the models' ability to focus on localized crack patterns, improving the overall classification accuracy.

Data Source

The dataset described in Section 3.2.3.2 was utilized, which comprises high-resolution GoPro images collected from diverse road conditions. To prepare the data for classification, the images were divided into smaller patches to manage their large size and high variability effectively. Each patch was then manually labeled into one of three classes, Longitudinal, Transverse, or Alligator cracks, to train the deep learning models. This manual labeling process ensured accurate and reliable ground truth annotations, forming the foundation for robust model training and evaluation.

(1) Dividing GoPro Images into small Patches

The procedure for dividing GoPro images into smaller patches is detailed below, outlining the steps taken to preprocess the large-scale images for classification. The process ensures that the resulting patches are optimized for training while removing irrelevant sections and noise.

Step 1: **Initial Splitting of Large GoPro Images.** The original GoPro image, obtained after applying photogrammetry to a GoPro video collected by the ITRE research team, with a total size of approximately 40GB, were divided into smaller patches with dimensions of 1488 × 744 pixels using a custom Python script (Figure 27). This step reduced the computational burden and prepared the images for further segmentation.

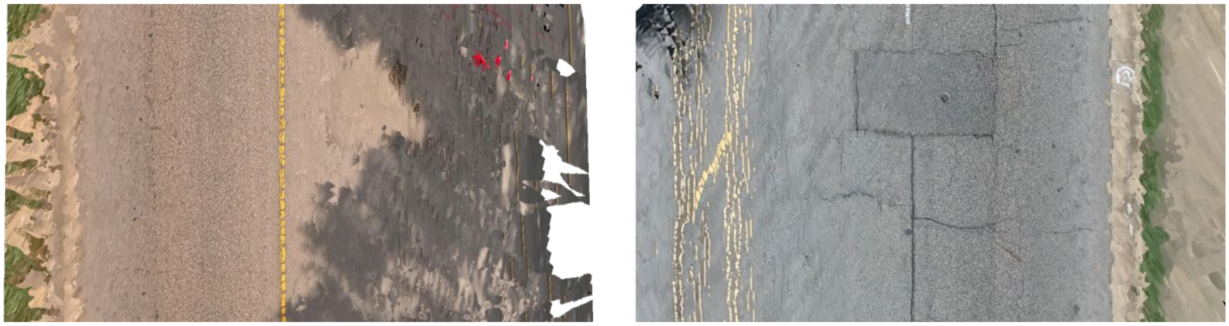


Figure 27. Initial Splitting of GoPro Images

Step 2: **Further Splitting into Smaller Patches.** Each image patch was further subdivided into smaller patches to ensure uniformity and compatibility with the deep-learning models (Figure 28). This step increased the granularity of the data, allowing the models to focus on localized crack features.

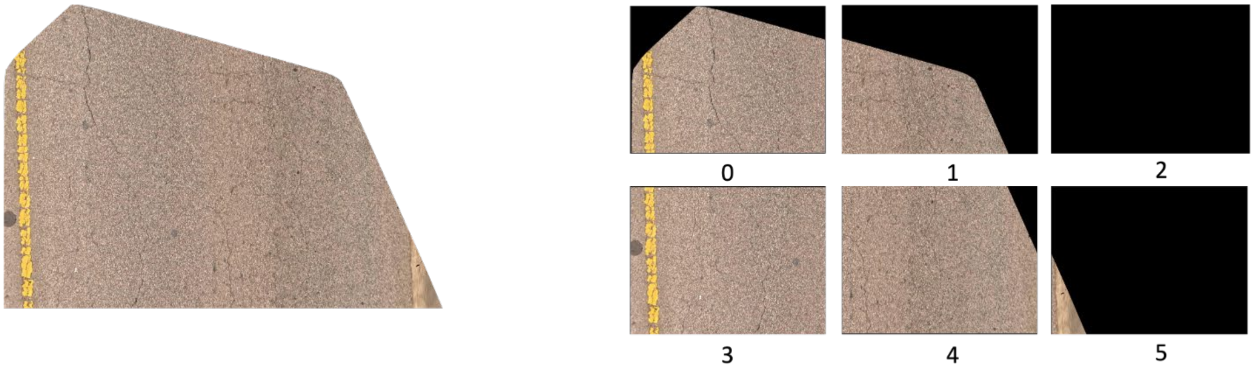


Figure 28. Further Splitting of GoPro Images

Step 3: **Filtering Unnecessary Patches.** To ensure only patches containing useful road sections were retained for further processing, noisy and irrelevant GoPro image patches were removed. These include areas with sidewalk trees, vehicles, and other objects that were not part of the road surface. A sample noisy patch is shown in Figure 29.

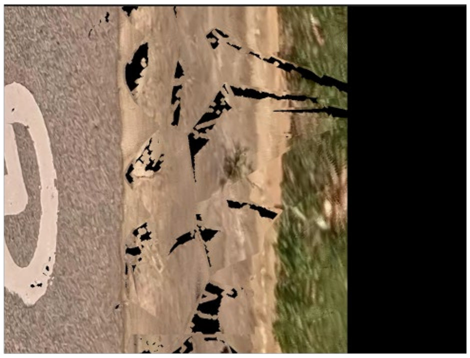


Figure 29. Sample Noise Patch

By following these steps, a high-quality dataset was prepared from a large, processed GoPro image, ensuring the final data was well-suited for classification tasks while minimizing noise and irrelevant content.

(2) Transfer into new dataset

To enhance model performance, crack regions were manually extracted from the GoPro images to create a new dataset. This dataset was then used to train two deep-learning models. Using transfer learning, the pre-trained weights from these models in section 4.1.1 were utilized and fine-tuned using the newly created dataset. This approach allowed researchers to efficiently adapt the models to the specific characteristics of road cracks captured in GoPro images, improving their accuracy and generalization.

Analysis Results

The performance of the classification models, ResNet and Vision Transformer, was evaluated using the created test dataset. As described in earlier sections, the test dataset was derived from the prepared dataset, ensuring a balanced distribution of crack types for fair evaluation.

The results of the evaluation for the two models, including standard metrics such as accuracy, precision, recall, and F1-score, are summarized in Table 9 below. It should be noted that due to resource constraints, only a limited number of images were annotated and used to train the models. This limitation could be the reason that both models, ResNet and ViT, achieved similar performance scores.

Table 9. The Performance of Classification Models

Network	**Pre-trained Model**	**Optimizer**	**SGD**	**Decay**	**Accuracy**	**F1**	**Recall**
ResNet	ImageNet	Adam	FALSE	1.00E-05	0.85	0.92	0.86
ViT	ImageNet	Adam	FALSE	1.00E-05	0.86	0.92	0.86

4.2.2 Crack Segmentation

4.2.2.1 Crack Segmentation using U-Net and DeepSegmentor

Segmenting road cracks in GoPro images introduces unique challenges due to their high resolution, wide-angle perspectives, and the presence of unrelated objects such as sidewalks, trees, and vehicles. To address these challenges, the Segment-Anything model [43] was used to preprocess the images by identifying and removing unrelated objects, ensuring the segmentation process focuses solely on the road surface and cracks.

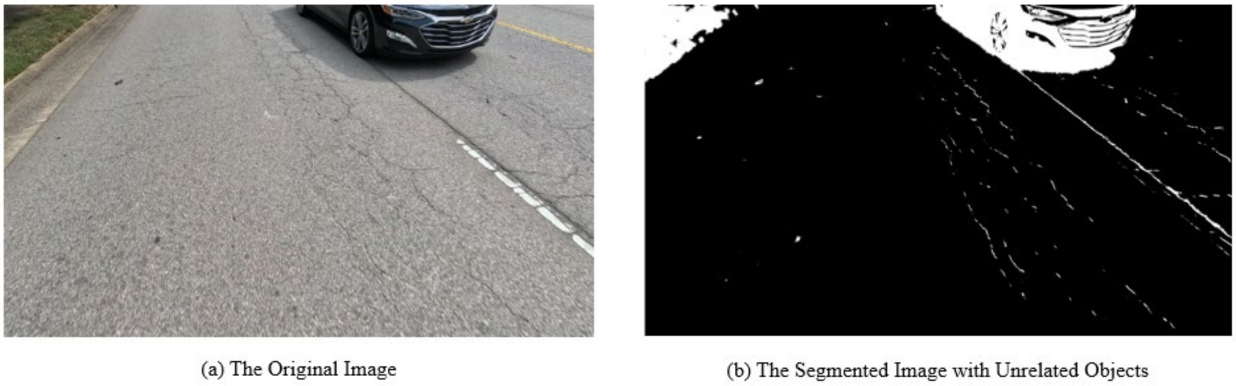
Following the removal of irrelevant elements, the same pre-trained networks, U-Net and DeepSegmentor, and weights from Section 4.1.2 were used to segment road cracks. These models were used without additional fine-tuning, utilizing their pre-trained capabilities to effectively segment cracks in the refined GoPro image patches.

Challenges Encountered during Segmentation

Applying segmentation models directly to high-resolution GoPro images introduces specific challenges, particularly in identifying and segmenting unnecessary objects on the road. GoPro images often capture not only road surfaces but also non-relevant elements such as sidewalks, trees, vehicles, and shadows due to their wide-angle perspectives and dynamic field of view (Figure 30). These elements can interfere with the segmentation process by introducing noise and distracting the models from focusing on road cracks.

The primary challenge lies in adapting segmentation models, originally designed for crack detection, to effectively contour and filter out these unrelated objects while preserving the road surface. High-resolution images further compound the difficulty by increasing computational demands, making preprocessing and object removal both time-sensitive and resource-intensive.

As described in a later section, Segment-Anything was employed to remove unrelated objects. By isolating and focusing on road surfaces, the segmentation process is streamlined, enabling subsequent crack segmentation tasks to focus solely on meaningful features.


(a) The Original Image

(b) The Segmented Image with Unrelated Objects

Figure 30. Challenges Encountered during Segmentation

Segment-Anything

Segment-Anything is a state-of-the-art segmentation framework designed to provide zero-shot segmentation capabilities across diverse image domains and tasks. Developed with a focus on generalization, the model eliminates the need for task-specific training, making it a versatile tool for segmenting arbitrary objects in various contexts. It has quickly become a foundational model in segmentation tasks due to its ability to handle complex images and scenarios with minimal preprocessing.

In the context of road analysis, Segment-Anything is particularly valuable for preprocessing tasks, such as identifying and removing non-road elements (e.g., vehicles, sidewalks, trees, and shadows). This preprocessing ensures that subsequent models can focus solely on the road surface, enhancing the accuracy and efficiency of segmentation tasks like crack detection.

Segment-Anything employs a highly adaptable architecture designed to handle diverse segmentation tasks. At its core, the model leverages a prompt-based segmentation approach, where user-defined inputs, such as points, bounding boxes, or text descriptions, guide the model to identify and segment objects of interest within an image. Once these prompts are provided, the mask decoder processes them to generate precise segmentation masks that accurately align with the desired objects (Figure 31).

The foundation of the model is a pre-trained Vision Transformer (ViT), which serves as its backbone for robust feature extraction. This backbone enables the model to process complex and diverse image inputs, making it exceptionally effective across varied domains. Additionally, the model's zero-shot capability, developed through training on a massive and diverse dataset, allows it to perform segmentation tasks in new domains without requiring any fine-tuning or retraining. This makes Segment-Anything a versatile tool for a wide range of applications.

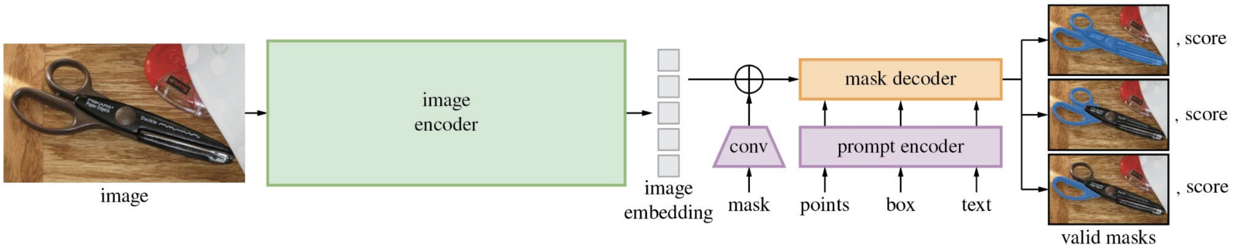


Figure 31. Segment-Anything Workflow

Segment-Anything offers several distinct advantages that make it an ideal choice for preprocessing road images:

1. Generalization: Its zero-shot segmentation ability allows it to handle a wide variety of objects and scenes without requiring domain-specific training.
2. Ease of Use: The prompt-based approach enables flexible and intuitive segmentation, reducing the complexity of manual preprocessing tasks.
3. High-Quality Masks: The model consistently produces accurate segmentation masks, even for complex objects and cluttered scenes, making it suitable for removing unrelated elements from road images.
4. Scalability: Its pre-trained nature and ability to process diverse image types make it highly scalable for large datasets, such as GoPro images with varying resolutions and content.

In this research project, Segment-Anything is integrated into the preprocessing pipeline for GoPro images. By isolating road surfaces and filtering out non-relevant elements, it significantly enhances the performance of downstream segmentation models, such as U-Net and DeepSegmentor, in detecting road cracks.

Workflow of GoPro Image Segmentation

The segmentation of cracks in GoPro images involves a systematic pipeline that integrates object removal and crack segmentation, ensuring that only relevant road surface areas are processed. The workflow is illustrated in Figure 32.

Step 1. Input GoPro Image: The process begins with high-resolution GoPro images that capture both the road surface and surrounding objects, such as vehicles, trees, and sidewalks.

Step 2. Object Masking: Using the generated masks, the non-relevant objects are removed from the image, isolating the road surface for further processing.

Step 3. Road Patches: The isolated road surface is split into smaller patches to make the data manageable for deep learning models and to focus on localized features.

Step 4. Crack Segmentation: Each patch is fed into pre-trained segmentation models, U-Net and DeepSegmentor, which produce binary masks highlighting crack regions in the road surface. Segmentation models are also applied to identify and generate masks for non-relevant objects in the image, such as vehicles, sidewalk areas, and vegetation as well.

Step 5. Segmentation Integration: The segmentation outputs from all patches are fused to reconstruct the full segmented road surface, ensuring consistency across the image.

Step 6. Output Segmentation Results: The final output is a fully segmented image of the road surface with cracks distinctly highlighted, ready for further analysis such as crack classification, measurement, or maintenance planning.

This workflow ensures the effective segmentation of cracks in GoPro images by first removing unrelated objects and then accurately identifying crack patterns. By leveraging pre-trained models and a patch-based approach, the process is both efficient and scalable for large datasets.

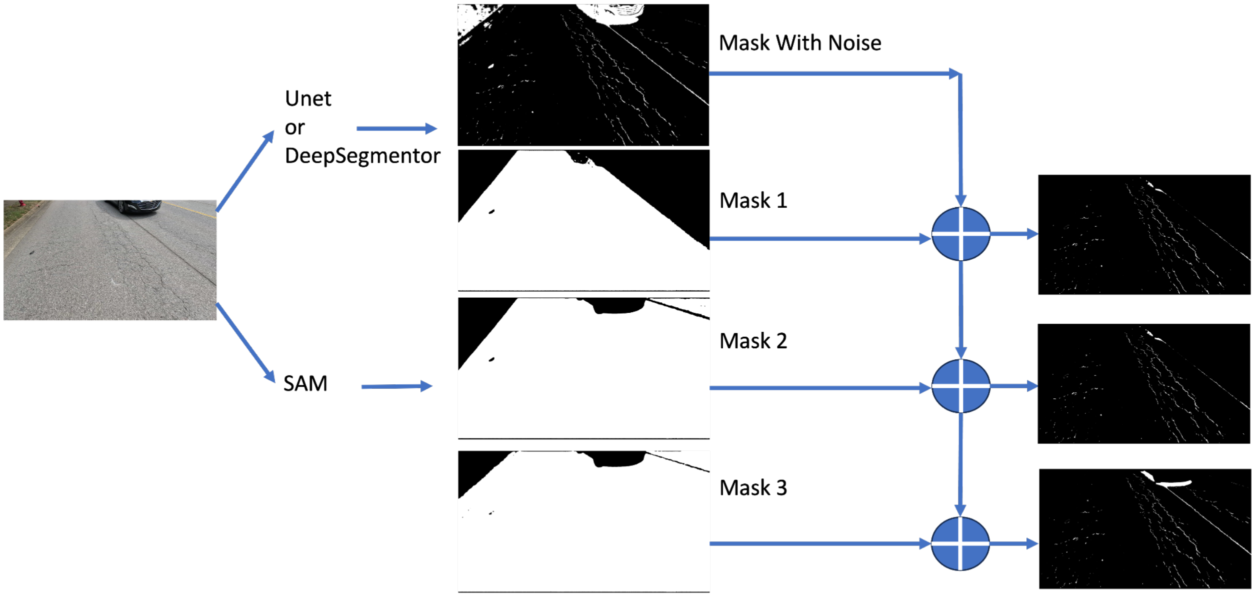


Figure 32. Workflow of GoPro Image Segmentation

4.2.2.2 Image Segmentation using Mask R-CNN

This section describes the use of another deep learning model, Mask R-CNN, for pavement crack segmentation. Mask R-CNN goes beyond simple object detection by performing instance segmentation. This means it can not only identify the presence of cracks in pavement images but also precisely delineate the shape and location of each crack. This capability is crucial for pavement assessment as it provides detailed information about the extent and severity of damage.

Dataset

This work used a dataset available online which combined and standardized 12 smaller datasets. The dataset description is available at https://github.com/khanhha/crack_segmentation. Some images in the dataset do not apply to the Mask R-CNN use case. Therefore, they are excluded from the training process. Specifically, 9,844 images out of 11,298 images are available for this research project. Those excluded are due to containing no cracking and for data quality issues such as lack of annotations. This work did not use data augmentation techniques to increase the dataset size, but it is suggested for future research to further improve training performance. The dataset was randomly split into two sets, 80% for training and 20% for validation.

Model Architecture

This research project used the Mask R-CNN implementation offered by PyTorch. This implementation has been shown to perform well on image segmentation tasks and has been pre-trained by the PyTorch team. Therefore, transfer learning was utilized to improve the training performance on pavement cracking based on the pre-trained weights.

Due to the nature of deep learning models, several hyperparameters can be fine-tuned to improve model performance. Hyperparameters used to train Mask R-CNN are included in Table 10.

Table 10. Hyperparameter for the Mask R-CNN Model

Hyperparameter	**Value**	**Description**
Epochs	300	Number of complete passes over the training dataset
Batch Size	16	Number of samples processed in one forward/backward pass
Initial Learning Rate (LR)	0.005	Learning rate at the start of training (epoch 0)
Momentum	0.75	Optimizer denoising parameter
Weight Decay	0.0005	Regularization parameter to prevent overfitting by penalizing large weights
LR Step Size	100	Learning rate is multiplied by LR Gamma at this epoch
LR Gamma	0.1	Multiply current LR by this parameter
Hidden Layer Size	256	Number of Mask R-CNN hidden layer neurons
Backbone Trainable Layers	5	Allow training on this many layers (max: 5)
Validation Dataset Split	0.2	Proportion of the dataset reserved for validation
Validation Epochs Every n	2	Validation is conducted every n training epochs
Data Loader Workers	2	Number of CPU threads for loading data to GPU
Start Epoch	0	Starting epoch for resuming training

Training Process

Training the Mask R-CNN model begins with the creation of a custom dataloader to parse the dataset. This dataloader is based on PyTorch examples and best practices. It supports dataset splitting (for training and validation), shuffling for randomized training, multiple workers for increased throughput, and exclusion of unwanted data to improve training.

Next, the Mask R-CNN model was defined. This customized model uses the *maskrcnn_resnet50_fpn* model as a base, with a *FastRCNNPredictor* predictor for Region of Interest (RoI) boxes and *MaskRCNNPredictor* predictor for RoI masks. These changes are made to support the new class and label (pavement cracking) rather than the default Mask R-CNN object labels such as horse, car, and person. The RoI masks predictor was configured to use a hidden layer of size 256 neurons.

The Mask R-CNN model was combined with a Stochastic Gradient Descent (SGD) optimizer and a Stepped Learning Rate (StepLR) scheduler. Their configuration options are listed in the table above. SGD and StepLR are common straightforward functions in deep learning and were chosen for their ease of use and validity for this task.

Analysis Results

Table 11 presents the quantitative results of the trained model. A selection of performance metrics is across the table from left to right. The metrics have descriptive statistics as rows from top to bottom.

The F1 score has a mean of 0.52 across all 9,844 images. Following is a similar Dice Score of 0.52. These results suggest that the model performs only very slightly above random (0.5). Additionally, the significant level of standard deviation (0.31) is cause for concern. With such a high amount of variation between results, it can be concluded that the Mask R-CNN model is not well-trained for the task of pavement crack segmentation. This conclusion is supported by the precision and recall metrics. A precision of 0.79 shows the model is fair at avoiding false positives, but a recall of 0.49 shows it is also unable to detect true positives. The basic metrics of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), are also included in the table. These final four metrics are used to compute the previous metrics.

The model has a mean bounding box Intersection over Union (IoU) value of 0.93. This value applies to the bounding boxes of segmentation predictions. An IoU value may range from 0.0 to 1.0. The result of 0.93 is very good but has limited practical use due to the low F1score of masks. From these values, it can be concluded that the Mask R-CNN model is very good at predicting the general area (bounding box) but not the specific pixels (mask) of a crack in the pavement.

Table 11. Mask R-CNN Validation Results

	F1 Score	Dice Score	Precision	Recall	TP	FP	TN	FN
Count	9,844	9,844	9,318	9,318	9,844	9,844	9,844	9,844
Mean	0.52	0.52	0.79	0.49	4,211	732	191,917	3,844
St. Deviation	0.31	0.31	0.15	0.30	6,129	1,002	7,774	4,122
Minimum	0.00	0.00	0.00	0.00	0	0	113,851	7
25th Quartile	0.24	0.24	0.71	0.19	738	180	190,374	1,260
50th Quartile	0.60	0.60	0.80	0.54	2,050	517	194,261	2,373
75th Quartile	0.79	0.79	0.89	0.76	4,997	937	196,427	5,167
Maximum	0.99	0.99	1.00	0.99	81,976	38,128	200,655	75,440

Qualitative Analysis

While a quantitative analysis is useful for comparing model performance against competing methods, a qualitative analysis is also included here to demonstrate the fair performance of the model. The series of figures below show a comparison between the annotated crack (ground truth) and the inference result (prediction). The bounding boxes are included in their respective lighter colors. Additionally, the predictions are provided with confidence values from 0.0 to 1.0. These confidences are per-pixel, providing a gradient of least-confident (transparent) to most-confident (opaque red). In Figure 33 below, it was observed that the model understands the general shape but not the finer details of the cracking.

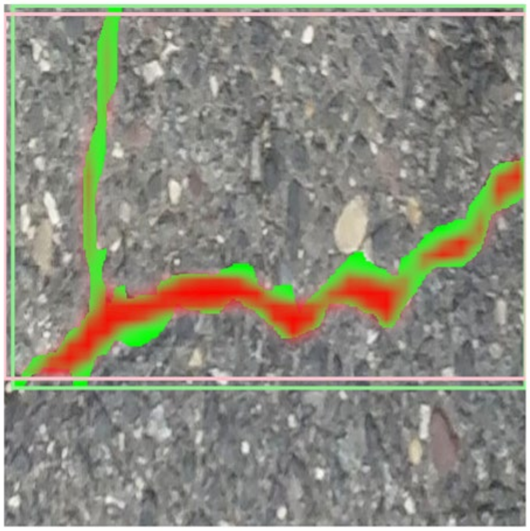


Figure 33. The Annotated Crack (Ground Truth) and The Inference Result (I)

Figure 34 supports the previous conclusion, with red predictions generally following the shape and most confident in the center of the cracking. While these two images appear to be of different spatial scales (the former being taken closer to the road surface), their predictions follow a common pattern.

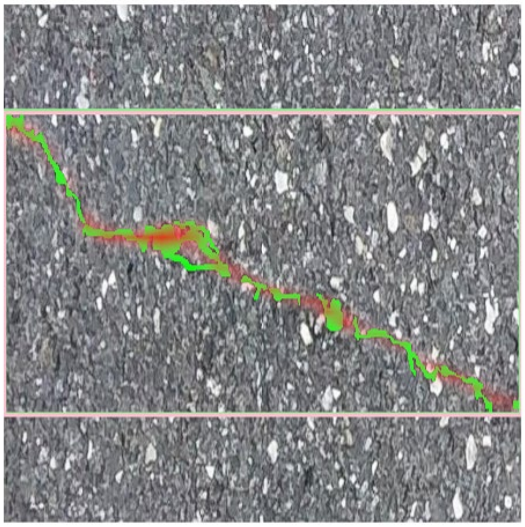


Figure 34. The Annotated Crack (Ground Truth) and The Inference Result (II)

In Figure 35, the model failed to predict a large portion of the cracking in the middle of the image. Interestingly, it over-predicted the width of the very thin cracking areas at the top and bottom of the image. This third image appears to be the furthest from the road surface, which may explain the thinness of the cracking pictured.

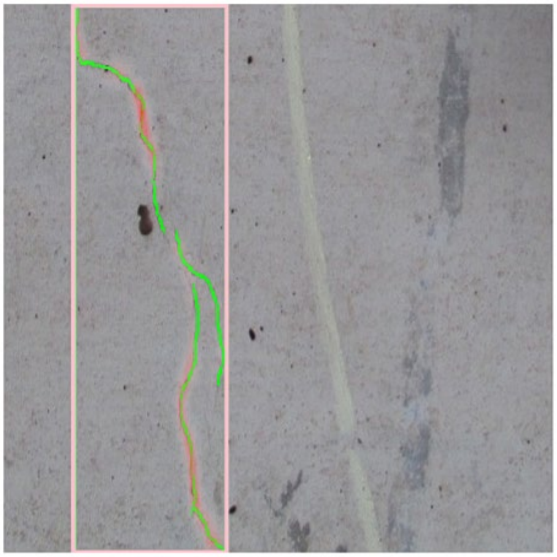


Figure 35. The Annotated Crack (Ground Truth) and The Inference Result (III)

Demonstration

A real-world demonstration of the trained Mask R-CNN model is provided. Photogrammetry was used to create a georeferenced orthomosaics of the road surface of a local road in Raleigh, North Carolina. The location of this road is shown in Figure 36 below. Subfigure A1 shows the State of North Carolina in black outline. Subfigure A2 shows Wake County, North Carolina in purple outline. Subfigure A3 shows the entire orthomosaics. Subfigure A4 shows a subset of the orthomosaics in full color (RGB). The Mask R-CNN prediction (inference) is overlaid with the orthomosaics using red color. Subfigure A4 demonstrates the Mask R-CNN model can find some cracking with imperfect results for the thinnest cracks present.

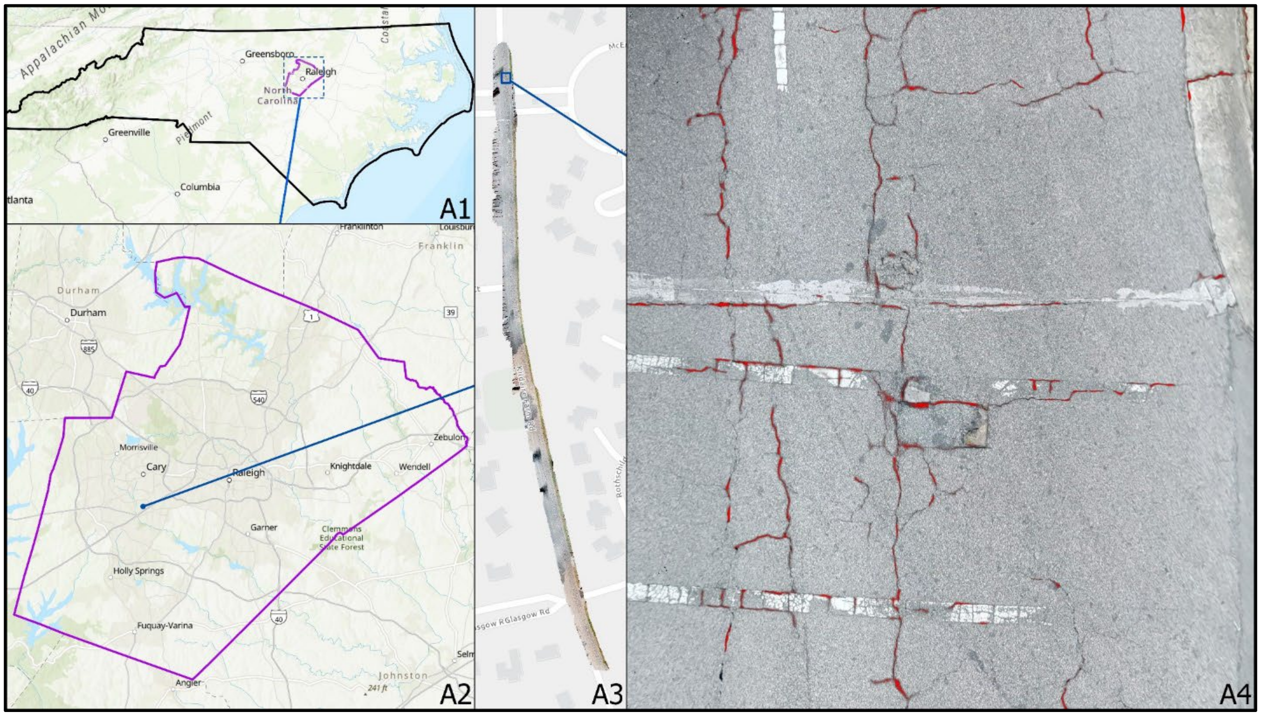


Figure 36. A Real-World Demonstration of Photogrammetry

Three additional figures (Figure 37, Figure 38, Figure 39) illustrate the performance of the Mask R-CNN model. In each figure, subfigures are related to their spatial location along the given orthomosaic using blue leader lines.

In subfigures B1, B2, and B3 (Figure 37), utility access holes are highlighted. It was observed that the Mask R-CNN model does a good job of ignoring the manhole covers, with few exceptions. Poor performance for segmenting alligator cracking can be observed in subfigure B1.

In subfigures A4 (Figure 36), B3 (Figure 37), C1, C2 (Figure 38), and D1 (Figure 39), it can be observed that cracks are in and around road makings. Some of these cracks are present in the pavement, but others appear to be of the marking paint itself. The Mask R-CNN model can find these cracks but cannot properly distinguish between them. This problem could be addressed through the implementation of additional categories in training such as road markings.

In subfigure C3 (Figure 38), a piece of debris is included in the photogrammetry orthomosaics. Its shadow is detected by the model. This is an example of how photogrammetry and the trained Mask R-CNN model cannot ignore unwanted objects during processing. Finally, in Subfigure D2 (Figure 39), some incomplete predictions were observed.

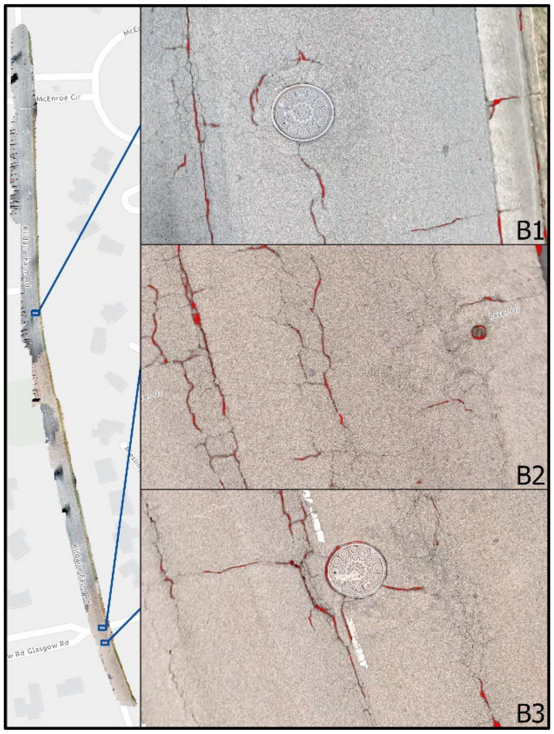


Figure 37. The Performance of the Mask R-CNN Model (I)

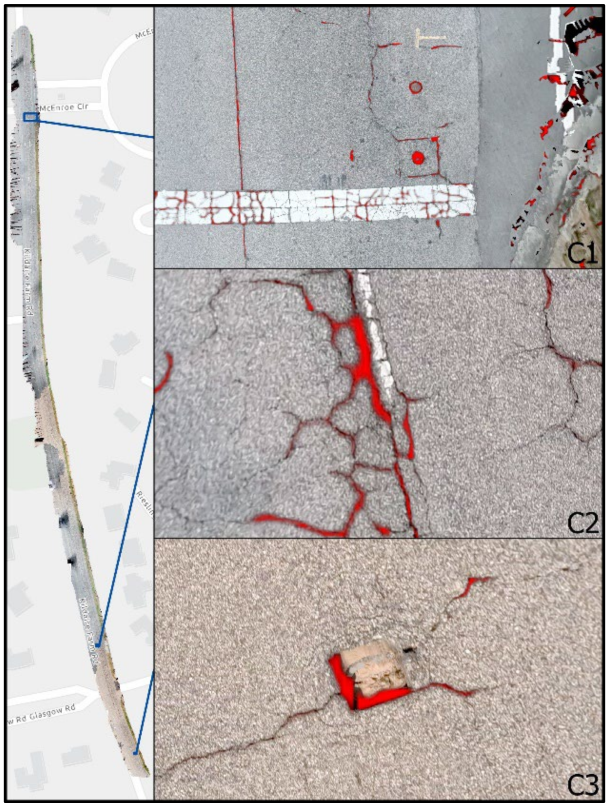


Figure 38. The Performance of the Mask R-CNN Model (II)

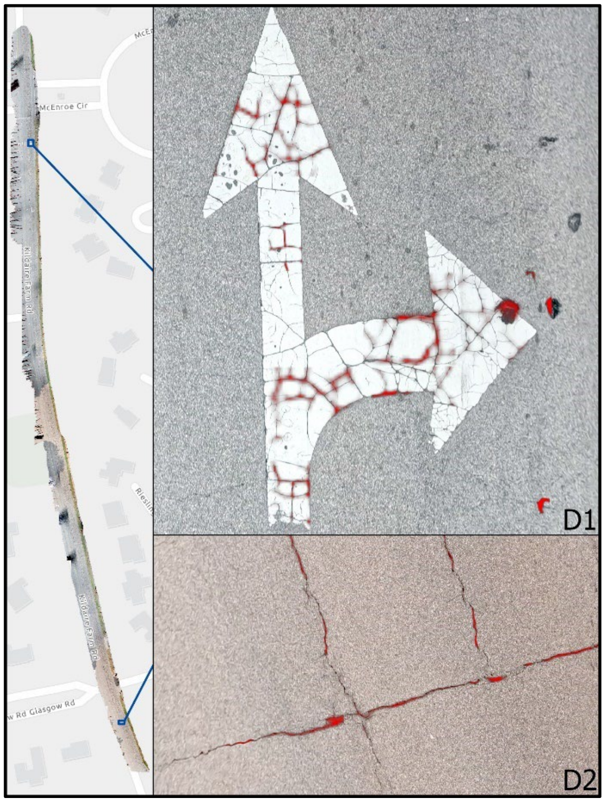


Figure 39. The Performance of the Mask R-CNN Model (III)

Conclusion

While significant progress was made towards automated crack detection, the Mask R-CNN model appears to struggle with the non-descript interior and relatively thin shape of pavement cracks. Researchers believe that there is potential in deep learning algorithms, especially instance segmentation, but Mask R-CNN may not be the ideal algorithm. Mask R-CNN excels at other tasks such as finding vehicles or animals where it can rely on the structure and internal texture of such objects. Pavement cracking has no such internal texture at the resolutions available by photogrammetry from GoPro cameras. The trained model could be improved through various enhancements in the future. First, the use of more training data would improve the variety of cracking detected. Second, data augmentation methods can be used to further increase the variety of training data and thus the robustness of the future model.

Analysis Results of U-Net and DeepSegmentor Models

Since segmentation ground truth is not available for the GoPro images, quantitative performance metrics such as Intersection over Union (IoU) or Dice Coefficient cannot be calculated. However, visual analysis (Figure 40) provides an effective way to assess the performance of the segmentation models.

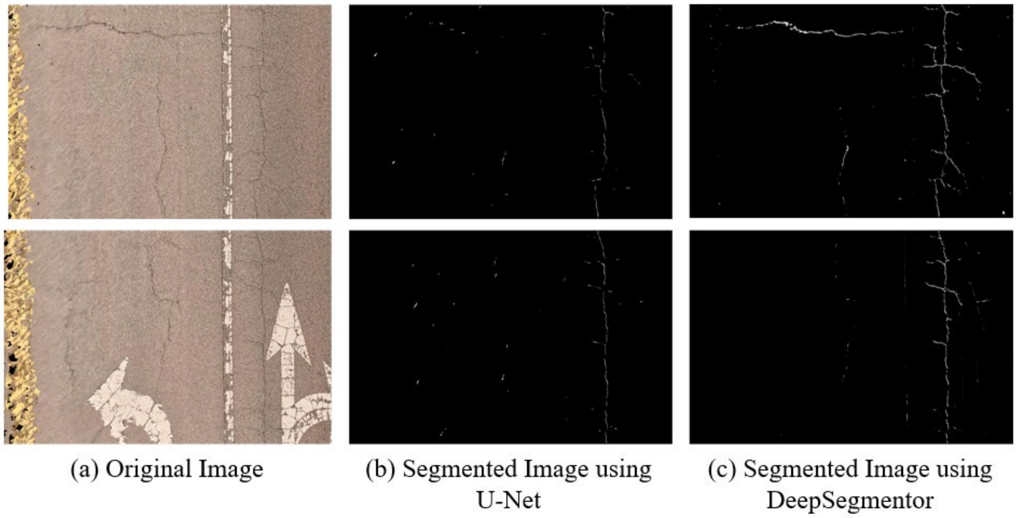

(a) Original Image
(b) Segmented Image using U-Net
(c) Segmented Image using DeepSegmentor

Figure 40. Visual Analysis of the Segmentation Models

The input images, which feature typical road surfaces with visible cracks, provide a challenging test for segmentation models due to their varying crack patterns, including thin longitudinal cracks and intersecting irregular cracks.

The U-Net model demonstrated a reasonable ability to capture the basic structure of visible cracks, particularly thin, straight longitudinal cracks in areas with uniform background textures. However, it faces limitations in identifying finer details in complex or intersecting cracks. In regions where the crack intensity was low or the patterns were irregular, the U-Net model often failed to produce continuous segmentation, resulting in fragmented crack masks.

In contrast, the DeepSegmentor model showed superior performance in capturing both prominent and subtle crack patterns. Its segmentation masks provide more details and can accurately trace the geometry of the cracks, including intersecting and complex patterns. This model excelled in regions with varying textures, effectively identifying cracks even in noisy backgrounds. However, in some cases, DeepSegmentor produced slightly over-segmented outputs, with crack masks appearing thicker than the actual cracks in the input images.

4.2.3 Crack Quantification

For crack quantification, the same process as described in Section 4.1.3 was utilized. Segmentation results and bounding boxes were utilized to quantify the size of cracks, i.e., width, length, and area. The actual dimension of a manhole was used to calibrate the quantification results. The quantification approach is described below:

- **Crack Length for Longitudinal Cracks and Transverse Cracks**: After segmentation, all pixels corresponding to longitudinal cracks and transverse cracks were identified. A series of small bounding boxes were generated along the crack's path, and the total crack length was calculated by summing the lengths of all detected segments. The pixel-based length was converted to real-world measurement using a scaling factor derived from the known dimensions of a manhole visible in the images.
- **Crack Width for Longitudinal Cracks and Transverse Cracks**: For longitudinal cracks and transverse cracks, bounding boxes were created perpendicular to the crack's path, with

a step size of 10 pixels. The segmentation results within these bounding boxes were analyzed to calculate the crack's minimum and maximum width. Similar to the crack length, the pixel-based width was converted to a real-world measurement using the same scaling factor.

- **Crack Area for Alligator Cracks**: Bounding boxes were generated around the segmented regions of alligator cracks to capture their overall shape. The maximum x and y coordinates of the segmented pixels were used to define the boundaries. The area of the alligator cracks was calculated based on these maximum coordinates, and the pixel-based area was converted to a real-world measurement using the same scaling factor.

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This research project was conducted to identify a cost-effective distress data collection method for secondary roadways, and to classify and quantify three types of cracks, longitudinal, transverse, and alligator, that are commonly observed in asphalt pavements in North Carolina. To this end, high-resolution images provided by NCDOT and GoPro videos/images collected by researchers were annotated at a pixel level, segmented using deep learning models, and quantified to obtain their length, width, and area. The following conclusions were drawn from the results of the data analysis process:

- Action cameras, such as GoPro, can be used as a low-cost image collection method for evaluating roadway pavement distresses.
    - To achieve optimal results, the GoPro camera should be mounted to the rear of a vehicle, the vehicle's speed should be less than 20 mph, and pavement images should not be collected during rain, snow, or under other conditions contributing to poor pavement visibility.
    - To evaluate pavement distress, video clips collected by an action camera are more suitable than still images. Video clips offer continuous coverage of the roadway surface, eliminating the need to determine appropriate intervals for still image capture. Additionally, video clips significantly minimize the impact of the double-counting crack issue.
    - To be used for machine learning, distortion correction needs to be conducted on images extracted from the abovementioned video clips. Two distortion correction methods were developed in this research project and the results are satisfactory. The first method involves the use of open-source software titled Hugin. The second method involves the use of leading photogrammetry software, Pix4DMapper.
- Fiji (ImageJ) has proven to be an effective image annotation tool for this research project. To quantify pavement cracks, the original roadway surface images need to be annotated at a pixel level. Low-precision annotation methods, such as bounding boxes, cannot meet the accuracy requirements. The Trainable Weka Segmentation (TWS) plugin within Fiji is capable of high-accuracy annotation and is highly productive. It uses a small number of manual annotations to train a classifier, and this classifier can be saved and reused to segment the remaining image automatically.
- For Primary roadways in North Carolina, either a ResNet or a Vision Transformer model is suitable for classifying cracks as longitudinal, transverse, or alligator. In this research project, top-down high-resolution roadway surface images were analyzed, and either one of these two methods consistently achieves outstanding performance, all with an accuracy of 97%.
- Either a ResNet or a Vision Transformer model is also recommended for classifying cracks in Secondary roadway pavements in North Carolina. In this case, GoPro videos were

recorded, and roadway surface images were extracted and analyzed, either one of these two methods consistently achieves robust performance, all with an accuracy of 85%.

- Either a U-Net or a DeepSegmentor model is recommended for segmenting cracks in Interstate and Primary roadway pavements in North Carolina. The performance of either model is highly satisfactory, with a Dice Coefficient score of 97%.
- A DeepSegmentor model is recommended for segmenting cracks in Secondary roadway pavements in North Carolina. The DeepSegmentor model showed superior performance in capturing both prominent and subtle crack patterns. Its segmentation masks provide more details and can accurately trace the geometry of the cracks, including intersecting and complex patterns. This model excelled in regions with varying textures, effectively identifying cracks even in noisy backgrounds.
- The deep learning models developed in this research project are highly efficient in processing pavement distress images. For two deep learning tasks, image segmentation and image classification, deep learning models were trained on UNC Charlotte's high-performance cluster computers, with training times of 13.4 hours and less than half an hour, respectively. Once deployed to standard office computers, image processing time for high-resolution images is extremely short, averaging 8 milliseconds and 3 centiseconds per image, respectively. In 2021, about 71 thousand images were collected for Division 5, and the total image processing time was less than an hour. Assuming similar data volumes for the other 13 divisions, the combined image processing time for these high-resolution images is around 10 and a half hours. It should also be noticed that the performance is highly satisfactory (Table 12).

Table 12. Efficiency of the Deep Learning Models

Deep Learning Task	**NCDOT Project Research Question**	**Deep Learning Model Training Time**	**Image Processing Time (per image)**	**Image Processing Time for Div. 5 (71,090 images)**	**Estimated Image Processing Time for all 14 Divisions**	**Performance Metric**
Image Segmentation and Evaluation	Is there a crack in the image?	13.4 hours	0.008 seconds	0.2 hours	2.2 hours	Dice Coefficient > 97%
	How long/ wide/large is it?					
Image Classification	What type of the crack?	0.4 hours	0.03 seconds	0.6 hours	8.3 hours	Accuracy > 98%

5.2 Recommendations

The following recommendations are provided for future research endeavors.

- Expanding annotated datasets: It is recommended that the proposed methods be used to annotate more crack images. The current annotated datasets, while valuable, still lack a sufficient number of comprehensive annotations needed for robust segmentation tasks. In this research project, to address this limitation, publicly available road crack datasets were utilized to supplement the existing data and improve the training of deep learning models. For future research, A larger, high-quality dataset with well-annotated ground truth is crucial for enhancing the accuracy and reliability of crack segmentation.
- Enhancing training data: It is recommended that data augmentation techniques should be used to increase the variety of training data, thus improving the robustness of future models.
- Addressing limitations of current models: The U-Net model's limitations in identifying finer details and complex cracks and the tendency for DeepSegmentor to slightly over-segment, suggest that future research should focus on further refinement of current models.
- Applying developed methods to other distress types: It is recommended that the developed methods should be applied to other non-cracking distresses managed by the NCDOT PMS. This would broaden the scope of the research and demonstrate the versatility of the deep learning approach in addressing various types of pavement issues beyond just cracking.
- Testing the GoPro method on secondary routes: It is recommended that the proposed GoPro method should be tested on selected secondary routes in North Carolina. This is an important step for validating the effectiveness of the low-cost data collection method in real-world scenarios and evaluating the feasibility of using GoPro cameras for routine pavement assessments on secondary roads.

These recommendations collectively aim to improve the robustness and applicability of using deep learning for pavement condition assessment, leading to more efficient and cost-effective infrastructure management practices.

CITED REFERENCES

1. North Carolina Report Card | North Carolina Infrastructure | ASCE's 2021 Infrastructure Report Card, https://infrastructurereportcard.org/state-item/north-carolina/
2. Coenen, T. B. J. and Golroo, A. (2017). A Review on Automated Pavement Distress Detection Methods. Cogent Eng, vol. 4, no. 1, p. 1374822, 2017, [Online]. Available: https://www.tandfonline.com/doi/pdf/10.1080/23311916.2017.1374822.
3. Mei, Q. and Gül, M. (2020). A Cost Effective Solution for Pavement Crack Inspection using Cameras and Deep Neural Networks. Constr Build Mater, vol. 256, p. 119397, 2020, [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0950061820314021
4. Leduc., E. and Assaf, G., (2020). Road Visualization for Smart City: Solution Review with Road Quality Qualification. December 2020, Internet of Things 12(6):100305, DOI:10.1016/j.iot.2020.100305.
5. Raoult, V., David, P.A., Dupont, S.F., Mathewson, C..P, O'Neill, S.J., Powell, N.N., and Williamson, J.E. (2016). GoPros™ as an Underwater Photogrammetry Tool for Citizen Science. PeerJ. 2016 Apr 25;4:e1960. doi: 10.7717/peerj.1960. PMID: 27168973; PMCID: PMC4860335.
6. Balletti, C., Guerra, F., Tsioukas, V., and Vernier, P. (2014). Calibration of Action Cameras for Photogrammetric Purposes. Sensors, 14(9), 17471-17490. https://doi.org/10.3390/s140917471
7. https://goprotelemetryextractor.com/accident-reconstruction-with-gopro-telemetry
8. Markus Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U., and Grosso, H. (2017). How to Get Pavement Distress Detection Ready for Deep Learning? A Systematic Approach. International Joint Conference on Neural Networks (IJCNN), May 14-19, 2017. Anchorage, AK, USA, pp 2039-2047.
9. Guo, Y., Wang, Z., Shen, X., Barati, K., and Linke, J. (2022). Automatic Detection and Dimensional Measurement of Minor Concrete Cracks with Convolutional Neural Network. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume X-4/W3-2022, The 7th International Conference on Smart Data and Smart Cities (SDSC), October 19-21, 2022, Sydney, Australia.
10. Schroeder AB, Dobson ETA, Rueden CT, Tomancak P, Jug F, Eliceiri KW. The ImageJ ecosystem: Open-Source Software for Image Visualization, Processing, and Analysis. Protein Science. 2021; 30: 234–249. https://doi.org/10.1002/pro.3993
11. Ignacio Arganda-Carreras, I., Kaynig, V., Rueden, C., Eliceiri, K., Schindelin, J., Cardona, A., and Seung, H. (2017). Trainable Weka Segmentation: A Machine Learning Tool for Microscopy Pixel Classification. Bioinformatics, Volume 33, Issue 15, August 2017, Pages 2424–2426, https://doi.org/10.1093/bioinformatics/btx180
12. Kanuri, N., Abdelkarim, A., Rathore, S. (2022). Trainable WEKA (Waikato Environment for Knowledge Analysis) Segmentation Tool: Machine-Learning-Enabled Segmentation

on Features of Panoramic Radiographs. Cureus. 2022 Jan 31;14(1):e21777. doi: 10.7759/cureus.21777. PMID: 35251847; PMCID: PMC8890604.

13. https://fiji.sc/
14. Tan, Y., Deng, T., Zhou, J. Y., & Zhou, Z. X. (2024). LiDAR-Based Automatic Pavement Distress Detection and Management Using Deep Learning and BIM. Journal of Construction Engineering and Management, 150(7), 04024069. https://doi.org/10.1061/Jcemd4.Coeng-14358.
15. Anderson, K., Westoby, M. J., & James, M. R. (2019). Low-Budget Topographic Surveying Comes of Age: Structure from Motion Photogrammetry in Geography and the Geosciences. Progress in Physical Geography-Earth and Environment, 43(2), 163-173. https://doi.org/10.1177/0309133319837454.
16. Carrivick, J. L., & Smith, M. W. (2019). Fluvial and Aquatic Applications of Structure from Motion Photogrammetry and Unmanned Aerial Vehicle/Drone Technology. Wiley Interdisciplinary Reviews-Water, 6(1), e1328. https://doi.org/10.1002/wat2.1328.
17. Cook, K. L. (2017). An Evaluation of the Effectiveness of Low-Cost UAVs and Structure from Motion for Geomorphic Change Detection. Geomorphology, 278, 195-208. https://doi.org/10.1016/j.geomorph.2016.11.009 .
18. Cucchiaro, S., Fallu, D. J., Zhao, P., Waddington, C., Cockcroft, D., Tarolli, P., & Brown, A. G. (2020). SfM Photogrammetry for GeoArchaeology. In Developments in Earth Surface Processes (Vol. 23, pp. 183-205). Elsevier.
19. Guisado-Pintado, E., Jackson, D. W. T., & Rogers, D. (2019). 3D Mapping Efficacy of a Drone and Terrestrial Laser Scanner over a Temperate Beach-Dune Zone. Geomorphology, 328, 157-172. https://doi.org/10.1016/j.geomorph.2018.12.013.
20. Iglhaut, J., Cabo, C., Puliti, S., Piermattei, L., O'Connor, J., & Rosette, J. (2019, Sep). Structure from Motion Photogrammetry in Forestry: a Review. Current Forestry Reports, 5(3), 155-168. https://doi.org/10.1007/s40725-019-00094-3.
21. Roncoroni, M., Mancini, D., Kohler, T. J., Miesen, F., Gianini, M., Battin, T. J., & Lane, S. N. (2022). Centimeter-Scale Mapping of Phototrophic Biofilms In Glacial Forefields using Visible Band Ratios and UAV Imagery. International Journal of Remote Sensing, 43(13), 4723-4757. https://doi.org/10.1080/01431161.2022.2079963.
22. Smith, M. W., & Vericat, D. (2015). From Experimental Plots to Experimental Landscapes: Topography, Erosion and Deposition in Sub-Humid Badlands from Structure-from-Motion Photogrammetry. Earth Surface Processes and Landforms, 40(12), 1656-1671. https://doi.org/10.1002/esp.3747.
23. Ecke, S., Dempewolf, J., Frey, J., Schwaller, A., Endres, E., Klemmt, H. J., Tiede, D., & Seifert, T. (2022). UAV-Based Forest Health Monitoring: A Systematic Review. Remote Sensing, 14(13), 3205. https://doi.org/10.3390/rs14133205.
24. Smith, M. W., Carrivick, J. L., & Quincey, D. J. (2015). Structure from Motion Photogrammetry In Physical Geography. Progress in Physical Geography: Earth and Environment, 40(2), 247-275. https://doi.org/10.1177/0309133315615805.

25. Oliveira, H. and Correia, P. L. (2013). Automatic Road Crack Detection and Characterization. IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 1, pp. 155-168, March 2013, doi: 10.1109/TITS.2012.2208630.
26. Murphy, K. P. (2012). Machine Learning: a Probabilistic Perspective. Cambridge, MA, MIT Press.
27. Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). Foundations of Machine Learning. Cambridge, MA, MIT Press.
28. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539.
29. Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016). Synthesizing the Preferred Inputs for Neurons in Neural Networks via Deep Generator Networks. Advances in Neural Information Processing Systems, 29.
30. Deng, L., Yu, D., & Platt, J. (2012, March). Scalable Stacking and Learning for Building Deep Architectures. In 2012 IEEE International conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2133-2136). IEEE.
31. Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Tao, S., Chen, C., ... & Li, B. (2018). Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet. Journal of Computing in Civil Engineering, 32(5), 04018041.
32. Q. Zou, Y. Cao, Q. Li, Q. Mao and S. Wang, CrackTree: Automatic Crack Detection from Pavement Images, Pattern Recognit. Lett., vol. 33, no. 3, pp. 227-238, Feb. 2012.
33. Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep Learning ‐ Based Crack Damage Detection using Convolutional Neural Networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361-378.
34. Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., ... & Chen, C. (2017). Automated Pixel ‐ Level Pavement Crack Detection on 3D Asphalt Surfaces using a Deep ‐ Learning Network. Computer-Aided Civil and Infrastructure Engineering, 32(10), 805-819.
35. Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Tao, S., Chen, C., ... & Li, B. (2018). Deep Learning–Based Fully Automated Pavement Crack Detection on 3D Asphalt Surfaces with an Improved CrackNet. Journal of Computing in Civil Engineering, 32(5), 04018041.
36. Tsai, Y., & Huang, Y. (2012). A Generalized Framework for Parallelizing Traffic Sign Inventory of Video Log Images Using Multicore Processors. Computer-Aided Civil and Infrastructure Engineering, 27(7), 476-493.
37. Shen, S., Zhang, W., Wang, H., & Huang, H. (2017). Numerical Evaluation of Surface-Initiated Cracking in Flexible Pavement Overlays with Field Observations. Road Materials and Pavement Design, 18(1), 221-234.
38. Hadjidemetriou, G. M., & Christodoulou, S. E. (2019). Vision-and Entropy-Based Detection of Distressed Areas for Integrated Pavement Condition Assessment. Journal of Computing in Civil Engineering, 33(3), 04019020.

39. Fukushima, K. and Miyake, S. (1982). Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition, Competition and Cooperation in Neural Nets. Springer, 1982, pp. 267–285.
40. Chellapilla, K., Puri, S., and Simard, P. (2006). High Performance Convolutional Neural Networks for Document Processing, Tenth International Workshop on Frontiers in Handwriting Recognition. Suvisoft, 2006.
41. Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105, 2012.
42. Simonyan, K. and Zisserman, A., (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv preprint arXiv:1409.1556, 2014.
43. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., (2015). Going Deeper with Convolutions, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9
44. He, K., Zhang, X., Ren, S., and Sun, J., (2016). Deep Residual Learning for Image Recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
45. Long, J., Shelhamer, E., and Darrell, T., (2015). Fully Convolutional Networks for Semantic Segmentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
46. Ronneberger, O., Fischer, P., and Brox, T., (2015). U-net: Convolutional Networks for Biomedical Image Segmentation, International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2015, pp. 234–241.
47. Kr¨ahenb¨uhl, P. and Koltun, V., (2011). Efficient Inference in Fully Connected CRFS with Gaussian Edge Potentials, Advances in Neural Information Processing Systems, vol. 24, pp. 109–117, 2011.
48. https://connect.ncdot.gov/resources/Asset-Management/AssetManagementDocs/NCDOT%20High%20Speed%20Distress%20Manual%20V1.2%202-20-2024.pdf
49. https://www.pix4d.com/product/pix4dmapper-photogrammetry-software/

Appendix A. Lessons Learned from Annotation of GoPro Images

The open-source software used for both stages of this process, distortion correction and annotation, took significant time to learn how to use. They were useful and adequate for this project, but they were not user-friendly, particularly as the ITRE research team was unfamiliar with either the application or the processes of distortion correction and image annotation in general. UNCC research students who were familiar with both applications but not proficient trained the ITRE researchers and students on how to use both applications through web meetings and tutorial videos. All in all, there was a significant learning curve for everyone involved. As such, upon receiving the training and traversing each stage of this process, ITRE researchers noted some steps that were unnecessary or insufficient and thereby caused disruptions or difficulties in both distortion correction and annotation. Likewise, ITRE researchers noted areas where the applications showed promise and where doubts had previously arisen. These are described below.

Image Distortion Correction

The Hugin software was not perfect in removing all distortion from each image, as previously noted, but it performed well enough. The biggest lesson learned as this phase of removing distortion from each image progressed was not to allow Hugin to take unhelpful steps. Mainly, when ITRE researchers first began using Hugin for distortion correction, the last steps of correcting distortion were to “calculate field of view”, “calculate optimum size”, and “fit crop to images”. However, team members eventually learned that these steps automatically cropped a significant portion of the image – specifically, the area of the image that is closest to the camera was removed, even though this is the area of the image that is the clearest due to its proximity to the camera. This left only the area that was further from the camera and thereby the most distorted and blurry. Upon realizing this, ITRE researchers no longer utilized these last three steps, resulting in much clearer images, which are better for the classifier to achieve higher intelligence. These images looked odd initially due to their shape – which is presumably the reason Hugin removed this portion of the image – but this had no negative effect on the classifier when upon training it on these new images 1) because, as described in an earlier section, these images were cropped a second time, and 2) the classifiers eventually became smart enough on their own to recognize that the bevels in the bottom corners of the images were not cracks. An example of the outcome from Hugin after omitting the last three steps is shown below in Figure 41.

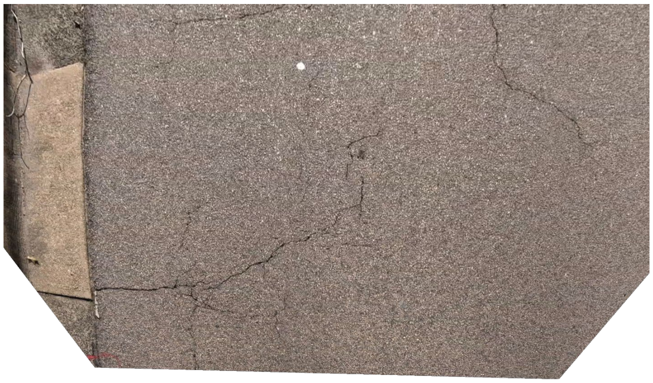


Figure 41. Hugin Output after Omitting Calculate FOV, Calculate Optimum Size, and Fit Crop to Images Steps

Image Annotation

There were two primary adjustments made between earlier annotations and later annotations. The first adjustment dealt with how cracks were selected and consisted of changing the width of the manual annotation line. Originally, ITRE researchers were trained to use a width of three pixels when manually annotating cracks to better "fill" the cracks – specifically the wider cracks. However, as the classifier would automatically fill in each crack regardless of width, the manual line width was reduced to a single pixel. The primary benefits of this change were fewer pixelated cracks and reduced image noise after training the classifier. The research team is not entirely certain, but anecdotally, it seemed that with a wider line width, portions of the pavement beyond the edges of the crack (i.e., background) were classified as a part of the crack, which confused the classifier. This created pixelated crack lines and noisier images. This is noted when zooming in on the image cracks in earlier images. Both issues are displayed in the following image Figure 42), with hatched or dotted marks composing thinner cracks and wider cracks having blurrier edges with less definition. This was simply a side effect of the classifier not being as intelligent as later iterations of the same classifier; however, upon reducing the line width the one pixel and making the adjustment noted below, this issue mostly resolved itself quickly, with only a few instances occurring in later annotations.

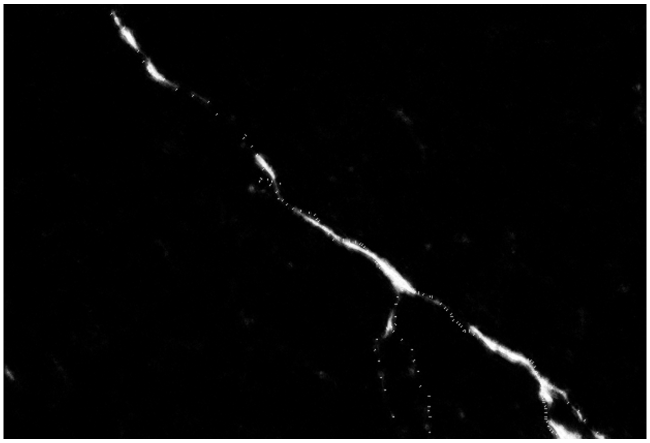


Figure 42. Pixelated and Cloudy Cracks

The second adjustment was in how the background was selected. Originally, the background was selected using the "rectangle" selection tool, which only allowed areas to be selected by a rectangle. The shape of the rectangle could vary, but it consisted of four straight edges, which did not allow the user to capture smaller, oddly shaped background areas. ITRE researchers eventually began using the "freehand" selection tool, which allowed for more accurate selections of background, particularly those areas near cracks. This resulted in a better definition of the cracks and significantly reduced noise in the background. Note in Figure 43 below the contrast on the edges of the thicker cracks – the reduced "cloudiness" of these lines – as well as the absence of hatched or dotted lines on the thinner cracks.

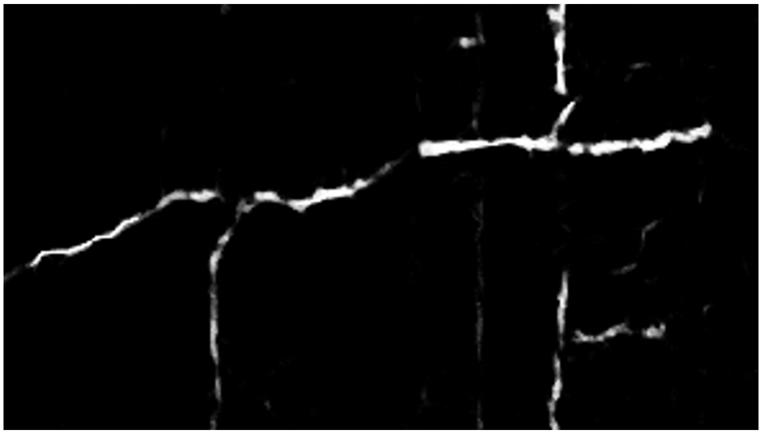


Figure 43. High Contrast, Connected Cracks

Lastly, Fiji performed well in areas that the research team had hypothesized would be problematic. The research team was initially concerned during the data collection process because of the presence of shadows, pavement markings, and debris in the collected imagery. The research team expected that the classifier would struggle to notice the difference between the edge of a shadow and a crack. Likewise, pavement markings and even spray-painted portions of the roadway due to utility work were areas of concern because of the contrast between those areas and the surrounding pavement. Lastly, debris being present on the pavement was expected to create confusion for the classifier, but the classifier passed this test, at least with small debris.

Fiji seemed to quickly adjust to the presence of shadows, with the annotations not displaying any indication of a difference between portions of roadway cloaked in shadow and those in direct sunlight. Examples are present below ranging from subtle shadows heavily present in an image (Figure 44) to a high contrast shadow traversing through an image (Figure 45). The corresponding annotations for these images are adjacent to each original image for reference.

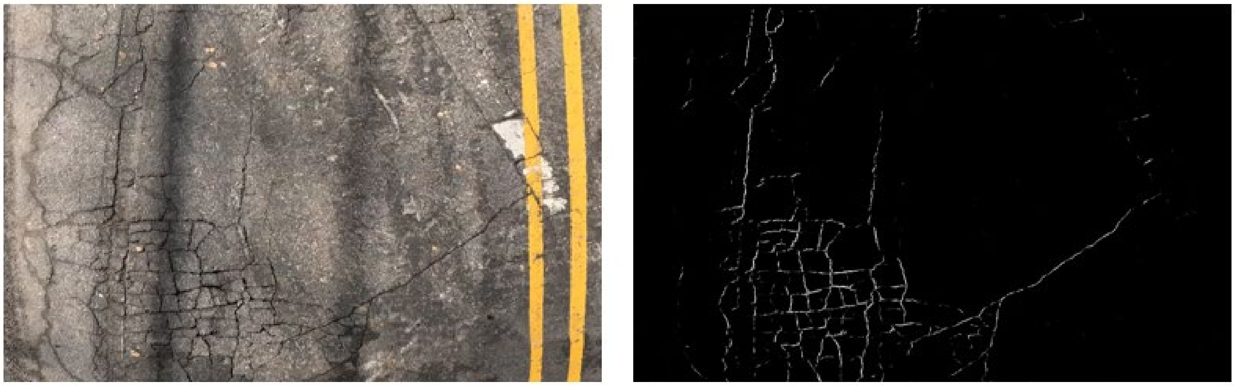


Figure 44. Subtle Shadows Present Throughout Image

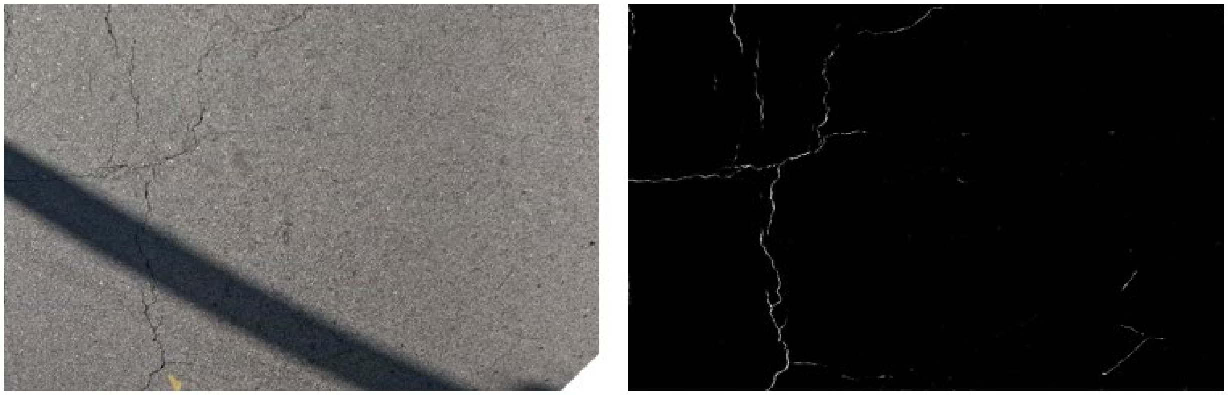


Figure 45. High Contrast Shadow Through Image

The presence of any paint or markings on the pavement was also thought to create confusion due to contrast. However, as can be seen below, this concern was also subdued. While some pavement markings could be noticed in annotations when cracks are not present, that is simply due to the lack of a crack rather than the contrast between the marking and the surrounding pavement. Two examples are shown below of pavement markings in images and their corresponding annotations (Figure 46, Figure 47).

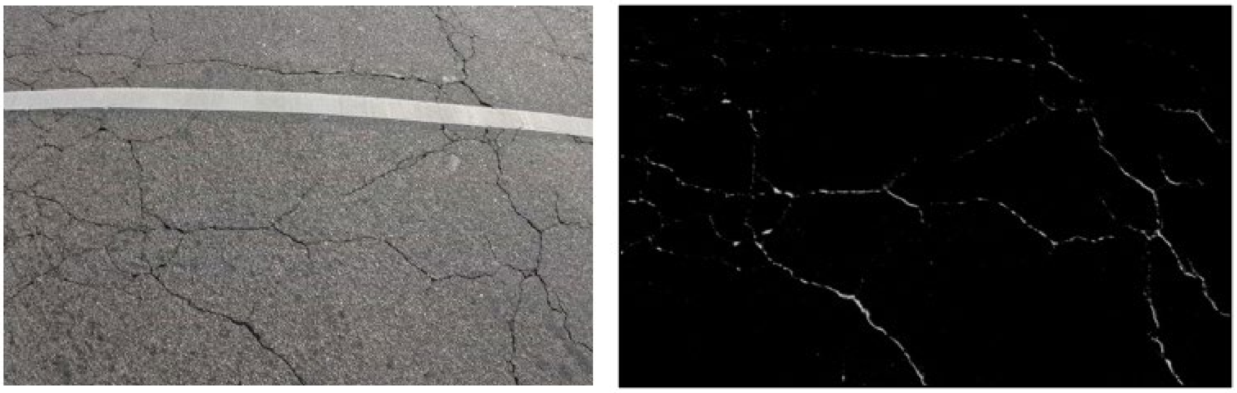


Figure 46. Horizontal Pavement Marking Through Image

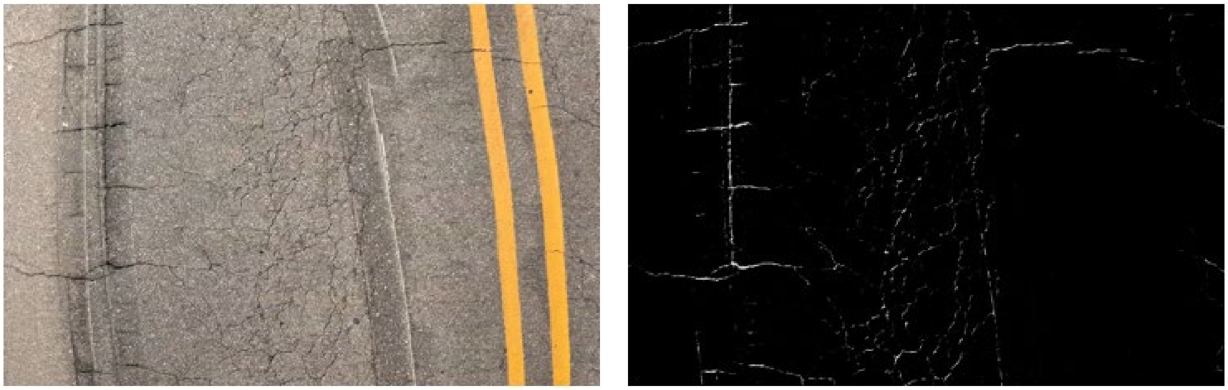


Figure 47. Vertical Pavement Markings Through Image

When paint was present on the pavement, the difference in color between the paint and adjacent pavement did not show through in the annotations, and if cracks were present in the painted portion, those displayed the same in the annotations as the unpainted areas. Examples are shown below of images with paint present on the pavement and their corresponding annotations (Figure 48, Figure 49).

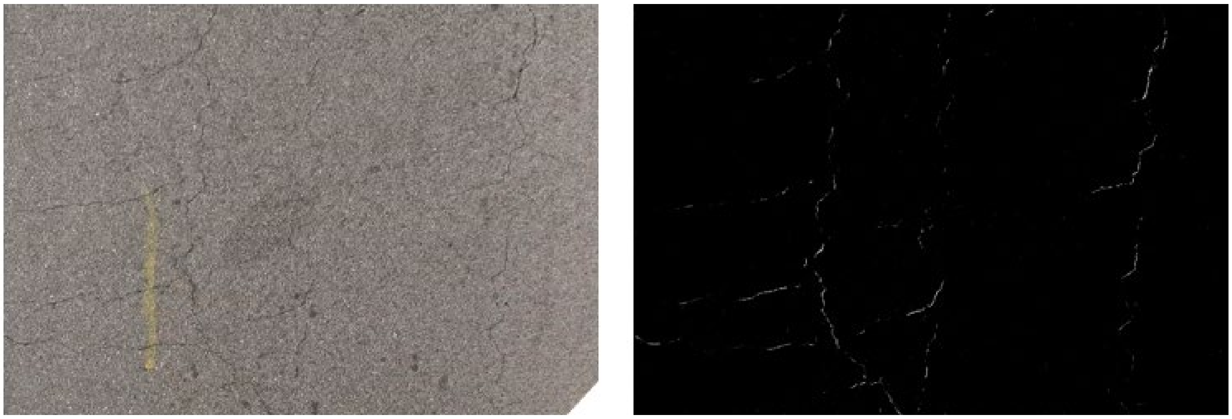


Figure 48. Paint Present on Pavement Across Cracks and Background (I)

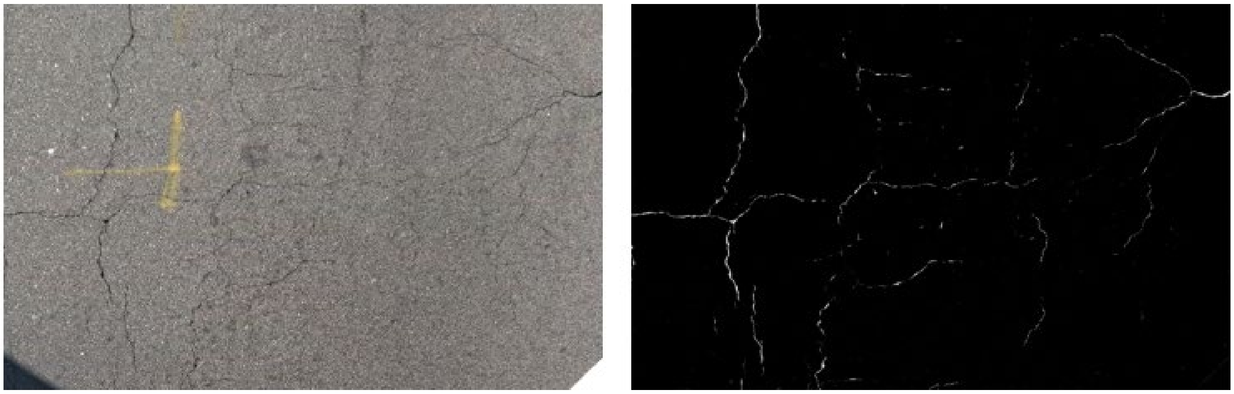


Figure 49. Paint Present on Pavement Across Cracks and Background (II)

Lastly, small debris present on the pavement likewise did not cause significant issues for the classifier. For the first image (Figure), the debris was slightly present initially (Figure 50) but was no longer present in the third and final annotation (Figure 51). For the second image (Figure 52), the provided annotation is the third annotation of the image (Figure 53), but the debris did not appear in any of the successive annotations for this image. The debris shown in these images is minimal, and the performance of the classifier with a larger quantity or different types of debris (such as coarser gravel or sticks) remain uncertain. However, one could easily argue that the presence of heavier debris on the pavement is not only an issue for computer vision software but also for human raters conducting windshield surveys.

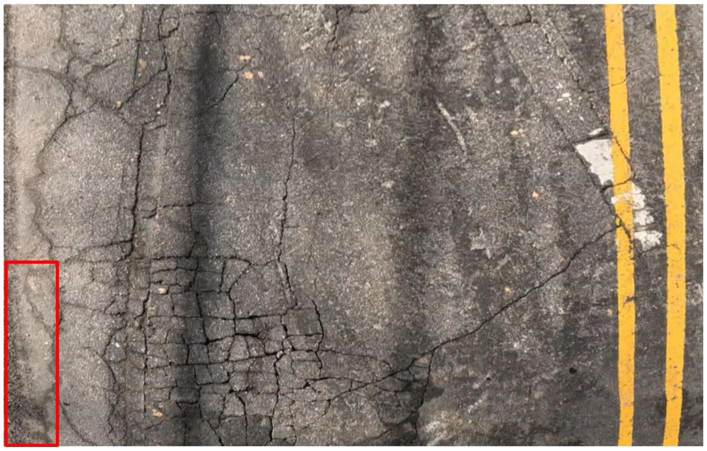


Figure 49. Original Image with Debris Present (Highlighted)

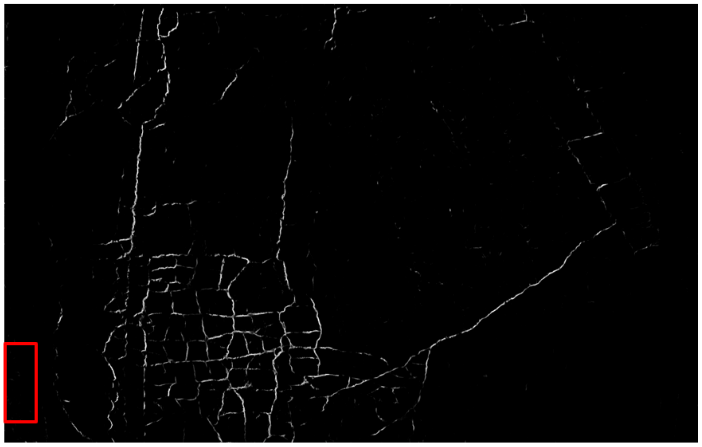


Figure 50. First Annotation of Above Image with Debris Showing Slightly (Highlighted)

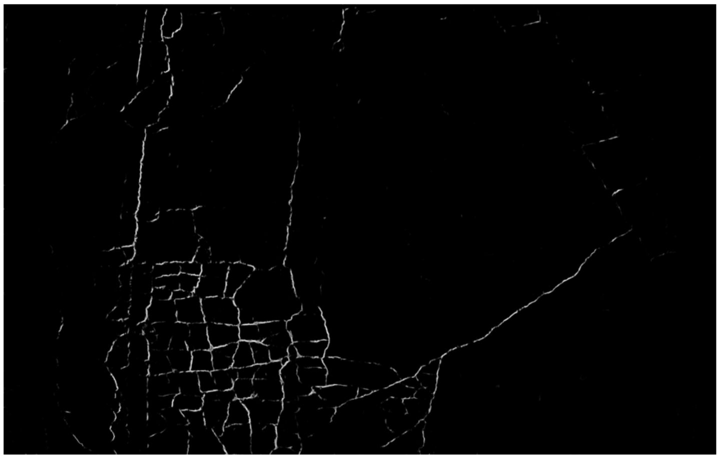


Figure 51. Third and Final Annotation of Above Image with No Debris Present and No Loss of Contrast in Cracks

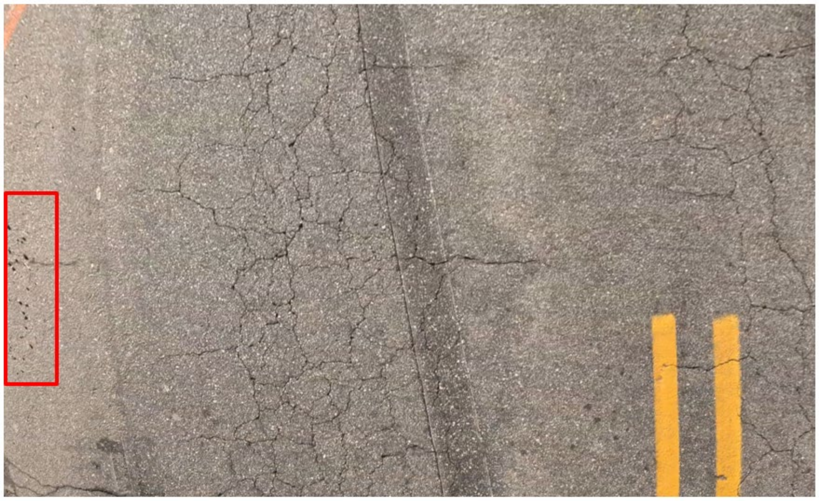


Figure 52. Original Image with Debris Present (Highlighted)

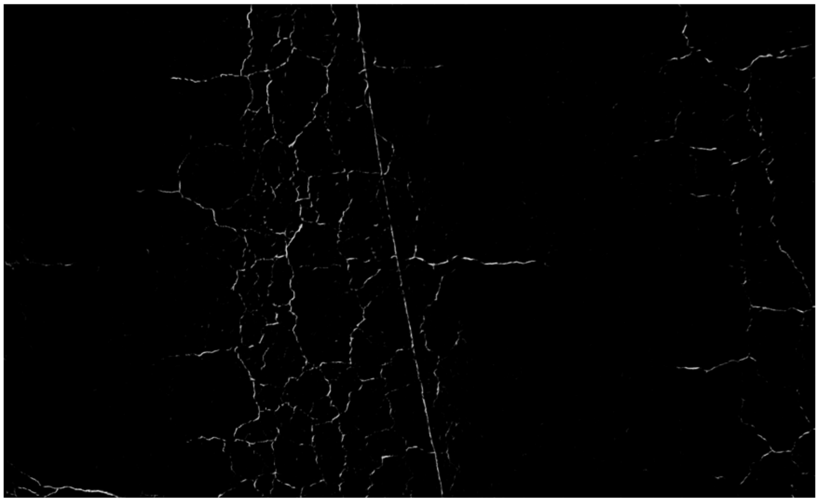


Figure 53. Annotation Not Showing Debris