TEXAS STATE UNIVERSITY

The rising STAR *of Texas*

TxDOT Report 0-7150-R1
Artificial Intelligence for Pavement Condition Assessment from 2D/3D Surface Images

Texas State University, Ingram School of Engineering; San Marcos, Texas
Feng Wang (PI)
Haitao Gong
Yongsheng Bai
Jelena Tesic
Xiaohua Luo

Submitted March 2026; Published April 2026

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
FHWA/TX-26-0-7150-R1			
4. Title and Subtitle Artificial Intelligence for Pavement Condition Assessment from 2D/3D Surface Images		**5. Report Date** March 2026 6. Performing Organization Code	
7. Author(s) Haitao Gong, Yongsheng Bai, Jelena Tesic, Xiaohua Luo, and Feng Wang		**8. Performing Organization Report No.** Report 0-7150-R1	
9. Performing Organization Name and Address Texas State University Ingram School of Engineering 601 University Drive San Marcos, TX 78666		**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** Project 0-7150	
12. Sponsoring Organization Name and Address Texas Department of Transportation Research and Technology Implementation Division P.O. Box 5080 Austin, TX 78763-5080		**13. Type of Report and Period Covered** Technical Report 09/2022-03/2026 **14. Sponsoring Agency Code**	
15. Supplementary Notes Project performed in cooperation with the Texas Department of Transportation and the Federal Highway Administration			

16. Abstract

The development of a national standard data format for two-dimensional/three-dimensional (2D/3D) pavement surface images and Artificial Intelligence (AI) Machine Learning (ML) in Computer Vision adopted by AASHTO and embraced by state DOTs has provided TxDOT the opportunity to develop new methods for automated pavement condition assessment. This research implements automated pavement condition evaluation using 2D/3D surface imagery and Artificial Intelligence. A comprehensive image library was established in the AASHTO standard data format, capturing diverse pavement conditions and surface types in Texas. The dataset includes 5,892 2D/3D image pairs and associated labels for Asphalt Concrete (ACP), 7,750 for Jointed Concrete (JCP), and 5,776 for Continuously Reinforced Concrete Pavement (CRCP). These correspond to 10,885, 16,943, and 13,779 individual distress instances, respectively, as defined by the TxDOT Pavement Management Information System (PMIS) for a total of 19,418 images. Neural network models such as YOLO (You Only Look Once) series were trained on the established library datasets for generalization and robustness of pavement distress measurements. Leveraging vision-based AI and ML models, the system automatically detects and measures surface distresses, achieving a mAP50 of over 0.80 for ACP, 0.70 for JCP, and 0.75 for CRCP on the validation datasets. These models were integrated into practical tools for calculating PMIS distress scores and delivered via a dedicated software application for TxDOT. A pilot study was implemented using both the library and vendor-collected field data in 2024 to validate the efficacy of the proposed methods. The research concludes with actionable recommendations and a strategic roadmap designed to transition these findings into full-scale operational implementation within TxDOT's pavement management system.

17. Key Words	18. Distribution Statement		
Automated pavement condition data collection, data quality management, quality assurance, data accuracy and precision, sampling method, data quality threshold	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161; www.ntis.gov.		
19. Security Classification (of this report)	**20. Security Classification (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified.	Unclassified.	216 pages	N/A

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Disclaimers

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official view or policies of the Federal Highway Administration or the Texas Department of Transportation (TxDOT). This report does not constitute a standard, specification, or regulation.

Acknowledgments

The authors would like to express appreciation to the Texas Department of Transportation for sponsoring this research. Special gratitude to the following individuals who provided support and guidance to the research team during the project study: Project Manager Jade Adediwura and all the Project Monitoring Committee members. We would also like to thank the following TxDOT personnel who provided support and technical input during the study: Jenny Li, Hui Wu, Arturo Perez, Andre Smit, Daniel Garcia, and Jesus Garcia.

ARTIFICIAL INTELLIGENCE FOR PAVEMENT CONDITION ASSESSMENT FROM 2D/3D SURFACE IMAGES

Final Report
March 2026

Principal Investigator
Feng Wang, PhD, PE
Ingram School of Engineering, Texas State University

Investigator
Haitao Gong, PhD
Ingram School of Engineering, Texas State University

Investigator
Yongsheng Bai, PhD
Ingram School of Engineering, Texas State University

Investigator
Jelena Tesic, PhD
Ingram School of Engineering, Texas State University

Investigator
Xiaohua Luo, PhD
Ingram School of Engineering, Texas State University

Authors
Feng Wang, Haitao Gong, Yongsheng Bai, Jelena Tesic, and Xiaohua Luo

Sponsored by
Texas Department of Transportation

A report from
Ingram School of Engineering
Texas State University
601 University Drive
San Marcos, TX 78666

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Introduction

According to the Federal Highway Administration (FHWA), there are approximately 4.20 million miles of public roads in the United States. Texas has a total of 0.68 million miles. Automated pavement condition survey systems including two-dimensional or three-dimensional (2D/3D) laser line scan cameras mounted on the vehicles have been used to scan about 100,000 lane miles each year since 2017 for pavement condition evaluation in Texas. The collected 2D/3D images are processed by computer vision algorithms to deliver information to evaluate pavement surface condition and provide support for maintenance and rehabilitation (M&R) decision-making. But it is challenging for agencies like the Texas Department of Transportation (TxDOT) to manage consistent pavement condition evaluation when different vendors are using different condition survey systems and producing vast amounts of image data in different formats. An initiative was launched to develop standard data format for 2D/3D pavement surface images from the FHWA to overcome this technical barrier. This project aimed to create a library of standard format 2D/3D pavement surface images to comply with this requirement. Also, there is a need for TxDOT to perform pavement condition assessment with more independence from the data collection vendors. In addition, it is important to develop artificial intelligence (AI) and machine learning (ML) methods to upgrade or replace the existing image processing algorithms to improve the accuracy and speed performance of the automated pavement surface condition evaluation systems.

The project started with reviewing the literature of available datasets, established AI models, and practices in the U.S. and other countries. The research team investigated the acquisition of high-resolution images with American Association of Highway and Transportation Officials (AASHTO) standard MP47 (AASHTO, 2020) and created a tool for viewing the vendor's data for TxDOT and labeling different distresses on three pavement types (Asphalt Concrete Pavement (ACP), Jointed Concrete Pavement (JCP), and Continuously Reinforced Concrete Pavement (CRCP), respectively). A comprehensive image library was established in the AASHTO standard data format, capturing diverse pavement conditions and surface types in Texas. Representative 2D/3D images were selected carefully to include diverse and various pavement distress types and instances for the annotation of distresses on the pavement images. For each distress type, the selected images were labeled with bounding boxes to manually locate and mark the pavement surface distresses, as defined in the Pavement Rater's Manual of TxDOT's Pavement Management Information System (PMIS) (TxDOT, 2023). Segmentation masks were created for a small portion of the images for potential applications as well. Model training was conducted to develop robust and generalized AI and ML models. Neural network models such as YOLO (You Only Look Once) series were trained on the established library datasets for generalization and robustness of pavement distress detection and measurements. These models were integrated into practical tools and application programming interface (API) for calculating PMIS distress scores and delivered via a dedicated software application for TxDOT. A pilot study was carried out to validate the developed AI/ML models and pavement condition evaluation methods using the image data newly collected in 2024.

The detailed investigation of the research is structured into seven chapters, each dedicated to a specific task as shown below:

Chapter 2 reports a review of the literature on automated pavement condition assessment and AI applications. It also includes the work to look into standards and protocols for pavement condition assessment and traditional pavement distress measurement.

Chapter 3 presents the findings for the creation of a 2D/3D image library in the AASHTO standard. The methodology to acquire and preprocess data collected by the TxDOT's vendor is addressed. Annotation details for library construction are provided, including the statistical analysis of data quality.

Chapter 4 addresses the study using rules-based methods for pavement distress measurement to explore the capabilities and limitations of the current practice.

Chapter 5 details how the research team used AI/ML-based methods for pavement distress detection and measurement across diverse pavement types. The performance of the models on different datasets is discussed, and various factors that influence the model's performance are addressed.

Chapter 6 presents the framework developed for the pavement condition assessment using PMIS formulas and following TxDOT's standards. The framework includes using the established AI/ML models to detect and measure various pavement distresses for each pavement type, and to establish the workflow to calculate distress scores to evaluate pavement conditions.

Chapter 7 summarizes findings from a pilot study that benchmarked model performance against training, validation, and testing using newly collected image data. It covers the evaluation of the latest AI/ML models and provides recommendations for future efforts to ensure the research results are fully implemented in practical pavement management.

Chapter 8 integrates the key conclusions derived throughout the research and outlines actionable recommendations to transition the project's deliverables from current phase to full-scale implementation for TxDOT.

Chapter 2 Literature Review

This chapter presents a review of the literature on automated pavement condition assessment and artificial intelligence applications.

2.1 Standards and protocols related to pavement condition assessment

The literature review in this chapter is mainly based on but not limited to the following:

- AASHTO MP 47: File Format of 2-Dimensional and 3-Dimensional (2D/3D) Pavement Image Data (AASHTO, 2020)
- AASHTO R 86: Collecting Images of Pavement Surfaces for Distress Detection (AASHTO, 2018a)
- AASHTO R 85-18: Quantifying Cracks in Asphalt Pavement Surfaces from Collected Pavement Images Utilizing Automated Methods (AASHTO, 2018b)
- TxDOT: Pavement Manual (TxDOT, 2021)
- TxDOT: Pavement Rater's Manual (TxDOT, 2023)

Pavement condition assessments are conducted to determine the functional and structural conditions of a highway section either for purpose of routine monitoring or planned corrective action (TxDOT, 2021). Visual condition surveys are one of the assessment types, which are performed to document aspects of both functional and structural pavement conditions but generally serve as a qualitative indicator of the overall condition (TxDOT, 2021). Currently, TxDOT is contracting with Pathway Services Inc. to implement network-level automated pavement condition surveys. Among all kinds of data collected in the survey, 2D/3D pavement surface images is one of the main sources for condition evaluation, especially for surface distress quantification.

2.1.1 TxDOT visual evaluation protocol

The latest TxDOT Pavement Rater's Manual (TxDOT, 2023) has defined the methods for conducting evaluations of Asphalt Concrete Pavement (ACP), Continuously Reinforced Concrete Pavement (CRCP), and Jointed Concrete Pavement (JCP) sections. Each pavement type has unique classifications of surface distresses, of which the ratings of a pavement section are combined to make the Distress Score. The Distress Score ranges from 2.1 (most distressed) to 100 (least distressed), with a score below 80 indicating problems. Table 1 shows the Distress Score classes used in TxDOT Pavement Management Information System (PMIS)

Table 2.1 Distress Scores

Distress score	Class	Description
90-100	"A"	Very good
80-89	"B"	Good
70-79	"C"	Fair
60-69	"D"	Poor
1-59	"F"	Very poor

2.1.1.1 Distress types for flexible pavement

The types of distresses for ACP according to the TxDOT Pavement Rater's Manual (TxDOT, 2023) are listed in Table 2.2. Among these distress types, rutting, patching, block cracking, alligator cracking, and longitudinal cracking are measured by the length. Transverse cracking and failures are measured by numbers. Raveling and flushing are measured by certain rating codes.

Table 2.2 Distress types for flexible pavement sections

1	Rutting - shallow
2	Rutting - deep
3	Patching
4	Block cracking
5	Alligator cracking
6	Longitudinal cracking
7	Transverse cracking
8	Raveling
9	Flushing
10	Failures

According to the current practice, the 2D/3D surface images are not used for the assessment of all distress types. For example, existing literature on distress measurement is mainly focused on transverse cracking, longitudinal cracking, alligator cracking, and block cracking, with patching fairly being studied. For raveling and flushing, there has not been any solid research on using 2D/3D surface images for assessment

2.1.1.2 Distress types for rigid pavement

The types of distresses for CRCP and JCP according to the TxDOT Pavement Rater's Manual (TxDOT, 2023) are listed in Tables 2.3 and 2.4, respectively. Compared to the distress types of flexible pavements, distress types of rigid pavements are a bit more complicated. For example, the punchouts are the composites of longitudinal cracks and transverse cracks according to the description in the manual. To accurately assess the condition of the rigid pavements, it is essential to further specify the definitions of each distress type (such effort is also focused in NCHRP 01-57B).

Table 2.3 Distress types for CRCP

1	Spalled cracks/Longitudinal cracking
2	Punchouts
3	Asphalt patches
4	Concrete patches
5	Average crack spacing

Table 2.4 Distress types for JCP

1	Failed joints and cracks
2	Failures
3	Shattered slabs
4	Slabs with longitudinal cracks
5	Concrete patches
6	Apparent joint spacing

2.1.2 Data format

The digital image is an image composed of pixels, each with finite, discrete quantities of numeric representation for its intensity and/or other properties that are output from its 2-dimensional functions fed as inputs by its spatial coordinates denoted as x, and y on the x-axis and y-axis, respectively. Based on the state-of-practice imagery technologies developed for pavement surface data collection, a standard for the file format of 2D and 3D pavement image data was proposed recently (AASHTO, 2020). The industry standard image representation for pavement surfaces is shown in Figure 2.1, in which a row parallel to x-axis and a column parallel to y-axis corresponds to a transverse profile and a longitudinal profile, respectively. Each image is recommended to cover one lane of 4 meters in width and about 5 meters in length, which can ensure a comprehensive evaluation of a full lane.

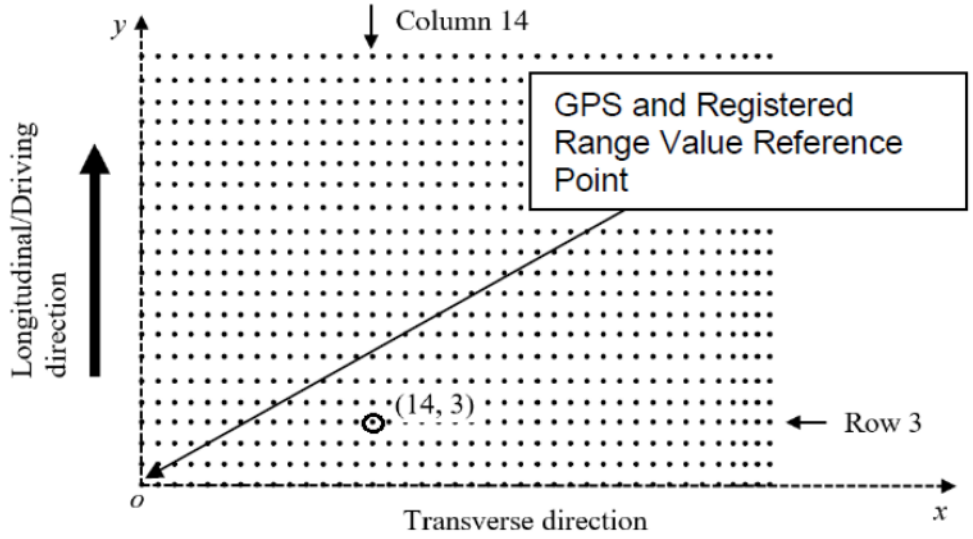


Figure 2.1 Standard image representation for pavement surface (AASHTO, 2020)

2.2 Traditional methods for distress measurement

Traditional automated distress measurement methods focus on separating the crack pixels from the pavement background. Intensity thresholding and edge detection are the two main methods for crack segmentation.

2.2.1 Intensity thresholding

Intensity thresholding is a classic approach in image segmentation, which converts the input grayscale image into a binary image based on the intensity threshold (Zhu et al., 2007). The basic assumption is that crack pixels are relatively darker than other pixels in grayscale digital images, thus cracks can be separated from the background by setting a proper intensity threshold. A dynamic threshold is developed to deal with distinctive mean pixel intensities of different images (Oliveira and Correia, 2009). However, it fails to cope well with images with non-uniform illumination. A sample of segmentation results using a thresholding method on a non-uniform 3D image is shown in Figure 2.2.

2.2.2 Edge detection

Edge detection approaches are adopted for crack detection, combined with other image processing methods to improve the measurement performance. Ayenu-Prah and Attoh-Okine (2008) combine bi-dimensional empirical mode decomposition with Sobel edge detection to remove noise in the pavement images. Wang et al. (2007) apply Wavelet Transform to decompose the original image into different subsamples to capture the details of the cracks on different scales. However, both methods are unable to generate complete crack profiles. Seed-based crack detection is implemented for real-time crack detection (Huang and Xu, 2006; Zhou et al., 2016). This method divides the image into cells of pixels and then decides the cells as crack or non-crack based on the contrast pixels. This method works very fast, but it is hard to find universal thresholds for images of dissimilar contrast. A dynamic optimization-based method aims to utilize global information to estimate the probability of crack existence (Tsai et al., 2010). This method formulates the problem as an optimization task with four primary parameters, which entails a long processing time. CrackTree extracts crack seeds through the crack probability map that is generated through local intensity contrast and tensor voting (Zou et al., 2012). This method can detect crack curves based on the constructed crack probability map. However, the crack width information is neglected. A sample of segmentation results using an edge detection method on a non-uniform 3D image is shown in Figure 2.3.

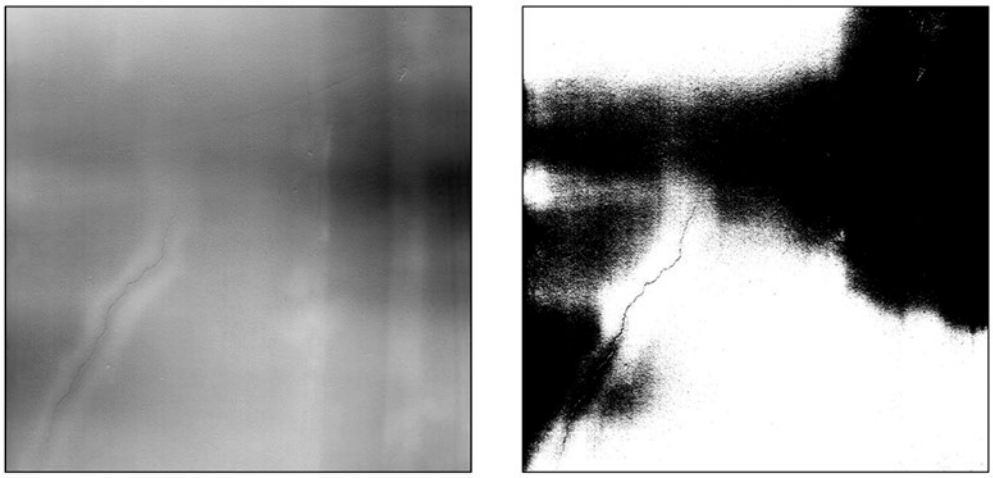


Figure 2.2 Segmentation result using thresholding method on a non-uniform 3D image

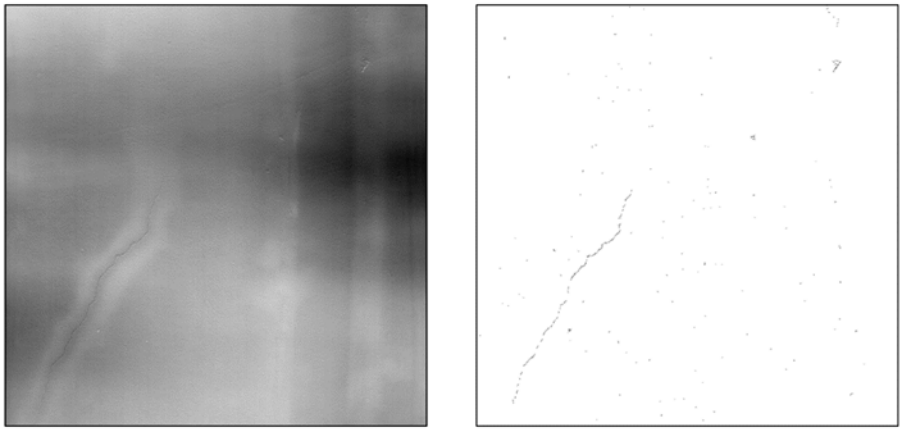


Figure 2.3 Segmentation result using edge detection method on a 3D image

One of the main challenges of distress measurement is to deal with diversified characteristics of pavement surface images. In some circumstances, a crack in either a 2D line scan image or a 3D intensity image may show brighter than the background or simply disappears from the image, due to the geometry relationship of the crack and light, including incident angle, and bottom flatness of the crack. Old and wide cracks, especially filled with deposits, may reflect more light than non-cracking areas. Due to the complexity of the pavement surface image characteristics, the traditional rules-based methods usually can hardly yield uniform performance over network-level datasets

2.3. Artificial Intelligence-based methods for distress measurement

2.3.1 Background

2.3.1.1 Segmentation

Image segmentation can be formulated as the problem of classifying pixels with semantic labels or the partition of individual objects (Minaee et al., 2021). Traditional image segmentation

methods, such as thresholding and edge detection, require predefined parameters and complex data pipelines for the application. In recent years, DL-based methods, which require little or no prior image processing, have been achieving remarkable performance improvement in image segmentation.

Fully Convolutional Network (FCN) is one of the earliest and simplest image segmentation methods (Long et al., 2015). It only includes convolutional layers and pooling layers. One issue in this specific FCN is that the resolution of the output features is gradually downsampled, resulting in blurred object boundaries. Most of the segmentation methods adopt an encoder-decoder architecture, which is further developed into an architecture called U-Net (Ronneberger et al., 2015). U-Net uses skip connections in DL-based models to solve the information loss caused by downsampling in typical encoder-decoder networks. Following U-Net, more techniques are developed to help improve prediction accuracy and efficiency. Atrous convolutions are adopted by many models to replace simple pooling operations and prevent significant information loss (Chen et al., 2017, Yu and Koltun, 2015, Wang et al., 2018). A multiscale network is developed to better learn the global context representation of a scene (Zhao et al., 2017). To recover boundary information, fully connected Conditional Random Fields (CRFs) are adopted to refine the coarse feature map based on the label (Chen et al., 2017). Recurrent Neural Networks and Generative Adversarial Networks are also used for image segmentation.

2.3.1.2 Object detection

DL-based object detection methods can mainly be categorized into two types: two-stage detection and one-stage detection. To localize objects in the image, corresponding bounding boxes need to be proposed in the prediction process. Two-stage detection methods typically have a separate module to generate regional proposals besides the feature extraction module (Zaidi et al., 2021). This methodology includes R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick et al., 2015), Faster R-CNN (Ren et al., 2015), R-FCN (Dai et al., 2016), FPN (Lin et al., 2017), and Mask R-CNN (He et al., 2017). One-stage detection methods consider detection as a regression problem, thus using a single neural network to perform both classification and localization (Zaidi et al., 2021). This methodology includes YOLO (Redmon et al., 2016) and its variations, SSD (Liu et al., 2016), RetinaNet (Lin et al., 2017), CenterNet (Duan et al., 2019), and EfficientDet (Tan and Le, 2019).

Two-stage object detection methods utilize a separate module to generate region proposals. For Faster R-CNN, Region Proposal Network (RPN) is used to take the input of the feature map and output a set of rectangular object proposals, each with an objective score (Ren et al., 2015). The RPN uses a sliding window with a specific size to slide all over the feature map and generate k anchor boxes (region proposals) of different sizes and shapes (as shown in Figure 4). These proposals are then fed to two layers: the *reg* layer for encoding the coordinates of k bounding boxes, and the *cls* layer for estimating the probabilities of object/no-object for each proposal. The corresponding loss function is expressed as Equation 2.1.

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \tag{2.1}$$

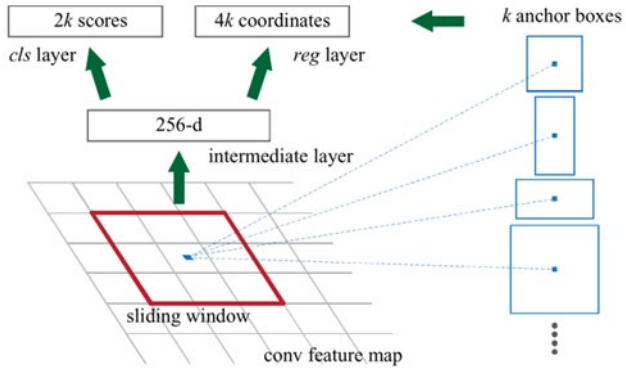


Figure 2.4 Region Proposal Network (Ren et al., 2015)

The RPN loss function contains two parts. The first part calculates the loss between the predicted probability of anchors containing an object and the ground truth label. In Equation 5: *i* is the index of an anchor in a mini-batch; p_i stands for the predicted probability of anchor i containing an object; p_i^* is the ground truth label, the value of which is set as 1 if the anchor is positive, and 0 otherwise. The second part of the function calculates the difference between the predicted bounding boxes and the ground truth boxes. t_i stands for a vector that represents the 4 parameterized coordinates of the predicted bounding box, and t_i^* is that of the ground truth box. L_{cls} and L_{reg} are loss functions. The two parts are normalized by N_{cls}, N_{reg}, and balancing weight λ.

For one-stage detection methods, there is no separate region proposal module. Predicted bounding boxes are learned along with classification through the same network. Figure 2.5 shows the architecture of YOLO, which is the most representative one-stage model. YOLO divides the input image into a $S \times S$ grid, and each grid generates B bounding boxes with confidence scores and class probabilities. In the prediction process, the convolutional layers are followed by two fully connected layers, which map the extracted features into the final prediction of shape $S \times S \times (B \times (5+n))$.

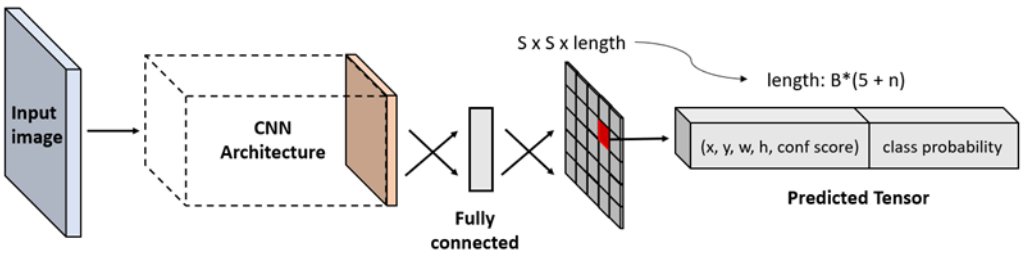


Figure 2.5 YOLO architecture

For each predicted bounding box, the prediction takes the form of $[x, y, w, h, confidence, c_0, c_1, \ldots, c_n]$, where: (x, y) and (w, h) denotes the center coordinate and size of the predicted bounding box, respectively, $confidence$ score indicates the likelihood that the bounding box contains an object, and $(c_0, c_1, \ldots c_n)$ indicates the predicated probabilities for all classes. The corresponding loss function is expressed in Equation 2.2 (Redmon et al., 2016):

$$
\begin{aligned}
L = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - x_i^*)^2 + (y_i - y_i^*)^2] \\
& + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{w_i^*})^2 + (\sqrt{h_i} - \sqrt{h_i^*})^2] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} (C_i - C_i^*)^2 \\
& + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{noobj} (C_i - C_i^*)^2 \\
& + \sum_{i=0}^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - p_i^*(c))^2
\end{aligned}
\tag{2.2}
$$

Where 1_i^{obj} denotes if object appears in cell i and 1_{ij}^{obj} denotes that the jth bounding box prediction in cell i is responsible for that prediction. By incorporating these two parameters, the loss function only penalizes bounding box coordinate error if that prediction is responsible for the ground truth box, and also only penalizes the classification error if an object is present in that cell.

Besides the models mentioned above, many other object detection models have been developed with modifications to model architecture, such as feature extraction backbone, loss function, and so on, aiming to further improve the performance of object detection.

2.3.1.3 Evaluation metrics

Precision, Recall, and F1 are usually used to evaluate the model performances. Precision is defined as the ratio of correctly detected objects to all detected objects. Recall is defined as the ratio of correctly detected objects to all actual objects. F1 is a weighted combination of Precision and Recall used to measure the overall performance, as there is always a trade-off between Precision and Recall. These three indicators can be expressed as the following equations:

$$
precision = \frac{tp}{tp + fp} \tag{2.3}
$$

$$recall = \frac{tp}{tp + fn} \tag{2.4}$$

$$F1 = \frac{2 \times (precision \times recall)}{precision + recall} \tag{2.5}$$

Where *tp* denotes the number of true positives, fp denotes the number of false positives, fn denotes the number of false negatives. Beside the metrics mentioned above, Average Precision (AP) is also used to indicate the overall performance of deep learning models. AP is the weighted mean of precisions at each threshold, which is also considered as the area under the recall/precision curve.

2.3.2 AI-based distress measurement

2.3.2.1 Distress segmentation

Almost all deep learning-based approaches for pavement distress segmentation in recent years adopt the encoder-decoder structure and use Fully Convolutional Networks (FCNs) as the basic concept for prediction. A basic FCN-based model is developed by Dung and Anh, with VGG16 as the encoder and deconvolution layers and upsampling layers as the decoder, as shown in Figure 6 (Dung and Anh, 2019). For the output feature maps of each level in the encoder, there are corresponding feature maps of the same size, obtained through upsampling and deconvolution based on the outputs of the former layers. The last layer is a detection layer, where each pixel is classified into 'crack' or 'non-crack' classes. One shortcoming of this method is that the maxpooling layer will cause the loss of spatial information, resulting in coarse predictions.

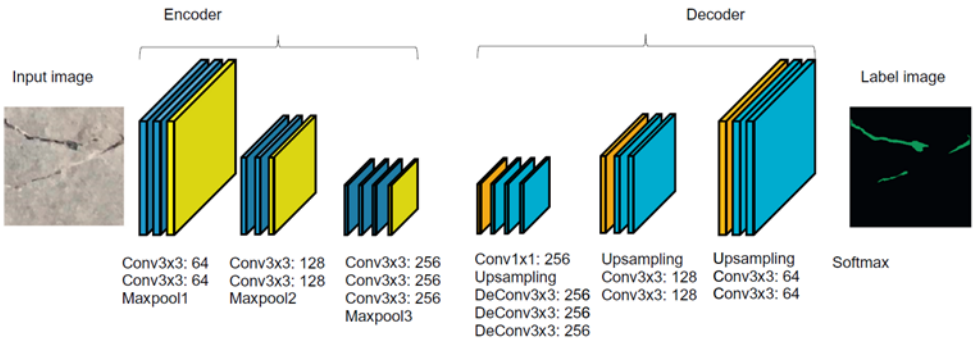


Figure 2.6 The network architecture of FCN for crack segmentation (Dung and Anh, 2019)

Yang et al. also developed an FCN-based method for crack segmentation (Yang et al., 2018). To reserve spatial information, high-level feature maps in the decoder are combined with low-level feature maps, which are commonly known as skip connections, as shown in Figure 2.7. As low-level feature maps contain fine spatial information, this structure can efficiently refine the boundaries of the segmentation while retaining high-level semantic information. This is of special significance for crack segmentation, as thin cracks may only be 1 to 2 lines thick. Yang et al. apply the developed method to a concrete image dataset, obtaining an F1 score of 0.80 (Yang et al., 2018). Bang et al. apply a similar method with different backbones for crack segmentation

using vehicle rear camera images (Bang et al., 2019). The best performance is yielded from a model with ResNet-152 as the backbone, with F1 score of 0.75.

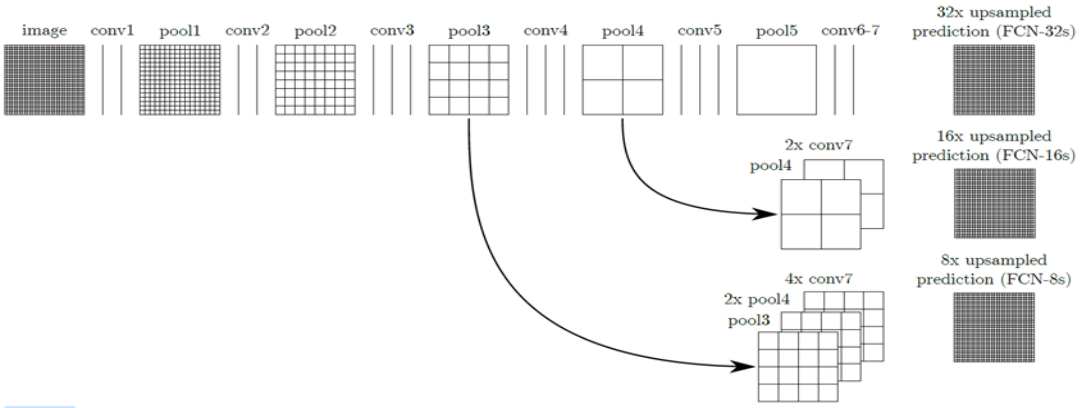


Figure 2.7 Combination of coarse high-level feature maps with fine low-level feature maps (Long et al., 2015)

Jenkins et al. adopt U-net, which performs skip connections on all levels of feature maps, as shown in Figure 2.8 (Jenkins et al., 2018). It is noticed that the size of the output image is much smaller than the input image, due to each convolution operation without padding. While the aforementioned studies are not tested on the same dataset, it is yet known how the skip connections would affect the final performance. Zou et al. develop DeepCrack, a similar architecture to U-net (Zou et al., 2018). Instead of relying on the final layer for loss calculation, DeepCrack assigns loss to each level, along with the final layer. In doing so, the author believes that the model could effectively capture information on thin cracks at each scale. Based on the test results over the open-source datasets, CrackTree 260 (Zou et al., 2012), CRKWH100, CrackLS315, and Stone331, DeepCrack yields better performance than that of U-net.

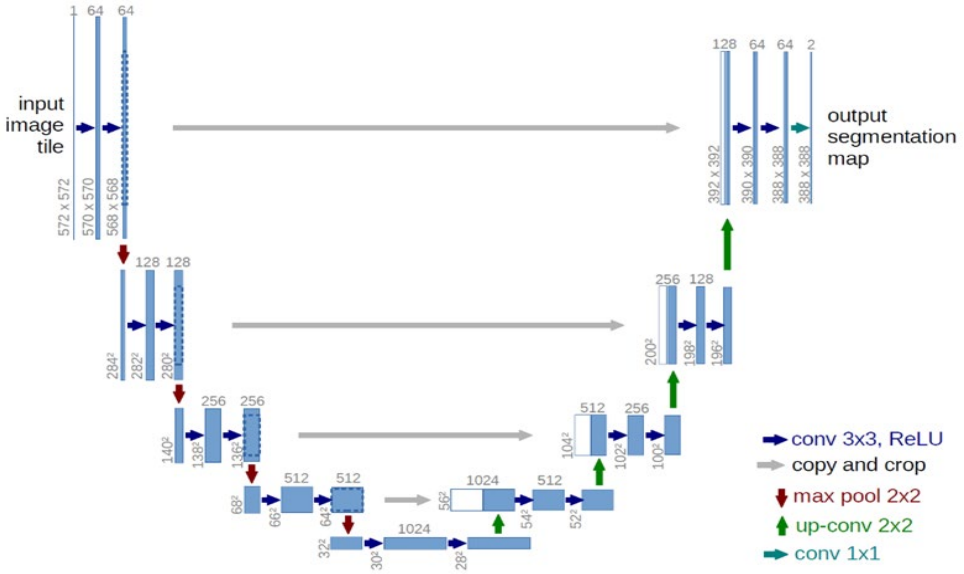


Figure 2.8 The architecture of U-net (Ronneberger et al., 2015)

Some researchers design specific network architecture for distress segmentation. Zhang et al. develop CrackNet for crack detection in 3D pavement images (Zhang et al., 2017). Unlike the FCNs, CrackNet deletes pooling layers so that spatial resolution could be preserved throughout the hidden layers. The model is tested on an annotated 3D pavement imaged dataset, and an overall F1 score of 0.89 is reached. It is worth noticing that because of the elimination of the pooling layers, CrackNet requires more computation than typical CNNs. In 2019, the same research team proposed CrackNet-R (Zhang et al., 2019), which is based on Recurrent Neural Network (RNN). The CrackNet-R outperforms CrackNet on both F1 score and inference speed. Beside that CrackNet-R yields better results than that of CrackNet, it is also noticed that both models underperform significantly for certain images. GAN is adopted for a self-supervised crack detection model (Zhang et al., 2020). This study demonstrates the capacity of DL models on performing crack segmentation without paired ground truth.

The performances of the aforementioned studies are listed in Table 2.5. It is noticed that most of these studies achieved good results with the datasets adopted in their experiments. However, as these studies adopted different datasets, it is impossible to directly compare the performances of these proposed methods. Besides, different datasets have significantly or slightly different protocols for distress annotation, and the models developed based on these datasets normally cannot be applied directly to a specific occasion. To develop a distress measurement model that accords with TxDOT's protocols, a specifically developed image dataset is needed.

Table 2.5 Summary of distress segmentation methods

Main frame	Reference	Method description	Data	Performance
Encoder-decoder structure	Dung and Anh, 2019	FCN with no feature fusion	Concrete (2D)	F1 = 0.89
	Yang et al., 2018	FCN with low-level feature fusion	Concrete (2D)	F1 = 0.80
	Bang et al., 2019		Mixed (2D)	F1 = 0.75
	Jenkins et al., 2018	U-net	Asphalt (2D)	F1 = 0.87
	Zou et al., 2018	U-net with multi-level loss calculation	Asphalt (2D)	F1 = 0.87
No pooling	Zhang et al., 2017	CNN with no pooling layer	Mixed(3D)	F1 = 0.89
RNN	Zhang et al., 2019	Recurrent neural network	Mixed(3D)	F1 = 0.92
Self-supervised learning	Zhang et al., 2020	Generative adversarial network	Asphalt (2D)	F1 = 0.91

To consistently evaluate the performance of the emerging models, some researchers apply 8 of the most recent DL models for crack segmentation, based on the same datasets, model training settings, and evaluation metrics (Hsieh and Tsai, 2020). The results are shown in Table 2.6. Results show different network structures perform differently over the same dataset, while deeper networks tend to yield better performance. An interesting observation is that the performance of most of the models in this study is significantly lower than that in the original studies where these models are brought up. It could be assumed that either the models are not trained under the optimized hyperparameter settings, or there is a wide range of variations between the performances of each model over different datasets. Either way, more work needs to be done for model evaluation.

Table 2.6 Segmentation performance of different models (Hsieh and Tsai, 2020)

Models	Precision	Recall	F1
FCN-VGG16	0.1292	0.8218	0.2199
FCN-VGG19	0.1364	0.8514	0.2313
FCN-ResNet	0.1905	0.8940	0.2989
DeepCrack	0.2007	0.8357	0.3115
U-Net	0.2732	0.8673	0.4010
CrackNetII	0.1143	0.6279	0.1856
GAN-U-Net	0.4025	0.4605	0.4032
GAN-ResNet	0.3948	0.4735	0.4071

2.3.2.2 Distress detection

Distress detection studies target two types of pavement images: top-down view image and rider's view image. Top-down view images are usually collected according to industry-level standards, using high-resolution cameras and/or laser-based sensors. These images are usually proprietary and are currently only available to a very small group of researchers. Eisenbach et al. offer one of the rare open-source industry-level top-down image datasets, which consists of 1,969 high-resolution gray pavement images (Eisenbach et al., 2017). This dataset annotates the distress with bounding boxes that have a size of lower than 64 x 64 pixels. This annotation is designed to suit patch-level classification instead of detection, thus did not become very popular for distress detection studies. The rider's view images are pavement images taken from an oblique angle, usually taken from the windshield of a moving vehicle using inexpensive cameras. The first open-source rider's view image dataset is proposed by Maeda et al., who collects 9,053 images of Japanese roadways with the smartphone mounted on the vehicle dashboard (Maeda et al., 2018). This dataset is further developed by adding datasets from two other countries (Arya et al., 2020). Another dataset is developed with Google Street View images (Majidifard et al., 2020). The cost of collecting rider's view images is much less than that of top-down view images. However, it still takes a lot of manual work for ground truth annotation. Although it was rarely discussed in prior literature, annotation quality can potentially affect detection performance.

Table 2.7 Datasets with pavement crack classification

Name	Image type	Number of classes	Image size (pixel*pixel)	Data size/Instance number
GAPs (Eisenbach et al., 2017)	Top-down view	6	1,920x1,280	1,969/-
RDD-2018 (Maeda et al., 2018)	Rider's view	8	600x600	9,053/15,435
RDD-2020 (Arya et al., 2020)	Rider's view	4	600x600	26,620/25,046
PID (Majidifard et al., 2020)	Rider's view	9	640x640	7,237/67,469
Du et al., 2020	Rider's view	7	-	45,788/-
Du and Jiao, 2022	Rider's view	4	640x640	5,600/-
Ghosh and Smadi, 2021	Top-down view	9	1,800x1,200	1,423/910

Distress detection combines the tasks of both distress localization and classification. Some research approach distress detection by manipulating classified patches to gain localization information (Cha et al., 2017; Chen et al., 2019; Li and Zhao, 2019; Wang and Hu, 2017). However, the localization results are always coarse, and it takes a long time to process one image. Other research incorporates state-of-the-art object detection methods developed for generic object detection, such as YOLO, SSD, RetinaNet, and Faster R-CN, aiming to get more accurate localization information and faster inferencing speed. This literature is summarized in Table 2.8. According to the literature, YOLO is the most widely used method for distress detection, which also generally yields equivalent or better performance than that of other methods. Another significant phenomenon is that the performance of the same detection methods varies a lot over different datasets, while different detection methods would also perform differently depending on the dataset.

Majidifard et al. test both YOLOv2 and Fast R-CNN on PID dataset. According to the result, YOLOv2 yields an overall F1 score of 0.84, indicating that DL-based algorithms have great potential in distress detection. Another similar study (Maeda et al., 2018) tests SSD on the RDD-2018 dataset, and the proposed model can yield the same level of accuracy. To facilitate the distress detection research, RDD-2018 and RDD-2020 datasets are used for global road damage detection competitions in 2018 and 2020, respectively. The best performance yielded on RDD-2020 dataset is an F1 score of 0.67, which is much lower than that of RDD-2018. This significant drop in performance can be attributed to the heterogeneous distress from different countries, but also it indicates that there is still space for improvement between distress detection algorithms and practical application. Ghosh and Smadi apply YOLO on a top-down view image dataset and achieve a F1 score of 0.96. However, this dataset is comparatively small, with an instance number of less than 1,000. According to the literature, building a comprehensive image dataset is the first and one of the steps for developing AI-based pavement condition assessment algorithms.

Table 2.8 Summary of distress detection methods.

Reference	Method	Dataset	Results
Mandal et al., 2018	YOLO	9,053 rider's view images taken by smartphone	F1=0.878
Maeda et al., 2018	SSD	9,053 rider's view images taken by smartphone	-
Carr et al., 2018	RetinaNet	118 grey scale images taken by smartphone	Precision=0.98
Nie and Wang, 2019	YOLO/Faster R-CNN/RetinaNet	4,200 grey scale images	AP= 0.51 (YOLO) AP= 0.53 (Faster R-CNN) AP=0.50 (RetinaNet)
Liu et al., 2020	YOLO	1,066 grey scale images taken by smartphone	F1=0.91
Majidifard et al., 2020	YOLO/Faster R-CNN	7,237 rider's view images extracted from Google Street View	F1= 0.84 (YOLO) F1= 0.65 (Faster R-CNN)
Du et al., 2020	YOLO	45,788 high-resolution images collected with a professional survey vehicle	Accuracy=0.74
Ghosh and Smadi, 2021	YOLO/Faster R-CNN	1,423 high-resolution 3D images collected with a professional survey vehicle	F1= 0.90 (YOLO) F1= 0.90 (Faster R-CNN)
Xiang et al., 2022	YOLO	26,620 rider's view images taken by smartphone	F1=0.67
Du and Jiao, 2022	YOLO	5,600 rider's view images extracted from Baidu Street View	F1= 0.7

Chapter 3 Data Preparation

The primary purpose of this chapter is to outline the objectives, methodology, and findings related to the creation of a library of pavement surface 2D/3D images in the AASHTO standard format. This library aims to serve as a resource for engineers and researchers who work with TxDOT to improve data quality and the technology of automated pavement condition data collection methods using pavement surface image data and image processing algorithms.

3.1 Background

3.1.1 Current Practice of Pavement Surface Condition Assessment

Pavement condition assessment is critical for pavement management which is required for managing the National Highway System under the Moving Ahead for Progress in the 21st Century (or MAP-21) and the Fixing America's Surface Transportation (or FAST) acts (Zimmerman, 2017). Walking and windshield surveys are two conventional pavement condition assessment methods that are intensively dependent on human labor. Considering the advantages of minimal impact on traffic, a significant increase in safety, more time efficiency, and the possibility of 100% network coverage, the automated pavement condition assessment has become a commonly acceptable data collection method in recent years (Pierce and Weitzel, 2019). As reported in a survey by the National Cooperative Highway Research Program (NCHRP), 45 out of 57 respondents had adopted automated data collection methods exclusively, six agencies used both manual and automated condition surveys, and only six agencies used manual pavement condition surveys by the time of March 2018 (Pierce and Weitzel, 2019). Currently, automated pavement condition surveys are conducted using specially designed vehicles to obtain 2D/3D pavement surface images and profile data (Pierce and Weitzel, 2019). The 2D/3D pavement surface images are used for pavement distress measurement using image processing techniques.

It is very important and yet difficult to extract distress information accurately and efficiently from pavement surface images of different pavement types and conditions. The main challenge of extracting distress information from digitized pavement images is to separate the distress features from the noisy and varying backgrounds. Distress measurement algorithms can be divided into rules-based algorithms and Machine Learning (ML) based algorithms. The rules-based algorithms are developed based on the explicit characteristics of pavement distress in digital images, such as the photometric variation and geometric variation of the pavement distress (Zakeri et al., 2017. Recently, researchers started to adopt Deep Learning (DL) methods in distress measurement tasks, after witnessing the great success DL made in Computer Vision (CV). Instead of using manual engineering for feature extraction from images, a DL method learns the feature extraction from the training data in an end-to-end manner, typically in the form of neural networks. Hundreds of studies have been conducted during the last five years for DL-based distress measurement. It was claimed that DL-based algorithms have achieved more robust and accurate distress segmentation compared with traditional methods.

3.1.2 The Need for an Image Library of Standard Format

The pavement condition data collected and stored by state highway agencies are critical to decision-making for pavement management of the state agencies and to fulfilling the objective of pavement performance reporting required by the FHWA. Pavement condition data enable state highway agencies to characterize network-level pavement conditions, predict future pavement conditions, and trigger pavement maintenance and rehabilitation actions. However, there is considerable variation in the ways that pavement condition survey vendors/agencies acquire, compress, store, transmit, analyze, and evaluate the 2D/3D pavement surface image data, while almost all of these data management methods are proprietary. Information stored in proprietary formats can be difficult to access, and ad-hoc formats increase software development costs and are not easily extended to widespread usage (Wang et al., 2016). A standard data format can help to: 1) reprocess historical pavement image data when new analysis algorithms are developed; 2) apply analysis algorithms to 2D/3D digital images from different sources; 3) support the structure of the AASHTO standards separating data collection from data analysis (similar to the longitudinal profile); 4) share data efficiently between uses, software tools, and electronic platforms; and 5) promote development and adoption of 2D/3D data collection technologies (Tsai et al., 2019). Therefore, a versatile and flexible standard format for 2D/3D image data could be very beneficial in achieving the desired objectives (AASHTO, 2023).

On the aspect of using AI/ML methods for pavement condition assessment, a comprehensive image library is also vital for the performance of the developed model. A comprehensive library ensures a wide variety of pavement conditions are covered, providing a rich dataset that can help AI models generalize better and work effectively across different scenarios. Eisenbach et al. offered one of the rare open-source industry-level top-down image datasets, which consisted of 1,969 high-resolution gray pavement images (Eisenbach et al., 2017). This dataset annotates the distress with bounding boxes that have a size lower than 64x64 pixels. This annotation is designed to suit patch-level classification instead of detection; thus, it is not suitable for network-level pavement condition assessment. Another research prepared a high-resolution 3D image dataset for a pavement distress detection study, with a total of 1,423 pavement surface images covering a variety of pavement types (*Ghosh and Smadi, 2021*). This dataset contains only 910 distress instances, which may not be sufficient given the complexity of the pavement types and distress classes. No established standard format 2D/3D pavement surface image dataset for training AI/ML models has been made publicly available as of the time this chapter was prepared.

3.1.3 Objectives

The ongoing development and improvement of the image library are guided by specific objectives outlined below, categorized into three key aspects: Data Collection, Data Annotation, and Library Architecture. These objectives are geared towards creating a comprehensive library of standard format 2D/3D pavement surface images to facilitate AI-based pavement condition assessment.

Data Collection. The aim is to collect a diversified and high-quality set of standard format 2D/3D pavement surface images. This includes collecting images with different distress classes and severities. The images should ideally come from a variety of pavement types, distress types, and distress severity levels to make a comprehensive library.

Data Annotation. The collected images will be annotated to identify various types of cracking distresses, such as longitudinal cracking, transverse cracking, and other pavement distress classes defined by TxDOT (TxDOT, 2023). In addition, the annotation should be performed according to certain guidelines to ensure consistency and accuracy. The new AASHTO standard "Crack Annotation and Crack Length and Width Computation on 2D/3D Pavement Images" was followed in this study (AASHTO, 2023). The focus here is to provide high-quality annotations that can be useful for machine learning algorithms, particularly deep learning methods.

Library Architecture. Designing the structure and format of the image library is crucial. Given that this library is intended for use in deep learning applications, the architecture should be designed to facilitate easy data retrieval, updating, and scalability. It should support AASHTO standard format images to ensure wide compatibility and easy usage.

3.2. Methodology

3.2.1 Data Acquisition

The standard format images utilized in this research are provided by TxDOT through a contractual agreement with Pathway Services Inc., which is responsible for taking highway pavement surface image data of selected counties. The images in the standard format have the .psi suffix and will be referred to as PSI files throughout the remainder of this chapter. Every year, Pathway Services Inc. provide standard format images of 500-mile highway sections. The research team will select from the annually provided raw pavement images to develop a comprehensive image library.

The PSI files utilized in this research follow the standard format defined by AASHTO: File Format of 2-Dimensional and 3-Dimensional (2D/3D) Pavement Image Data (6). Each image comprises six sequential sections: (1) File Signature, (2) File Header, (3) 2D Image Data (intensity), (4) 3D Image Data (Range), (5) User Defined Metadata, and (6) File Trailer. The File Signature identifies a file defined by this specification. The File Header describes the properties of the 2D/3D data stored in the file. Each property is denoted by a variable of defined data type and byte length. For example, the GPS Longitude is required to be denoted by an 8-byte float type. The 2D Image Data and 3D Image Data sections store blocks of compressed or uncompressed binary data. The standard image representation is shown in Figure 3.1.

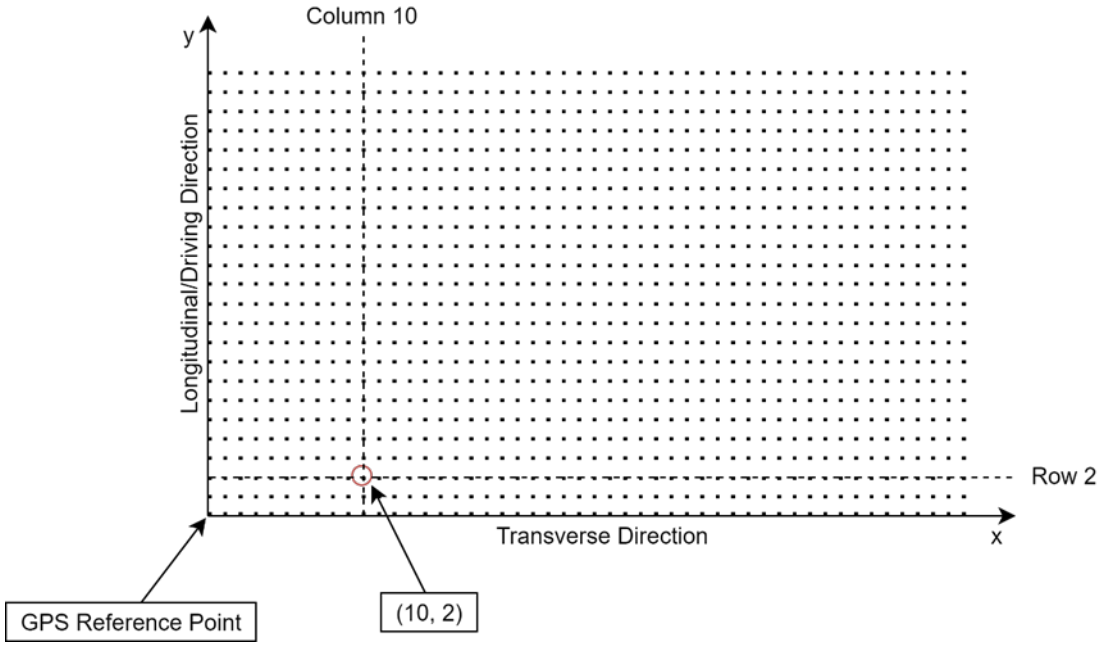


Figure 3.1 Graph of standard image representation (Ghosh and Smadi, 2021)

3.2.2 Data Preprocessing

Customized PSI file review software is developed to assist PSI file reading and preprocessing. The main objective of the data preprocessing phase is to extract relevant 2D/3D pavement surface images accurately and efficiently from existing standard format image data files. This ensures that the extracted images are prepared for AI-based pavement condition assessment.

This critical phase involves a series of steps aimed at parsing and exporting the necessary 2D/3D images from the standard format data files:

File Parsing: The header file of the PSI file contains metadata that allows for the identification and extraction of the 2D/3D image data. Extraction of 2D/3D image data is correctly processed by reading the header files associated with the standard format image data files.

Data Decompression: If the standard format files are compressed, they will be decompressed to enable image extraction.

Image Tagging: Tagging or labeling each image to make the image easily locatable for future reference and analysis.

Export Image: Once the images have been tagged, they will be exported to a dedicated folder or database for further processing and analysis. Metadata for each image, such as the location of capture and resolution of the images, will be associated during the export process.

3.2.3 Image Annotation

The objective of the image annotation phase is to meticulously label and classify the 2D/3D pavement surface images. The labels are crucial for training and validating deep learning models that specialize in assessing pavement conditions. The annotation of the image data follows the procedure described below.

Preliminary Inspection: Before beginning the annotation, the research team manually inspects the image dataset to understand the diversity and prevalence of specific pavement conditions like pavement types and distress classes. This initial evaluation helps in formulating an annotation strategy, including deciding on the types of labels that will be necessary.

Annotation Tools: The selection of annotation tools is the next crucial step. The chosen software or platform must allow for detailed and efficient image annotation, and it must be compatible with the data formats used in the project. The pavement distress bounding box annotation tool is developed to fit the specific annotation needs of the project.

Label Definition: Based on the pavement distress definition of the TxDOT Pavement Rater's Manual (TxDOT, 2023), the project develops a modified set of distress classes specially adapted for computer vision applications. The manual provides a valuable baseline for classifying and understanding different pavement distress classes. However, the requirements for computer vision tasks, particularly in deep learning models, often necessitate a more nuanced or granular approach to labeling. Therefore, labels may be subdivided or merged to create distress classes that are more easily distinguishable by machine learning algorithms. Special attention is paid to ensuring that these labels can be operationalized within a computer vision context. For example, labels may be defined in such a way as to make it easier for a machine to identify boundaries, contrasts, and features within the pavement images.

Annotation Guidelines: To ensure uniformity and precision, detailed guidelines for annotation are developed. These guidelines include instructions on how to apply labels, under what conditions, and how to deal with edge cases or ambiguities. The guidelines are shared and explained to all annotators to ensure compliance and consistency.

Manual Annotation: The process involves two specific types of annotation: bounding box annotation and pixel-level segmentation annotation. Bounding box annotation aims to identify and locate areas of interest in the image, such as specific types of pavement distress. Annotators draw rectangular boxes around areas of interest according to the guidelines. Each box is tagged with the appropriate label from the label reference guide. Special guidelines are provided for handling overlapping or clustered areas that may require multiple bounding boxes. Random samples of bounding box annotations are reviewed by senior annotators or domain experts to ensure accuracy and consistency. Pixel-level segmentation aims to provide a finely detailed classification of pavement distress. Unlike bounding box annotation, this method involves the classification of individual pixels and is crucial for pavement condition assessment requiring highly detailed information. Annotators begin by creating a new transparent layer over the existing pavement image using Photoshop. On this newly created layer, annotators proceed to

color-code individual pixels that correspond to specific types of distresses. The layer is transparent, allowing annotators to see the original image underneath. This facilitates more accurate annotation by providing context. This is a more labor-intensive but highly precise form of annotation.

Quality Checks: The aim of quality checks is to ensure the highest level of accuracy, consistency, and reliability in pixel-level annotations. These checks are crucial for maintaining the integrity of the dataset. Before engaging in the full-scale annotation task, annotators are required to annotate a smaller, test batch of images. These test annotations are reviewed by senior annotators or domain experts. Feedback is given, and adjustments are made to the annotation process if needed. During the annotation process, portions of the annotated images are randomly selected for quality review. The review focuses on the correct application of labels and adherence to annotation guidelines.

Annotation Export: Once the annotation is complete and verified, the annotations are exported in a machine-readable format. The bounding box information is stored in TXT files, while all segmentation annotations are stored as PNG files. These files are designed to be directly importable into the deep learning frameworks used in the project, and each file associates an image with its corresponding labels and other relevant metadata.

3.2.4 Library Construction

The methodology for this study pivots around the utilization of the Standard Format 2D/3D Pavement Surface Image Library, a sophisticated database developed to aid in the collection, storage, and analysis of pavement distress information. The library is designed with a user-friendly interface featuring three main components: the Distress Database, PSI File Reviewer, and PSI File Query. Below is a detailed description of each of the components, their functions and features, and how they are used for this research.

Distress Database: The Distress Database serves as a comprehensive repository for systematically cataloging, storing, and managing vital information related to pavement distress. Various attributes, as outlined in Table 3.1, contribute to a holistic understanding of each recorded pavement distress instance.

PSI File Reviewer: The PSI File Reviewer is engineered to extract and visualize 2D and 3D image data of pavement surfaces. The component processes metadata and parameters stored in the head file and also provides annotation information for each PSI file.

PSI File Query: The PSI File Query is integrated into the default dashboard and aims to facilitate the quick retrieval of specific pavement images based on types of distresses. The interface allows keyword-based and attribute-filtered searches through drop-down menus.

Table 3.1 Attributes of the distress table

Attribute title	Definition	Utility and Functionality
Section name	Alphanumeric descriptor to identify the specific road or pavement segment	Allows for pinpointing the distress to an exact road segment
Image name	Reference to the image file capturing the distress	Allows for fast retrieval and visual assessment of the particular distress
Pavement type	Classification of the pavement material	Provides context for distress-related information
Distress code	Standardized code to categorize the distress class	Facilitates quick identification and analysis
Distress class	Types of distress	Offers categorization for different distress classes
Bounding box	Spatial details referring to a single image	Provides spatial context for each distress instance

3.3. Construction of Library

The completion of the standard format image library unfolds in two crucial stages: (1) the Initialization Stage, and (2) the Enhancement Stage. During the Initialization Stage, a preliminary library is constructed using a varied assortment of pavement sections. This foundational library serves as the cornerstone for the development of a more expensive and comprehensive pavement image repository. In the Enhancement Stage, additional pavement sections are strategically chosen to address the gaps or deficiencies identified in the initial library. This chapter primarily concentrates on the establishment and completion of this foundational library.

3.3.1 Raw data retrieval

A cumulative total of 149,191 standard format images, covering a distance of 530 miles, have been delivered by TxDOT at two separate times. The geographical distribution of the collected PSI files is shown in Figure 3.2, and the number of files from each county is listed in Table 3.2. The first batch of data contains PSI files collected in 2022 and cover six counties in Texas and are used for the development of the foundational library. The second batch contains PSI files collected in Jefferson County in 2023.

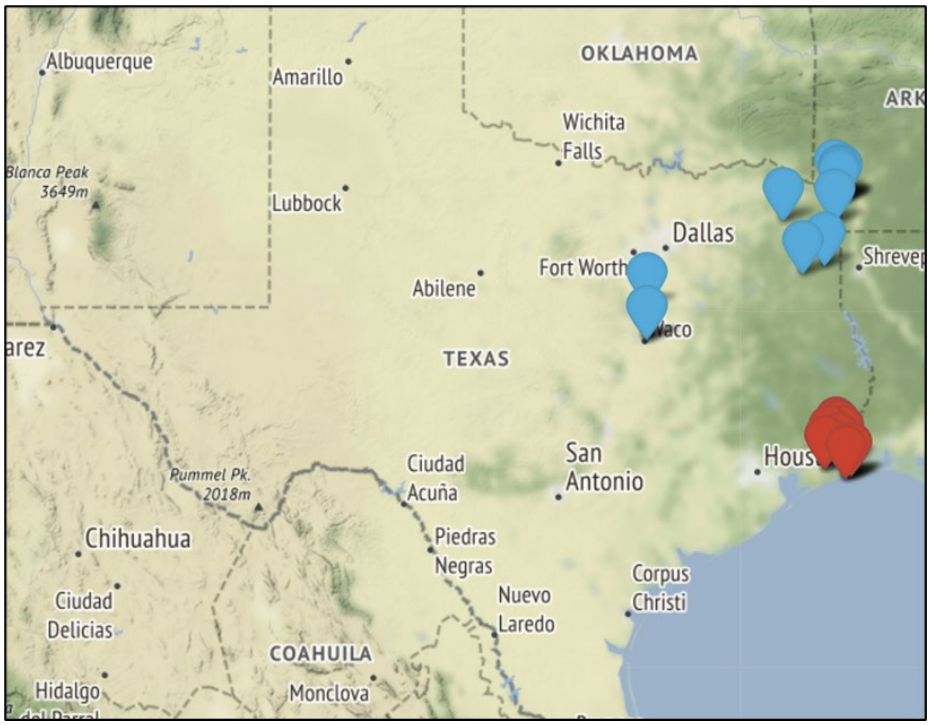


Figure 3.2 Geographical distribution of collected PSI files (blue marks indicating locations of the first batch data, red marks indicating locations of the second batch data)

Table 3.2 Geographical distribution of collected PSI files

County	Collection year	Number of images
Bowie	2022	9,557
McLennan	2022	7,501
Harrison	2022	1,796
Hill	2022	1,222
Titus	2022	1,726
Cass	2022	1,217
Jefferson	2023	126,145

Based on initial research, Jefferson County has been selected for library enhancement because it offers representative samples across ACP, JCP, and CRCP. Figure 3.3 shows the distribution of each pavement type of dataset collected in Jefferson. The most prevalent type of pavement in Jefferson County is ACP, with a total length of 334.578 miles. On the other end of the spectrum, CRCP has the shortest total length, at just 32.714 miles.

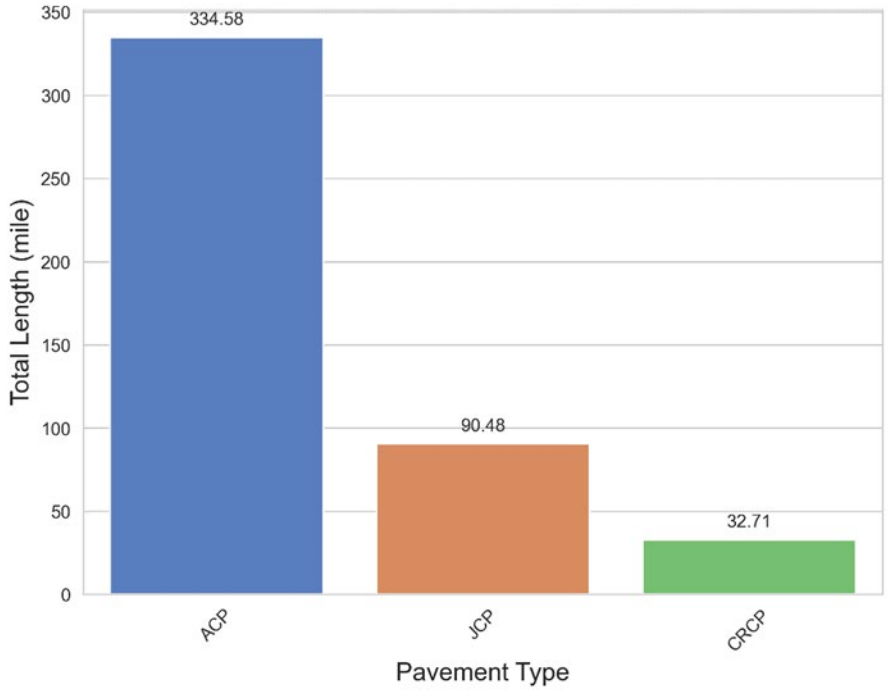


Figure 3.3 Total lengths of pavement types of Jefferson County dataset

To further understand the representations of pavement surface distresses of the Jefferson County dataset, specific distress quantities for each pavement type were extracted from PMIS as shown in Tables 3.3 to 3.5. For the ACP dataset in Jefferson County, there are rich representations of alligator cracking, longitudinal cracking, and transverse cracking, respectively. Failure is underrepresented and block cracking is entirely missing. For the JCP dataset, there is also a rich representation of each distress class, except for the shattered slabs. However, many distress classes need further annotations to distinguish the sub-types. For example, the distress classes of JCP failure include seven sub-types, which must be well-represented individually too. For the CRCP dataset, spalled cracking, punchouts, and ACP patches are significantly underrepresented. PCC patches are the most abundant, but even this data group might not have enough data for robust model training. Overall, the Jefferson County dataset can potentially serve as a good representation for ACP and JCP, but not for CRCP. More CRCP data may need to be collected in the future.

Table 3.3 Distress statistics of ACP in Jefferson County

Total Patching length	2.76 mile
Total Failures	96
Total Block Cracking length	0
Total Alligator Cracking length	2.99 mile
Total Longitudinal Cracking	14.16 mile
Total Transverse Cracking Quantity	398

Table 3.4 Distress statistics of JCP in Jefferson County

Failed Joints and Cracks quantity	390
Failures quantity	1353
Shattered Slabs quantity	27
Slabs with Longitudinal Cracking quantity	304
PCC Patches quantity	895
Average apparent joint space	19.79
Failed Joints and Cracks quantity	390

Table 3.5 Distress statistics of CRCP in Jefferson County

Spalled Cracking quantity	29
Punchout quantity	13
ACP Patches quantity	5
PCC Patches quantity	106

3.3.2 Annotation process

3.3.2.1 Bounding box annotation

Some refinements and subcategories are introduced based on the distress definitions in the TxDOT Pavement Rater's Manual (TxDOT, 2023), as shown in Table 3.6.

Table 3.6 Modifications of distress classification (ACP)

No.	Definition (TxDOT)	Modified distress classes
1	Transverse Cracking	Transverse cracking
2		Sealed transverse cracking
3	-	Joint
4	Longitudinal Cracking	Longitudinal cracking
5		Sealed longitudinal cracking
6	-	Lane longitudinal cracking
7	Block Cracking	Block cracking
8	Alligator cracking	Alligator cracking
9	Failures	Potholes

The modifications aim to have more accurate classifications in the context of computer vision. For transverse cracking, the modified distress classes now differentiate between Transverse

Cracking and Sealed Transverse Cracking. This could be useful for a DL model to understand both the original and post-repair conditions of the road. For Longitudinal Cracking, similarly, a distinction is made between Longitudinal Cracking and Sealed Longitudinal Cracking. Again, this differentiation can help a DL model become more nuanced in its predictions and understanding of road conditions. The definitions of Block Cracking and Alligator Cracking remain unchanged, as the conditions are not close enough in the observed dataset to warrant further recognition. Failures are replaced with Potholes, which would add specificity to DL models and could be an improvement. Considering most of the potholes had already been patched when the images were being collected, pothole patches are also considered as potholes. Non-distress classes, such as Joint, Lane Longitudinal Cracking are added due to their similar features to other distress classes. The inclusion of these non-distress classes can help the model identify what is not a sign of a pavement distress, thus reducing false positives.

Table 3.7 Modifications of distress classification (JCP)

No.	Definition (TxDOT)	Annotation
1	Failed joints and cracks	Failed joints and cracks
2	Shattered slabs/Failures	Corner break
3		Punchout
4		Asphalt patch
5		Failed concrete patch
6		D-cracking
7		Spall
8		Popout
9	Slabs with longitudinal cracking	Longitudinal crack
10		Sealed longitudinal
11	Concrete patch	Concrete patch
12	Apparent joint spacing	Transverse crack
13		Joint crack
14		Sealed transverse crack

Each distress class can be identified by certain characteristics specific to each type of crack. During the annotation process, each distress class of JCP is identified based on the following descriptions: Transverse cracks run horizontally/perpendicular across the road and have not been treated with sealant or filling. Transverse cracks can be quite straight or more crooked. Untreated Transverse cracks will appear as a dark line on the 2D/ 3D images but can appear lighter if the damage is not extreme. Joint cracks also run horizontally across the road, but unlike Transverse cracks, they are exclusively straight lines. Joints can be filled with fill/sealant if needed. During annotation, joints were usually found on the road's far left/right side. Sealed transverse cracks run horizontally across the road and have been treated with a fill/sealant. Usually, sealed transverse cracks have a distinctly visible outline in the 2D/3D images from the sealant used. Longitudinal cracks run vertically/parallel down the road and have not been treated with sealant or filling. Longitudinal cracks can be quite straight or more crooked. Untreated Longitudinal cracks will appear as a dark line on the 2D/3D images but can appear lighter if the damage is not

extreme. Lane Longitudinal cracks also run vertically down the road, but unlike Longitudinal cracks, they are extremely straight, not crooked, and can be filled with fill/sealant if needed.

Sealed longitudinal cracks run vertically across the road and have been treated with a fill/sealant. Usually, sealed longitudinal cracks have a distinctly visible outline in the 2D/3D images from the sealant used. Block cracking is characterized by the intersection of longitudinal and transverse cracking to form “blocks”. These blocks can range in size anywhere from 1.0x1.0 ft to 10.0x10.0 ft. Block cracking does not always appear as perfect blocks but instead can be an oblong rectangle. Like the name, alligator cracking appears similar to the scales on an alligator's back. Alligator cracking can look similar to block cracking, but alligator cracking/bunching cannot exceed 1.0x1.0 ft per division. Potholes are characterized by large circular divots/holes in the roadbed. Usually, a large dark spot will be clear on the 3D image when detecting potholes.

Table 3.7 shows the definition modifications of JCP distress classes based on the TxDOT Pavement Rater's Manual (TxDOT, 2023). Similar to the modified classifications for ACP, this table adds nuance and subcategories to standard distress classes. For Failed joints and cracks, the original definition remains intact, as the anticipated DL models aim to capture the broader category of both failed joints and cracks in JCP. For Shattered Slabs/Failures, both categories are broken down into multiple specific types like Corner Break, Punchout, Asphalt Patch, Failed Concrete Patch, D-Cracking, Spall, and Popout. This variety allows for a much more nuanced understanding and classification by the model, aiding in both identification and possible remediation strategies. Slabs with Longitudinal Cracking are split into Longitudinal Cracking and Sealed Longitudinal Cracking, similar to how ACP categories were divided. This can help the model understand the initial and post-repair conditions of the road. Concrete Patch remains a single category as Concrete Patch, due to its straightforward nature. Apparent Joint Spacing is replaced with Transverse Cracking, Joint Crack, and Sealed Transverse Cracking, as all three types of cracking are to be used to calculate the value of Apparent Joint Spacing.

During the annotation process, each distress class of JCP is identified based on the following descriptions: Failed Joint, which encompasses multiple problems within a joint arising from localized pavement failures like D-cracking, extreme spalling, and pop-outs; Corner Break, denoting cracks at the juncture of longitudinal and transverse joints, distinguished by considering crack size; Punchout, akin to pop-outs but occurring at the crossroads of transverse and longitudinal cracks or joints; Asphalt Patch, representing pavement patches using asphalt, annotated based on color and absence of utility cuts or defined edges; Failed Concrete Patch / Concrete Patch, describing concrete patches indicating utility cuts or lighter color, showcasing other distress classes within the patch; D-cracking, characterized by "wrinkles" along transverse joints in JCP, annotated for even minimal amounts of cracking; Spall, linked with longitudinal or transverse cracking, annotated with minimal image distortion; Popout, annotated with bounding boxes around visible pop-outs, minimizing box size for accuracy; Longitudinal Crack, evident cracks parallel to lane marking lines; Sealed Longitudinal Crack, sealed parallel cracks indicating previous spalling; Transverse Crack, horizontal cracks sometimes resembling joint cracks, annotated for non-straight lines and other distress forms; Joint Crack, clean cracks perpendicular to lane marking lines mimicking natural joints.

Table 3.8 Modifications of distress classification (CRCP)

No.	Definition (TxDOT)	Modified distress classes
1	For Average Crack Spacing calculation	Transverse cracking
2		Sealed transverse cracking
3	Spalled cracks	Spalled transverse cracking
4	Longitudinal Cracking	Longitudinal cracking
5		Sealed longitudinal cracking
6	Punchout	Punchout
7	Asphalt patch	Asphalt patch
8	Concrete patch	Concrete patch

Table 3.8 shows the definitions for the CRCP distresses. As a concrete pavement, the CRCP is similar to JCP except for the joints on the latter. Therefore, the annotation for the CRCP follow the same rule of the JCP.

3.3.2.2 Distress segmentation annotation

Distress segmentation annotation is performed using the commercial software Photoshop. The process involves overlaying both 2D and 3D images onto separate layers within a single canvas. A new transparent layer is created above these images to facilitate the annotation process. Subsequently, crack pixels are accurately delineated on the transparent layer, with careful consideration of visual cues from both the 2D and 3D images. The Brush Tool is employed to mark crack areas, and adjustments are made to match crack widths. Fine-tuning involves adjusting layer opacity to align with crack characteristics. Regular comparison between 2D and 3D images ensures consistent annotation. After verification, the original image layers are concealed, leaving the annotated transparent layer visible. The final step involves exporting the annotated layer as a PNG file, serving as mask annotation, where crack pixels are depicted in black against a white or transparent background.

3.3.3 Annotation analysis

A total of 19,418 images for the bounding box and a total of 229 images for segmentation have been annotated. A summary of the annotation for each pavement type is listed in Table 3.9. In the Detection category, which involves annotations with bounding boxes and distress classification, there are 5,892 images available for ACP, while there are 8,066 and 5,776 images for JCP and CRCP, respectively. In the Segmentation category, which involves more granular, pixel-level annotations, ACP has 114 images, JCP has 53, and CRCP has 62. This data is especially useful for tasks that require more precise measurement of pavement distress.

Table 3.9 Annotation summary

Annotation type	Pavement Type	Number of images
Detection (Bounding box & distress class)	ACP	5,892
	JCP	7,750
	CRCP	5,776
Segmentation (Pixel-level mask)	ACP	114
	JCP	53
	CRCP	62

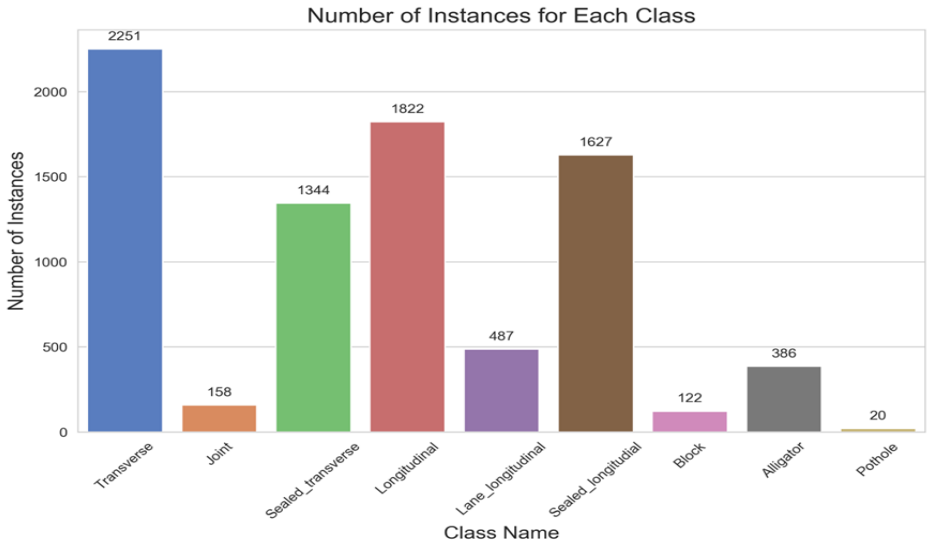


Figure 3.4 Distribution of numbers of distress classes of ACP dataset

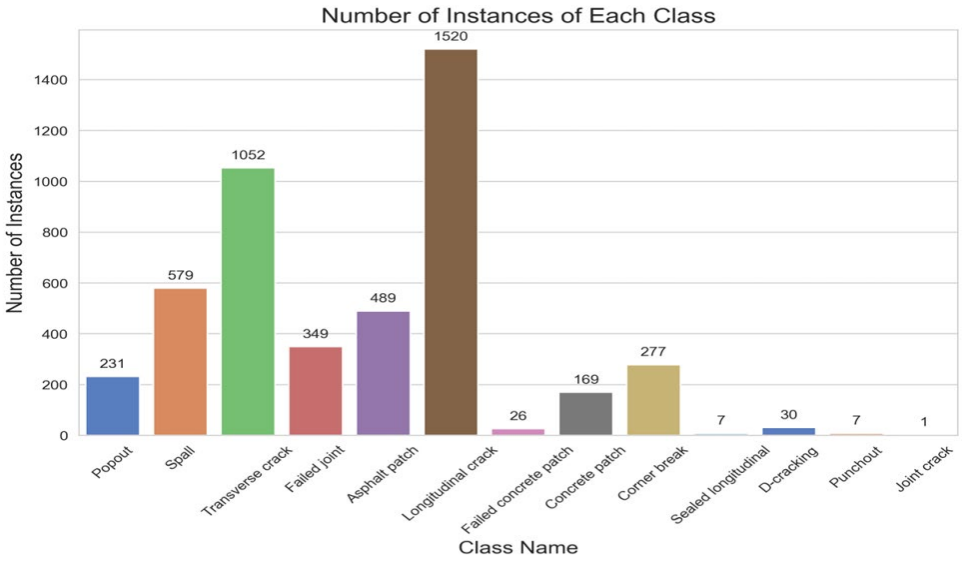


Figure 3.5 Distribution of numbers of distress classes of JCP dataset

Figure 3.5 shows the distribution of the number of distress classes of the JCP dataset. According to the figure, the JCP dataset presents a significant class imbalance that must be carefully managed in the deep learning training process. High-frequency classes like Longitudinal Cracking with 1,520 instances and Transverse Cracking with 1,052 instances could disproportionately influence the model's learning, potentially causing it to be biased towards these classes. Mid-frequency classes such as Spall with 579 instances, Asphalt Patch with 489, and Failed Joint with 349 instances offer a moderate level of representation, reducing the risk of model bias toward these categories but still providing ample data for effective learning. Low-frequency classes like Popout with 231 instances, Concrete Patch with 169, and Corner Break with 277 could potentially be underrepresented, affecting the model's ability to generalize well for these classes. Extremely rare classes like Failed Concrete Patch, D-cracking, Sealed Longitudinal Cracking, Punchout, and Joint Crack, with instances ranging from just 1 to 30, are at the highest risk of being poorly represented, posing challenges for achieving reliable performance for these classes.

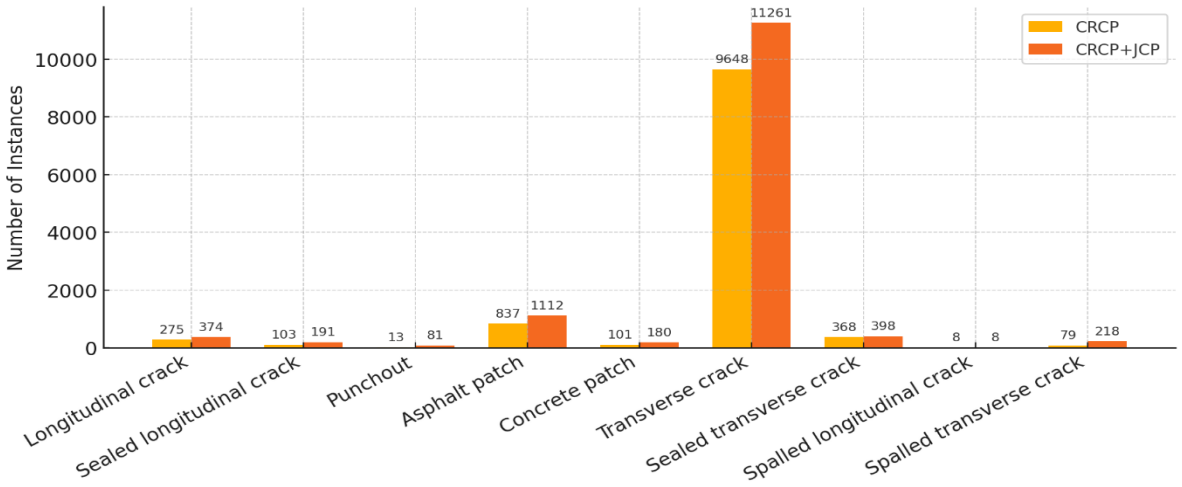


Figure 3.6 Distribution of numbers of distress classes of JCP dataset

Figure 3.6 indicates the distribution of number of distress classes of the CRCP dataset and combined dataset of JCP and CRCP. The latter aims to include JCP data to enhance underrepresented categories like punchouts and spalled transverse cracks. High-frequency classes like Transverse Cracking with 11,261 instances could disproportionately influence the model's learning, potentially causing it to be biased towards these classes. Mid-frequency classes such as Longitudinal cracks with 374 instances, Asphalt patch with 1,112, and Sealed longitudinal crack with 398 instances are at a moderately representative level. On the other hand, Punchout and Spalled transverse crack are low-frequency classes.

3.3.4 Discussion

3.3.4.1. Accuracy of the annotation

To analyze the impact of subjectivity on the annotation, two inspectors were asked to work on the same pavement section, performing the annotation of that section independently. The

inconsistency of the two annotation sets was measured by the precision score, using one set as the ‘ground truth’ and the other as the ‘prediction’. The IoU threshold was set to 0.5, above which the bounding box was considered consistent across two annotation sets. The result is shown in Figure 3.7. According to the figure, pavement distress classes exhibit varying levels of consistency between two sets of annotations. Classes such as Joint and Alligator demonstrate extremely high consistency rates of 96.30% and 97.37%, respectively, signifying strong agreement between the sets and potentially high reliability for maintenance planning. Conversely, Lane Longitudinal Cracking and Sealed Longitudinal Cracking show considerably lower consistency rates of 44.00% and 37.00%, indicating discrepancies that may necessitate further investigation. Classes like Transverse Cracking, Longitudinal Cracking, and Pothole fall in the moderate range of consistency, with rates between 51.14% and 57.14%, suggesting a moderate level of agreement. Other notable observations include the Block Cracking and Sealed Transverse Cracking classes, which also have relatively high consistency rates of 73.08% and 93.96%, further enriching the landscape of the dataset's reliability.

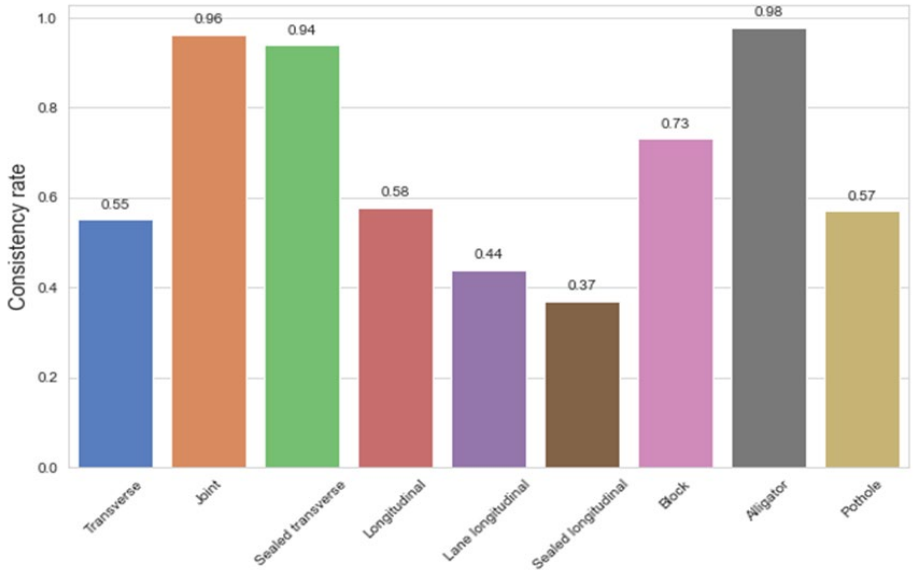


Figure 3.7 The consistency rates of two ACP annotation sets independently developed by two inspectors

The same experiment was conducted with JCP annotation, and the result is shown in Figure 3.8. Like the ACP annotation consistency experiment, the JCP experiment also reveals a wide range of consistency rates across different pavement distress classes. Notably, Longitudinal Cracking and Sealed Transverse Cracking demonstrate extremely high consistency with rates of 98.85% and 100%, respectively, indicating strong agreement between annotation sets and high reliability for maintenance actions. D-cracking and Sealed Longitudinal Cracking continue to exhibit low consistency with rates of 37% and 33.33%, respectively. Meanwhile, Failed Joint, Corner Break, and Asphalt Patch fall into the high-consistency category with rates above 80%, and classes like Spall, Popout, and Concrete Patch show moderate consistency, ranging between 61.54% and 76.92%. The zero-consistency rate of Punchout and Joint Crack indicates that there are no such distress classes annotated in the dataset.

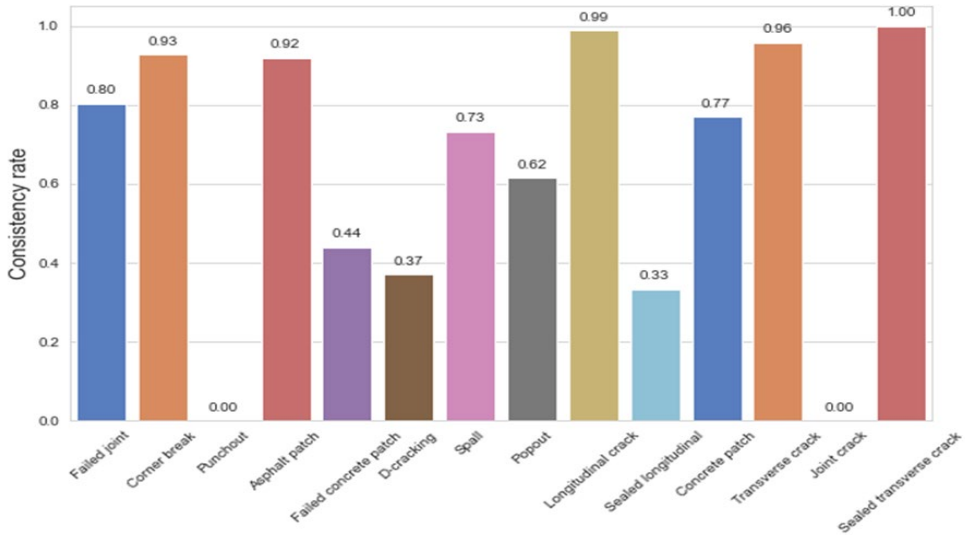


Figure 3.8 The consistency rate of two JCP annotation sets independently developed by two inspectors

High levels of annotation consistency suggest that the labeled data is both objective and reliable, often leading to improved model accuracy. In contrast, low annotation consistency indicates that the labeling is more subjective and less dependable. A model trained on such inconsistent data may excel on the training set but underperform on new, unseen data. This inconsistency may arise from the ways distress is represented in 2D or 3D digital images, among other factors. Moving forward, it will be important to closely examine the impact of annotation consistency on the performance of machine learning models.

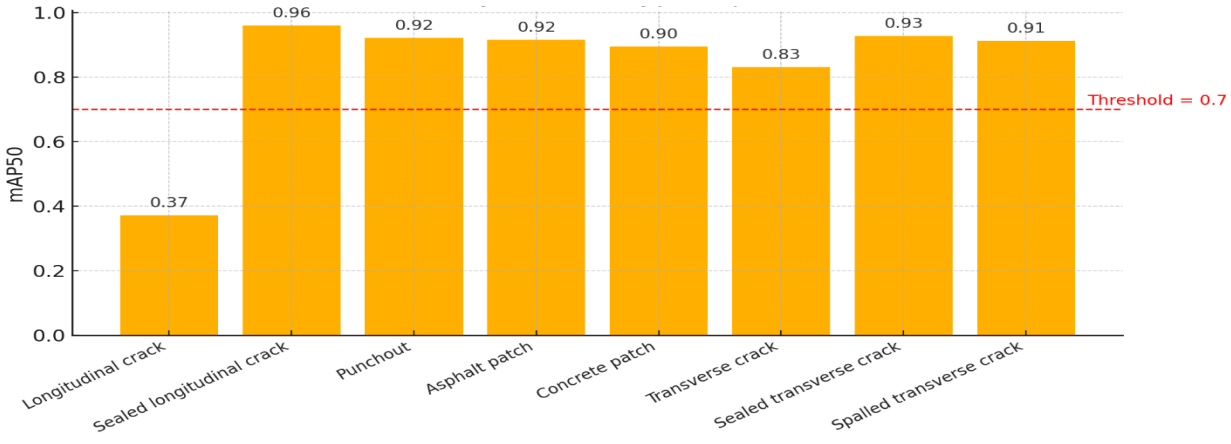


Figure 3.9 The consistency rate of two CRCP annotation sets independently developed by two inspectors

Figure 3.9 shows the performance of an AI model on CRCP dataset prepared in Section 3.3.3. The metric is mAP50, which is more comprehensive but considering the model's precision and recall across multiple thresholds. If the threshold is set as 0.7, all the classes except the Longitudinal crack have performance over this value on mAP50. But the longitudinal crack still needs improvement.

3.3.4.2. Underrepresented distress classes

It is noticed that there is a significant imbalance in the distribution of distress classes represented in three datasets. The model is more likely to predict high-frequency classes like Longitudinal Cracking and Transverse Cracking because of their greater impact on the loss function during training. This could lead to high false-positive rates for these classes. Distress classes with fewer instances may not provide the model with enough context to learn the nuanced differences between them and other more frequent classes. As a result, the model might underperform in identifying these low-frequency classes in new, unseen data. The imbalance in class distribution can also lead to overfitting, particularly for the high-frequency classes, where the model learns the training data too well but fails to generalize to new data. Standard evaluation metrics like accuracy can be misleading. For instance, a model that always predicts Longitudinal Cracking would still perform fairly well based on accuracy, even though it fails to identify any other class.

To improve the dataset for training a more robust pavement distress detection model, several steps can be taken. In the aspect of enhancing the image library, additional data for underrepresented classes should be collected. Synthetic data generation methods like GANs could be explored for extremely rare classes.

3.4. Summary

This chapter employs a detailed methodology for acquiring and annotating pavement images with the aim of AI-based condition assessment. The completion of the standard format image library consists of two crucial stages: (1) the Initialization Stage, and (2) the Enhancement Stage. In Task 3, the fundamental image library was developed for the Initialization Stage using the PSI files randomly selected. A total of 19,418 images for the bounding box and a total of 229 images for segmentation have been annotated. In the Detection category, which involves annotations with bounding boxes and distress classification, there are 5,892, 7,750, and 5,776 images available for ACP, JCP, and CRCP, respectively. In the Segmentation category, which involves more granular, pixel-level annotations, ACP has 114 images, JCP has 53, and CRCP has 62.

.

Chapter 4 Rules-based Automated Methods

The primary purpose of this chapter is to outline the objectives, methodology, and findings related to the evaluation of the existing methods of image-based pavement condition assessment. More specifically, this chapter focuses on using rules-based methods for pavement distress measurement to explore the capabilities and limitations of the current practice.

4.1. Objectives

In the realm of pavement crack detection, the period before the widespread application of artificial intelligence (AI) and machine learning (ML) technologies was marked by the development and use of conventional digital image processing techniques, which are also known as rules-based methods. The distress identification based on rules-based methods focuses on separating the crack pixels from the pavement background. Thresholding and edge detection are the two main methods for crack segmentation. Thresholding is a classic approach in image segmentation, which converts a grayscale image into a binary image based on the intensity threshold (Zhu et al., 2007). The basic assumption is that crack pixels are relatively darker than other pixels in grayscale digital images, thus cracks can be separated from the background by setting a proper intensity threshold. A dynamic threshold was developed to deal with distinctive mean pixel intensities of different images (Oliveira and Correia, 2009). However, it fails to cope well with images with non-uniform illumination. Edge detection approaches were adopted for crack detection, combined with other image processing methods to improve the measurement performance. Ayenu-Prah and Attoh-Okine combined bi-dimensional empirical mode decomposition with Sobel edge detection to remove noise in the pavement images (Ayenu-Prah and Attoh-Okine, 2008). Wang et al. (2007) applied Wavelet Transform to decompose the original image into different subsamples to capture the details of the cracks on different scales. However, both methods are unable to generate complete crack profiles. Seed-based crack detection was implemented for real-time crack detection (Huang and Xu, 2006; Zhou et al., 2016). This method divides the image into cells of 8x8 pixels and then decides whether the cells belong to crack or non-crack based on the contrast pixels. This method works very fast, but it is hard to find universal thresholds for images of dissimilar contrast. A dynamic optimization-based method was developed to utilize global information to estimate the probability of crack existence (Tsai et al, 2010). This method formulates the problem as an optimization task with 4 primary parameters, which entails a long processing time.

In this chapter, the main image processing techniques are adopted to explore the capabilities and limitations of the current practice of pavement condition evaluation. The specific objectives are to:

- Explore the main challenges that impose difficulties on distress identification with digital images.
- Explore the capabilities and limitations of rules-based distress identification methods.

4.2. Methodology

The main goal of the experiment is to evaluate the performance of four rules-based image processing techniques in accurately segmenting pavement distresses across three different pavement types: Asphalt Concrete Pavement (ACP), Jointed Concrete Pavement (JCP), and Continuously Reinforced Concrete Pavement (CRCP). For each pavement type, both 2D/3D image data and corresponding annotation are to be extracted from the image library built in Task 3. Python will be used as the main coding language to perform this task. The segmentation result of each image processing method will be compared with the annotation to evaluate the performance.

4.2.1 Image processing steps

In this study, the workflow of distress segmentation is organized into three phases: preprocessing, image processing, and postprocessing.

In the **preprocessing phase**, the aim is to prepare the images for analysis by improving their overall quality. This involves reducing image noise through filters like Gaussian blur and median blur, which helps minimize distractions without significantly affecting the details. In this study, the median blur filter is applied, which operates by replacing each pixel's value in an image with the median value of the intensity levels in the neighborhood of that pixel. Image enhancement techniques, such as histogram equalization or contrast stretching, are applied to increase the visibility of cracks by enhancing the contrast between them and the surrounding pavement.

The image processing phase is dedicated to the core tasks of analyzing and segmenting the images to isolate the cracks from the rest of the image. In this study, four main image processing methods will be evaluated, which will be discussed in detail in the later section.

The **post-processing phase** plays a pivotal role in ensuring the accuracy and usability of the detection results. The primary focus of this phase is on refining the identified features by connecting potentially broken segments of cracks and eliminating noise that may have been misidentified as cracks during the earlier stages of processing.

4.2.2 Metrics

Performance will be evaluated in two aspects, accuracy and generality. For accuracy, the method should be able to accurately separate the crack pixels from the background. For generality, the method should be able to produce acceptable performance across diversified scenarios. Precision, recall, and F1 will be used to evaluate the accuracy of the vendor’s assessment results. Precision is defined as the ratio of correctly detected distresses to all detected distresses. Recall is defined as the ratio of correctly detected distresses to all annotated distresses. F1 is a weighted combination of precision and recall used to measure the overall performance. These three indicators can be expressed as in Equations 4.1 to 4.3:

$$precision = \frac{tp}{tp + fp} \tag{4.1}$$

$$recall = \frac{tp}{tp + fn} \tag{4.2}$$

$$F1 = \frac{2 \times (precision \times recall)}{(precision + recall)} \tag{4.3}$$

where *tp* denotes the number of true positives, *fp* denotes the number of false positives, and *fn* denotes the number of false negatives. The overlap percentage of the detection and the annotation, commonly denoted by Intersection of Union (IoU) (Equation 4.4), is calculated to decide whether a distress is successfully identified. Performances of different scenarios (e.g., different surface types) will be evaluated separately to explore the effects of different parameters (e.g., texture) on the assessment accuracy.

$$IoU = \frac{tp}{tp + fp + fn} \tag{4.4}$$

4.2.3 Methods

4.2.3.1 Thresholding

Thresholding is a simple, yet effective, image processing technique used for segmenting objects from the background. The basic assumption is that crack pixels are relatively darker than other pixels in grayscale digital images, thus cracks can be separated from the background by setting a proper intensity threshold. The method involves selecting a threshold value, and then classifying each pixel in the image as either crack or background based on whether its intensity exceeds this threshold. Methods based on thresholding have been developed by researchers to solve the problem of crack segmentation with intensity images (Zhu et al., 2007; Oliveira and Correia, 2009). In this study, four thresholding-based methods are implemented, including Simple Global Thresholding, Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding.

Simple Global Thresholding applies a single threshold value to the entire image. This method assumes that there is a clear bimodality in the image's histogram, typically representing the background and the cracks to be segmented. For each image, A global threshold value ***T*** is selected. Then, for each pixel ***p*** in the image with intensity ***I(p)***, if ***I(p) > T***, ***p*** is classified as a crack (foreground), otherwise as background. Regular thresholding is straightforward and fast,

making it suitable for images with good contrast between the pavement surface and the cracks. However, its effectiveness diminishes when dealing with images that have uneven lighting or where the contrast between cracks and the pavement surface is not consistent across the image.

Otsu Global thresholding is an advanced global thresholding method that automatically determines the optimal threshold value to separate the objects from the background in an image. The algorithm exhaustively searches for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes, as expressed by Equation 4.5:

$$\sigma_{\omega}^2(t) = \sigma_0\ (t)\sigma_0^2(t) + \ \sigma_1\ (t)\sigma_1^2(t) \tag{4.5}$$

Weights$\omega_0$and ω_1 are the probabilities of the two classes separated by a threshold t, and σ_0^2 and σ_1^2 are variances of these two classes. Unlike simple global thresholding, which requires manual selection of the threshold value, Otsu's method calculates this value automatically, making it particularly effective for images where the histogram of the pixel intensities is bimodal and the distinction between the foreground (cracks) and background (pavement) is not immediately apparent.

Adaptive Gaussian Thresholding overcomes some limitations of regular thresholding by adjusting the threshold value based on the local image properties. This method calculates a unique threshold for each pixel based on a weighted sum of the local neighborhood pixels, where the weights are a Gaussian window. For each image, the threshold for a pixel ***p*** is calculated using the Gaussian average of the intensities in a neighborhood around ***p***, subtracting a constant ***C*** to adjust the sensitivity. This allows the threshold to vary across the image, adapting to different lighting conditions and crack intensities. This approach is particularly useful for images with varying illumination or where the pavement texture varies across the image. The Gaussian weighting helps to smooth local variations, making it more effective at identifying cracks in challenging lighting conditions.

Adaptive Mean Thresholding is similar to adaptive Gaussian thresholding but uses the arithmetic mean of the local neighborhood pixels, rather than a weighted sum, to determine the local threshold. For each pixel ***p***, the threshold is determined by calculating the mean intensity of the pixels in a neighborhood around ***p***, then subtracting a constant ***C***. This method ensures that the threshold adapts to the local intensity variations across the image. Adaptive mean thresholding is effective in scenarios where cracks are present with varying intensities, and the background pavement texture is not uniform. It is less sensitive to local intensity spikes than Gaussian thresholding, making it suitable for images with a more homogeneous appearance but still requiring localized thresholding adjustments.

4.2.3.2 Edge detection

Edge detection is a crucial technique in image processing, particularly effective for identifying discontinuities in intensity, which correspond to edges in an image. In the context of pavement crack detection, edge detection methods are employed to highlight the boundaries of cracks,

differentiating them from the relatively uniform pavement surface. These methods are designed to enhance the visibility of cracks, facilitating their subsequent analysis and measurement. In this study, four edge detection-based methods are implemented, which are Simple Edge Detection, Mean Gradient Edge Detection, Median Gradient Edge Detection, and Otsu Adaptive Edge Detection.

Simple Edge Detection involves the application of standard edge detection operators or filters to identify the boundaries within an image. Common operators include Sobel Operator and Canny Edge Detector.

Sobel Operator estimates the gradient of the image intensity function, highlighting regions with high spatial frequency corresponding to edges. It operates by convolving the image with two separate kernels (filters): one to detect changes in brightness in the horizontal direction (Sobel-X) and another to detect changes in the vertical direction (Sobel-Y). These kernels are designed to respond maximally to edges running vertically and horizontally relative to the pixel grid, with one kernel for each of the two perpendicular orientations. For each pixel, the magnitude of the gradient can be computed using the following equation:

$$G = \sqrt{G_x{}^2 + G_y}{}^2 \tag{4.6}$$

Where G_x denotes the gradient in the horizontal direction, and G_y denotes the gradient in the vertical direction. The direction (or orientation) of the edge can also be calculated using the arctangent:

$$Angle(\theta) = \tan^{-1}(\frac{G_y}{G_x}) \tag{4.7}$$

After getting the gradient magnitude and direction, a full scan of the image is done to remove any unwanted pixels that may not constitute the edge. For this, at every pixel, the pixel is checked if it is a local maximum in its neighborhood in the direction of the gradient. This step is commonly known as non-maximum suppression.

For Canny Edge Detector, Hysteresis thresholding is applied to decide which edges are real edges and which are not. For this, two threshold values, minVal and maxVal, need to be decided. Any edges with an intensity gradient more than maxVal are sure to be edges and those below minVal are sure to be non-edges, so discarded. Those who lie between these two thresholds are classified as edges or non-edges based on their connectivity. If they are connected to "sure edge" pixels, they are considered to be part of the edges. Canny Edge Detector aims to detect edges with minimal error rate, good localization, and minimal response (one per edge), based on gradients and non-maximum suppression. These operators work by convolving the image with a kernel that is designed to respond strongly to edges in the image, typically where there is a significant change in intensity.

Mean Gradient Edge Detection involves modifying the gradient calculation to incorporate the mean value of gradients in a local neighborhood. This method can smooth out the noise and emphasize larger, more significant edges. Calculating the gradient at each pixel, then adjusting the gradient value based on the average gradient within a surrounding window, helps to reduce the impact of small, noisy gradients while preserving the more significant edges corresponding to cracks. This method is useful in images with variable texture or noise, where standard edge detection might highlight too many irrelevant features.

Median Gradient Edge Detection is similar to mean gradient edge detection but uses the median of gradients in a local neighborhood instead of the mean. The median is less sensitive to outliers, making this approach particularly robust to noise. This method computes the gradient at each pixel, and then adjusts it based on the median gradient value within a surrounding window. This technique effectively reduces the impact of outlier gradients that are not representative of true edges. Also like the mean gradient edge detection, this method could be effective in noisy images or when the pavement surface contains a lot of texture or debris.

Otsu Adaptive Edge Detection combines the principles of Otsu's thresholding method with edge detection to adaptively select the threshold for edge detection based on the histogram of the image gradients. First, the method computes the gradient magnitude for each pixel. Then, it applies Otsu's method to these gradient magnitudes to find an optimal threshold that separates the edges from the rest of the image. This threshold is then used to identify significant edges. This method offers a robust way to detect edges in images with varying lighting conditions or contrast levels. Particularly effective when the distinction between cracks and pavement surface is not consistent across the image.

4.2.3.3 Seed-based crack detection

Seed-based crack detection was implemented for real-time crack detection in a research study sponsored by TxDOT (Huang and Xu, 2007). In this method, a pavement image is divided into grid cells of 8 x 8 pixels, and each cell is classified as a non-crack or crack cell using the grayscale information of the border pixels. Whether a crack cell can be regarded as a basic element (or seed) depends on its contrast to the neighboring cells. Several crack seeds can be called a crack cluster if they fall on a linear string. A crack cluster corresponds to a dark strip in the original image that may or may not be a section of a real crack. Additional conditions to verify a crack cluster include the requirements in the contrast, width, and length of the strip. If verified crack clusters are oriented in similar directions, they will be joined to become one crack.

4.2.3.4 Multiscale wavelets

Multiscale wavelets-based method is a sophisticated technique used in image processing for detecting edges at various scales, making it particularly effective for identifying cracks in pavement images. Wavelets are mathematical functions that can decompose a signal or an image into different frequency components, allowing for the analysis of various details at multiple scales. This characteristic is highly beneficial for crack identification, as cracks can vary

significantly in width, depth, and visibility. Researchers have already applied Wavelet Transform to decompose the original image into different subsamples to capture the details of the cracks on different scales (Wang et al., 2007).

The principle behind multiscale wavelet-based edge detection involves using wavelet transforms to decompose the image into a series of images at different scales or resolutions. Each decomposed image highlights features of the pavement at different sizes, enabling the detection of both small and large cracks that might be missed by single-scale edge detection methods.

4.3. Experiment Results

This chapter presents a comprehensive analysis of the performance of the four rules-based image processing techniques applied to segment pavement distresses. Given the diverse characteristics and conditions of ACP, JCP, and CRCP, the outcomes for each pavement type will be discussed separately.

4.3.1 Thresholding

4.3.1.1 ACP

For ACP, unsealed cracking and sealed cracking are examined separately. 3D images are extracted for unsealed cracking detection, while 2D images are extracted for sealed cracking detection. This is because the depth information makes it easier to detect and delineate unsealed cracks but not sealed cracks, as the filling materials are meant to level the surface.

For unsealed cracks of ACP, Figure 4.1 shows the segmentation results of three different 3D images using different thresholding methods. For each row in the figure, the first image is the original 3D image, and the second image is the ground truth, a binary image with white pixels indicating the cracking. The following images are segmentation results using Simple Global Thresholding, Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding, respectively.

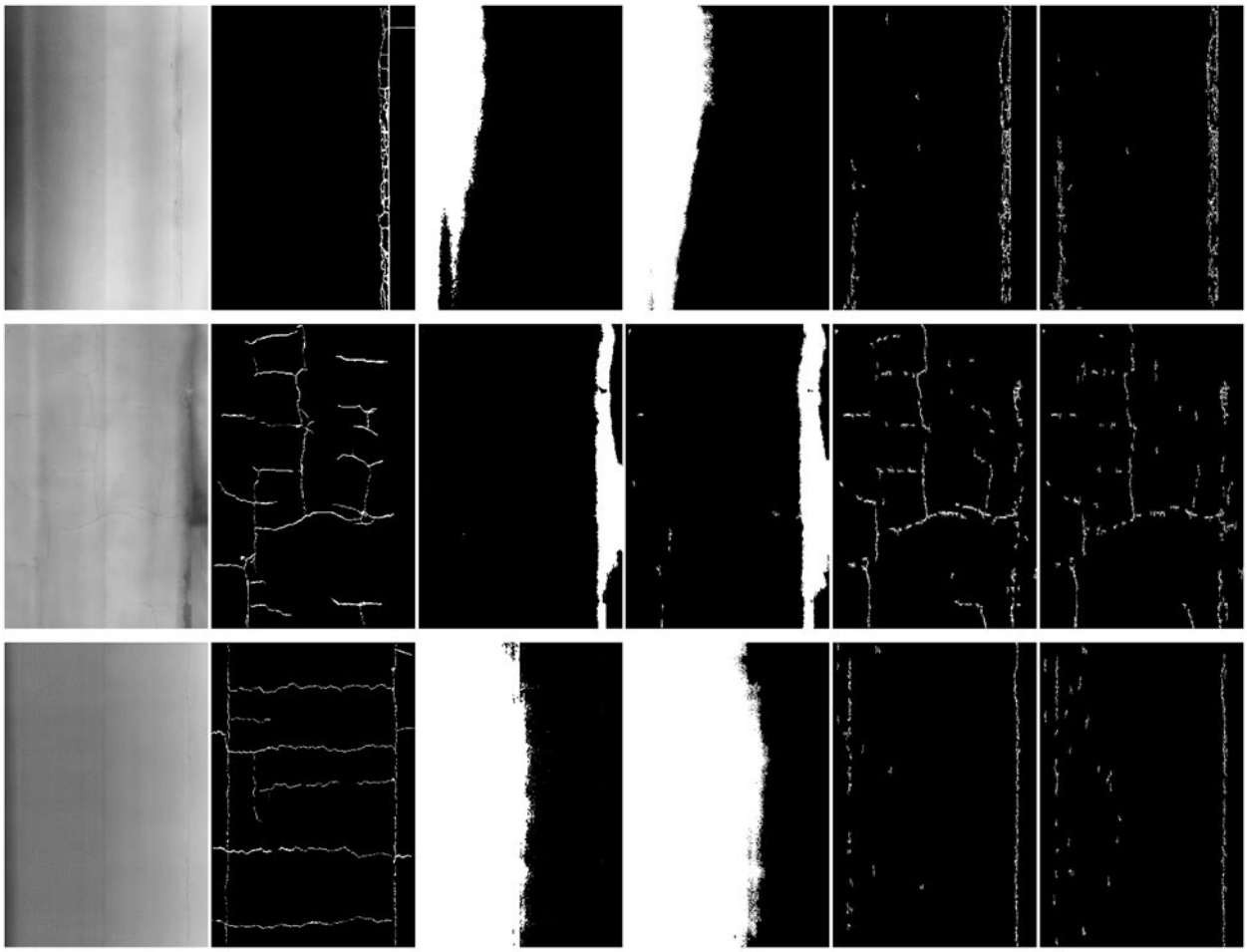


Figure 4.1 Samples of segmentation results using thresholding methods (from left to right: original ACP 3D image, ground truth, Simple Global Thresholding, Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding)

From Figure 4.1, the following interpretations can be drawn:

Global Thresholding Inadequacy: The segmentation results from Simple Global Thresholding and Otsu Global Thresholding suggest that these methods are not entirely effective for this application. The likely cause of their suboptimal performance is the non-uniform background of the images, which can be attributed to the inherent unevenness often found in flexible pavement surfaces. Global thresholding assumes consistent intensity distribution across the entire image, which is not the case here, leading to inaccurate segmentation.

Improved Performance of Adaptive Thresholding: The images resulting from Adaptive Mean Thresholding and Adaptive Gaussian Thresholding show a marked improvement in the segmentation of pavement cracks. These adaptive methods outperform global thresholding likely due to their ability to apply different threshold values across the image. This local or regional approach to thresholding takes into account the variability in lighting and texture, which is

characteristic of pavement surfaces, and adjusts the thresholds accordingly, resulting in a more accurate delineation of the cracks.

False Positives in Adaptive Thresholding: Despite the improved performance, the adaptive thresholding methods are not without their issues. They seem to be prone to false positives, where certain image features are incorrectly identified as cracks. This can occur due to the presence of foreign objects, surface deformation, or texture patterns that these local thresholding methods may misinterpret as cracks due to their similar intensity profiles.

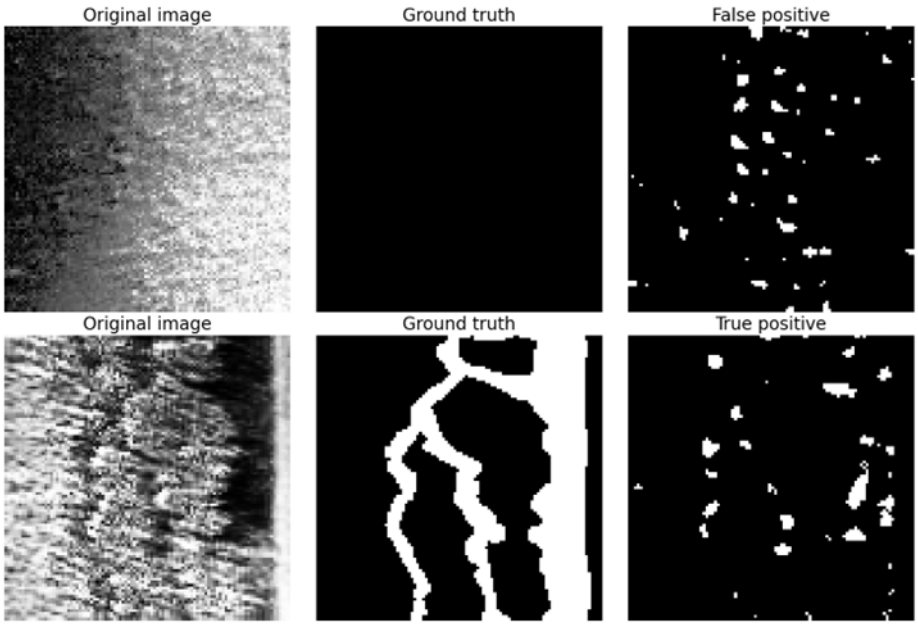


Figure 4.2 Segmentation results of two patches from the same ACP 3D image using Adaptive Mean Thresholding

Figure 4.2 shows a sample of two patches cropped from the same image. For the first row, the pavement texture is falsely identified as cracks, due to significant local depth disparity. Potentially, such false positives can be eliminated by postprocessing, as these false positives generally show as isolated spots. However, the true positives can also be represented by isolated spots, subjecting to the local threshold values. It is tricky to eliminate the false positive without affecting the true positive.

False Negatives in Adaptive Thresholding: There is also an indication of false negatives, particularly with thin and shallow cracks. These types of cracks present a challenge for all the thresholding methods tested. Their low contrast and minimal presence compared to the surrounding pavement texture make them difficult to detect, as shown in Figure 4.3. Since thresholding methods rely on a clear distinction between the crack and the pavement in terms of intensity, any crack that does not significantly differ from the background is at risk of being missed, leading to false negatives.

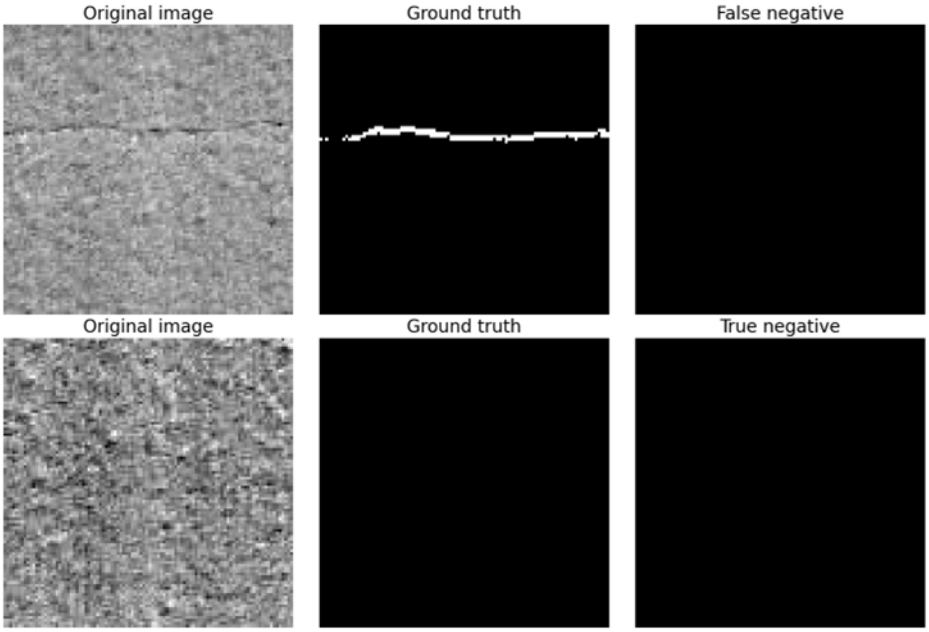


Figure 4.3 Segmentation results of two patches from the same ACP 3D image using Adaptive Mean Thresholding (top row: thin cracking, bottom row: no cracking)

Table 4.1 shows the metric values of average precision, average recall, average F1, and average IoU using different thresholding methods on the ACP 3D image dataset. It is worth noting that objects like lane marking and curbside are not treated, which has introduced a certain amount of noise into the final prediction. According to the metrics, global thresholding methods are significantly insufficient for ACP distress segmentation, with average IoU values of 1.2% and 1.3%, respectively. This is possibly due to the universal surface range variation of flexible pavement. In comparison, the adaptive thresholding methods show significant improvement in the overall segmentation performance, with Adaptive Mean Thresholding delivering the best performance. However, the average IoU value of 14.7% is not yet satisfactory for network-level pavement condition evaluation. According to the former analysis, there are still many factors that could affect the performance of adaptive thresholding methods.

Table 4.1 Metric values of ACP 3D images using different thresholding methods.

Method	Average Precision	Average Recall	Average F1	Average IoU
Simple Global Thresholding	1.2%	**60.8%**	2.3%	1.2%
Otsu Global Thresholding	1.3%	55.1%	2.5%	1.3%
Adaptive Mean Thresholding	**29.7%**	26.0%	**24.3%**	**14.7%**
Adaptive Gaussian Thresholding	25.9%	16.8%	17.8%	10.3%

Figure 4.4 shows the sealed cracking segmentation results of three different 2D images using different thresholding methods. For each row in the figure, the first image is the original 2D image, and the second image is the ground truth, a binary image with white pixels indicating the cracking. The following images are segmentation results using Simple Global Thresholding,

Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding, respectively.

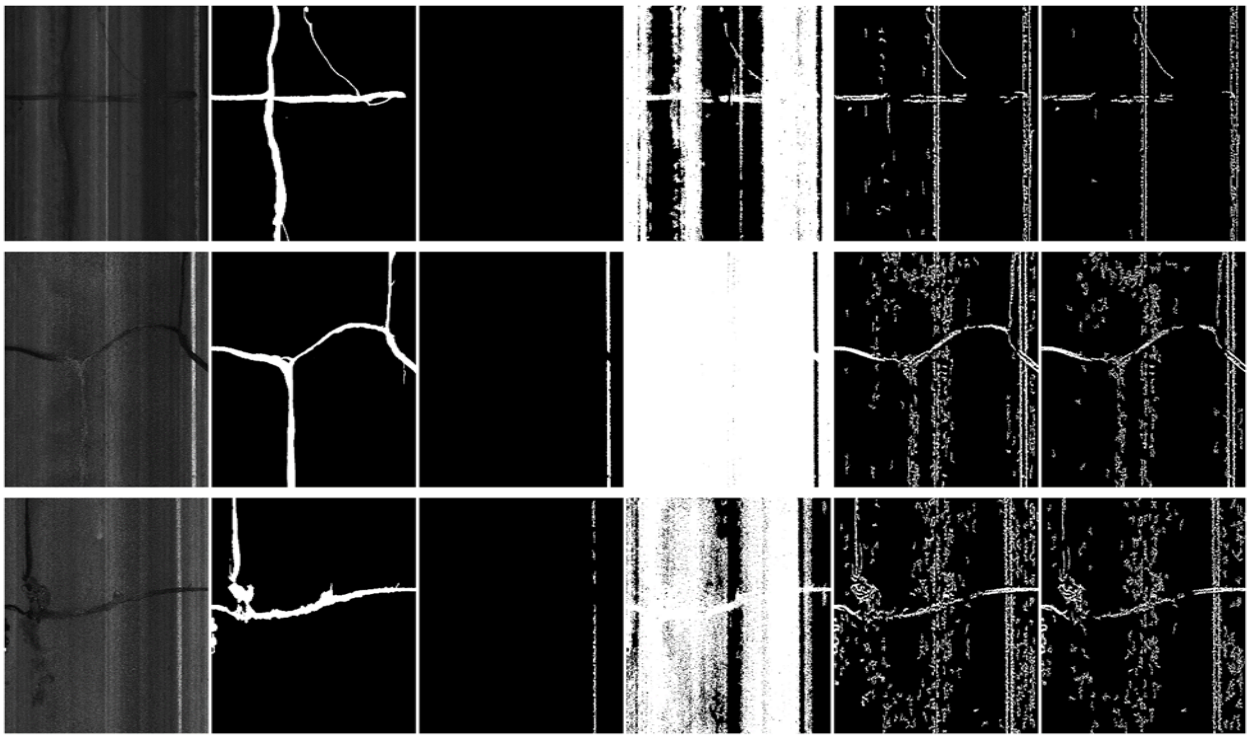


Figure 4.4 Samples of segmentation results using thresholding methods (from left to right: original ACP 2D image, ground truth, Simple Global Thresholding, Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding)

Similar to the performance with 3D images, global thresholding methods failed to perform the sealed cracking task, while adaptive thresholding methods generated comparatively better results. From Figure 4.4, the following interpretations can be drawn.

Complex Background. The 2D image of the ACP shows a more complex background than that of the 3D image. This complexity can largely be ascribed to the intrinsic properties and constituents of the pavement material itself. The heterogeneous mixture of aggregates—varying in size, shape, and color—combined with the bituminous binder creates a multifaceted texture that is further complicated by the effects of weathering, aging, and repairs. Such complexity in the background can cause threshold-based segmentation methods to produce errors by either missing the actual cracks or identifying non-crack areas as cracks (false positives).

Contrast between Sealed Crack and Background. The sealed cracks are filled with a material that has a reflectivity similar to the surrounding asphalt, making the contrast between the cracks and the background small (shown in Figure 4.5). This similarity in reflectivity can lead to challenges in distinguishing between the sealed cracks and the intact asphalt. Sometimes the sealing material, mostly asphalt, can be very reflective, causing it to appear white or light-

colored in the 2D image (shown in Figure 4.5). Such low or contrary contrast situations make it difficult for thresholding methods to accurately segment out the sealed cracks.

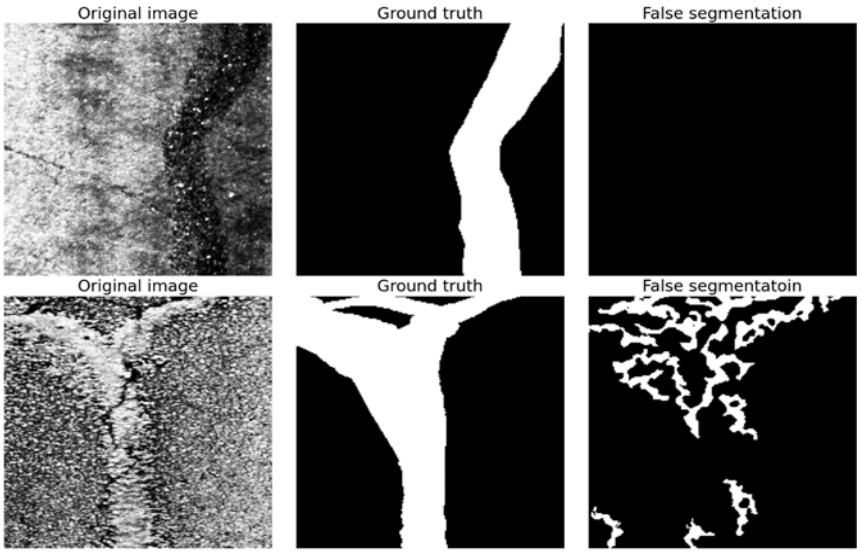


Figure 4.5 Segmentation results of two ACP 2D image patches with low contrast (top) and high reflectivity (bottom)

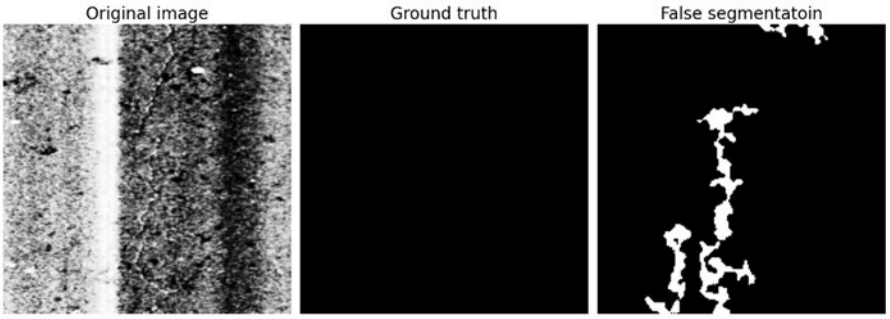


Figure 4.6 Segmentation result of an ACP 2D image patch with white longitudinal line noise introduced by image collection sensor

Noise and False Segmentation. The presence of noise in the image can exacerbate the difficulty of accurately segmenting the sealed cracks. Noise can arise from a variety of sources, including but not limited to camera sensor noise (shown in Figure 4.6), variations in lighting conditions, or shadows cast on the pavement surface. Noise can lead to false segmentation results, where non-crack features are incorrectly identified as cracks (false positives), or actual cracks are missed (false negatives).

4.3.1.2 JCP

Figure 4.7 shows the segmentation results of three different 3D images of JCP using different thresholding methods. For each row in the figure, the first image is the original 3D image, and the second image is the ground truth, a binary image with white pixels indicating the cracking.

The following images are segmentation results using Simple Global Thresholding, Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding, respectively.

According to Figure 4.7, global thresholding methods failed to detect the cracking, while adaptive thresholding methods generated comparatively better results. This is similar to that of the ACP images. The main shortage of adaptive thresholding methods for crack segmentation with JCP images are:

Incomplete Segmentation of wide and spalled cracks. The adaptive thresholding methods have not completely identified wide cracks, particularly those that are spalled. This incomplete segmentation is likely due to the use of local threshold values that may not adequately capture the full extent of these wide or irregular features (as shown in Figure 4.8). Local thresholding might be sensitive to variations in the crack width and the texture around it, leading to parts of the crack being missed if they do not meet the specific threshold criteria set for a given local area.

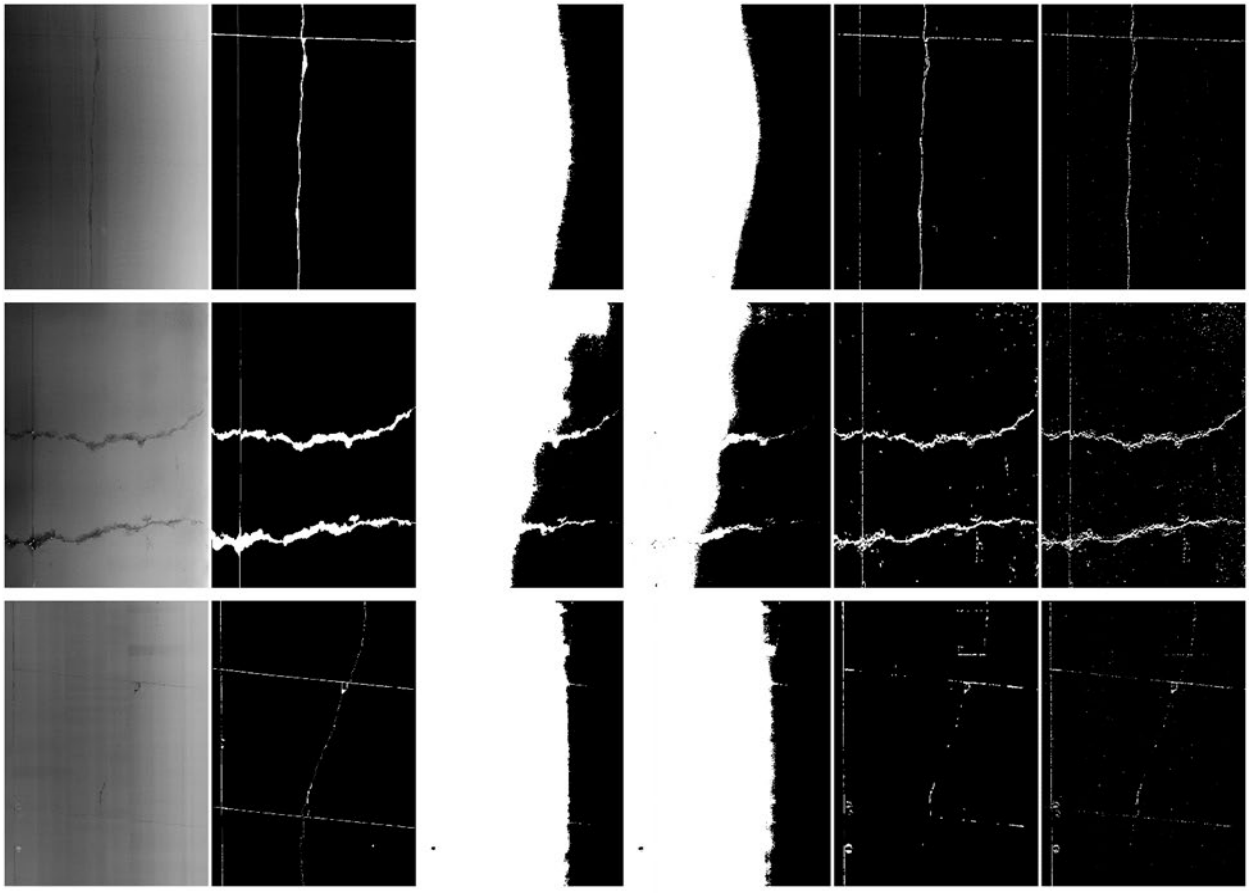


Figure 4.7 Samples of segmentation results using thresholding methods (from left to right: original JCP 3D image, ground truth, Simple Global Thresholding, Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding)

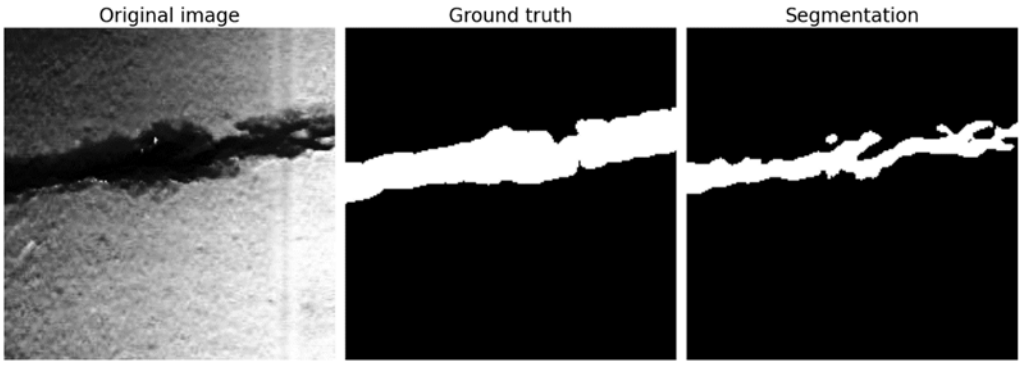


Figure 4.8 Segmentation result of a JCP 3D image patch with spalled crack

Discontinuity in segmented cracks. The segmented cracks appear to be discontinuous, especially in the case of thinner cracks (Figure 4.9). This suggests that the segmentation process may have difficulty detecting and tracing the entirety of narrower cracks, likely due to the crack width being close to the resolution limit of the imaging or the thresholding technique. As a result, the algorithm may pick up segments of the crack where the contrast or crack width is sufficient to exceed the threshold, leading to a segmented appearance rather than a continuous line.

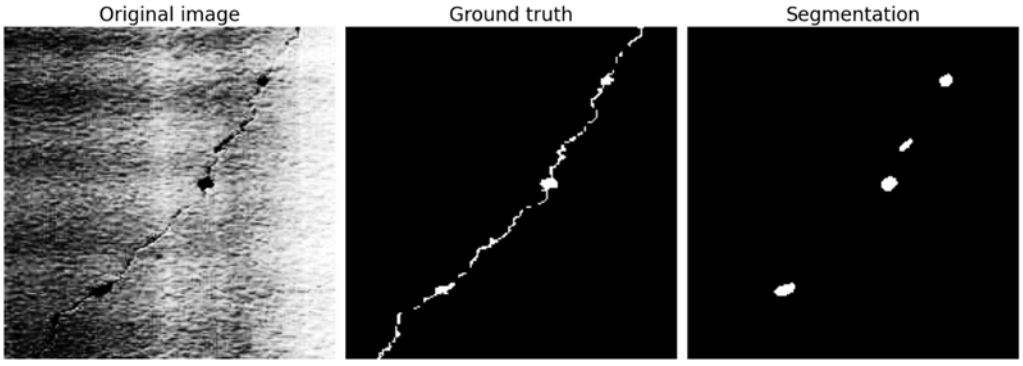


Figure 4.9 Segmentation result of a JCP 3D image patch with thin cracks

Table 4.2 shows the metric values of average precision, average recall, average F1, and average IoU using different thresholding methods on the JCP 3D image dataset.

Table 4.2 Metrics values of JCP 3D images using different thresholding methods.

Method	Precision	Recall	F1	IoU
Simple Global Thresholding	2.9%	**86.2%**	5.5%	2.8%
Otsu Global Thresholding	12.0%	64.3%	6.4%	3.5%
Adaptive Mean Thresholding	**74.9%**	49.0%	**57.9%**	**41.7%**
Adaptive Gaussian Thresholding	70.6%	38.2%	48.3%	32.3%

According to the table, Global thresholding methods (Simple Global and Otsu Global) have very low IoU values, indicating a complete failure in accurately segmenting cracks in JCP. On the

other hand, adaptive thresholding methods (Adaptive Mean and Adaptive Gaussian) show comparatively better performance, with much higher IoU values. This is consistent with the process of ACP dataset. Compare with ACP with the highest IoU value of 14.7%, the highest IoU value with JCP is 41.7%, which is significantly higher. This is probably due to that the contrast between the cracks and the background in JCP images is generally more pronounced than in ACP images. JCP, with its more uniform material and color, and less complex surface textures, may provide clearer delineation of cracks. However, the presence of false negatives, particularly in detecting extremely wide and very thin cracks, points to limitations in the adaptive thresholding methods with JCP. These methods may not accurately segment the entire breadth of crack types, possibly because wide cracks exhibit internal variations that are misclassified as background and thin cracks do not alter the local pixel values sufficiently to be detected. This highlights the need for further methodological advancements to ensure comprehensive and accurate crack detection across different pavement types.

4.3.1.3 CRCP

Figure 4.10 shows the segmentation results of three different 3D images of CRCP using different thresholding methods. The primary challenges identified in Figure 4.10 involve detecting wide or spalled cracks and thin cracks. These difficulties suggest that for the CRCP, while thresholding methods can successfully segment certain features of the pavement surface, they struggle with accurately identifying cracks that either have a significant width or are very fine. The segmentation results on CRCP images are similar to those for the JCP indicating that the challenges and effectiveness of the thresholding methods are consistent across these two types of concrete pavements. This similarity is likely due to the comparable material properties and conditions (such as texture, color, and crack patterns) found in both pavement types.

Table 4.3 Metrics values of CRCP 3D images using different thresholding methods.

Method	Precision	Recall	F1	IoU
Simple Global Thresholding	2.1%	9.0%	1.1%	0.6%
Otsu Global Thresholding	0.7%	21.6%	1.3%	0.7%
Adaptive Mean Thresholding	44.1%	**53.0%**	**45.1%**	**30.7%**
Adaptive Gaussian Thresholding	**51.0%**	41.3%	42.6%	28.3%

Table 4.3 shows the metric values of average precision, average recall, average F1, and average IoU using different thresholding methods on the CRCP 3D image dataset. Despite the similarities between CRCP and JCP, the best average IoU value of CRCP (30.7%) is significantly lower than that of JCP (41.7%). This is possibly due to the image variation distribution within the CRCP and JCP datasets. According to Table 4.3, further methodological advancements are also needed to ensure comprehensive and accurate crack detection across different pavement types.

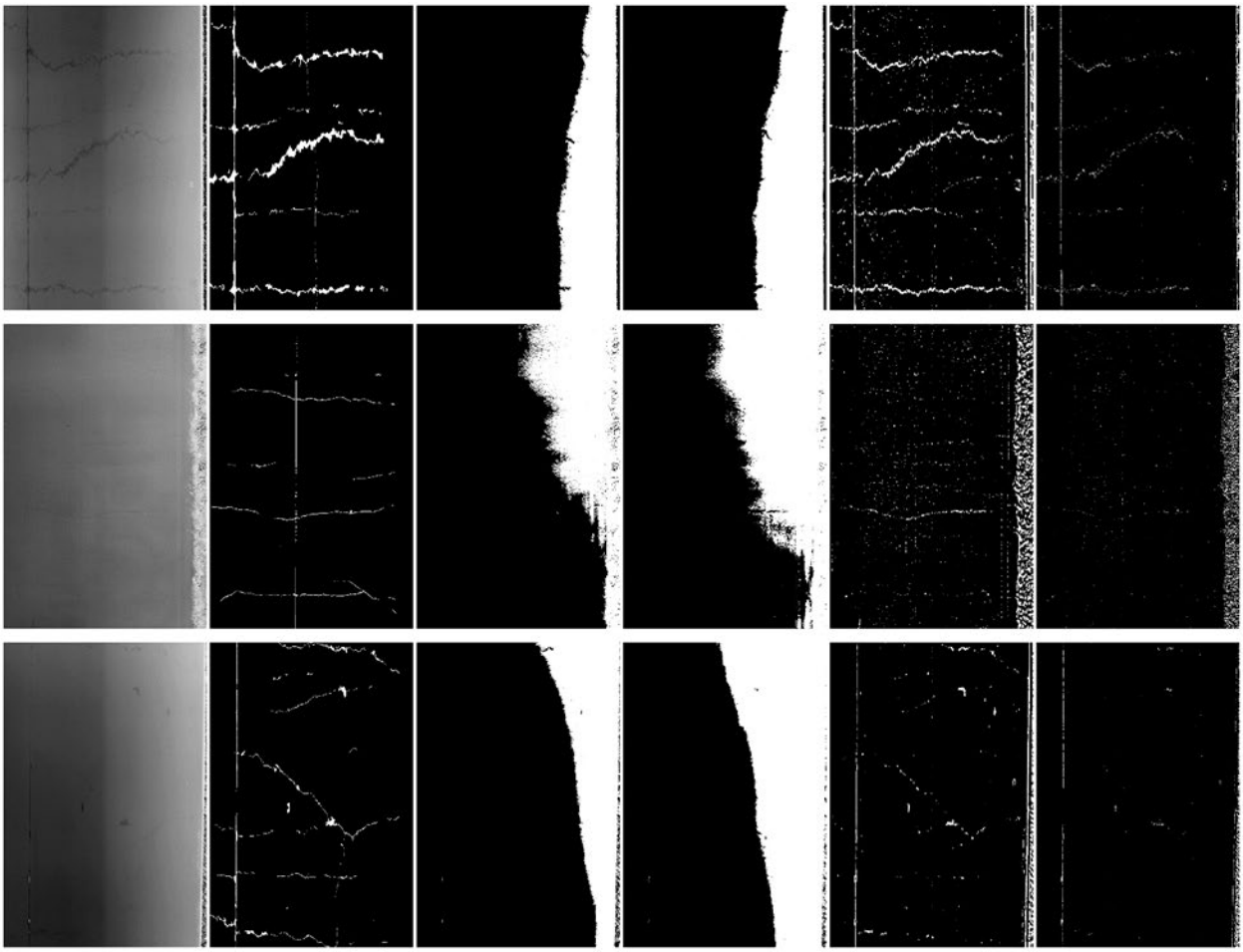


Figure 4.10 Samples of segmentation results using thresholding methods (from left to right: original CRCP 3D image, ground truth, Simple Global Thresholding, Otsu Global Thresholding, Adaptive Mean Thresholding, and Adaptive Gaussian Thresholding)

4.3.2 Edge detection

This section interprets the experimental findings from applying four edge detection methods, Simple Global Thresholding, Mean Gradient Edge Detection, Median Gradient Edge Detection, and Otsu Adaptive Edge Detection, on three types of pavements, ACP, JCP, and CRCP. The experiment's objective was to evaluate the effectiveness of these methods in detecting pavement cracking through image segmentation techniques.

4.3.2.1 ACP

Figure 11 shows the edge detection results of three different 3D images using different edge detection methods. For each row in the figure, the first image is the original 3D image, and the second image is the ground truth, a binary image with white pixels indicating the cracking. The

following images are edge detection results using Simple Global Thresholding, Mean Gradient Edge Detection, Median Gradient Edge Detection, and Otsu Adaptive Edge Detection, respectively.

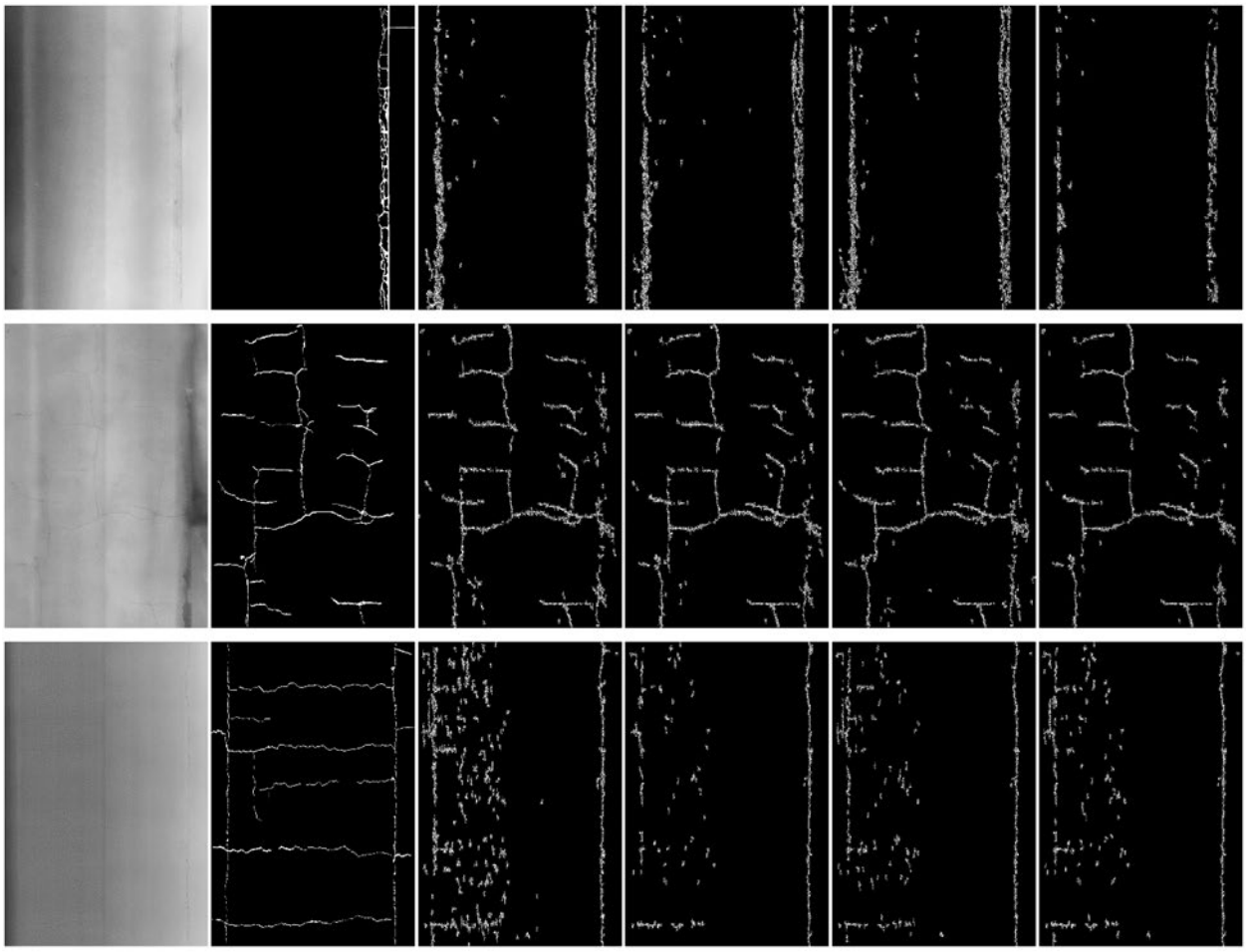


Figure 4.11 Samples of segmentation results using edge detection methods (from left to right: original ACP 3D image, ground truth, Simple Edge Detection, Mean Gradient Edge Detection, Median Gradient Edge Detection, and Otsu Adaptive Edge Detection)

From Figure 4.11, the following interpretations can be drawn:

Similar Performance Across Methods. The fact that all four edge detection methods produced similar results indicates that the complexity of the edge detection algorithm might not be the primary factor in improving crack detection accuracy in ACP pavements. This could mean that the basic structural and textural characteristics of ACP cracks are sufficiently captured by even the simplest edge detection method, suggesting a small variance of gradient features across the test dataset. This observation is crucial for practical applications, as it suggests that simpler, less computationally intensive methods could be equally effective for certain tasks, potentially reducing processing time and resource consumption.

False Positives. The high incidence of false positives in the first row's images, particularly with the Simple Edge Detection method, highlights the challenge of distinguishing between actual cracks and pavement textures. This suggests that these methods lack the specificity to accurately differentiate between crack edges and textural patterns inherent in ACP surfaces. This could lead to over-detection, where normal textural features of the pavement are incorrectly identified as cracks, compromising the precision of the detection process.

False Negatives on Thin Cracks. The failure to detect thin cracks, resulting in many false negatives, especially in the third row's images, points to a limitation in the sensitivity of the applied methods. This could be due to the resolution of the input images or the inherent limitations of the edge detection algorithms, which might not be fine-tuned to recognize very narrow, subtle features as cracks. This observation underscores the need for enhancing the edge detection methods or preprocessing steps to improve crack detection accuracy, especially for minor features that are crucial for early-stage pavement maintenance and repair decisions.

Table 4.4 Metrics values of ACP 3D images using different edge detection methods.

Method	Precision	Recall	F1	IoU
Simple Edge Detection	26.6%	72.9%	33.8%	9.7%
Mean Gradient Edge Detection	37.9%	56.2%	38.0%	9.8%
Median Gradient Edge Detection	35.7%	59.8%	**39.1%**	10.2%
Otsu Adaptive Edge Detection	33.4%	63.6%	36.4%	9.4%

Table 4.4 shows the precision, recall, F1 score, and Intersection over Union (IoU) metrics for the four edge detection methods. In evaluating the effectiveness of edge detection methods for pavement crack detection, it's crucial to consider the inherent challenges posed by the sensitivity of edge location information and the subjectivity involved in manual annotation. To address these challenges and ensure a fair and consistent assessment of the edge detection algorithms' performance, detected edges that are within two pixels of the manually annotated labels have been considered as true positives in the analysis. According to Table 4.4, the Median Gradient Edge Detection method stands out with the highest F1 Score and IoU, indicating its superior balance in accurately identifying true crack features while minimizing false positives and negatives. This suggests that despite the simplicity of the Simple Edge Detection method, which shows a notably high recall but lower precision, the Median method's nuanced approach to gradient evaluation offers a more effective strategy for ACP crack detection. The close performance metrics among the methods, however, highlight a relatively uniform ability to capture crack features in ACP, with slight variations in precision and recall emphasizing the trade-offs between detecting more cracks and avoiding misclassification of non-crack features.

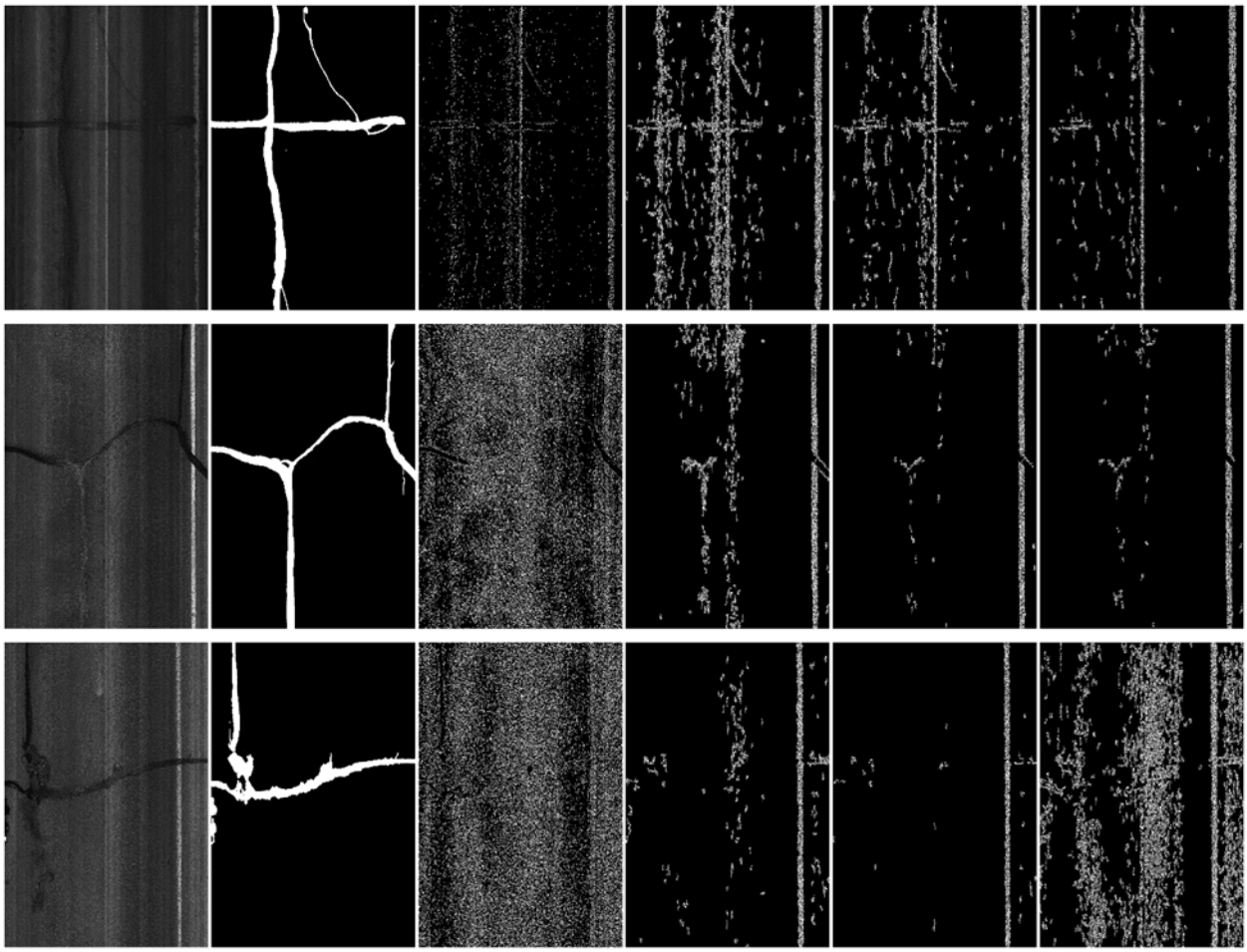


Figure 4.12 Samples of segmentation results using edge detection methods (from left to right: original ACP 2D image, ground truth, Simple Edge Detection, Mean Gradient Edge Detection, Median Gradient Edge Detection, and Otsu Adaptive Edge Detection)

Figure 4.12 and Table 4.5 show the sealed cracking segmentation results of the ACP 2D images. According to the results, all four edge detection methods failed to achieve effective segmentation of sealed cracks. This unanimous failure across diverse methodologies signals a significant challenge specific to the nature of sealed cracking in ACP. Typically, these cracks are sealed with materials such as asphalt, which closely resembles the visual and textural properties of the surrounding pavement, leading to extremely low contrast between the sealed crack and the background in 2D images. Such low contrast severely undermines the edge detection algorithms' ability to discern the boundaries between crack and non-crack areas, as these methods fundamentally rely on detecting notable changes in intensity or texture to identify edges. To enhance the low contrast between sealed cracking and background, features of other irrelevant objects, such as pavement texture, could also be amplified, leading to abundant false positives. This scenario underscores a critical limitation of current edge detection techniques for handling the nuanced task of identifying sealed cracks in ACP 2D images. It suggests the necessity for developing more sophisticated segmentation approaches that can overcome the challenges of low contrast and high false-positive rates, potentially through leveraging advanced pattern

recognition and machine learning technologies that can learn to distinguish the subtle differences between sealed cracks and their surrounding pavement.

Table 4.5 Metrics values of ACP 2D images using different edge detection methods.

Method	Precision	Recall	F1	IoU
Simple Edge Detection	4.2%	92.9%	7.5%	2.9%
Mean Gradient Edge Detection	5.6%	31.4%	9.0%	2.9%
Median Gradient Edge Detection	6.7%	23.3%	8.5%	2.4%
Otsu Adaptive Edge Detection	6.1%	34.3%	9.3%	2.8%

4.3.2.2 JCP

Figure 4.13 illustrates the segmentation results on JCP, following the same layout as for ACP. From Figure 4.13, the following interpretations can be drawn.

Comparable Performance Among Methods. The observation that Simple Edge Detection's performance closely matches that of more sophisticated methods in JCP pavements suggests that the key features of cracks in JCP are not significantly better captured by complex algorithms. This might be due to the distinct nature of cracks in concrete pavements, which could be more defined and less influenced by textural noise compared to asphalt pavements. This finding could influence the selection of edge detection techniques for different pavement types, favoring simpler, faster methods where applicable.

Wide Spalled Cracks Detection. The identification of wide-spalled cracks as areas of edges rather than distinct lines indicates a challenge in handling the spatial variability within cracks. This might reflect the difficulty in edge detection algorithms to distinguish between the depth variations and the actual edges of the cracks. Such a challenge is particularly pronounced in JCP, where spalled areas can present a complex pattern of shadows and texture changes that mimic edge-like features, potentially confusing the detection algorithms.

False Negatives for Thin Cracks. Similar to ACP, the consistent issue of false negatives for thin cracks in JCP suggests a common limitation across pavement types in detecting narrow, less pronounced cracks. This indicates that the edge detection algorithms might require adjustments or supplementary techniques, such as enhanced preprocessing or postprocessing steps, to improve their sensitivity to such features. Addressing this limitation is crucial for comprehensive pavement evaluation and the early detection of deterioration signs.

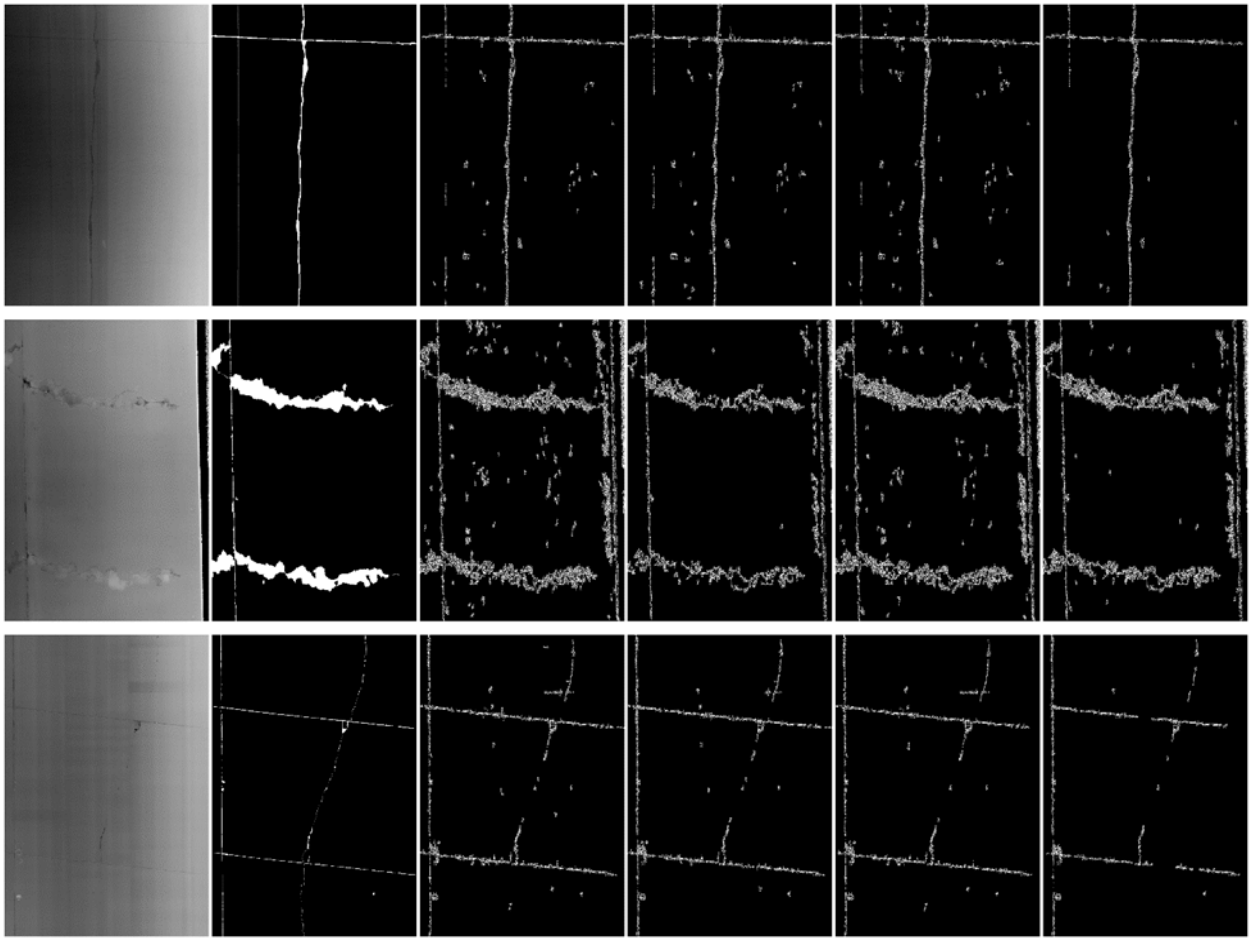


Figure 4.13 Samples of segmentation results using edge detection methods (from left to right: original JCP 3D image, ground truth, Simple Edge Detection, Mean Gradient Edge Detection, Median Gradient Edge Detection, and Otsu Adaptive Edge Detection)

Table 4.6 Metrics values of JCP 3D images using different edge detection methods.

Method	Precision	Recall	F1	IoU
Simple Edge Detection	36.8%	95.3%	49.9%	14.3%
Mean Gradient Edge Detection	47.5%	79.7%	55.3%	14.0%
Median Gradient Edge Detection	42.6%	89.5%	55.1%	14.9%
Otsu Adaptive Edge Detection	48.1%	86.9%	56.3%	14.1%

Table 4.6 shows the precision, recall, F1 score, and Intersection over Union (IoU) metrics for the four edge detection methods applied to JCP. The data reveals a relatively uniform performance across all methods, with Simple Edge Detection slightly underperforming in comparison to the others. This slight underperformance could be attributed to the inherent simplicity of the Simple Edge Detection method, which, while effective in identifying a broad range of cracks, may lack the refined sensitivity and specificity of the more sophisticated methods. The highest F1 score recorded, 56.3% by the Otsu Adaptive Edge Detection method, represents a significant

improvement over the best performance observed in ACP (39.1%), highlighting the relative effectiveness of these methods in detecting cracks in JCP surfaces. However, despite this notable improvement, the F1 score of 56.3% is still considered below the threshold typically regarded as acceptable for highly reliable crack detection systems. Ideally, an F1 score approaching or exceeding 70% would be indicative of a highly effective detection mechanism, suggesting a balanced and robust capability to accurately identify true positives while minimizing false positives and negatives. The current shortfall indicates a critical need for methodological enhancements.

4.3.2.3 CRCP

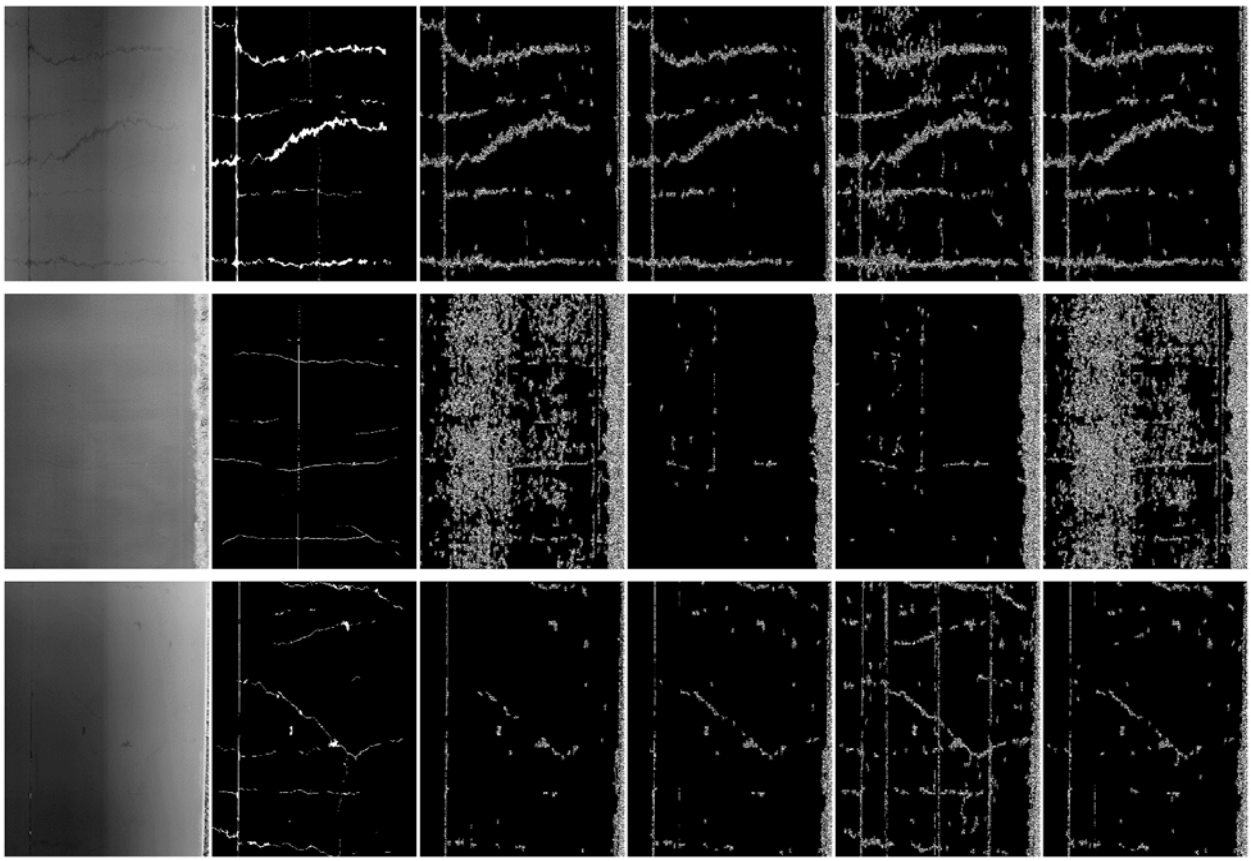


Figure 4.14 Samples of segmentation results using edge detection methods (from left to right: original CRCP 3D image, ground truth, Simple Edge Detection, Mean Gradient Edge Detection, Median Gradient Edge Detection, and Otsu Adaptive Edge Detection)

Figure 4.14 illustrates the segmentation results on CRCP, following the same layout as for ACP and JCP. From Figure 4.14, the following interpretations can be drawn.

Inconsistent Detection. The varied success of the edge detection methods in identifying crack features in CRCP images points to a significant challenge in applying these techniques to continuously reinforced concrete pavements. The failure of these methods for certain images suggests that the features used for edge detection, such as gradients, might not be sufficient for

capturing the complex patterns of cracks in CRCP. This inconsistency highlights the potential need for more sophisticated or specialized algorithms that can better account for the unique characteristics of CRCP cracking, including its often more chaotic and less predictable patterns.

False Positives and Negatives. The observation that cracks are either overly identified (leading to many false positives) or not identified at all (resulting in false negatives) underscores the difficulty in distinguishing crack edges from the background in CRCP pavements. This indicates a fundamental limitation in the current edge detection approaches, where the algorithms are either too sensitive, mistakenly identifying normal pavement features as cracks, or not sensitive enough, missing genuine crack features. This challenge suggests a need for refining the edge detection criteria or incorporating additional contextual or textural information to improve discrimination between cracks and non-crack features in CRCP pavements.

Table 4.7 shows the precision, recall, F1 score, and Intersection over Union (IoU) metrics for the four edge detection methods applied to CRCP. According to the table, the Simple Edge Detection delivered the highest F1 score of 52.3. This outcome suggests that the mean gradient, median gradient, and adaptive thresholding techniques, as utilized in the more complex edge detection methods, might not be key or adequate for deciding the thresholds in edge detection, particularly for the unique challenges presented by CRCP. This realization opens avenues for reevaluating the criteria and methodologies used in crack detection algorithms, suggesting a need for a more tailored approach that better aligns with the specific detection challenges of CRCP. It calls for further research and development efforts to explore alternative features or methods that could enhance the precision and reliability of crack detection in CRCP pavements, potentially incorporating a combination of simple and complex analysis techniques to achieve a more effective balance between sensitivity to crack detection and specificity in distinguishing true cracks from other pavement features.

Table 4.7 Metrics values of CRCP 3D images using different edge detection methods.

Method	Precision	Recall	F1	IoU
Simple Edge Detection	47.2%	76.5%	52.3%	13.4%
Mean Gradient Edge Detection	39.2%	76.8%	42.3%	11.2%
Median Gradient Edge Detection	43.6%	77.1%	48.6%	12.9%
Otsu Adaptive Edge Detection	42.7%	80.6%	49.3%	13.3%

4.3.3 Seed-based crack detection

The experiment in this section was designed to evaluate the effectiveness of seed-based crack detection applied to ACP, JCP, and CRCP. Seed-based crack detection involves initially identifying small, distinct features or "seeds" within the pavement images that are highly likely to be parts of cracks. These seeds serve as starting points for a broader crack detection process, where algorithms iteratively expand from these initial points to trace the full extent of the cracks across the pavement surface. To apply seed-base crack detection, a four-step process was implemented. First, 2D/3D images are loaded in grayscale to simplify the analysis by focusing on intensity variations. Following this, the image is divided into cells with 8x8 pixels, which

allows for a more granular examination of the image's features. Within each of these cells, a set of predetermined features, such as edge density, intensity contrast, and texture entropy, are calculated to capture various aspects of the cell's content, ranging from edge presence and intensity variability to texture complexity. Finally, the calculated values for each selected feature are aggregated, each contributing equally to constructing an integrated map that highlights potential crack cells.

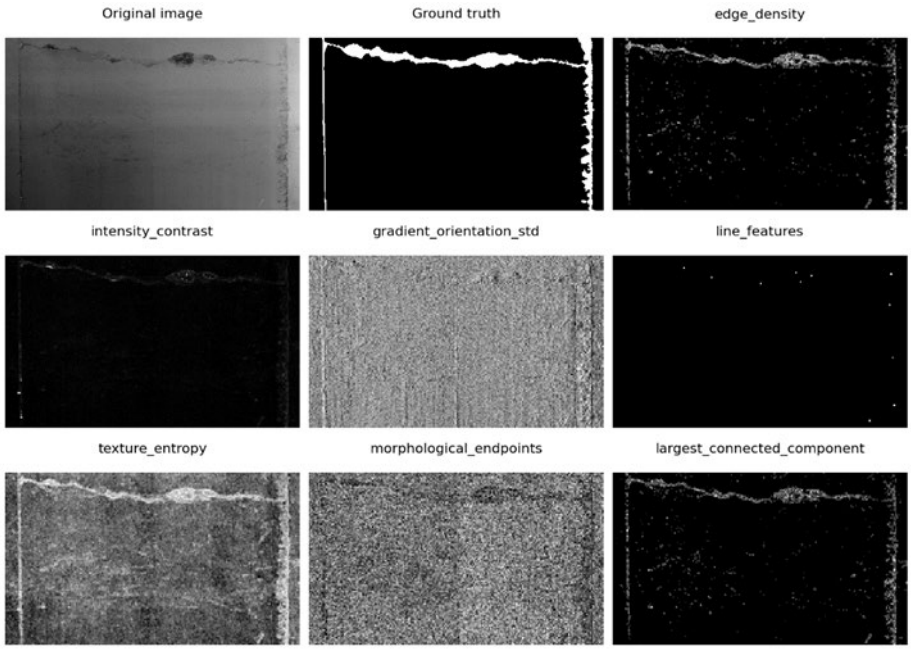


Figure 4.15 Maps of different features based on a sample 3D image

A critical determinant of success in this seed-based technique is the algorithm's ability to accurately assess whether a given cell (or image segment) contains crack pixels. In order to explore the most relevant features of a cell to its probability of containing a crack, we looked into 7 features that characterize the appearance of cracks, including edge density, gradient orientation consistency, line features, intensity contrast, texture analysis, morphological characteristics, and connectivity, as shown in Figure 4.15. Based on the comparison of different features, three of the features were selected for crack cell classification, including edge density, intensity contrast, and texture entropy.

Figures 4.16 and 4.17 show the individual feature maps and final detection of ACP and JCP, respectively. The final detection is the aggregation of the three individual feature maps in an equal fashion. According to these figures, cracks are most visible in the edge density map among the three feature maps. This phenomenon is likely because edge density directly measures the presence and concentration of edges within each cell, which naturally aligns with the physical characteristics of cracks. The intensity contrast map, by detecting variations in pixel brightness, can reveal cracks that result in significant contrast differences with their surroundings. Texture entropy, quantifying the complexity or randomness in the image texture, can help identify areas where the regular pattern of the surface is disrupted by cracks. However, these latter features can

be influenced by many other factors and are not exclusively related to cracks, making the feature maps prevalent with irrelevant information.

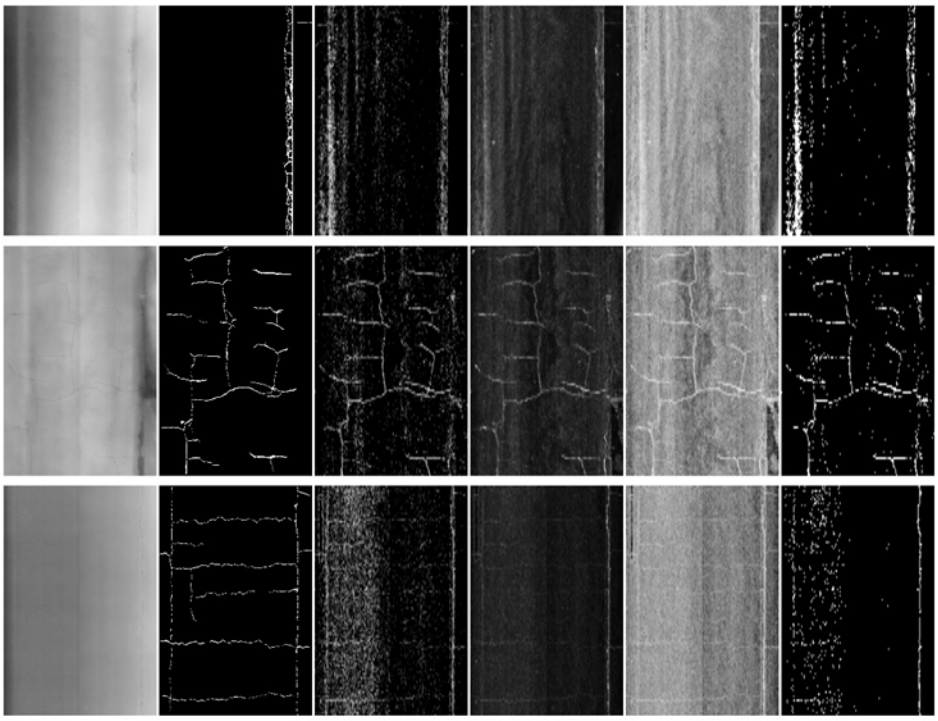


Figure 4.16 Samples of segmentation results using seed-based crack detection methods (from left to right: original ACP 3D image, ground truth, edge density map, intensity contrast map, texture entropy map, and final detection)

Despite integrating information from three distinct feature maps, the presence of significant false positives and false negatives in the final detection in these figures underscores the complexity of automatic crack detection. These inaccuracies can stem from a variety of sources, including the inherent limitations of the selected features, which may not be exclusively or sufficiently sensitive to cracks, leading to their misidentification amid similar patterns or textures in the image. Moreover, the method of aggregating these features, especially if done with equal weighting, may not accurately reflect the relative importance or reliability of each feature in identifying cracks, thereby diluting the effectiveness of more pertinent signals. Additionally, the quality and characteristics of the input images, such as varying lighting conditions, background noise, and the texture of the material being analyzed, can further complicate the distinction between crack and non-crack elements. These challenges highlight the delicate balance required in feature selection, the sophistication needed in feature aggregation methods, and the importance of preprocessing steps to enhance image quality for more accurate crack detection.

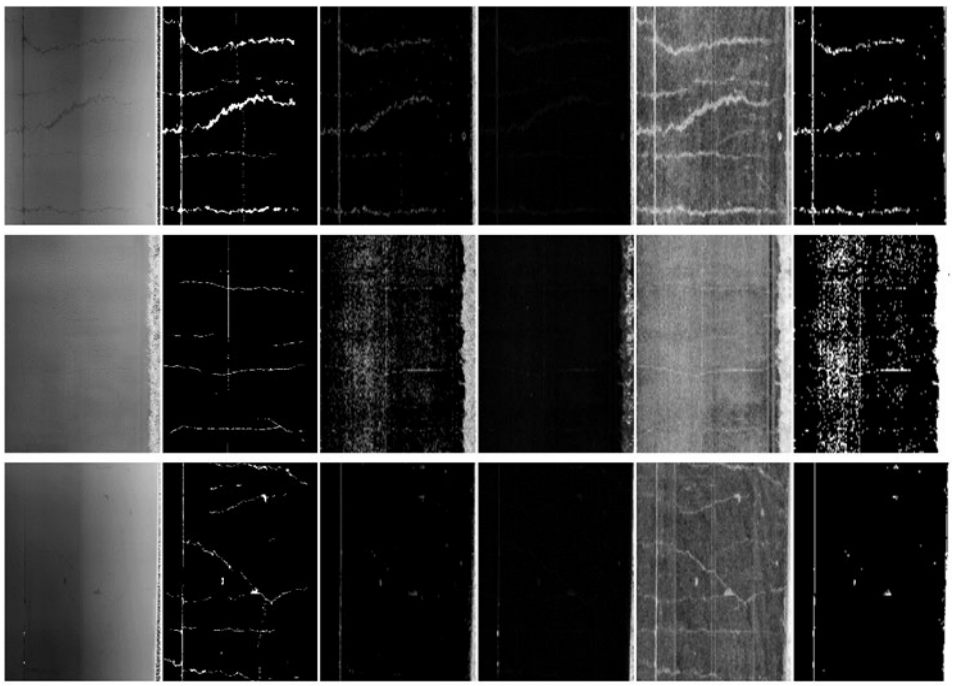


Figure 4.17 Samples of segmentation results using seed-based crack detection methods (from left to right: original JCP 3D image, ground truth, edge density map, intensity contrast map, texture entropy map, and final detection)

4.3.4 Multiscale wavelets

The process of applying multiscale wavelets involves three major steps. First, the wavelet transformation is applied to the image at different levels to decompose the image into approximation and detail coefficients that capture various frequency bands. This decomposition allows for the isolation of features at different scales, where each level of decomposition reduces the image size and increases the abstraction level, thereby focusing on more significant structures over finer details. Following the decomposition, the next step involves the reconstruction of the image from the approximation coefficients obtained at each level. This step is crucial as it generates images that emphasize the major structures at varying scales, making it easier to identify and process edges corresponding to these structures. The third step is to apply Canny Edge detection to the reconstructed images. The application of edge detection at this stage highlights the edges within the multiscale images, with parameters potentially adjusted to suit the characteristics of each reconstructed image.

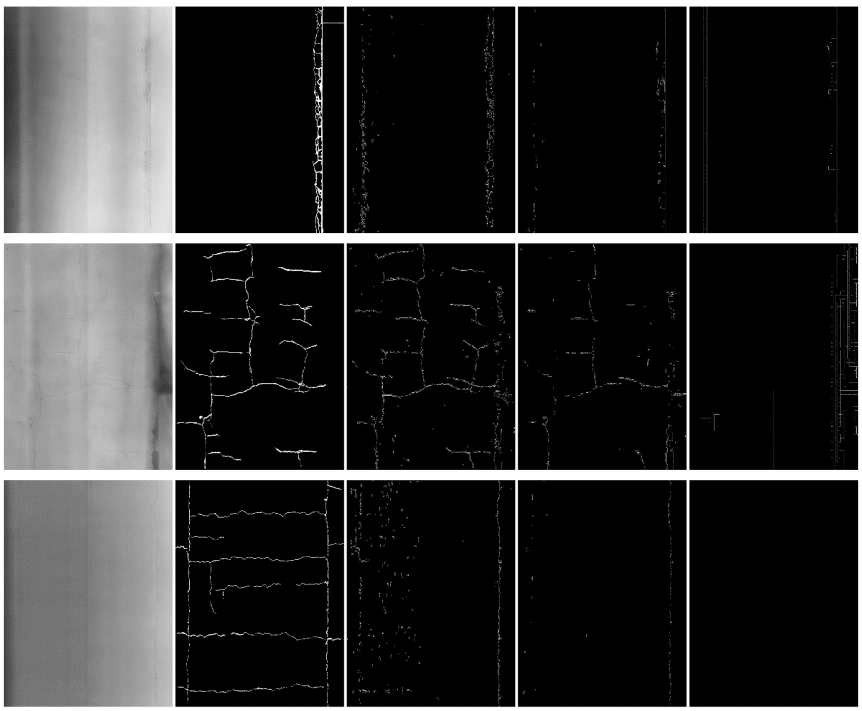


Figure 4.18 Samples of segmentation results using multiscale wavelets edge detection method (from left to right: original ACP 3D image, ground truth, edge detection at level 2^0, edge detection at level 2^1, and edge detection at level 2^2)

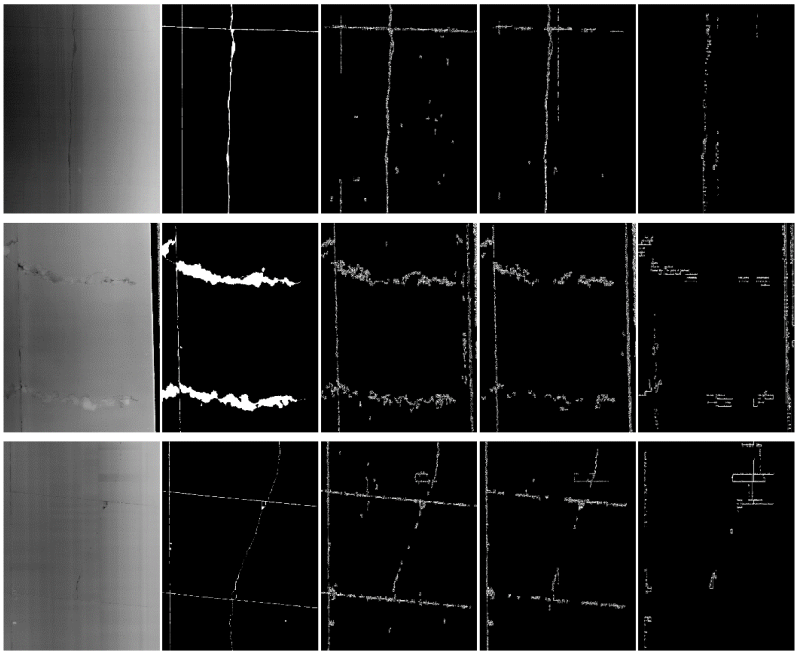


Figure 4.19 Samples of segmentation results using multiscale wavelets edge detection method (from left to right: original JCP 3D image, ground truth, edge detection at level 2^0, edge detection at level 2^1, and edge detection at level 2^2)

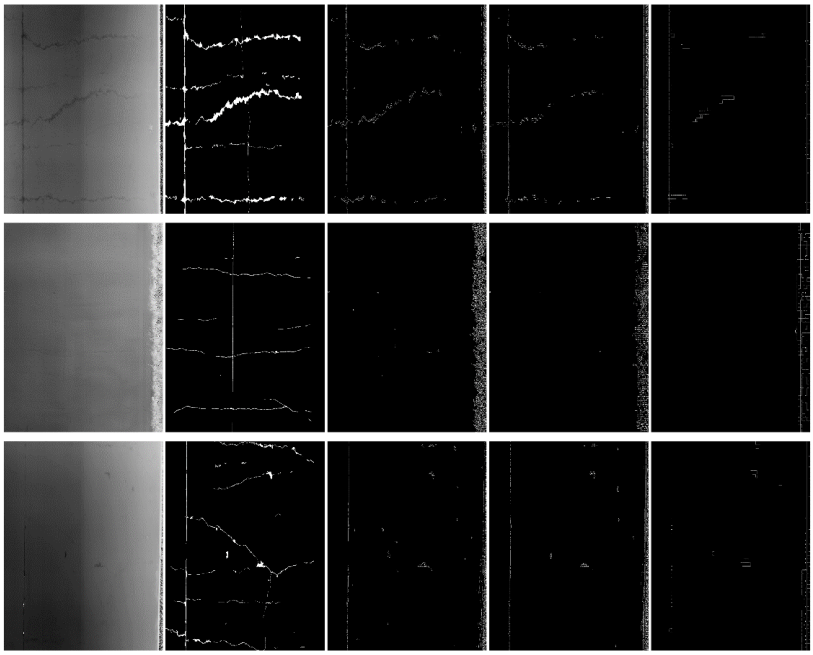


Figure 4.20 Samples of segmentation results using multiscale wavelets edge detection method (from left to right: original CRCP 3D image, ground truth, edge detection at level 2^0, edge detection at level 2^1, and edge detection at level 2^2)

Figures 4.18 to 4.20 show the segmentation results of ACP, JCP, and CRCP, respectively. For each original image, three levels of images are decomposed to, which are levels 2^0, 2^1, and 2^2. The level 2^0 means no decomposition. It was observed that almost all cracks predicted on level 2^1 and level 2^2 images are also predicted on level 2^0 images, which suggests that while the method might be capable of identifying major structural defects, it does not significantly enhance the detection of finer details beyond what is observable in the original images. Besides, the deficiency of detecting thin cracks that prevail in the aforementioned methods still exists with the multiscale wavelet method. Despite the inherent advantages of multiscale wavelets in capturing and analyzing features across different scales, the results indicate that this approach may not be ideally suited for the test dataset, particularly concerning the detection of thin cracks.

4.3.5 Discussions

4.3.5.1 Challenges with pavement surface images

Despite the wealth of data contained within pavement surface images, many challenges persist when utilizing this information for the identification of pavement distress. These challenges stem from the complexity of interpreting features within the images that are not always straightforward indicators of distress. The following discussion details three primary factors that impose difficulties for accurate pavement distress identification.

Inherent Complexity of Distress Patterns Due to Various Formation Mechanisms. The inherent complexity of distress patterns stems from their varied formation mechanisms and deterioration patterns. Pavement distress can originate from multiple sources, such as thermal

fluctuations causing cracks, overloading leading, or water damage. Each type of distress exhibits a distinct pattern based on its underlying cause. Compounding this complexity, the deterioration of pavement does not follow a singular narrative, given the various geographical and environmental situations. The deterioration is influenced by a myriad of factors including the quality of materials, construction practices, environmental conditions, and the volume and type of traffic. These diverse mechanisms result in unique patterns of wear and failure, which can evolve in unpredictable ways over time, making the task of automated pavement distress identification highly challenging. Algorithms must be designed to interpret these complex patterns and discern between the different stages and types of pavement deterioration.

Background Noises. Background noise in pavement images encompasses a range of features that are not distress but may interfere with their detection. Pavement textures vary greatly depending on the material composition and the finish of the surface when laid, as shown in Figure 4.21. These textures can create patterns that are easily confused with cracking or rutting. Road markings, designed to guide traffic, can also be mistaken for linear distress patterns by automated systems. Additionally, tire marks, oil spills, and various stains left by vehicles often resemble cracking or patching, leading to potential false positives in distress identification. Effective segmentation must filter out these various forms of background noise, distinguishing them from genuine distress indicators.

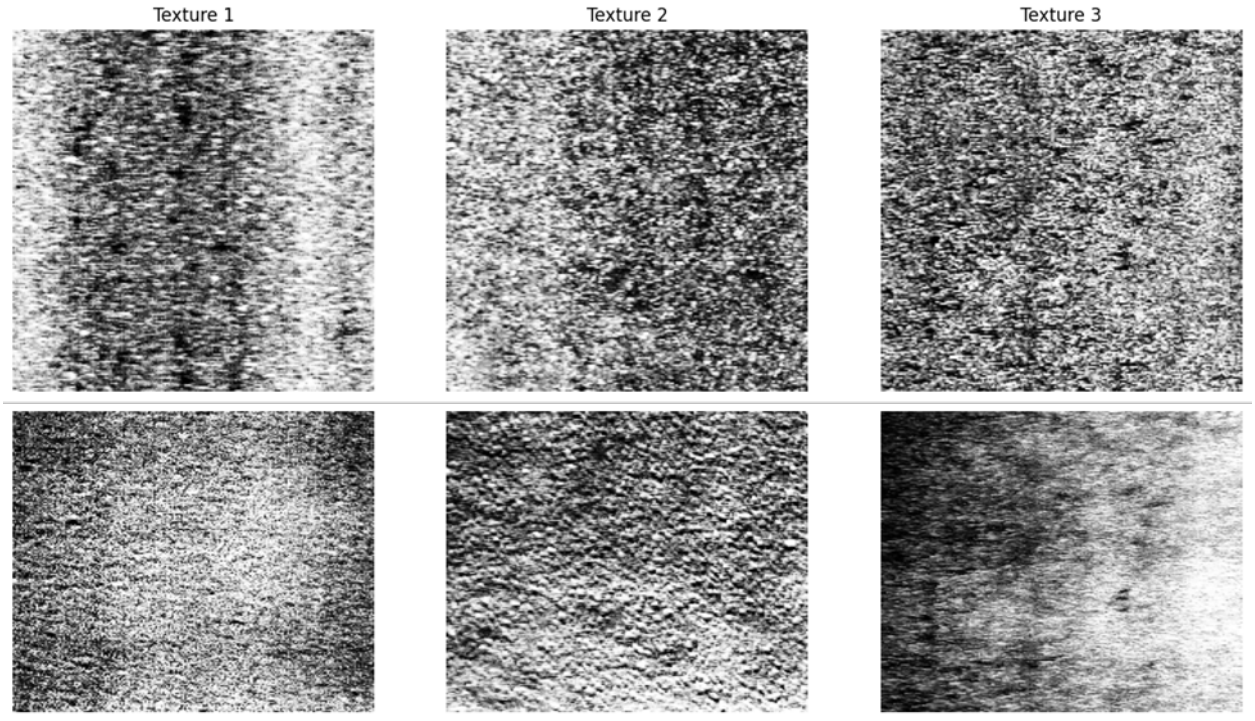


Figure 4.21 Representations of different ACP textures (top row: 2D images, bottom row: 3D images)

Sensor-induced noise. Sensor-induced features can significantly affect the quality and usability of pavement images for distress analysis. Image resolution is a critical factor; higher resolutions capture more detail, which is necessary for detecting small distresses but can result in

prohibitively large data sets. The value range that a sensor can capture determines how well it can differentiate between subtle variations in pavement conditions, which is crucial for identifying early-stage distresses. Figure 4.22 shows a transverse crack collected by two different sensors. The longitudinal/transverse resolution of one sensor was set to 8/2.75 mm, while the other was set to 5/1mm. In Figure 4.22, a crack is represented as fragments or black spots in the low-resolution image, while it is visualized as a continuous line with several spalls in the high-resolution image. According to the figure, crack features can be different across images with different resolutions. Sensor noise, which can be introduced by a variety of environmental and technical factors, can obscure distress features or create false patterns that mimic distress. Addressing these sensor-related issues often requires pre-processing steps to standardize images and filter out noise without losing important distress information.

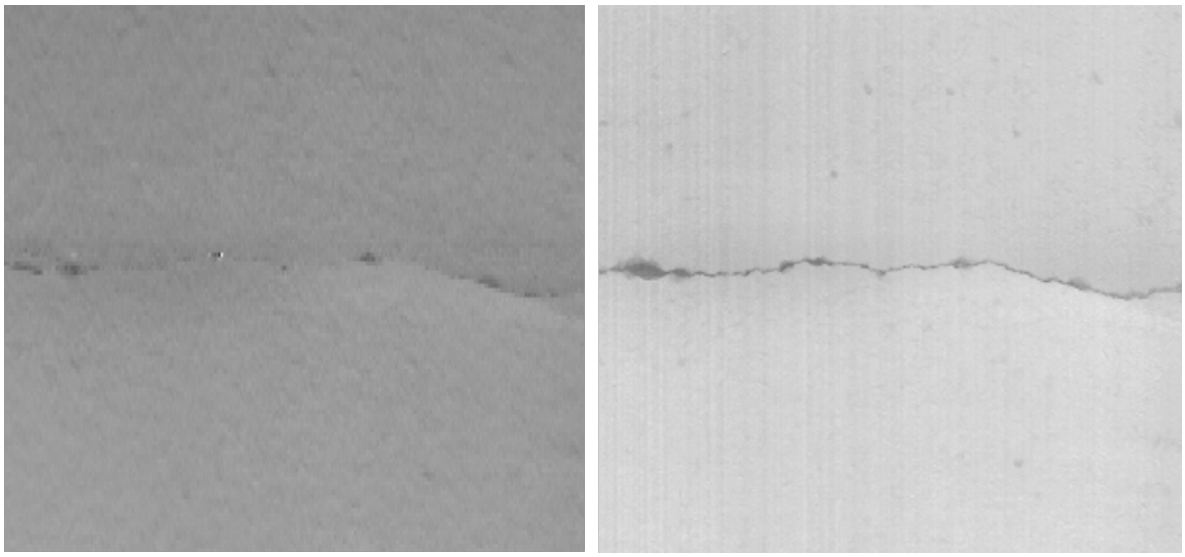


Figure 4.22 A transverse cracking collected by two different sensors (longitudinal/transverse resolution of the left image: 8/2.75 mm; longitudinal/transverse resolution of the right image: 5/1 mm)

The intricacies of these challenges are not isolated, they often interact with each other, compounding the difficulty of the problem. For instance, comparatively thin cracks can blend into a highly textured pavement surface, which is a common scenario with sealcoat pavements. Addressing these intertwined challenges may require sophisticated, integrated approaches that combine advanced image processing techniques and a deep understanding of pavement engineering.

4.3.5.2 Limitations of rules-based image processing methods

Based on the results of the aforementioned experiments, the limitations of rules-based image processing methods can be broadly classified into two categories: 1) diminished efficacy in complex scenarios and 2) limited generalizability.

Due to the deficiency of applied rules-based image processing methods, cracks cannot be accurately identified in some complex scenarios. Almost all the methods applied in the study struggled to identify thin cracks, especially when these cracks are embedded in highly textured pavement surfaces. The subtle nature of such distresses often falls below the detection threshold of the algorithms. On the other hand, certain pavement textures can be falsely identified as cracks by the applied algorithms due to crack-like features, as shown in Figure 4.2 and Figure 4.6. Increasing the sensitivity of these methods to detect more cracks often results in a higher rate of false positives. Achieving a balance where the detection of cracks is enhanced without introducing additional errors remains a significant challenge. This suggests that a reliance solely on intensity and edge features is insufficient. More nuanced features, perhaps capturing the broader context or specific characteristics of pavement distress, are needed for more accurate detection.

Another challenge is that the parameter setting of the rules-based image processing methods cannot be generalized to large-scale datasets. Such parameter settings are usually manually decided by experience or through a test of small datasets. Adaptive thresholding methods demonstrated relatively good performance due to their capability to adjust local thresholds automatically. However, the determination of the neighborhood size for calculating these thresholds required manual selection. This parameter setting can be highly specific to the particularities of each image, making it difficult to apply a single setting effectively across a diverse dataset. Similarly, edge detection methods depend on the robust selection of gradient thresholds. These thresholds are vital for distinguishing between actual distress features and background noise. According to the experiment, none of the features, including mean gradient and median gradient, are sufficient to generate the proper thresholds.

In summary, these challenges highlight the need for more sophisticated image processing techniques that can adapt to the variability inherent in pavement surfaces. There is a clear indication that beyond basic intensity and edge detection, additional features and more advanced algorithms are necessary for effective pavement distress analysis. This may include machine learning-based methods that can learn from a wide range of data and generalize across different pavement conditions, leading to more accurate and reliable distress segmentation.

4.4. Summary

This chapter presents an in-depth analysis using various image processing techniques for pavement distress segmentation. The primary objectives are to explore the capabilities and limitations of rules-based distress identification methods and main challenges on distress identification with digital images. Four rules-based image processing techniques, thresholding, edge detection, seed-based crack detection, and multiscale wavelets were selected to address the pavement distress segmentation problem. The findings are as follows:

- Inadequate detection of thin cracks. A common shortfall across the methods was their general failure to accurately segment thin cracks. This issue highlights a significant gap in the sensitivity of current image processing techniques, as thin cracks are often early indicators of underlying pavement issues.

- Pavement texture leads to false positives. The natural texture of pavement surfaces posed a considerable challenge, especially in distinguishing genuine distresses from textural features. This issue was particularly pronounced, leading to a high rate of false positives where normal variations in pavement texture were misclassified as distresses.

- Lack of generalization across diverse datasets. The parameter settings for each method struggled to adapt to the variability inherent in different pavement conditions and distress types. This limitation underscores the difficulty of developing a universally applicable approach with the current methods, as they require extensive manual tuning to be effective across various datasets.

- While each method holds potential in specific contexts, their limitations underscore the need for further innovation and development in pavement distress segmentation technologies. The inability to effectively detect thin cracks, the challenge of distinguishing distress from pavement texture, and the lack of generalization capability call for a more adaptable, intelligent approach, potentially leveraging advancements in machine learning and artificial intelligence to overcome these hurdles.

Chapter 5 Development of Artificial Intelligence Models

This chapter's primary purpose is to report the development of machine learning (ML)/deep learning (DL)-based models for pavement distress measurement for different pavement types. More specifically, this task focuses on the experiments to select the most promising ML method and develop suitable ML algorithms based on which the distress measurement models are developed. Two types of distress measurement tasks are focused on: distress detection and distress segmentation. An update for the AI/ML model developments is provided in Chapter 7.

5.1 Objectives

In the past couple of years, pavement crack detection research has focused on adapting and improving state-of-the-art ML/DL to identify cracks accurately and efficiently.

The work stems from significant advances in the field of computer vision in the past decade in object detection using Deep Neural Networks (DNN). Several of those DNN networks were shown to be quite reasonable in detecting cracks: YOLO (You Only Look Once) revolutionized the field with its unified approach, predicting bounding boxes and class probabilities directly from full images in a single forward pass, enabling real-time processing while maintaining high accuracy, developed by Redmon et al. (Redmon et al., 2016). Subsequent versions of YOLO, such as YOLOv2 (Redmon and Farhadi, 2018), YOLOv3 (Redmon et al., 2018), YOLOv4 (Bochkovskiy, 2020), and beyond have incorporated advanced techniques to achieve better performances, including multi-scale prediction, improved network architectures, and data augmentation methods. Among the YOLO family, YOLOv5 is widely adopted for applications and serves in many studies as the base model for developing new approaches. The latest version was proposed by Wang et al. (Wang et al., 2024), which proposed the concept of Programmable Gradient Information (PGI) for more reliable training. The Mask R-CNN utilizes a Region Proposal Network (RPN) to generate potential object regions, which are then refined by a Fast R-CNN detector. This technique achieves remarkable precision in real-time object detection and predicts segmentation masks in parallel with bounding box recognition He et al. (He at al., 2017). Cascade R-CNN further enhances detection performance by employing a multi-stage object detection architecture that progressively improves the quality of detected objects through a series of detectors trained with increasing IoU thresholds introduced by Cai and Vasconcelos (Cai and Vasconcelos, 2019). The U-Net's encoder-decoder architecture enables precise localization (Ronneberger et al., 2015), and the DeepLab employs convolution and pyramid pooling modules to capture multi-scale context Chen et al. (Chen et al., 2017).

In this chapter, both state-of-the-art semantic segmentation and object detection methods are investigated to explore suitable methods specifically for pavement distress detection using standard 2D/3D images. The specific objectives are to:

1. Develop a semantic segmentation model that is able to segment 2D/3D images into crack/non-crack regions pixel-wise.

2. Develop a viable ML/DL detection model that is capable of detecting a wide range of distress classes over different pavement surface types.

3. Perform a comprehensive analysis of the performance of the developed models

5.2 Methodology

5.2.1 Distress segmentation methods

The rise of deep learning models has enabled more robust segmentation approaches that can capture complex patterns and variations in images. State-of-the-art methods like UNet have proven effective in handling biomedical image segmentation, and their adaptability has led to their integration with other powerful architectures. For example, combining UNet with frameworks like Fully Convolutional Networks (FCN) and DeepLabV3 has shown great promise in improving seg-mentation performance. Additionally, modern approaches like PSPNet, Segformer, and DDRNet bring a more sophisticated understanding of spatial information and feature representation. These architectures, along with DeepLabV3+ and SegNet, offer varying strengths in accuracy, speed, and complexity, providing diverse tools for pavement distress detection and pavement surface analysis. The following sections will delve into the specifics of these prominent segmentation models: UNet, DeepLabV3, PSPNet, DDRNet, Segformer, and SegNet, highlighting the comparative advantages. In Figure 5.1, UNet is a convolutional neural network architecture designed for image segmentation, originally developed for biomedical applications (Ronneberger et al., 2015) but widely used across various domains. It features a symmetric encoder-decoder structure, where the encoder captures context through convolutions and max-pooling layers, reducing spatial dimensions while increasing feature complexity. The decoder reconstructs the image resolution using up-convolutions, complemented by skipping connections that link corresponding layers in the encoder and decoder, preserving spatial details. This architecture enables precise segmentation by combining abstract and high-resolution features, making UNet effective for tasks like satellite image analysis, autonomous driving, and general object segmentation.

DeepLab is a semantic image segmentation architecture that enhances accuracy by leveraging atrous convolution to capture multi-scale contextual information without reducing spatial resolution (Chen et al, 2017). It introduces the Atrous Spatial Pyramid Pooling (ASPP) module (Figure 5.2), which samples input features at multiple scales through parallel atrous convolutions with varying dilation rates 2. This architecture simplifies previous models by focusing on a powerful encoder-only structure, often built on top of backbone networks like ResNet or Xception, making it efficient and effective for tasks such as autonomous driving and medical image analysis.

Pyramid Scene Parsing Network (PSPNet) is a deep learning architecture designed for semantic segmentation, which involves classifying each pixel in an image (Zhao et al., 2017). Its key innovation is the Pyramid Pooling Module (PPM), which captures contextual information at multiple scales by pooling feature maps at different levels, allowing the network to understand both local and global contexts. PSPNet typically uses a pre-trained ResNet as its backbone for

feature extraction and then aggregates multi-scale features to produce a detailed pixel-wise classification. Known for its strong performance on benchmark datasets like PASCAL VOC and Cityscapes, PSPNet is widely used in applications such as autonomous driving and medical image analysis.

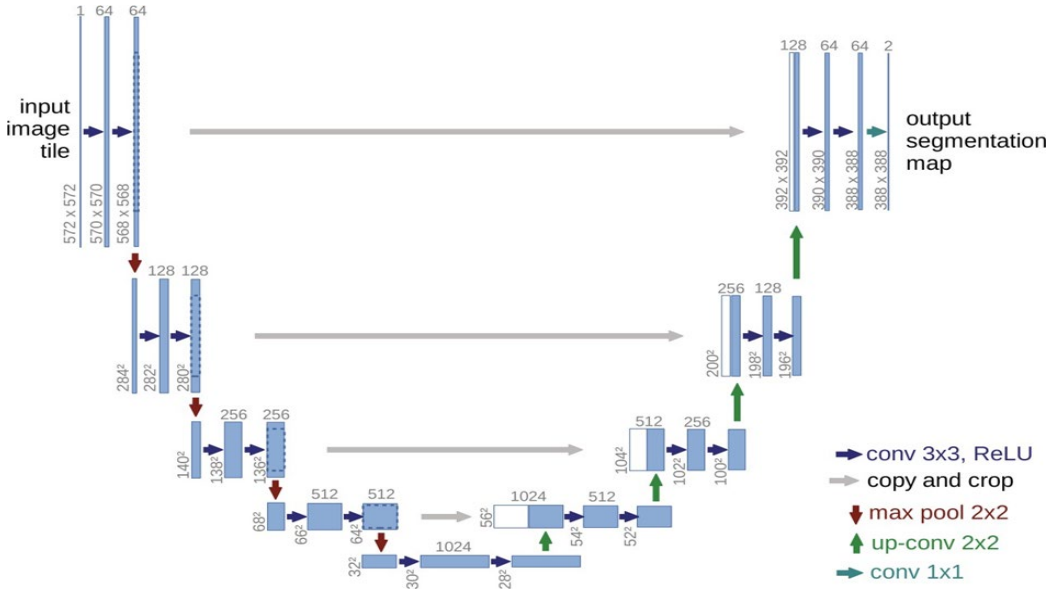


Figure 5.1 U-Net architecture (Ronneberger et al., 2015)

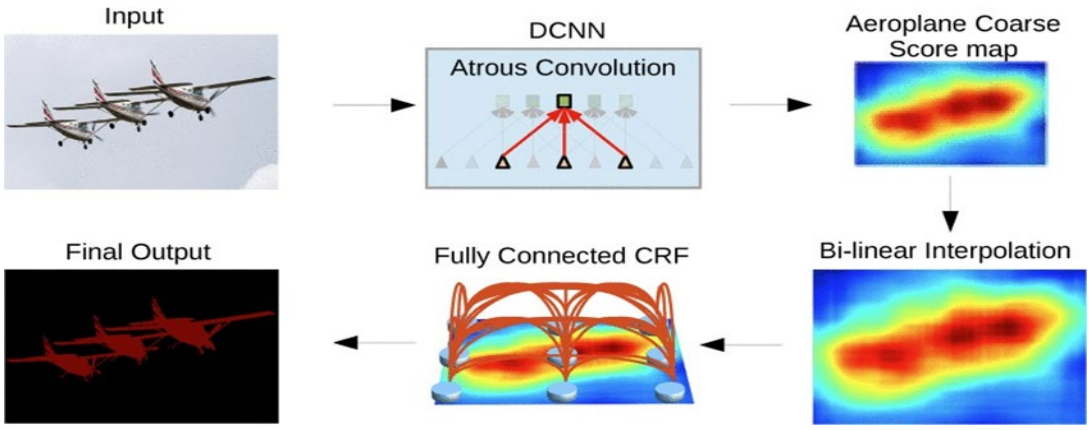


Figure 5.2 Atrous Convolution (Chen et al, 2017)

Deep Dual-resolution Network (DDRNet) is a deep learning architecture designed for se-mantic segmentation, excelling in scenarios where both accuracy and real-time performance are needed (Hong et al., 2021). It features a dual-resolution design, processing images through parallel high-resolution and low-resolution streams to capture fine details and broader contextual information. These streams are iteratively fused, enabling the network to aggregate multi-scale context effectively. Despite its detailed processing, DDRNet is optimized for real-time applications. It is ideal for tasks like autonomous driving and video analysis, where maintaining high-resolution outputs with low latency is essential.

Segformer is a transformer-based architecture for semantic segmentation that combines efficiency, scalability, and accuracy (Xie et al., 2021). Unlike traditional CNNs, Segformer utilizes a transformer backbone, which excels at capturing both local and global features across an image. Its hierarchical design processes images at multiple resolutions, enabling it to maintain high accuracy across different scales while remaining lightweight and efficient, suitable for deployment in resource-constrained environments. Segformer avoids the need for positional encoding and a heavy decoder module, streamlining the architecture for faster performance. It has achieved state-of-the-art results on various benchmarks, making it ideal for applications requiring a balance of detail and speed, such as autonomous driving and real-time image analysis.

SegNet is a deep learning architecture designed for semantic segmentation, characterized by its encoder-decoder structure, which efficiently maps input images to segmentation maps (Badrinarayanan et al., 2017). The encoder downsamples the image while capturing essential features, and the decoder upsamples it back to the original resolution using max-pooling indices from the encoder to guide the process. This approach helps retain spatial details and improves segmentation accuracy, especially around object boundaries. SegNet omits fully connected layers, reducing the model's complexity and making it more memory-efficient and suitable for real-time applications like autonomous driving and robotics, where both speed and accuracy are critical.

5.2.2 Distress detection methods

5.2.2.1 One-stage detection

One-stage object detection has emerged as a popular approach in computer vision for its balance of accuracy and speed, making it highly suitable for real-time applications like autonomous driving, video surveillance, and pavement distress detection. Unlike two-stage detectors, which first generate region proposals and then classify them, one-stage detectors bypass the proposal generation step, directly predicting bounding boxes and class probabilities from the input image in a single forward pass. This streamlined process significantly reduces computational complexity while maintaining competitive detection accuracy. The key innovation in one-stage detectors is their ability to perform dense prediction by treating object detection as a simple regression problem.

YOLO, which stands for "You Only Look Once," is the most adopted one-stage object detection model, introduced by Joseph Redmon and colleagues (Redmon et al., 2016). It approaches object detection as a single regression problem, directly predicting bounding boxes and class probabilities from an entire image in one evaluation, making it exceptionally fast and streamlined. YOLO divides the input image into a grid of cells, with each cell responsible for predicting bounding boxes, confidence scores, and class probabilities, all in a single forward pass through the network (Figure 5.3). This efficiency allows YOLO to achieve real-time object detection, which sets it apart from earlier, slower methods like R-CNN. Each grid cell predicts multiple bounding boxes, defined by parameters such as coordinates, dimensions, and confidence scores, along with class probabilities. A custom loss function balances precision in bounding box

predictions and classification accuracy while penalizing incorrect predictions, contributing to YOLO's overall effectiveness and speed.

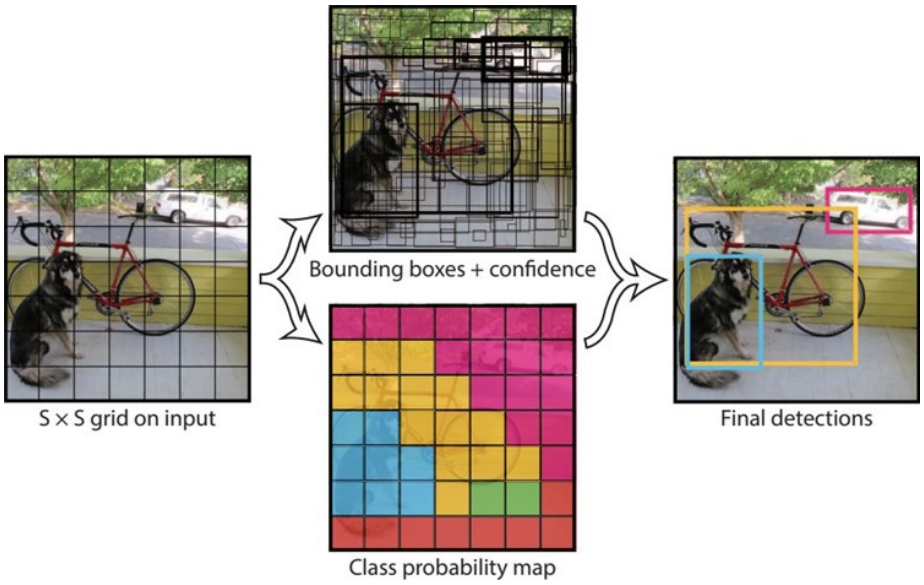


Figure 5.3 YOLO architecture (Redmon et al., 2016)

In YOLO, bounding boxes are calculated through a process that begins by dividing the input image into a grid of cells. Each grid cell is responsible for predicting a fixed number of bounding boxes. For each bounding box, the model predicts five key values: x and y coordinates for the center of the box (relative to the grid cell), the width w and height h of the box (relative to the entire image dimensions), and a confidence score C (Figure 5.4). The confidence score is a product of the probability that the bounding box contains an object and the Intersection Over Union (IoU) between the predicted box and the ground truth. The coordinates x and y are normalized within the grid cell, while w and h are normalized with respect to the image dimensions, and these values are further refined using anchor boxes in later versions of YOLO (starting from YOLOv2). After predicting the bounding boxes, YOLO applies Non-Maximum Suppression (NMS) to filter out redundant boxes, keeping only those with the highest confidence scores, which results in the final object detections. This approach allows YOLO to generate bounding boxes efficiently and accurately in a single forward pass, enabling real-time object detection.

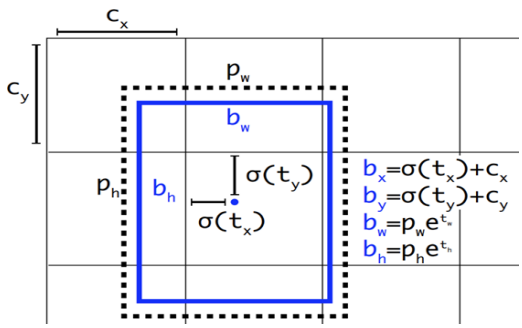


Figure 5.4 YOLO bounding box prediction (Redmon et al., 2017)

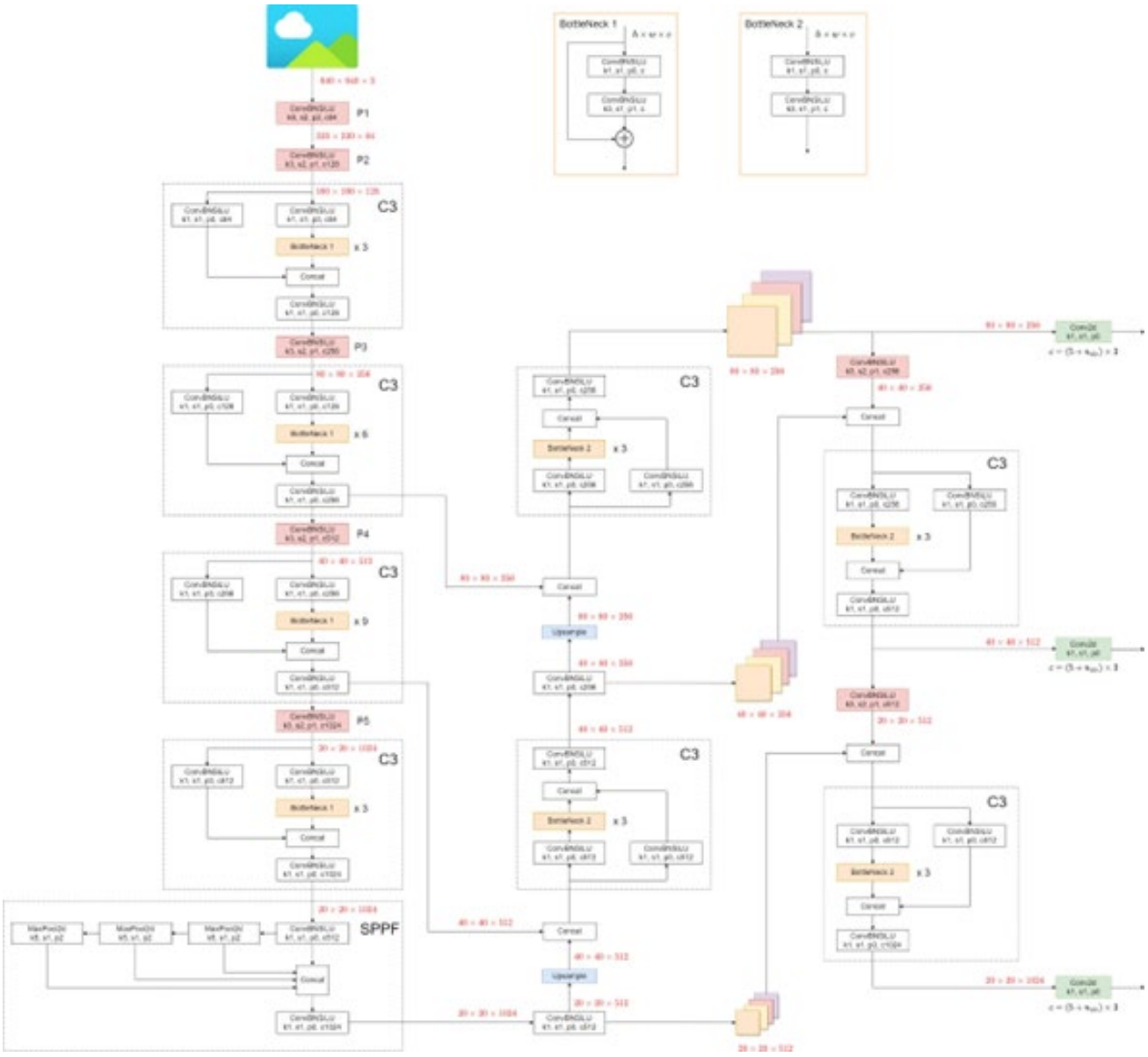


Figure 5.5 YOLOv5 architecture

YOLOv5 and YOLOv8 represent significant advancements in the YOLO series, each bringing innovative changes to the architecture to enhance object detection performance. YOLOv5 introduced the CSPDarknet backbone, which utilizes Cross Stage Partial (CSP) connections to reduce computational cost while maintaining high accuracy (Figure 5.5). It also integrated PANet (Path Aggregation Network) in the neck for better feature fusion across different scales and included automated anchor box learning, which adapts to various datasets. Additionally, YOLOv5 leverages advanced data augmentation techniques like Mosaic, which improves the model's robustness and generalization. YOLOv8 moves towards an anchor-free architecture, which simplifies the detection process by eliminating the need for predefined anchor boxes. This anchor-free approach allows YOLOv8 to directly predict object centers and scales, improving its ability to detect small and overlapping objects more accurately and efficiently.

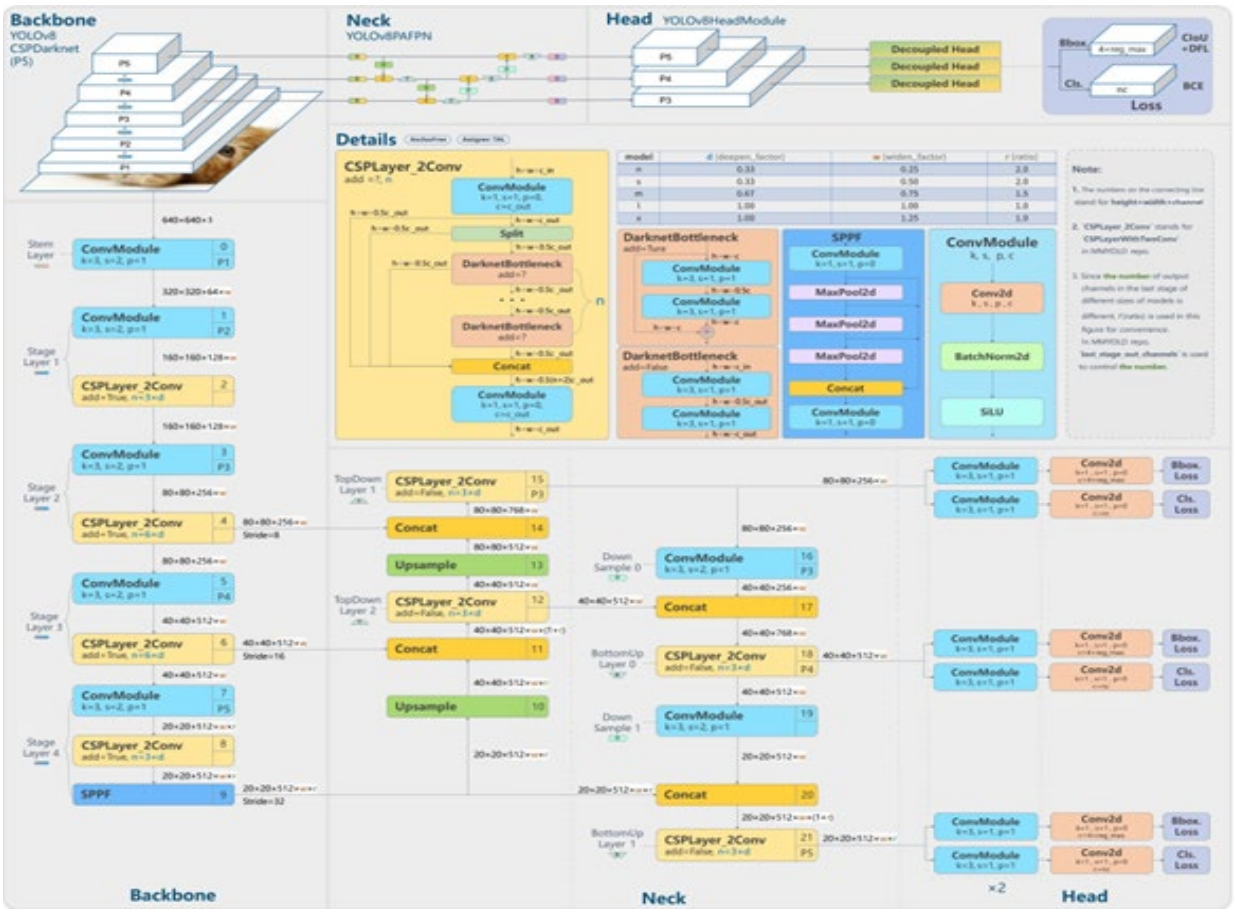


Figure 5.6 YOLOv8 architecture

Another significant improvement in YOLOv8 is the introduction of a decoupled head, which separates the tasks of classification and localization into different branches (Figure 5.6). This decoupling allows the model to optimize each task independently, resulting in better performance in both aspects compared to YOLOv5's coupled head architecture. YOLOv8 also enhances post-processing with dynamic NMS and soft-NMS, which more effectively handle overlapping bounding boxes, reducing the chances of missing detections. Furthermore, YOLOv8 integrates more advanced training techniques, such as dynamic data augmentation and auto-labeling tools, which streamline the training process and improve the model's generalization capabilities. These improvements are anticipated to make YOLOv8 more accurate, efficient, and versatile than YOLOv5, particularly in challenging detection scenarios.

5.2.2.2 Two-stage detection

Two-stage object detection has been a cornerstone of high-accuracy detection frameworks, particularly in applications where precision is paramount. Unlike one-stage detectors, two-stage methods first generate a set of region proposals that potentially contain objects and then refine these proposals through a second stage that classifies the objects and adjusts their bounding boxes. This divide-and-conquer approach allows two-stage detectors to focus on more relevant

areas of the image, achieving superior accuracy compared to their one-stage counterparts, though at the cost of longer inference times.

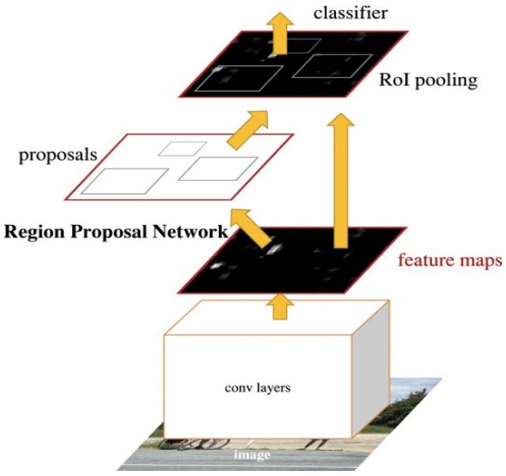


Figure 5.7 Faster R-CNN architecture (Ren et al., 2015)

As shown in Figure 5.7, faster R-CNN is one of the widely applied two-stage models designed for object detection and is known for its efficiency and accuracy. It builds on the R-CNN series by introducing RPN, which generates region proposals directly from the shared feature maps of a backbone network like VGG16 or ResNet (Ren et al., 2015). These proposals are refined through a process called RoI Pooling, which ensures that the proposals are converted into fixed-size feature maps, regardless of their original size. These maps are then processed to classify objects and refine bounding boxes. Faster R-CNN is trained end-to-end, optimizing both region proposal generation and object detection simultaneously, leading to significant improvements in speed and accuracy over previous models. While highly accurate, it is computationally intensive and slower compared to single-shot detectors like YOLO, making it more suitable for applications where precision is prioritized over real-time performance.

In Faster R-CNN, the prediction of bounding boxes is a multi-step process involving both the RPN and the final detection network. The process begins with extracting feature maps from the input image using a CNN backbone, such as VGG16 or ResNet. These feature maps capture essential visual information, including edges, textures, and shapes, which are crucial for detecting objects. The RPN is then used to slide over the feature map and to generate a set of potential bounding boxes, known as region proposals. At each position on the feature map, the RPN considers a predefined set of anchor boxes, which are boxes of various scales and aspect ratios that serve as initial guesses for where objects might be located. For each anchor box, the RPN predicts two things: the objectiveness score and the bounding box adjustment. The objectiveness score is the likelihood that the anchor box contains an object versus a background. The bounding box adjustment is a set of four values that adjust the position and size of the anchor box, refining it to fit the object better. These values represent the adjustments to the center coordinates (x, y) and the width and height (w, h) of the anchor box. Just like YOLO, NMS removes redundant proposals, retaining only the most confident ones. The selected region

proposals pass through the RoI Pooling layer next. RoI Pooling converts the varying-sized proposals into fixed-size feature maps to ensure that regardless of the proposal size, the subsequent classification and regression layers receive a uniform input. After RoI Pooling, the fixed-size feature maps are classified in the final fully connected layers.

5.2.3 Proposed new methods

Two new methods address the issues of 1) visibility discrepancy across intensity and range images and 2) distress class imbalance.

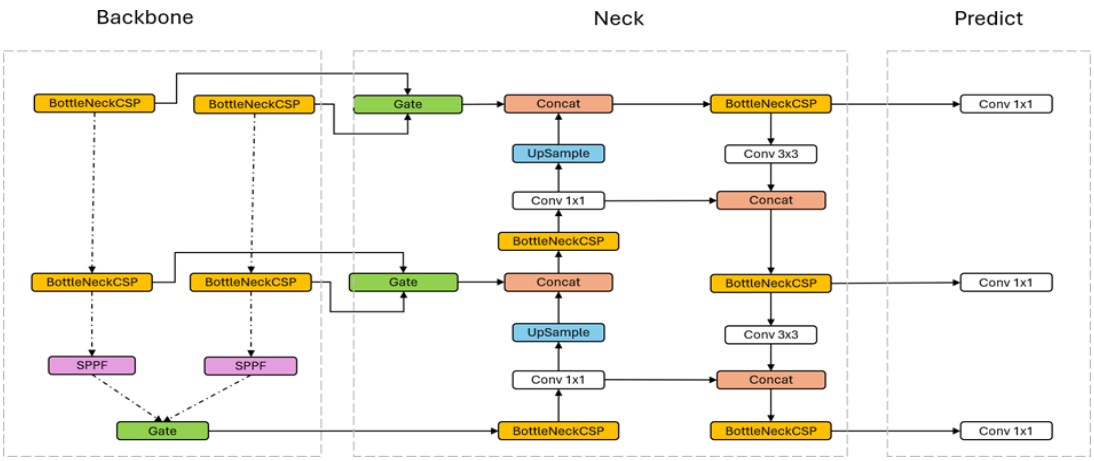


Figure 5.8 Proposed architecture

The dataset includes two distinct types of images: range, which captures the pavement's depth from the sensor, and intensity, which reflects the pavement surface's reflectivity. Some kinds of distress are only identifiable through intensity images, and others through range images. Utilizing both types provides a more holistic view of pavement surface distress and surface conditions. However, merging information from these different image types poses challenges due to limited existing research. Conventional object detection models typically use three-channel RGB images, where each channel represents intensity variations. These models might process and weigh each channel's data differently compared to a combined approach of intensity and range information. In this research, we developed a new model architecture that simultaneously processes and learns from both types of data (as shown in Figure 5.8), enhancing the model's ability to identify distress across both range and intensity images.

Due to the nature of pavement distress class distribution, some of the distress classes may not have enough samples for DL model training. The research team will explore the benefits of using KNN/SVM to address class imbalance issues.

5.3. Experiments

5.3.1 Datasets

They are aligned with the defined objectives. Two distinct datasets are utilized in this project: one for distress segmentation and another for distress detection. Both datasets are initially extracted from the image library developed in Task 3 and are subsequently processed and refined to facilitate the development of deep learning models.

Each image has a resolution of 1,536 × 900 pixels and covers an area of 4.3 meters in the transverse direction and 1 to 8 meters in the longitudinal direction. It should be noted that the driving speed determines the longitudinal length represented by each pixel at the time of image collection. As the survey vehicle's speed increases, a single pixel captures a longer area in the longitudinal direction.

5.3.1.1 Segmentation

The dataset adopted for segmentation includes both 2D and 3D images and is categorized into three groups based on pavement types. Number of images for each group is listed in Table 5.1. Each image is annotated on a pixel level by trained annotators, distinguishing between distress and background regions.

Table 5.1 Segmentation dataset breakdown by pavement type

Dataset	Annotation Count
ACP	114
CRCP	62
JCP	53
Total	229

5.3.1.2 Detection

The detection task is evaluated on a total of 6,922 images for JCP and 5,201 images for ACP at this moment (not the final). It is worth noting that the main principle of annotating distress is whether the distress can be clearly identified in either the intensity image or range image. For example, unsealed cracks tend to be more visualized in range (3D) images, while most sealed cracks can only be identified in intensity (2D) images. Thus, a crack should be annotated as long as it can be identified on either or both images.

Table 5.2 JCP Detection Dataset

Distress Class	Number of Instances
Joint crack	3,534
Longitudinal crack	1,045
Transverse crack	561
Asphalt patch	243
Corner break	44
Failed joint	351
Popout	23
Concrete patch	747
D-cracking	5
Failed concrete patch	38
Sealed longitudinal	150
Punchout	2
Slab edge	7,007
Sealed transverse crack	0
Total	**13,750**

Table 5.2 provides an overview of the JCP Detection Dataset at this moment (not the final), which includes a total of 13,750 instances distributed across various distress classes. Slab edge is the most prevalent distress type, with 7,007 instances, followed by Joint crack, with 3,534 instances, highlighting their significance in jointed concrete pavements. Other distress types, such as Longitudinal crack (1,045 instances) and Concrete patch (747 instances), have moderate representation. In comparison, distress types like D-cracking (5 instances), Punchout (2 instances), and Corner break (44 instances) are severely underrepresented, which may pose challenges for detection models in learning these classes effectively. Notably, the Sealed transverse crack has no instances in the dataset. The imbalance in distress class representation suggests that techniques like data augmentation or class weighting might be necessary to ensure accurate detection across all distress types.

Table 5.3 ACP Detection Dataset

Distress Class	Number of Instances
Sealed longitudinal crack	3,303
Transverse crack	863
Longitudinal crack	724
Sealed transverse crack	1,771
Lane longitudinal crack	1,229
Alligator crack	521
Joint	433
Block crack	134
Pothole	84
Total	**9,062**

Table 5.3 summarizes the ACP Detection Dataset, containing a total of 9,062 instances distributed across various pavement distress classes. Sealed longitudinal crack is the most dominant class with 3,303 instances, followed by Sealed transverse crack (1,771 instances) and Lane longitudinal crack (1,229 instances), which are well-represented in the dataset. Distresses such as Transverse crack (863 instances), Longitudinal crack (724 instances), and Alligator crack (521 instances) offer moderate representation. In comparison, less common classes like Block crack (134 instances) and Pothole (84 instances) have relatively fewer instances. Although there is still some imbalance in the dataset, with a focus on sealed cracks, the ACP dataset is comparatively more balanced than the JCP dataset, which is heavily dominated by Slab edge and Joint crack classes. This relative balance may allow for more consistent model performance across distress types, though additional techniques may still be required to handle underrepresented classes like Potholes and Block cracks effectively.

5.3.2 Evaluation metrics

Precision, Recall, and F1 were usually used to evaluate the model performance. Precision is defined as the ratio of correctly detected objects to all detected objects. Recall is defined as the ratio of correctly detected objects to all actual objects. F1 is a weighted combination of Precision and Recall used to measure the overall performance, as there is always a trade-off between Precision and Recall. These three indicators can be expressed as the following equations:

$$Precision = \frac{TP}{TP+FP} \tag{5.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{5.2}$$

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{5.3}$$

where TP denotes the number of true positives, *FP* denotes the number of false positives, and *FN* denotes the number of false negatives.

Besides the metrics mentioned above, Average Precision (AP) is also used to indicate the overall performance of deep learning models. AP is the weighted mean of precisions at each threshold, which is also considered as the area under the recall/precision curve (Equation 4). AP is often calculated for each class separately in multi-class detection problems. The mean of these AP scores, termed Mean Average Precision (mAP), is then used as a single metric to evaluate the overall effectiveness of the detection model across all classes. mAP is especially significant in datasets with multiple object classes and is widely used in benchmarks and competitions, like the Pascal VOC challenge, COCO challenge, etc. AP considers both false positives and false negatives, providing a balanced view of the model's performance. By integrating across all thresholds, AP provides a single metric that summarily accounts for the trade-off between precision and recall without being tied to a specific decision threshold. Besides, AP can be calculated for individual categories, making it useful for assessing model performance on a per-class basis in complex scenarios with multiple object types.

$$AP = \int_0^1 P(r)dr \tag{5.4}$$

where $P(r)$ is the precision at recall level r.

For segmentation, the following metrics are used during evaluation:

- aAcc (Average Accuracy): Average accuracy of pixel-wise predictions across all classes. It is calculated by summing up the intersection of the prediction and ground truth for all classes and then dividing by the sum of the area of the ground truth for all classes.

$$aAcc = \frac{\sum_i^N TP_i}{\sum_i^N (TP_i + FN_i)} \tag{5.5}$$

- mAcc (Mean Accuracy): Mean accuracy of pixel-wise predictions for each class. It is calculated as the mean accuracy for each class.

$$mAcc = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i} \tag{5.6}$$

- mIoU (Mean Intersection over Union): Mean IoU is a commonly used segmentation metric that measures the average overlap between predicted and ground truth segments across all classes. It is calculated as the intersection over the union for each class and then averaged.

$$mIoU = \frac{1}{N} \sum_{i=1}^N \frac{TP_i}{TP_i + FP_i + FN_i} \tag{5.7}$$

- mDice (Mean Dice Coefficient): Mean Dice measures the similarity between predicted and ground truth segments. It is calculated for each class and then averaged as follows:

$$mDice = \frac{1}{N} \sum_{i=1}^N \frac{2 \times TP}{(TP + FP) + (TP + FN)} \tag{5.8}$$

- mFscore (Mean F-score): Mean F-score is the harmonic mean of precision and recall. It is calculated for each class and then averaged as follows:

$$mFscore = \frac{1}{N} \sum_{i=1}^N \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5.9}$$

- mPrecision (Mean Precision): Mean Precision calculates the average precision across all classes. It is calculated for each class and then averaged as follows:

$$mPrecision = \frac{1}{N} \sum_{i=1}^N \frac{TP}{TP + FP} \tag{5.10}$$

- mRecall (Mean Recall): Mean Recall calculates the average recall across all classes. It is calculated for each class and then averaged as follows:

$$mRecall = \frac{1}{N} \sum_{i=1}^{N} \frac{TP}{TP+FN} \quad (5.11)$$

5.3.3 Preprocessing

5.3.3.1 Segmentation

All datasets were independently split into the train, validation, and test subsets with ratios of 0.7, 0.1, and 0.2, respectively. Because of some model constraints and the dataset size, we decided to split our original images and annotation into a grid of 16. During data loading, the intensity and range images were stacked on top of each other to form a two-channel image. The process is fully outlined in Figure 5.9, while the exact sample split is shown in Table 5.4.

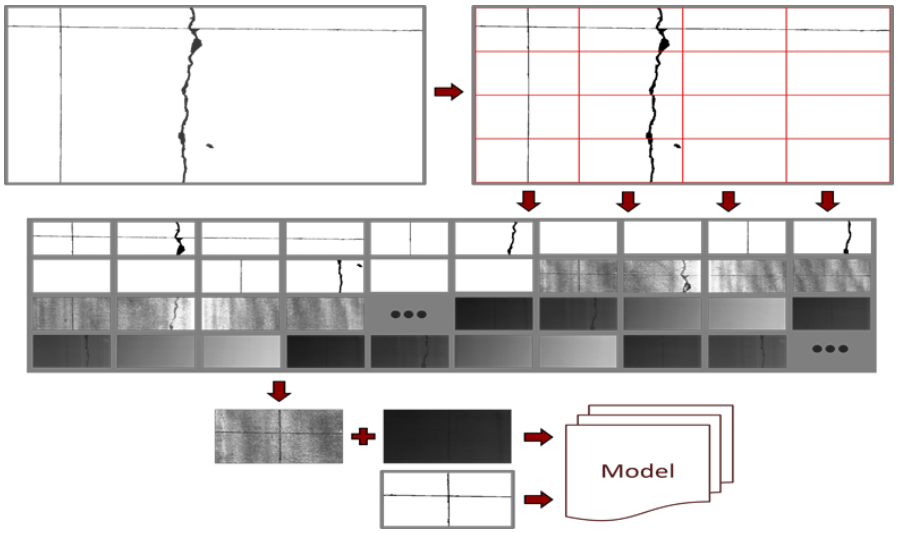


Figure 5.9 Segmentation preprocessing pipeline

Table 5.4 Split of Segmentation Dataset

Dataset	Train	Val	Test	Total
ACP	912	128	288	1,328
JCP	592	80	176	848
CRCP	688	96	208	992

5.3.3.2 Detection

Both the JCP and ACP datasets were independently split into the train, validation, and test subsets with ratios of 0.7, 0.15, and 0.15, respectively. For ACP, the dataset is divided into a

training dataset of 3,640 images, a validation dataset of 780 images, and a test dataset of 781 images. For JCP, the dataset is divided into a training dataset of 4,845 images, a validation dataset of 1,038 images, and a test dataset of 1,039 images. Before using the dataset, the annotations were revised from a machine learning perspective to ensure they would support the optimal performance of the developed models. The number of instances for each distress class across different datasets after revision is shown in Tables 5.5 and 5.6.

The revision of the dataset annotations was necessary due to the use of Google Street View to aid in the annotation process, which may have led annotators to label insignificant distresses. Additionally, since the original images were annotated by a single individual, despite the annotators being well-trained, a second opinion was required to ensure consistency and accuracy. The revision process primarily focused on removing ambiguous annotations that could not be validated without external references like Google Street View. Furthermore, a small portion of the annotations were modified to make the bounding boxes more consistent across the dataset, ensuring that the dataset would better support the performance of machine learning models by reducing errors and inconsistencies in the labeled data.

Table 5.5 Split of ACP Detection Dataset

Class Code	Distress class	Number of Instances		
		Train	Val	Test
0	Transverse crack	614	110	139
1	Joint	315	61	57
2	Sealed transverse crack	1,240	251	280
3	Longitudinal crack	520	95	109
4	Lane longitudinal crack	889	173	167
5	Sealed longitudinal crack	2,339	471	493
6	Block crack	93	24	17
7	Alligator crack	351	84	86
8	Pothole and others	56	13	15
Total		**6,417**	**1282**	**1,363**

Table 5.5 provides a breakdown of the ACP Detection Dataset split into training, validation, and test sets across nine different distress classes. The distress type, Sealed Longitudinal crack, has the highest number of instances, with 2,339 in the training set, followed by 471 in validation and 493 in the test set, indicating a significant representation in the dataset. In contrast, classes like Block crack and Pothole and Others have the fewest instances, making them more challenging for model detection due to their smaller sample sizes. Distresses such as Transverse crack, Longitudinal crack, and Lane Longitudinal crack have moderate representation, ensuring that the model has a balanced distribution of training data for these classes. The total number of instances across all datasets amounts to 6,417 for training, 1,282 for validation, and 1,363 for testing. The distribution of individual distress classes suggests that while some distress classes are well-represented, the model may require techniques to handle the imbalance across different distress types for optimal detection performance.

Table 5.6 presents the split of the JCP Detection Dataset, detailing the number of instances for various distress classes across the training, validation, and test sets. Slab edge and Joint crack are the most represented classes, with 4,910 and 2,511 instances, respectively, in the training set, ensuring sufficient data for model training. In contrast, distress types like Punchout, D-Cracking, and Popout have very few instances, making them challenging for the model to learn due to insufficient data. Notably, the Sealed transverse crack has no instances in the training set, limiting the model's ability to detect this class. Other distress types, such as Longitudinal cracks and Concrete patches, are moderately represented, providing a more balanced foundation for model development. The dataset totals 9,659 training instances, 2,072 validation instances, and 2019 test instances, though the imbalance across distress classes may require more developed techniques to ensure consistent model performance.

Table 5.6 Split of JCP Detection Dataset

Class Code	Distress class	Number of Instances		
		Train	Val	Test
0	Failed joint	219	72	60
1	Corne break	33	6	5
2	Punchout	1	1	0
3	Asphalt patch	170	48	25
4	Failed concrete patch	25	4	9
5	D-cracking	4	1	0
6	Popout	17	3	3
7	Longitudinal crack	741	159	145
8	Sealed longitudinal	114	14	22
9	Concrete patch	533	120	94
10	Transverse crack	381	83	97
11	Joint crack	2,511	524	499
12	Sealed transverse crack	0	0	0
13	Slab edge	4,910	1,037	1,060
Total		**9,659**	**2,072**	**2,019**

5.3.4 Distress segmentation

In this section, we present the results of the distress segmentation experiments conducted on the JCP, ACP, and CRCP datasets. The segmentation experiments aimed to evaluate how well the models perform in identifying and delineating distress areas across these datasets, considering the unique challenges posed by each, such as the uniform background in JCP, the roughness of aggregates in ACP, and the presence of tining in CRCP. Through this comparative analysis, we assess the effectiveness of various segmentation models and highlight the factors that impact their performance across these diverse pavement surfaces.

5.3.4.1 JCP

The experiments on the JCP dataset were conducted using two distinct types of annotations to explore how different labeling strategies impact the performance of segmentation models.

The first type of annotation is a pixel-perfect mask, which precisely marks the entire area of the distress, capturing the full geometry and width of each crack. This method is highly detailed and provides an accurate representation of the actual surface damage, making it ideal for models that require high precision in distress detection. The second annotation type is a one-pixel line, which delineates only the centerline of the crack. This annotation strategy simplifies the labeling process by focusing solely on the central path of the distress, providing less information about the crack's width and boundaries. While this method is more efficient and easier to annotate, it may reduce the model's ability to detect the full extent of the damage. However, it can still be useful for crack detection tasks where the exact shape and width of the crack are not critical.

Several state-of-the-art segmentation models were evaluated on the dataset to compare their performance across various metrics, including aAcc, mIoU, mAcc, mDice, mFscore, mPrecision, and mRecall. The models included DeepLabV3, FCN, PSPNet, and SegNet, each configured with Weighted Cross-Entropy (WCE) loss to address the class imbalance, an issue revealed in data preparation. These models were all trained and tested on the ACP dataset using the same setting described earlier. Additionally, FCN with Focal loss was included in the experiment, which proved to be effective at handling imbalanced datasets by focusing on hard-to-classify examples. Given that the ACP dataset contains a mix of both well-represented and underrepresented distress classes, the inclusion of FCN-Focal was intended to see if Focal Loss could improve performance, especially for minority classes that are harder to detect. The experimental setup aimed to determine which model and loss function combination provides the best performance in terms of segmentation accuracy and detection of various distress types.

Table 5.7 Model performance metrics on JCP dataset

Model	aAcc	mIoU	mAcc	mDice	mFscore	mPrecision	mRecall
DeepLabv3-WCE	99.29	86.20	92.60	92.06	92.06	91.53	92.60
FCN-Focal	99.12	84.17	93.91	90.69	90.69	87.93	93.91
FCN-WCE	99.30	86.13	92.02	92.01	92.01	91.99	92.02
PSPNet-WCE	99.31	86.48	92.51	92.24	92.24	91.98	92.51
SegNet-WCE	98.78	80.37	93.76	87.95	87.95	83.58	93.76

Table 5.8 Model performance metrics on JCP (one pixel) dataset

Model	aAcc	mIoU	mAcc	mDice	mFscore	mPrecision	mRecall
DeepLabv3-WCE	99.73	65.80	76.23	74.10	74.10	72.30	76.23
FCN-WCE	99.74	66.13	75.94	74.48	74.48	73.17	75.94
PSPNet-WCE	99.72	65.59	76.17	73.86	73.86	71.92	76.17
SegNet-WCE	99.54	63.01	83.72	70.83	70.83	65.11	83.72

Table 5.7 shows the metrics over the dataset with the pixel-perfect annotation, focusing on key metrics like mFscore, mPrecision, and mRecall, which are crucial for balancing detection accuracy in binary segmentation tasks with potential class imbalances. In terms of the F1-score, PSPNet-WCE achieves the highest value (92.24%), demonstrating the most balanced performance between precision (91.98%) and recall (92.51%). This suggests that it handles both correct detection and false positives effectively, making it a robust choice for JCP segmentation tasks. FCN-WCE closely follows with an F1-score of 92.01%, reflecting strong all-around performance with the highest precision (91.99%), meaning it is particularly effective in minimizing false positives.DeePLabv3-WCE also performs well, achieving a high F1-score (92.06%) and recall (92.60%), though with slightly lower precision compared to FCN-WCE. FCN-Focal, while scoring lower on F1-score (90.69%), excels in recall (93.91%), indicating that it effectively captures true positives, though it sacrifices precision (87.93%) compared to the other models. SegNet-WCE, despite showing the lowest F1-score (87.95%), still delivers acceptable performance, particularly in recall (93.76%), but its precision (83.58%) is noticeably lower, making it more prone to false positives.

Table 5.8 presents the performance of segmentation models on the JCP dataset using one-pixel annotations, with a focus on the F1-score (mFscore) as a key indicator of performance in this setting. Among the models, FCN-WCE achieves the highest F1-score (74.48%), making it the most balanced model in terms of precision (73.17%) and recall (75.94%), which indicates strong overall performance in segmenting the centerlines of cracks. DeepLabv3-WCE follows closely with an F1-score of 74.10%, showing similarly balanced precision and recall. Meanwhile, PSPNet-WCE achieves a slightly lower F1-score (73.86%) but leads in precision (71.92%), suggesting it is better at avoiding false positives, though slightly less effective in capturing all instances (recall of 76.17%). SegNet-WCE shows the lowest F1-score (70.83%) despite having the highest recall (83.72%), indicating it captures most of the crack centerlines but struggles with precision (65.11%), leading to more false positives. Overall, FCN-WCE stands out as the most reliable model for this one-pixel annotation approach, balancing precision and recall effectively.

The comparison between one-pixel annotations and pixel-perfect annotations on the JCP dataset reveals a clear difference in model performance. With pixel-perfect annotations, PSPNet-WCE achieved the highest F1-score of 92.24%, indicating strong performance with high precision (91.98%) and recall (92.51%), possibly due to the detailed, full-area distress information. In contrast, with one-pixel annotations, FCN-WCE performed best but with a significantly lower F1-score of 74.48%, along with reduced precision (73.17%) and recall (75.94%), reflecting the challenges of using simplified, less detailed centerline annotations. The pixel-perfect approach clearly leads to superior segmentation results, as it provides more comprehensive information about the distress area, while one-pixel annotations, though functional, result in a noticeable drop in accuracy and balance between identifying true positives and minimizing false positives.

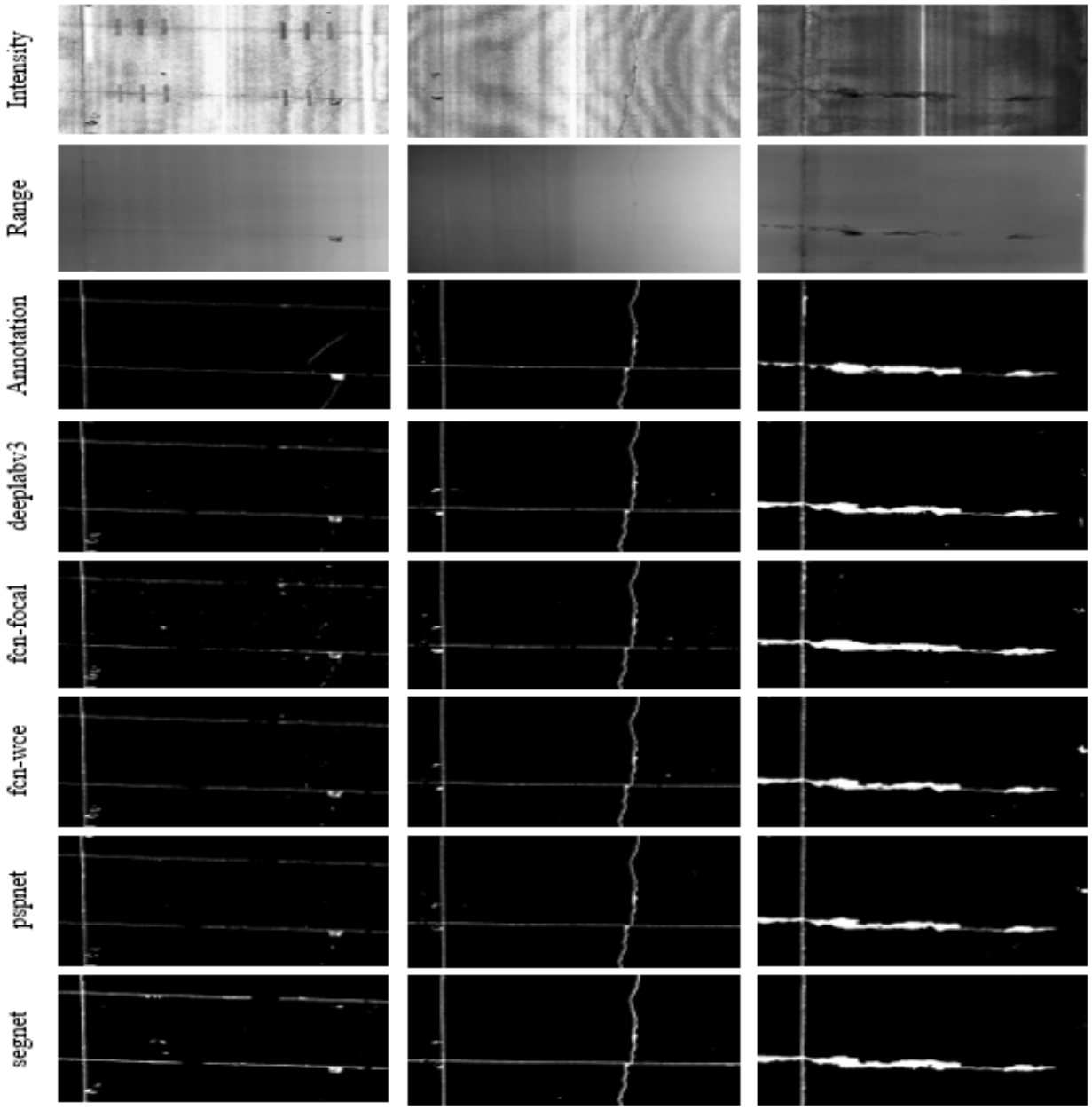


Figure 5.10 Comparison between different JCP samples

Figure 5.10 shows the segmentation results of different models, including DeepLabv3, FCN-Focal, FCN-WCE, PSPNet, and SegNet, using intensity and range images of JCP, where the respective rows are the intensity, range, annotated ground truth, and predictions of the trained models DeepLabv3, FCN-Focal, FCN-WCE, PSPNet, and SegNet, while the columns show the different samples. The first column shows low- and mid-severity cracks. The second column shows mid-severity cracks, and the third column shows high-severity cracks. All models demonstrate the ability to capture most of the mid-and high-severity crack regions, but they underperformed in detecting thin cracks, often missing finer details. DeepLabv3, FCN-WCE, and PSPNet generally show balanced segmentation, while FCN-Focal tends to make false

positive detections. SegNet also shows a tendency to over-segment, resulting in less accurate crack delineation.

5.3.4.2 ACP

Table 5.9 compares the performance of segmentation models on the ACP dataset, with a focus on the F1-score (mFscore) as the most comprehensive metric for this imbalanced binary segmentation task. FCN-WCE achieves the highest F1-score (77.78%), making it the best-performing model for balancing precision (81.57%) and recall (74.85%), ensuring accurate distress detection with minimal false positives and missed detections. DeepLabv3 follows closely with an F1-score of 77.53%, showing strong overall performance. FCN-Focal, while having a slightly lower F1-score (76.44%), excels in recall (80.02%), making it particularly effective at capturing true positives, though at the cost of lower precision (73.67%). PSPNet-wce also performs well with an F1-score of 76.86%, maintaining a good balance between precision and recall. SegNet has the lowest F1-score (76.29%) but still demonstrates acceptable performance across all metrics. Overall, FCN-WCE provides the most balanced performance.

Table 5.9 Model performance metrics on ACP dataset

Model	aAcc	mIoU	mAcc	mDice	mFscore	mPrecision	mRecall
DeepLabv3-WCE	98.13	68.51	75.00	77.53	77.53	80.69	75.00
FCN-Focal	97.55	67.32	80.02	76.44	76.44	73.67	80.02
FCN-WCE	98.18	68.76	74.85	77.78	77.78	81.57	74.85
PSPNet-WCE	98.11	67.86	74.04	76.86	76.86	80.50	74.04
SegNet-WCE	97.66	67.20	78.41	76.29	76.29	74.48	78.41

Figure 5.11 compares the segmentation performance of various models, DeepLabv3, FCN-Focal, FCN-WCE, PSPNet, and SegNet, on pavement distress detection using intensity and range images of ACP, where the respective rows are the intensity, range, annotated ground truth, and predictions of the trained models DeepLabv3, FCN-Focal, FCN-WCE, PSPNet, and SegNet, while the columns show the different samples.

The first column shows mid-severity cracks, the second shows sealed cracks, and the third shows high-severity block cracks. All models captured the skeletons of the cracks effectively, particularly in mid- and high-severity cases, but underperformed in detecting sealed cracks. Sealed cracks can be challenging due to the similarity of intensity between the sealed cracks and the background. FCN-Focal tends to generate more false positives than the other models. The relatively balanced performance of DeepLabv3, FCN-WCE, PSPNet, and SegNet still failed to delineate a completely sealed crack. Compared to similar models applied to JCP, these segmentation models tend to make more false predictions for ACP, highlighting the challenges associated with detecting and differentiating distress types in asphalt surfaces.

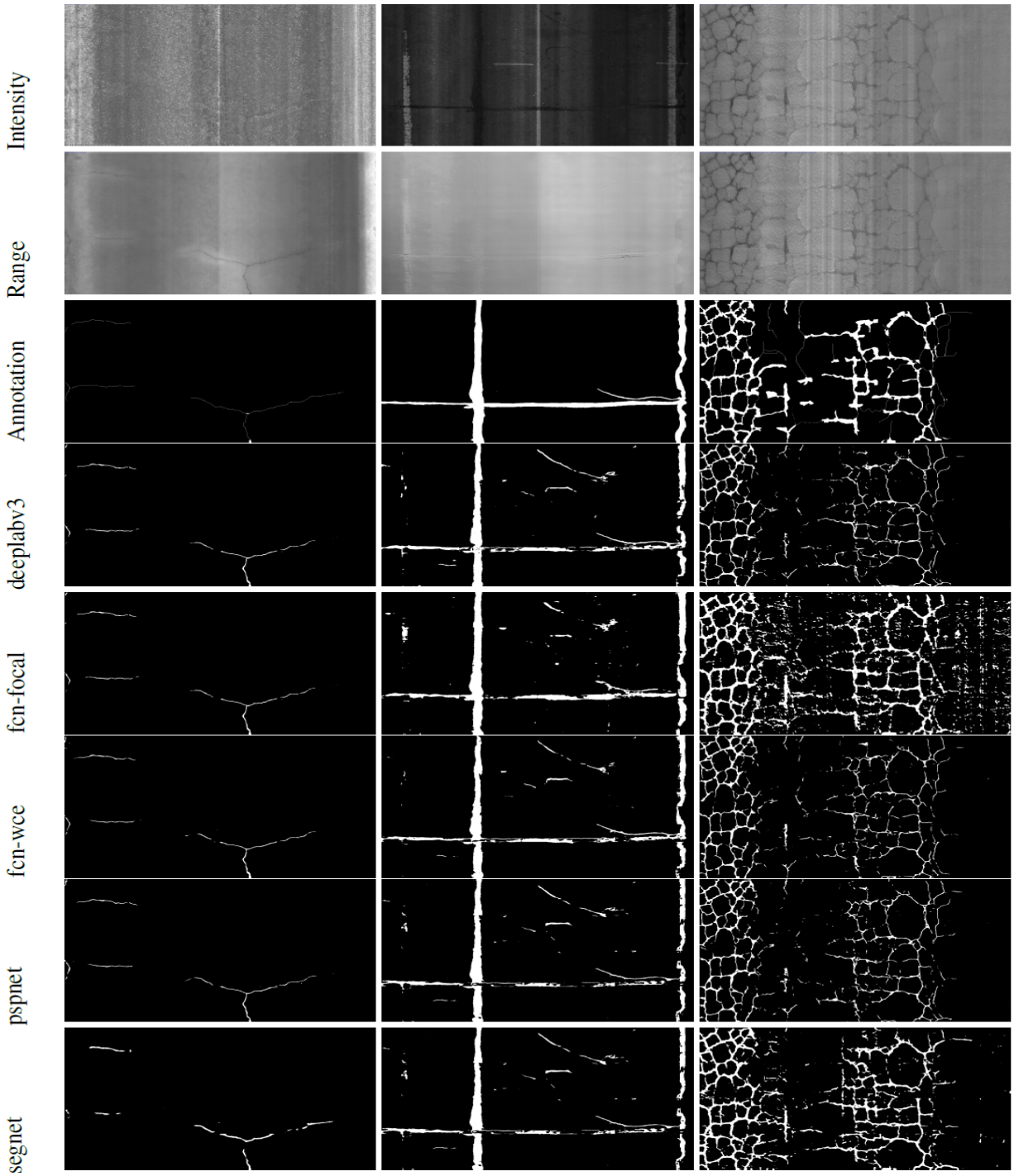


Figure 5.11 Comparison between different ACP samples

5.3.4.3 CRCP

Table 5.10 presents the performance metrics of segmentation models on the CRCP dataset, directly following the experiment settings of the ACP. Similar metrics are used here, including

mFscore), mPrecision, and mRecall, which are particularly important for analyzing the segmentation performance on this dataset. Among the models selected, DeepLabv3 achieves the highest F1-score (77.40%) and precision (83.33%), demonstrating its strong capability in maintaining a balance between correctly detecting distress segments and minimizing false positives. Its mIoU (68.46%) and mDice (77.40%) are also slightly higher than the other models, suggesting that it excels in segmenting the CRCP dataset with better overlap between predicted and ground truth segments. FCN-WCE follows closely, with an F1-score of 77.20% and nearly the same precision (83.34%), indicating a well-balanced performance. It achieves similar mIoU (68.26%) and mDice (77.20%) values, showing comparable effectiveness to DeepLabv3, though with marginally lower recall (73.04%). FCN-Focal, while having the lowest F1-score (76.93) and precision (80.37%), performs slightly better in terms of recall (74.23%), meaning it is more effective at capturing true positives but at the cost of more false positives. Its mDice (76.93%) and mIoU (67.98%) are slightly lower than the other two models, reflecting its slightly lower segmentation performance. Overall, DeepLabv3-we and FCN-WCE deliver better performance on the CRCP dataset, with the highest F1-scores and precision, indicating a good balance between accurate distress detection and false positives. FCN-Focal has a higher recall, lower precision, and overall segmentation accuracy.

Table 5.10 Model performance metrics on CRCP dataset

Model	aAcc	mIoU	mAcc	mDice	mFscore	mPrecision	mRecall
DeepLabv3-WCE	98.42	68.46	73.33	77.40	77.40	83.33	73.33
FCN-Focal	98.30	67.98	74.23	76.93	76.93	80.37	74.23
FCN-WCE	98.42	68.26	73.04	77.20	77.20	83.34	73.04
PSPNet-WCE	98.52	70.04	75.45	78.99	78.99	83.76	75.45
SegNet-WCE	98.12	68.31	78.18	77.33	77.33	76.54	78.18

Figure 5.12 compares the segmentation performance of models on detecting pavement dis-tress in CRCP, where the respective rows are the inten-sity, range, annotated ground truth, and predictions of the trained models DeepLabv3, FCN-Focal, FCN-WCE, PSPNet, and SegNet, while the columns show the different samples. The first column shows sealed cracks, the second column shows low-severity cracks, and the third column shows mid-severity cracks. All models demonstrate the ability to capture the major structure of salient cracks, but there is variation in segmentation quality across crack severity levels. According to the second column of Figure 12, all models failed to segment the fine crack located at the top-center area, while false positives were produced. DeepLabv3 and PSP-Net effectively capture the main crack structures but occasionally suffer from over-segmentation. FCN-Focal frequently produces false positives, especially in low-severity areas, while FCN-WCE tends to under-segment and miss finer crack details. SegNet exhibits a tendency to over-segment, leading to inaccurate, overly thick crack detections.

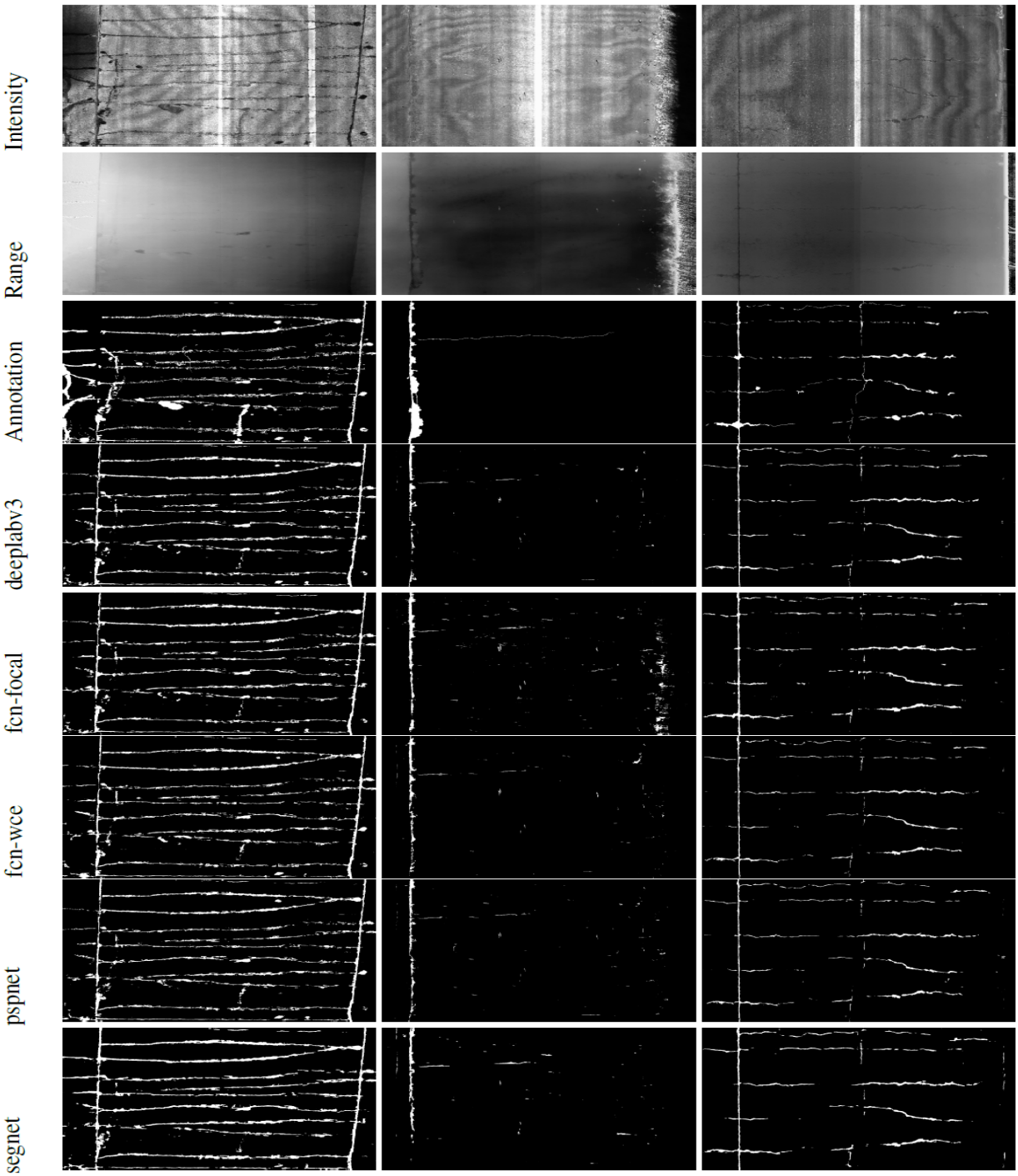


Figure 5.12 Comparison between different CRCP samples

The comparison of the best-performing models across the JCP, ACP, and CRCP datasets shows that the JCP dataset achieved the highest segmentation performance, with the best F1-score of 92.24%. This superior performance is possibly due to the more uniform background in JCP, which makes distress segmentation easier. In contrast, the ACP dataset yielded a lower F1-score of 77.78%, likely due to the rough surface texture caused by varying aggregate sizes, which

makes it harder to distinguish cracks, especially their boundaries. For the CRCP dataset, the best F1-score was 77.40%, slightly lower than ACP, and the presence of tining lines in CRCP may have interfered with crack detection, making it more difficult for models to segment the cracks accurately. Overall, the uniformity of JCP surfaces appears to enhance segmentation performance, while ACP and CRCP present additional challenges due to their surface textures and structural features.

Model speed and hardware utilization are important factors to consider when choosing a model. If a model can only predict ten images per second and the input is hundreds of images per second, the model will lag behind. Likewise, hardware constraints can determine how expensive a model is to run in the real world. Judging these constraints, we can balance model performance with the speed necessary to make a given model practical for real-world usage. In this task, all the segmentation benchmarks

While segmentation provides an essential step in identifying and outlining pavement distresses, achieving accurate individual crack location and classification requires further work beyond the segmentation phase. Separating connected crack regions, distinguishing between over-lapping distresses, and refining the boundaries of segmented areas post-segmentation improve the localization and identification of the individual cracks.

Additionally, advanced classification methods need to be applied to correctly label each crack type, particularly when multiple types of distresses appear close together or exhibit com-plex shapes. Techniques like post-processing algorithms, object detection models, and geometric feature extraction could be employed to improve crack localization and classification accuracy. Therefore, while segmentation lays on the groundwork, additional steps are necessary to achieve precise crack-level analysis and classification for practical applications in pavement assessment.

5.3.5 Distress detection

Unlike segmentation, which only outlines distressed regions, distress detection methods are de-signed to pinpoint the precise coordinates of individual distresses while also estimating their size and categorizing them into specific types. In this section, the ACP and JCP datasets are adopted to develop Deep Learning-based distress detection models, respectively. CRCP will be tested once the data is ready.

5.3.5.1 ACP

In this section, we present a series of experiments to evaluate the performance of various object detection models and analyze their applicability to pavement distress detection on ACP. First, we compare the performance of different models, including YOLOv5, YOLOv8, and Faster RCNN, using a consistent evaluation framework. This comparison helps identify the most effective model for detecting distress types in asphalt concrete pavements. Next, we conduct a detailed analysis of the performance for each distress class using the best-performing model, providing insights into which kinds of distress are better detected and where improvements are needed. We

further explore the impact of using different types of images, intensity, range, or both on the detection performance of the selected model. Lastly, we introduce our proposed model, highlighting the key architectural choices and improvements that address the limitations identified in previous experiments.

Overall performance of different methods

Table 5.11 shows the overall metrics of different models. The configurations for YOLOv5s are as follows: Setting 1 uses the original image size, a single-channel input, and anchor boxes; Setting 2 uses the original image size, a single-channel input, and no anchor boxes; Setting 3 uses a reduced image size, a three-channel input, and anchor boxes. Evaluation metrics for different YOLOv5 settings, YOLOv8, and Faster RCNN on the ACP dataset indicate that YOLOv5s with Setting 3 (reduced image size, three-channel input, and anchor boxes) outperforms others, achieving the highest recall (0.54), mAP50 (0.64), and mAP50-95 (0.33), demonstrating the best overall detection capability. YOLOv5s with Setting 1 (original image size, single-channel input, and anchor boxes) shows the highest precision (0.70), suggesting effective false positive control while using original image size as input. Setting 2, which excludes anchor boxes, underperforms in precision and mAP compared to settings with anchor boxes. The underperformance of YOLOv5s (Setting 2) and YOLOv8s suggests that using anchor boxes can yield better performance for pavement distress detection. YOLOv8s delivers balanced but unremarkable metrics, while Faster RCNN exhibits the weakest performance across all metrics. The results emphasize the advantages of anchor boxes and reduced image size, which do not necessarily lower detection performance or enhance detection performance.

Table 5.11 Evaluation metrics for different methods on ACP image dataset

Method	Images	Instances	P	R	mAP50	mAP50-95
YOLOv5s(setting 1)	780	1,282	**0.70**	0.43	0.56	0.29
YOLOv5s(setting 2)	780	1,282	0.56	0.48	0.52	0.22
YOLOv5s(setting 3)	780	1,282	0.69	**0.54**	**0.64**	**0.33**
YOLOv8s	780	1,282	0.56	0.47	0.51	0.23
Faster RCNN	780	1,282			0.33	0.13

Performance per distress class

In this section, we explore the use of YOLOv5s for distress detection on the ACP dataset using intensity images. The experiment aims to assess the model's ability to identify and classify distress directly from the minute accurately. It offers insights into its performance in detecting pavement distress in asphalt concrete pavements.

The class-wise performance of the model applied to the ACP intensity image dataset shows varying levels of success across different pavement distress types (Table 5.12). For well-represented classes like Joint, Sealed transverse, Lane longitudinal, and Sealed longitudinal, the model per-forms well, with high precision and recall values. For instance, Joint distress has a

precision of 0.708 and a recall of 0.852, reflecting that the model successfully identifies most instances of this class. Similarly, Lane longitudinal and Sealed longitudinal show strong performance with both precision and recall values above 0.78, indicating that the model is effective in detecting these distresses. However, the model struggles with classes like Transverse and Longitudinal, where lower recall values (0.294 and 0.326, respectively) indicate that the model is missing a significant portion of actual instances. The Pothole class, in particular, performs poorly, with very low re-call (0.0769) and low precision (0.333), suggesting that the model struggles to identify pothole instances effectively.

When looking at the mean performance across all classes, the model achieves an average Precision of 0.686 and Recall of 0.543, indicating that while the model is reasonably precise in its predictions, it misses a substantial portion of true positives. The mean mAP50 of 0.636 indicates moderate accuracy in terms of the model's ability to balance precision and recall at a 0.5 IoU threshold. The model has mAP50-95 of 0.33 indicates that more related threshold is needed for the IoU for this application.

Table 5.12 Evaluation metrics for different classes of ACP intensity image dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
Transverse	780	110	0.782	0.294	0.555	0.192
Joint	780	61	0.708	0.852	0.810	0.276
Sealed transverse	780	251	0.785	0.640	0.736	0.370
Longitudinal	780	95	0.574	0.326	0.442	0.212
Lane longitudinal	780	173	0.804	0.782	0.815	0.457
Sealed longitudinal	780	471	0.788	0.811	0.849	0.453
Block	780	24	0.695	0.583	0.663	0.509
Alligator	780	84	0.708	0.524	0.621	0.362
Pothole	780	13	0.333	0.0769	0.229	0.138
Mean	780	1,282	0.686	0.543	0.636	0.33

According to Table 5.12, the model performs well for most visually salient pavement distress types, such as Joint, Sealed transverse, and Lane longitudinal, with high precision and recall. However, the low recall and precision for classes like Pothole and Longitudinal indicate that the model has difficulty identifying certain distress types. The overall mean performance is moderate, with room for improvement in both recall and generalization across different IoU thresholds. To further enhance the model's effectiveness, additional efforts could be focused on improving the detection of underrepresented classes and fine-tuning the model to perform better at stricter IoU thresholds.

The Precision-Confidence Curve (Figure 5.13) demonstrates that the model performs well for distress classes such as Sealed longitudinal, Lane longitudinal, and Joint, maintaining high precision (above 0.8) across a range of confidence thresholds, indicating reliable predictions even at lower confidence levels. However, the model struggles with classes like Pothole and Block, where precision starts low (around 0.4) and only improves significantly at higher

confidence levels, suggesting difficulties in making accurate predictions unless the model is highly confident. Overall, the model achieves strong aggregate performance, with precision reaching 1.00 at a confidence threshold. Still, there is room for improvement in handling certain challenging or underrepresented distress types, especially at lower confidence thresholds.

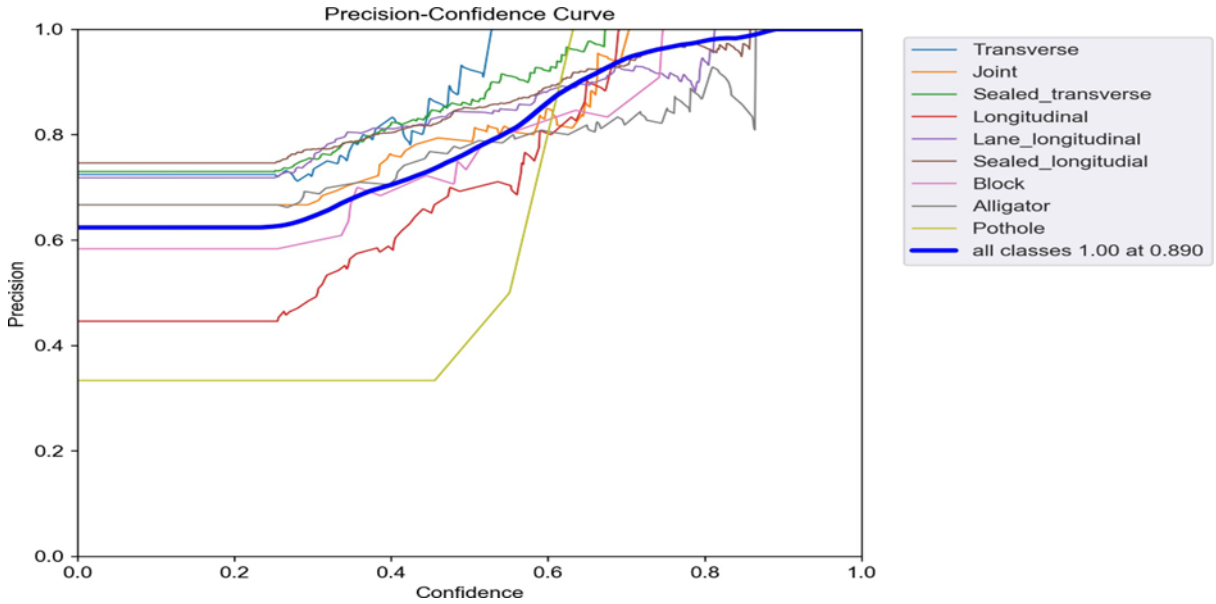


Figure 5.13 Precision curve for different classes of ACP intensity image dataset

The Recall-Confidence Curve (Figure 5.14) shows that the model performs well in detecting distress classes like Joint, Lane longitudinal, and Sealed longitudinal, maintaining high recall values (above 0.8) across much of the confidence threshold range, indicating that the model successfully detects most instances of these classes. However, for classes like Transverse, Longitudinal, and particularly Pothole, recall is consistently low, with Pothole showing a flat zero recall, meaning the model fails to detect any instances of this class. The overall performance, as represented by the blue line, starts with a reasonable recall of 0.57 at lower confidence. However, levels decrease as confidence increases, highlighting that the model becomes more conservative in its predictions, missing more instances when making high-confidence predictions. This suggests that while the model is effective for some common classes, it struggles with detecting challenging or underrepresented classes at higher confidence thresholds.

The F1-Confidence Curve (Figure 5.15) illustrates that the model performs well for classes like Joint, Lane longitudinal, and Sealed longitudinal, maintaining high F1 scores (above 0.8) across a wide range of confidence thresholds, reflecting a good balance between precision and recall. However, for classes like Transverse, Longitudinal, and Pothole, the model struggles, with significantly lower F1 scores, indicating difficulties in balancing precision and recall. Pothole, in particular, shows almost no improvement, maintaining a near-zero F1 score across all confidence levels. The overall model performance, represented by the thick blue line, starts with an F1 score of 0.58, but gradually declines as the confidence threshold increases, suggesting that while the

model maintains a reasonable balance at lower confidence levels, it becomes overly conservative at higher thresholds, missing more true positives as it prioritizes precision over recall.

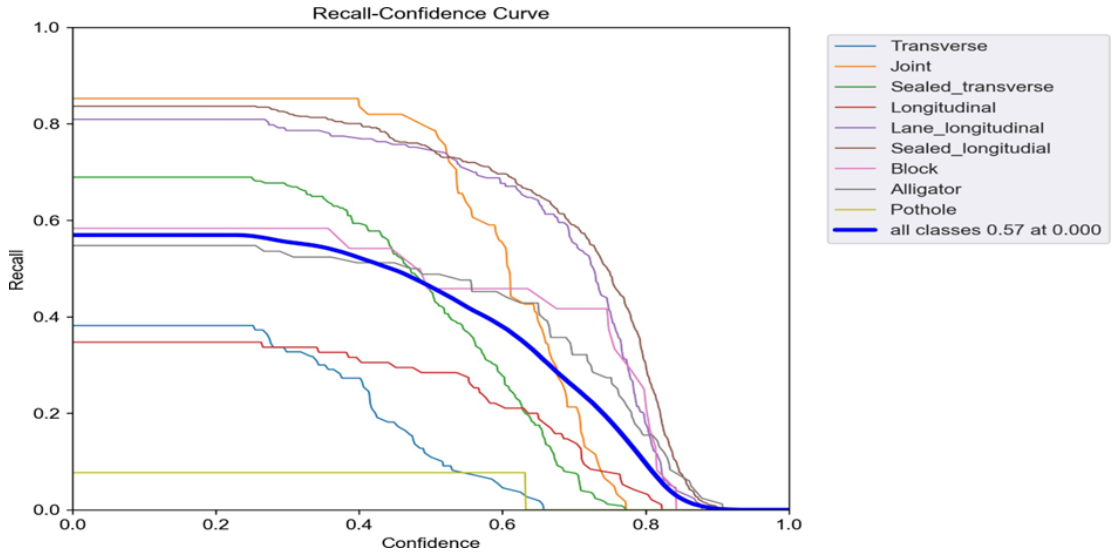


Figure 5.14 Recall curve for different classes of ACP intensity image dataset

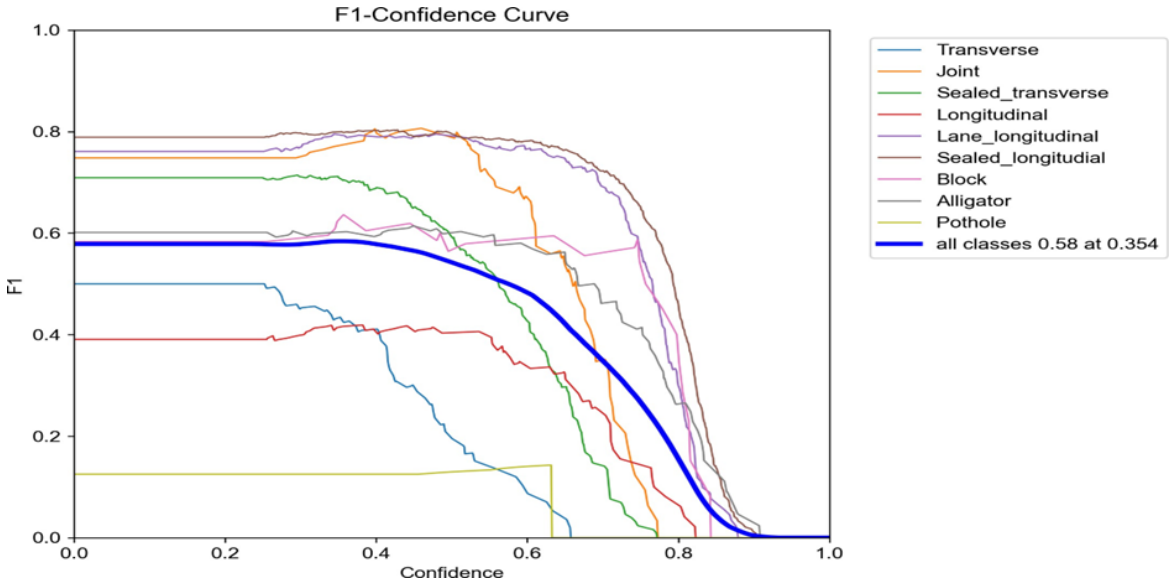


Figure 5.15 F1 curve for different classes of ACP intensity image dataset

The Precision-Recall Curve (Figure 5.16) shows that the model performs well for classes like Joint, Lane longitudinal, and Sealed longitudinal, with precision and recall values above 0.8, indicating that the model is effective at identifying these classes with a low rate of false positives. However, classes like Transverse and Sealed transverse cracks show moderate performance, with the model trading off between precision and recall. The model struggles significantly with classes like Longitudinal and Pothole, where precision and recall are much lower, suggesting difficulties in accurately detecting these distress types. The overall model performance, represented by the thick blue line, shows a mean average precision (mAP) of 0.636

at an IoU threshold of 0.5, indicating a moderate balance between precision and recall across a. Still, but there is room for improvement in handling more challenging or underrepresented classes.

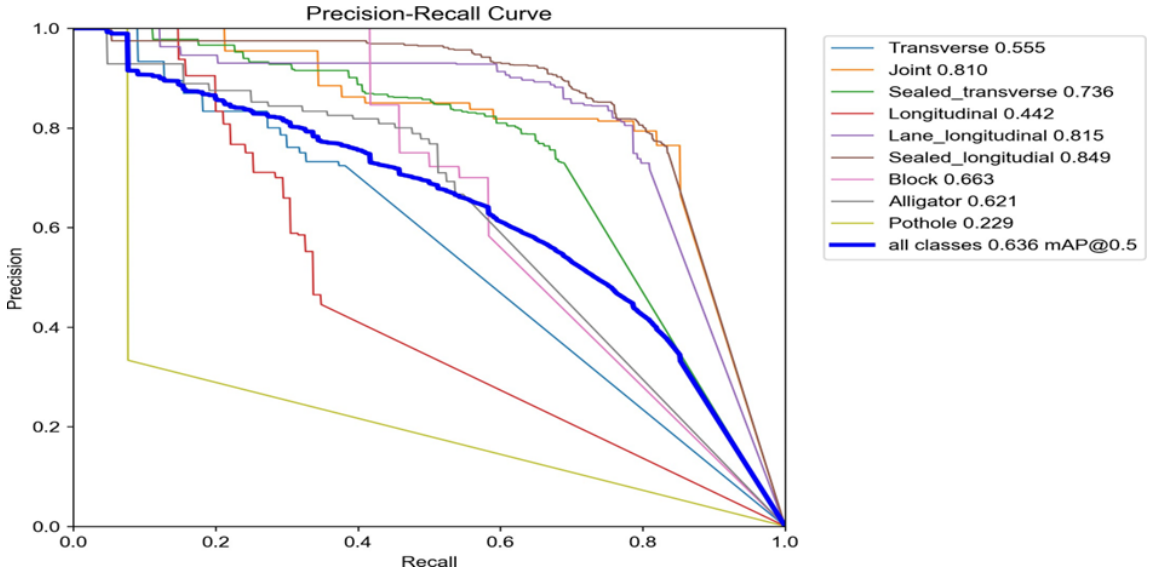


Figure 5.16 Precision/Recall curve for different classes of ACP intensity image dataset

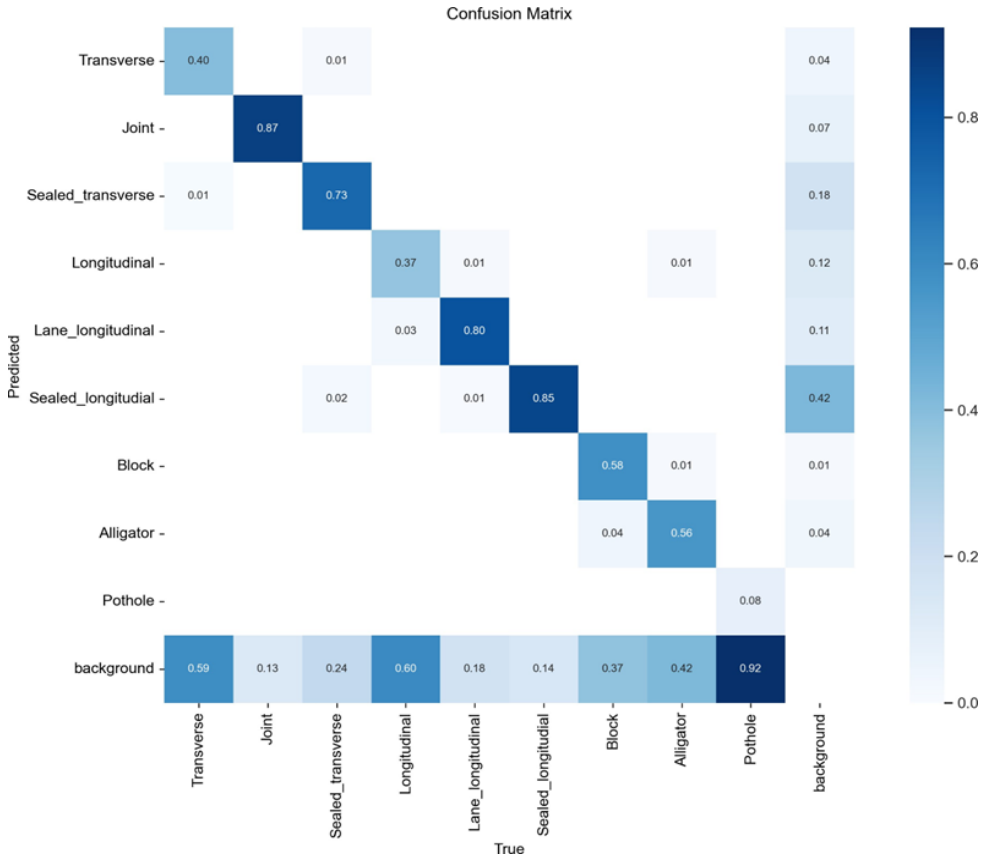


Figure 5.17 Confusion matrix for different classes of ACP intensity image dataset

The Confusion Matrix (Figure 5.17) reveals that the model performs relatively well for classes like Joint (87% accuracy) and Lane longitudinal (80% accuracy), but struggles significantly with other classes, particularly Pothole, where only 8% of instances are correctly classified, making it the worst-performing class. A large portion of Pothole instances (92%) are misclassified as background, indicating the model's difficulty in distinguishing this class. Similarly, Transverse and Longitudinal classes also exhibit poor performance, with only 40% and 37% of instances correctly identified and many misclassified as background. While Sealed transverse and Sealed longitudinal show moderate performance, the confusion matrix highlights the model's challenge with subtle or visually similar distress types, especially in differentiating them from the background class. Overall, the model's performance is reasonable for some classes but requires significant improvement in accurately detecting Potholes and other challenging distress types.

The Confidence Score Distribution histogram (Figure 5.18) shows that the model primarily generates predictions with moderate to high confidence, with a noticeable spike in the 0.7 to 0.8 range, where the frequency of predictions reaches its peak at around 160. The confidence score distributions per pavement distress classes (Figure 5.19) reveal that the model is highly confident in detecting classes like Sealed longitudinal, Lane longitudinal, and Alligator, where most predictions are concentrated in the 0.7 to 0.9 range. In contrast, classes like Sealed transverse, Longitudinal, and Pothole show more dispersed and lower confided. This indicates that the model is less certain when predicting these classes, particularly for Transverse and Pothole, where confidence peaks around 0.4 and 0.5, respectively. Overall, while the model performs confidently for some distress types, it struggles with others, resulting in less reliable predictions for those classes.

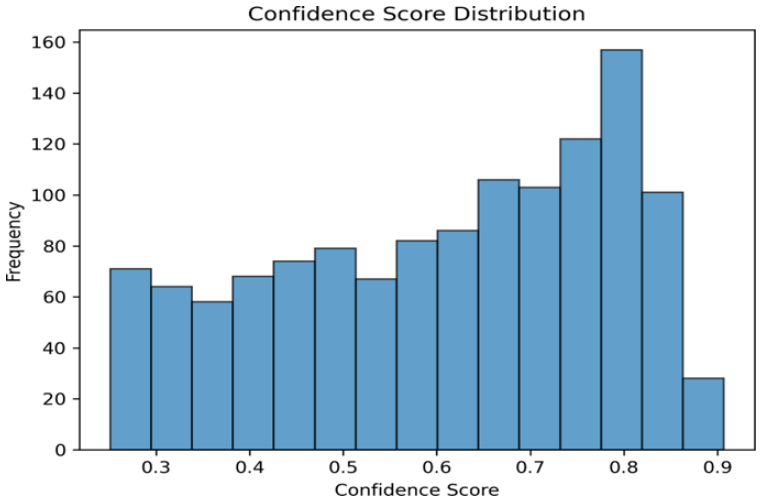


Figure 5.18 Confidence score distribution of the ACP intensity model

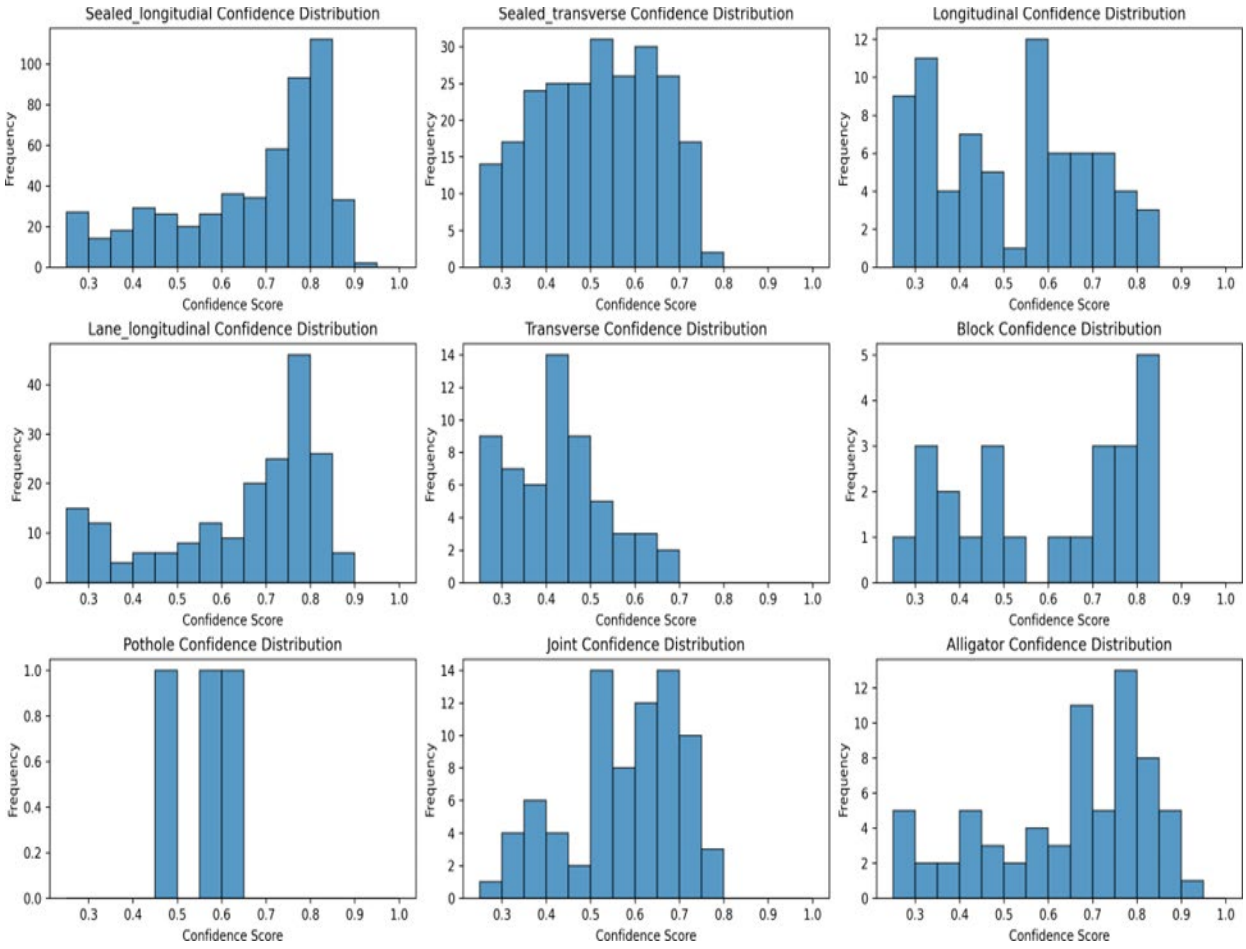


Figure 5.19 Confidence score distribution per distress class of the ACP intensity model

Performance by utilizing different images

Table 5.13 compares YOLOv5's performance using intensity images, range images, and a combination of both, showing that the combined approach yields the best results across all metrics. Precision and recall are highest when both image types are used (0.71 and 0.702, respectively), leading to the highest mAP50 (0.72) and mAP50-95 (0.383), outperforming intensity images (P: 0.686, R: 0.543, mAP50: 0.636, mAP50-95: 0.33) and range images (P: 0.678, R: 0.593, mAP50: 0.66, mAP50-95: 0.341) individually. A possible reason for this performance boost is that certain types of pavement distress may only be visible in either intensity or range images, making the combination more comprehensive for detection tasks.

Table 5.13 Comparison of YOLOv5s Performance using Different ACP Images

Image Type	P	R	mAP50	mAP50-95
Intensity image	0.686	0.543	0.636	0.33
Range image	0.678	0.593	0.66	0.341
Both	**0.71**	**0.702**	**0.72**	**0.383**

Figure 5.20 compares mAP50 scores across different pavement distress types using intensity images, range images, and a combination of both. For several pavement distress types, combining both intensity and range images improved the detection performance. These include Joint, Lane longitudinal crack, Block crack, Alligator crack, and Pothole. The improvement can be attributed to the additional information provided by the intensity and range data together. For example, Joint and Lane longitudinal crack may benefit from range images that capture depth variations, while intensity images capture texture and color variations. Distress cracks and Pothole are often complex in shape and structure, and combining both image types enhances the model's ability to detect subtle variations in the pavement surface that either image type alone might miss. This suggests that these distress types benefit from extra data, leading to better feature representation and improved detection accuracy. For Longitudinal crack and Transverse crack, the combined images did not show better performance compared to using range images alone. This may be because these types of cracks are primarily surface-level features with clear depth differences, which are effectively captured by range images. Intensity images, which reflect variations in color or brightness, may not contribute significantly to detecting these cracks, as their appearance is less dependent on surface texture or color changes and more on structural depth information. In Additionally, for the Sealed transverse crack and Sealed longitudinal crack, the combined images did not show a notable improvement over range images, likely because sealed cracks lack range variation, making it harder for intensity data to add meaningful information. However, it's worth noting that even though sealed cracks are difficult for humans to identify in range images, a large portion can still be detected, possibly due to subtle patterns in the range data that the model can pick up.

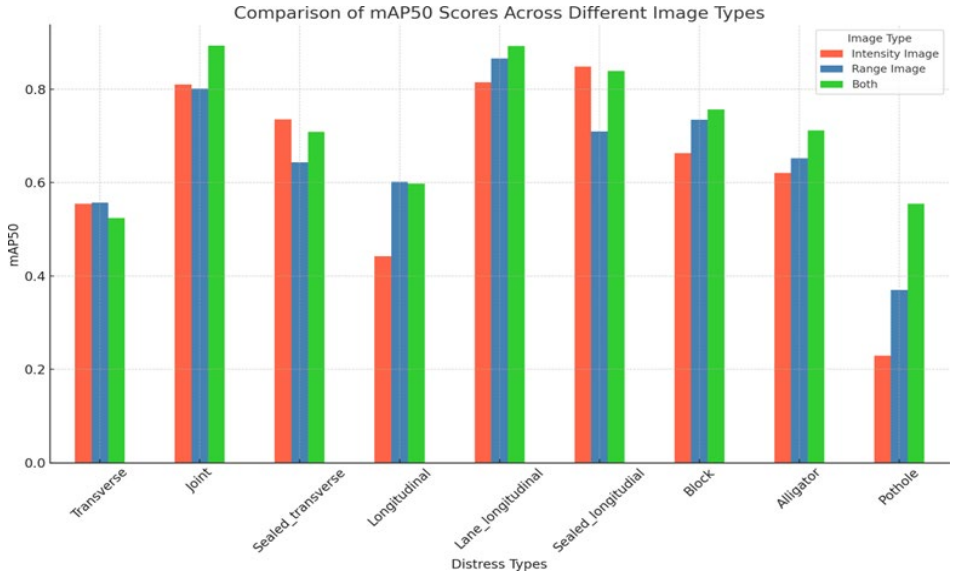


Figure 5.20 Comparison of mAP50 scores across different image types

Figure 5.21 compares precision scores across different image types for various pavement distress types, showing that combining both intensity and range images does not necessarily improve the distress detection precision. For distress types such as Joint crack, Block crack, and Pothole, combining both image types yields higher precision, likely due to the additional information from both texture and depth, which helps reduce false positives. However, for Transverse crack, Sealed transverse crack, Longitudinal crack, and Sealed longitudinal crack, the combined images

do not result in better precision compared to using range/intensity images alone. This indicates that for distress types primarily dependent on one specific image type, adding extra information may actually reduce precision by introducing noise or irrelevant data that does not contribute meaningfully to the detection of these classes.

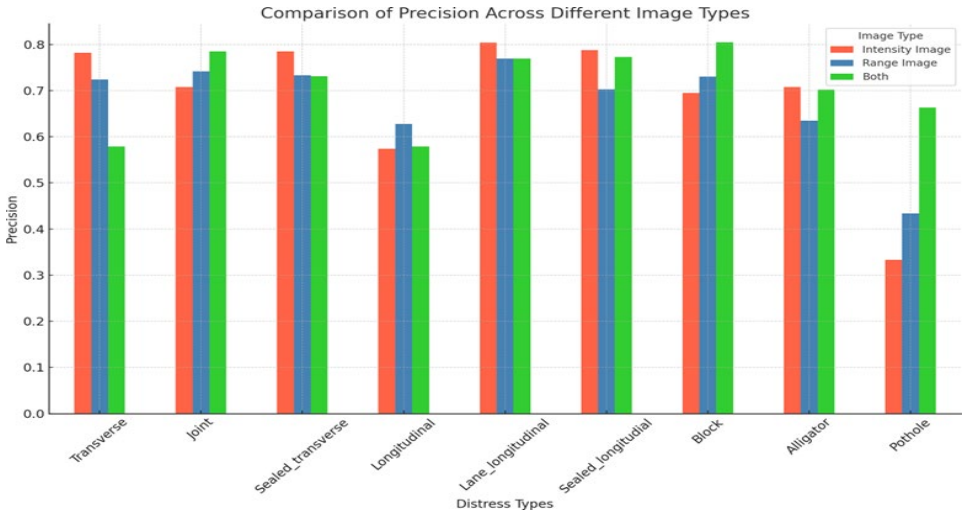


Figure 5.21 Comparison of precision scores across different image types

Figure 5.22 compares recall scores across different image types for various pavement distress classes. Recall score across all distress classes with range images is higher than those with intensity images, except for Joint, Sealed transverse, and Sealed longitudinal. This indicates that range images generally provide more useful information for detecting most pavement distresses. This suggests that depth variations, captured by range images, play a crucial role in identifying these distresses. However, for Sealed transverse, Longitudinal, and Joint cracks, intensity images perform slightly better or comparably, indicating that for these specific distresses, surface texture and color information (captured by intensity images) may contribute more to detection than depth information alone. Combining both intensity and range images greatly improves recall across all distress classes, indicating that the complementary nature of these image types provides more comprehensive information, leading to better identification of true positive instances. This combination is particularly effective for complex distresses like Potholes, which include both patched or unpatched surface texture (intensity) and depth (range) are essential for accurate detection of Potholes. For most distresses, combining both image types helps capture subtle features that might be missed by either image type alone, making the model more robust and improving overall detection performance.

A notable observation is that cracks on rough pavement surfaces are challenging to distinguish from the background due to the resolution of 2D/3D images. These cracks often appear as a series of segments rather than continuous lines, and their characteristics closely resemble the texture of the non-distressed areas. This similarity can trigger a chain reaction if these cracks are considered during the analysis.

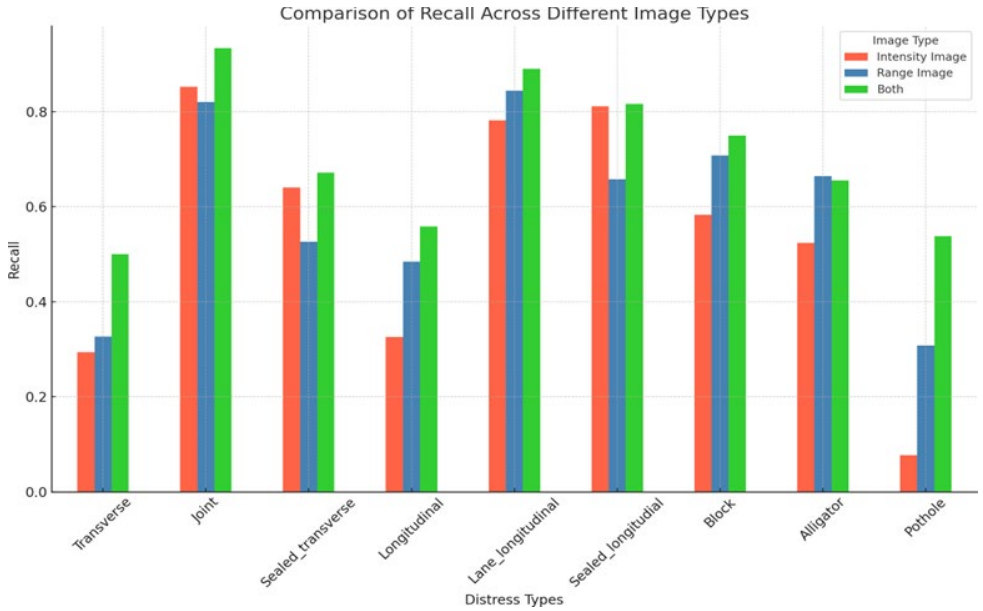


Figure 5.22 Comparison of recall scores across different image types

Beginning with the annotation phase, such cracks can confuse annotators, leading to inconsistent labeling. Consequently, the trained model may suffer from either overfitting or poor convergence due to the high similarity between the cracks and the background. To address this issue, higher-resolution images are desirable to overcome this limitation.

Proposed method

The proposed distress detection architecture builds upon the YOLO framework, adapted to process the two distinct channels: intensity and range. This new model, designed to enhance the identification of pavement surface distresses, operates under a dual-channel input strategy where each channel is processed independently yet in parallel. This method allows the model to maintain the unique characteristics of each type of data while efficiently handling them simultaneously.

- Base Model: The architecture uses YOLO as its foundational model due to YOLO's robust-ness and efficiency in real-time object detection tasks. YOLO's architecture is modified to accept two separate data streams instead of the traditional single three-channel RGB input.

- Dual-Channel Processing: In this architecture, the intensity and range channels are fed into separate but parallel branches of the network right from the initial layer. Each branch con-sists of a series of convolution and pooling layers specifically tuned to optimize the features extracted from its respective data type. This parallel processing ensures that the model lever-ages the distinct physical properties captured by each channel—depth and reflectivity.

- Attention Mechanism: After initial feature extraction, an attention mechanism is introduced. This mechanism plays a critical role in integrating the features from the intensity and range channels. It dynamically assigns weights to different feature maps generated by each chan-nel. The weighting is determined by the relevance of each feature map to the distress detec-tion task, allowing the model to prioritize more informative features dynamically.

- Feature Integration and Detection: The weighted features from both channels are then com-bined and passed through additional layers of the network to refine the feature representation further. This integration is crucial for allowing the model to capitalize on the complementary information provided by both types of data. The combined features ultimately lead to the final detection layer, which predicts the presence, types, and locations of distresses on the pavement surface.

This architecture addresses the challenge that different types of distress manifest variably across intensity and range images. By learning to assign appropriate importance to different fea-tures depending on their relevance to the detection task, the model should be able to enhance its accuracy and performance in identifying pavement distresses.

Table 5.14 compares the performance of three models, YOLOv5s, Model 1, and Model 2, on various pavement distress types using the mAP50 metric. Model 1 uses a single weight for the output features of intensity and range channels, respectively, while Model 2 applies different weights for different feature scales. Both proposed models generally exhibit better performance than YOLOv5s, with particular strengths noted in detecting Transverse cracks and Sealed transverse cracks. Model 2, which utilizes varying weights for different feature scales, demonstrates the more improved performances, with 10.6% increase for Transverse crack detection compared to that of YOLOv5s. However, both Model 1 and Model 2 underperform in detecting potholes, which is likely due to a limited number of pothole samples in the training dataset.

Table 5.14 Comparison of Proposed Models with YOLOv5s on ACP dataset

Distress class	YOLOv5s	Model 1		Model 2	
		mAP50	Change	mAP50	Change
Transverse	0.524	0.584	6.0%	**0.629**	10.5%
Joint	**0.894**	0.853	-4.1%	0.859	-3.5%
Sealed transverse	0.709	0.756	4.7%	**0.77**	6.1%
Longitudinal	0.598	0.598	0.0%	0.598	0.0%
Lane longitudinal	**0.893**	0.881	-1.2%	0.878	-1.5%
Sealed longitudinal	0.839	**0.846**	0.7%	0.841	0.2%
Block	0.757	**0.79**	3.3%	0.776	1.9%
Alligator	**0.712**	0.699	-1.3%	0.7	-1.2%
Pothole	**0.555**	0.423	-13.2%	0.379	-17.6%

Data augmentation

To enhance the performance of the pavement distress detection model, we augmented the original dataset, which consisted of pavement images collected in 2022, by adding a new set of images from 2023. These recent images were sourced entirely from Jefferson County and served as a crucial supplement to the existing dataset. Notably, a significant portion of the 2023 images were ones that had been misclassified or incorrectly detected by the distress detection model trained on the original 2022 dataset. This augmentation aims to address specific weaknesses in the model by incorporating diverse and challenging instances of pavement distress, thereby improving its robustness and detection accuracy across varied conditions. The addition of misclassified images provides an opportunity to refine the model's learning process by focusing on correcting previously identified deficiencies, leading to better generalization and adaptability in distress detection.

The results of the dataset augmentation are presented in Table 5.15. As shown, the number of instances for each distress class increased significantly after augmentation, leading to notable improvements in the model's performance. The mAP50 metric increased from 0.72 to 0.84 overall. Specific distress classes, such as transverse, longitudinal, and pothole, exhibited significant gains in mAP50, demonstrating that the augmented dataset successfully addressed areas where the original model was underperforming. However, the performance for pothole detection remains relatively low due to the small sample size, even after augmentation. These improvements highlight the effectiveness of the augmentation in enhancing the model's ability to detect a wide range of pavement distresses more accurately.

Table 5.15 Augmentation of the ACP Detection Dataset

Distress class	**Number of instances**		**mAP50**	
	Before augment	**After augment**	**Before augment**	**After augment**
Transverse	863	1,637	0.52	**0.81**
Joint	433	883	0.89	**0.90**
Sealed transverse	1,771	2,102	0.71	**0.87**
Longitudinal	724	1,698	0.60	**0.84**
Lane longitudinal	1,229	1,481	0.89	0.89
Sealed longitudinal	3,303	3,805	0.84	**0.93**
Block	134	297	0.76	**0.88**
Alligator	521	855	0.71	**0.86**
Pothole	84	160	0.56	**0.59**
Total	9,062	12,518	0.72	**0.84**

5.3.5.2 JCP

In this section, we present a series of experiments to evaluate the performance of various object detection models and analyze their applicability to pavement distress detection on JCP. First, we compare the performance of different models, including YOLOv5, YOLOv8, and Faster R-CNN, using a consistent evaluation framework as we did with ACP. Next, we conduct a detailed analysis of the performance for each distress class using the best-performing model, providing insights into which types of distress are better detected and where improvements are needed. Then we further explore the impact of using different types of images, intensity, range, or both, on the detection performance of the selected model.

Overall performance of different methods

Table 5.16 shows the overall metrics of different models. The configurations for YOLOv5s are the same as described in Section 3.5.1. Based on Table 5.16, YOLOv5s Setting 3 achieves the highest performance metrics across all categories: precision (0.58), recall (0.52), mAP50 (0.53), and mAP50-95 (0.33), confirming that it provides the best overall detection capability. YOLOv5s Setting 1 demonstrates effective false positive control with a precision of 0.50, but lower recall (0.47) compared to Setting 3. Setting 2, without anchor boxes, performs the weakest among the YOLOv5 settings, supporting the conclusion that anchor boxes enhance detection performance. YOLOv8s delivers balanced yet lower metrics. The result is consistent with that of Table 5.11, with Setting 3 achieving the best performance, and models with anchor boxes outperforming those without anchor boxes.

Table 5.16 Evaluation metrics for different methods on JCP image dataset

Method	Images	Instances	P	R	mAP50	mAP50-95
YOLOv5s (setting 1)	1,038	2,072	0.50	0.47	0.50	0.30
YOLOv5s (setting 2)	1,038	2,072	0.37	0.45	0.40	0.18
YOLOv5s (setting 3)	1,038	2,072	**0.58**	**0.52**	**0.53**	**0.33**
YOLOv8s	1,038	2,072	0.44	0.42	0.43	0.22
Faster RCNN	1,038	2,072			0.26	0.14

Performance per distress class

Based on the performance comparison, we explored the use of YOLOv5s for distress detection on the JCP dataset using intensity images. Like with the ACP dataset, the experiment aims to assess the model's ability to accurately detect distresses, directly from the intensity data.

The class-wise performance of the model (Table 5.17) applied to the JCP intensity image dataset reveals a varying degree of accuracy across different pavement distress types. For well-represented classes such as Failed joint, Asphalt patch, Longitudinal crack, Sealed longitudinal, and Concrete patch, the model demonstrates good performance, with precision and recall values consistently above 0.7. Notably, Joint crack and Slab edge are detected with high accuracy, with both precision and recall values above 0.9, indicating the model excels in identifying these types

of distress with minimal false positives or false negatives. However, for some underrepresented classes like Corner break, Punchout, Failed concrete patch, and D-cracking, the model performs poorly, with both precision and recall values of 0, signifying it fails to detect these classes entirely.

When analyzing the overall mean performance, the model's precision, recall, and mAP50 values stand at 0.583, 0.517, and 0.534, respectively, reflecting moderate performance across all distress classes. The mAP50-95 score, which measures performance across a range of IoU thresholds, is lower at 0.331, indicating that the model's accuracy decreases as the IoU threshold becomes stricter. However, when underrepresented distress classes are excluded from the calculation, the performance improves significantly. The precision rises to 0.822, recall to 0.840, and mAP50 to 0.868, showing that the model is much more effective at identifying common and well-represented distress types. The mAP50-95 score also improves to 0.538, indicating a more robust performance across IoU thresholds for these classes.

According to Table 5.17, the model performs well for common distress types but struggles with the underrepresented classes in the currently used JCP dataset. The exclusion of these underrepresented classes leads to a marked improvement in performance metrics, highlighting the need for further training or data augmentation for rare classes. Addressing these deficiencies in detecting less frequent pavement distresses could result in a more balanced and accurate model, capable of identifying both common and rare pavement conditions with high precision and recall.

Table 5.17 Evaluation metrics for different classes of JCP intensity image dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
Failed joint	1,038	72	0.783	0.778	0.781	0.44
Corner break	1,038	6	1	0	0	0
Punchout	1,038	1	0	0	0	0
Asphalt patch	1,038	48	0.916	0.854	0.922	0.547
Failed concrete patch	1,038	4	0	0	0	0
D-cracking	1,038	1	0	0	0	0
Popout	1,038	3	0	0	0	0
Longitudinal crack	1,038	159	0.721	0.843	0.815	0.416
Sealed longitudinal	1,038	14	0.818	0.857	0.904	0.601
Concrete patch	1,038	120	0.8	0.802	0.839	0.531
Transverse crack	1,038	83	0.746	0.687	0.758	0.392
Joint crack	1,038	524	0.903	0.941	0.959	0.634
Slab edge	1,038	1,037	0.891	0.956	0.964	0.74
mean			0.583	0.517	0.534	0.331
mean (without underrepresented distress classes)			0.822	0.840	0.868	0.538

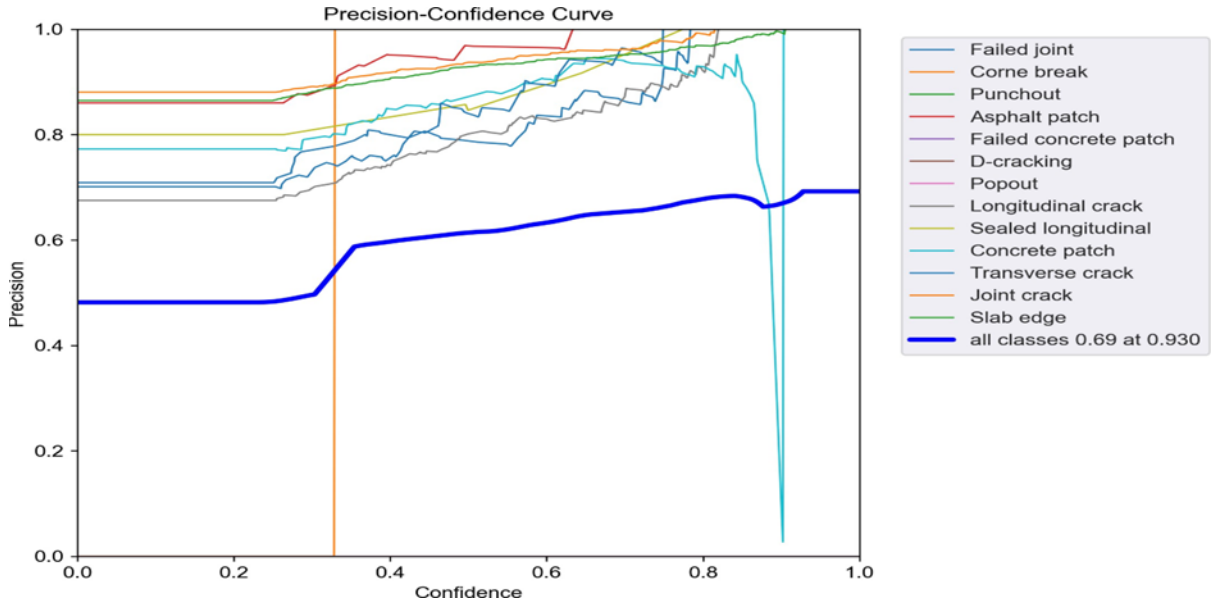


Figure 5.23 Precision curve for different classes of JCP intensity image dataset

The F1-Confidence Curve (Figure 5.23) shows how the F1 score, which combines precision and recall, changes as the confidence threshold varies across different pavement distress classes. Classes like Slab edge, Concrete patch, and Joint crack maintain high F1 scores even at higher confidence thresholds, indicating that the model performs reliably for these distress types. Conversely, classes such as Failed joint and Asphalt patch see a noticeable drop in F1 score as the confidence threshold. This suggests that the model struggles to maintain accuracy for these categories when it becomes more conservative. The overall performance, represented by the bold blue line, peaks with an F1 score of 0.51 at a confidence level of 0.36, highlighting this as the optimal threshold for balancing precision and recall across all distress types.

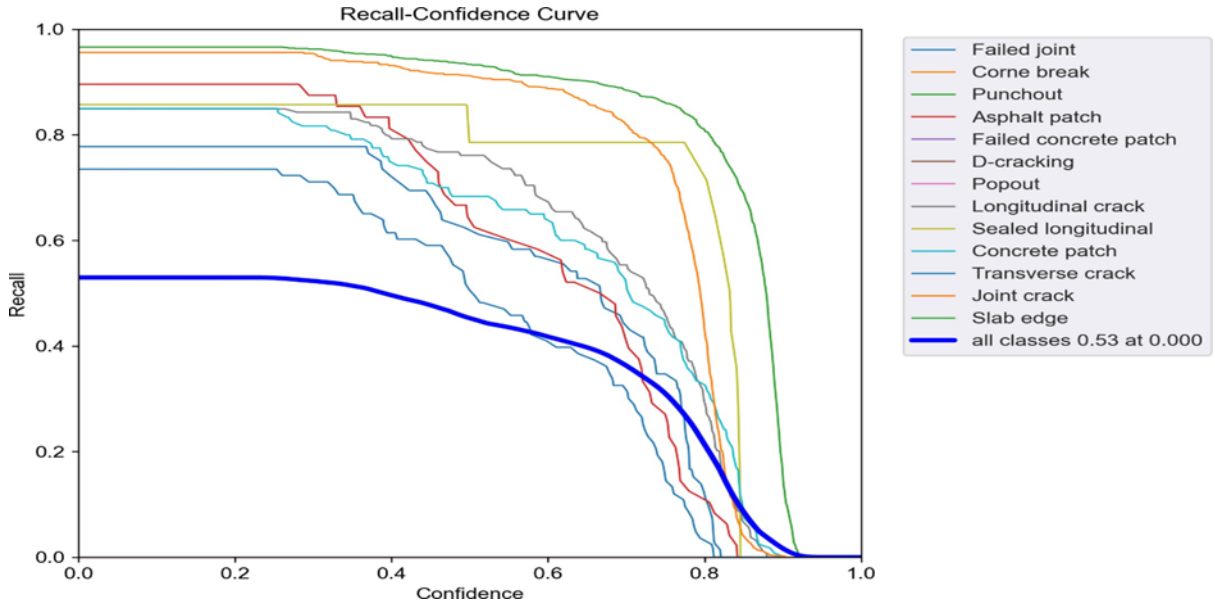


Figure 5.24 Recall curve for different classes of JCP intensity image dataset

The Recall-Confidence Curve (Figure 5.24) shows that the model performs well in terms of recall for classes such as Slab edge, Joint crack, and Sealed longitudinal, maintaining high recall even at higher confidence thresholds, indicating the model is capable of identifying most instances of these distresses with increased certainty. However, for classes like Failed joint, Longitudinal crack, and Asphalt patch, recall declines significantly as the confidence threshold rises, meaning the model begins missing more instances as it becomes more selective. The overall recall, represented by the thick blue line, starts at 0.53 and decreases as confidence increases, indicating that while the model captures more true positives at lower confidence levels, it becomes more conservative at higher thresholds, resulting in missed detections.

The F1-Confidence Curve (Figure 5.25) shows how the F1 score, which combines precision and recall, changes as the confidence threshold are varied across different pavement distress classes. Classes like Slab edge, Concrete patch, and Joint crack maintain high F1 scores even at higher confidence thresholds, indicating that the model performs reliably for these distress types. Conversely, classes such as Failed joint and Asphalt patch see a noticeable drop in F1 score as the confidence threshold. This suggests that the model struggles to maintain accuracy for these categories when it becomes more conservative. The overall performance, represented by the bold blue line, peaks with an F1 score of 0.51 at a confidence level of 0.36, highlighting this as the optimal threshold for balancing precision and recall across all distress types.

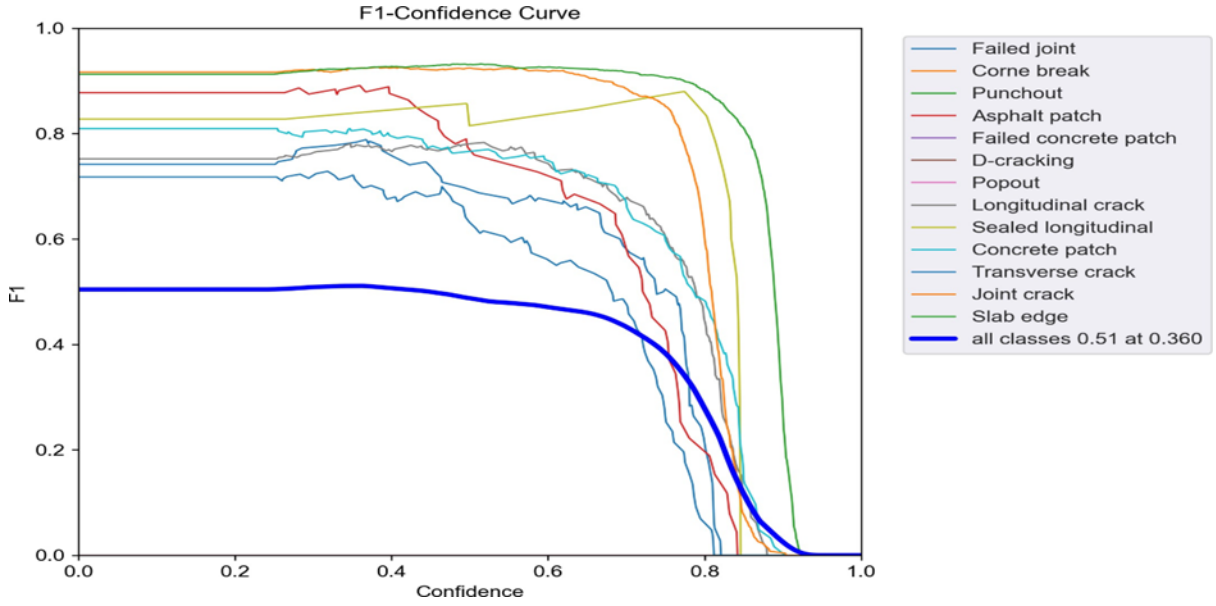


Figure 5.25 F1 curve for different classes of JCP intensity image dataset

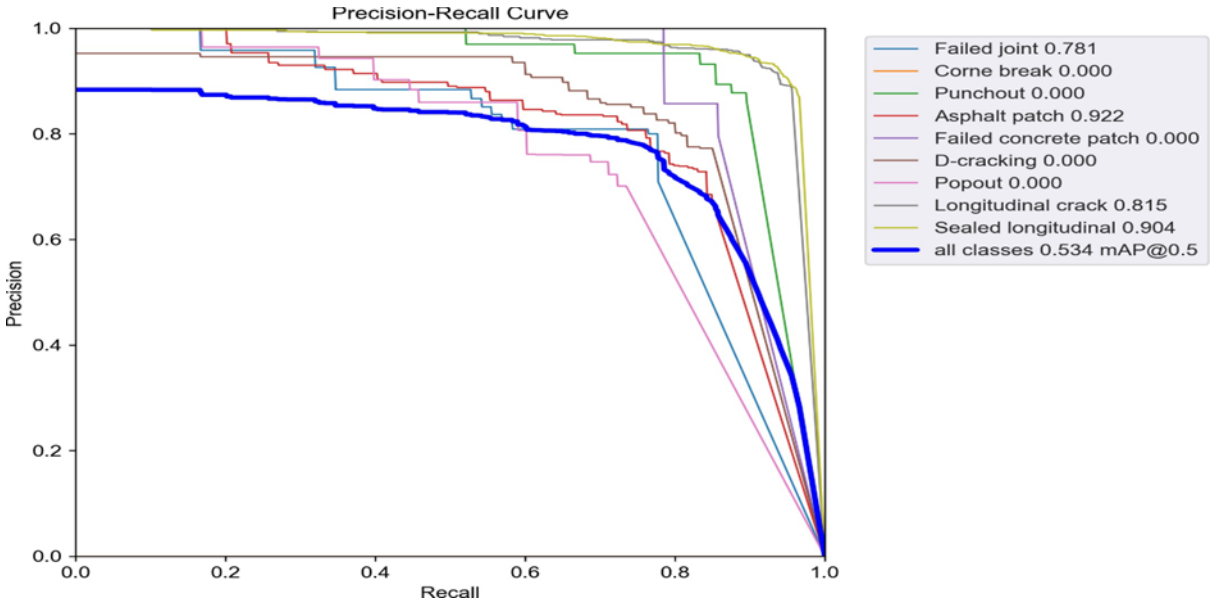


Figure 5.26 Precision/Recall curve for different classes of JCP intensity image dataset

The Precision-Recall Curve (Figure 5.26) displayed in the image highlights the model's performance across various pavement distress classes, demonstrating a significant trade-off between precision and recall. High-performing classes such as Asphalt patch, Sealed longitudinal, and Longitudinal crack exhibit curves near the top-right corner of the plot, indicating robust precision and recall values, with Asphalt patch reaching the highest precision at 0.922. Conversely, several classes like Corner break, Punchout, Failed concrete patch, and Popout show zero precision and re-call, indicating the model's complete inability to detect these types accurately. The overall model performance, represented by a bold blue line, shows a mean average precision (mAP) of 0.534 at an IoU of 0.5, suggesting moderate effectiveness with a need for improvement, particularly in enhancing detection capabilities for less common distress types.

The Confusion Matrix (Figure 5.27) provides a detailed view of the model's performance by comparing the true labels (on the horizontal axis) to the predicted labels (on the vertical axis) for different classes of pavement distress. The intensity of the color reflects the proportion of correctly and incorrectly classified instances, with darker shades indicating a higher proportion. According to Figure 5.27, the model performs very well for certain classes, particularly Slab edge (0.97), Joint crack (0.95), Longitudinal crack (0.87), Concrete patch (0.82), and Asphalt patch (0.81), where the predictions closely align with the true labels, indicating high accuracy. For these classes, the majority of instances are correctly classified with minimal confusion with other classes. Classes like Failed joint (0.71), Sealed longitudinal crack (0.79), and Transverse crack (0.75) show moderate performance. While most instances are classified correctly, there is some confusion with other classes, as seen by small proportions of misclassified instances. For example, a significant portion of Failed joints are misclassified as Joint. A substantial portion of Sealed longitudinal cracks and transverse cracks are misclassified as background, indicating a need for recall improvement.

Several classes, such as Corner break, Punchout, Failed concrete patch, D-cracking, Popout, and Sealed transverse crack, have no predictions or very little data, as indicated by empty or pale boxes. These classes are either underrepresented or confused with others, reflecting the model's inability to detect these distress types effectively. Notably, all instances of Failed concrete patches are misclassified as Concrete patches, as evidenced by the absence of any correctly predicted failed concrete patches and the presence of misclassified instances in the Concrete patch category. This indicates that the model struggles to differentiate between these two distress types, which may appear visually similar.

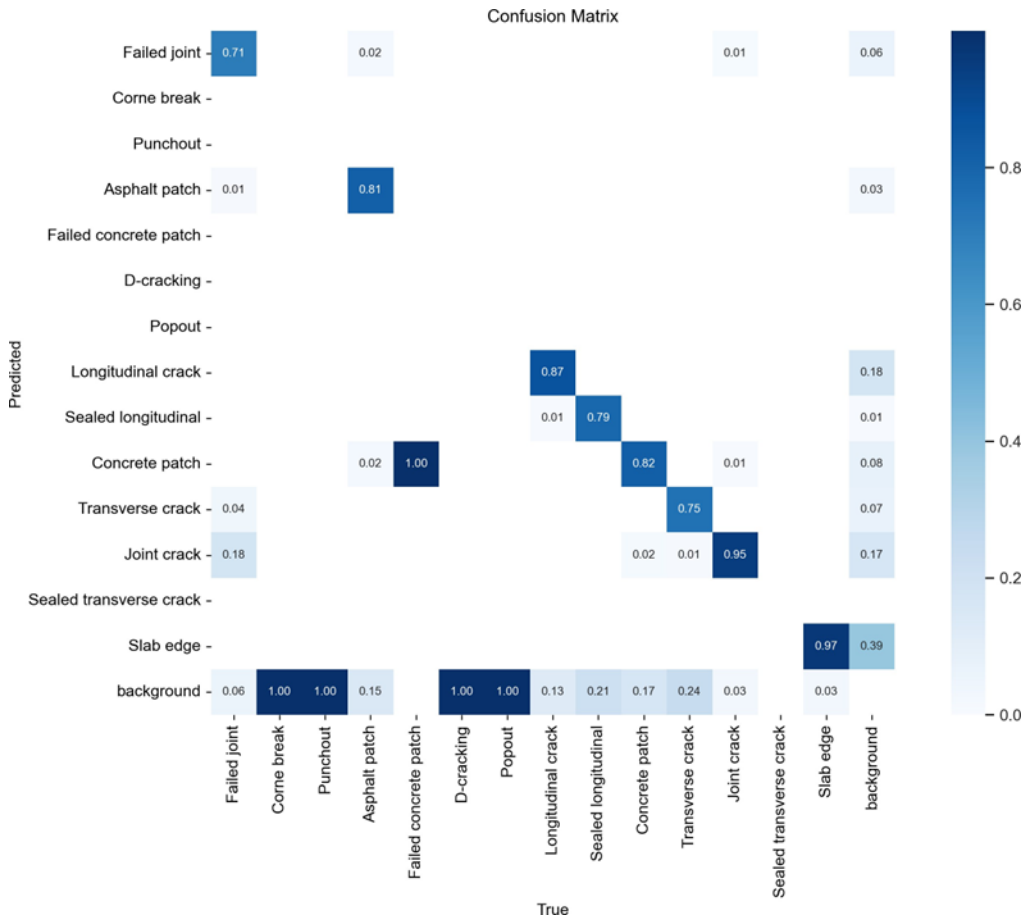


Figure 5.27 Confusion matric for different classes of JCP intensity image dataset

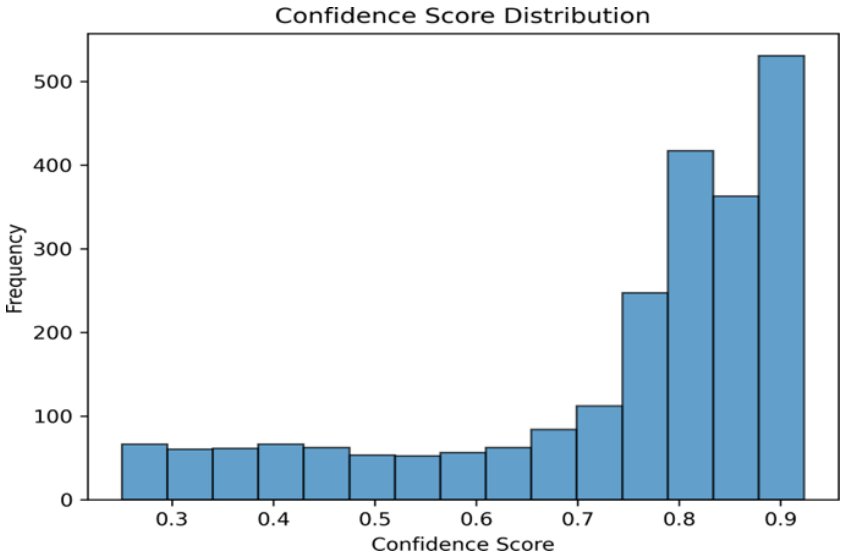


Figure 5.28 Confidence score distribution of the JCP intensity model

The confidence score distribution (Figure 5.28) shows that the majority of the model's predictions fall within the higher confidence range (0.7 to 0.9), with the largest frequency concentrated around a score of 0.9. This suggests that the model is generally confident in its predictions. Lower confidence scores (between 0.3 and 0.6) occur much less frequently, indicating that the model rarely produces uncertain predictions. Overall, the distribution implies that the model tends to make predictions with a high level of certainty, which could be a positive indication of its performance, even though further investigation into accuracy and error rates would provide a more complete assessment.

Metrics by utilizing different images

Table 5.18 compares YOLOv5s performance on the JCP dataset using intensity images, range images, and their combination, both with and without underrepresented distress classes. Overall, the combination of both image types yields the best performance (Figure 5.29), with slightly higher mAP50 (0.554) and mAP50-95 (0.350) than intensity or range images alone. When underrepresented distress classes are excluded, performance improves significantly across all metrics, particularly for intensity images, where precision (0.822), recall (0.840), and mAP50 (0.868) are highest. Although the combination of both image types still delivers the best mAP50-95 (0.548), the results suggest that using intensity images alone can achieve strong performance for better-represented distress types, while combining both image types ensures more balanced and comprehensive detection.

In contrast, the performance on intensity images is nearly the same, while range images consistently perform slightly lower across these distress types. This suggests that intensity images are a reliable source of information for distress detection in the JCP dataset. This observation contrasts with the ACP dataset, where range information plays a more crucial role in distress detection. In JCP, surface texture and appearance captured by intensity images seem to be the dominant features, while depth information from range images is less important. Despite

these variations in detection performance across image types, all distress classes, including Asphalt patch, Joint crack, Slab edge, and others, yield acceptable mAP50 scores, generally in the range of 0.8 to 0.9. This reflects that the model performs well across the board in identifying and detecting different distress types in JCP, as long as sufficient training samples are fed to the model. The high scores indicate that even though some image types (such as intensity images) may be more reliable for specific distresses, all three image types provide sufficient information for acceptable distress detection.

Table 5.18 Comparison of YOLOv5s Performance using Different Images

Image Type	P	R	mAP50	mAP50-95
Mean				
Intensity image	**0.583**	0.517	0.534	0.331
Range image	0.544	0.510	0.543	0.336
Both	0.525	**0.536**	**0.554**	**0.350**
Mean (without underrepresented distress classes)				
Intensity image	**0.822**	**0.840**	**0.868**	0.538
Range image	0.793	0.787	0.825	0.516
Both	0.794	0.833	0.863	**0.548**

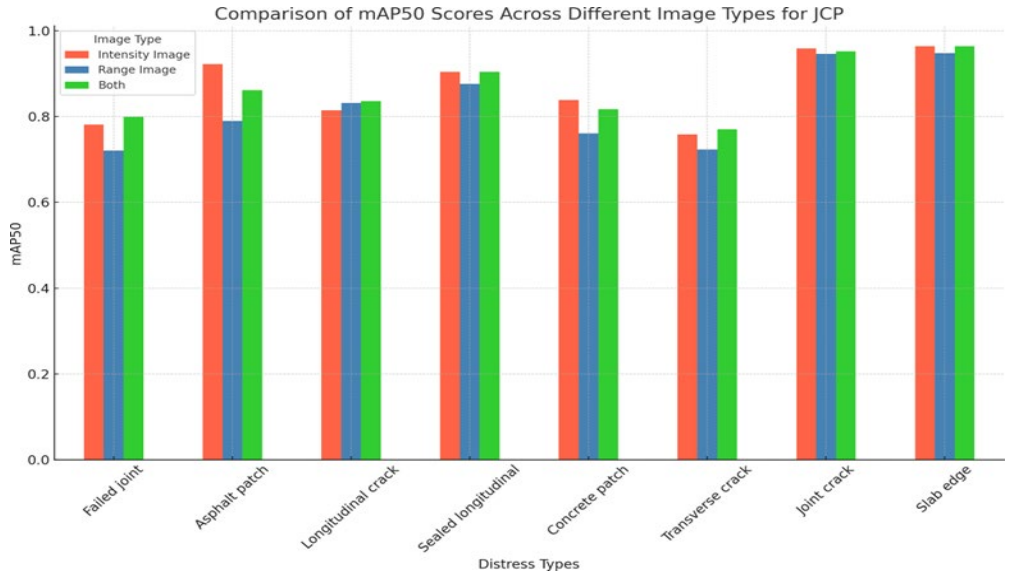


Figure 5.29 Comparison of mAP50 scores across different image types

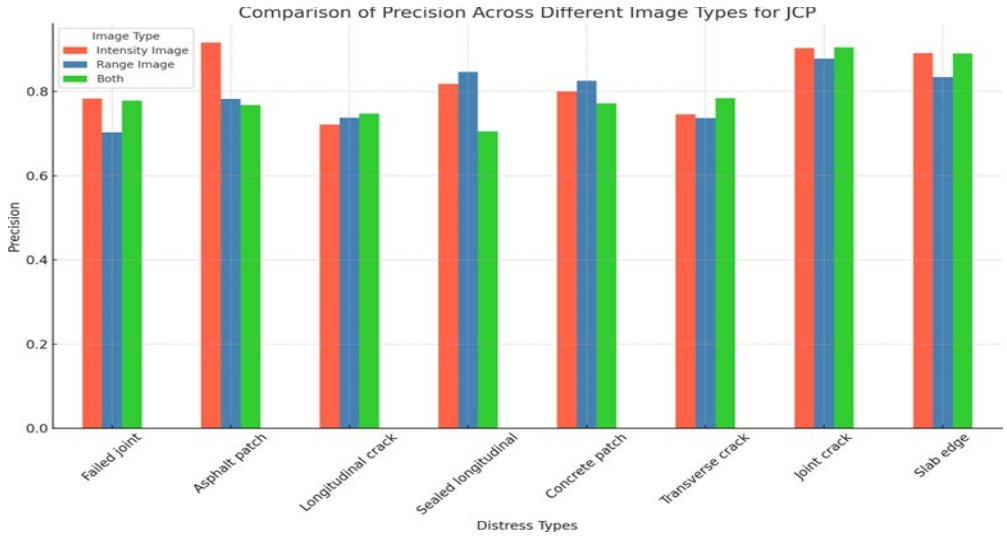


Figure 5.30 Comparison of precision scores across different image types

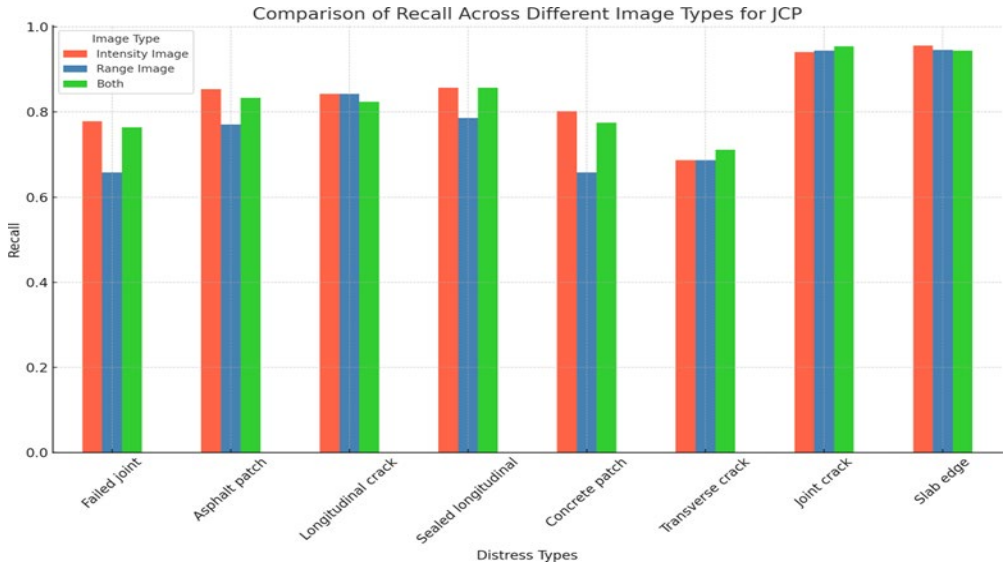


Figure 5.31 Comparison of recall scores across different image types

Figures 5.30 and 5.31 compare precision and recall scores across the three im-age types in the JCP dataset. Asphalt patch detection shows the highest precision using intensity images (red bar), with both range and combined images performing slightly worse. This suggests that surface texture and color features captured by intensity images are more reliable for detecting this distress type, while range data does not contribute as effectively to increasing precision. For most of the other distress types, precision remains consistent across intensity, range, and combined images. The small differences across image types suggest that all three provide comparable information for accurately identifying these distresses. The slight variation, particularly for Failed joint, Asphalt patch, Sealed longitudinal crack, and Concrete patch, where range images perform slightly lower than intensity and combined images, indicates that intensity information is crucial for extracting more accurate distress features from the JCP dataset.

5.3.5.3 Discussion

The performance of pavement distress detection can be influenced by various factors, which are discussed below. One key factor is the image configuration, mainly spatial resolution, which significantly impacts the ability to detect fine details of distress types. Another important consideration is the presence of underrepresented distress classes in the dataset, which can lead to biased detection performance, particularly for rare distress types that may not have enough training examples. Additionally, the inherent characteristics of different distress classes and pavement surface types can affect the detection performance differently, necessitating tailored model configurations for optimal results.

Image configuration plays a crucial role in the detection of pavement distress using machine learning models. One important factor is spatial resolution. Large spatial resolution can make some cracks appear very thin or even discontinuous, which complicates the labeling process for annotators and may lead to inaccurate annotations. These inaccuracies can further compromise the model's ability to learn and detect such distress types effectively. Additionally, artifacts introduced by image collection sensors, though not common, can sometimes create unexpected features that might be mistakenly labeled as distress. These artifacts confuse the model during training, resulting in false positives during detection. Furthermore, variations in image resolution can also influence detection performance by creating inconsistencies in the features that the model learns. These factors make it challenging for the model to accurately differentiate between genuine pavement distress and variations introduced by image configuration, thereby affecting the detection system's robustness and generalizability. These issues can be addressed by increasing spatial resolution to enhance detail visibility with computational efficiency, implementing preprocessing techniques to reduce sensor artifacts, and using data augmentation to account for variations in image resolution.

Underrepresented distress classes can significantly impact the detection performance of machine learning models. According to the experiment results, distress classes with sample sizes smaller than 200 do not yield acceptable detection performance, suggesting that insufficient data limits the model's ability to generalize effectively. Moreover, some distress classes, despite having large sample sizes, still exhibit sub-optimal performance due to the inherent complexity or variability of their features. To address the imbalance in data representation, collecting additional samples for underrepresented classes and applying data augmentation should be further explored to enhance the model's performance for these distress types. The inherent characteristics of different distress classes and pavement surface types also significantly affect distress detection performance. Distress types such as sealed cracking and joint have distinct visual patterns, making it easier for models to learn and identify their features com-pared to distress types with more subtle or inconsistent appearances, such as unsealed longitudinal and transverse cracking. Additionally, different pavement surface types, like asphalt and concrete, exhibit unique textural properties that influence the model's ability to detect distress effectively. For instance, cracks on asphalt surfaces might be less pronounced compared to those on concrete, resulting in challenges for the model to differentiate them from the complex surface textures. Addressing these challenges requires tailored model configurations, extensive data collection across different pavement types, and feature engineering techniques that enhance the model's ability to learn from diverse visual characteristics.

5.4. Performance Comparison

In this section, we compare the predictions made by the proposed model with the evaluation results from the commercial system, PathView. Since there is no ground truth available, a direct performance comparison between the two systems is not possible. Instead, we calculate the differences in distress detection based on pavement sections and identify sections or distress classes where significant discrepancies occur. The goal is to gain insights into the areas where the proposed model and PathView differ, helping to highlight potential strengths and weaknesses of the proposed approach in comparison to the commercial system.

5.4.1 ACP

Table 5.19 compares the performance of the proposed method to the existing PathView system in detecting various classes of pavement distress across 4 ACP segments.

Table 5.19 Comparison of Common ACP Distress Detection between PathView and Proposed Method

Distress class	Unit	Detection summary		Difference	
		PathView	Proposed	Absolute	Relative
Transverse crack	(ft)	1,948.6	2,493.1	-544.5	-21.8%
Longitudinal crack	(ft)	2,100.3	2,333.6	-233.3	-10.0%
Alligator crack	(ft^2)	6	8.3	-2.3	-27.7%
Block crack	(ft^2)	3,934.7	0	-3,934.7	-
SH0073-R_8.234_13.818 (5.584 mile)					
Transverse crack	(ft)	7,440.5	5,667.9	1,772.6	31.27%
Longitudinal crack	(ft)	20,849	30,443.4	-9,594.4	-31.52%
Alligator crack	(ft^2)	2,132.5	133.4	1,999.1	1,498.58%
Block crack	(ft^2)	0	0	0	-
IH0010-A_851.930_854.253 (2.323 mile)					
Transverse crack	(ft)	2,090.7	1,877.0	213.7	11.4%
Longitudinal crack	(ft)	4,375.3	4,217.8	157.5	3.7%
Alligator crack	(ft^2)	711.7	0	711.7	-
Block crack	(ft^2)	0	0	0	-
SS0380-R_3.671_5.521 (1.85 mile)					
Transverse crack	(ft)	2,675.1	2,224.7	450.4	20.2%
Longitudinal crack	(ft)	7,643.3	6,312.6	1,330.7	21.1%
Alligator crack	(ft^2)	1,519.7	54.2	1,465.5	2,703.9%
Block crack	(ft^2)	0	0	0	-

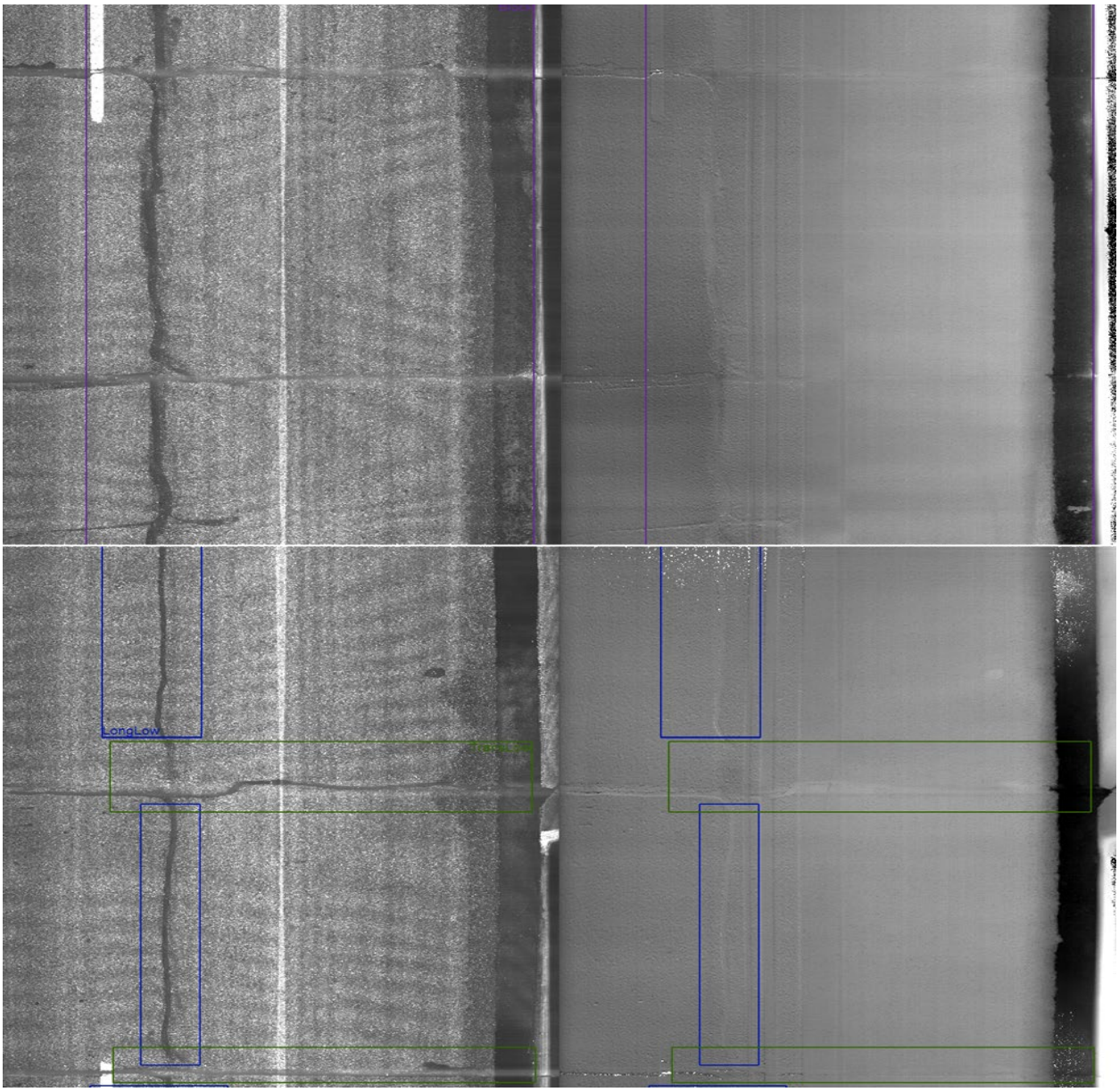


Figure 5.32 Distress predictions on Section SH0347-R_10.819_11.338

Detection results with significant changes are marked with red colors. Considerable change in this context refers to instances where the relative difference between the two systems' detection metrics is greater than 100%, or where one system detects a distress class that the other completely misses. Notable examples include: 1) Block Crack in SSH0347-R_10.819_11.338: Detected by PathView as 3934.7 ft² but completely missed by the proposed system. 2) Alligator Crack in SH0073-R_8.234_13.818, SS0380-R_3.671_5.521, and IH0010-A_851.930_854.253: Large quantities detected by PathView but very few by the proposed system. The original image analysis suggests PathView might be misclassifying reflective Transverse and Longitudinal cracks as Block cracks (Figure 5.32), which explains the overestimation in Block crack detection and also the underestimation in Transverse and Longitudinal cracks. For Alligator crack prediction, significant quantities of false positive predictions by PathView are observed in the selected section.

Figure 5.32 shows distress predictions on Section SH0347-R_10.819_11.338 from PathView system: (1) Top row images show that intersected reflective Longitudinal crack and Transverse crack are classified as Block crack (shown in purple bounding box), (2) Bottom row images show the reflective Longitudinal crack and Transverse crack from the same pavement section that are correctly classified.

Figure 5.33 shows alligator predictions (shown in red bounding boxes) from PathView system: (1) from left to right, intensity image and range image; (2) from top to bottom, images from IH0010-A_851.930_854.253, SS0380-R_3.671_5.521, and SH0073-R_8.234_13.818.

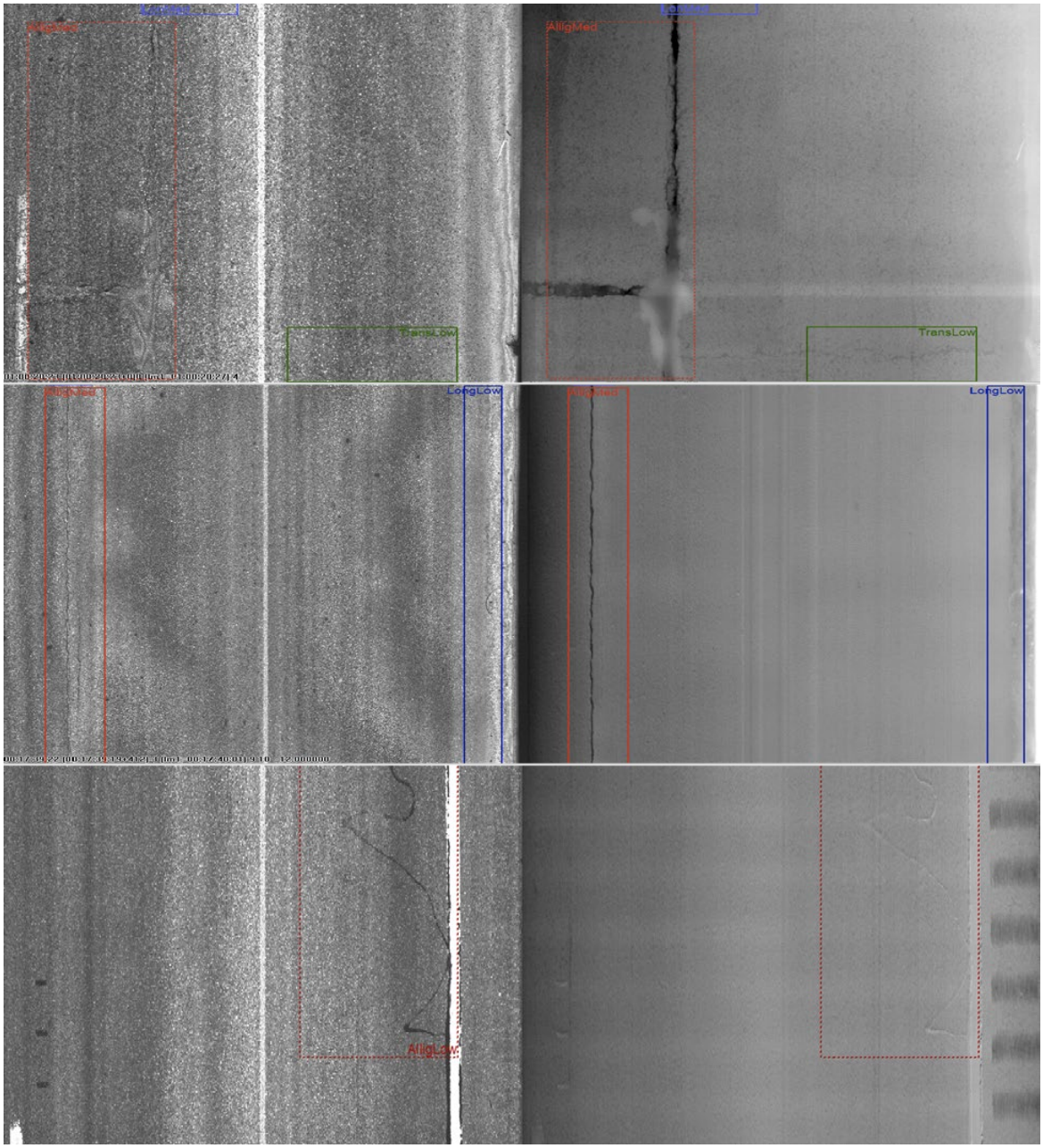


Figure 5.33 Alligator predictions (shown in red bounding boxes) from PathView system

5.4.2 JCP

Table 5.20 compares the performance of the proposed method to the existing PathView system in detecting various classes of pavement distress across 4 JCP segments. Compared to the earlier ACP distress detection table, this JCP distress detection table shows even larger discrepancies in the detection of Transverse and Longitudinal cracks between the PathView and the proposed system. These larger discrepancies might indicate a variation in how the two systems interpret crack width, length, or severity or possibly how they are calibrated for different types of pavement surfaces. A notable trend observed is that the PathView system consistently detects more Transverse cracks compared to the proposed model, often by a significant margin. This discrepancy may stem from the JCP data revision phase, where many thin and ambiguous transverse cracks were excluded to enhance model performance. Such exclusions, while aimed at improving accuracy, might lead to the proposed model underperforming in detecting low-severity cracks. This indicates a potential trade-off between the model's precision and its sensitivity to less pronounced distress features.

Table 5.20 Comparison of Common JCP Distress Detection between PathView and Proposed Method

Distress class	Unit	Detection summary		Difference	
		PathView	Proposed	Absolute	Relative
US090-K_730.772_733.845 (3.073 mile)					
Transverse crack	(ft)	127	63.9	63.1	98.7%
Longitudinal crack	(ft)	589.1	925	-335.9	-36.3%
Joint	(ft)	13,129.8	12,980.3	149.4	1.2%
US0069-X_332.500_327.715 (4.485 mile)					
Transverse crack	(ft)	2,357.4	1,672.8	684.6	40.9%
Longitudinal crack	(ft)	1,263.2	1,340.7	-77.5	-5.8%
Joint	(ft)	13,310.1	17,236.2	-3,926.1	-22.8%
SS0380-R_1.478_2.963 (1.476 mile)					
Transverse crack	(ft)	157.8	83	74.8	90.1%
Longitudinal crack	(ft)	55.2	43.8	11.4	26.0%
Joint	(ft)	5,887.7	5,771.7	156.0	2.0%
IH0010-X_854.028_853.356 (0.668 mile)					
Transverse crack	(ft)	389.9	232.9	157	67.4%
Longitudinal crack	(ft)	213	556.1	-343.1	-61.7%
Joint	(ft)	1,531.1	1,672.3	-141.2	-8.4%

Across most sections, the detection of Joint distress exhibits remarkable consistency between the two models, indicating that both systems might be using similar criteria are equally effective in recognizing joint-related issues. However, a notable exception is observed in section US0069-X, where the proposed system detects significantly more Joint area (17,236.2 ft) com-pared to

PathView (13,301.1 ft), showing an increase of 3,926.1 ft (22.8%). This outlier suggests that while both systems generally align well in Joint detection, there can be substantial variations depending on specific pavement conditions or segment characteristics. Substantial discrepancies and occasional consistencies underscore the need for further statistical analysis to delve deeper into the causes behind these differences. Such analysis could include regression analysis to understand the dependency of detection on various factors like pavement type or environmental conditions, and a confusion matrix to evaluate the type and frequency of misclassifications made by each system. This would provide more detailed insights into each system's strengths and weaknesses, facilitating improvements in pavement distress detection algorithms.

5.5. Summary

In this summary, we provide an overview of the key findings from this task on distress detection and segmentation. First, we summarize the performance of the models across different datasets, highlighting their strengths and weaknesses. Next, we discuss the various factors that influence the model's performance, including the size of the distress samples, the characteristics of the pavement surfaces and distresses, and the methods employed for detection. Finally, we outline potential areas for future work, such as expanding the dataset by including more samples from underrepresented distress classes and exploring new model architectures to improve detection accuracy and robustness further.

For segmentation, the performance across the JCP, ACP, and CRCP datasets showed that the models performed well on uniform surfaces like JCP, but struggled more on complex textures, such as in ACP and CRCP, where crack boundaries were harder to detect. While the models effectively outlined distress areas, additional work is required to obtain accurate extent and classification results for individual cracks. Furthermore, the dataset sizes used in these experiments were relatively small, and a larger dataset is needed to advance this method towards practical application.

The distress detection performance demonstrated several successful outcomes and areas for improvement. On the positive side, the model generally delivered satisfactory results for distress classes with sufficient samples, such as Sealed longitudinal crack and Sealed transverse crack in ACP, as well as Joint and Patches in JCP. Additionally, when compared with the PathView system, the model excelled in precision, producing significantly fewer false positives. However, there are several areas needing improvement. The model struggled with underrepresented distress classes, such as Potholes and Block crack in ACP, and Corner Break, Punchout, D-cracking, Failed concrete patch, and Popout in JCP, where performance was notably poor. Despite having many samples, the model's performance on Transverse crack and Longitudinal crack in ACP was also suboptimal, likely due to the complex textures of the pavement surface. Furthermore, experiments indicated that the type of images used (intensity, range, or both) had a significant impact on performance: for ACP, using both images yielded the best results, while for JCP, using intensity images alone performed just as well as using both. These findings suggest the need for more data, particularly for underrepresented classes, and a refined approach to image type selection for optimal performance. Future work should focus on addressing the limitations identified in the current distress detection model to improve overall performance and

applicability. First, expanding the dataset by collecting more samples, particularly for underrepresented distress classes such as Potholes, Block crack, Corner Break, Punchout, and D-cracking, will be essential for better generalization and accuracy across all distress types. Additionally, improving the model's ability to handle complex pavement textures, particularly for Transverse crack and Longitudinal crack in ACP, is crucial. Exploring advanced image processing techniques or incorporating multi-scale feature extraction may help in these areas.

The primary goal of this study is to develop AI/ML models for the implementable detection and measurement of pavement distresses on ACP, JCP, and CRCP. As a very important task of the research approach to utilizing modern neural networks trained on a dedicated library of preprocessed and labeled image datasets, Chapter 7 details a pilot study that employs both the 2D/3D image data library and newly collected pavement image data sourced from the TxDOT's vendor for model testing and further model refinement.

Chapter 6 Practical Tools for Pavement Condition Assessment

This chapter's primary purpose is to document the procedure of using the distress detection results from Chapter 5 to obtain the distress score, which is part of the calculation of PMIS condition score. In this chapter, details of how the detection results are post-processed and how the distress score is calculated are documented.

6.1 Objectives

The objective of this chapter is to integrate automated distress detection results into the calculation of the distress score, a critical component of the Pavement Management Information System (PMIS) condition score. By leveraging data-driven distress evaluation, this TM aims to improve the accuracy, consistency, and reliability of pavement condition assessments, ultimately enhancing pavement management decision-making.

According to the Overview of Calculation of PMIS Condition Score (TxDOT, 2009), the distress score is computed from distress ratings, which are converted into utility values before contributing to the final condition score (Figure 6.1). In the case of asphalt concrete pavement (ACP), rutting measurements are considered in distress scoring. The distress score, combined with the ride score (derived from ride quality measurements and adjusted by ADT-Speed Limit categories), determines the overall PMIS Condition Score.

The current distress detection system developed in Chapter 5 can successfully identify most of the pavement distresses, such as cracking (e.g., transverse, longitudinal, block cracking), pothole, and patch. These detected distresses provide valuable insights into pavement health and serve as the foundation for distress score computation. However, due to the research focus of this research project, certain distress types—such as flushing and rutting—are not currently captured in the detection results. In this case, the values of these distress classes could be manually input if necessary.

This TM directly supports the broader project goal of developing an AI/ML-based pavement condition assessment that enhances the accuracy and efficiency of existing automated pavement inspections. While current inspection methods are already automated, they suffer from ac- curacy limitations and lack an efficient validation process. By ensuring that AI-driven distress detection results are effectively incorporated into PMIS distress scoring, this TM will contribute to a more reliable and scalable validation framework. This will enable highway agencies to efficiently verify automated condition reports, reduce inconsistencies in pavement assessments, and support more data-driven and confident decision-making in maintenance planning.

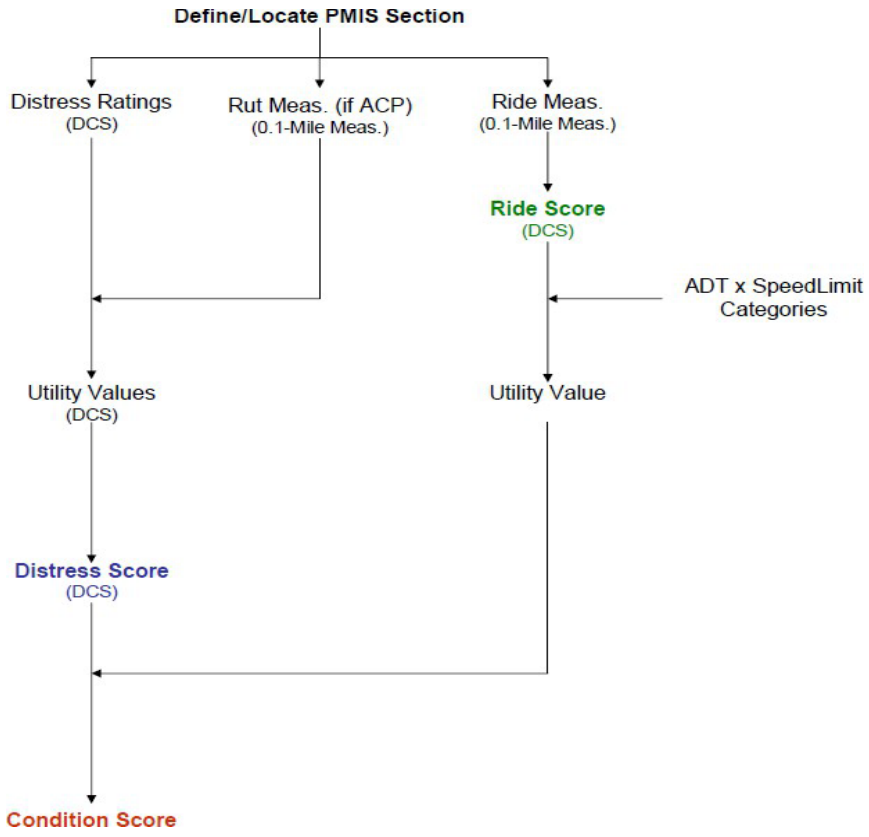


Figure 6.1 Process used to calculate PMIS condition score

6.2 Distress Detection Post-Process

This section describes the post-processing rules applied to ensure compliance with TxDOT's distress classification and scoring framework. Post-processing is essential for converting raw AI-based distress detections into structured outputs that align with predefined distress categories used in PMIS scoring. Python program was used to handle all the necessary post-processing of the raw detected distress datasets.

6.2.1 Post-processing rule for ACP

According to TxDOT's Rater's Manual (TxDOT, 2023), 10 distress types are considered for ACP: Shallow Rutting, Deep Rutting, Patching, Block Cracking, Alligator Cracking, Longitudinal Cracking, Transverse Cracking, Raveling, Flushing, and Failures (see Table 6.1). Among these, Raveling and Flushing are excluded from the utility value calculation. Due to the research scope of this project, the values for Shallow Rutting and Deep Rutting will be either set to a constant or manually input for distress score calculation.

Table 6.1 Distress types considered for distress score calculation (ACP)

No.	Distress Type (TxDOT)	Distress Type (Detection model)
1	Rutting - Shallow	Not measured
2	Rutting - Deep	Not measured
3	Patching	Potholes (patched)
4	Block Cracking	Block Cracking
5	Alligator Cracking	Alligator Cracking
6	Longitudinal Cracking	Unsealed Longitudinal Cracking, Sealed Longitudinal Cracking, Lane Longitudinal Cracking
7	Transverse Cracking	Unsealed Transverse Cracking, Sealed Transverse Cracking
8	Raveling	-
9	Flushing	-
10	Failures	Potholes (unpatched and other)

Since the format of the distress detection results differs from TxDOT's distress definitions, a post-processing step is required to convert the raw detection outputs into the format necessary for compliance with TxDOT's distress classification and scoring framework. The following are how detection results of the six distress types required by the PMIS distress score calculation are extracted from the distress detection model:

Longitudinal Cracking: As shown in Table 6.1, Unsealed Longitudinal Cracking, Sealed Longitudinal Cracking, and Lane Longitudinal Cracking are consolidated into a single distress class under Longitudinal Cracking. Since the captured images may include areas beyond the target pavement lane, longitudinal cracks detected near the left and right edges of the image are excluded. This approach statistically filters out most longitudinal cracks from adjacent lanes, minimizing false detections (see Figure 6.2). Additionally, it effectively removes a significant portion of Lane Longitudinal Cracking, as these cracks predominantly occur along the lane edges.

Transverse Cracking: As shown in Table 6.1, Unsealed Transverse Cracking and Sealed Transverse Cracking are consolidated into a single distress class under Transverse Cracking. Since the distress detection model captures all size of transverse cracks, the results are filtered according to the Raters Manual: 1) cracks shorter than 5 feet are ignored, 2) cracks with length between 5 and 10 feet are counted as half a crack, and 3) cracks 10 feet and longer are counted as a full crack (see Figure 6.3).

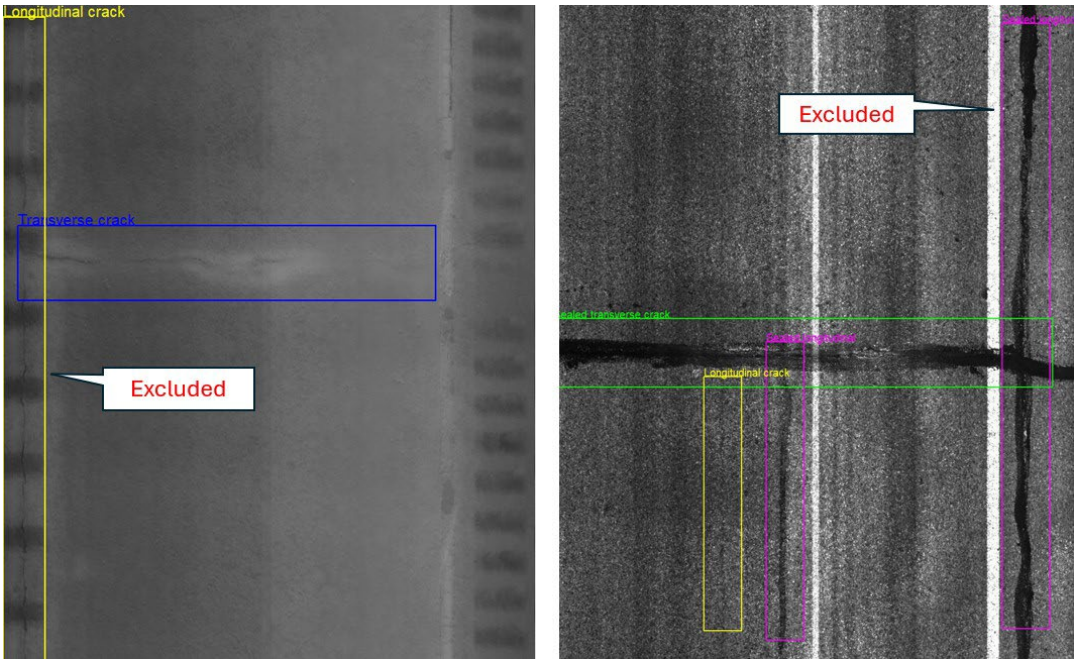


Figure 6.2 Detected longitudinal cracks close to left and right borders are excluded.

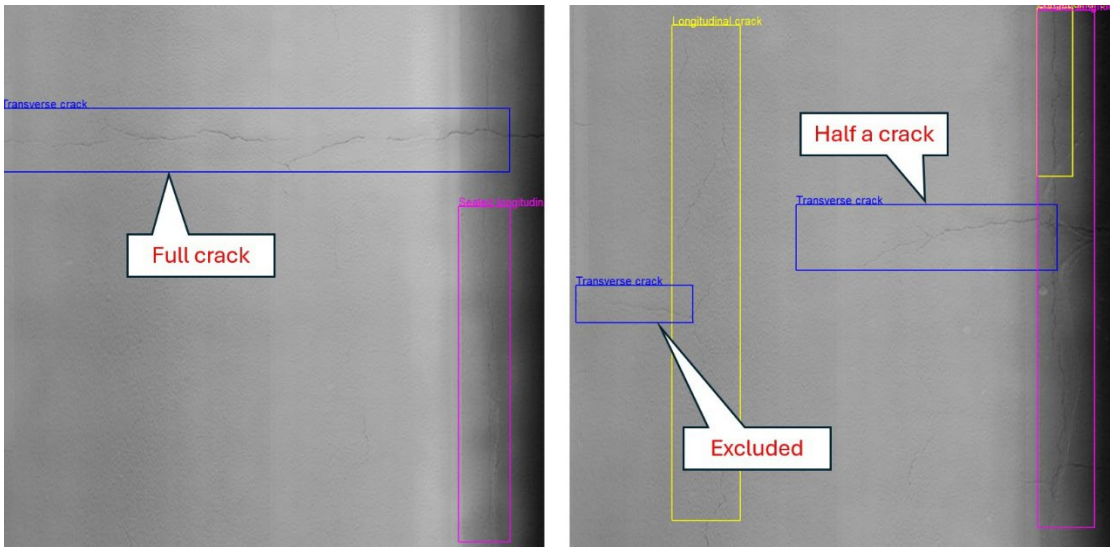


Figure 6.3 Transverse cracks with different lengths are filtered differently.

Block Cracking: The detection results for Block Cracking from the model are directly used to calculate its utility score. Since Block Cracking is measured based on its extent along the longitudinal direction, only the heights of the detected bounding boxes are extracted for further processing, with no additional modifications applied. Additionally, to prevent over-prediction from influencing the distress score calculation, any other detected distress types within a Block Cracking bounding box are excluded, except for Alligator Cracking.

Alligator Cracking: The detection results for Alligator Cracking from the model are directly used to calculate its utility score. Like Block Cracking, Alligator Cracking is also measured based on its extent along the longitudinal direction; only the heights of the detected bounding boxes are extracted for further processing, with no additional modifications applied.

Patching: Patching is partially detected by the distress detection model under the distress type Pothole (patched). According to the Rater's Manual, level-ups, overlays, seal coats, and strip seals shorter than 500 feet are classified as Patching. However, since a single pavement image typically covers a range of 5 to 20 feet, these types of Patching cannot be reliably identified from a single image alone. In this TM, the utility of Patching is manually set instead of being calculated using the detection results.

Failure: Failure is partially detected by the distress detection model under the distress type Pothole. Since only one failure can be recorded per 40 feet of pavement, the detection results are adjusted to ensure that multiple detected failures within a 40-foot segment are counted as a single failure. As the detection model does not include rutting, failures caused by severe rutting cannot be identified. As a result, the calculated utility value for Failure may be higher than its actual condition, potentially leading to an overestimation of pavement quality.

6.2.2 Post-processing rule for JCP

According to TxDOT's Rater's Manual (TxDOT, 2023), six distress types are considered for Jointed Concrete Pavement (JCP): Failed Joints and Cracks, Failures, Shattered Slabs, Slabs with Longitudinal Cracks, Concrete Patches, and Apparent Joint Spacing (see Table 6.2). Among these distress types, Apparent Joint Spacing is excluded from the utility value calculation.

Table 6.2 Distress types considered for distress score calculation (ACP)

No.	Distress Type (TxDOT)	Distress Type (Detection model)
1	Failed Joints and Cracks	Failed joints and cracks, Transverse crack, Asphalt patch
2	Failures	Corner break, Punchout, Asphalt patch, Failed concrete patch, D-cracking, Popout, Transverse crack, Longitudinal crack
3	Shattered Slabs	Corner break, Punchout, Asphalt patch, Failed concrete patch, D-cracking, Popout, Transverse crack, Longitudinal crack
4	Slabs with Longitudinal Cracks	Unsealed longitudinal crack, Sealed longitudinal crack
5	Concrete Patches	Concrete patch
6	Apparent Joint Spacing	-

Failed Joints and Cracks: Failed Joints and Cracks include both spalled joints and transverse cracks, and asphalt patches of spalled joints and transverse cracks. For the distress detection

results, the category of Failed joints and cracks include the spalled joints, spalled transverse cracks, and spalled joints patched with asphalt. The asphalt patches of transverse cracks are screened by checking the detected asphalt patches and transverse cracks. If a detected transverse crack is overlapped with one or multiple asphalt patches, this detected transverse crack is counted as one Failed Joints and Cracks instance.

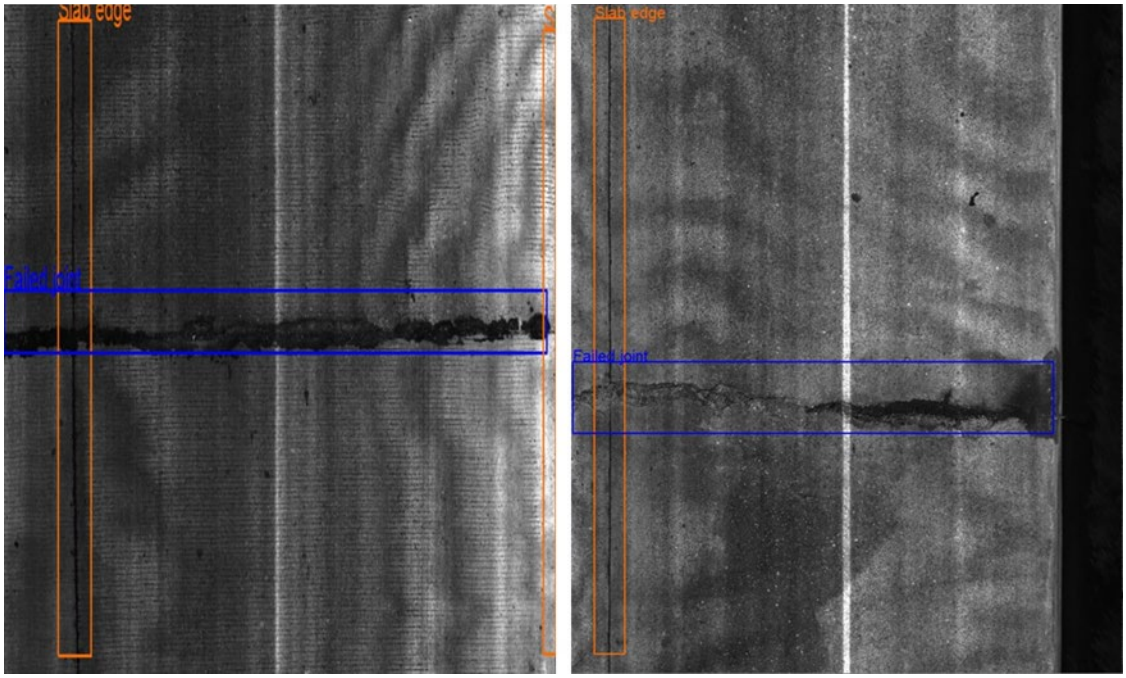


Figure 6.4 Two samples of Failed joints and cracks: 1) a joint patched with asphalt, and 2) a seriously spalled transverse crack

Failures: Failures are the most complex distress category to quantify in JCP. According to the Rater's Manual, Failures encompass various distress types, including corner breaks, punchouts, asphalt patches, failed concrete patches, D-cracking, spalls longer than 10 inches and wider than 12 inches, and popouts wider than 12 inches and deeper than 3 inches. Additionally, a Failed Joints and Cracks instance may also be classified as a Failure if certain conditions are met. Table 6.3 outlines the process for screening detected distresses and converting them into Failure counts. Notably, if a slab is classified as a Shattered Slab, all other detected distresses, including Failures, are excluded from the count.

Shattered Slabs: Shattered Slabs are defined as slabs with either five or more Failure instances or one or more failures covering more than half of the slab's area. Failed Joints and Cracks are not considered Failures unless specific conditions are met. The process for counting Shattered Slabs consists of two sequential steps: 1) The total number of Failures on the slab is counted. If there are five or more, the slab is classified as a Shattered Slab. 2) If the count is less than five, the total area of all Failures is assessed. If they collectively cover more than half of the slab, the slab is classified as a Shattered Slab. Notably, once a slab is classified as a Shattered Slab, all other detected distresses are excluded from the count.

Table 6.3 Screening and converting detected distress into Failure counts

No.	Distress Type (Detection model)	Conditions for counting as Failures
1	Failed Joints and Cracks	If an asphalt patch overlaps with a failed joint and is longer than 10 inches and wider than 12 inches, a Fail- ure instance is counted. If this occurs on both sides of the joint, two Failure instances are counted (Figure 6.5).
2	Corner break	If the detected corner break is wider/longer that 1.0 foot, one failure is counted.
3	Punchout	If a detected punchout is longer than 10 feet, then one failure is counted for each 10 feet of length.
4	Asphalt patch	Shallow-depth asphalt patches are not counted as failures. While these patches usually have irregular shapes, only detected asphalt patches with rectangular shapes are counted for failures.
5	Failed concrete patch	A failed concrete patch is counted as one Failure instance.
6	D-cracking	A D-cracking instance is counted as one Failure instance.
7	Popout	A detected popout greater than 12 inches wide is counted as one failure.
8	Transverse crack	If a detected transverse crack overlaps with one or more asphalt patches, and at least one patch is longer than 10 inches and wider than 12 inches, the transverse crack is counted as a Failure instance.
9	Longitudinal crack	If a detected longitudinal crack overlaps with one asphalt patch, and the patch is longer than 10 inches and wider than 12 inches, the transverse crack is counted as a Failure instance.

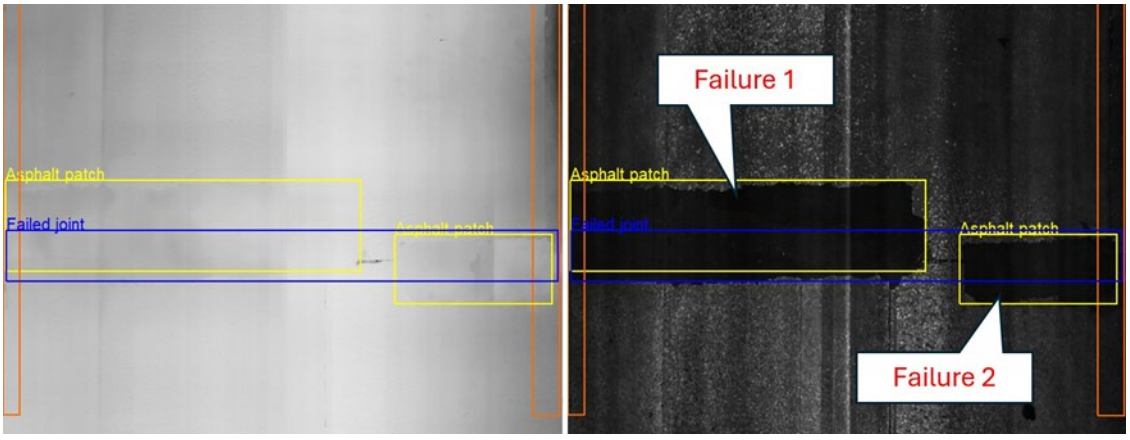


Figure 6.5 A sample of two Failure instances on each side of the detected joint

Slabs with Longitudinal Cracks: Slabs with Longitudinal Cracks are defined as slabs containing a longitudinal crack that extends from one transverse joint to the next transverse joint, or from a transverse joint to an edge joint, with a total length exceeding half the slab's length. For classification, a transverse joint includes both actual joints and cleanly-defined transverse cracks. If a slab is divided into multiple smaller slabs by apparent joints (i.e., cleanly-defined transverse cracks), and a longitudinal crack with severe spalling or faulting extends from one apparent joint to the next, each resulting smaller slab is classified as a Slab with Longitudinal Cracks. To qualify, the total detected length of longitudinal cracks within a slab must exceed half of the slab's length.

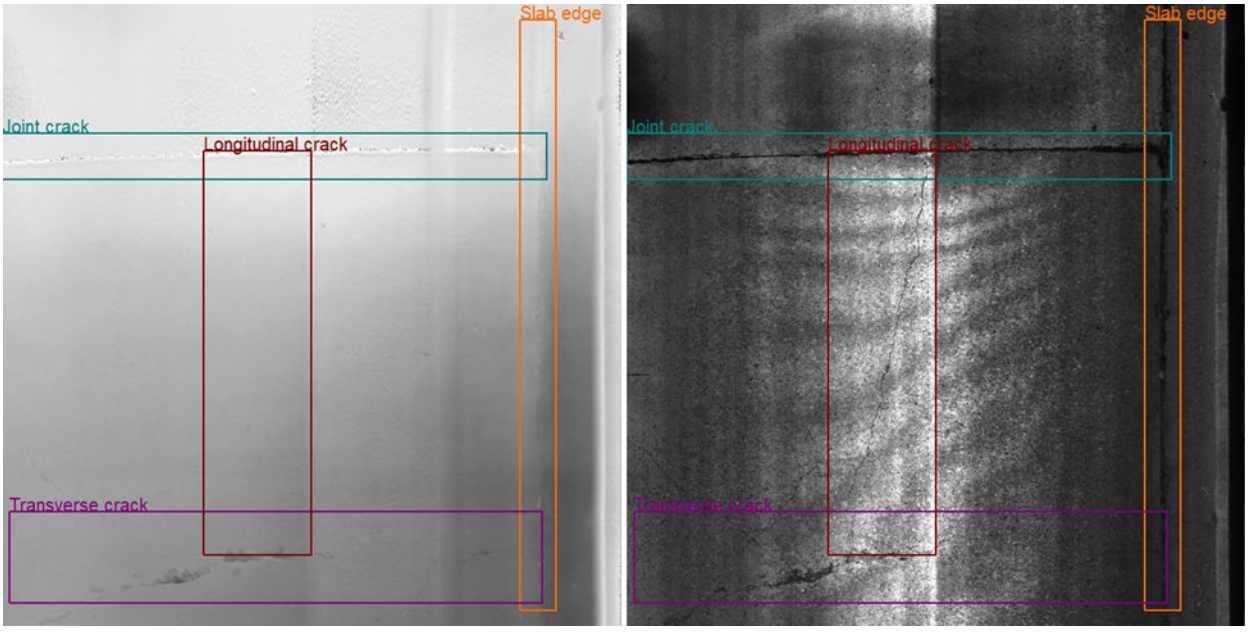


Figure 6.6 A sample of Slabs with Longitudinal Cracks instance defined by a joint, a cleanly-defined transverse crack, and a longitudinal crack extending from the joint to the transverse crack

Concrete Patches: Concrete Patches are defined as localized areas of newer concrete placed to the full depth of the existing slab to correct surface or structural defects. To qualify as a Concrete Patch, the patch must be greater than 10 inches in length, which is the average depth of a JCP slab. The width of the patch is not a factor in the rating. For longer patches, they are rated as one patch for every 10 feet. For example, a 15-ft concrete patch would be rated as 2 concrete patches. If an entire slab has been replaced, it is not rated as Concrete Patches. Additionally, Failed Concrete Patches are considered as Failures and are excluded from being counted as Concrete Patches.

6.2.3 Post-processing rule for CRCP

According to TxDOT's Rater's Manual (TxDOT, 2023), 5 distress types are considered for Continuously Reinforced Concrete Pavement (CRCP): Spalled Cracks/Longitudinal Cracking, Punchouts, Asphalt Patches, Concrete Patches, and Average Crack Spacing (see Table 6.4).

Among these distress types, Average Crack Spacing is excluded from the utility value calculation.

To clarify, Spalled Cracks and Longitudinal Cracks are distinct distress types in CRCP evaluation, though they can occur together. Spalled Cracks refer to concrete deterioration along crack edges, where fragments break off due to water infiltration, freeze-thaw cycles, or traffic-induced stress. While spalling may develop along Longitudinal Cracks, the two are classified separately in PMIS distress assessment. Longitudinal Cracking is a structural distress that occurs due to shrinkage, reinforcement issues, or thermal movement, whereas Spalled Cracks are rated based on the severity of material loss along these cracks. Therefore, Spalled Cracks do not inherently include Longitudinal Cracks, but they are often associated with them in distress evaluations.

Table 6.4 Distress types considered for distress score calculation (CRCP)

No.	Distress Type (TxDOT)	Distress Type (Detection model)
1	Spalled Cracks	Spalled transverse crack, Transverse crack, Asphalt patch
2	Punchouts	Punchout, Asphalt patch, Concrete patch
3	Asphalt Patches	Asphalt patch
4	Concrete Patches	Concrete patch
5	Average (Transverse) Crack Spacing	--

Spalled Cracks: Detection results for spalled transverse cracks are generally used to calculate the utility value for spalled cracks (Figure 6.7), while spalled longitudinal cracks are not considered, as specified in the Rater's Manual. Since the manual does not define a specific length to qualify as a spalled crack, all detected spalled transverse cracks are included. Additionally, spalled cracks filled with asphalt should still be counted. Therefore, detection results for asphalt patches and transverse cracks are screened, and any transverse crack overlapping or connected to an asphalt patch is counted as a single spalled crack.

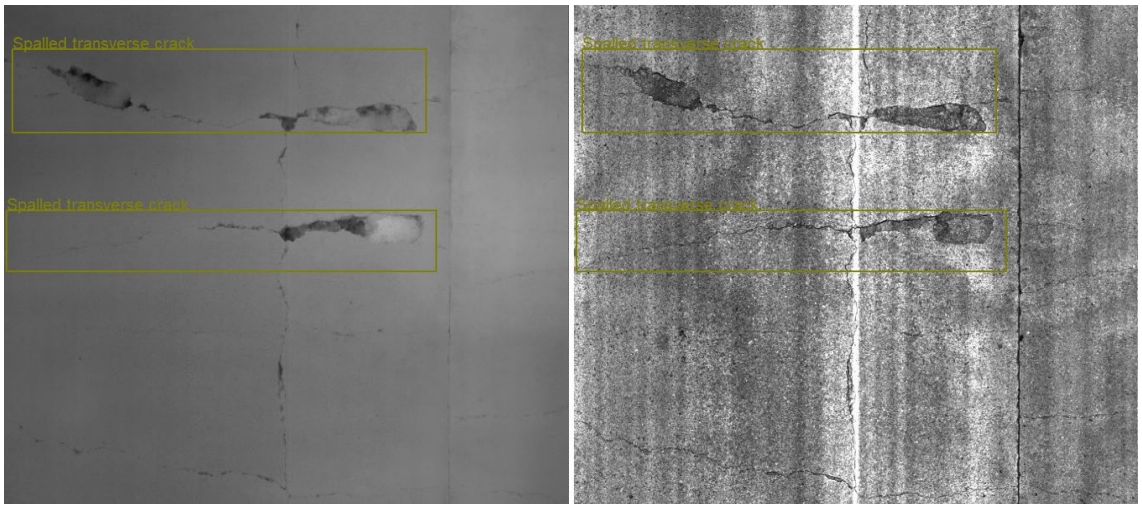


Figure 6.7 A sample of two Spalled Cracks instances

Punchouts: Detection results for punchouts (Figure 6.8) are primarily used to calculate the utility value for Punchouts. Punchouts with both width and length shorter than 12 inches are excluded, while those exceeding 10 feet in length (typically along the longitudinal direction) are counted as one punchout for every 10 feet. Since punchouts that meet the criteria for a patch should be rated as both punchout and patch, all detected punchouts are included, regardless of overlap with a detected patch.

Asphalt Patches: According to the Rater's Manual, an asphalt patch should be placed to the full depth of the surrounding concrete slab; therefore, not all detected asphalt patches should be counted. Detected patches shorter than 12 inches are excluded, while those exceeding 10 feet in length are counted as one patch for every 10 feet. Similar to Punchouts, all detected asphalt patches are included, regardless of overlapping with a detected spalled crack.

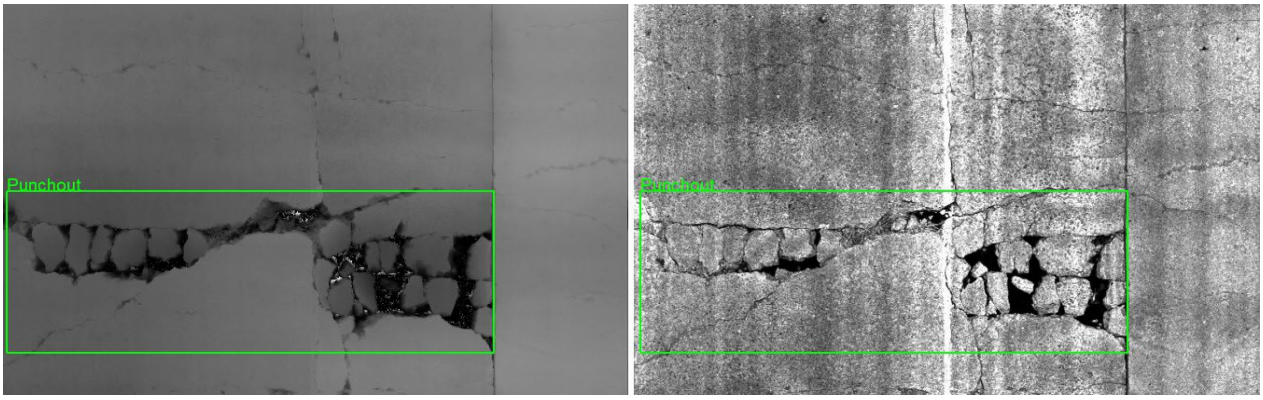


Figure 6.8 A sample of Punchouts

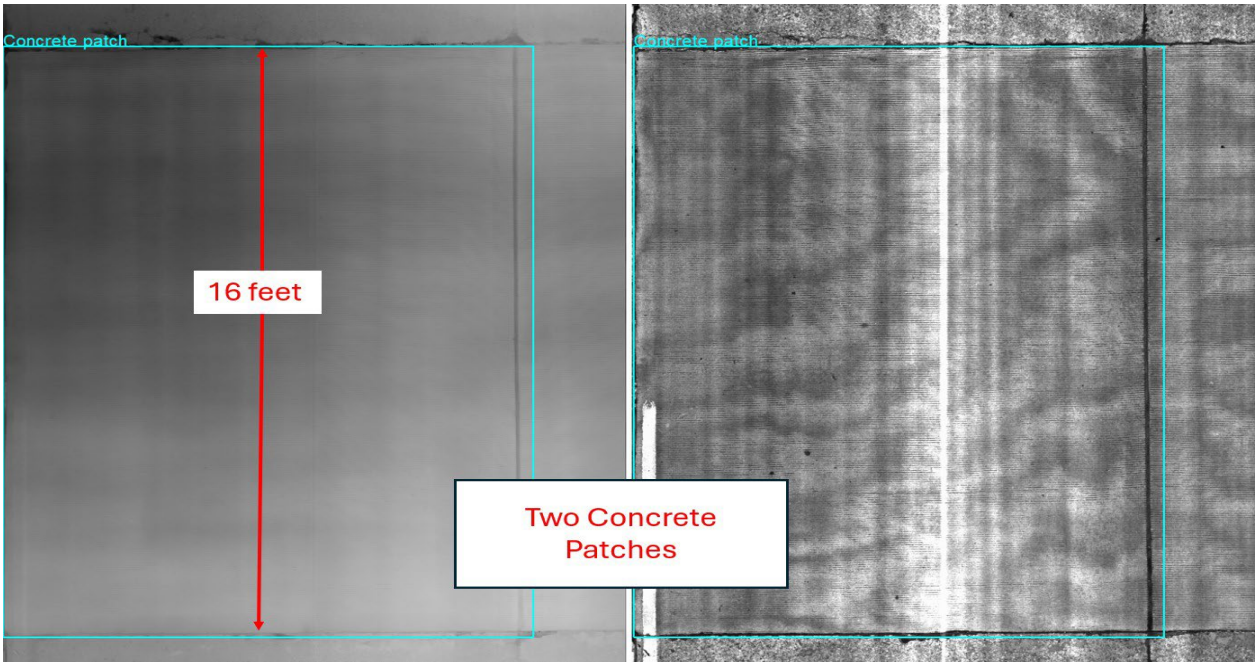


Figure 6.9 A sample of Concrete Patches instance with length of 16 feet counted as 2 patches

Concrete Patches: Like asphalt patches, concrete patches are placed to the full depth of the existing slab. Detected concrete patches shorter than 12 inches are excluded, while those exceeding 10 feet in length are counted as one patch for every 10 feet (Figure 6.9). All qualified detected concrete patches are included, regardless of overlap with a detected spalled crack. A special case arises when two detected concrete patches are adjacent or partially overlapping; these are counted as a single patch unless their combined length exceeds 10 feet, in which case they are counted as one patch for every 10 feet. Additionally, concrete patches longer than 20 feet may extend beyond a single image, requiring further consideration in future evaluations.

6.3 Distress Score Calculation

6.3.1 General data flow description

The distress score calculation process initiates with outputs generated by the AI/ML model, which identifies pavement surface distresses. These outputs comprise bounding boxes that pinpoint distress locations along with classifications indicating specific distress types (e.g., longitudinal crack, transverse crack). The detection results are subsequently post-processed at the pavement section level according to predefined criteria outlined in Section 6.2. This step converts the raw detection data into PMIS ratings based on guidelines provided in the Rater's Manual (TxDOT, 2023). These PMIS ratings represent the severity and condition of each identified pavement distress within the evaluated pavement section. Following this, the PMIS ratings are normalized into L_i Value values. Normalization scales each rating onto a standardized numerical range, placing all pavement sections on a level playing field and facilitating consistent evaluation and comparison across various types and severities of pavement distress. The L_i Value values are then converted into utility values, which range between 0 and 1. Utility values make it possible to compare different distress types and ride quality across various pavement sections, providing a consistent, reliable, and defendable description of each section's condition. Utility values reflect the pavement's relative condition and the urgency for maintenance intervention, calculated through utility functions specifically designed for each distress type's unique impact and importance. In the final step, the calculated utility values are combined using a predetermined weighting system, resulting in a comprehensive distress score. This distress score encapsulates the overall pavement condition, enabling informed decision-making for maintenance and management activities.

6.3.2 Converting detection results to PMIS ratings

6.3.2.1 ACP

Figure 6.10 shows the flow chart for ACP distress score calculation. Seven distress types are included, such as Sealed/Unsealed/Lane longitudinal crack, Sealed/Unsealed transverse crack, block crack, and alligator crack.

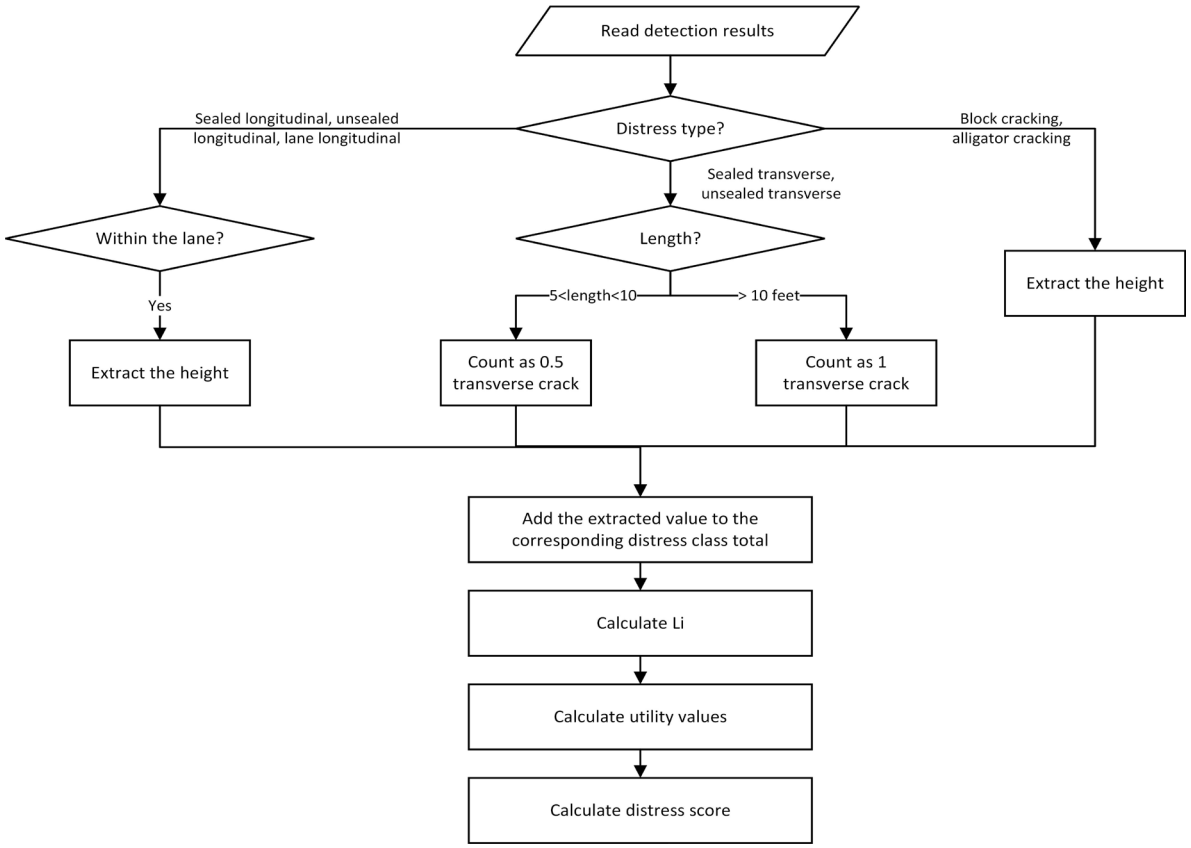


Figure 6.10 Flowchart for ACP distress score calculation

The process begins by reading the detection results and identifying the distress type. If the distress is a longitudinal crack (sealed, unsealed, or lane longitudinal), it is checked whether it lies within the lane; if so, its height is extracted. For transverse cracks (sealed or unsealed), the length is evaluated—if the length is between 5 and 10 feet, it is counted as 0.5 transverse crack; if it is 10 feet or more, it is counted as 1 transverse crack. In the case of block cracking or alligator cracking, the height is extracted directly. All extracted values—whether height or equivalent crack count—are then added to the corresponding distress class total, setting the stage for further calculations in the distress scoring process.

6.3.2.2 JCP

For JCP, the flow chart is a bit more complicated compared to that of ACP (Figure 6.11). Due to that the condition assessment is performed by slabs, the images of the section in the original cut (Figure 6.12) are combined and then re-cut by the apparent joints (Figure 6.13). AI/ML models are applied again to the re-cut images, and the new detection results are imported for PMIS rating conversion.

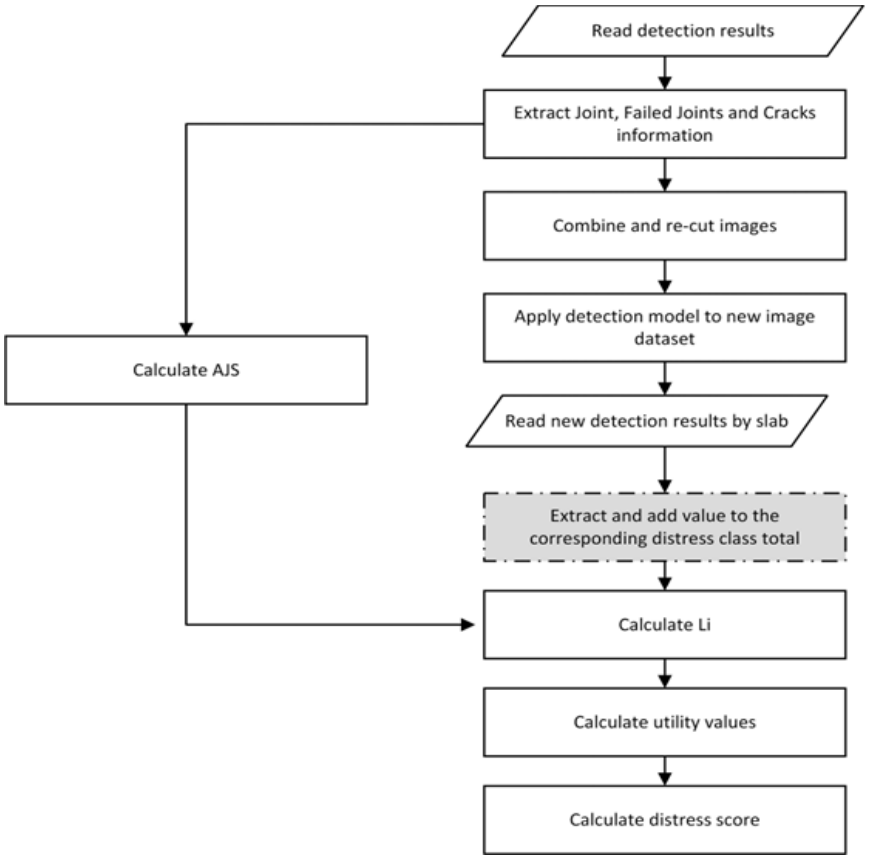


Figure 6.11 Flowchart for JCP distress score calculation

The detailed conversion flow chart is shown in Figure 6.14. The flowchart describes the process of converting detected JCP distress data into PMIS ratings by analyzing slab-level distress types and aggregating them into section-level indicators. It begins with reading new detection results by slab and identifying the distress type. For failed joints and transverse cracks, if they have an asphalt patch longer than 10 inches and wider than 12 inches, they are counted as failures; otherwise, failed joints are separately counted as such. For longitudinal cracks, if there's an asphalt patch wider than 10 inches and longer than 12 inches, it's counted as a failure. No matter if a longitudinal crack is overlapped with an asphalt patch, the crack length is added to the slab's total longitudinal crack length. Other distresses like corner breaks, punchouts, asphalt patches, failed concrete, D-cracking, and popouts are also directly counted as failures. Concrete patches are only counted if their length exceeds 12 inches. If they exceed 10 feet, they are counted for every 10 feet. A special case is that, if a concrete patch is overlapped with a joint or an apparent joint, this concrete patch is counted as two patches. After identifying the failures per slab, values are added to totals such as the number of failed joints/cracks, total slab failures, and total longitudinal length. If a slab has fewer than 5 failures and the overall coverage of the failure is less than 50% of the slab, it contributes to either the count of failed joints, failures, concrete patches, or slabs with longitudinal cracks, depending on the crack length. Slabs with 5 or more failures or failure coverage larger than 50% are classified as shattered slabs. At the section level, these slab counts are then compiled into metrics for failed joints, shattered slabs, failures, slabs with longitudinal cracks, and concrete patches, supporting the overall PMIS distress rating computation.

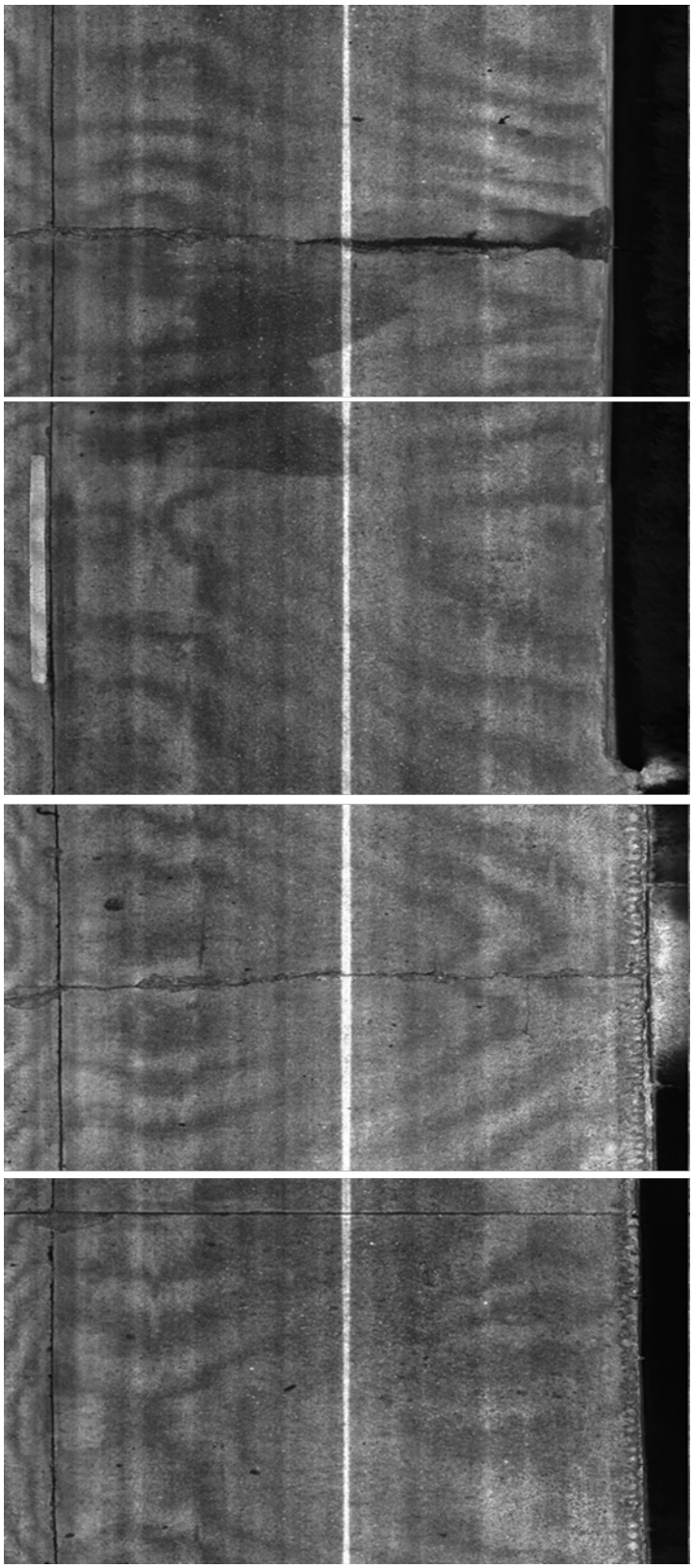


Figure 6.12 Images in original cut

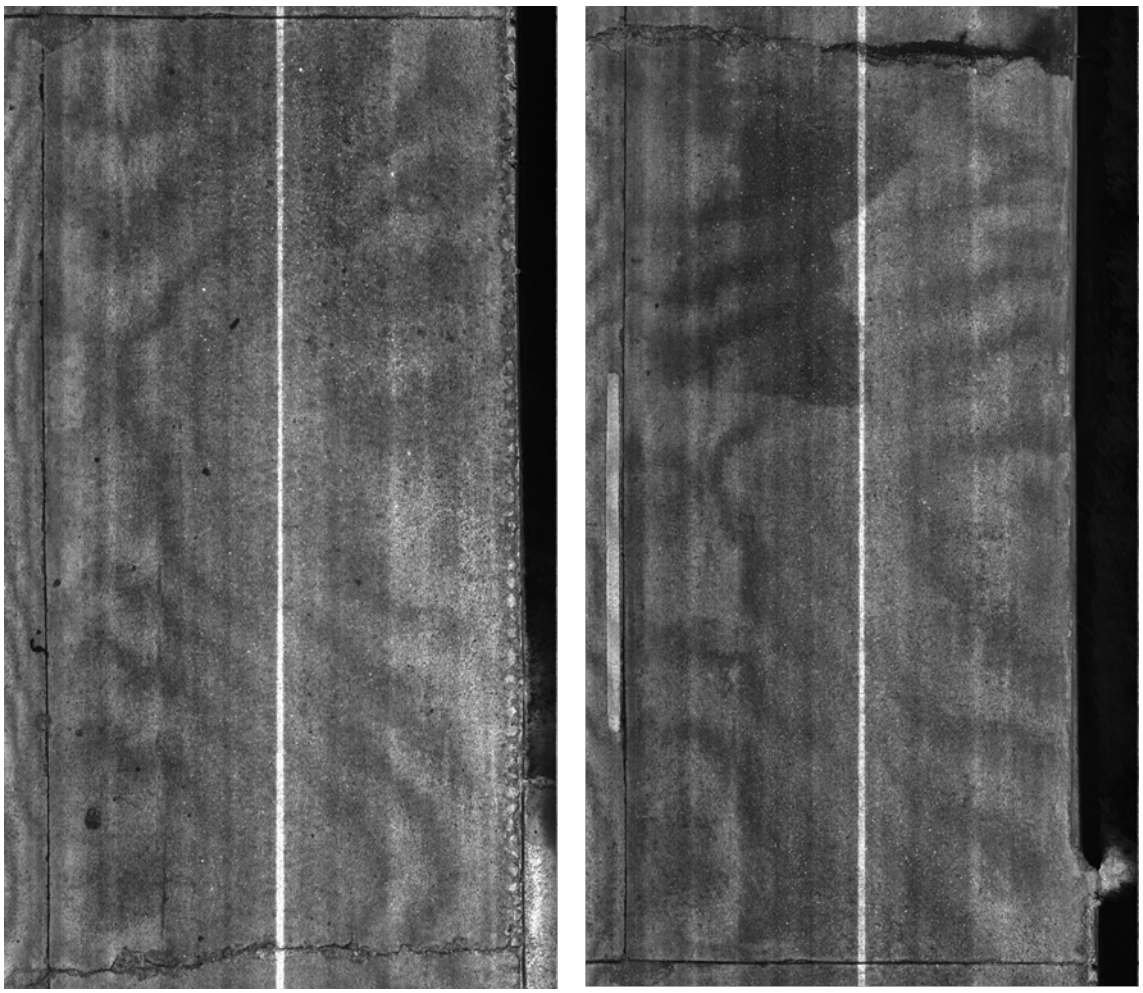


Figure 6.13 Images after combining and re-cut. Each image contains only a single slab defined by Joint, Failed Joint and Crack, or Transverse Crack

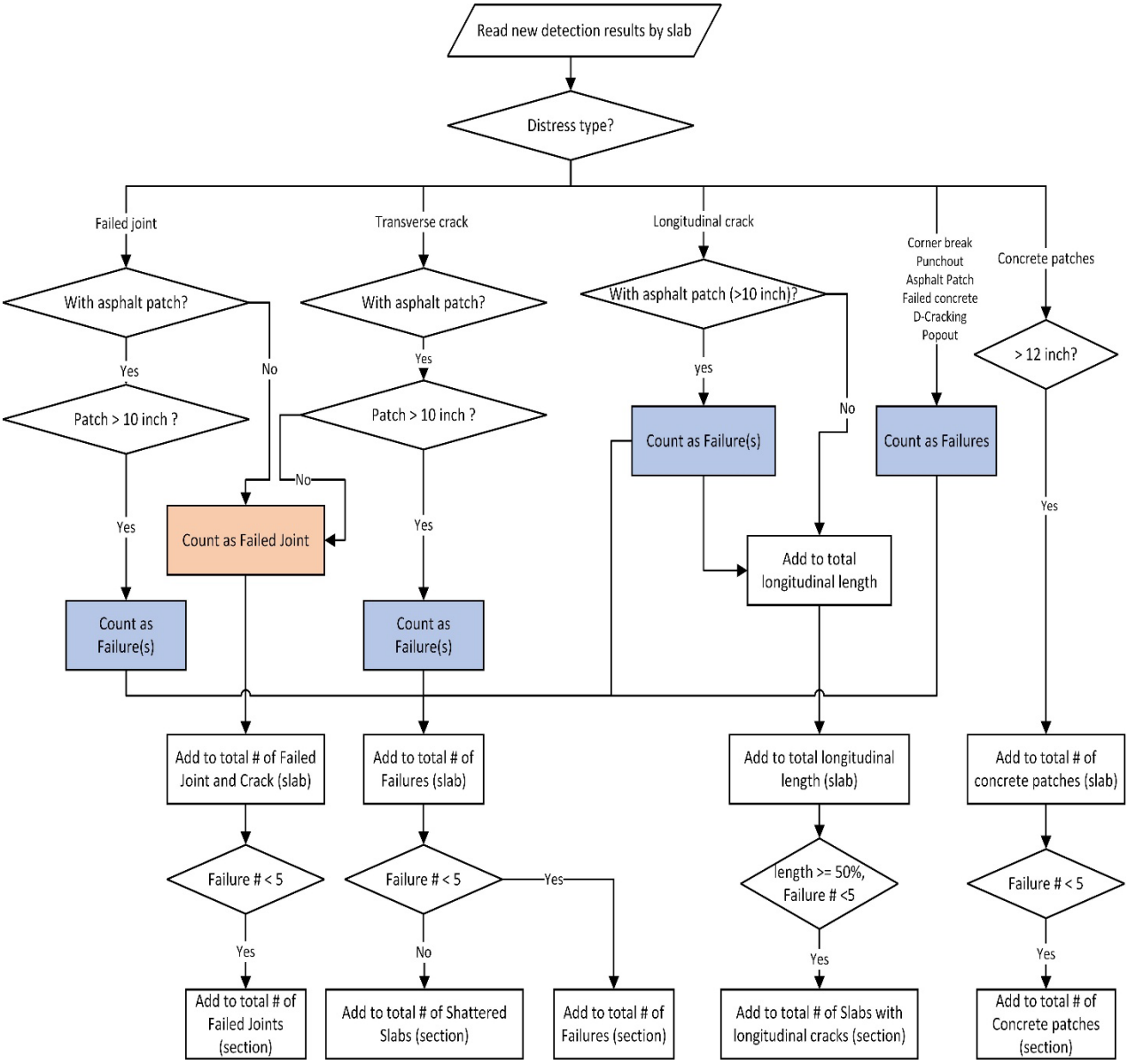


Figure 6.14 Flowchart for JCP distress score calculation: details about how to count each distress class

6.3.2.3 CRCP

The flowchart (Figure 6.15) outlines the process of converting detected CRCP distress data into a distress score. It begins with reading detection results and identifying the distress type. For transverse or sealed transverse cracks, the presence of an asphalt patch leads to counting it in the total number of spalled cracks. Additionally, spalled transverse cracks longer than 5 feet are also added to the spalled crack count. Other distress types such as punchouts, asphalt patches, and concrete patches are each tallied in their respective totals. Like JCP, punchouts, asphalt

patches, and concrete patches are all counted as one for every 10 feet. Once all relevant distress quantities are collected, these values are added to the corresponding distress class totals.

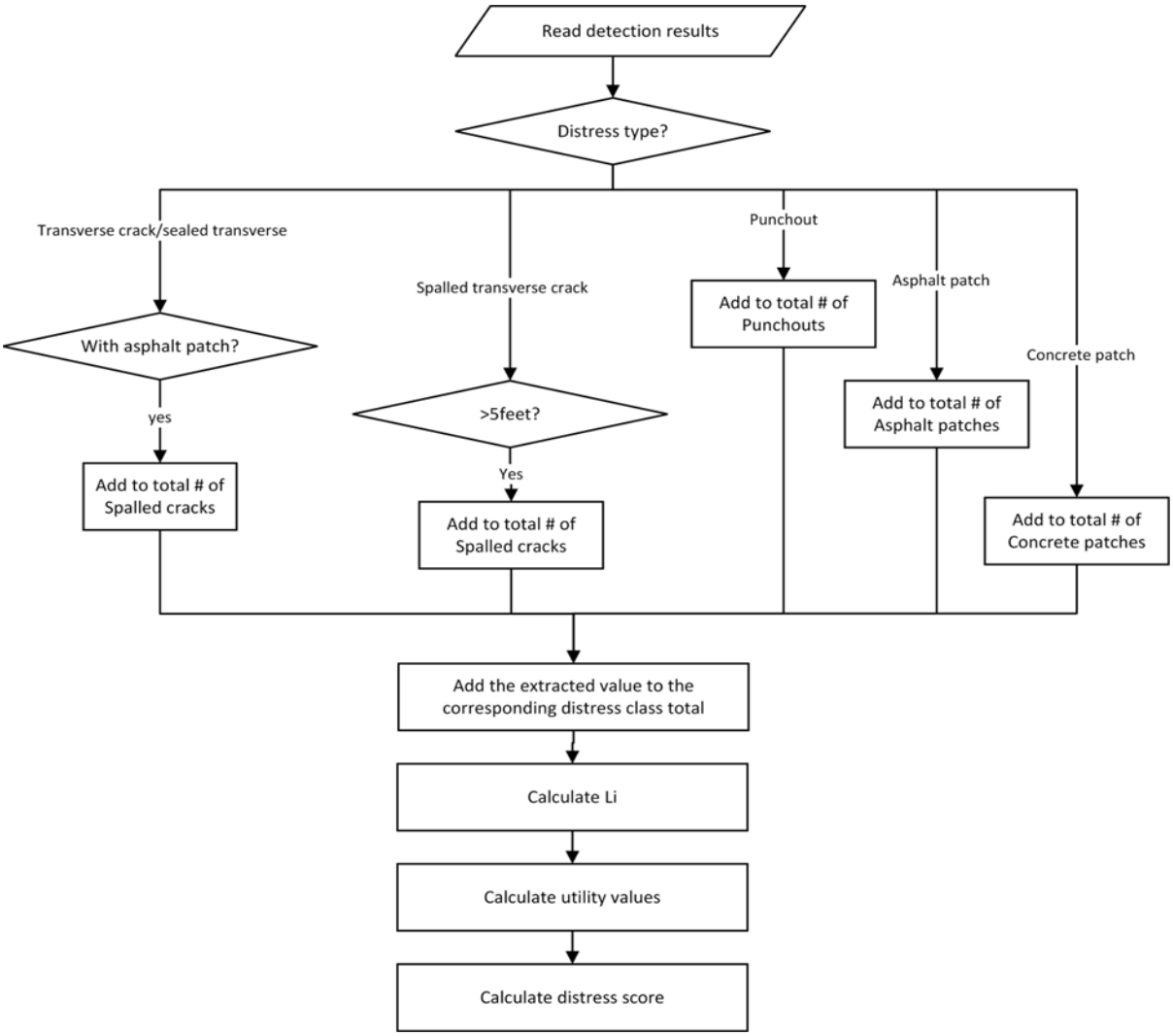


Figure 6.15 Flowchart for CRCP distress score calculation

6.3.3 Calculation of Normalized Distress

The normalized distress value (L_i) is a critical parameter used to standardize distress measurements across different pavement sections. Since pavement distress ratings depend on section length, normalization ensures consistency and comparability in the assessment process.

Different distress types are measured using distinct criteria; some are based on percentage coverage (e.g., block cracking, alligator cracking), while others are counted as discrete occurrences (e.g., failures, punchouts). The normalization process ensures that all distress values are converted into a standardized unit (e.g., occurrences per mile), facilitating accurate distress scoring.

Table 6.5 outlines how to determine the L_i values, which quantify the severity of different ACP distresses. For most distress types, such as shallow rutting, deep rutting, patching, block cracking, alligator cracking, longitudinal cracking, and transverse cracking, the L_i value is directly taken from the PMIS rating, which is typically expressed as a percentage of lane area or wheel-path length, or as a count per 100-foot station. For failures, L_i is calculated as the number of failures per mile. Raveling and flushing are optional distress types described qualitatively as none, low, medium, or high, and no L_i value is computed for them. This standardized approach ensures consistent conversion of PMIS distress ratings into numerical inputs for pavement condition evaluation.

Table 6.5 Distress Types and Computation of L_i Value (TxDOT, 2009)

ACP Distress Type	PMIS Rating	Computing L_i Value
Shallow Rutting	Percent of wheelpath length (0 to 100)	L_i = PMIS rating
Deep Rutting	Percent of wheelpath length (0 to 100)	L_i = PMIS rating
Patching	Percent of lane area (0 to 100)	L_i = PMIS rating
Failures	Total number (0 to 99)	$L_i = Rating/Length$
Block Cracking	Percent of lane area (0 to 100)	L_i = PMIS rating
Alligator Cracking	Percent of wheelpath length (0 to 100)	L_i = PMIS rating
Longitudinal Cracking	Length per 100-foot station (0 to 999)	L_i = PMIS rating
Transverse Cracking	Number per 100-foot station (0 to 99)	L_i = PMIS rating
Raveling	None, low, medium, or high	None (optional distress type)
Flushing	None, low, medium, or high	None (optional distress type)

Table 6.6 outlines how to calculate the L_i values for various types of distresses found in JCP. It includes six distress types: Failed Joints and Cracks, Failures, Shattered (Failed) Slabs, Slabs with Longitudinal Cracks, Concrete Patches, and Apparent Joint Spacing. For most types, the PMIS rating is the total number of occurrences (ranging from 0 to 999), and the L_i value is computed based on the number of distress instances adjusted for pavement length and joint spacing. Specifically, for Failed Joints and Cracks, Shattered Slabs, and Slabs with Longitudinal Cracks, the value is adjusted by AJS (Apparent Joint Spacing) in feet. For general Failures and Concrete Patches, L_i is calculated as rating per mile. The Apparent Joint Spacing itself is not

assigned an L_i value but is instead used as a normalization factor for other distress calculations, typically ranging from 15 to 75 feet. This table ensures consistent computation of distress severity for condition assessments and scoring in JCP pavement evaluation.

Table 6.6 JCP Distress Types and Computation of L_i Value (TxDOT, 2009)

JCP Distress Type	PMIS Rating	Computing L_i Value
Failed Joints and Cracks	Total number (0 to 999)	$L_i = 100 \times \left(\frac{Rating}{5280 \times Length/AJS}\right)$
Failures	Total number (0 to 999)	$L_i = \frac{Rating}{Length}$
Shattered (Failed) Slabs	Total number (0 to 999)	$L_i = 100 \times \left(\frac{Rating}{5280 \times Length/AJS}\right)$
Slabs with Longitudinal Cracks	Total number (0 to 999)	$L_i = 100 \times \left(\frac{Rating}{5280 \times Length/AJS}\right)$
Concrete Patches	Total number (0 to 999)	$L_i = \frac{Rating}{Length}$
Apparent Joint Spacing	Spacing, in feet (15 to 75)	None (used to normalize Failed Joints, Cracks, Shattered Slabs, and Slabs With Longitudinal Cracks)

Table 6.7 CRCP Distress Types and Computation of L_i Value (TxDOT, 2009)

CRCP Distress Type	PMIS Rating	Computing L_i Value
Spalled Cracks	Total number (0 to 999)	$L_i = Rating/Length$
Punchouts	Total number (0 to 999)	$L_i = Rating/Length$
Asphalt Patches	Total number (0 to 999)	$L_i = Rating/Length$
Concrete Patches	Total number (0 to 999)	$L_i = Rating/Length$
Average Crack Spacing	Spacing, in feet (1 to 75)	None (used as an indicator of the concrete slab's structural strength)

Table 6.7 outlines how to calculate the L_i values for different types of distresses found in CRCP. For Spalled Cracks, Punchouts, Asphalt Patches, and Concrete Patches, the PMIS rating is recorded as the total number of occurrences (ranging from 0 to 999), and the corresponding L_i value is computed using the formula, where "Length" refers to the pavement segment length in miles, and the result represents the rating per length. This approach standardizes the number of distresses per mile for condition assessment. The final row, Average Crack Spacing, is measured in feet (ranging from 1 to 75) and does not contribute to a computed L_i value. Instead, it serves as a qualitative indicator of the concrete slab's structural strength, offering insight into slab integrity rather than distress severity.

6.3.4 Calculation of utility values and distress score

6.3.4.1 Utility Values

The utility rating (U_i) represents the relative condition of pavement based on distress severity. It is computed using predefined parameters α, β, and ρ, ensuring a standardized assessment approach.

The utility function is defined as follows:

$$U_i = \begin{cases} 1, & \text{when } L_i = 0 \\ 1 - \alpha e^{-\left(\frac{\rho}{L_i}\right)^{\beta}}, & \text{when } L_i > 0 \end{cases} \quad (6.1)$$

where:

- **U_i**: Utility Value.
- **i**: PMIS distress type (e.g., alligator crack, transverse crack, or ride quality loss).
- **e**: Base of the natural logarithm ($e \approx 2.7182818$).
- α (Alpha): A horizontal asymptote factor that controls the maximum utility loss.
- β (Beta): A slope factor that determines how steeply utility declines as distress increases.
- ρ (Rho): A prolongation factor that controls how long the utility curve "lasts" before significant deterioration.
- **L_i**: The level of distress, which is normalized depending on the distress type.
- PMIS subdivides ACP, JCP, and CRCP further into ten "detailed" pavement types, assigned the code from 1 to 10. According to the distinct characteristics of each detailed pavement type and its surface distress, different sets of parameters α, β, and ρ are used (see Tables 6.8 to 6.11).

Table 6.8 Parameters for Distresses on ACP (Type 4,5,6,9 and 10)

Distress Type	Alpha (α)	Beta (β)	Rho (ρ)
Alligator Cracking	0.5300	1.0000	8.0100
Block Cracking	0.4900	1.0000	9.7800
Deep Rutting	0.6900	1.0000	16.2700
Failures	1.0000	1.0000	4.7000
Longitudinal Cracking	0.8700	1.0000	184.0000
Patching	0.2398	1.6978	12.0300
Severe Rutting	0.7661	1.0000	5.4604
Shallow Rutting	0.3100	1.0000	19.7200
Transverse Cracking	0.6900	1.0000	10.3900

Table 6.9 Parameters for Distresses on ACP (Type 7 and 8)

Distress Type	Alpha (α)	Beta (β)	Rho (ρ)
Alligator Cracking	0.4200	1.0000	18.7700
Block Cracking	0.3100	1.0000	13.7900
Deep Rutting	0.3200	1.0000	9.0400
Failures	1.0000	1.0000	4.7000
Longitudinal Cracking	0.3700	1.0000	136.9000
Patching	0.3200	1.0000	17.2800
Severe Rutting	0.6472	1.0000	5.0608
Shallow Rutting	0.2300	1.0000	17.5500
Transverse Cracking	0.4300	1.0000	9.5600

Table 6.10 Parameters for Distresses on JCP (Type 2 and 3)

Distress Type	Alpha (α)	Beta (β)	Rho (ρ)
Concrete Patches	1.1000	0.9900	64.0000
Failed Joints and Cracks	0.5298	1.0000	21.4000
Failures	1.4555	1.0000	22.1500
Shattered Slabs	1.1710	1.0000	16.3100
Slabs with Longitudinal Cracks	1.0058	1.0000	47.8000

Table 6.11 Parameters for Distresses on CRCP

Distress Type	Alpha (α)	Beta (β)	Rho (ρ)
Asphalt Patches	1.6000	0.2500	50.0000
Concrete Patches	0.9000	0.6600	13.6100
Punchouts	0.9849	1.0000	5.1400
Spalled Cracks	1.0000	0.6900	106.0000

6.3.4.2 Distress score

The distress score (DS) represents the overall pavement condition based on the severity of different distress types. It ranges from 1, indicating the worst condition, to 100, indicating the best condition. The score is derived from the computed utility values of individual distress types, ensuring an objective assessment of pavement quality.

The distress score is calculated using the product of the utility values for all considered distress types and is scaled to a range of 1 to 100. The formula used is:

$$DS = 100 \times \prod_{i=1}^{n} U_i \tag{6.2}$$

where:

- DS is the distress score,
- U_i is the utility value of distress type i,
- n is the total number of distress types included in the evaluation.

The multiplicative approach ensures that if any distress type has a very low utility value, the overall distress score is significantly reduced, accurately reflecting the pavement's deteriorated condition. The final scaling to 100 provides a standardized measure for comparison across different pavement sections. Special Case: When the computed DS is less than 0.0001, a minimum value of 0.0001 is assigned for calculation purposes to avoid numerical instability.

6.3.5 Calculation example

6.3.5.1 ACP

To illustrate the application of the distress score calculation methodology, this section presents a sample computation for an ACP section. The calculation involves post-processing detection results, computing utility values, and aggregating them to obtain the final distress score.

Step 1: Post-processing detection results. Table 6.12 shows the summary of the distress detection results of a sample ACP section. The total length of this section is 5021.73 feet. Following the rules described for ACP detection result conversion, the raw detection results are converted to PMIS ratings shown in Table 6.13.

Table 6.12 Distress detection results of a sample ACP section

Distress Type	Count #	Width (feet)	Length (feet)	Area (square feet)
Transverse crack	154	929.20	213.40	1395.41
Joint	0	0.00	0.00	0.00
Sealed transverse crack	22	248.81	40.76	473.86
Longitudinal crack	410	453.44	37,29.28	4,172.22
Lane longitudinal	10	9.07	142.98	129.07
Sealed longitudinal	88	106.10	972.85	1,172.58
Block crack	49	559.49	745.33	8,548.22
Alligator crack	14	35.37	218.74	554.61

The adjusted quantities in Table 6.13 are derived from the raw detection results presented in Table 6.12. For Alligator Cracking, Block Cracking, Failures, and Patching, the adjusted quantities remain essentially unchanged from the raw data. In contrast, the adjusted quantity for Longitudinal Cracking is calculated as the sum of the lengths of longitudinal, lane longitudinal, and sealed longitudinal cracks in Table 6.12. However, due to the exclusion of cracks located outside the travel lane, the final adjusted total (4845.12 feet) is slightly lower than the raw sum. For Transverse Cracking, the adjusted quantity is based on the total count of both transverse and sealed transverse cracks. Applying the standard rule that cracks shorter than 5 feet are ignored, those between 5 and 10 feet count as 0.5, and those longer than 10 feet count as 1, the adjusted total (72.5) is significantly smaller than the raw count (176), indicating that a substantial portion of the detected transverse cracks are short in length and fall below the threshold for full inclusion. The adjusted quantities are subsequently translated into PMIS ratings based on the guidelines provided in the Rater's Manual.

Table 6.13 Summary of post-processing the detection results

Distress Type	Unit	Adjusted quantity	PMIS Rating
Alligator Cracking	Total feet	218.74	0.0218
Block Cracking	Total feet	745.33	0.1484
Failures	Total count	0	0
Longitudinal Cracking	Total feet	4,845.12	96.48
Patching	Total count	0	0
Transverse Cracking	Total count	72.5	1.4437

Table 6.14 Summary of L_i and Utility values calculation

Distress Type	PMIS Rating	L_i	Utility Value
Alligator Cracking	0.02	0.02	1.00
Block Cracking	0.145	0.15	1.00
Deep Rutting	0.00	0.00	1.00
Failures	0.00	0.00	1.00
Longitudinal Cracking	96.48	96.48	0.87
Patching	0.00	0.00	1.00
Severe Rutting	0.00	0.00	1.00
Shallow Rutting	0.00	0.00	1.00
Transverse Cracking	1.44	1.44	1.00

Step 2: Computing utility values. Table 6.14 presents the PMIS ratings, computed L_i values, and corresponding utility values for various distress types observed in the sample ACP section. The L_i values are calculated based on the methodology described in Table 6.5, which outlines

how to derive severity levels from PMIS ratings. The utility values are then computed using the L_i values as inputs to Equation 1, with parameter settings specified in Table 6.8. Currently, the PMIS ratings for all rutting-related distresses (deep, severe, and shallow rutting) are set to 0 by default, as these types of distresses are not captured by the current detection model. Among all the distress types, longitudinal cracking is the most dominant in this ACP section, reflected by its significantly high PMIS rating and L_i value, resulting in the lowest utility value in the table.

Step 3: Computing distress score. The calculation of the distress score of the sample ACP section is based on Equation 2. More specifically, Equation 3 is applied. The distress score of the sample section is 87, falling under the category of "Good".

$$DS_{\text{ACP}} = 100 \times U_{\text{ShaRut}} \times U_{\text{DRut}} \times U_{\text{Patch}} \times U_{\text{Fail}} \times U_{\text{Allig}} \times U_{\text{Blk}} \times U_{\text{Trn}} \times U_{\text{Lng}} \times U_{\text{SevRut}} \quad (6.3)$$

6.3.5.2 Distress score

To demonstrate the methodology for distress score calculation in JCP, this section presents a step-by-step computational example. The distress score is determined by evaluating key distress types such as failed joints and cracks, failures, slabs with longitudinal cracks, and concrete patches. The calculation involves post-processing detection results, computing utility values, and aggregating them to obtain the final distress score.

Step 1: Post-processing detection results. Table 6.15 shows the summary of the distress detection results of a sample JCP section. The total length of this section is 451.5 feet. Following the rules described for JCP detection result conversion, the raw detection results are converted to PMIS ratings shown in Table 6.16.

The adjusted quantities in Table 6.16 are derived from the raw detection results presented in Table 6.15. For Concrete Patches, the adjusted quantity remains zero, consistent with the raw data where no instances of concrete patch were detected. For Failed Joints and Cracks, the adjusted quantity is 5, directly corresponding to the 5 failed joints identified in the detection results. No transverse cracks overlapping with asphalt patches were detected that would otherwise contribute to this category. For Failures, one instance is counted in the adjusted results, which can be attributed to the single corner break detected in Table 6.15. Although one asphalt patch was also detected, its area is minimal and thus excluded from failure classification. No other distresses met the criteria to be counted as failures. The adjusted quantity for Shattered Slabs is zero, indicating that no slabs exhibited ei- ther five or more failures or failure coverage exceeding 50% of the slab area—both of which are required to classify a slab as shattered. For Slabs with Longitudinal Cracks, the adjusted quantity is one, even though two longitudinal cracks are reported in the raw data. This suggests that either both cracks are located within a single slab or that one of them does not exceed 50% of the slab's length, thereby not qualifying as a separate instance. The adjusted quantities are subsequently translated into PMIS ratings according to the procedures outlined in the Rater's Manual.

Table 6.15 Distress detection results of a sample JCP section

Distress Type	Count #	Width (feet)	Length (feet)	Area (square feet)
Failed joint	5	62.69	4.20	52.90
Corner break	1	1.71	0.35	0.59
Punchout	0	0.00	0.00	0.00
Asphalt patch	0	0.00	0.00	0.00
Failed concrete patch	0	0.00	0.00	0.00
D-cracking	0	0.00	0.00	0.00
Popout	0	0.00	0.00	0.00
Longitudinal crack	2	2.58	9.41	11.48
Sealed longitudinal	0	0.00	0.00	0.00
Concrete patch	0	0.00	0.00	0.00
Transverse crack	15	144.85	12.99	132.86
Joint crack	13	157.40	9.86	114.47
Sealed transverse crack	0	0.00	0.00	0.00
Slab edge	55	42.11	205.32	159.63

Table 6.16 Summary of post-processing the detection results

Distress Type	Unit	Adjusted quantity	PMIS Rating
Concrete Patches	Total count	0	0
Failed Joints and Cracks	Total count	5	5
Failures	Total count	1	1
Shattered Slabs	Total count	0	0
Slabs with Longitudinal Cracks	Total count	1	1

Step 2: Computing utility values. Table 6.17 presents the PMIS ratings, computed L_i values, and corresponding utility values for various distress types observed in the sample JCP section. The L_i values are calculated based on the methodology described in Table 6.6, which outlines how to derive severity levels from PMIS ratings. The utility values are then computed using the L_i values as inputs to Equation 1, with parameter settings specified in Table 6.10. Among the listed distresses, Failures and Failed Joints and Cracks are identified as the two most critical types based on their relatively low utility values—0.78 and 0.80, respectively—indicating greater severity and impact on pavement conditions. Notably, even a single Failure instance can lead to a significant reduction in utility value, highlighting its impact on the overall distress score. In

contrast, a distress like Slabs with Longitudinal Cracks, which shares the same PMIS rating as Failures, maintains a high utility value of 1.00.

Table 6.17 Summary of L_i and Utility values calculation

Distress Type	PMIS Rating	L_i	Utility Value
Concrete Patches	0	0.00	0.00
Failed Joints and Cracks	5	21.74	0.80
Failures	1	11.69	0.78
Shattered Slabs	0	0	1.00
Slabs with Longitudinal Cracks	1	4.35	1.00

Step 3: Computing distress score. The calculation of the distress score of the sample JCP section is based on Equation 2. More specifically, Equation 4 is applied. The distress score of the sample section is 63, falling under the category of "Poor".

$$DS_{\text{JCP}} = 100 \times U_{\text{FailJnt}} \times U_{\text{Fail}} \times U_{\text{SSlab}} \times U_{\text{Lng}} \times U_{\text{PCPatch}} \tag{6.4}$$

6.3.5.3 CRCP

This section presents a step-by-step computational example for CRCP distress score evaluation. The calculation involves post-processing detection results, computing utility values, and aggregating them to obtain the final distress score.

Table 6.18 Distress detection results of a sample CRCP section

Distress Type	Count #	Width (feet)	Length (feet)	Area (square feet)
Longitudinal crack	20	18.78	223.98	217.59
Sealed longitudinal crack	18	20.42	209.92	243.63
Punchout	1	10.94	4.25	46.51
Asphalt patch	0	0.00	0.00	0.00
Concrete patch	4	40.45	28.43	319.98
Transverse crack	203	1,396.29	223.86	1,660.16
Sealed transverse crack	196	1,653.89	215.91	,1970.91
Spalled longitudinal crack	0	0.00	0.00	0.00
Spalled transverse crack	9	95.20	15.32	161.19

Step 1: Post-processing detection results. Table 6.18 shows the summary of the distress detection results of a sample ACP section. The total length of this section is 917.97 feet. Following the rules described for CRCP detection result conversion, the raw detection results are converted to PMIS ratings shown in Table 6.19.

The adjusted quantities in Table 6.19 are derived from the raw detection results presented in Table 6.18. For Asphalt Patches, Concrete Patches, and Punchouts, the adjusted counts are generally carried over directly from Table 6.18 with minimal modification. However, a standard adjustment rule is applied: for instances longer than 10 feet, one count is added for every 10 feet of length. This explains why the count of Concrete Patches increases from 4 in Table 6.18 to 5 in Table 6.19—one of the detected patches falls within the 10–20-foot range and thus contributes two counts.

Table 6.19 Summary of post-processing the detection results

Distress Type	Unit	Adjusted quantity	PMIS Rating
Asphalt Patches	Total count	0	0
Concrete Patches	Total count	5	5
Punchouts	Total count	1	1
Spalled Cracks	Total count	9	9

In the case of Spalled Cracks, only Spalled Transverse Cracks are included in the adjusted quantity. Spalled Longitudinal Cracks are excluded from consideration, which is why the count of 9 in Table 6.18 directly becomes the adjusted quantity in Table 6.19. Other distresses such as nonspalled Longitudinal Cracks and Transverse Cracks are generally not included for distress score calculation unless they meet specific qualification criteria, which do not apply in this instance.

Step 2: Computing utility values. Table 6.20 presents the PMIS ratings, computed $L_i$values, and corresponding utility values for various distress types observed in the sample CRCP section. The L_i values are calculated based on the methodology described in Table 6.7, which outlines how to derive severity levels from PMIS ratings. The utility values are then computed using the L_i values as inputs to Equation 1, with parameter settings specified in Table 6.11. Among the listed distresses, Concrete Patches and Punchouts stand out as the two most critical based on their relatively low utility values—0.51 and 0.60, respectively—indicating a substantial negative impact on overall pavement condition. Spalled Cracks also contribute notably, with a utility value of 0.81, suggesting moderate severity. Notably, despite being represented by only a single count, Punchouts significantly reduce the utility value, demonstrating their high weighting and critical influence on the distress score.

Table 6.20 Summary of L_i and Utility values calculation

Distress Type	PMIS Rating	L_i	Utility Value
Asphat Patches	0	0.00	1.00
Concrete Patches	5	28.76	0.51
Punchouts	1	5.75	0.60
Spalled Cracks	9	51.77	0.81

Step 3: Computing distress score. The calculation of the distress score of the sample CRCP section is based on Equation 2. More specifically, Equation 5 is applied. The distress score of the sample section is 25, falling under the category of "Very Poor".

$$DS_{CRCP} = 100 \times U_{Spall} \times U_{Punch} \times U_{ACPatch} \times U_{PCPatch} \quad (6.5)$$

6.4 Summary

This chapter outlines a comprehensive workflow for transforming AI-based pavement distress detection outputs into standardized PMIS distress scores. The purpose is to enhance the accuracy, efficiency, and consistency of pavement condition assessments by aligning machine learning detection results with TxDOT's established scoring framework.

Post-processing rules are tailored for three pavement types, such as ACP, JCP, and CRCP, to convert raw detection outputs into distress categories recognized by the PMIS. Each pavement type has specific criteria to filter, consolidate, and qualify detected distresses, ensuring compatibility with the Rater's Manual.

Following post-processing, distress data are converted into PMIS ratings and normalized to generate L_i values. These values are then used to compute utility values through an exponential decay function. The final distress score, a product of all Utility values, reflects the overall pavement condition on a scale from 1 (worst) to 100 (best). Through worked examples for ACP, JCP, and CRCP sections, this study demonstrates how distress scores are computed in practice. These examples validate the robustness and adaptability of the framework in real-world scenarios.

Chapter 7 Pilot Study

The primary purpose of this chapter is to document the procedure and findings of a pilot study designed to evaluate the generalization PERFORMANCE of the AI-based pavement condition assessment model. Specifically, this study assesses how well the distress detection models, developed and documented in Chapter 5, apply to pavement sections located in Brazoria County that were not included in the model's training dataset. Also, new AI/ML models are developed.

7.1 Objectives

The objective of this pilot study is to evaluate the performance and generalization capability of an AI-based distress detection model when applied to pavement sections from a county not represented in the training dataset. The evaluation is conducted at four distinct levels:

- Training dataset – to assess the model's ability to fit the data it was trained on and confirm that it effectively learned the underlying distress patterns.
- Validation dataset – to evaluate the model's ability to generalize during development by measuring its performance on data withheld during training.
- Real-world dataset (pilot counties) – to test the model's robustness and real-world generalization on entirely new pavement sections from a different geographic region.
- Model development and improvement – to fully utilize the available datasets and latest developed AI methods to further improve the performance of AI models.

This multi-level evaluation aims to comprehensively assess the model's capabilities of learning, validation, and generalization. Insights gained from this pilot study will help identify potential limitations, validate model robustness, and recommend future strategies for statewide implementation of AI-based pavement condition assessment.

This chapter details the preparation of the test datasets, the post-processing of detection results, and the evaluation of the current AI model (YOLOv5 parallel) performance in terms of consistency and accuracy. The pilot study represents an initial step toward validating the model's robustness and assessing its potential for broader implementation at the state level.

7.2 Performance of Current Model over Image Library

This chapter presents a detailed evaluation of the performance of AI models across different pavement types using the image library (labeled data). The focus is on three pavement types: Asphalt Concrete Pavement (ACP), Jointed Concrete Pavement (JCP), and Continuously Reinforced Concrete Pavement (CRCP). For each pavement type, performance is analyzed separately over the training and validation datasets.

The goal of this analysis is to assess how well the model learns distress characteristics specific to each pavement type and to identify potential weaknesses or biases in detection. By examining precision, recall, and other performance metrics at the dataset level, this section provides insight into the model's capability to generalize within the trained pavement categories:

(1) Precision (P), recall (R) and F1 score

$$Precision = \frac{TP}{TP+FP} \tag{7.1}$$

$$Recall = \frac{TP}{TP+FN} \tag{7.2}$$

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{7.3}$$

where *TP* and *FP*, TN and *FN* represent true and false positives, true and false negatives, respectively.

(2) mAP50 and mAP50-95

Average precision (AP) is a metric for how accurate of predicted bounding box over the ground-truth one, and can be calculated as the area under the precision-recall curve:

$$AP = \int_0^1 P(r)dr \tag{7.4}$$

where $P(r)$ is the precision at recall level r.

Mean average precision (mAP) represents the overlaid prediction and ground truth. mAP50 is calculated when the AP is at an Intersection over Union (IoU) threshold of 0.5 for a class. mAP50-95is metric to average the AP by calculating it at IoU thresholds from 50% to 95% with a step size of 5%.

7.2.1 Test on training and validation datasets of Asphalt Concrete Pavements

Table 7.1 shows the detection performance of the model on the ACP training dataset, highlighting its strong learning ability with an overall mAP50 of 0.958. Performance across individual distress types is also consistently high, with the lowest mAP50 observed for Joint distress at 0.907 and the highest for Block and Alligator cracking, both reaching 0.983. These results indicate that the model has effectively learned the characteristics of the training data and can accurately detect a wide range of distress types. However, further analysis on the validation dataset is necessary to ensure that high training performance does not reflect overfitting.

Table 7.1 Detection performance of the model on the ACP training dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	4,853	8,842	0.891	0.919	0.958	0.606
Transverse	4,853	1,147	0.885	0.863	0.936	0.502
Joint	4,853	316	0.853	0.886	0.907	0.387
Sealed_transverse	4,853	1,475	0.927	0.882	0.959	0.538
Longitudinal	4,853	1,259	0.874	0.923	0.963	0.611
Lane_longitudinal	4,853	1,005	0.855	0.964	0.961	0.606
Sealed_longitudinal	4,853	2,771	0.912	0.946	0.975	0.622
Block	4,853	202	0.885	0.95	0.983	0.839
Alligator	4,853	568	0.912	0.981	0.983	0.720
Pothole	4,853	99	0.918	0.879	0.952	0.634

Table 7.2 indicates the detection performance of the model on the ACP validation dataset, providing insight into its effectiveness on unseen but similar data. The overall mAP50 is 0.812 and mAP50-95 is 0.414, indicating reasonable detection accuracy across the dataset. Performance varies among distress types, with Joint distress achieving the highest mAP50 of 0.933, reflecting strong detection capability for that class. In contrast, Pothole detection shows the lowest performance, with an mAP50 of 0.480 and mAP50-95 of 0.184, likely due to its limited sample size and higher variability. Other classes such as the Block cand Sealed longitudinal cracking also demonstrate relatively high detection accuracy, with mAP50 values of 0.882 and 0.859, respectively. These results suggest that while the model performs well for several types of distress, performance remains uneven and may benefit from further refinement or class-specific improvements.

Table 7.2 Detection performance of the model on the ACP validation dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	1,039	2043	0.838	0.781	0.812	0.414
Transverse	1,039	222	0.828	0.694	0.824	0.329
Joint	1,039	84	0.885	0.827	0.933	0.369
Sealed transverse	1,039	327	0.897	0.749	0.892	0.415
Longitudinal	1,039	306	0.753	0.686	0.788	0.399
Lane longitudinal	1,039	251	0.834	0.773	0.850	0.459
Sealed longitudinal	1,039	629	0.886	0.836	0.899	0.461
Block	1,039	50	0.830	0.720	0.882	0.629
Alligator	1,039	143	0.834	0.703	0.820	0.484
Pothole	1,039	31	0.799	0.323	0.480	0.184

Figure 7.1 compares the mAP50 scores for each distress type on the ACP training and validation datasets, providing a visual assessment of the model's ability to generalize from learned data to

unseen examples. Overall, the model achieves consistently high mAP50 scores across all distress types on the training set, typically above 0.90. However, a performance drop is observed across nearly all categories when evaluated on the validation, indicating some degree of overfitting.

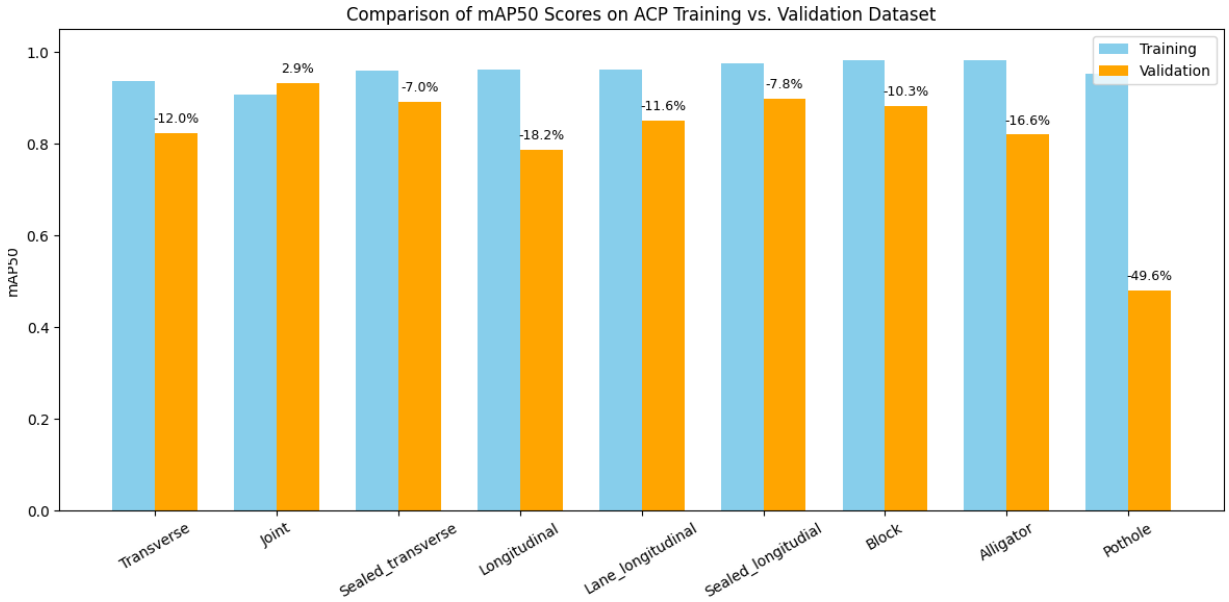


Figure 7.1 Comparison of mAP50 scores on ACP training and validation datasets

The most significant drop is observed for pothole detection, which declines by 49.6%, likely due to the small number of training instances, leading to substantial overfitting. Interestingly, the Joint distress shows a 2.9% increase, possibly due to a favorable distress distribution in the validation dataset. The Longitudinal and Alligator cracking exhibit drops greater than 15%, indicating moderate challenges in generalization for these classes. The remaining distress types, including the Sealed transverse, Lane longitudinal, and Block cracking, show relatively strong and stable performance, with drops under 13%, suggesting more consistent model behavior across training and validation data.

It is worth mentioning that: the training and validation datasets used in this study were randomly split from the same image library of ACP pavement sections. As a result, while the validation dataset was not seen during training, both datasets may share similar visual characteristics, environmental conditions, or distress patterns. This split allows for evaluating the model's ability to generalize to new images with familiar features, but it may not fully reflect performance on entirely unseen or geographically diverse datasets.

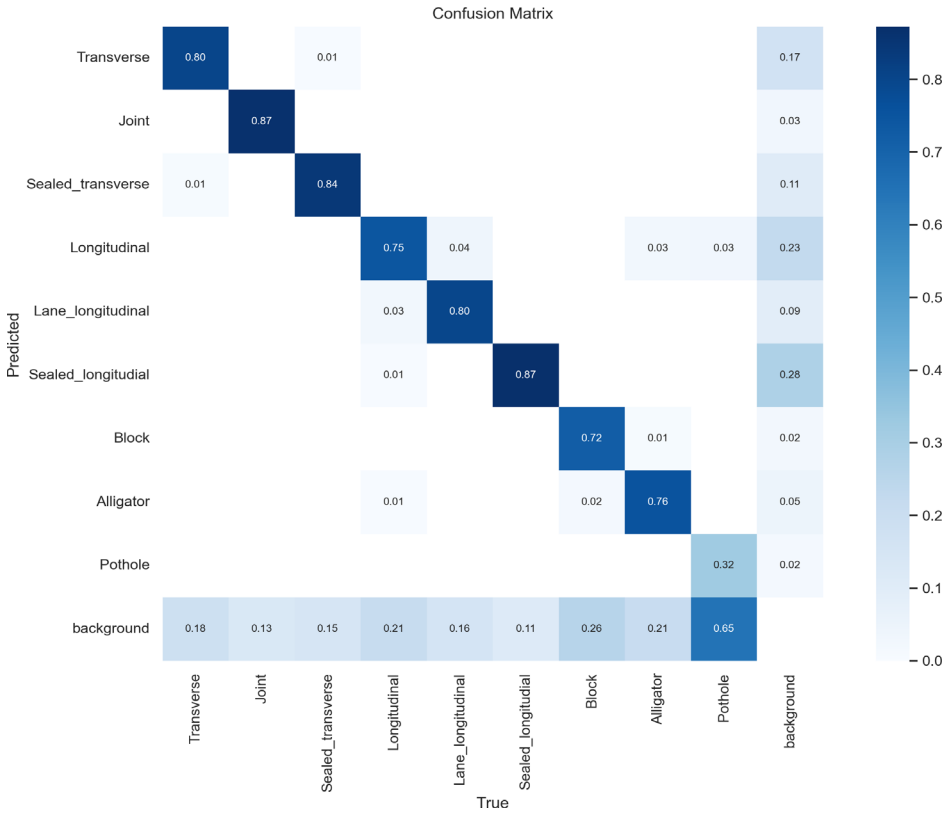


Figure 7.2 ACP confusion matrix

The confusion matrix (see Figure 7.2) illustrates the model's classification performance across various ACP distress types on the validation dataset. Most classes show strong correct prediction rates along the diagonal, particularly Sealed longitudinal (0.87), Joint (0.87), and Sealed transverse (0.84), indicating effective identification of these distresses. However, the Pothole exhibits a lower correct classification rate of 0.65 and a high confusion rate with the background class (0.32), suggesting difficulty in distinguishing potholes from surrounding pavement, possibly caused by limited training samples and visual ambiguity. Other classes showing notable confusion with background include Longitudinal (0.23), Alligator (0.21), and Block (0.26), indicating that the main portion of missed detections are due to the model's limited ability to accurately extract distress features from visually complex or noisy backgrounds. This highlights that the primary challenge lies in distinguishing distresses from background textures.

In addition to background confusion, Figure 7.2 also reveals specific pairs or groups of dis- tress types that are prone to misclassification due to visual similarity. One such group includes transverse and sealed transverse cracks, where a small but consistent portion of Sealed transverse instances are misclassified as the Transverse crack, likely due to partial sealing or unclear sealing boundaries. Another common confusion occurs within the longitudinal crack categories: Longitudinal cracks are occasionally misclassified as Lane longitudinal, Sealed longitudinal, or even Alligator cracking. This may be attributed to overlapping spatial patterns or morphological similarities in crack structure. Similarly, both Block and Alligator cracking show a sign of mutual misclassification, reflecting the difficulty in distinguishing between these two interconnected and networked crack types. These observations suggest that beyond background

separation, improving the model's ability to capture finer intra-class distinctions, possibly through enhanced labeling consistency or feature-level attention, could further reduce confusion between visually similar distress types.

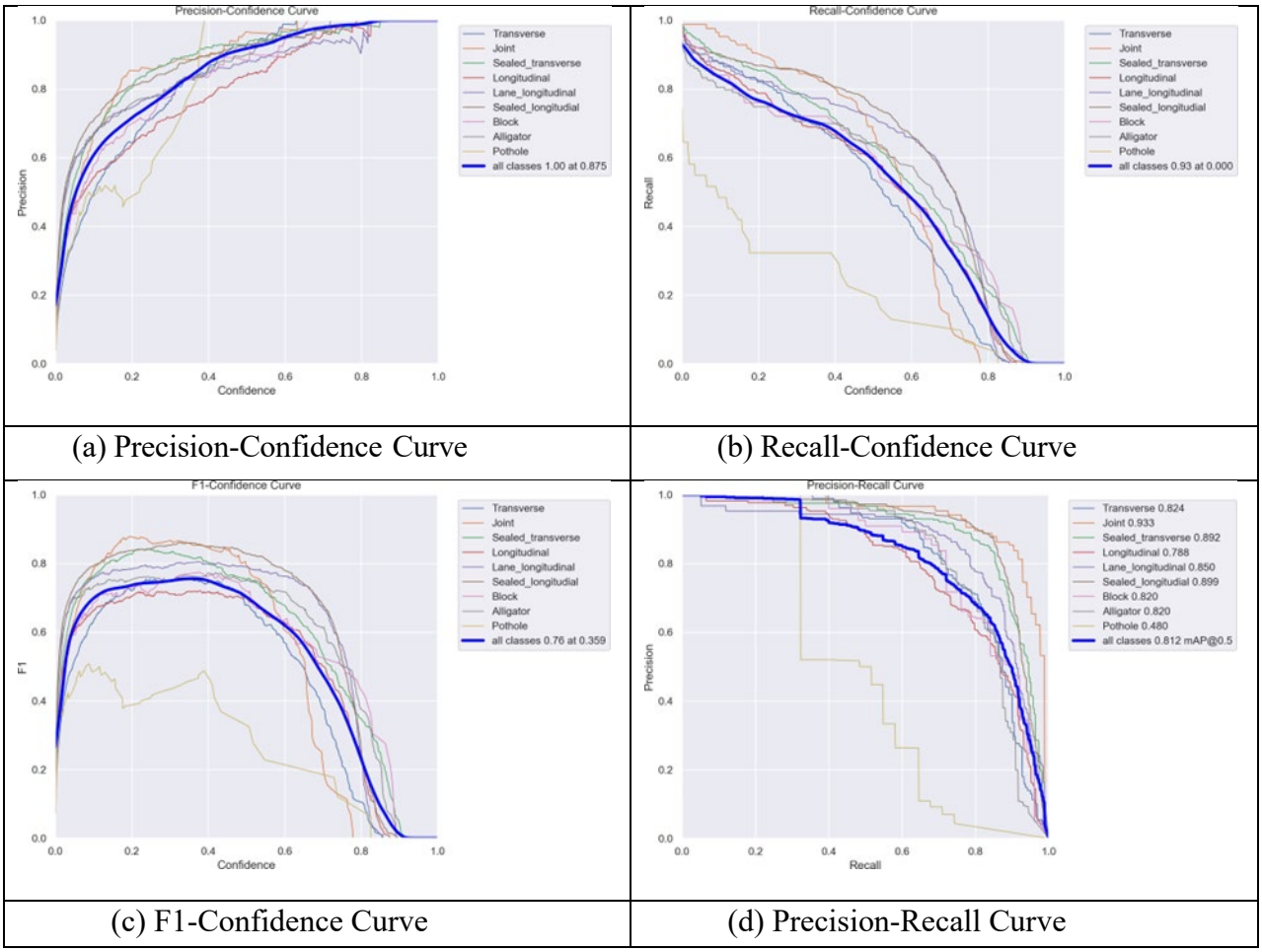
Figure 7.3 illustrates the model's performance across various ACP distress types, with a focus on how confidence thresholds impact prediction quality.

- Precision–Confidence Curve (a): This plot shows that model precision improves steadily as the confidence threshold increases. For most distress types, especially Sealed longitudinal, Sealed transverse, and Joint, precision rises sharply and approaches 1.00 at higher confidence levels, indicating highly reliable predictions when the model is confident. In contrast, Pothole displays erratic and consistently lower precision across the entire confidence range, suggesting unreliable detection.

- Recall–Confidence Curve (b): This plot shows how recall changes with increasing confidence thresholds. Most distress types exhibit a similar and expected trend: recall gradually decreases as the confidence threshold increases, reflecting the typical trade-off between sensitivity and prediction certainty. Sealed longitudinal and Lane longitudinal maintain relatively high recall over a wide range of thresholds, indicating the model's consistent ability to detect these distresses even at higher confidence levels. In contrast, The Pothole distress displays a distinct pattern with a sharp and early decline in recall, dropping below 0.5 at very low confidence thresholds. This deviation highlights the model's poor sensitivity to potholes.

- F1–Confidence Curve (c): The F1 curve balances precision and recall, peaking at an optimal confidence threshold of 0.359, where the overall F1 score reaches 0.76. Most classes display smooth, peaked curves with strong F1 performance, especially the Joint, Sealed longitudinal, and Sealed transverse. In contrast, Pothole again stands out with a low and unstable F1 curve, reinforcing the model's limited effectiveness for this class.

- Precision–Recall Curve (d): This plot provides a comprehensive view of the model's trade- off between precision and recall. Joint achieves the highest area under the curve with a mAP@50 of 0.933, followed by Sealed longitudinal (0.899) and Sealed transverse (0.892). These curves are well-formed and maintain high precision across a wide recall range, indicating robust detection. Pothole, however, shows a steep decline in both precision and recall, yielding the lowest mAP50 of 0.480, confirming its detection remains the model's most significant weakness.



(a) Precision-Confidence Curve

(b) Recall-Confidence Curve

(c) F1-Confidence Curve

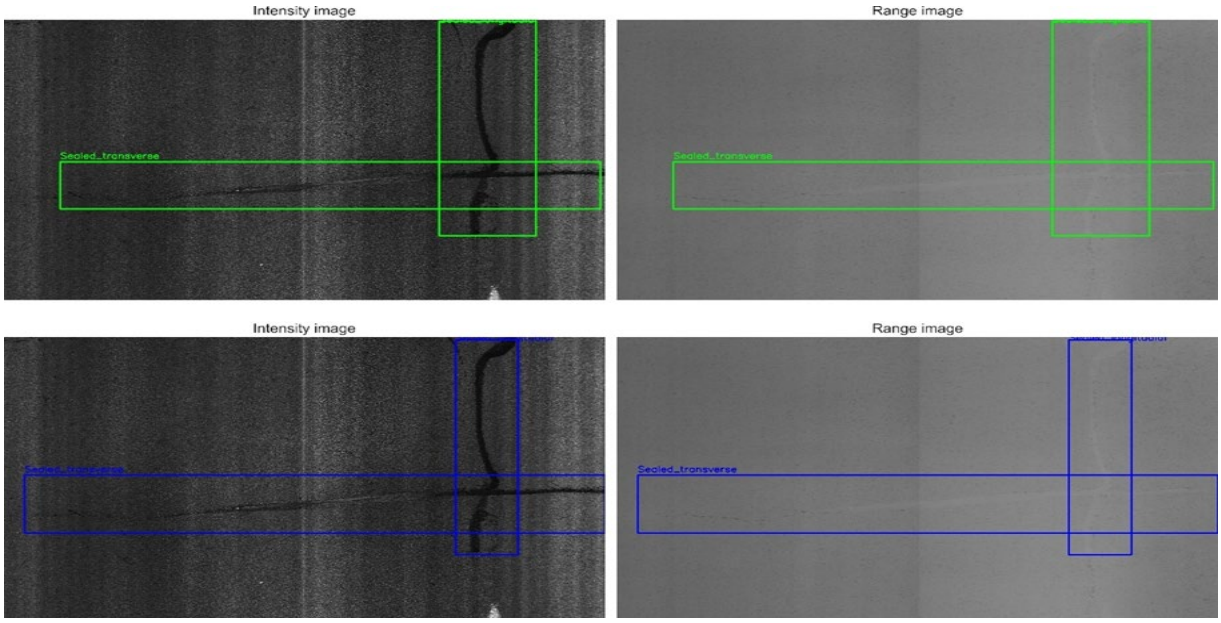
(d) Precision-Recall Curve

Figure 7.3 Model evaluation plots for the detection system over the ACP dataset

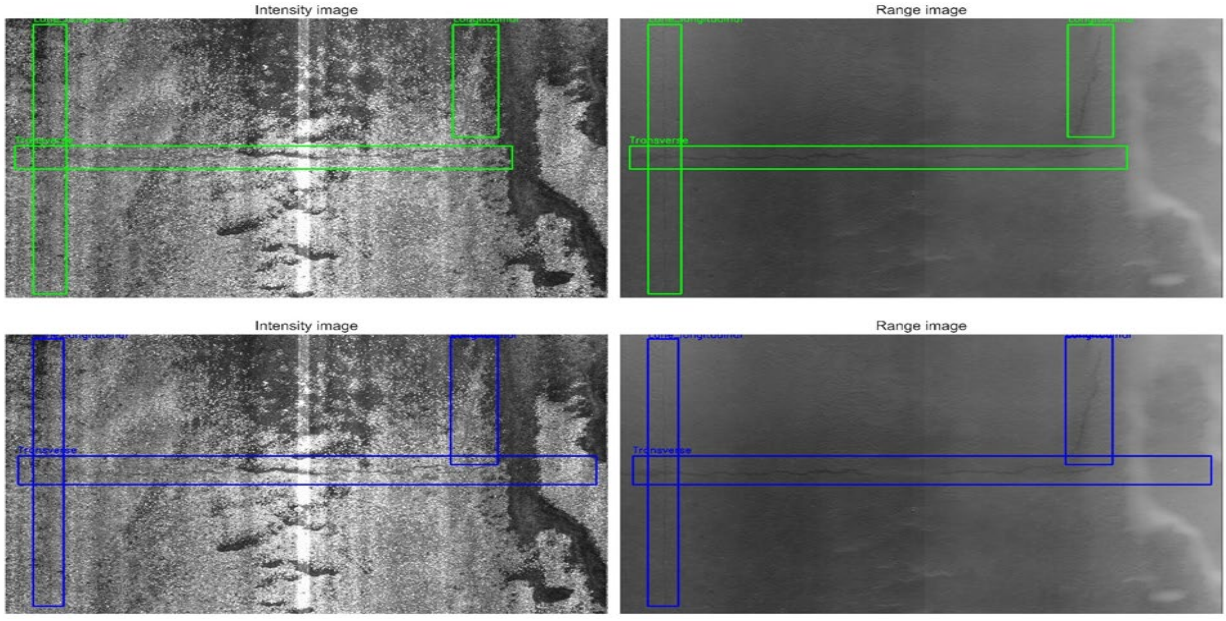
Figure 7.4 presents examples of correct detection results from the ACP model, with green boxes representing ground truth annotations and blue boxes representing model predictions. In Figure 7.4 (Subfigure 1), the model correctly identifies both Sealed transverse and Sealed longitudinal cracks in a relatively clean background, demonstrating strong alignment between predictions and annotated distress locations. In Figure 7.4 (Subfigure 2), the model accurately detects unsealed transverse and longitudinal cracks in a much more visually complex environment, where surface noise, markings, and texture variation are prominent. The accurate detection under such challenging conditions highlights the model's robustness and its ability to generalize distress patterns even in the presence of significant background clutter.

Figure 7.5 illustrates incorrect prediction examples made by the ACP model, with green boxes representing ground truth and blue boxes representing model predictions. In Figure 7.5 (Subfigure 1), the ground truth includes two longitudinal cracks and one transverse crack, all with moderate contrast in both the intensity and range images. The model correctly detects the transverse crack and one longitudinal crack but fails to detect the second longitudinal crack, which is spatially separated from other predictions. This indicates a true missed detection, likely due to weak feature activation in that area or underrepresentation of similar cases in the training data. In Figure 7.5 (Subfigure 2), the ground truth labels the distress as a block crack, while the

model detects a transverse crack in the same region. Although technically a misclassification, this prediction is still partially valid, as block cracking is often composed of intersecting longitudinal and transverse cracks. In this case, the model appears to have responded to local transverse features within the block-cracked region and output a transverse label instead of identifying the full pattern as block cracking. Further experiments are needed to decide if this is a class underrepresented issue or model deficiency issue.

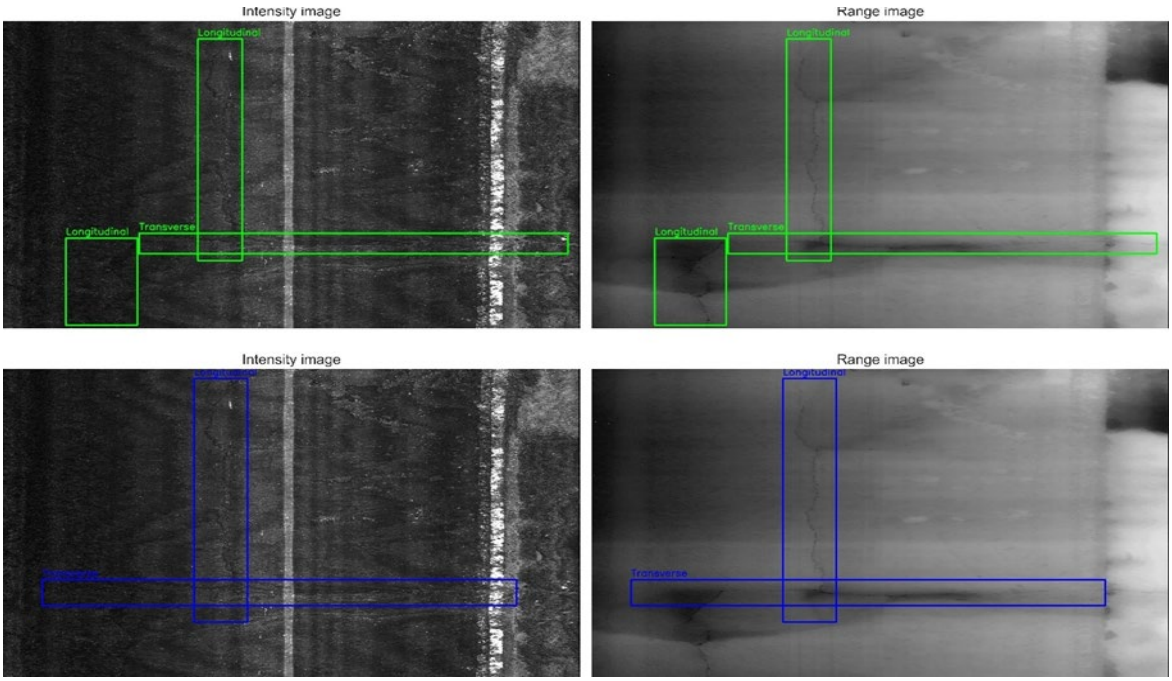


(1) Detection of sealed cracks

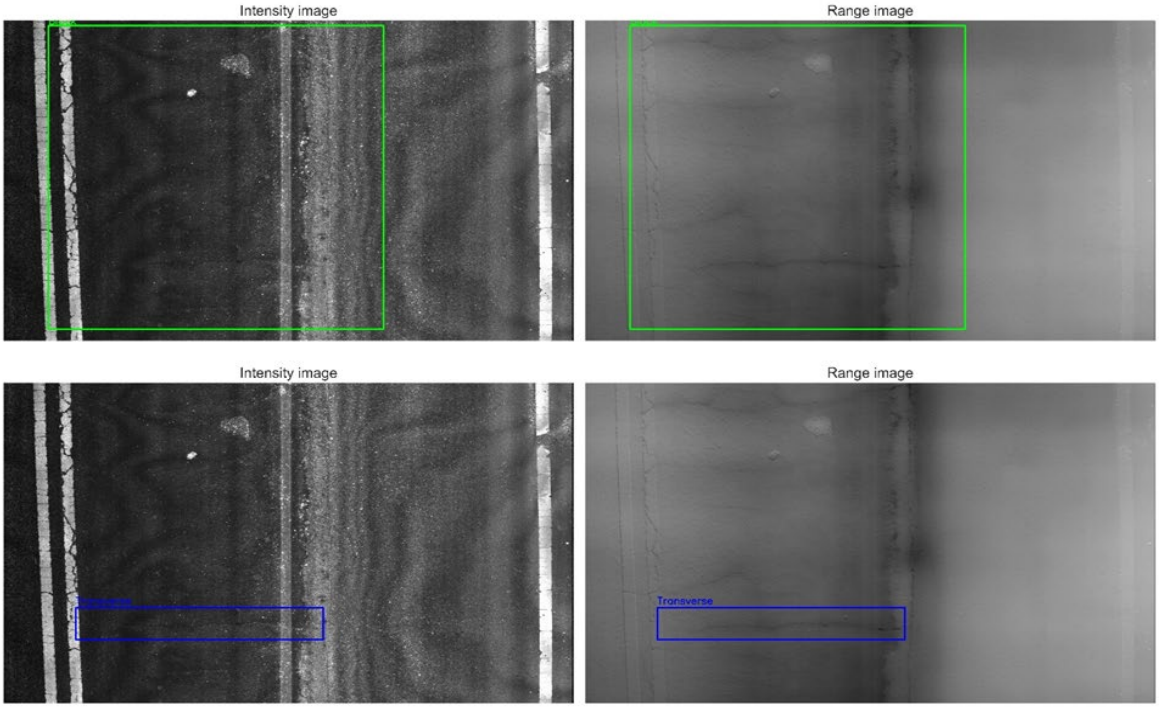


(2) Detection of unsealed cracks

Figure 7.4 Correct detection samples of the ACP model



(1) Missing detection



(2) Wrong and misclassified detection

Figure 7.5 False detection samples of the ACP model

Overall, the model demonstrates strong learning ability and generalization across most ACP distress types, particularly when the distress patterns are visually distinct and well-represented in the training data. The evaluation also reveals some limitations, including poor performance

on underrepresented classes such as potholes, frequent confusion with background features, and occasional failure to recognize the global structure of certain distress types, especially when only local features are detected. These issues suggest that, while the model performs reliably under familiar conditions, its robustness may be challenged when applied to unfamiliar environments.

As the next section analyzes the model's performance on a new dataset from a previously unseen county, particular attention should be given to these areas of concern. Specifically, it is important to assess whether the performance will still stay strong with the new dataset, whether rare or complex distresses such as potholes and block cracking are detected consistently, and whether the model can generalize beyond locally learned features to recognize full distress patterns under new conditions. These factors will be critical in evaluating the model's readiness for deployment.

7.2.2 Test on training and validation datasets of Jointed Concrete Pavements

Table 7.3 Detection performance of the model on the JCP training dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	6,200	13,551	0.909	0.801	0.869	0.584
Failed joint	6,200	413	0.962	0.985	0.993	0.688
Corner break	6,200	228	0.867	0.883	0.896	0.501
Punchout	6,200	73	0.871	0.932	0.944	0.650
Asphalt patch	6,200	262	0.917	0.874	0.948	0.643
Failed concrete patch	6,200	34	0.999	0.735	0.954	0.658
D-cracking	6,200	5	1.000	0.000	0.079	0.0459
Popout	6,200	29	0.742	0.241	0.581	0.322
Longitudinal crack	6,200	905	0.882	0.934	0.946	0.620
Sealed longitudinal	6,200	351	0.920	0.953	0.972	0.656
Concrete patch	6,200	721	0.955	0.933	0.970	0.693
Transverse crack	6,200	546	0.868	0.954	0.976	0.633
Joint crack	6,200	3,296	0.954	0.987	0.990	0.698
Sealed transverse crack	6,200	38	0.885	0.868	0.896	0.581
Slab edge	6,200	6,650	0.953	0.967	0.985	0.762

Table 7.3 presents the detection performance of the model on the JCP training dataset, encompassing 6,200 images and over 13,000 annotated distress instances across 15 classes. The overall detection precision and recall are 0.909 and 0.801, respectively, with a mean average precision of 0.869 at IoU = 0.5 (mAP50) and 0.584 across IoU thresholds (mAP50–95). Most classes exhibit high precision and recall, indicating effective learning during training. Notably, frequent and visually distinctive distresses such as joint cracks, slab edge, and sealed longitudinal cracks achieve both high mAP50 and mAP50–95 (e.g., slab edge with 0.985 and 0.762, respectively). Conversely, rare classes such as D-cracking and Popout show poor

generalization, reflected in their low recall and mAP metrics, likely due to their limited instances (e.g., 5 and 29, respectively). In particular, D- cracking achieves perfect precision but zero recall, implying complete failure to detect any instance despite its inclusion in training. These results suggest that while the model performs well for dominant distress types, additional strategies such as data augmentation or class-balancing are needed to improve performance on underrepresented categories.

Table 7.4 presents the detection performance of the model on the JCP validation dataset, which includes 1,550 images and a total of 3,392 annotated distress instances across 14 classes. The overall detection metrics indicate moderate performance, with a precision of 0.762, recall of 0.624, mAP50 of 0.670, and mAP50–95 of 0.367. The model performs well in dominant classes such as Slab edge and Joint crack, achieving high mAP50 values of 0.975 and 0.961, respectively, and strong precision-recall balance. However, for less frequent distresses like the Popout, Failed concrete patch, and punchout, both recall and mAP scores are significantly lower. These results suggest that while the model can reliably detect common and well-defined distresses, it struggles with rare types, likely due to limited representation in the validation dataset.

Table 7.4 Detection performance of the model on the JCP validation dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	1,550	3,392	0.762	0.624	0.670	0.367
Failed joint	1,550	106	0.751	0.708	0.780	0.409
Corner break	1,550	72	0.635	0.514	0.577	0.198
Punchout	1,550	23	0.497	0.387	0.399	0.169
Asphalt patch	1,550	84	0.745	0.583	0.698	0.428
Failed concrete patch	1,550	8	0.388	0.250	0.227	0.131
Popout	1,550	10	1.000	0.000	0.027	0.000942
Longitudinal crack	1,550	192	0.759	0.724	0.821	0.378
Sealed longitudinal	1,550	90	0.840	0.833	0.827	0.446
Concrete patch	1,550	154	0.756	0.805	0.821	0.516
Transverse crack	1,550	103	0.822	0.647	0.720	0.352
Joint crack	1,550	831	0.920	0.954	0.961	0.592
Sealed transverse crack	1,550	12	0.861	0.750	0.881	0.446
Slab edge	1,550	1,707	0.930	0.933	0.975	0.709

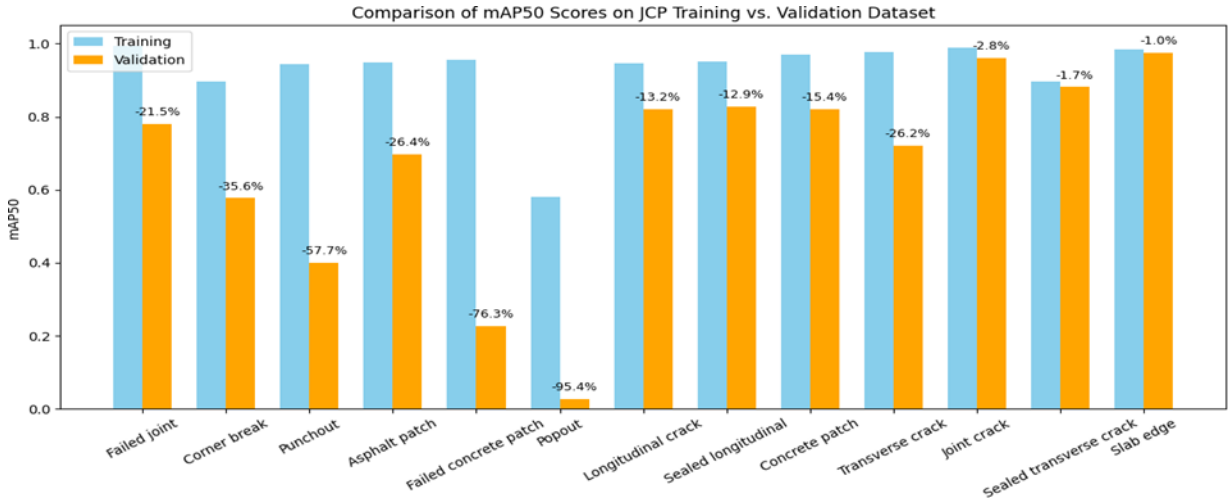


Figure 7.6 Comparison of mAP50 scores on JCP training and validation datasets

Figure 7.6 compares mAP50 scores for each distress class between the JCP training and validation datasets, with blue bars representing training performance and orange bars showing validation performance. The percentage values indicate the relative drop in performance, highlighting the model's generalization capability.

Significant drops are observed for rare classes, such as the Popout and Failed concrete patch, with declines of 95.4% and 76.3%, respectively. For the Popout, the model shows no good fit even with the training dataset. For the Concrete patch, the performance drop could be due to overfitting or inadequate generalization likely due to limited sample sizes and weak visual patterns. Other classes like Punchout, Corner break, and Asphalt patch also exhibit notable performance degradation above 25%, indicating difficulties in transferring learned features to unseen data. On the other hand, classes such as Joint crack, Sealed transverse crack, and Slab edge demonstrate strong generalization with minimal drop, reflecting the model's robustness for well-defined and frequently occurring distresses. Common structural cracks including the Longitudinal and Sealed longitudinal cracks show moderate drops in mAP50 around 13%, indicating that while generalization is acceptable, there is still room for improvement. For the Transverse crack, the drop of 26% indicates a significant generalization issue, possibly due to a large portion of thin transverse cracks with ambiguous features. Overall, the trend suggests that the model performs well on dominant, visually consistent classes but struggles with underrepresented or ambiguous ones, emphasizing the need for targeted strategies to address imbalance and enhance robustness.

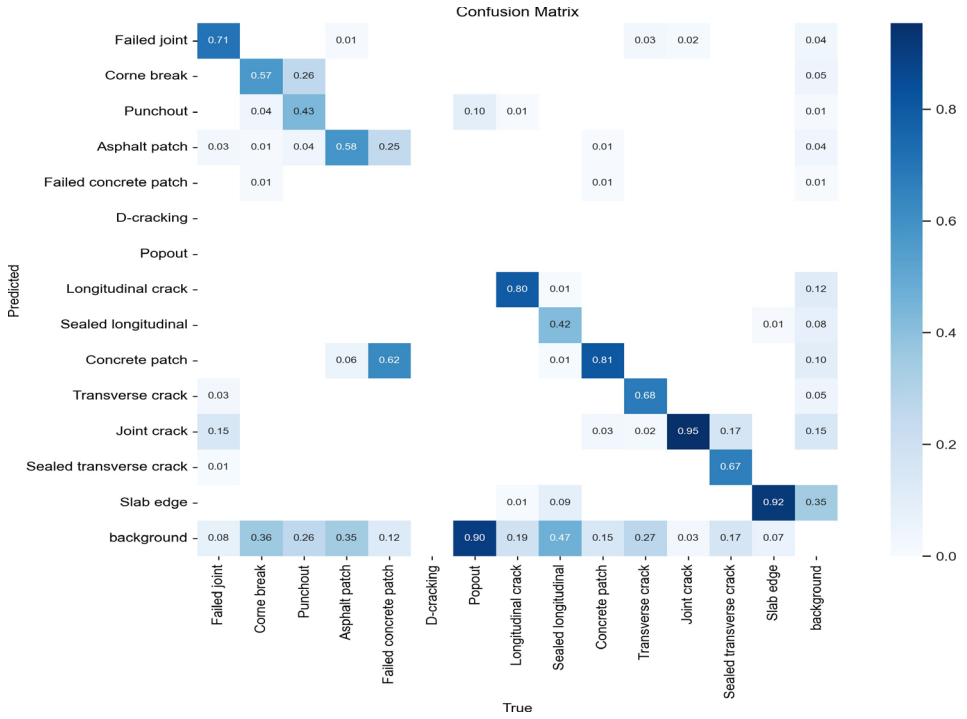
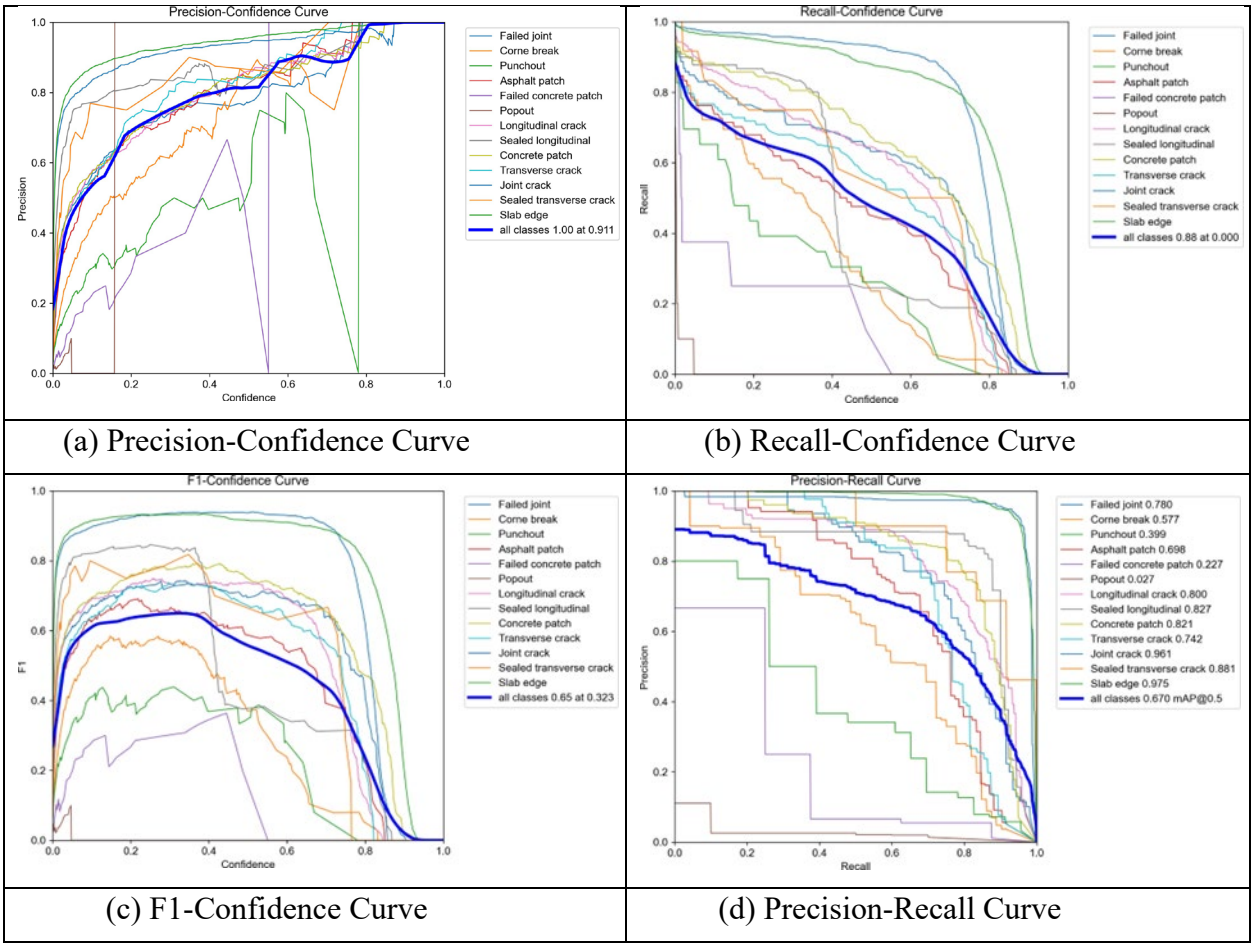


Figure 7.7 JCP confusion matrix

The confusion matrix (Figure 7.7) illustrates the normalized classification results across different JCP distress classes, with true labels along the x-axis and predicted labels on the y-axis. Diagonal values represent correct predictions, while off-diagonal values indicate misclassifications. High diagonal values for classes, such as Joint crack (0.95) and Slab edge (0.92), reflect strong detection performance and class separability.

Several classes exhibit notable confusion, especially among visually similar or overlapping types. The Failed joint is often misclassified as Joint crack, which is understandable since the Failed joints typically manifest as joint cracks overlaid with other distresses, such as spalling or severe faulting. A similar situation exists between Failed concrete patch and Concrete patch, where the former is essentially a degraded version of the latter, making visual differentiation difficult. Sealed transverse crack is also frequently mistaken for the Joint crack, likely due to overlapping features such as linear geometry and filled fissures. Additionally, The Punchout is often confused with the Corner break; both classes are partially defined by visual indicators such as transverse cracks or joints, which are shared elements in their definitions and appearance. Beyond these specific misclassifications, several classes, such as Corner break, Punchout, Asphalt patch, Popout, Sealed longitudinal, and Transverse crack, further reveal the model's limited ability to consistently distinguish these distress types from the surrounding pavement surface. For the Transverse crack, the reason could be the subtle features of most of the extreme thin cracks, while it could be largely due to the lack of instances for other distress classes. Overall, while the model handles dominant classes well, it struggles with rare or visually ambiguous categories, especially those that involve fine-grained distinctions or occur in similar spatial regions.

Figure 7.8 illustrates the model's performance across various JCP distress types, with a focus on how confidence thresholds influence prediction quality.



(a) Precision-Confidence Curve

(b) Recall-Confidence Curve

(c) F1-Confidence Curve

(d) Precision-Recall Curve

Figure 7.8 Model evaluation plots for the detection system over the JCP dataset

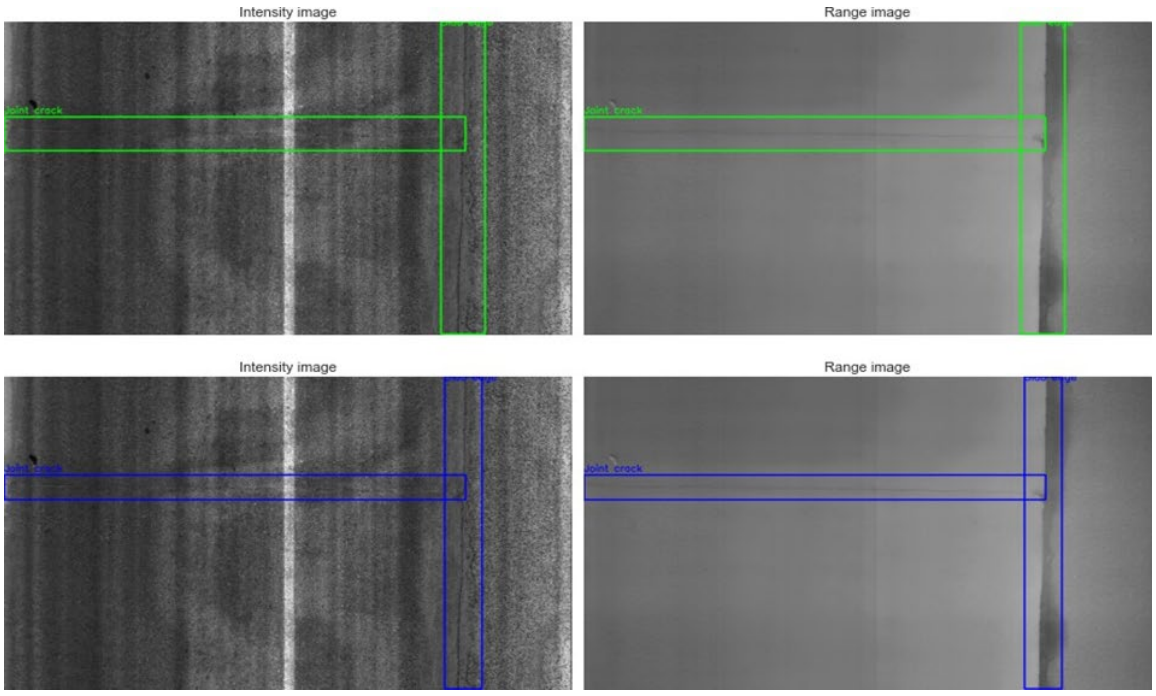
- Precision–Confidence Curve (a): This plot shows that model precision generally increases as the confidence threshold rises. Distress types such as Slab edge, Joint crack, and Sealed longitudinal crack reach near-perfect precision above a threshold of 0.9, indicating strong reliability when the model is confident. Conversely, the Popout and Failed concrete patch exhibit low and unstable precision throughout the entire confidence range, reflecting high rates of false positives and unreliable predictions regardless of threshold.
- Recall–Confidence Curve (b): As expected, recall decreases as the confidence threshold increases. Most classes, including Slab edge, Failed joint, and Concrete patch, maintain relatively high recall across a broad range of thresholds, suggesting consistent detection. However, classes such as Failed concrete patch, Popout, and Punchout exhibit sharp declines in recall even at low confidence levels, indicating poor sensitivity and early rejection of true positives.
- F1–Confidence Curve (c): The F1 curve reflects the balance between precision and

recall, with an overall peak at a confidence threshold of 0.323, where the macro F1 score reaches 0.65. Strong, peaked F1 curves are observed for Slab edge, Joint crack, and Sealed transverse crack, showing well-balanced performance. In contrast, Popout, Punchout, and Failed concrete patch yield flat and low F1 curves, reinforcing the model's difficulty in detecting these distress types reliably.

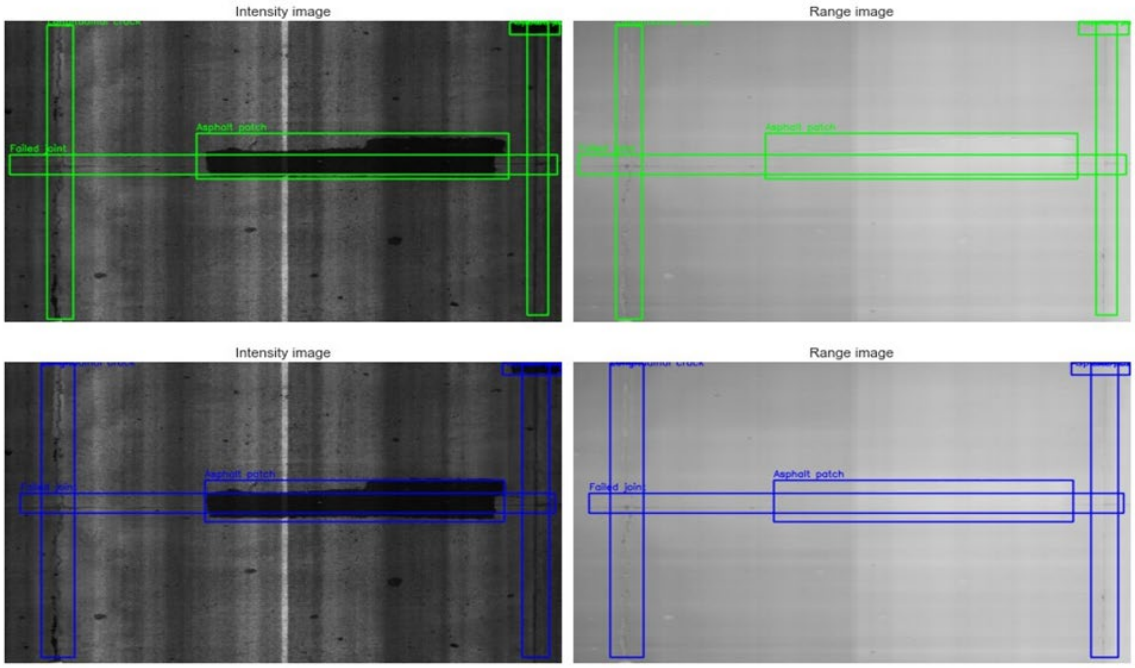
- Precision–Recall Curve (d): This curve captures the full range of the trade-off between precision and recall. Slab edge achieves the highest area under the curve with an mAP50 of 0.975, followed by Joint crack (0.961) and Sealed transverse crack (0.881). These classes maintain high precision even as recall increases, demonstrating robust detection capabilities. In contrast, Popout has the lowest mAP50 of 0.027, with a steep drop in both precision and recall, confirming it as the most problematic class for the model.

Figure 7.9 presents examples of correct detection results from the JCP model, with green boxes representing ground truth annotations and blue boxes representing model predictions. In Figure 7.9 (Subfigure 1), the model accurately detects both joint cracks and slab edges in a clean surface con- text, with well-aligned bounding boxes indicating precise localization of key structural features. In Figure 7.9 (Subfigure 2), the model correctly identifies a failed joint where an asphalt patch overlaps a joint crack, capturing the composite nature of the distress. This classification aligns with the definition of failed joints, which often occur as a combination of existing joint cracks and additional surface deterioration. The successful detection of these compound patterns demonstrates the model's ability to interpret overlapping features and contextual distress relationships in JCP.

Figure 7.10 illustrates incorrect prediction examples made by the JCP model, with green boxes representing ground truth and blue boxes representing model predictions. In Figure 7.10 (Subfigure 1), the ground truth includes a punchout near the slab corner and two thin, oblique longitudinal cracks. The model misclassifies the punchout as a corner break, likely due to their shared geometric cues such as crack location near the slab edge and associated transverse features. Additionally, both longitudinal cracks are missed entirely, suggesting weak feature activation or insufficient representation of oblique cracks in the training data. In Figure 7.10 (Subfigure 2), a small corner break located at the bottom-right of the slab is not detected. The missed detection may be attributed to its subtle contrast against the surrounding background and relatively small spatial footprint, which can challenge the model's ability to localize and classify fine-grained corner distress. These examples reflect the model's current limitations in distinguishing distress types with overlapping visual features and detecting low-contrast or small-area distresses.

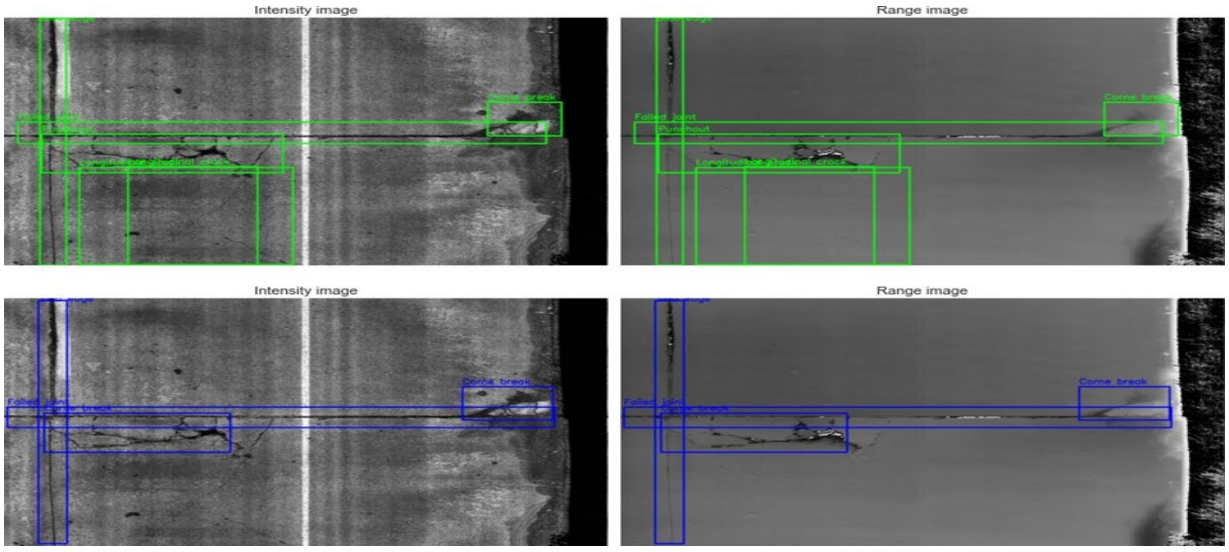


(1) Correct detection of Joints and Slab Edges

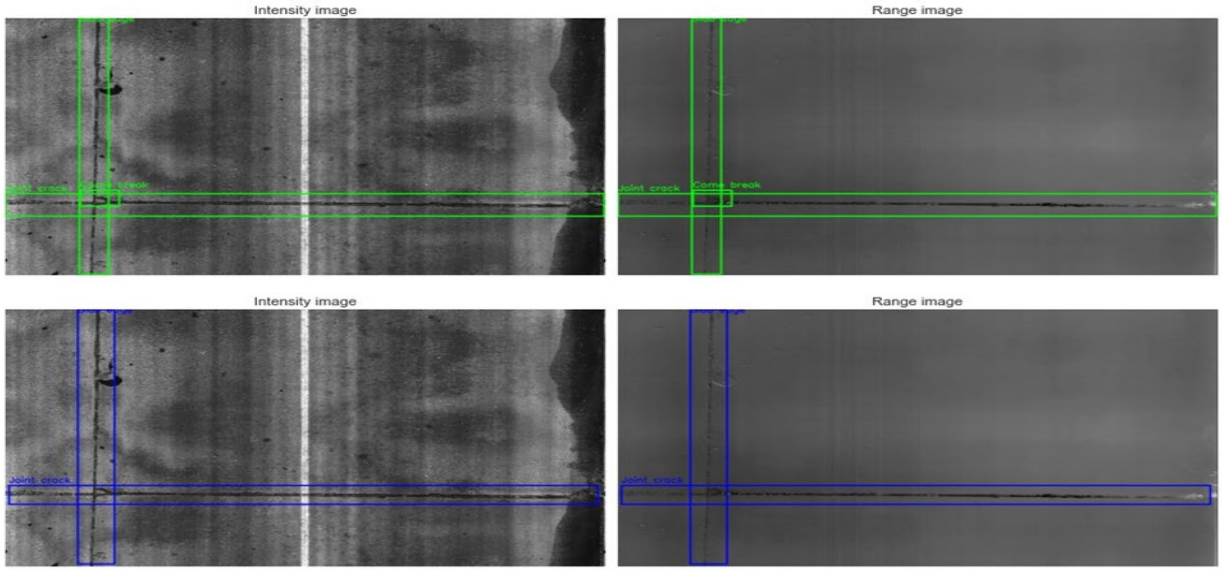


(2) A Joint crack overlapped with Asphalt Patch and thus considered as Failed Joint and Cracks

Figure 7.9 Correct detection samples of the JCP model



(1) Misclassified punchout and two missed longitudinal cracks



(2) Missed detection of a corner break

Figure 7.10 False detection samples of the JCP model

Overall, the model demonstrates strong detection capability and moderate generalization across most JCP distress types, especially for frequent and structurally well-defined classes such as Slab edge, Joint crack, and Sealed cracks. Evaluation metrics and qualitative examples confirm that the model can reliably detect these categories with high precision and recall, particularly when the visual patterns are clear and consistently represented in the training data. However, some limitations are evident. The model struggles with rare and visually subtle classes such as Popout, Failed concrete patch, and Punchout, which show low detection scores and frequent misclassifications. Common errors include misclassification due to overlapping visual features,

such as punchouts being confused with corner breaks, or missing small or oblique cracks. In addition, confusion with background textures is a recurring issue for distress types with weak contrast or irregular shape boundaries, revealing limitations in both sensitivity and spatial pattern aggregation. These findings suggest that while the model's robustness may degrade when faced with visual ambiguity, low-frequency classes, or insufficient spatial context.

As the next section analyzes the model's performance on a new dataset from a previously unseen county, particular attention should be given to these areas of concern. Specifically, it is important to assess whether the performance would remain strong with the new dataset, and whether rare or complex distresses such as the Popout and Failed patches are detected consistently. These aspects will be crucial for evaluating the model's suitability for real-world deployment across diverse pavement conditions.

7.2.3 Test on training and validation datasets of Continuously Reinforced Concrete Pavements

Table 7.5 presents the detection performance of the model on the CRCP training dataset, encompassing 4,620 images and 11,026 annotated distress instances. Overall, the model achieved high precision (P = 0.886) and a reasonably strong recall (R = 0.711), with a mean average precision of 0.761 at IoU = 0.5 (mAP50) and 0.461 across IoU thresholds from 0.5 to 0.95 (mAP50-95). Performance varies significantly by distress type. Patches and punchouts were detected with high accuracy, e.g., Asphalt patch (mAP50 = 0.987), Concrete patch (mAP50 = 0.959), and Punchout (mAP50 = 0.953)—indicating the model's robustness for large or high-contrast features. In contrast, longitudinal crack detection showed limitations, with a low recall of 0.239 and mAP50 of 0.371, suggesting difficulty in identifying thin or low-contrast features. While transverse cracks possess similar subtle features as longitudinal cracks, their performance is much stronger than that of longitudinal cracks, indicating that with enough training samples, acceptable performance can still be achieved. These results highlight strong learning performance for common and visually prominent distresses, while revealing vulnerabilities in detecting subtle or rare defects.

Table 7.5 Detection performance of the model on the CRCP training dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	4,620	11,026	0.886	0.711	0.761	0.461
Longitudinal crack	4,620	299	0.749	0.239	0.371	0.133
Sealed longitudinal crack	4,620	161	0.851	0.677	0.777	0.401
Punchout	4,620	63	0.913	0.921	0.953	0.617
Asphalt patch	4,620	884	0.952	0.965	0.987	0.664
Concrete patch	4,620	149	0.925	0.911	0.959	0.652
Transverse crack	4,620	8,975	0.814	0.850	0.893	0.552
Sealed transverse crack	4,620	321	0.879	0.930	0.942	0.543
Spalled longitudinal crack	4,620	7	1.000	0.000	0.000	0.000
Spalled transverse crack	4,620	167	0.894	0.909	0.967	0.586

Table 7.6 Detection performance of the model on the CRCP validation dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	1,156	2,753	0.712	0.696	0.739	0.384
Longitudinal crack	1,156	73	0.520	0.205	0.190	0.0444
Sealed longitudinal crack	1,156	21	0.686	0.714	0.754	0.376
Punchout	1,156	18	0.730	0.754	0.829	0.338
Asphalt patch	1,156	226	0.898	0.934	0.965	0.619
Concrete patch	1,156	31	0.629	0.581	0.714	0.415
Transverse crack	1,156	2,259	0.736	0.826	0.835	0.485
Sealed transverse crack	1,156	77	0.801	0.870	0.877	0.497
Spalled transverse crack	1,156	48	0.696	0.688	0.751	0.328

Table 7.6 summarizes the model's detection performance on the CRCP validation dataset, comprising 1,156 images and 2,753 distress instances. The overall performance remains solid, with a precision of 0.712 and recall of 0.696, resulting in a mAP50 of 0.739 and mAP50-95 of 0.384. The model demonstrates particularly high accuracy in detecting asphalt patches (P = 0.898, R = 0.934, mAP50 = 0.965) and sealed transverse cracks (P = 0.801, R = 0.870, mAP50 = 0.877), indicating strong generalization for these visually distinct features. However, longitudinal cracks exhibit weak performance, with the lowest recall (0.205) and mAP50-95 (0.0444), suggesting persistent challenges in identifying fine, elongated distress types in more diverse scenarios. While punchouts, concrete patches, and spalled cracks achieved moderate detection quality, their relatively low instance counts may limit consistency across wider deployments. Overall, the model shows promising generalization capabilities, particularly for common and well-defined distress types, while highlighting areas to improve in detecting less prominent or underrepresented features.

Figure 7.11 presents a comparison of mAP50 scores between the CRCP training and validation datasets across different distress types, revealing notable differences in generalization performance. The most significant drop is seen in the detection of longitudinal cracks, with a 48.8% decrease from training to validation, indicating the model's limited ability to generalize this subtle and often low-contrast distress type. Similarly, concrete patches and spalled transverse cracks show considerable declines of 25.5% and 22.3%, respectively, which may reflect challenges stemming from limited instance diversity or changes in background texture. In contrast, asphalt patches demonstrate strong generalization with only a 2.2% decrease, likely due to their distinct visual characteristics and consistent appearance. Sealed longitudinal cracks, sealed transverse cracks, and transverse cracks exhibit relatively small declines, suggesting that the model has learned to detect these features robustly across different data contexts. Moderate performance drops are also observed for punchouts, indicating potential sensitivity to image variability. These results emphasize that while the model performs well on prominent and well-represented distress types, its generalization capability is reduced for rare, fragmented, or less distinct features, underscoring the need for improved training diversity or targeted model enhancements.

The confusion matrix (see Figure 7.12) provides insight into the classification accuracy and mis-classification trends of the model across CRCP distress types. Diagonal values represent correct predictions, and off-diagonal values indicate confusion with other classes or the background.

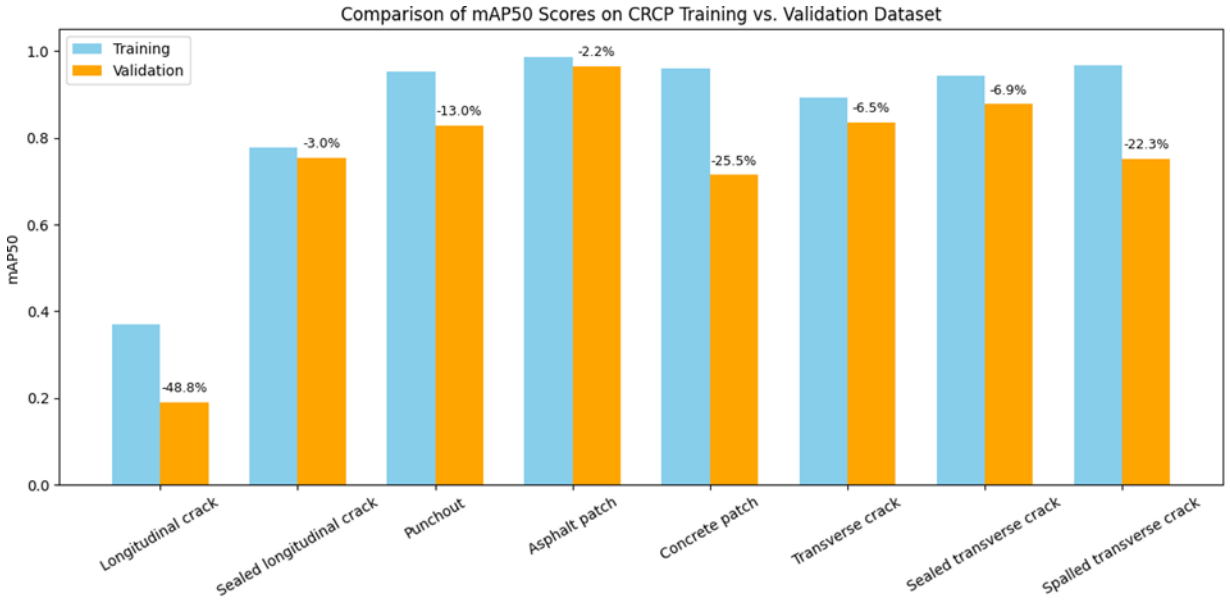


Figure 7.11 Comparison of mAP50 scores on CRCP training and validation datasets

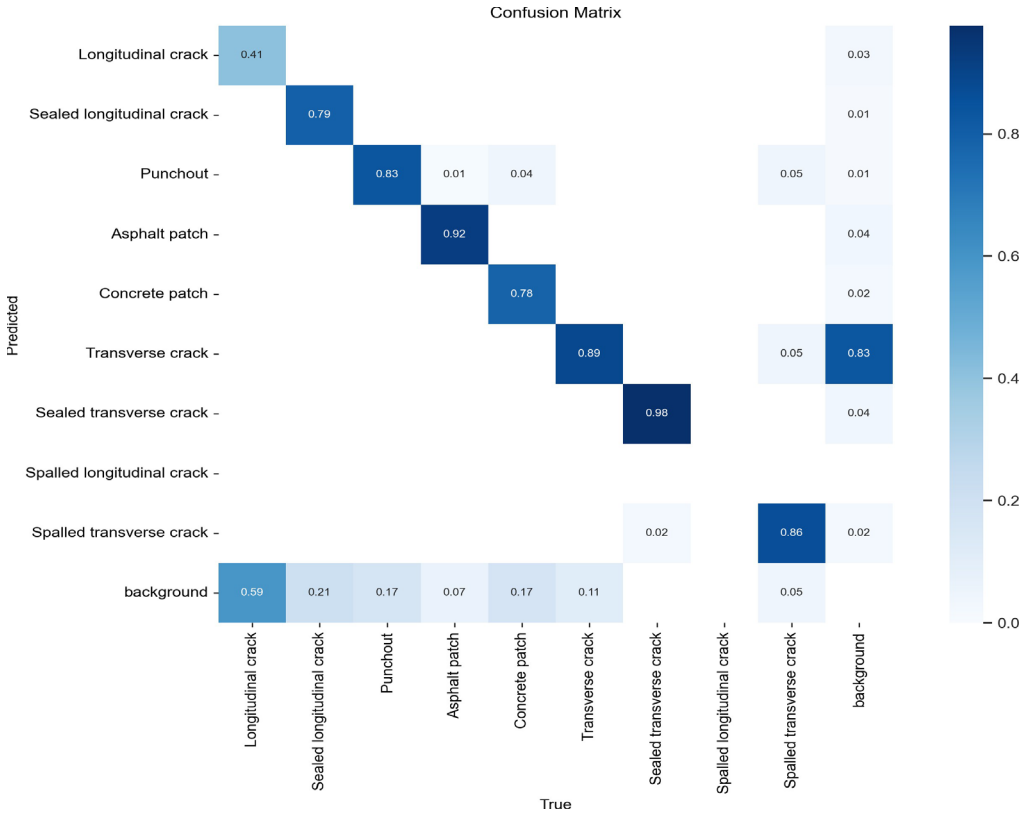


Figure 7.12 CRCP confusion matrix

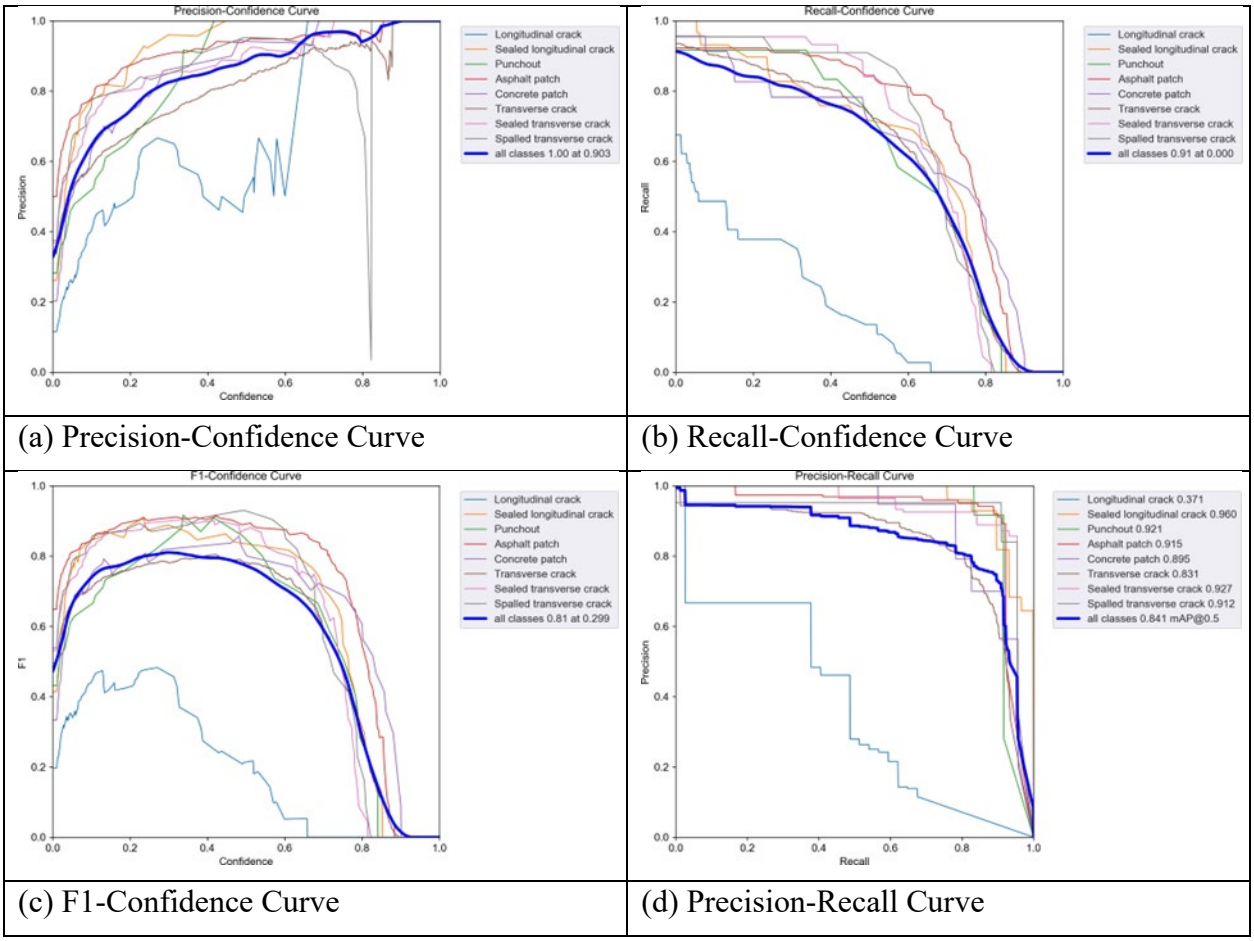
Overall, the model performs well in distinguishing most distress types, with high true positive rates for sealed transverse cracks (0.96), asphalt patches (0.92), transverse cracks (0.89), spalled transverse cracks (0.86), and punchouts (0.83). These values reflect strong confidence and discriminative ability for these visually distinct and structurally consistent features. Concrete patches and sealed longitudinal cracks are also reasonably well predicted, with accuracy values of 0.78 and 0.79, respectively.

However, longitudinal cracks show notable confusion, with only 0.41 of instances correctly classified, and the remaining 59% misclassified as background. This highlights the model's difficulty in detecting thin or faint longitudinal features, which often lack contrast and may need more training samples. Additionally, punchouts are misclassified as asphalt patches and transverse cracks in small proportions (0.04 and 0.01, respectively), suggesting some overlap in visual appearance or shape under certain conditions.

The last row and column labeled "background" further confirm that misclassification into background is a primary issue for several distress types, particularly longitudinal and sealed longitudinal cracks, as well as concrete patch and punchout classes to a lesser extent. These trends point to limitations in the model's sensitivity to low-contrast or underrepresented features and suggest the need for enhanced feature learning, data balancing, or improved pre-processing to better distinguish distress from background noise.

Figure 7.13 illustrates the model's performance across various CRCP distress types, focusing on how confidence thresholds affect precision, recall, and overall detection quality.

- Precision–Confidence Curve (a): This plot shows that precision generally increases as the confidence threshold rises. Most distress types, such as Sealed longitudinal, Asphalt patch, and Sealed transverse, demonstrate high and stable precision across a wide range of thresh- olds, with values approaching 1.00 at higher confidence levels, indicating reliable predictions when the model is confident. In contrast, the Longitudinal crack exhibits noisy and consistently low precision, suggesting a lack of reliable confidence calibration and frequent false positives for this class.
- Recall–Confidence Curve (b): As expected, recall decreases with increasing confidence thresholds for most classes. While the decline is gradual for well-detected distresses like Sealed transverse, Spalled transverse, and Asphalt patch—indicating that the model retains good sensitivity even at higher thresholds—the curve for the Longitudinal crack drops off sharply, with recall falling below 0.5 early and degrading quickly, reflecting weak sensitivity and poor class detection even under lenient confidence settings.
- F1–Confidence Curve (c): The F1-score curve highlights the balance between precision and recall, peaking at a confidence threshold of 0.299 where the overall F1 score reaches 0.81. Most classes show smooth, unimodal curves with strong F1 performance, especially Asphalt patch, Punchout, and Sealed transverse. However, the Longitudinal crack is again an outlier, with a low and unstable F1 curve, underscoring the model's inability to achieve a reliable balance between precision and recall for this distress type.



(a) Precision-Confidence Curve

(b) Recall-Confidence Curve

(c) F1-Confidence Curve

(d) Precision-Recall Curve

Figure 7.13 Model evaluation plots for the detection system over the CRCP dataset

- Precision–Recall Curve (d): This plot summarizes the trade-off between precision and recall across all thresholds. Sealed longitudinal crack achieves the highest area under the curve with an mAP50 of 0.960, followed closely by Sealed transverse (0.927), Punchout (0.921), and Asphalt patch (0.915). These classes show consistent high precision across nearly the entire recall range, indicating robust generalization. In contrast, Longitudinal crack has the lowest mAP50 of 0.371 and a steeply descending curve, reaffirming it as the model's most underperforming and error-prone class.

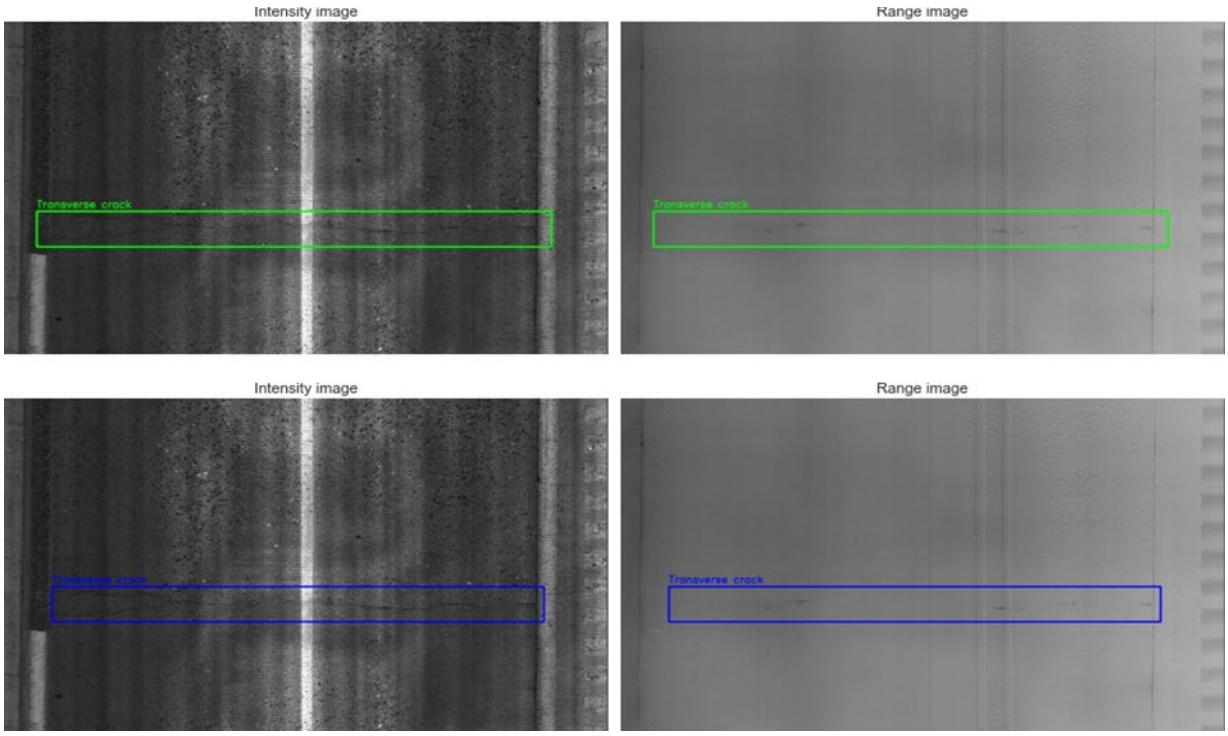
Figure 7.14 presents examples of correct detection results from the CRCP model, with green boxes indicating ground truth annotations and blue boxes representing model predictions. In Figure 7.14 (Subfigure 1), the model successfully detects transverse cracks in both intensity and range images with precise localization and minimal background interference, suggesting reliable performance for this distress type under standard visual conditions. In Figure 7.14 (Subfigure 2), the model correctly identifies multiple distress types, including transverse and spalled transverse cracks, and asphalt patch, across a more complex scene featuring overlapping features and surface variation. The consistent alignment between predictions and ground truth in these samples

highlights the model's strong detection capability across both simple and visually cluttered scenarios.

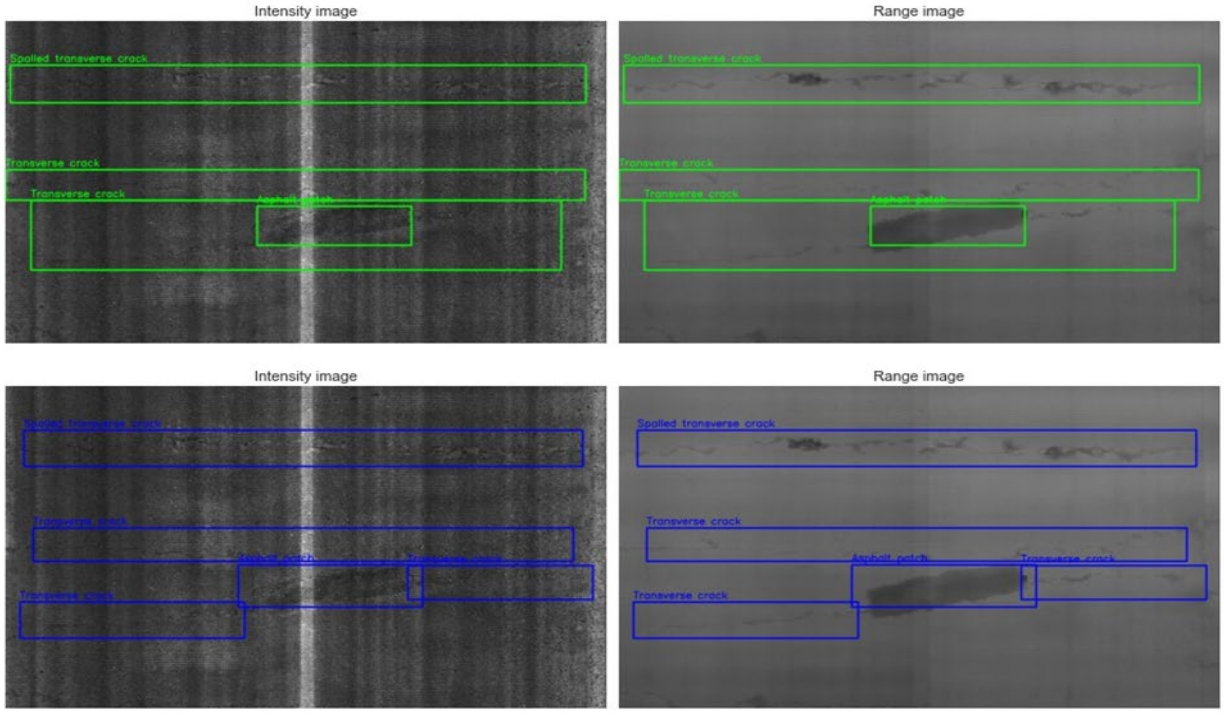
Figure 7.15 illustrates incorrect prediction examples made by the CRCP model, with green boxes representing ground truth and blue boxes representing model predictions. In Figure 7.5 (Subfigure 1), the ground truth indicates the presence of a longitudinal crack with moderate visual contrast in both the intensity and range images. However, the model fails to detect this feature entirely, resulting in a true missed detection. This failure may be attributed to the low representation of similar longitudinal crack patterns in the training set or insufficient feature activation in this region, both of which hinder the model's sensitivity to subtle linear distresses. In Figure 7.15 (Subfigure 2), a concrete patch containing a centered manhole is misclassified as a punchout. While the geometric and grayscale features of the manhole may superficially resemble punchout patterns, this object is not part of the training data and therefore represents an unknown or out-of-distribution input. This misclassification suggests that the model may rely heavily on local structural cues without adequate contextual understanding, leading to errors when encountering unfamiliar surface elements. These examples highlight key limitations in both sensitivity to weakly defined cracks and robustness to non-distress artifacts.

Overall, the model demonstrates strong learning ability and moderate generalization across most CRCP distress types, particularly for features that are visually distinct and consistently represented in the training data, such as sealed cracks, patches, and transverse cracks. Evaluation results show high precision and recall for these classes across both training and validation datasets, as well as consistent performance in F1 and precision-recall metrics. However, some limitations are evident. The model struggles with underrepresented or visually ambiguous classes, most notably longitudinal cracks, which exhibit unstable precision-recall behavior, frequent misclassification as background, and poor F1 score performance. Confusion matrix analysis further highlights the dominance of background errors for certain classes, while qualitative inspection reveals missed detections and classification errors, particularly in complex scenes or when the model encounters novel surface features like manholes that were not included during training. These findings suggest that, although the model is effective under familiar conditions, its robustness is weakened when facing subtle, low-contrast, or previously unseen patterns.

As the next section analyzes the model's performance on a real-world dataset from a previously unseen county, particular attention should be given to these areas of concern. Specifically, it is important to examine whether performance would remain strong with the new dataset, and whether rare or composite distresses such as longitudinal cracks and concrete patches are detected consistently. These factors will be critical in evaluating the model's readiness for deployment across diverse pavement environments.

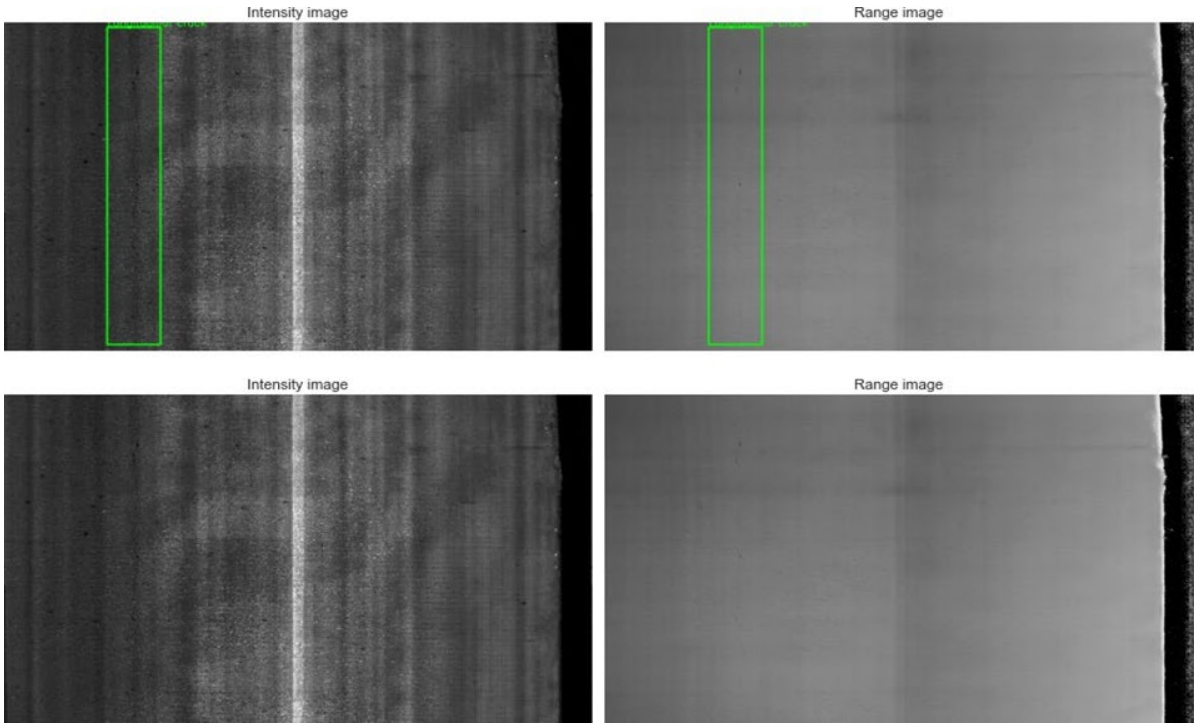


(1) Correct detection of transverse cracks

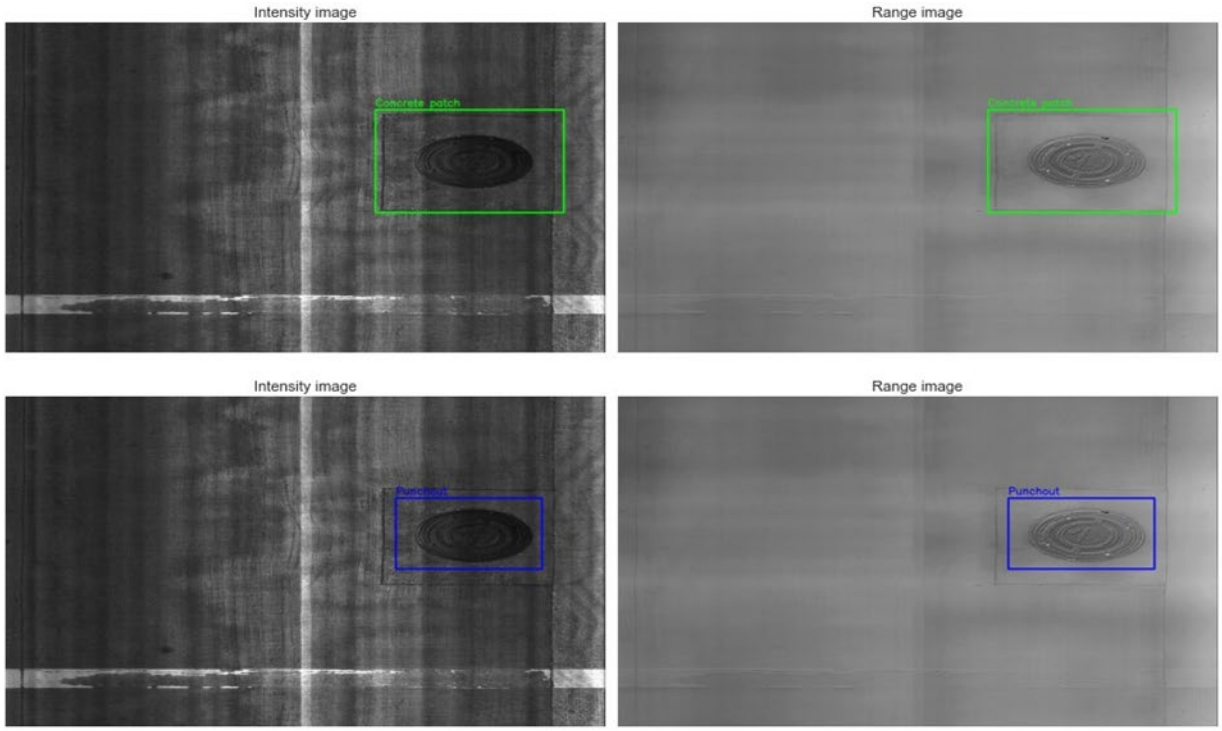


(2) Correct detection of transvers and spalled transverse cracks, and asphalt patches

Figure 7.14 Correct detection samples of the CRCP model



(1) One missed Longitudinal Crack



(2) One misclassified Concrete Patch

Figure 7.15 False detection samples of the CRCP model

7.3 Model's Performance on Datasets Collected from Brazoria County in Texas in 2024

This section examines the model's performance on the Brazoria County dataset, which represents a previously unseen environment and was not included in the training data. As such, this evaluation provides a critical test of the model's generalization ability beyond the distribution of the original dataset. Given the inclusion of different pavement types, including the ACP, JCP, and CRCP, this analysis is structured to evaluate both the overall detection performance and the performance on individual distress types within each pavement category. Particular attention is paid to whether the model maintains consistent accuracy across varying surface textures and structural characteristics, and whether it can correctly distinguish complex or rare distresses such as potholes, block cracking, punchouts, or corner breaks within the respective pavement contexts. These insights are essential for assessing the model's robustness and its readiness for deployment across diverse real-world roadway conditions.

7.3.1 Datasets

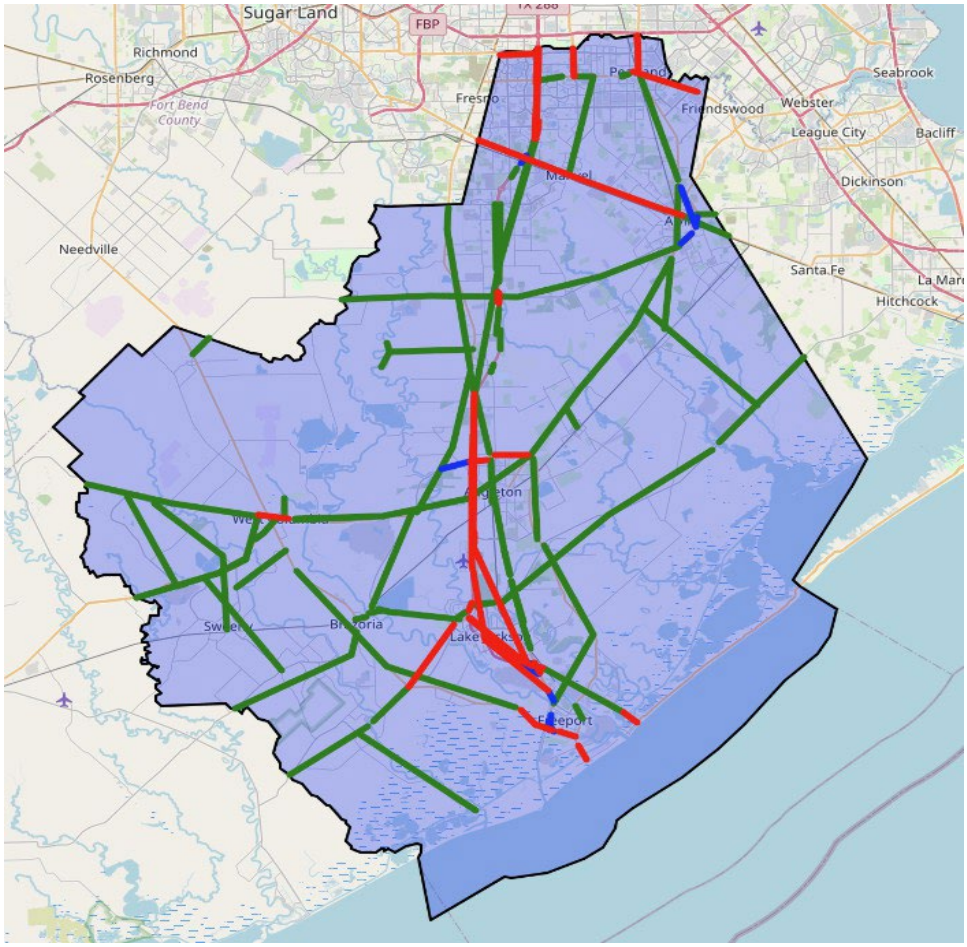


Figure 7.16 Distribution of different pavement types in Brazoria County based on 2024 PIS data: 1) green denotes ACP pavement, 2) blue denotes JCP pavement, and 3) red denotes CRCP pavement

This study utilizes pavement condition data collected in 2024 as part of the Pavement Management Information System (PMIS) maintained by the Texas Department of Transportation (TxDOT). The dataset focuses on Brazoria County, Texas, which was selected as a representative testbed due to its diverse pavement types and extensive roadway network.

As illustrated in Figure 7.16, the roadway segments within Brazoria County comprise three major pavement types: ACP, JCP, and CRCP. These pavement types are visualized using distinct color codes: green for ACP, blue for JCP, and red for CRCP. The distribution shows a predominance of ACP segments, with JCP and CRCP segments distributed more sparsely across the network. The data were collected using automated survey vehicles and recorded in the standard 2D/3D format (.psi), which includes synchronized intensity and range images.

To ensure representative coverage and manageable annotation scope, the collected roadway sections were first grouped by pavement type and then aggregated based on highway names. From each highway group, a continuous segment ranging randomly from 0.5 to 2 miles in length was selected to serve as the representative section for this study. While this sampling strategy does not guarantee a balanced distribution of pavement types, it allows for the inclusion of diverse surface conditions and construction characteristics observed across the network.

The selected pavement sections were manually annotated using guidelines aligned with the image library developed Chapter 3. This consistent annotation framework enhances comparability across pavement types and supports downstream model evaluation. This dataset offers a well-structured and diverse bench- mark for assessing the performance and generalizability of the developed models under real-world conditions.

A total of 17 representative ACP pavement sections were selected for analysis, as summarized in Table 7.7. These sections are primarily located along farm-to-market roads and state highways, reflecting the typical distribution of ACP surfaces within Brazoria County. The total length of each selected segment ranges from approximately 0.7 to 1.5 miles, with the number of image files per section varying accordingly.

Figure 7.17 illustrates the distribution of distress instances across eight annotated distress types within these ACP sections. The most frequently observed classes are sealed longitudinal cracks (2,081 instances), longitudinal cracks (1,864), and transverse cracks (1,169), followed by sealed transverse (1,014) and lane longitudinal (431) cracks. In contrast, block cracking and alligator cracking are sparsely represented, with only 8 and 51 instances, respectively, and joint cracking appears only twice. Due to their extremely low sample sizes, distress types with fewer than 50 instances, namely block and joint cracking, are not considered statistically meaningful and are therefore excluded from the evaluation of the model's performance.

Table 7.7 Selected ACP pavement sections

Section Name	# Files	Total Length (miles)
BS0288BK	150	0.735
FM0517-K	266	1.041
FM0518-R	600	1.426
FM0521-K	201	0.959
FM0523-K	285	0.786
FM0524-K	304	1.236
FM0655-K	278	1.180
FM1128-K	333	0.946
FM1301-K	305	1.505
FM1459-K	300	1.322
FM1462-K	300	1.481
FM2004-K	300	1.503
SH0006-L	459	1.330
SH0035-K	305	1.321
SH0036-K	285	1.192
SH0288-X	301	0.931
SH0332-K	299	1.375

A total of four representative JCP pavement sections were selected for analysis, as listed in Table 7.8. These segments are drawn from both farm-to-market roads and state highways, with total section lengths ranging from approximately 0.7 to 1.9 miles.

Table 7.8 Selected JCP pavement sections

Section Name	# Files	Total Length (miles)
FM0523-K	429	1.932
SH0035-L	357	1.306
SH0288-A	262	0.808
SH0288-R	266	0.693

Figure 7.18 presents the distribution of annotated distress instances for the JCP sections. The most frequently observed classes are the Slab edge (1,837 instances) and Joint crack (1,086 instances), followed by the Transverse crack (141) and Longitudinal crack (85). Several other distress types, including the Asphalt patch, Failed joint, and Concrete patch, have lower occurrence counts, while types such as the Corner break, Punchout, Popout, and Sealed longitudinal crack appear only in very small numbers (10 instances). As with the ACP dataset, any distress class with fewer than 50 instances is not considered statistically meaningful and is therefore excluded from the evaluation of the model's performance.

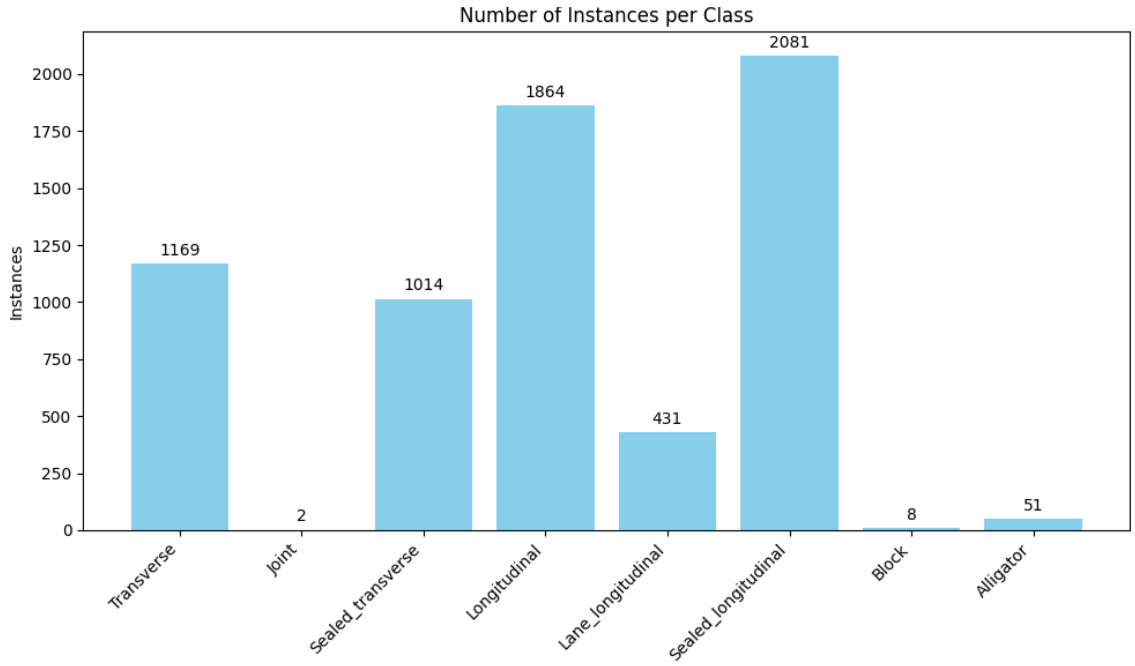


Figure 7.17 Distribution of the number of individual distress classes of selected ACP sections

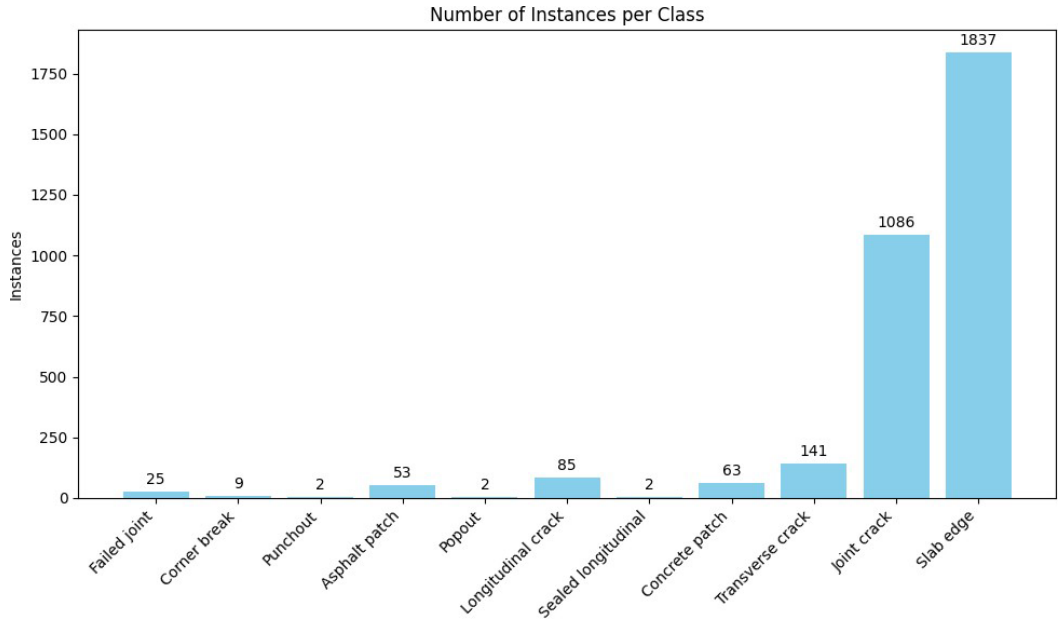


Figure 7.18 Distribution of the number of individual distress classes of selected JCP sections

In addition to ACP and JCP sections, nine representative CRCP pavement sections were selected, as shown in Table 7.9. These sections are distributed across both farm-to-market roads and state highways, with segment lengths ranging from approximately 0.49 to 1.18 miles.

Table 7.9 Selected CRCP pavement sections

Section Name	# Files	Total Length (miles)
FM0518-K	397	1.175
FM0523-K	215	0.977
FM0865-L	206	0.729
FM1495-K	258	0.818
FM2004-K	233	0.671
SH0035-L	219	0.492
SH0036-K	223	1.091
SH0288-X	294	0.848
SH0332-K	265	0.981

Figure 7.19 presents the distribution of distress instances for the CRCP sections. The Transverse cracking is by far the most dominant distress type, with 2,838 instances, followed by spalled transverse cracks (231) and longitudinal cracks (58). All other distress types, including sealed cracks, concrete patches, and asphalt patches, appear infrequently, with instance counting well below the 50-instance threshold. As in the ACP and JCP analyses, only distress classes with at least 50 occurrences are considered statistically meaningful and included in model performance evaluation. The strong skew toward the transverse cracking in CRCP sections highlights the characteristic distress behavior of this pavement type and underscores the importance of evaluating detection performance on dominant versus rare classes.

7.3.2 Evaluation metrics

To assess the generalization ability of the model on the previously unseen Brazoria County dataset, two evaluation metrics were employed. These metrics are designed to capture both the overall detection consistency across all pavement sections and the model's ability to distinguish individual distress types under varying conditions. The analysis is conducted separately for ACP, JCP, and CRCP, as each pavement type presents unique distress characteristics and surface textures that may affect model performance.

7.3.2.1 Class-wise Generalization Drop (Validation vs. Brazoria)

To explicitly measure the impact of domain shift, the generalization drop is computed as the difference in mAP50 for each distress class between the validation dataset (included in the image library) and the Brazoria County test set. This metric highlights the degree to which performance deteriorates when the model encounters new visual environments, helping to identify distress types that are especially vulnerable to overfitting or underrepresentation during training. To ensure the reliability of this comparison, distress classes with fewer than 50 instances in the Brazoria County test set are excluded from the generalization drop analysis.

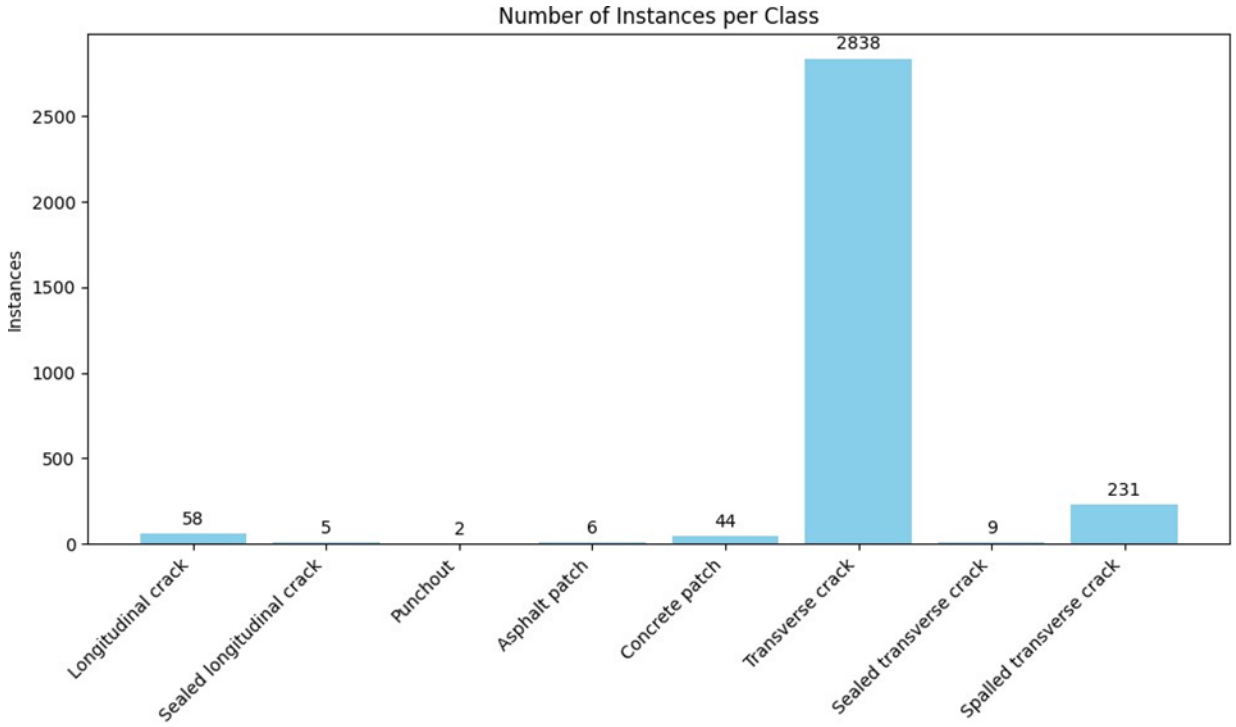


Figure 7.19 Distribution of the number of individual distress classes of CRCP in Brazoria County

7.3.2.2 mAP50 per Distress Class per Section

To quantify the model's ability to differentiate between specific distress types, the mAP50 is calculated for each class across different pavement sections. This class-level analysis enables evaluation of how well the model generalizes its learned representations of individual distresses under varying visual and structural pavement contexts. For reliability, distress classes with fewer than 50 instances within a given pavement section are excluded from the analysis, as their occurrence is not statistically meaningful for performance interpretation. Together, these metrics provide a comprehensive view of the model's robustness and limitations when applied to real-world pavement conditions beyond the training domain. The results of this evaluation are discussed in the following subsections for each pavement type.

7.3.3 ACP

Table 7.10 summarizes the detection performance of the model on the ACP dataset in Brazoria County. The model demonstrates strong performance on major distress types with sufficient sample sizes, including transverse cracks (0.818), sealed transverse cracks (0.958), longitudinal cracks (0.823), and sealed longitudinal cracks (0.899). These results suggest that the model is capable of reliably detecting common and visually distinctive distress types in ACP sections.

In contrast, the performance on lane longitudinal cracks (e.g., mAP50 is 0.599) and alligator cracking (e.g., mAP50 is 0.577) is noticeably lower, indicating that these distress types present

greater detection challenges. Distress types with fewer than 50 instances, including the Joint (2 instances) and Block cracking (8 instances), are excluded from interpretation in accordance with the statistical threshold previously established.

Table 7.10 Detection performance of the model on the ACP dataset (Brazoria County)

Class	Images	Instances	P	R	mAP50	mAP50-95
All	,5265	6,620	0.611	0.604	0.605	0.383
Transverse	5,265	1,169	0.821	0.706	0.818	0.459
Joint	5,265	2	0.0294	0.500	0.0217	0.0152
Sealed_transverse	5,265	1,014	0.920	0.855	0.958	0.532
Longitudinal	5,265	1,864	0.805	0.740	0.823	0.546
Lane_longitudinal	5,265	431	0.529	0.599	0.599	0.427
Sealed_longitudinal	5,265	2,081	0.820	0.855	0.899	0.654
Block	5,265	8	0.167	0.125	0.142	0.0909
Alligator	5,265	51	0.797	0.451	0.577	0.340

Figure 7.20 compares the mAP50 scores of the model for various ACP distress types between the validation dataset and the 2024 collection from Brazoria County, providing insight into the model's generalization capability across domains. Distress types with fewer than 50 instances in the new dataset, such as joints and block cracking, are excluded from this comparison due to their lack of statistical significance.

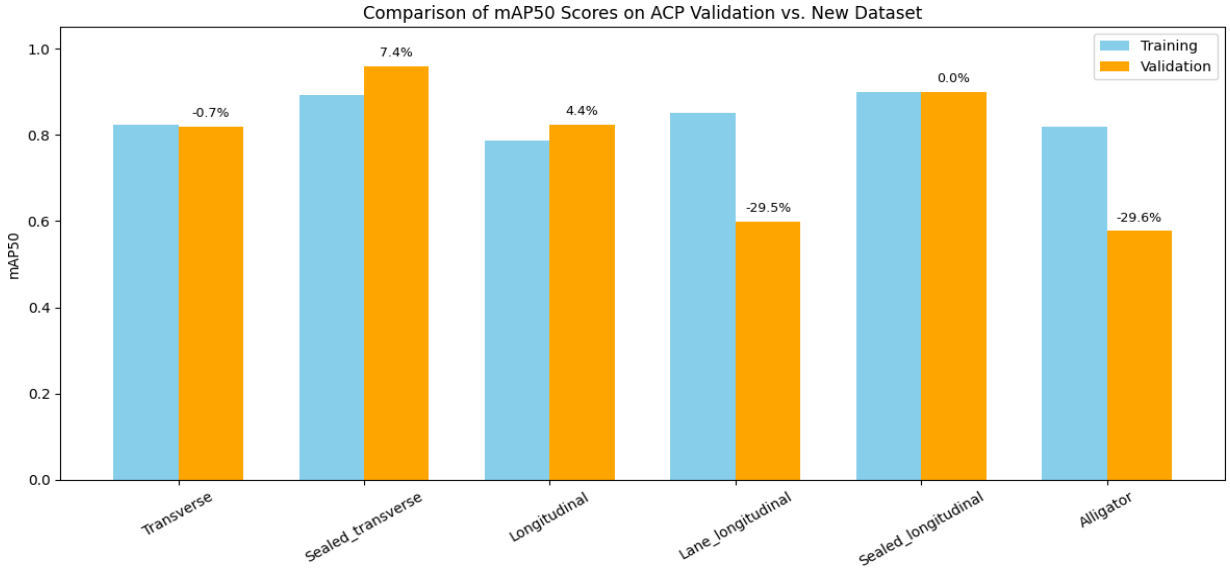


Figure 7.20 Comparison Of mAP50 scores on ACP validation and real-world datasets

Most distress types show moderate generalization behavior. Sealed transverse and longitudinal cracks exhibit positive gains of 7.4% and 4.4%, respectively. While these improvements may suggest effective feature transfer, they are more likely influenced by the specific distribution or visual characteristics of this new dataset, which may have favored detection of these distress

types. Sealed longitudinal cracks demonstrate stable performance, with no change observed between training and test datasets.

In contrast, lane longitudinal cracks and alligator cracking experience substantial drops of 29.5% and 29.6% in mAP50, respectively, highlighting their susceptibility to domain shift. These types may suffer from visual ambiguity, contextual dependence, or overfitting to features present in the training data but absent in the new domain. Transverse cracks remain relatively stable, with only a 0.7% drop, indicating good generalizability.

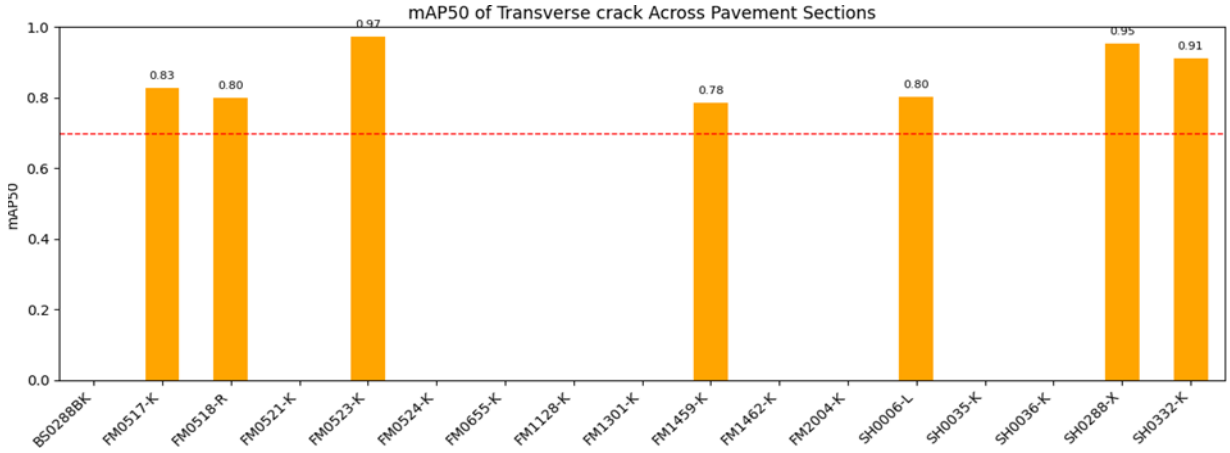


Figure 7.21 The mAP50 scores of transverse cracks across ACP pavement sections

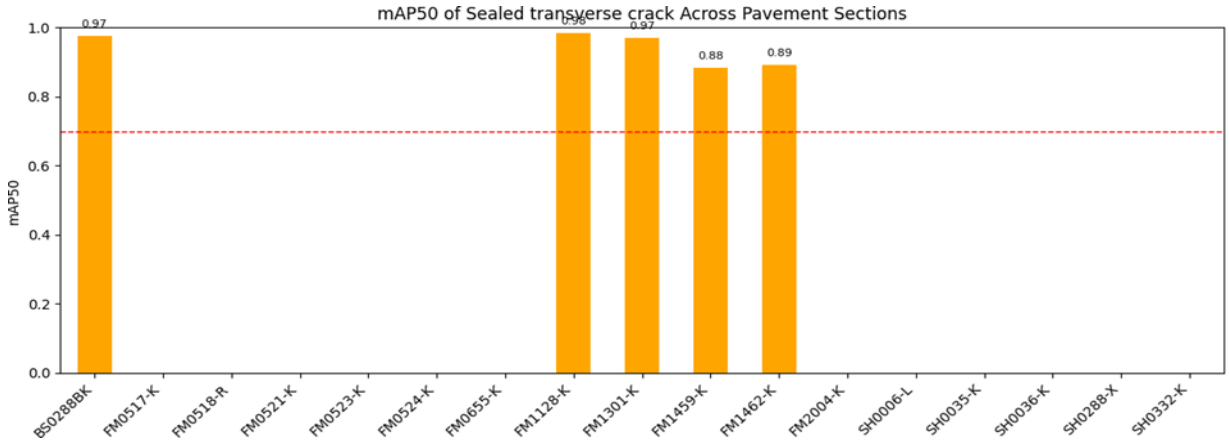


Figure 7.22 The mAP50 scores of sealed transverse cracks across ACP pavement sections

Figures 7.21 to 7.25 present the mAP50 scores for five major ACP distress classes across the selected pavement sections in Brazoria County. Only distress types with more than 50 instances are included in these figures; empty bars indicate that either the distress type is not present in the section, or the number of instances is below the 50-instance threshold and thus excluded from analysis. The results show consistently strong model performance on transverse, sealed transverse, and sealed longitudinal cracks, with all observed mAP50 values exceeding 0.7. Notably, sealed transverse and sealed longitudinal cracks achieve particularly high scores (all above 0.8 across all relevant sections), indicating that the model detects these distress types with high confidence and robustness in new environments.

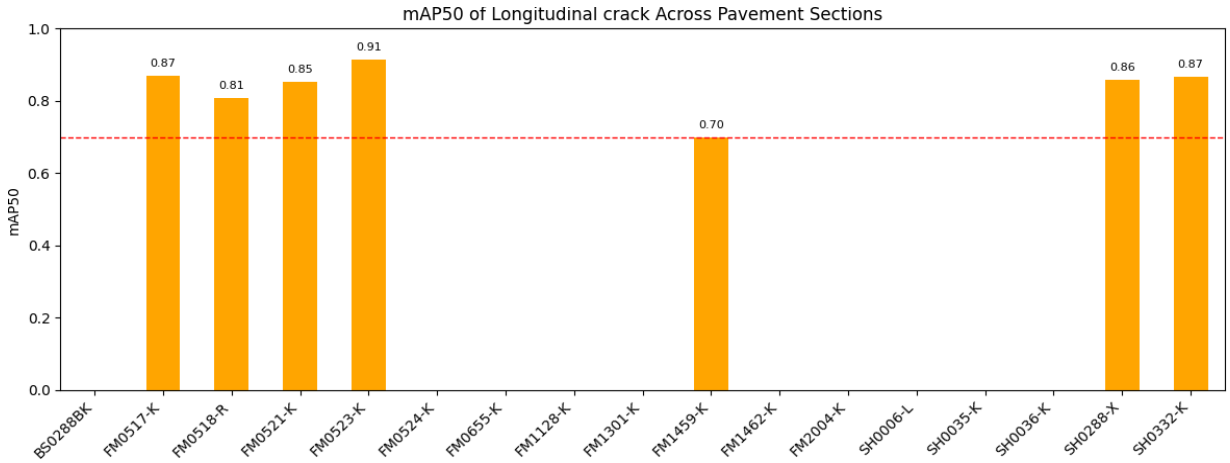


Figure 7.23 The mAP50 scores of longitudinal cracks across ACP pavement sections

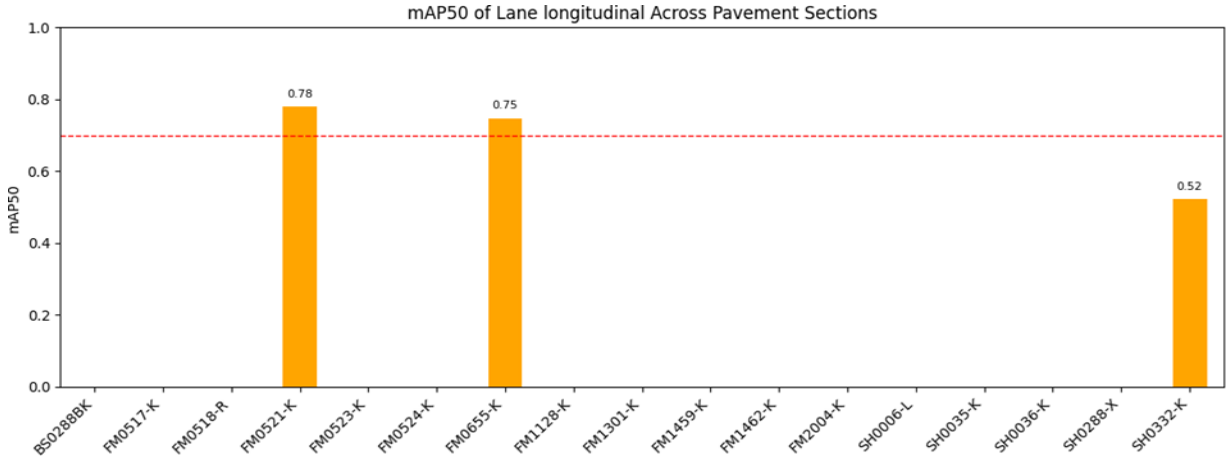


Figure 7.24 The mAP50 scores of lane longitudinal cracks across ACP pavement sections

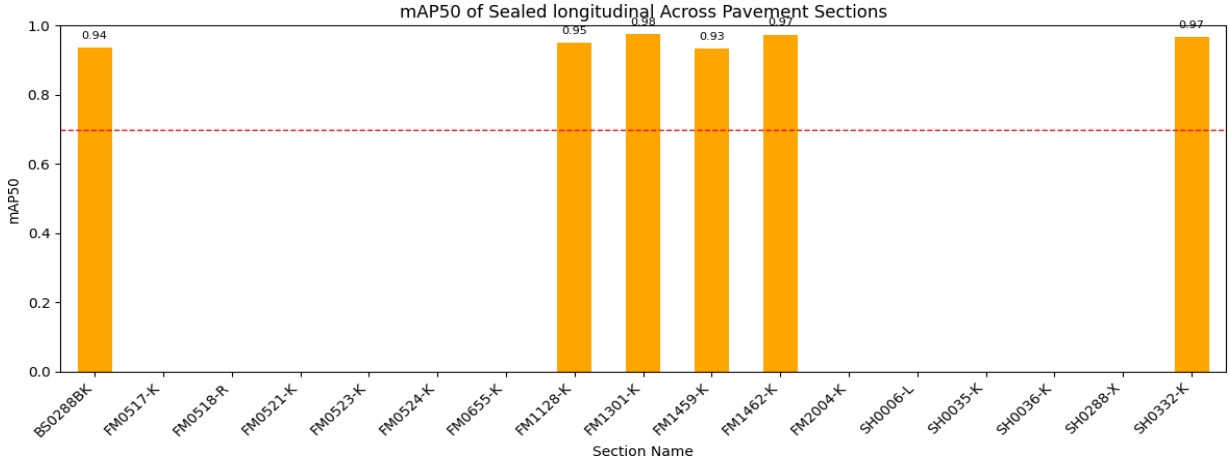


Figure 7.25 mAP50 scores of sealed longitudinal cracks across ACP pavement sections

For longitudinal cracks, which were detected in seven pavement sections (Figure 7.23), the model achieved mAP50 scores above 0.8 in six cases, with one section (FM1495-K) showing slightly lower performance at 0.70. This suggests generally strong generalization for this class, with some sensitivity to localized surface characteristics. Figure 7.26 shows detection results of longitudinal cracks on the FM1495-K section, where most cracks are sealed and appear as high-contrast features in the intensity image. These sealed cracks are consistently detected by the model, reflecting its ability to recognize the distinctive visual patterns of sealing material. However, an unsealed longitudinal crack located at the bottom left of the image is not detected by the model. One possible reason for this omission is that the surrounding high-contrast sealed cracks create a visually complex background that may obscure the more subtle appearance of the unsealed crack. Such background conditions, dominated by dense, visually prominent sealed cracks, may not be well represented in the training dataset, leading to reduced model sensitivity in these scenarios.

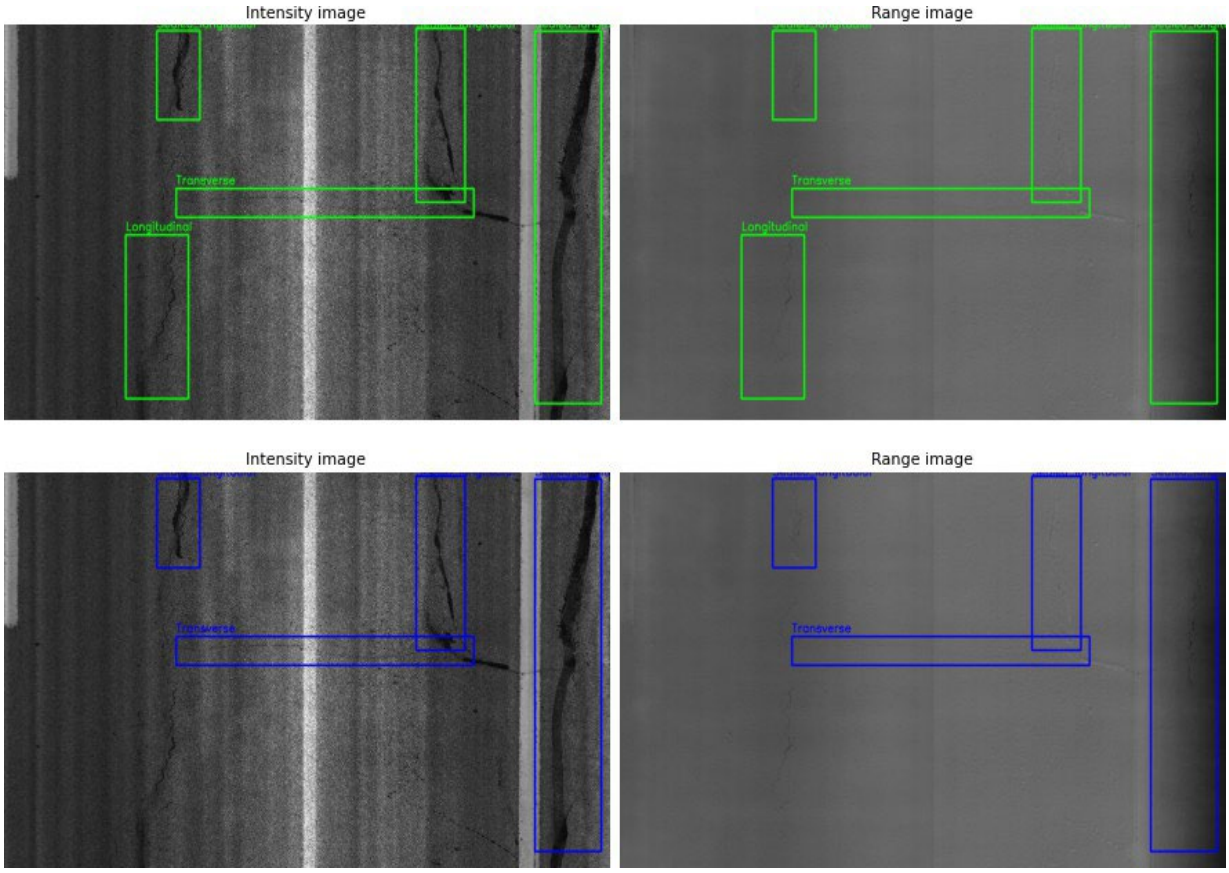


Figure 7.26 Detection sample of longitudinal cracks on FM1495-K section

Lane longitudinal cracks were identified in only three pavement sections, and their detection performance shows more variability. Two sections achieved scores near or above the 0.75 mark, while one section (SH0332-K) had a substantially lower mAP50 score of 0.52, indicating less reliable detection for this distress type. Figure 7.27 presents detection results of lane longitudinal cracks on the FM1495-K section, comparing the ground-truth annotations (top row) with model predictions (bottom row) in both intensity and range images. The model classifies the lane longitudinal cracks on the right as general longitudinal cracks. This misclassification likely stems

from the high visual similarity between the two distress types, both of which typically appear as continuous, narrow vertical cracks with similar contrast and shape. Because these features are largely shared, and the dataset contains relatively few labeled examples of lane longitudinal cracks, the model may not have learned sufficient discriminative cues to differentiate them. This confusion has already been reflected with the validation dataset and needs to be further addressed.

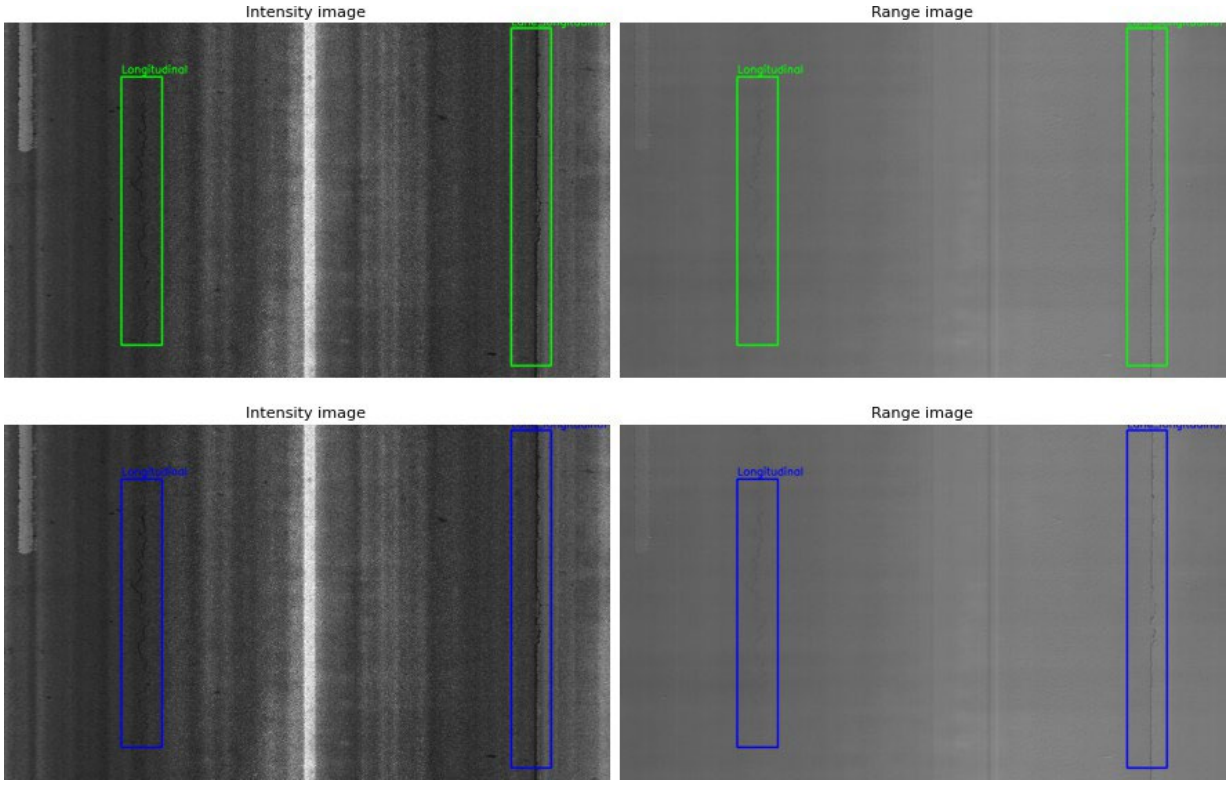


Figure 7.27 Detection sample of lane longitudinal cracks on FM1495-K section

Overall, these section-level evaluations indicate that the model generalizes well across ACP pavement sections for most distress types, especially those with consistent and distinguishable visual features. However, performance for distress types with more complex or ambiguous appearances, such as the lane longitudinal cracking, may be more susceptible to local variation and warrant further investigation.

7.3.4 JCP

Table 7.11 and Figure 7.28 collectively evaluate the model's performance on the JCP dataset from Brazoria County, focusing on detection accuracy for various distress types. The table presents class-wise performance metrics computed over 1,308 images and 3,305 total instances. To ensure statistical significance, only classes with more than 50 instances were considered for the figure comparison.

Table 7.11 Detection performance of the model on the JCP dataset (Brazoria County)

Class	Images	Instances	P	R	mAP50	mAP50-95
All	1,308	3,305	0.557	0.440	0.483	0.271
Failed joint	1,308	25	0.454	0.640	0.648	0.418
Corner break	1,308	9	0.480	0.333	0.465	0.146
Punchout	1,308	2	0.243	0.500	0.557	0.354
Asphalt patch	1,308	53	0.320	0.0755	0.111	0.0419
Popout	1,308	2	1.000	0.000	0.0438	0.0193
Longitudinal crack	1,308	85	0.509	0.294	0.337	0.181
Sealed longitudinal	1,308	2	0.000	0.000	0.0155	0.012
Concrete patch	1,308	63	0.447	0.619	0.506	0.285
Transverse crack	1,308	141	0.783	0.567	0.718	0.407
Joint crack	1,308	1,086	0.947	0.908	0.951	0.498
Slab edge	1,308	1,837	0.940	0.898	0.963	0.616

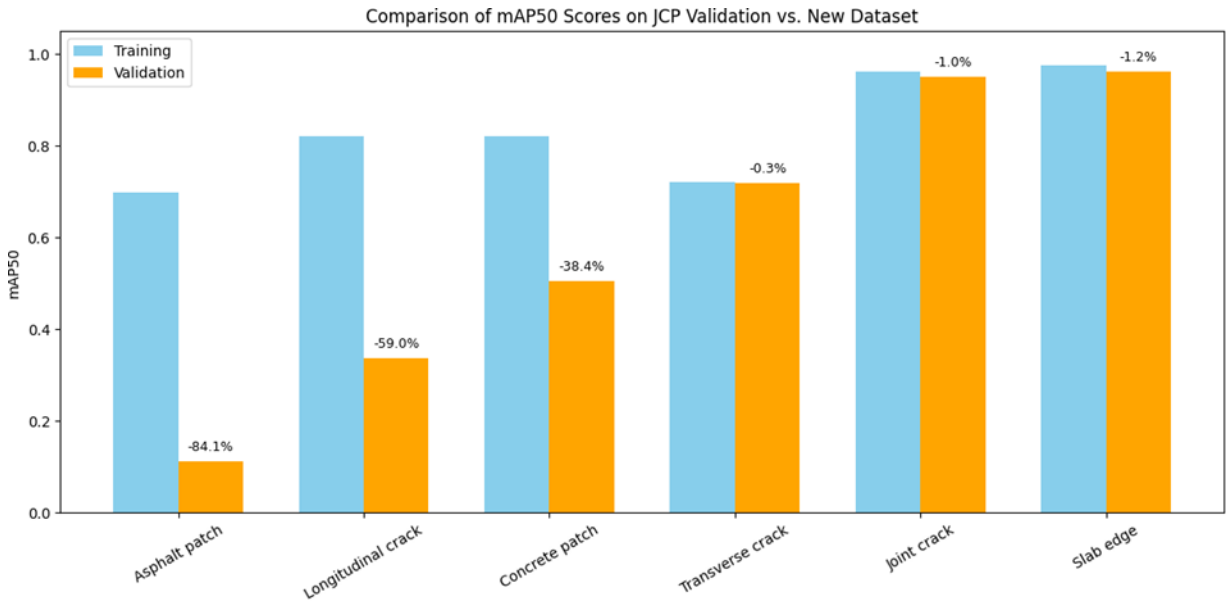


Figure 7.28 Comparison of mAP50 scores on JCP validation and new datasets

According to Table 7.11, it is evident that the model achieves high performance on the Joint crack (mAP50 = 0.951) and Slab edge (mAP50 = 0.963), both of which are supported by a large number of instances (1,086 and 1,837 respectively). In contrast, classes with fewer than 50 instances, such as the Sealed longitudinal, Popout, and Corner break, exhibit low detection metrics.

The bar chart in Figure 7.28 visualizes the generalization ability of the model by comparing mAP50 scores between the JCP validation dataset and a new, unseen dataset. Notably, classes like the Asphalt patch, Longitudinal crack, and Concrete patch show substantial drops of 64.1%,

59.0%, and 38.4% in mAP50 on the new dataset, respectively, indicating weak generalization, possibly due to limited variation in training data or visually complex backgrounds. Conversely, the Joint crack and Slab edge maintain strong performance across both datasets, with minimal mAP50 drops of 1.0% and 1.2%, respectively, underscoring their consistent model robustness.

Figures 7.29 to 7.32 present the mAP50 performance of the model for different distress types across JCP pavement sections, providing insights into the generalizability of the model at the section level.

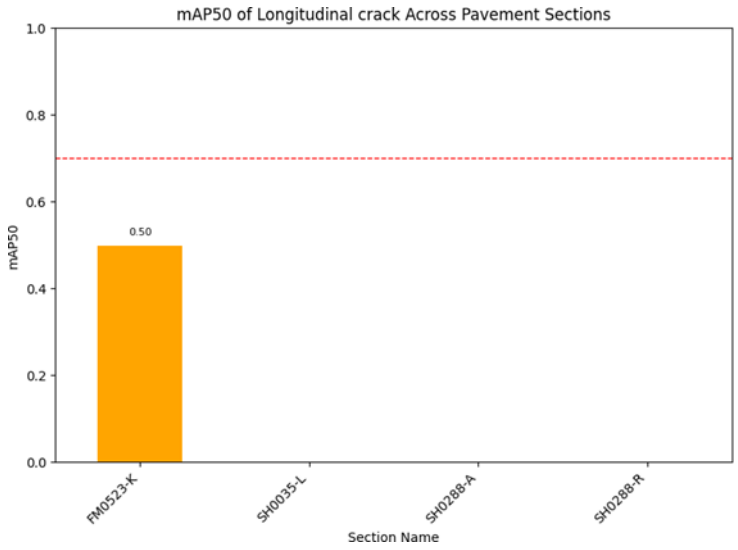


Figure 7.29 The mAP50 scores of longitudinal cracks across JCP pavement sections

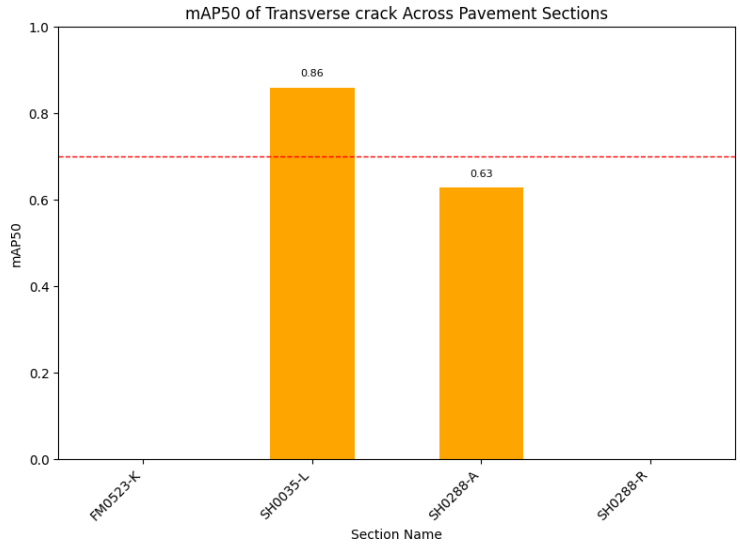


Figure 7.30 The mAP50 scores of transverse cracks across JCP pavement sections

Figures 7.31 and 7.32 demonstrate consistently strong performance for joint crack and slab edge detection, with mAP50 scores exceeding 0.90 across all four pavement sections. This indicates excellent generalization and robustness of the model for these two distress types, likely due to their strong visual patterns and high instance counts in the training data. In contrast, Figure 7.29 shows that longitudinal cracks were only present in one section (PM0325-X) with an mAP50 of 0.50, reflecting low generalization and potentially limited representation in the training dataset. This suggests the model struggles to maintain reliable detection of longitudinal cracks across diverse environments. Figure 7.30 shows transverse cracks in two sections, where the model performs very well on section SH0035-L (mAP50 = 0.86) but drops to 0.65 on SH0036-A. This variability implies inconsistent generalization and potential sensitivity to differences in surface texture or crack morphology between sections.

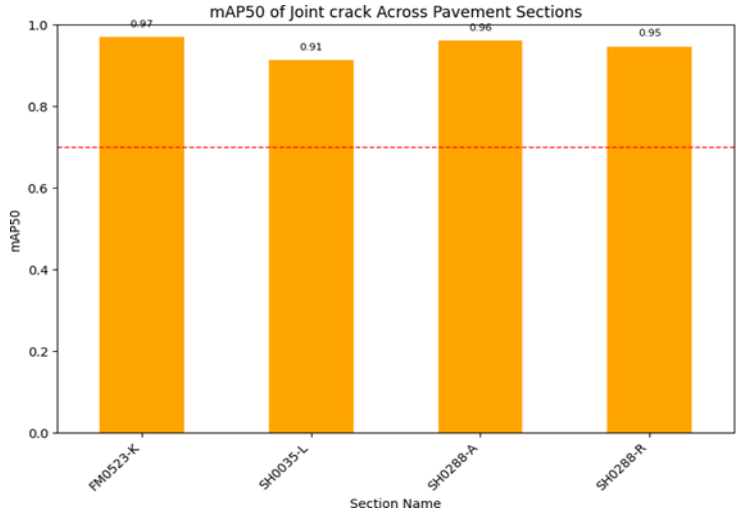


Figure 7.31 The mAP50 scores of joint cracks across JCP pavement sections

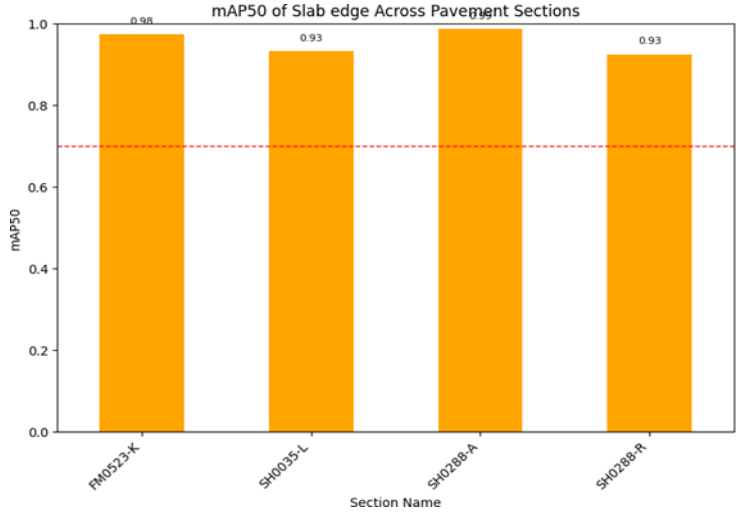


Figure 7.32 The mAP50 scores of slab edges across JCP pavement sections

Figure 7.33 illustrates the detection results of longitudinal cracks on the FM0523-K section using both intensity and range images, with ground truth annotations shown in green (top row) and model predictions in blue (bottom row). While some longitudinal cracks are correctly identified by the model, two of them are missed entirely, indicating limited recall. Figure 7.34 shows the detection results of transverse cracks on the SH0288-A section. While two lane longitudinal cracks at the edges are correctly detected, the bottom transverse crack is completely missed by the model. Although this crack has moderate contrast, it is not recognized, indicating a recall issue. A likely reason is the presence of asphalt spots scattered across the pavement, which serve as a new background not represented in the training dataset. These unfamiliar visual patterns likely introduce background noise that disrupts feature extraction, leading to reduced detection accuracy and contributing to the relatively low mAP50 score (0.65) observed for transverse cracks in this section.

Overall, the analysis suggests that while the model performs well on dominant and well-represented classes, its performance deteriorates significantly for distress types with fewer or more visually complex examples, highlighting the need for more diverse and balanced training data to enhance generalization.

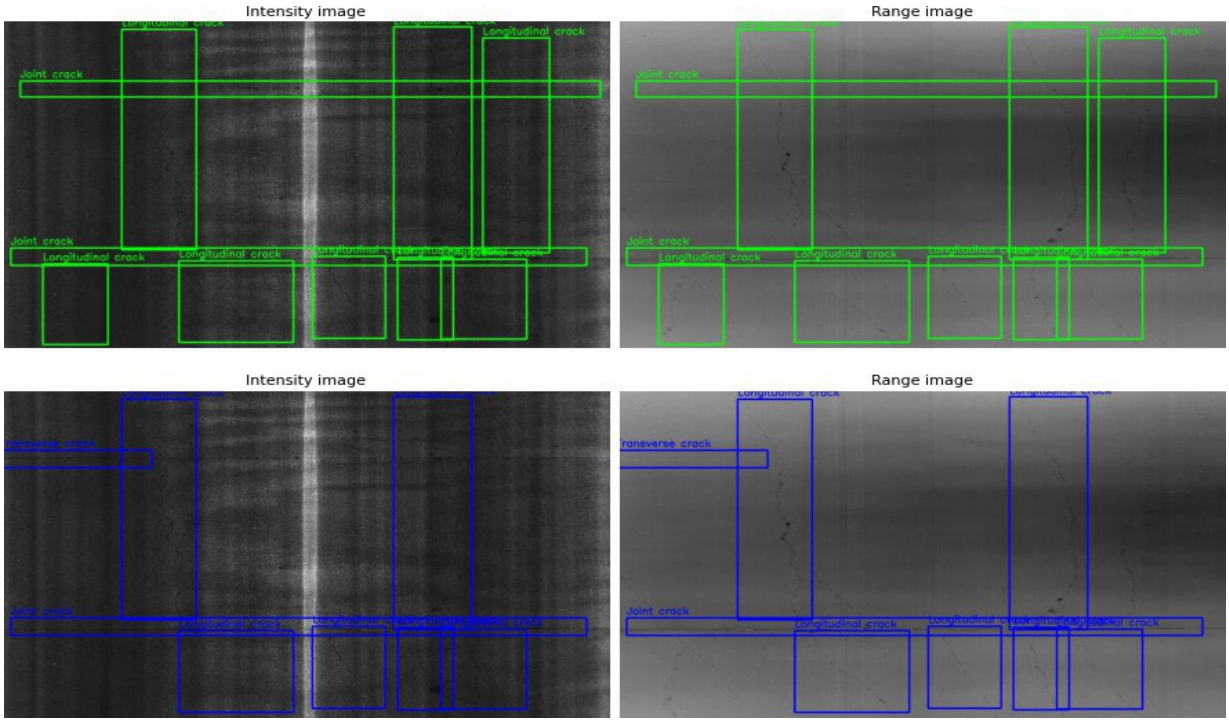


Figure 7.33 Detection sample of longitudinal cracks on FM0523-K section

7.3.5 CRCP

Table 7.12 and Figure 7.35 show the detection performance and generalization ability of the model on the CRCP dataset from Brazoria County. The table summarizes class-wise performance metrics across 2,308 images and 3,193 total instances. To ensure statistical reliability, only classes with more than 50 instances, namely longitudinal cracks, transverse

cracks, and spalled transverse cracks, are considered in the figure for comparison between training and validation datasets.

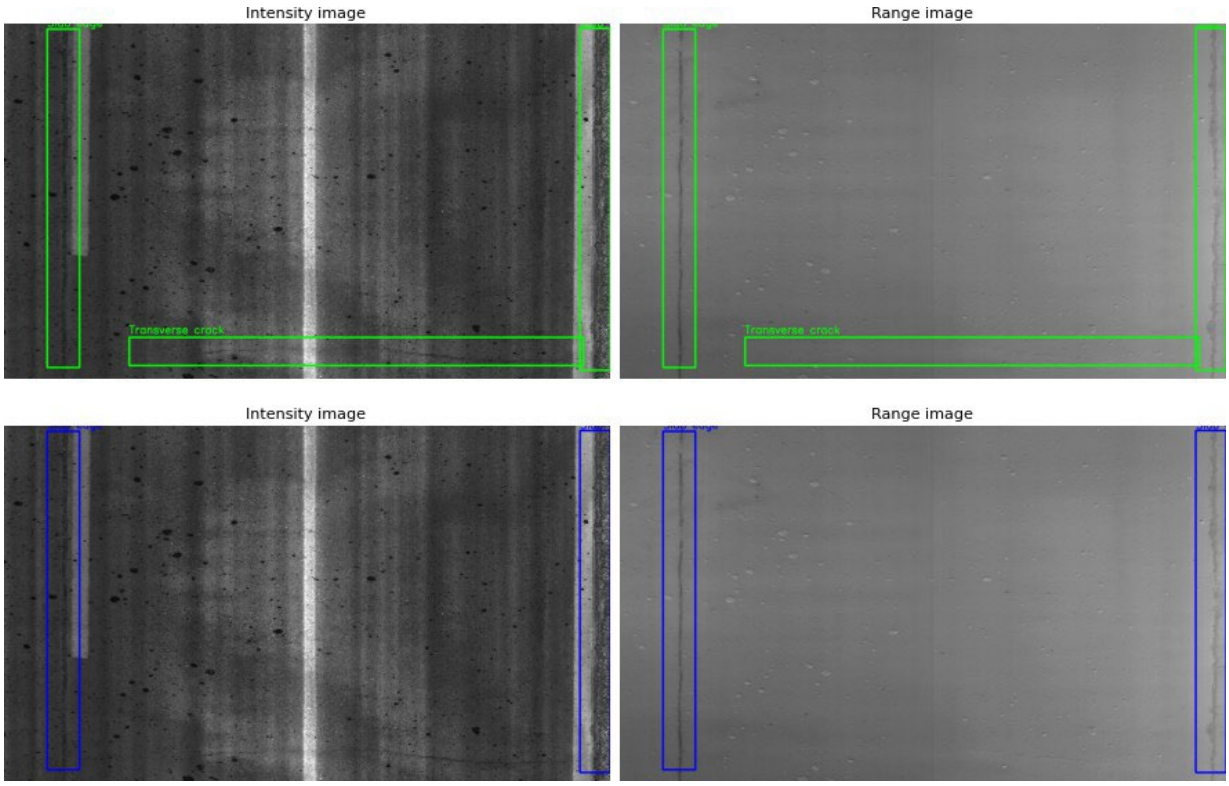


Figure 7.34 Detection sample of transverse cracks on SH0288-A section

Table 7.12 Detection performance of the model on the CRCP dataset (Brazoria County)

Class	Images	Instances	P	R	mAP50	mAP50-95
All	2308	3193	0.672	0.558	0.594	0.360
Longitudinal crack	2308	58	0.402	0.0469	0.0457	0.227
Sealed longitudinal crack	2308	5	0.434	0.800	0.678	0.481
Punchout	2308	2	0.473	1.000	0.745	0.422
Asphalt patch	2308	6	0.376	0.667	0.519	0.345
Concrete patch	2308	44	0.872	0.386	0.413	0.325
Transverse crack	2308	2838	0.904	0.702	0.873	0.487
Sealed transverse crack	2308	9	1.000	0.000	0.524	0.393
Spalled transverse crack	2308	231	0.917	0.861	0.946	0.488

According to Table 7.12, transverse crack and spalled transverse crack stand out with strong performance: both classes show high precision and recall, and mAP50 scores of 0.873 and 0.946,

respectively. In contrast, longitudinal cracks, despite having 58 instances, show very poor recall (0.0469) and a low mAP50 of 0.0457, indicating that the model rarely detects them correctly.

Figure 7.35 highlights the model's generalization across datasets. For longitudinal cracks, there is a dramatic 75.9% drop in mAP50, signaling poor generalization and potential underrepresentation in the training data. Conversely, transverse cracks and spalled transverse cracks exhibit positive generalization, with 4.6% and 26.0% improvements in mAP50, respectively, on the new dataset. These results suggest that for well-represented and visually consistent classes, the model generalizes effectively, while for classes with lower frequency and higher visual variability, such as longitudinal cracks, the model struggles to transfer learned patterns to unseen data. This reinforces the need for more comprehensive and balanced training samples to improve robustness and reduce class-specific performance degradation.

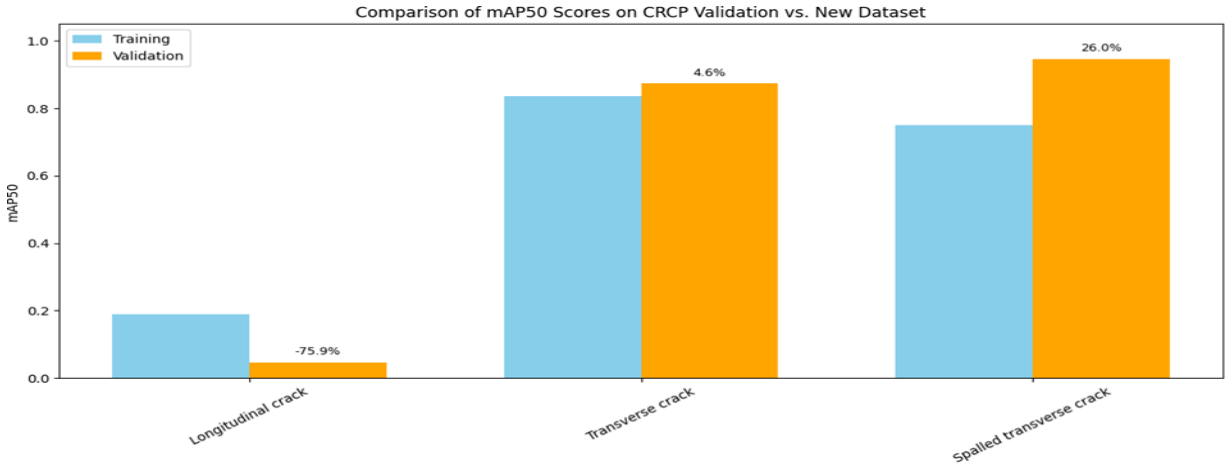


Figure 7.35 Comparison of mAP50 scores on CRCP validation and real-world datasets

Figures 7.36 and 7.37 evaluate the model's mAP50 scores for transverse and spalled transverse cracks across individual CRCP pavement sections, highlighting its section-level performance and generalization. As shown in Figures 7.36, transverse cracks are consistently detected with high accuracy, with mAP50 scores exceeding 0.79 in all eight sections, and reaching up to 0.97 in SH0035-K. This indicates the model performs reliably across diverse environments for this class. Similarly, Figure 7.37 shows the model achieves a very high mAP50 of 0.95 for spalled transverse cracks in SH0035-K, although this distress type is only present in one section at a statistically valid instance count.

Together, these results confirm very strong model performance and generalization for both transverse and spalled transverse cracks, the only two CRCP distress types with sufficient instance numbers for robust evaluation. The consistently high mAP50 scores across multiple sections (for transverse cracks) further support the model's robustness to spatial and environmental variability. Other distress classes, such as longitudinal cracks, punchouts, and asphalt patches, are excluded from these figures due to insufficient instance counts, making it inappropriate to draw conclusions about their generalization capability.

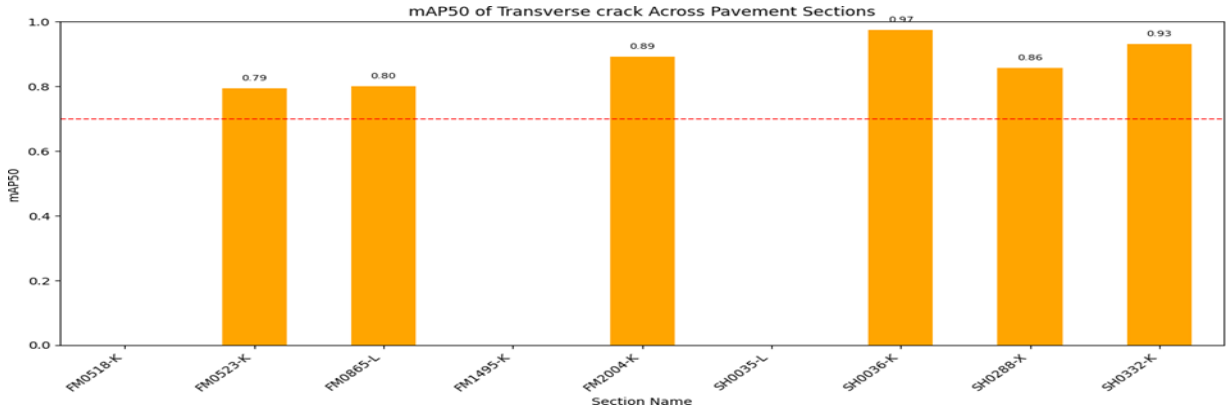


Figure 7.36 mAP50 scores of transverse cracks across CRCP pavement sections

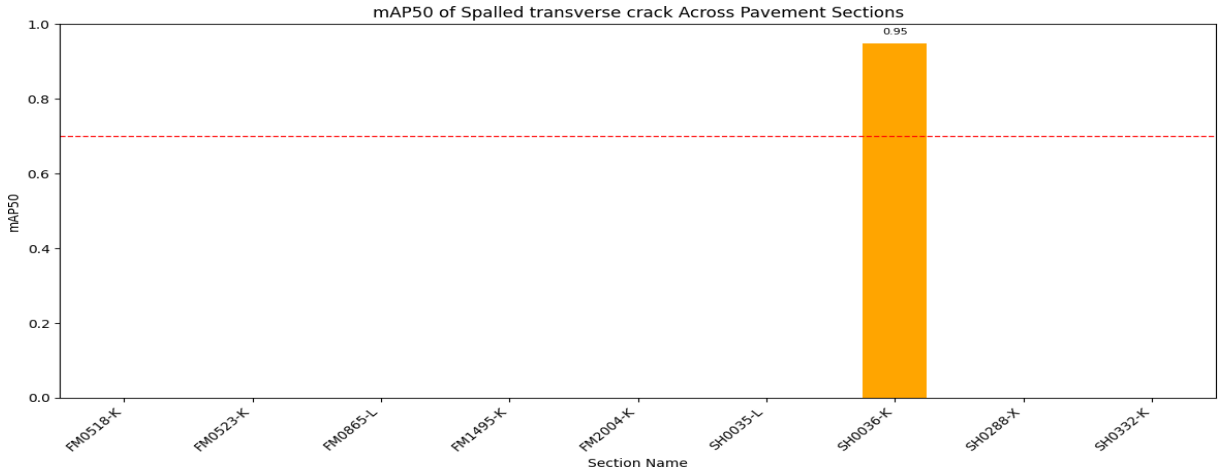


Figure 7.37 mAP50 scores of spalled transverse cracks across CRCP pavement sections

7.4 Development of New AI Models for Generalization and Robustness

7.4.1 New Development on the model of automated pavement distress detection

To address persistent bottlenecks in pavement distress detection, namely long-tailed class distributions, low contrast, and slender/irregular geometries, a minimal-intrusion upgrade was employed while keeping labels and evaluation protocol unchanged. We migrated from the YOLOv5 to YOLOv8's anchor-free, decoupled head to reduce anchor matching dependence and improve sensitivity to fine cracks and adopted a rare-class–aware training recipe (inverse-frequency weighting with focal-style loss plus rarity-aware sampling) to reallocate gradient toward scarce, hard positives without degrading head classes. Additionally, we fused 2D/3D images as inputs (no relabeling) to inject geometric cues that improve separability in low-contrast scenes. This recipe yields stable, reproducible gains on JCP/CRCP, confirming its generality. However, on ACP we observe a single-class limitation (pothole learnability), reflecting scene- and category-specific boundaries. The subsequent sections and tables first

report dataset-level comparisons and key class outcomes, then analyze error modes and sensitivity to thresholds/sampling, culminating in targeted recommendations.

7.4.1.1 ACP Dataset

To overcome ACP's long-tailed distribution and channel-dependent distress responses, two-channel, class-aware augmentation pipeline that operates jointly on intensity (2D) and range (3D) images was created, inspired by the observation that different pavement distresses rely on different channel cues. Unlike conventional augmentation that applies identical operations to all channels and all classes, our policy is asymmetric and targeted:

- geometric alignment is preserved between channels,
- but pixel-level augmentation is differentiated across channels and across classes, so that augmentation amplifies the right cues instead of introducing noise.

Examples include:

- longitudinal-type cracks rely more on intensity continuity, so we enhance contrast and avoid aggressive range noise;
- potholes rely strongly on 3D depression cues, so we emphasize range-channel multiplicative noise and CLAHE while keeping intensity changes mild;
- sealed cracks require balanced augmentation to preserve weak boundaries.

Table 7.13 Detection performance of the YOLOv5 model with data augment model on the ACP validation dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	1039	2043	0.814	0.778	0.858	0.534
Transverse	1039	222	0.742	0.815	0.875	0.476
Joint	1039	84	0.926	0.889	0.956	0.528
Sealed_transverse	1039	327	0.907	0.881	0.95	0.56
Longitudinal	1039	306	0.749	0.781	0.85	0.511
Lane_longitudinal	1039	251	0.803	0.829	0.869	0.575
Sealed_longitudial	1039	629	0.878	0.893	0.954	0.597
Block	1039	50	0.749	0.74	0.829	0.676
Alligator	1039	143	0.822	0.79	0.891	0.591
Pothole	1039	31	0.75	0.387	0.547	0.291

Table 7.13 shows the results of newly developed model on ACP validation dataset. Compared to Table 7.2 with the same ACP validation dataset (1,039 images and 2,043 instances), the proposed two-channel and class-aware augmentation strategy brings consistent improvements over the previous YOLOv5 model. The enhanced model increases the overall mAP50 from 0.812 to 0.858 (+4.6%) and mAP50-95 from 0.414 to 0.534 (+12.0%), while maintaining comparable precision and recall. The most significant gains come from rare or structurally fragile categories,

such as Alligator (+7.1%), Longitudinal (+6.2%), Sealed transverse/longitudinal (+5.0–5.5%), and Pothole (+6.7%). For dominant classes, such as Lane longitudinal and Joint, the model stabilizes high performance and achieves mAP50 scores of 0.869 and 0.956, respectively. These gains arise from two key factors: (i) channel-decoupled augmentation, which enhances Intensity (2D) and Range (3D) differently to emphasize class-specific visual cues, and (ii) rarity-aware enhancement, which selectively increases gradient contribution from scarce distress patterns without harming head-class convergence. Together, the strategy effectively boosts the learnability of low-contrast cracks and long-tailed categories on ACP.

7.4.1.2 JCP Dataset

On the JCP validation dataset (1,550 images for 3,392 instances), the newly developed YOLOv8 model improves 6.8% (P), 14.4% (R), 13.9% (mAP50), and 27.0% (mAP50-95) from the previous YOLOv5 model (see Tables 7.14 and 7.4). The YOLOv8 model (Biswas et al., 2026) makes a dramatic increase on rare classes, such the Punchout, Failed concrete patch, Concrete patch, and Transverse crack. Regarding dominant classes, the Slab edge and Joint crack achieve a mAP50 of 0.974 and 0.947. This is because (i) re-allocating learning signal via class-balanced focal loss and rarity-aware sampling, and (ii) geometric cues from Range that help detect subtle, low-contrast distresses. On the other hand, the model is still struggling at popouts (only 10 instances).

Table 7.14 Detection performance of the new model on the JCP validation dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	1550	3392	0.814	0.714	0.763	0.466
Failed joint	1550	106	0.715	0.726	0.782	0.363
Corner break	1550	72	0.708	0.694	0.742	0.405
Punchout	1550	23	0.579	0.739	0.682	0.45
Asphalt patch	1550	84	0.89	0.677	0.798	0.558
Failed concrete patch	1550	8	0.494	0.375	0.464	0.336
Popout	1550	10	1.0	0.0	0.007	0.00214
Longitudinal crack	1550	192	0.83	0.815	0.886	0.498
Sealed longitudinal	1550	90	0.914	0.944	0.906	0.565
Concrete patch	1550	154	0.876	0.825	0.889	0.634
Transverse crack	1550	103	0.819	0.777	0.848	0.462
Joint crack	1550	831	0.909	0.924	0.947	0.509
Sealed transverse crack	1550	12	0.911	0.852	0.937	0.532
Slab edge	1550	1707	0.931	0.929	0.974	0.693

7.4.1.3 CRCP Dataset

There are 2,753 instances on 1,156 images, after training, the newly upgraded YOLOv8 model improves 3.9% (P), 6.5% (R), 4.6% (mAP50), and 10.7% (mAP50–95) compared to the

YOLOv5 model (see Tables 7.15 and 7.6). The largest gains of mAP50 occur on the Longitudinal crack 179%), Concrete patch (16%), and Sealed longitudinal crack (14%), while dominant classes remain high (e.g., Asphalt patch is 0.953). A few categories show mild decreases (e.g., Spalled transverse, Transverse and Punchout drop 0.085, 0.057, and 0.083, respectively), indicating potential for targeted sampling or threshold tuning. Overall, the CRCP results confirm the same simple recipe yields consistent, measurable increase across pavements with different distress profiles, supporting its use as a practical upgrade.

Table 7.15 Detection performance of the new model on the CRCP validation dataset

Class	Images	Instances	P	R	mAP50	mAP50-95
All	1,156	2,753	0.740	0.741	0.773	0.425
Longitudinal crack	1,156	73	0.624	0.384	0.530	0.223
Sealed longitudinal crack	1,156	21	0.609	0.920	0.857	0.503
Punchout	1,156	18	0.746	0.722	0.746	0.314
Asphalt patch	1,156	226	0.904	0.957	0.953	0.634
Concrete patch	1,156	31	0.834	0.812	0.830	0.596
Transverse crack	1,156	2,259	0.842	0.667	0.778	0.425
Sealed transverse crack	1,156	77	0.765	0.883	0.824	0.414
Spalled transverse crack	1,156	48	0.597	0.583	0.666	0.295

7.4.2 Add new training datasets from other vendors to train models for generalization

In this study, we conducted experiments using single-source data (e.g., from one vendor) and two-source data (e.g., from two vendors) to train the deep learning models and improve the models' generalization. The 2D/3D imagery is used in this part (Bai et al., 2026). Two vendor datasets with identical distress definitions but different formats are included. Both datasets use bounding boxes to label distresses for ACP and CRCP. The dataset (Dataset 7150) that was used in Section 7.2 has a resolution of 1,536×900 (pixels) covering a pavement surface with a width of 4.3 m and a length ranging from 1.0 to 8.1 m, depending on the driving speed. Meanwhile, a new dataset from our NSF project (Dataset NSF), which can cover a 4.3×14.3-m region of a pavement with an image size of 4,096×2,048 (pixels), is introduced into this study. Due to optical devices, lighting, and road types, visual disparities can be observed on the examples of two datasets (see Figure 7.38). Dataset 7150 comprises 6,943 and 5.791 images of ACP and CRCP, and Dataset NSF includes a total of 5,823 and 7,699 images for ACP and CRCP datasets, respectively. Both datasets include nine distress types as presented in Section 7.3.

The YOLOv11 model is utilized to do this study. The objective is to address a knowledge gap in automated pavement distress detection: Given available datasets, what represents the best strategy for effectively leveraging two distinct datasets, independent of deep learning model selection when the strategies such as transfer learning, knowledge distillation, or data merging are used?

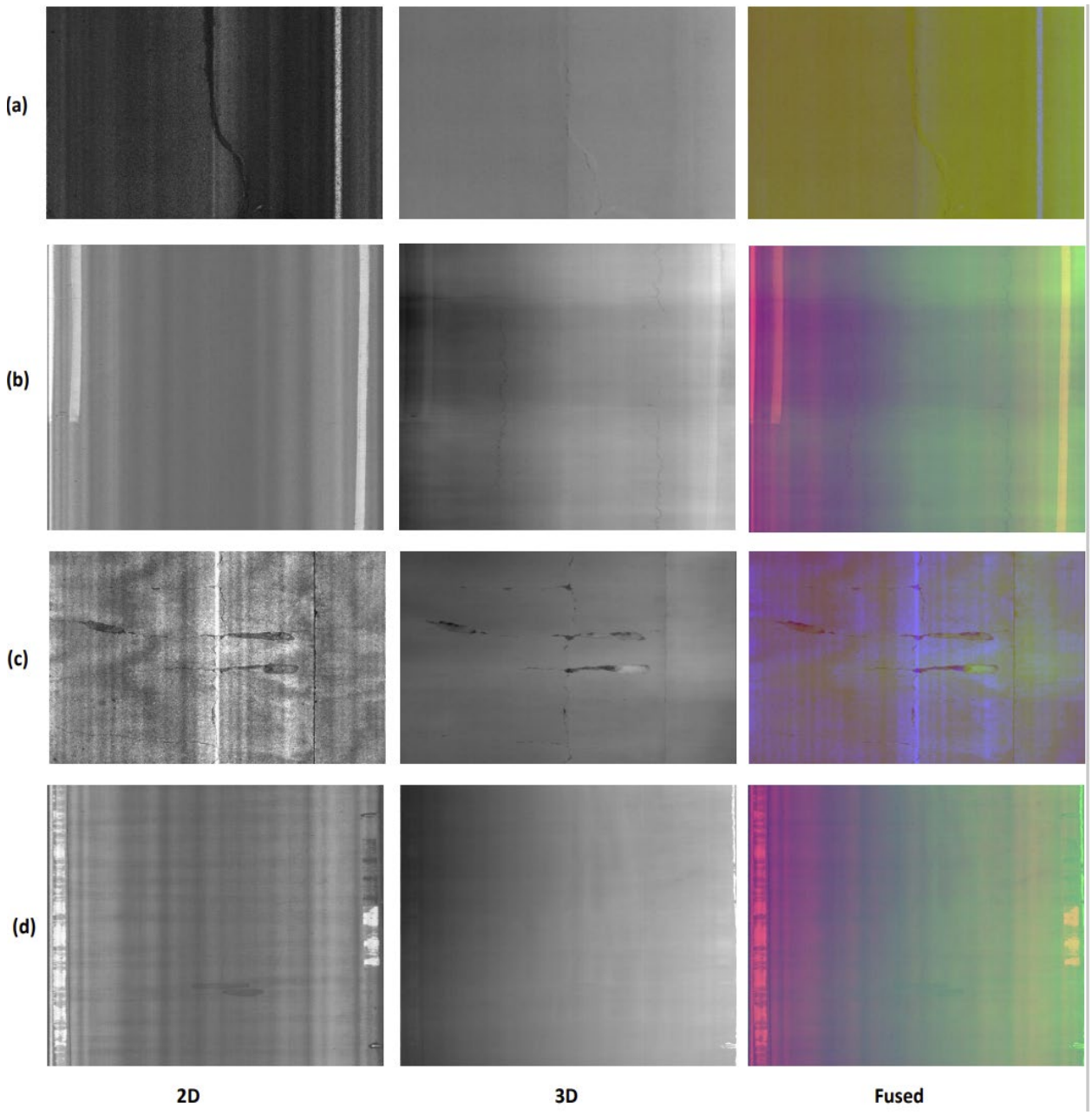


Figure 7.38 Two different datasets collected in this research. (a) and (c) , (b) and (d) are 2D, 3D, and fused images of ACP and CRCP for Datasets 7150 and NSF, respectively

Our findings show that models refined through either transfer learning or knowledge distillation with a new dataset cause severe performance degradation when applied to previous datasets, and achieve near-zero scores in precision, recall, and mAP50 during cross-validation. However, the comprehensive model trained with both datasets can obtain the same overall performance as the models trained by an individual dataset over the same validation dataset see Table 7.16). For a better generalization, the two datasets should be merged and used for the finetuning of a new model.

7.5 Discussion and Summary

7.5.1 Discussions

The analysis reveals a clear correlation between sample size and detection performance. Distress classes with more than 1000 training instances almost universally achieved strong results, with mAP50 scores exceeding 0.8. This includes the Longitudinal, Sealed longitudinal, Transverse, and Sealed transverse cracks in the ACP dataset; Joint and Slab edge in JCP; and Transverse crack in CRCP.

The only exception is Lane longitudinal crack in ACP, which, despite having over 1000 samples, failed to reach comparable performance. This underperformance may be attributed to frequent misclassification as standard longitudinal cracks, likely due to shared geometric characteristics and limited inter-class variation in the training data.

Interestingly, the Spalled transverse crack in CRCP stands out as the only distress type that achieved an mAP50 above 0.8 despite having fewer than 1000 training instances (167 samples). This anomaly could be explained by the distinct and visually salient features of spalling, which make it easier for the model to learn, or by the possibility that Spalled transverse cracks exhibit low intra-class variation, leading to easier generalization despite the limited data. This case highlights that while larger datasets generally yield better performance, class-specific factors such as visual distinctiveness and consistency also play a crucial role in model effectiveness. However, to validate that for each distress class, more data will still be needed.

7.5.1.1 Variety coverage

While sample size plays a critical role in model performance, the results also highlight the importance of variety coverage, the degree to which training data captures the visual diversity of each distress type. Several classes with ample sample sizes still show significant performance degradation when applied to validation and test datasets, suggesting insufficient representation of their visual variability during training. For instance, the Asphalt patch, Longitudinal crack, and Concrete patch of JCP, despite having fair numbers of training instances, experience mAP50 drops exceeding 40% on test data. These drops suggest that their real-world appearances may vary substantially across pavement sections or regions, and the training dataset fails to capture this diversity.

Conversely, the Spalled transverse crack in the CRCP dataset achieved high performance on the test dataset (mAP50 = 0.946) despite a relatively small training sample (167 instances). This implies that the class may exhibit consistent visual patterns, or that its features are especially salient and easily distinguishable, reducing the need for broad variety coverage. Additionally, some classes like Lane longitudinal crack in ACP, despite adequate sample size, suffered from frequent confusion with similar classes (e.g., Longitudinal crack), reflecting insufficient distinction or diversity in the training examples to help the model learn subtle differences.

Table 7.16 Performance of models trained with individual and combined datasets over validation dataset

Models	ACP			CRCP		
	P	R	mAP50	P	R	mAP50
YOLOV11 model trained and validated with Dataset 7150	0.76	0.71	0.78	0.83	0.62	0.75
YOLOV11 model trained and validated with Dataset NSF	0.61	0.58	0.59	0.63	0.46	0.48
YOLOV11 model trained with two datasets but validated on Dataset 7150	0.77	0.66	0.75	0.80	0.62	0.68
YOLOV11 model trained with two datasets but validated on Dataset NSF	0.63	0.56	0.59	0.60	0.42	0.46
YOLOV11 model trained and validated on Datasets NSF and 7150	0.67	0.62	0.67	0.65	0.51	0.56

Findings from this pilot study emphasize that model robustness depends not only on the quantity of training data but also on its ability to capture the full spectrum of visual variability within each distress type. Future dataset development should prioritize not just increasing sample sizes but also ensuring that samples span a broad range of pavement textures, distress severities, and contextual variations to improve generalization across diverse field conditions.

7.5.1.2 Model Readiness for the YOLOv5 Model (Deliverable Software)

The pilot study demonstrates strong detection capability for distress types with abundant training data, as reflected by consistently high mAP50 scores. Notably, this includes longitudinal and sealed longitudinal cracks, as well as transverse and sealed transverse cracks in ACP; joint and slab edge distresses in JCP; and both transverse and spalled transverse cracks in CRCP. These results suggest that the model has effectively learned the visual patterns of commonly occurring distresses. However, for distress types with limited sample sizes, the model exhibits poor generalization, as evidenced by its underperformance on the test data from Brazoria County. This likely stems from overfitting during training, where the model adapts too closely to the scarce examples seen, failing to generalize to new instances in varied conditions.

Table 7.17 summarizes the readiness of individual distress types across pavement types based on the pilot study. Aside from the classes showing strong performance, the remaining distress types either suffer from poor generalization or could not be evaluated due to insufficient instances in the pilot study.

Table 7.17 Summary of distress types based on model readiness for implementation

Pavement types	Ready to be implemented	Needs further test/improvement
ACP	Transverse, Sealed transverse, Longitudinal, and Sealed longitudinal cracks	Block crack, Alligator crack, Patch/Pothole
JCP	Joint, Slab edge	Failed joint, Corner break, Punchout, Asphalt patch, Popout, Longitudinal crack, Sealed longitudinal crack, Concrete patch, Transverse crack
CRCP	Transverse crack, Spalled transverse crack	Longitudinal crack, Sealed longitudinal crack, Punchout, Asphalt patch, Concrete patch, Sealed transverse crack

7.5.2 Analysis summary

Table 7.18 presents the mAP50 performance changes for various ACP distress types from the training to the validation and test datasets, with grey color indicating metrics based on insufficient instances. Distress types such as the Sealed transverse and Sealed longitudinal cracks exhibit minimal performance drops (within 8%), indicating strong generalization and consistent detection across datasets. In contrast, classes like the Joint and Block cracking suffer dramatic declines, with the Joint class showing a near-complete breakdown in test performance (from 0.907 to 0.0217), which may not be solid proof of model's generalization due to limited testing samples. Moderate drops are observed in the Transverse and Longitudinal cracks (13% to 18%), which, while not ideal, still reflect acceptable generalization. However, significant test performance degradation is observed for Alligator, suggesting severe overfitting likely due to limited and unrepresentative training samples. The Pothole class, with a 50% drop on validation and no test data available, underscores issues related to class imbalance and insufficient evaluation for rare types.

Table 7.18 mAP50 performance change from training to validation and test datasets for ACP

Class	Train	Val	Change (Train-Val)	Test	Change (Train-Test)
Transverse	0.936	0.824	-12%	0.818	-13%
Joint	0.907	0.933	+3%	0.0217	-98%
Sealed_transverse	0.959	0.892	-7%	0.958	0%
Longitudinal	0.963	0.788	-18%	0.823	-15%
Lane_longitudinal	0.961	0.850	-12%	0.599	-38%
Sealed_longitudial	0.975	0.899	-8%	0.899	-8%
Block	0.983	0.882	-10%	0.142	-86%
Alligator	0.983	0.823	-16%	0.577	-41%
Pothole	0.952	0.480	-50%	-	-

Table 7.19 summarizes the mAP50 performance changes across training, validation, and test datasets for JCP distress types. A notable trend is the significant drop in performance from training to validation for the majority of distress types, such as Punchout, Corner break, Popout, and Failed concrete patch, suggesting overfitting likely driven by underrepresented or imbalanced training samples. In contrast, the Joint crack and Slab edge show consistently high and stable performance across all datasets, with minimal drops (within 3%), implying that sufficient sample sizes provide a strong foundation for model generalization. On the test dataset, substantial mAP50 declines are observed for key classes like Asphalt patch, Longitudinal crack, and Concrete patch, despite adequate validation performance. These results warrant further investigation, potentially into annotation inconsistencies, regional distress pattern differences, or domain shift effects between training and test environments.

Table 7.19 mAP50 performance change across training, validation, and test datasets for JCP distress types

Class	Train	Val	Change (Train-Val)	Test	Change (Train-Test)
Failed joint	0.993	0.780	-21%	0.648	-35%
Corner break	0.896	0.577	-36%	0.465	-48%
Punchout	0.944	0.399	-58%	0.557	-41%
Asphalt patch	0.948	0.698	-26%	0.111	-88%
Failed concrete patch	0.954	0.227	-76%	-	-
D-cracking	0.079	-	-	-	-
Popout	0.581	0.027	-95%	0.0438	-92%
Longitudinal crack	0.946	0.821	-13%	0.337	-64%
Sealed longitudinal	0.972	0.827	-15%	0.0155	-98%
Concrete patch	0.970	0.812	-16%	0.506	-48%
Transverse crack	0.976	0.720	-26%	0.718	-26%
Joint crack	0.980	0.961	-2%	0.951	-3%
Sealed transverse crack	0.896	0.881	-2%	-	-
Slab edge	0.985	0.975	-1%	0.963	-2%

Table 7.20 highlights the mAP50 performance change from the training and validation to the test dataset for CRCP distress types, with a focus on generalization to new data. Notably, the model demonstrates strong test performance on the Transverse crack (0.873) and Spalled transverse crack (0.946), both showing minimal performance drops from training, indicating robust generalization for these well-defined and sufficiently represented classes. In contrast, the Longitudinal crack suffers a substantial decline in test performance, from 0.371 (Training) to just 0.0457, an 88% drop that points to a severe generalization failure, potentially due to underrepresentation or inconsistency in its visual patterns across datasets. For the remaining classes, including the Punchout, Asphalt patch, Concrete patch, and Sealed longitudinal crack, performance interpretation is limited by small instance counts in the test dataset, as indicated by

grayed-out values. This lack of data restricts meaningful evaluation and underscores the importance of ensuring adequate sample sizes for reliable assessment across all distress types.

Table 7.20 mAP50 performance change across training, validation, and test datasets for CRCP distress types

Class	Train	Val	Change (Train-Val)	Test	Change (Train-Test)
Longitudinal crack	0.371	0.190	-49%	0.0457	-88%
Sealed longitudinal crack	0.777	0.754	-3%	0.678	-13%
Punchout	0.953	0.829	-13%	0.745	-22%
Asphalt patch	0.987	0.965	-2%	0.519	-47%
Concrete patch	0.959	0.714	-26%	0.413	-57%
Transverse crack	0.893	0.835	-6%	0.873	-2%
Sealed transverse crack	0.942	0.877	-7%	0.524	-44%
Spalled longitudinal crack	0	-	-	-	-
Spalled transverse crack	0.967	0.751	-22%	0.946	-2%

On the other hand, we explored alternative methodologies to address the limitations of the previously developed YOLOv5 model in Chapter 5. The upgraded YOLOv8 model, trained on the same dataset, delivers significant performance improvements on the JCP and CRCP datasets. Also, progress on model training for previous YOLOv5 model has been made on the ACP dataset. These achievements encourage us to continuously dive deep into algorithm development for the AI/ML models. Additionally, a parallel study using datasets from another vendor demonstrates improved robustness and generalization. It also shows the effectiveness of merging multiple data sources to generalize the AI models for real applications.

Chapter 8 Conclusions and Recommendations

8.1 Conclusions

This study focused on developing automated pavement assessment methods for TxDOT to reduce dependency on specific data collection contractors and proprietary equipment. Leveraging AASHTO's national standard data format for 2D/3D pavement images and recent advancements in vision-based technologies, the research team created a standardized image library. This library categorizes pavement types into distinct datasets (see Table 8.1) for training and testing artificial intelligence (AI)/machine learning (ML) models against distresses defined in the Pavement Management Information System (PMIS) manual. Specifically, the team utilized YOLO (You Only Look Once) architectures to ensure generalized and robust distress measurements. Following a feedback-driven refinement process, the final application software was delivered to TxDOT. Our research indicates that the proposed AI and ML models can automatically detect and measure pavement surface distresses with a mAP50 of over 0.80 for ACP, 0.70 for JCP, and 0.75 for CRCP on the validation datasets. The pilot study using the real-world data collection has not only delivered promising results but also confirmed the effectiveness of the proposed methods.

Based on these research components, there are a few conclusions drawn below:

1. **Review literature on pavement condition assessment and Artificial Intelligence (AI)**: the literature review started with the standards and protocols for pavement condition assessment used by the AASHTO and TxDOT. Distress types for flexible and rigid pavements in TxDOT's PMIS manual are summarized to guide data library preparation. Traditional distress measurement methods were investigated, although the focus was on the AI-based methods for pavement distress measurements. Methods such as Faster R-CNN, YOLOv2, U-Net, GANs (Generative Adversarial Networks), and other deep learning methods were carefully reviewed for distress detection. In addition, datasets that were available in public were analyzed and compared, especially distinguishing the top-down and rider-view images.
2. **Establish a library of pavement 2D/3D Images in the AASHTO standard format**: the fundamental image library was developed for the Initialization Stage using the randomly selected PSI files. A total of 19,418 images (final data library) for bounding boxes and a total of 229 images for segmentation masks had been annotated. In the Detection category, which involves annotations with bounding boxes and distress classification, there were 5,892, 7,750, and 5,776 images available for ACP, JCP and CRCP. Instances for these distresses varied as shown in Table 8.1. In the Segmentation category, which involves more granular, pixel-level annotations, ACP had 114 images, JCP had 53, and CRCP had 62 (see Chapter 3). and metric calculation, illustrating the significance of

accuracy and precision in interpreting pavement condition data for informed decision-making in pavement asset management.

ACP		JCP		CRCP	
Class	Instances	Class	Instance	Class	Instances
All	10,885	All	16,943	All	13,779
Transverse	1,369	Failed joint	519	Longitudinal crack	372
Joint	400	Corner break	300	Sealed longitudinal crack	182
Sealed transverse	1,802	Punchout	96	Punchout	81
Longitudinal	1,565	Asphalt patch	346	Asphalt patch	1,110
Lane longitudinal	1,256	Failed concrete patch	42	Concrete patch	180
Sealed longitudinal	3,400	Popout	15	Transverse crack	11,234
Block	252	Longitudinal crack	221	Sealed transverse crack	398
Alligator	711	Sealed longitudinal	995	Spalled transverse crack	55
Pothole	130	Concrete patch	505		
		Transverse crack	824		
		Joint crack	1,377		
		Sealed transverse crack	3,308		
		Slab edge	1,745		

Table 8.1 Number of distress instances in datasets of each pavement type

** The number of images for ACP, JCP, and CRCP is 5,892, 7,750, and 5,776, respectively.

3. **Evaluate current practices of pavement condition assessment**: The research team presented an in-depth analysis of the application of various image processing techniques

for pavement distress segmentation. The primary objectives were to explore the capabilities and limitations of rules-based distress identification methods and the main challenges that impose difficulties on distress identification with digital images. Four rules-based image processing techniques, thresholding, edge detection, seed-based crack detection, and multiscale wavelets were selected to address the pavement distress segmentation problem. While each method held potential in specific contexts, their limitations underscored the need for further innovation and development in pavement distress segmentation technologies. The inability to effectively detect thin cracks, the challenge of distinguishing distress from pavement texture, and the lack of generalization capability indicated a more adaptable, intelligent approach, potentially leveraging advancements in AI technologies to overcome these hurdles.

4. **Develop machine learning-based models for pavement distress measurement:** the performance of the models was provided across different datasets, highlighting their strengths and weaknesses. Various factors that influence the model's performance, including the size of the distress samples, the characteristics of the pavement surfaces and distresses, and the methods employed for detection, were discussed. Suggestions such as expanding the dataset by including more samples from underrepresented distress classes and exploring new model architecture were provided to improve detection accuracy and robustness further. For distress segmentation, the performance across the JCP, ACP, and CRCP datasets showed that the models performed well on uniform surfaces like JCP, but struggled more on complex textures, such as in ACP and CRCP, where crack boundaries were harder to detect. For distress detection, the model generally delivered satisfactory results for distress classes with sufficient samples. Additionally, when compared with the PathView system, the model excelled in precision, producing significantly fewer false positives. But the model struggled with underrepresented distress classes, such as Potholes and Block cracks in ACP, and Corner Break, Punchout, D-cracking, Failed concrete patch, and Popout in JCP, where performance was notably poor. Despite having many samples, the model's performance on Transverse crack and Longitudinal crack in ACP was also suboptimal. Furthermore, experiments indicated that the type of images used (intensity, range, or both) had a significant impact on performance: for ACP, using both images yielded the best results, while for JCP, using intensity images alone performed just as well as using both. These findings suggested the need for more data, particularly for underrepresented classes, and a refined approach to image type selection for optimal performance.

5. **Develop practical tools for pavement condition assessment**: a comprehensive workflow was outlined for transforming AI-based pavement distress detection outputs into standardized PMIS distress scores. The purpose was to enhance the accuracy, efficiency, and consistency of pavement condition assessments by aligning machine learning detection results with TxDOT's established scoring framework. Post-processing

rules were tailored for ACP, JCP, and CRCP to convert raw detection outputs into distress categories recognized by the PMIS. To ensure compatibility with the Pavement Rater's Manual, each pavement type had specific criteria to filter, consolidate, and qualify detected distresses. Following post-processing, distress detection results were transformed into PMIS ratings and normalized to generate L_i values. These values were then used to compute utility values through an exponential decay function. The final distress score, a product of all Utility values, would reflect the overall pavement condition on a scale from 1 (worst) to 100 (best). These examples for ACP, JCP, and CRCP sections were validated the robustness and adaptability of the framework in real-world scenarios.

6. **A pilot study for machine learning-based pavement condition assessment implementation**: a pilot study had been conducted on the 2024 collection of pavement data in Brazoria County, as well as the new development of AI/ML models. The prepared data library did not have any 2D/3D images selected from this county. For ACP, one trained model demonstrated strong performance on major distress types with sufficient sample sizes, including transverse cracks (0.818), sealed transverse cracks (0.958), longitudinal cracks (0.823), and sealed longitudinal cracks (0.899) for mAP50. But its performance on lane longitudinal cracks (0.599) and alligator cracking (0.577) was noticeably lower for mAP50, indicating that these distress types present greater detection challenges. For JCP, it was evident that the model achieves high performance on the Joint crack (0.951) and Slab edge (0.963), both of which were supported by a large number of instances (1,086 and 1,837, respectively). In contrast, classes with fewer than 50 instances, such as the Sealed longitudinal, Popout, and Corner break, exhibited low detection metrics. For CRCP, transverse and spalled transverse cracks stood out with strong performance: both classes show high precision and recall, and mAP50 scores of 0.873 and 0.946, respectively. In contrast, longitudinal cracks, despite having 58 instances, showed very poor recall (0.0469) and a low mAP50 of 0.0457, indicating that the model rarely detects them correctly and needs more annotated data.

8.2 Recommendations

Based on these conclusions, the following recommendations are proposed to improve the proposed framework.

1. **Continuously review pavement condition assessment and AI technologies**: state-of-the-art AI/ML advancements and global and local pavement condition assessment practices should be investigated and adopted to overcome current models' limitations. Also, better practices and tools for data preparation, particularly those supporting robust AI/ML model development should be studied.

2. **Expand the AASHTO-standardized 2D/3D pavement image library**: it is in great need to prepare more targeted 2D/3D pavement surface images in the AASHTO standard. To support accurate distress detection and segmentation, manual and semi-automated annotation of various pavement distresses on 2D/3D images should be implemented. A statistical analysis can be applied to this image library to identify gaps and execute targeted data collection to ensure balanced and comprehensive datasets for different pavement types.
3. **Continuously evaluate current pavement condition assessment practices**: the efficacy of current automated pavement condition assessment technologies should be investigated with on a dual-track assessment. This involves an analysis of the pavement condition results provided by data collection contractor of TxDOT. The rules-based assessment methods should be assessed using the standardized pavement image library that has been expanded purposely.
4. **Develop AI/ML-based models for automated pavement distress quantification**: AI/ML-based models for the detection and segmentation of distress types should be the main focus for automated pavement distress measurement. The final goal is to find the most effective neural network architecture and develop specialized algorithms. These models will be trained and optimized using the updated image library to perform two primary tasks: distress detection with bounding box prediction and pixel-level segmentation (i.e., isolate distresses from the background).
5. **Develop deployable pavement condition evaluation tools**: although current application software has been updated based on TxDOT's feedback, more functions could be added since new models could be trained with new image data library. The tools used to expand the data library should be provided and possibly incorporated into the application software for better data management.
6. **Perform Scalable pilot study to evaluate the AI/ML-based system**: a pilot study should be conducted utilizing a diverse sample of pavement sections covering a variety of pavement types, pavement conditions, age groups, locations, and other parameters that would affect the characteristics of the pavement surface. The practical feasibility of the automated assessment system could be determined. All identified limitations of the developed application revealed in the pilot study could be documented for end-users on the software's appropriate deployment and establish a strategic roadmap for future research and development.

By implementing these recommended action items, TxDOT can significantly improve the efficiency and accuracy of AI based technologies for automated pavement condition assessment using 2D/3D surface images. This will ultimately assist TxDOT in enhancing the data quality assurance and reliability of automated pavement condition data collection and processing. The conclusions and recommendations constitute a robust framework for overcoming existing

hurdles on the automation of pavement condition assessment and help the State of Texas improve its pavement performance.

References

American Association of State Highway and Transportation Officials. (2018a). *Collecting images of pavement surfaces for distress detection* (AASHTO R 86).

American Association of State Highway and Transportation Officials. (2018b). *Quantifying cracks in asphalt pavement surfaces from collected pavement images utilizing automated methods* (AASHTO R 85).

American Association of State Highway and Transportation Officials. (2020). *File format of 2-dimensional and 3-dimensional (2D/3D) pavement image data* (AASHTO MP 47).

American Association of State Highway and Transportation Officials. (2023). *Crack annotation and crack length and width computation on 2D/3D pavement images* (Subcommittee 5a).

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., Mraz, A., Kashiyama, T., & Sekimoto, Y. (2020). Transfer learning-based road damage detection for multiple countries. *arXiv*.

Ayenu-Prah, A., & Attoh-Okine, N. (2008). Evaluating pavement cracks with bidimensional empirical mode decomposition. *EURASIP Journal on Advances in Signal Processing*, *2008*, 1–7.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495.

Bai, Y., Wang, F., Gong, H., & Luo. X. (2026). Evaluation of deep learning strategies using two different image datasets for automated pavement distress detection methodology. United States. *The 105th Annual TRB Meeting*, Washington, DC, January 5-9.

Bang, S., Park, S., Kim, H., & Kim, H. (2019). Encoder–decoder network for pixel-level road crack detection in black-box images. *Computer-Aided Civil and Infrastructure Engineering*, *34*(8), 713–727.

Biswas, D., Scouten, A., Gong, H., Wang, F., & Tešić, J. (2026). MoPac+: Multimodal Modeling of the Pavement cracks using hard-example mining and context-aware feature aggregation. *Measurement*, 120896.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv*.

Cai, Z., & Vasconcelos, N. (2019). Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(5), 1483–1498.

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.

Chen, K., Yadav, A., Khan, A., Meng, Y., & Zhu, K. (2019). Improved crack detection and recognition based on convolutional neural network. *Modelling and Simulation in Engineering*, *2019*.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848.

Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems* (pp. 379–387).

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., & Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 6569–6578).

Dung, C. V., & Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. *Automation in Construction*, *99*, 52–58.

Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U., & Gross, H. M. (2017). *How to get pavement distress detection ready for deep learning? A systematic approach*. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2039–2047). IEEE.

Ghosh, R., & Smadi, O. (2021). Automated detection and classification of pavement distresses using 3D pavement surface images and deep learning. *Transportation Research Record*, *2675*(9), 1146–1157.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440–1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580–587).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2961–2969).

Hong, Y., Pan, H., Sun, W., & Jia, Y. (2021). Deep dual-resolution networks for real-time and accurate semantic segmentation of road scenes. *arXiv*.

Hsieh, Y. A., & Tsai, Y. J. (2020). Machine learning for crack detection: Review and model performance comparison. *Journal of Computing in Civil Engineering*, *34*(5), 04020038.

Huang, Y., & Xu, B. (2006). Automatic inspection of pavement cracking distress. *Journal of Electronic Imaging*, *15*(1), Article 013017.

Jenkins, M. D., Carr, T. A., Iglesias, M. I., Buggy, T., & Morison, G. (2018, September). *A deep*

convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks. In 2018 26th European Signal Processing Conference (EUSIPCO) (pp. 2120–2124). IEEE.

Li, S., & Zhao, X. (2019). Image-based concrete crack detection using convolutional neural network and exhaustive search technique. *Advances in Civil Engineering*, *2019*.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2117–2125).

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2980–2988).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). *Ssd: Single shot multibox detector*. In European Conference on Computer Vision (pp. 21–37). Springer, Cham.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431–3440).

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1127–1141.

Majidifard, H., Jin, P., Adu-Gyamfi, Y., & Buttlar, W. G. (2020). Pavement image datasets: A new benchmark dataset to classify and densify pavement distresses. *Transportation Research Record*, *2674*(2), 328–339.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N., & Terzopoulos, D. (2021). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Oliveira, H., & Correia, P. L. (2009, August). *Automatic road crack segmentation using entropy and image dynamic thresholding*. In 2009 17th European Signal Processing Conference (pp. 622–626). IEEE.

Pierce, L. M., & Weitzel, N. D. (2019). *NCHRP Synthesis 531: Automated pavement condition surveys*. Transportation Research Board.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779–788).

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7263–7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (Vol. 28, pp. 91–99).

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-net: Convolutional networks for biomedical image segmentation*. In Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 (pp. 234–241). Springer.

Tan, M., & Le, Q. (2019). *Efficientnet: Rethinking model scaling for convolutional neural networks*. In International Conference on Machine Learning (pp. 6105–6114). PMLR.

Texas Department of Transportation. (2009, June 29). *Pavement Manual*.

Texas Department of Transportation. (2021, January 21). *Overview of calculation of PMIS condition score*.

Texas Department of Transportation. (2023). *Pavement management information system: Pavement rater's manual*.

Tsai, Y. C., Kaul, V., & Mersereau, R. M. (2010). Critical assessment of pavement distress segmentation methods. *Journal of Transportation Engineering*, *136*(1), 11–19.

Tsai, Y. J., Wang, Z., & Liu, T. (2019). *Evaluation of proposed standard data format and compression algorithms for 2D/3D pavement surface image*. Federal Highway Administration.

Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2024). Yolov9: Learning what you want to learn using programmable gradient information. *arXiv*.

Wang, K. C., Li, Q., & Gong, W. (2007). Wavelet-based pavement distress image edge detection with a trous algorithm. *Transportation Research Record*, *2024*(1), 73–81.

Wang, K. C. P., Li, Q. J., & Chen, C. (2016). *Development of standard data format for 2-dimensional and 3-dimensional (2D/3D) pavement image data used to determine pavement surface condition and profiles task 2 - research current practices*. url: https://pooledfund.org/Document/Download/7858.

Wang, P., Chen, P., Yuan, Y., Liu, D., Huang, Z., Hou, X., & Cottrell, G. (2018, March). *Understanding convolution for semantic segmentation*. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1451–1460). IEEE.

Wang, X., & Hu, Z. (2017). *Grid-based pavement crack analysis using deep learning*. In 2017 4th International Conference on Transportation Information and Safety (ICTIS) (pp. 917–924). IEEE.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., & Luo, P. (2021). SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural*

Information Processing Systems (Vol. 34, pp. 12077–12090).

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1090–1109.

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv*.

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2021). A survey of modern deep learning based object detection models. *arXiv*.

Zakeri, H., Nejad, F. M., & Fahimifar, A. (2017). Image based techniques for crack detection, classification and quantification in asphalt pavement: A review. *Archives of Computational Methods in Engineering*, *24*(4), 935–977.

Zhang, A., Wang, K. C., Fei, Y., Liu, Y., Chen, C., Yang, G., Li, J. Q., Yang, E., & Qiu, S. (2019). Automated pixel-level pavement crack detection on 3D asphalt surfaces with a recurrent neural network. *Computer-Aided Civil and Infrastructure Engineering*, *34*(3), 213–229.

Zhang, A., Wang, K. C., Li, B., Yang, E., Dai, X., Peng, Y., Fei, Y., Liu, Y., Li, J. Q., & 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.

Zhang, K., Zhang, Y., & Cheng, H. D. (2020). Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks. *Journal of Computing in Civil Engineering*, *34*(3), 04020004.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2881–2890).

Zhou, Y., Wang, F., Meghanathan, N., & Huang, Y. (2016). Seed-based approach for automated crack detection from pavement images. *Transportation Research Record*, *2589*(1), 162–171.

Zou, Q., Cao, Y., Li, Q., Mao, Q., & Wang, S. (2012). CrackTree: Automatic crack detection from pavement images. *Pattern Recognition Letters*, *33*(3), 227–238.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & Wang, S. (2018). Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, *28*(3), 1498–1512.

Zhu, S., Xia, X., Zhang, Q., & Belloulata, K. (2007, December). *An image segmentation algorithm in image processing based on threshold segmentation*. In 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System (pp. 673–678). IEEE.

Zimmerman, K. A. (2017). *Pavement management systems: Putting data to work*. (No. Project 20-05, Topic 47-08).