Pavement Defect Classification and Localization Using Hybrid Weakly Supervised and Supervised Deep Learning and GIS

Amir Jamali

A Thesis in

The Department of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirement

For the Degree of

Master of Applied Science (in Information Systems Security) at

Concordia University

Montreal, Quebec, Canada

August 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Amir Jamali

Entitled: Pavement Defect Classification and Localization Using Hybrid Weakly Supervised and Supervised Deep Learning and GIS

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Information Systems Security)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

______________________________ Chair

Dr. Chun Wang

______________________________ Examiner

Dr. Manar Amayri

______________________________ Supervisor

Dr. Amin Hammad

Approved by ______________________________

Dr. Jun Yan, Graduate Program Director

__________ ______________________________

Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

ABSTRACT

Pavement Defect Classification and Localization Using Hybrid Weakly Supervised and Supervised Deep Learning and GIS

Amir Jamali

Automated detection of road defects has historically been challenging for the pavement management industry. As a result, new methods have been developed over the past few years to handle this issue. Most of these methods relied on supervised machine learning techniques, such as object detection and segmentation methods, which need a large, annotated image dataset to train their models. However, annotating pavement defects is difficult and time-consuming due to their ununiformed and complex shapes. To address this challenge, a hybrid pavement defect classification and localization framework using weakly supervised and supervised deep learning methods is proposed in this thesis. This framework has two steps: (1) A robust hierarchical two-level classifier that classifies the defects in images, and (2) A method for defect localization combining weakly supervised and supervised techniques. In the localization method, first, defects are primarily localized using a weakly supervised method (i.e. Class Activation Mapping (CAM)). Next, based on the results of the first classifiers, the defects are segmented from the localized patches obtained in the previous step. The feature maps extracted from the CAM method are used to train a segmentation network once (i.e. U-Net or Mask R-CNN) to localize and segment the defects in the images. Thus, the proposed framework combines the advantages of weakly supervised and supervised methods. The supervised modules in the framework are trained once and can be used for any new data without the need to train. In other words, to use our framework on new dataset only the classifiers should be fine-tuned. Furthermore, the proposed framework introduced an innovative method designed to calculate the maximum crack width in pixels within linear segmented defect patches, derived from the localization module of the proposed framework. This method is particularly advantageous as it provides critical information that can be further employed in the calculation of the Pavement Condition Index (PCI).

Additionally, the proposed method benefits from an asset management inspection system based on Geographic Information System (GIS) technology to prepare the dataset used in the training and testing. Thus, this advanced system serves a dual role within our framework. Firstly, it assists in the assembly and preparation of the dataset used in the model training process, providing a geographically organized collection of images and related data. Secondly, it plays a crucial role in the testing phase, offering a spatially accurate platform for evaluating the effectiveness of the model in real-world scenarios.

A dataset from Georgia State in the USA was used in the case study. The proposed framework obtained high precision of 97%, 88%, 92% and 97% for localizing the alligator, block, longitudinal and transverse cracks, respectively. Considering all factors, such as annotation cost, and performance on the test dataset, the proposed localization method outperforms the supervised localization methods, such as instance segmentation and object detection for localizing road pavement defects.

ACKNOWLEDGEMENTS

My heartfelt appreciation goes to my supervisor, Dr. Amin Hammad, who generously offered me invaluable learning opportunities and unwavering support throughout my master's journey. His commitment, patience, and substantive contributions have greatly enriched my academic experience. His guidance, particularly in navigating the inevitable trials and uncertainties of research work, has significantly molded my intellectual growth and will indubitably have a lasting impact on my future endeavors.

I also extend my profound gratitude towards the members of my defense committee, Dr. Chun Wang and Dr. Manar Amayri. Their willingness to dedicate their time to review this thesis and provide insightful feedback has significantly contributed to the successful completion of this work. I truly appreciate their expert critique and constructive suggestions, which have served to enhance the quality and depth of my research.

I am also sincerely grateful to the dedicated team at Rival Solutions Inc. (Claude Laflamme & Rob Huber), with whom I had the honor of collaborating through a Mitacs partnership. Their crucial industry insights, unwavering support, and substantive contributions significantly impacted the trajectory of this research. This collaboration has not only amplified my learning journey, but also enriched my comprehension of the intricate facets of our research area.

Lastly, but certainly not least, I wish to dedicate this work to my late father. His unwavering faith in me, his timeless wisdom, and the enduring lessons he taught me have been my guiding lights throughout this journey. Although he is no longer with us, his memory continues to inspire and motivate me every day. His influence and values have profoundly shaped who I am today and continue to echo in every endeavor I undertake.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
ACM	Active Contour Model
AP	Average Precision
BFPN	Bidirectional Feature Pyramid Network
CAM	Class Activation Mapping
CBAM	convolutional block attention module
CCD	Charge-Coupled Device
CNN	Convolutional Neural Network
CRF	Conditional Random Field
CV	Computer Vision
DL	Deep Learning
FFT	Fast Fourier Transform
FHT	Fast Haar Transform
FN	False Negative
FP	False Positive
GIS	Geographic Information System
GPU	Graphical Processing Unit
HMM	Hidden Markov Model
IoU	Intersection over Union
IS	Intelligent Scissors
LBP	Local Binary Pattern

LiDAR	Light Detection and Ranging
LRCN	Long-term Recurrent Convolutional Network
LSSVC	least squares support vector classification
LSTM	Long Short-Term Memory
MRF	Markov Random Fields
NMS	Non-Maximum Suppression
PAMS	Pavement Asset Management System
PCA	Principal Component Analysis
PCI	Pavement Condition Index
R-CNN	Region-based Convolutional Neural Network
RGB	Red, Green, and Blue
RNN	Recurrent Neural Network
scSE	Spatial-Channel Squeeze and Excitation
SSD	Single Shot Detector
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
WSOD	Weakly Supervised Object Detection
YOLO	You Only Look Once

Chapter 1: Introduction

1.1 Background and Problem Statement

Pavement Asset Management System (PAMS) is a software that assists transportation authorities and decision-maker agencies in efficiently managing their pavement assets. PAMS assesses the condition of the pavement, calculates its remaining useful life, and chooses the most economical maintenance and rehabilitation options using data and analytical tools. By using PAMS, agencies can optimize their budget allocation and prioritize maintenance and rehabilitation activities based on the condition of their pavement assets. Overall, a PAMS is a valuable tool for transportation experts seeking to maximize the lifespan of their pavement assets and improve safety for road users by minimum costs [1].

In any robust PAMS, the current and historical state of roads are crucial indicators for preparing optimized long-term maintenance and rehabilitation plans. In other words, a model generated by the current pavement situation can be adopted, not just for maintenance strategy, but also to achieve an appropriate design as well.

The following data is required for making maintenance decisions in PAMS [1]:

1. Pavement inventory data: Inventory data includes information about the pavement's construction, such as its thickness and materials type. This data is vital for understanding the pavement's structural capacity and determining its design life.
2. Pavement condition data: Condition data refers to the current physical state of the pavement degradation, such as defect types, roughness, etc. Collecting condition data regularly allows engineers to track the progression of pavement distress over time and make appropriate decisions about maintenance and rehabilitation strategies.
3. Design data: Design data refers to the information such as pavement and mix design details, material characteristics, and specifications of pavement layers, etc.
4. Maintenance history data: Timing and methods adopted to execute previous maintenance activities.
5. Auxiliary data: Auxiliary data comprises information such as traffic volume, climate, and pavement age. This data is important for understanding how external factors may contribute to pavement deterioration and predict future pavement condition.

Database and data collection methods for PAMS is shown in Figure 1-1. Regarding the aforementioned data illustrated in Figure 1-1, inventory, condition and auxiliary data are crucial to evaluate the pavements and should be collected and monitored periodically [1].

A key element for planning is predicting future road conditions using deterioration rates or models from condition data. Deterioration modeling are best achieved when asset conditions are monitored over time and further projected into the future. Therefore, accurate and reliable road condition assessment is critical, and automation in this step plays a central role in making the overall PAMP more reliable. Data analysis and collection, including data from distress surveys, is one of the major challenges to evaluating pavement conditions and prioritizing maintenance and rehabilitation activities [1]. Artificial intelligence tools handle this issue to a great extent by automating and expediting both data collecting and processing processes. For instance, automatic defect type and severity recognition can be done by analyzing the images with image processing and computer vision tools. Next, machine learning algorithms can predict the future pavement condition based on the information extracted by the previous stage about the defects [1].

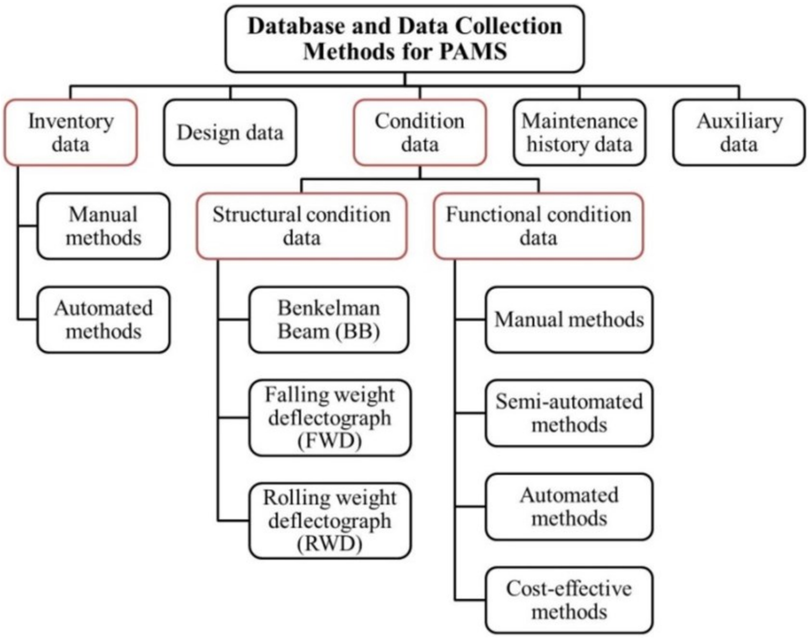


Figure 1-1 Database and data collection methods for PAMS [1]

The more distress the pavement has, the more maintenance it needs to keep it in service structurally. Pavement defects are the visible part of the deficiency of roads and are symptomatic of broader pavement degradation issues. Thus, regularly evaluating pavement defects would be an urgent aspect of pavement management, as it allows engineers and managers to recognize and address deficiencies in a timely manner before they get worse and become costly to repair. By recognizing the patterns and trends of pavement defects, engineers can develop effective maintenance and rehabilitation strategies that are tailored to the specific needs of the pavement. Moreover, monitoring pavement defects can optimize pavement maintenance costs. For instance, if pavement defects are recognized and repaired promptly, minor preservation measures such as crack sealing or pothole patching may be urgent to prevent further degradation and extend the pavement’s lifespan [2]–[5]. On the other hand, if pavement distresses are not repaired promptly, it costs experts and managers a lot. That is because, in this case, the pavement may require more extensive and costly maintenance activities such as resurfacing or reconstruction. In summary, regular observation of distresses has a vital role in the handling of pavement degradation in a timely and cost-effective manner, which helps to optimize pavement performance and leads to extending its lifespan. Recently, thanks to drastic developments in computer vision and artificial intelligence approaches, road defect detection is not only automated but also the performance of the overall process is also highly enhanced [2]–[5]. These methods analyze different aspects of defects to determine the cause of deterioration. They analyze huge data emanating from defects. The more adequate data provided about the age, traffic, and other pavement variables the system has, the more reliable prediction it forms. Therefore, analyzing these defects with respect to their patterns, shapes, topology, severity, and quantity makes it possible to find their root causes. Since the defect localization has urgent role in pavement condition assessment, a vast number of techniques have been developed to detect and localize the defects in recent years [2]. After detecting, classifying and quantifying the distresses, The pavement condition index (PCI)[6] is used for pavement assessment. PCI is a numerical metric that quantizes visual inspection of the pavement surface in the range from 0 to 100. The higher value the PCI has, the better the condition of the pavement is. The PCI methodology comprises a visual inspection of the pavement surface by a skillful inspector who evaluates the number of distresses, such as cracking, potholes, and rutting and their severities concerning their length and extension. The severity ratings are then combined to calculate an overall PCI for the pavement. The PCI is an effective tool for prioritizing maintenance and rehabilitation efforts on roads and parking lots. It allows agencies to identify pavement sections that require immediate attention and those that can wait until a later time. The PCI can also be used to evaluate the effectiveness of different maintenance strategies over time. Figure 1-2 shows the standard PCI rating scales.

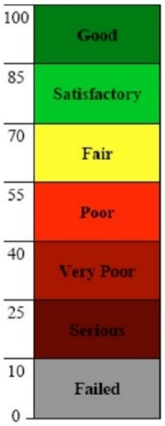
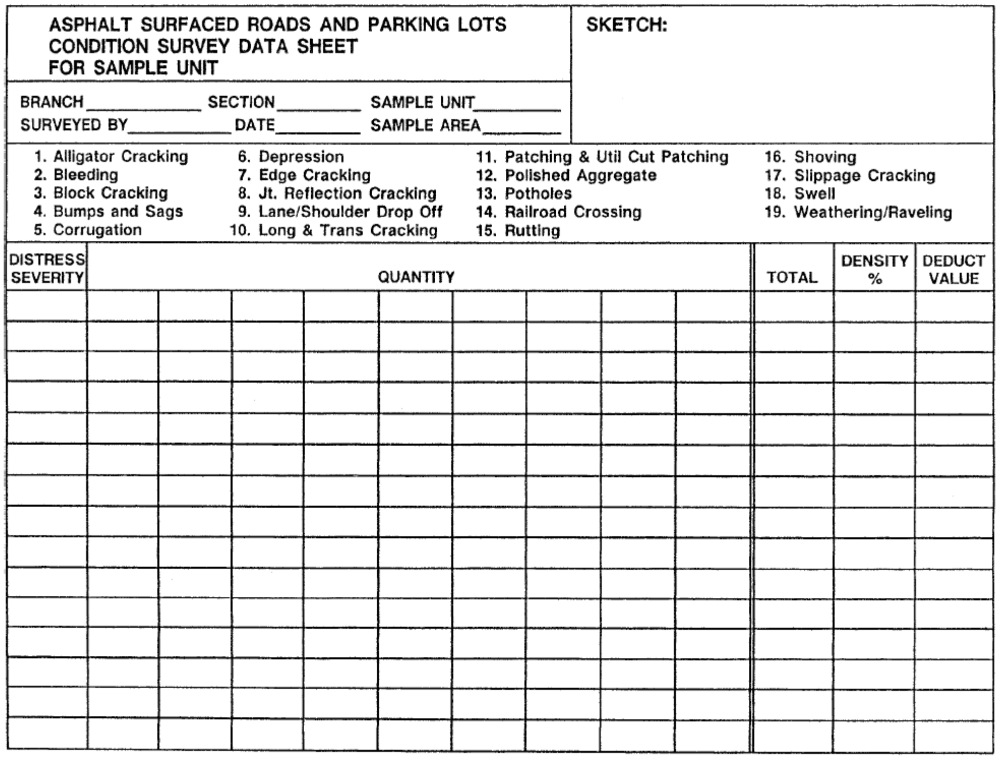


Figure 1-2 Standard PCI rating scales and suggested colors for their representation [6]

The suggested colors may vary depending on the agency or organization using the PCI system [6]. They allow agencies to identify areas that require maintenance or repair promptly on the maps. A PCI data sheet should provide the following information: date, location, branch, section, sample unit size, slab number and size, distress types, severity levels, quantities, and names of surveyors [6]. This information is then used to calculate the PCI and to develop maintenance and repair plans. A flexible example of PCI data sheet is shown in Figure 1-3.



ASPHALT SURFACED ROADS AND PARKING LOTS
CONDITION SURVEY DATA SHEET
FOR SAMPLE UNIT

SKETCH:

BRANCH____________ SECTION_________ SAMPLE UNIT_________
SURVEYED BY____________ DATE_________ SAMPLE AREA_________

1. Alligator Cracking
2. Bleeding
3. Block Cracking
4. Bumps and Sags
5. Corrugation
6. Depression
7. Edge Cracking
8. Jt. Reflection Cracking
9. Lane/Shoulder Drop Off
10. Long & Trans Cracking
11. Patching & Util Cut Patching
12. Polished Aggregate
13. Potholes
14. Railroad Crossing
15. Rutting
16. Shoving
17. Slippage Cracking
18. Swell
19. Weathering/Raveling

DISTRESS SEVERITY	QUANTITY									TOTAL	DENSITY %	DEDUCT VALUE

Figure 1-3 A flexible PCI data sheet [6]

Once the types and severity of distresses have been evaluated, a pavement condition rating is assigned based on a standardized rating rule. The mentioned rating system first assigns a score to each type of defect. Next, the scores are weighted regarding the extent of each defect. The final PCI value is then calculated as a weighted average of the individual distress scores. The PCI calculation can be performed manually or using specialized software designed for this purpose. PCI calculation's performance is affected by two factors: (1) The inspector's expertise and experience, and (2) the quality and consistency of the rating system being employed. Hence, to achieve an accurate PCI result, both the aforementioned factors should meet high standards.

As a result, automating the PCI calculation can tremendously reduce the time, cost and subjectivity associated with manual pavement surveys done by a human. It also improves the accuracy and consistency of the PCI calculation. The automation of PCI calculation can be accomplished as follows [1]:

1. Pavement data collection: The first step is to collect pavement data in the form of images or videos from the pavement surface using specialized pavement surveying equipment, such as a pavement condition assessment vehicle.

2. Machine learning and image processing process: The pavement images or videos are then processed using computer vision and machine learning techniques to classify, detect and segment the pavement distresses.

3. Feature extraction: Once the distresses have been segmented, relevant features such as size, shape, and texture are extracted from the distresses.

4. PCI calculation: Finally, the PCI is calculated using the machine learning output, which assigns a severity score to each type of distress. The severity scores are then weighted based on the extent of each distress, and the final PCI value is calculated as a weighted average of the individual distress scores.

However, it is significant to note that automated methods could not detect all defects of any shape and structure. Thus, human inspection is still required to ensure accuracy.

1.2 Research Objectives

This thesis aims to develop a hybrid pavement defect type and severity classification, and defect localization framework using weakly supervised and supervised deep learning methods based on images captured via passive Charge-Coupled Device (CCD) cameras to automate the pavement condition evaluation. The specific objectives are: (1) Developing a supervised method for classifying the types and severity of defects; and (2) Developing a hybrid approach integrating weakly supervised and supervised methods for defect localization; and (3) Estimating the width of the linear cracks, which can be used for calculating the PCI of the pavement. The proposed framework also integrates a Geographic Information System (GIS)-based inspection asset management system called RUBIX [117], which provides comprehensive monitoring and evaluating of the pavement's condition. The method proposed in this study combines the advantages of both supervised and weakly supervised approaches and demonstrates superior performance compared to previous methods.

1.3 Thesis Organization

The structure of this thesis is as follows:

Chapter 2 - Literature Review: In this chapter, a comprehensive review of the literature on the previous methods for common defect recognition, detection, and segmentation is presented. The main focus is on techniques that have been used across a variety of domains, like computer vision, image processing, 3D computer vision, and methods based on deep learning.

Chapter 3 – Proposed method: This chapter introduces the hybrid proposed methodology for detecting and recognizing pavement cracks, utilizing a weakly supervised and supervised learning approaches. Detailed explanations are provided for each submodule within this innovative framework. By integrating weakly supervised and supervised methods, this hybrid approach reaps the benefits of both techniques, thereby enhancing the efficiency and effectiveness of pavement crack classification and localization. The proposed method further provides a mechanism for precisely estimating the width of the linear cracks in pixel measurements using patches derived from the localization module. This information will be used later in PCI calculation.

Chapter 4 – Case Study: This chapter shows the empirical outcomes derived from the proposed framework. Additionally, a comparative study between the proposed hybrid localization approach and established supervised learning techniques, including instance segmentation and object detection methods, is conducted. It reveals that, upon considering all factors, the proposed framework significantly surpasses the performance of supervised methods.

Chapter 5 – Summary, Contributions and Future Work: In this chapter, a summary of the thesis is presented, encompassing its noteworthy contributions and advancements in the field. Moreover, any challenges faced during the research are also discussed. Furthermore, to direct further exploration in this domain, recommendations for future research avenues are put forth.

Chapter 2: Literature Review

2.1 Introduction

This chapter endeavors to present a thorough and inclusive examination of the various techniques previously utilized for the purpose of crack identification and localization. The methodologies encompass a broad spectrum, encompassing both machine learning and computer vision methods. First, classical image processing and computer vision techniques are examined for their applicability in pavement crack detection. They mostly involve several steps such as thresholding, edge detection and morphological operations. Next, 3D computer vision methods that leverage the 3D point clouds and 3D imaging by capturing detailed spatial information and integrating state-of-the-art imaging technologies are considered. Finally, a comprehensive investigation is carried out on deep learning-based methods, including both supervised and weakly supervised approaches, owing to their exceptional performance. This chapter also considers the effectiveness and drawbacks of the mentioned methods to highlight the areas where further improvements can be made.

2.2 Classic Machine Learning and Computer Vision Methods

Some approaches developed image processing and computer vision techniques to detect defects in pavement images obtained from the CCD cameras. CCD camera sensors convert light into an electrical charge in pixels. In general, these methods comprise the major steps as follows [1]:

1. **Pre-processing**: Different climate conditions may challenge the data collection process by affecting the image contrast and illumination, which may affect crack detection in the next steps. Thus, In pre-processing, some image processing techniques such as contrast stretching, histogram equalization and filtering are performed to recover the image from the environmental artifact noise to some extent [7].
2. **Segmentation** is the process in which the pixels belonging to specific objects in the image, such as the defects, are specified as the region of interest. After segmentation, the extent and severity of the defects can be calculated by some other techniques from the extracted regions from the segmentation process. There are various techniques and methods for image segmentation including: (1) Edge-based segmentation: This method involves detecting edges or boundaries between regions in an image based on changes in intensity or color. Since this method is noise-sensitive, it is advantageous to smooth the image to some extent before using the approach. (2) Region-based segmentation: In this method, pixels with the same features, such as color, texture, etc. are considered as one group. This method is helpful to a great extent for images in which the edges are not well-defined due to noise. (3) Threshold-based segmentation: This method categorizes the pixels into the foreground and background groups by comparing their intensity values with a specific threshold. This method is simple and fast, but may not work well for images with varying lighting or contrast. (4) Fuzzy-based segmentation: This method uses fuzzy logic to assign degrees of membership to pixels based on their similarity to different regions or objects. It can handle variations in texture and color but may require more computational resources. (5) Partial differential equation-based segmentation: This method involves solving partial differential equations to identify boundaries or regions in the image. It can be computationally intensive but can produce accurate results.

When compared to employing only one method, hybrid segmentation techniques, which incorporate several segmentation techniques, frequently produce better results. Since a combination of multiple segmentation techniques outperforms a single method in overcoming the drawbacks or limits of certain segmentation approaches, it may be better suited to various types of images or regions of interest [8]. To handle variations in lighting or contrast within the image, for instance, a hybrid approach might combine threshold-based segmentation with fuzzy-based segmentation. Alternatively, it might combine edge-based segmentation with region-based segmentation to better capture the edges and texture of an object of interest. However, combining the procedures without calibration does not yield better results. Therefore, the parameters and weighting of each technique must be carefully considered and optimized while designing a hybrid segmentation method. The computational cost of a hybrid technique may also be higher than employing a single method; hence, the methods should be selected regarding the particular application and available resources.

3. **Feature extraction and selection**: Feature extraction techniques can eliminate noises such as shadow regions and extract the region of interest, including the defects. Various feature extraction techniques, such as curvelet, Haar, Fourier, Wavelet, were used to extract defects from the images [9], [10]. In order to construct a representation of the crack in a skeleton shape, that is also useful for estimating the crack's width and other properties, the skeleton approach was employed [11]. Next, extracted features were classified to recognize defect types and estimate their lengths. A Canny edge detector has been applied to locate cracks [12]. Comparing Fast Haar Transform (FHT), Fast Fourier Transform (FFT), Sobel, and Canny edge detection algorithms, FHT demonstrated greater performance compared to the other edge detection methods [13]. An Active Contour Model (ACM) has been also used to obtain crack location and geometry [14]. An ACM is a type of deformable model that seeks to minimize an energy functional defined over the contour of the object being segmented. The Markov Random Fields (MRF) model has also been used to segment cracks [15]. However, both ACM and MRF are computationally expensive. Furthermore, ACM is sensitive to initialization and parameter settings. MRF performance also depends on the quality of the image and the choice of model parameters. Hence, it cannot be generalized for detecting defects with complex shapes.
4. **Detection and classification**: Extracted features from the previous stage are used in this stage to detect and classify the defect types and their severities. For example in [16] the images were segmented by 1D and 2D thresholding to segment the defects. Next, the segmented defects are categorized into the classes such as cracking, rutting, and potholes with different severities. The fused features from the local binary pattern (LBP) and principal component analysis (PCA) were classified by SVM to detect the cracks [17]. PCA is a statistical technique used for dimensionality reduction and feature extraction. LBP (Local Binary Pattern) is a texture descriptor used for texture classification. A novel crack segmentation method based on the Random Structured Forest was proposed [18]. A tile-based crack assessment technique has been developed to detect cracks from the 2D and 3D images [19]. However, the types of the defects cannot be detected by this method [20]. Hybridization of some computer vision and machine learning methods such as least squares support vector classification (LSSVC) and forensic-based investigation (FBI) were applied to recognize pavement rutting [21]. LSSVC is a supervised learning classification algorithm that finds the hyperplane that maximally separates the data points of different classes. Forensic-based Investigation (FBI) is a process that involves a detailed analysis of

pavement distresses to identify the root causes. The goal of the FBI is to determine the factors that have contributed to pavement distress and develop appropriate maintenance or rehabilitation strategies.

However, classical image processing and computer vision methods have certain limitations. They tend to rely heavily on handcrafted features, requiring expert knowledge and domain-specific expertise to design effective algorithms. This process can be time-consuming and may not generalize well to diverse datasets or complex scenarios [1]. Additionally, in contrast to deep learning methods, using shallow learning techniques make their performance susceptible in tackling the problems of illumination and background changes. Thus, their results are inferior to the deep-learning methods [22].

2.3 3D Methods

Since some pavement distresses, such as potholes and ruttings, are inherently three-dimensional, 3D defect detection algorithms, which capture the complete pavement geometry, are extremely helpful for identifying and quantifying such distresses [5]. 3D defect detection methods use 3D data generated by 3D photogrammetry, such as 3D laser-based imaging [5]. 3D data generated by structured light imaging was used to detect cracks [23]. Structured light technique projects a pattern of beams onto a scene and extract the 3D model of the scene by measuring the distortion in the projected pattern. 3D data can be used as 3D point cloud or 3D image. A 3D image is just like a regular image with the scene's depth information included. Each 3D pixel (i.e. a voxel), comprises information about the 3D space with color and other properties of the scene. 3D point clouds contain three-dimensional spatial information as well as others like color. In contrast to the 3D image, the 3D point clouds provide a sparse representation of the scene.

Light Detection and Ranging (LiDAR) scanner data has also been used to detect pavement defects [3], [25], [29]–[34]. Although 3D supervised-based algorithms showed superior performance, they rely on a large amount of 3D point cloud that takes a lot of time and effort. Hence, to address this issue a 3D semi-supervised point -level method was proposed by Feng et al [29]. Ravi et al. [33] proposed an automated method to detect pavement defects such as potholes and foreign object debris (FOD) using LiDAR. Understanding how the installation of a LiDAR sensor affects its performance is critical for effective point cloud analysis. For instance, the placement and orientation of the sensor can significantly impact the quality and accuracy of the resulting point cloud data, which are used later in 3D computer vision tasks such as object detection, classification, and segmentation. The field of view of LiDAR, called FOV, is the space area covered during the scanning process for collecting 3D point clouds by the LiDAR sensor. The LiDAR should be installed in the way that its FOV covers the area of interest for the application. Another important consideration is the mounting angle of the sensor. The mounting angle refers to the angle between the sensor and the ground surface, and it can affect the distribution and density of the point cloud data. A mounting angle that is too high or too low can result in missing or sparse data in certain areas, which can affect the accuracy of object detection and segmentation. The best-optimized installation parameters, such as height and the rotation angle, are presented by Lin et al. [34] using low-channel LiDAR for crack detection in highway and urban areas. Low-altitude unmanned aerial vehicle light detection and ranging (UAV LiDAR) and random forest classification (RFC) were used for defect classification [35]. Low-altitude unmanned aerial vehicle light detection and ranging is an unmanned vehicle equipped with a LiDAR sensor operating at low altitudes for collecting 3D data. Random Forest Classification (RFC) is a machine-learning algorithm which combines multiple decision trees to make predictions. The final classification decision is made based on the majority vote of the individual decision trees. High-quality 3D point cloud datasets collected with LiDAR scanners are used to train neural networks such as PointNet for defect classification [30].

Moving Least Squares (MLS) point clouds acquired with Mobile Laser Scanning systems have been clusters to detect cracks [24]. MLS point clouds are a smoothed representation of the original 3D point cloud data. As a result, owing to noise removal from the point clouds, MLS point clouds provide a more accurate representation of the underlying surface. However, one common challenge with MLS data is the lack of topology, which refers to the absence of information about the relationships between points in the point cloud. In traditional point cloud analysis, topology is often defined by the connectivity between points. This topology provides a geometric representation of the object or scene being scanned. Thus, to handle this issue, a two-dimensional index is assigned for each 3D point by Zhong et al. [25].

The Microsoft Kinect camera generates a combination of depth map of the scene using an infrared sensor and RGB color image that can be used to detect defects [2], [26]. Depth images collected by Kinect camera were used for pothole imaging [27]. The Kinect camera's main drawback is that ambient light sources can interfere with its depth sensor. Other approaches used the stereo image technique to reconstruct 3D point clouds from corresponding 2D points extracted from two images captured by two cameras with a sole translation to each other in one axis. Then, the reconstructed point clouds were used to detect cracks [5]. A 3D crack segmentation algorithm has been used to segment the cracks from the 3D point clouds [28].

Since the 3D point cloud annotations should be done by humans, these methods' annotations are intensively time-consuming and need some software skills to do; additionally, capturing high-quality 3D data can be challenging and resource-intensive [41].

2.4 Deep Learning Methods

Most of the recent studies on defect applications applied methods based on deep learning models because of their superior performance and lower computational costs compared to the other techniques [1], [20], [42], [43].

2.4.1 Supervised Methods

2.4.1.1 Object Detection

Some proposed supervised methods localize the defects in terms of object detection with bounding box shapes that can be categorized into two groups: one-stage and two-stage object detection approaches.

(a) One-stage detectors:

One-stage detectors, as exemplified by YOLO (You Only Look Once) and SSD (Single Shot Detector), are a single-step process where the model directly predicts the object class and the bounding box coordinates in a single feed-forward pass over the image without generating region proposal. The network divides the image into a grid and predicts the class probabilities and bounding boxes for each cell in the grid. YOLO deep learning framework developed by Redmon, et al. [55] has been used for defect detection [44]–[54]. YOLO is a deep learning model for object detection. Although it can detect objects in real-time, it has more missing small detected objects compared to other detection methods owing to the lack of complex pipeline. As illustrated in Figure 2-1, in the first step, it divides the image into some grid cells in which a class object is localized, and its class is predicted. In the next step, a bounding box regression is used to find the best-enclosed rectangle around the objects with the highest score. As mentioned previously, YOLO is unable to detect small objects. Thus, the crack detection performance of the original YOLO model was improved by adding attention modules, particularly to detect small cracks [45]. Zhang et al. [49] proposed an improved model of YOLO3 by adding a convolutional block attention module (CBAM) to detect cracks. CBAM is an attention module aiming to improve the feature representation of convolutional neural networks (CNNs) by selectively attending to the most informative features. It achieves this by incorporating two types of attention mechanisms, namely Channel Attention and Spatial Attention, into the feature extraction process. In [112] circular smooth label (CSL) method is used to detect the arbitrarily oriented objects.

[113] implemented and combined the CSL method with YOLO5 [114] and presented an oriented bounding box object detection called YOLO5 (OBB). In [113], each object is defined with six parameters: (cx, cy)(center of the rectangle), width (rectangle width), height (rectangle height), angle (rectangle rotation angle), and object class. An improved version of YOLO was has been also proposed for defect detection by using BFPN (Bidirectional Feature Pyramid Network) [46]. BIFPN is a modification of the Feature Pyramid Network (FPN) architecture, which is designed to address the problem of scale variation in object detection by generating a multi-scale feature pyramid. The feature pyramid consists of a set of feature maps at different scales. The resolutions of the feature maps are increased from the top to the bottom of the pyramid. BIFPN extends FPN by introducing a bidirectional pathway that connects adjacent levels of the pyramid.

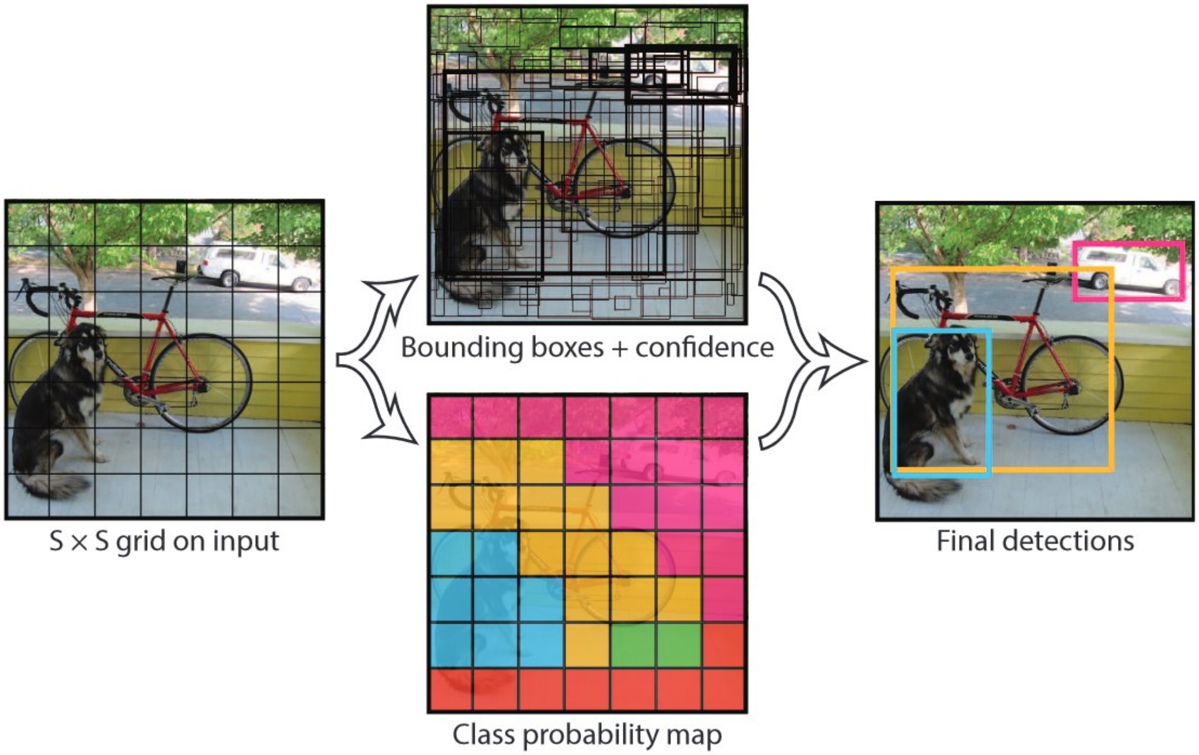


Figure 2-1 YOLO object detection [55]

(b) Two-stage detectors:

On the other hand, two-stage detectors, such as Faster R-CNN (Region-based Convolutional Neural Network), is are two-step process where the model first generates a set of regions of interest (region proposals) using reference boxes (anchors) and then predicts the object class and bounding box coordinates for each region proposal. The network uses a separate Region Proposal Network (RPN) to generate the region proposals, which are then refined by a separate detection network. Thus, the two-stage object detection has one more stage called region proposal generation compared to the one-stage object detection. One-stage object detection methods are ideal for situations where speed is critical since they are simple, effective, and real-time. On the contrary, two-stage objects detection methods tend to be more accurate, especially for small objects and cluttered scenes, due to their more complex architecture and extra region proposal generator compared to the one-stage object detection approaches. Additionally, they are not as real-time as one-stage methods due to their higher computational costs. In other words, the application's specific needs, such as accuracy, speed, computational costs and complexity, indicates whether to select one-stage or two-stage object detection methods.

Some other methods used two-stage object detection methods, such as Faster R-CNN, to detect cracks [57]–[63]. Faster R-CNN is a popular deep learning-based object detection model that was introduced by Ren et al. [64]. Faster R-CNN builds upon the Region-based Convolutional Neural Network (R-CNN) and Fast R-CNN models, which were previously developed for object detection. Faster R-CNN is designed to improve the speed and accuracy of these models by introducing a Region Proposal Network (RPN) that shares convolutional features with the object detection network. As illustrated in Figure 2-2, the RPN generates region proposals by sliding a small network over the convolutional feature maps of the input image. The network is designed to output a set of bounding box proposals, each of which is associated with a score that reflects the likelihood of the proposal containing an object. The top-scoring proposals are then fed into the object detection network for classification and refinement. The object detection network in Faster R-CNN is typically a Fast R-CNN model that uses the same convolutional features as the RPN. The network takes the region proposals generated by the RPN and uses them to crop and warp feature maps from the convolutional feature maps. The cropped feature maps are then fed into a sequence of fully connected layers for object classification and bounding box regression.

Li et al. [58] proposed a UAV-based crack detection using Faster R-CNN. Gou et al. [61] have developed a neural network based on the Faster-RCNN with improved feature extraction and region proposal. Faster R-CNN has also been used to automate pavement defect detection using images from the satellite imagery web-mapping service (Google Maps) [62]. To achieve a high-efficiency pavement crack detection, Zhai et al. [63] improve Faster-RCNN in two ways: (1) by introducing residual connections in the convolutional layers of the backbone, classification and regression network, which enable the learning of more complex features; and (2) by designing a feature ensemble structure for Faster R-CNN, which combines the shallow and deep feature maps of the backbone. This approach involves selecting a subset of feature maps from each layer of the backbone network and concatenating them to form a new feature map. The resulting feature map is then used as input to the object detection network, improving the overall accuracy of the model.

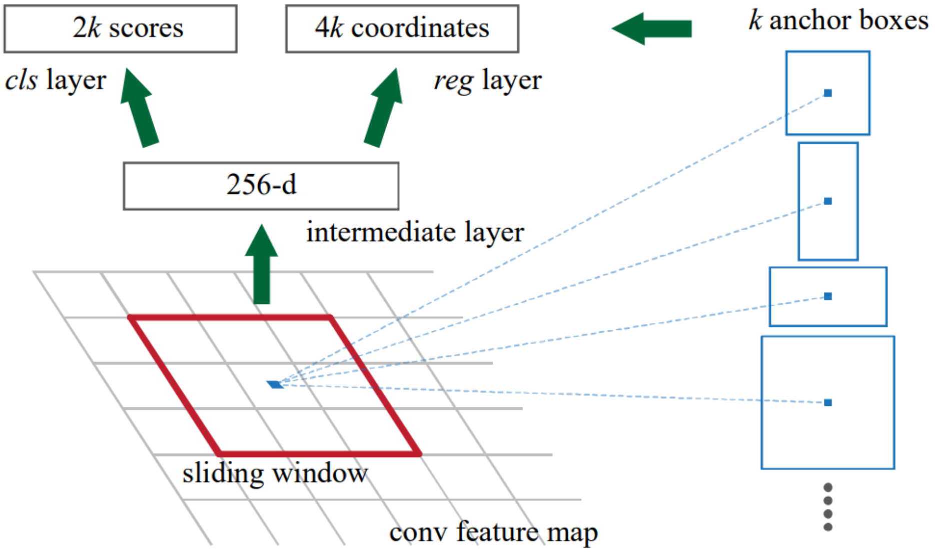


Figure 2-2 Region proposal network (RPN) in the Faster R-CNN [64]

2.4.1.2 Segmentation Methods

Some proposed supervised methods localize the defects in terms of pixel-wise segmentations shapes using deep learning segmentation models, such as the Mask R-CNN, U-Net and SegNet [65]–[78].

(a) Mask R-CNN

Mask R-CNN is a neural network architecture that builds upon the Faster R-CNN object detection model to also perform instance segmentation. Instance segmentation is a computer vision task that involves detecting objects in an image and identifying their precise boundaries at the pixel level. Mask R-CNN extends Faster R-CNN by adding a third branch to the network that outputs a binary mask for each detected object. As illustrated in Figure 2-3, the mask branch takes as input the region of interest (RoI) proposals generated by the RPN and generates a binary mask that specifies which pixels in the RoI belong to the object and which do not. The mask branch consists of a set of convolutional layers followed by a fully connected layer that outputs a binary mask of the same size as the RoI. The output mask is then resized to the original size of the input image and combined with the object classification and bounding box regression results to produce the final instance segmentation output. Compared to Faster R-CNN, Mask R-CNN adds additional computational overhead to the model due to the mask branch. However, it also provides more detailed information about the objects in an image by allowing for precise pixel-level segmentation. This additional information can be beneficial in applications where it is important to identify the exact boundaries of objects in an image, such as cracks.

An improved Mask R-CNN with optimized RPN was proposed to segment asphalt road cracks by replacing Soft Non-Maximum Suppression (Soft-NMS) by None-Maximum Suppression (NMS) [68]. NMS is a traditional post-processing algorithm used in object detection to eliminate redundant bounding boxes generated by a model. It works by selecting the bounding box with the highest confidence score and suppressing all other overlapping boxes with a certain threshold. The overlapping boxes are suppressed by setting their confidence score to zero, which removes them from consideration for the final output. On the other hand, Soft-NMS is a variation of NMS that retains some of the suppressed bounding boxes by reducing their confidence score instead of setting it to zero. This is achieved by using a Gaussian function to gradually reduce the confidence score of the suppressed bounding boxes based on their degree of overlap with the selected box. Hence, Soft-NMS can help retain important object details that may be lost with traditional NMS, especially in cases where small overlapping boxes may contain additional information. This can lead to more accurate object detection results with fewer missed detections.

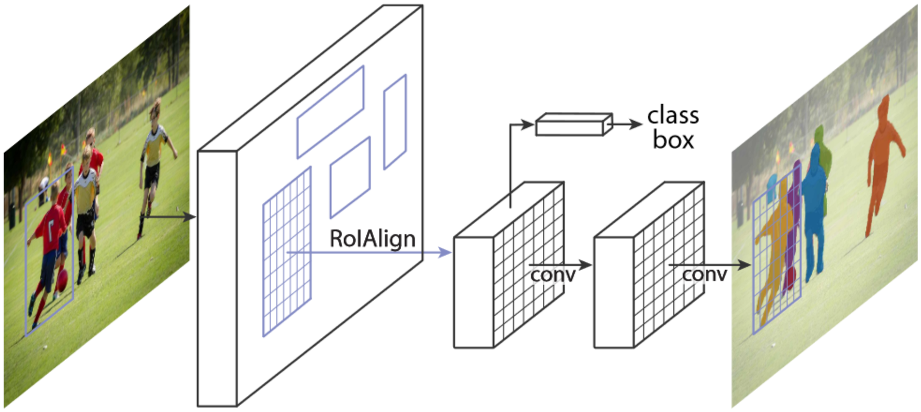


Figure 2-3 Mask R-CNN framework for instance segmentation [79]

Li et al. Xiao [72] proposed a new convolutional neural network based on the Mask R-CNN called C-Mask Region-based convolutional neural network (R-CNN) to identify pavement defects, such as transverse, longitudinal and alligator cracks, by adjusting the anchor ratio and using cascaded detectors with multiple intersections over union (IOU) thresholds. Joint training process for Mask R-CNN and Faster R-CNN has also been proposed to improve the defect detection [57].

(b) U-Net

Other methods used encoder-decoder-based segmentation networks such as U-Net and SegNet to segment the cracks. U-Net was specifically designed for biomedical image segmentation [80], and later it has also been used for other segmentation tasks, such as crack detection in civil engineering [73]–[76], [81]–[83].

As illustrated in Figure 2-4, U-Net architecture consists of two main parts: the contracting path and expanding path. The contracting path is similar to the encoder part of an encoder-decoder network. It comprises several layers of convolutional and max-pooling operations used to extract high-level features from the input image while reducing its spatial resolution. The expanding path is similar to the decoder part of an encoder-decoder network. It consists of several layers of up-convolutional and concatenation operations, which are used to increase the spatial resolution of the feature maps and perform pixel-wise segmentation. The up-convolutional layers are used to up-sample the feature maps and produce a segmentation map of the same size as the input image. U-Net also includes skip connections that connect corresponding layers of the contracting and expanding paths. These skip connections help to preserve spatial information and enable the network to learn both global and local features. Specifically, the output feature maps of each layer in the contracting path are concatenated with the input feature maps of the corresponding layer in the expanding path, before being passed through the up-convolutional layer.

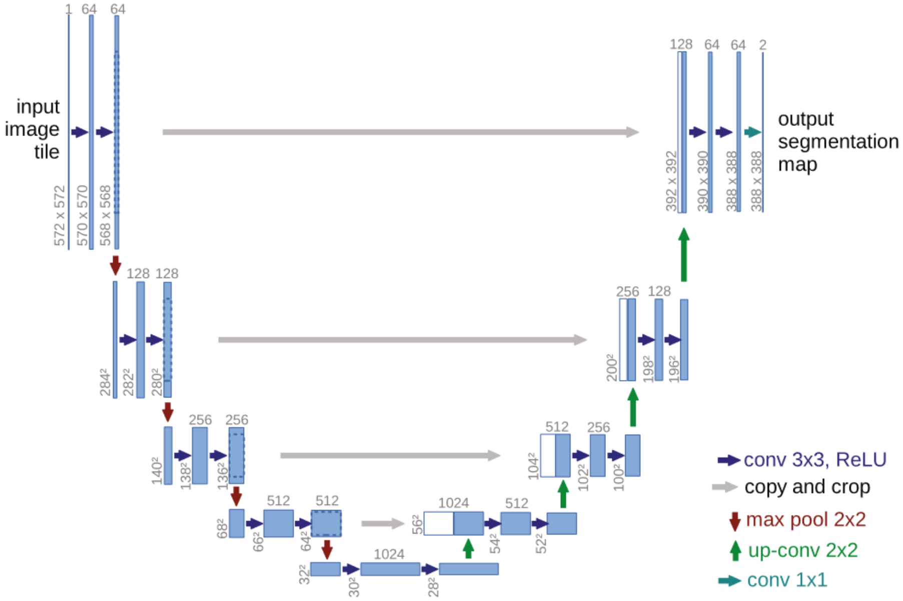


Figure 2-4 U-Net architecture [80]

Liu et al. [81] changed the depth and type of the encoder in the U-Net and considered its effect on not only the accuracy metric but also on computational costs and the model complexity for crack detection. Considering all factors, UNet-VGG19, UNet-InceptionResNetv2, and UNet-EfficientNetb3 have been ranked as the top three encoders. Wang et al. [82] proposed an improved U-Net called I-UNet for crack segmentation, in which dilated convolution instead of regular convolution was used to avoid losing information during the upsampling stage. Yu et al. [83] used residual blocks from the Resnet in the encoder part to improve the model generalization. They also applied an attention mechanism called Spatial-Channel Squeeze and Excitation (scSE) to improve the crack segmentation performance. This mechanism was designed to enhance the important features in an image while suppressing the less important ones. The scSE attention module works by reducing the spatial and channel dimensions of the input feature map through global average pooling and two fully connected layers, respectively. This allows the module to compute a global descriptor that summarizes the most important features in the input.

The issue of imbalance in the number of crack and background pixels has been a significant challenge for road crack image analysis. That is because, owing to the larger number of background pixels, the network tends to predict more background labels compared to the crack labels, which results in detection outputs that are completely black. Thus, an imbalanced number of labels leads to more missing detected pixels for thin cracks. To address this problem a U-Net and residual attention module-based network called RAO-UNet was proposed [76]. A novel pixel segmentation was proposed by modifying the original version of U-Net called CrackW-Net for crack segmentation by using a skip-level sampling method [84].

(c) SegNet

Similar to U-Net, SegNet is an encoder-decoder-based architecture for semantic segmentation. They differ in the way they transfer information between the encoder and decoder. SegNet only transfers max-pooling indices from the encoder to the decoder, while U-Net transfers entire feature maps, which requires more memory. This makes SegNet more memory-efficient than U-Net, but U-Net's approach helps to preserve spatial information and combine high-level and low-level features for more accurate segmentation. A segmentation network based on SegNet has been proposed to segment pavement and bridge cracks [77]. In this network, VGG16 without the top layer was used as encoder, which allows the model to be converged faster with lower computational power for pavement crack detection.

A two-step crack detection and segmentation has also been developed by Li et al [78]. First, a crack classification algorithm based on Interleaved Low-rank Group Convolution Hybrid Deep Network (ILGCHDN) was used to classify the images into two groups: (1) image with cracks and (2) image without cracks. ILGCHDN is a type of neural network architecture that combines multiple techniques to improve the efficiency and accuracy of deep learning models. By using low-rank approximation and group convolution, ILGCHDN reduces the computational cost of the convolutional layers. Interleaved execution then allows the network to process multiple layers in parallel, reducing the memory footprint and improving the overall efficiency of the network. Next, a fused network which combined SegNet and dense conditional random field (DCRF) was used to segment the cracks. DCRF is a probabilistic graphical model used for image segmentation and labeling. DCRF can consider both local and global information to make pixel-wise predictions. It models the probability distribution of the labels for all pixels in an image, given the observed image data and any other relevant information, such as the spatial relationships between neighboring pixels. An encoder-decoder neural network has also been developed based on SegNet to segment cracks [77].

2.4.1.3 Hybrid Methods

Some methods applied other imaging techniques, such as Infrared thermal imaging (IRT), to detect cracks since the temperature is different between cracks and pavement surface [86]. An artificial neural network was proposed to calculate pavement condition index (PCI) from the defect types and severities [87]. Some methods classified the whole images [88], whereas others divided images into equal size patches and classified the patches to detect the defects by two concepts: (1) by dividing the image into sub patches, more data can be provided to train the deep learning model; and (2) image localization map can be generated by putting the defect patches altogether [22], [89]–[92]. The Google Net convolutional neural network was used to classify the image patches for crack detection in a nuclear power plant [93]. The features extracted from the patches with the size of (256*256) by the Adaptive Salp Swarm Algorithm (ASSA) were used to classify the defects by ResNet50 [94].

A novel hybrid method was proposed to measure the crack width [95]. A probability fusion-based model has been developed to detect and measure the length and width of the cracks [96]. A combination of CNN and LSTM was proposed for pavement maintenance data [97].

2.4.1.4 3D Deep Learning Methods

3D point clouds and 2D image fusion were proposed to detect cracks [38]. The proposed data fusion has decreased background noise and increased crack distinguishability. A car-mounted ARM-based platform was proposed by Asadi et al. [39] to detect cracks by using RGB-D (RGB color data combined with depth image map) sensor data. Faster R-CNN and YOLO were used to detect the cracks using 3D pavement images [40]. Feng et al. [31] proposed a Stratified Contrastive Learning dual-branched Graph Convolution Network (GCF) to boost the accuracy and efficiency of pavement crack detection. A 3D deep convolutional neural network called Pavement Crack Detection Net (PCDNet), which uses patch-level instead of pixel-level data, has also been proposed to apply to the 3D images for defect classification [36]. The CrackNet-V model was also proposed to detect cracks from the 3D pavement images [37]. Despite the fact CrackNet-V comprised a deeper architecture compared to the CrackNet network, it has a fewer number of parameters than CrackNet; as a result, it generated more promising results with fewer computational costs.

2.4.2 Weakly Supervised Methods

Most of the proposed deep learning-based methods are supervised techniques, which need a large dataset to attain a good results. Thus, the main drawback of supervised deep learning methods is that they would not show their superior performance when they are not trained with a sufficient number of annotated objects. Two major supervised deep-learning-based methods are object detection and object segmentation, both of them need annotation of a large dataset using bounding boxes and polygon shapes, respectively. The problem gets worse when it comes to defect detection. Defect annotation is more challenging task than other types of object annotation because of the following reasons: (1) Due to the various complex shapes of defects, we need a very large dataset to train the models; and (2) In some cases, some defects with different types overlap with each other, which makes it difficult to find a clear boundary to separate them, especially in highly damaged asphalt pavement roads. For instance, Figure 2-5 shows overlapping of longitudinal, transverse and alligator cracks, and potholes in complex forms. Moreover, in such images, there are always some defects that cannot be definitely categorized into one type. Consequently, due to these drawbacks, defect localization using weakly supervised methods is a more appropriate alternative compared to the supervised ones.

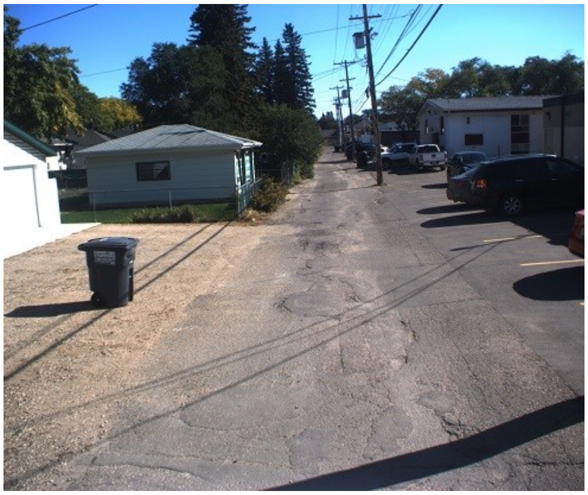


Figure 2-5 A sample image of complex defects for highly damaged asphalt pavements

In general, weakly supervised methods for pavement crack detection refers to a type of machine learning approaches that can localize cracks in pavement surfaces without relying on pixel-level or bounding box annotations of the images. In contrast to supervised methods, weakly supervised learning methods can use easier forms of supervision, such as image-level labels or partial annotations, to train models for detecting cracks in pavement surfaces.

A weakly supervised instance segmentation (WSIS) was developed for crack segmentation by Zhang et al. [98]. They generate pixel-wise pseudo labels from the bounding box-level annotation using the region growing algorithm with the GrabCut algorithm. Inoue et al. [99] developed a weakly-supervised approach to tackle the issue of crack annotation. They proposed a two-branch framework that can maintain a high level of accuracy in detecting cracks even when the annotations are of low quality. Two-branch fused the supervised model rough annotation at the first branch and annotation obtained from the darkness calculation in the second branch inspired by the human annotation. Tang et al. [100] proposed a weakly supervised learning U-Net (WSL U-Net) for pavement crack segmentation using a weakly labelled image in which only marginal pixels were labelled, which was beneficial in reducing the cost of the pixel-wise annotation.

Some methods used class activation mapping (CAM) [101] to localize the pavement cracks [102]–[104]. CAM is a technique used to visualize the spatial information within a neural network that contributes to its classification decision. It can provide the approximate location and shape of the object in terms of which the classifier is trained without the need to use bounding box or pixel-wise annotations. In the CAM method, first, the last fully connected layer of the classifier is removed. Next, a weighted sum of the feature maps generated by the last convolutional layer of the network is calculated. Each feature map is multiplied by its corresponding weight in the final classification layer, and the resulting feature maps are summed up. As illustrated in Figure 2-6, this process results in a 2D heatmap that represents the importance of different regions of the input image for a particular class, which can be segmented for object localization.

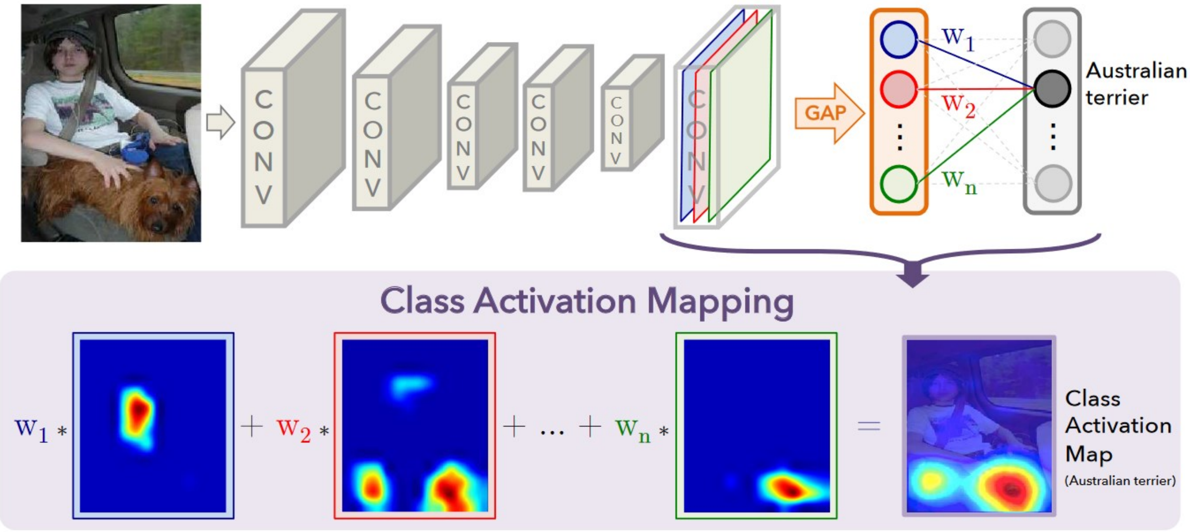


Figure 2-6 Class Activation Mapping [101]

The pixel-level semantic requirement cannot be met adequately by solely utilizing the default image size passed into the classification network, as it fails to activate sufficient features that identify objects. Thus, multi-scale localization using CAM method with self-attention (SA) module was used to address this problem [104]. In the same way, a weakly supervised patch label inference network (WSPLIN) is proposed to tackle this issue by dividing the images with different scales into patches with different collection strategies and classifying them [103]. A patch-based weakly supervised semantic segmentation network has also been developed for crack segmentation [102]. First, the images are divided into some patches. Next, the classifier is trained by their corresponding image labels and CAM method is used to extract the heatmaps of the cracks. Finally, Conditional Random Fields (CRF) are used to generate synthetic labels. CRF is used as a post-processing step to refine the output of a neural network. Neural networks often output pixel-wise or object-wise probabilities for a given task, such as image segmentation or object detection. However, these probabilities are often noisy and can result in fragmented or inaccurate segmentations or object boundaries. However, since they divide an image to small patches, the texture of the crack cannot be kept to classify the defect type.

The smoothed CAM method [109] is an extension of the CAM method, which can also be used to localize the objects. Similar to the CAM method, the smoothed CAM method localizes the defects in the images. This method calculates the first principal component of feature maps multiplied by weights instead of using their average sum. It also augments the images and applies the CAM method multiple times to the combined augmented images. This method reduces the noise of localization and centralizes the detector result at the center of the object [109].

2.5 Crack Severity Estimation and Width Measurement

Some methods are designed specifically for the purpose of estimating the width of pavement cracks. The results obtained from these methods can be used in calculating the Pavement Condition Index (PCI). Ong et al. [115] developed a hybrid method using merging the shortest and orthogonal projection methods to estimate the pavement crack width by identifying the points in close proximity to the skeleton, which are aligned to the orthogonal vector. Wang et al. [116] introduces a novel automatic technique for crack width measurement utilizing two algorithms including the crack blob extraction and crack boundary extraction by applying the Laplacian equation.

Some other methods, in addition to localizing the detects, classify their severities as well. Majidifard et al. [44] gathered 2D crack images from Google Street View and trained the YOLO object detection model with this data to detect cracks in 2D images. However, since YOLO did not quantify the cracks' density, they used the U-Net network for segmentation and quantifying defect density. A two-step crack detection and crack severity classification using Mask R-CNN has also been proposed [66]. In the first step, Mask R-CNN was used to localize and classify defect types for linear cracks, such as longitudinal and transverse cracks. In this step, Mask R-CNN was also used to classify the fatigue defect severity in three levels. In the second step, a combination of image processing techniques, such as connected component analysis, morphological operation (dilation and erosion), Gaussian filter technique, were applied to segment the linear defects more precisely and estimate their severities. However, the aforementioned image processing methods are beneficial for high-resolution pavement images. For images with lower resolution, such image processing techniques, especially erosion, deforms the crack's structure. In addition to the defect type, defect severities were estimated by calculating the maximum crack width in linear cracks and the density of cracks in area cracks [85].

Automatic crack width estimation can be done by analyzing the images with image processing and computer vision tools such as Interactive Segmentation with Intelligent Scissors (IS) method [110].

IS is an intuitive and user-friendly method that automatically detects object edges in an image. This method was introduced by Mortensen et al. [110] and was primarily designed to enable interactive boundary delineation of objects of interest in images.

The algorithm makes use of the idea of "costs" or "weights" for each pixel, which are calculated based on various features such as pixel intensity, gradient, and direction. The IS behaves like a "live wire" that snaps onto the object boundary based on these cost values. Seed points are fed to the algorithm which then seeks the path of least cost between the points, effectively delineating the object.

IS can handle small breaks in object boundaries. They are particularly useful for delineating objects with complex, irregular shapes. They are often used in applications like medical imaging, where accurate boundary delineation is crucial. Figure 2-7 shows a sample of delineating the intricate boundary of an object against a complex background using the IS method.

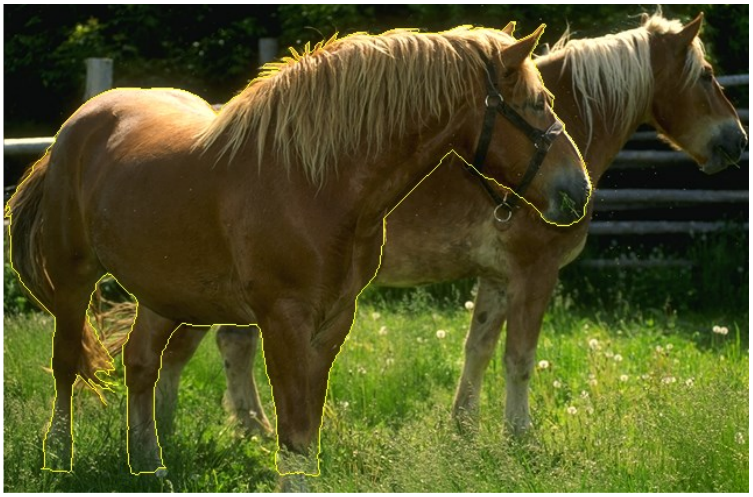


Figure 2-7 Delineating the intricate boundary of an object against a complex background using the Intelligent Scissors method [110]

2.6 Summary and Conclusions

This chapter delved into the existing body of literature concerning crack detection, localization, and severity estimation techniques. Moreover, it identified the drawbacks associated with these methods. Although classical image processing and computer vision methods are often computationally efficient and can work well in scenarios with limited computational resources, they struggle with handling large-scale datasets, as they may lack the scalability and adaptability of modern deep learning approaches. Providing a more comprehensive understanding of the three-dimensional world through 3D computer vision techniques leads to more accurate object localization and shape estimation. However, it relies on specialized equipment, such as depth sensors or stereo cameras, which may limit their applicability and increase the cost of implementation. Ultimately, a framework based on deep learning is taken into consideration. The examined methods encompassed consideration of various attributes, such as the classification of defect types, localization of defect through supervised or weakly supervised approaches, including defect detection and segmentation, as well as defect severity estimation.

Table 2-1 summarizes the pavement crack detection and recognition methods discussed in this chapter as well as the proposed method. As shown in the table, considering all factors, the proposed hybrid method aims to combine the benefits of the previous methods. By integrating both supervised and weakly supervised methods to localize defects and categorize their types and severities.

Table 2-1 Summary of pavement crack detection and recognition papers based on deep learning

Methods	Defect type classification	Defect localization (detection/segmentation/patch classification)		Defect severity classification or estimation
		Supervised method	Weakly supervised method	
[98], [99], [100], [102-104]	No	No	Yes	No
[22], [39], [49-54], [57], [61], [62], [65], [67]–[69], [70-71], [73]–[78], [81], [82], [84], [86], [89]–[93]	No	Yes	No	No
[45]–[48], [58], [60], [63], [72], [94]	Yes	Yes	No	No
[44], [66], [85]	Yes	Yes	No	Yes
[115], [116]	No	No	No	Yes
The proposed method	**Yes**	**Yes**	**Yes**	**Yes**

Chapter 3: Proposed Method

3.1 Introduction

The proposed framework consists of deep learning-based classification and localization modules integrated with a GIS-based inspection and asset management system called RUBIX [117]. The proposed hybrid deep learning modules benefit from the advantages of both supervised and weakly supervised methods. It is also shown that, in general, considering all factors, the proposed method outperforms the supervised methods in defect localization and classification.

Figure 3-1 shows the block diagram of our proposed method. In the classification module, images and extracted patches from localization are classified in terms of defect type and severity, respectively. The defects' localization module, first, feature maps are extracted from the defect type classifier. Then, these feature maps are used as weakly supervised features to localize the defects using Class Activation Mapping (CAM) (16). Finally, U-Net or Mask R-CNN are used to segment the defects from the CAM method heatmaps as supervised methods; and the segmented patches are sent to the second-level classifier to classify their severities.

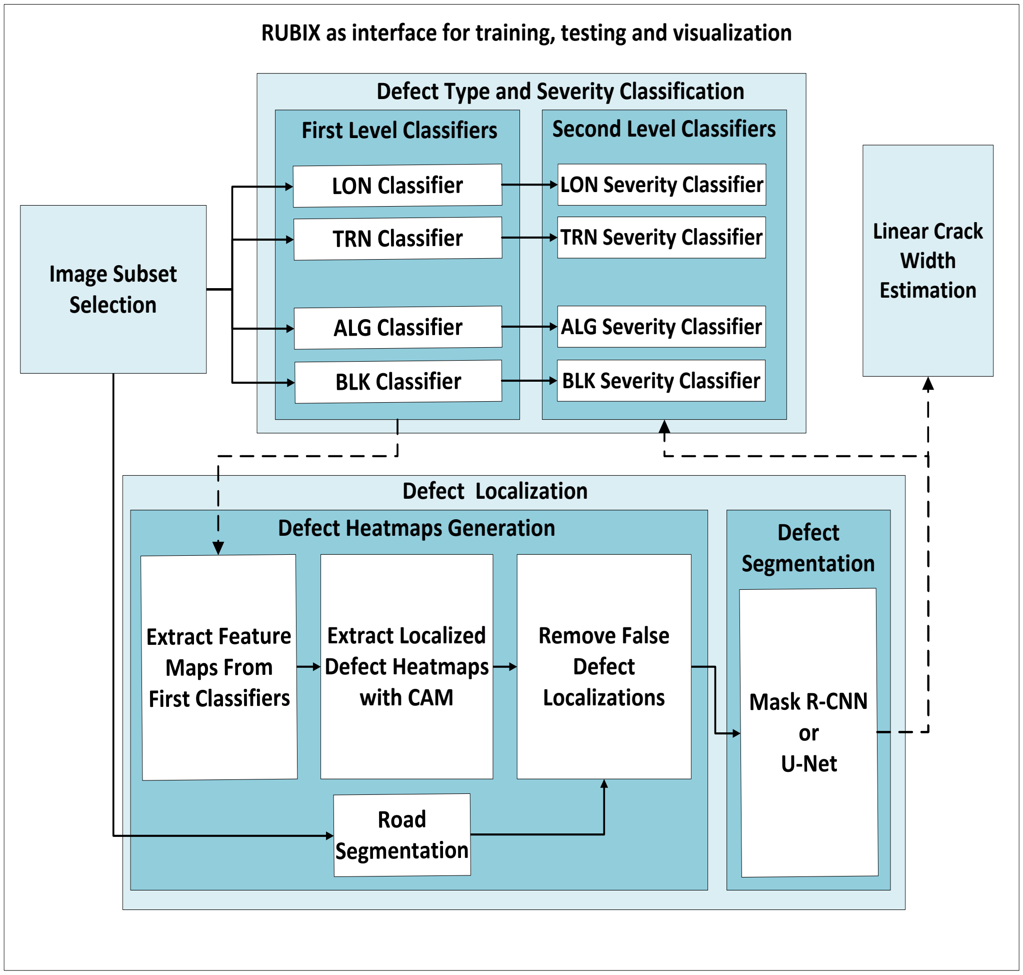
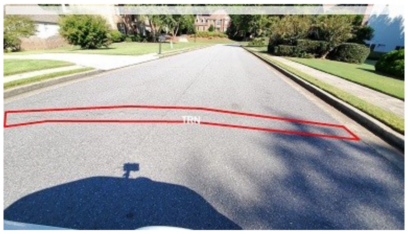
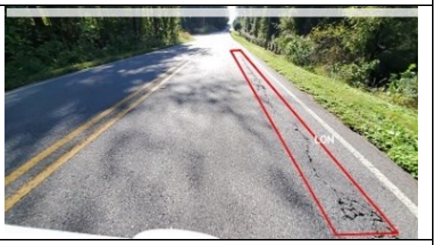
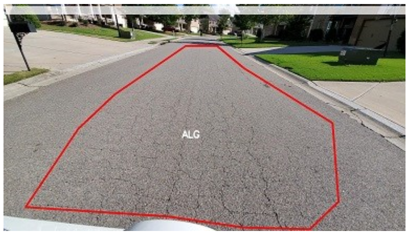
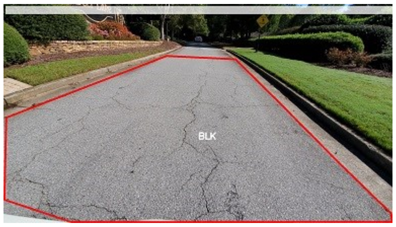


Figure 3-1 Proposed method

Table 3-1 Brief explanation of the selected defects

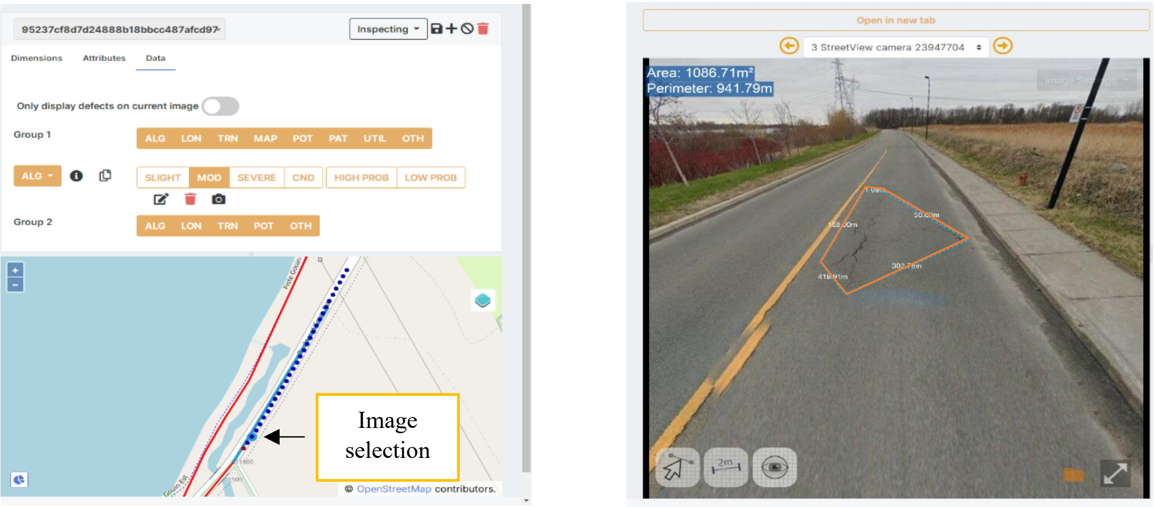
Defect type	Explanation	Abbreviation	Example
Transverse crack	Transverse defects are predominantly located perpendicular to the pavement centerline.	TRN	
Longitudinal crack	Longitudinal cracks are predominantly parallel to the pavement centerline.	LON	
Alligator crack	Alligator cracks mostly occur in areas subjected to repeated traffic loadings (wheel paths). It is a series of interconnected cracks in the early stages of development. It develops into many-sided, sharp angled pieces.	ALG	
Block crack	Block crack is a pattern of cracks that divides the pavement into approximately rectangular pieces.	BLK	

Hence, this hybrid framework combines the advantages of weakly supervised and supervised methods. In this research, four types of defects (i.e. longitudinal (LON), transverse (TRN), alligator (ALG), and block (BLK) cracks) are localized in pavement images. Table 3-1 provides a brief explanation and illustration of the selected defects.

3.2 Pre- and Post-Processing Module (RUBIX)

RUBIX is a GIS driven asset inspection and management system that integrates field data collection tools, dashboard reporting and decision-making aiding for long-term maintenance and rehabilitation planning. RUBIX has tools to perform: (1) visual inspection from field captured images, (2) PCI calculation to report the current infrastructure status, and (3) model degradation to help infrastructure managers with maintenance and rehabilitation planning. For this study, it served as the primary tool for field data collection, and the analysis platform used to annotate defects, train models, and test the datasets for object classification and localization. All captured images with GPS locations were uploaded into RUBIX for condition processing, analysis, and visualization of the generated information. RUBIX is a fully customizable platform and can be easily configured to match the scope of this research. As part of the pre-processing, each image is assigned to a road section, defined by a centerline in the GIS source map dataset. A road section in the RUBIX map is selected to extract the images for annotation in terms of defect types and severities. Figure 3-2 shows an example of image selection (Figure 3-2(a)) and annotation in the RUBIX GIS interface (Figure 3-2(b)).

Following the user's request, RUBIX generates the URL encoding query of the selected images and uses the hyperparameters specified by the user to start the training or testing processes. RUBIX executes a job for each process, puts the processes in a queue, and runs them based on their priorities. Thus, by creating a query for images in arbitrary locations and using the pre-trained models, the conditions of roads can be evaluated based on the types and severities of defects so that proper future pavement maintenance measures can be applied.



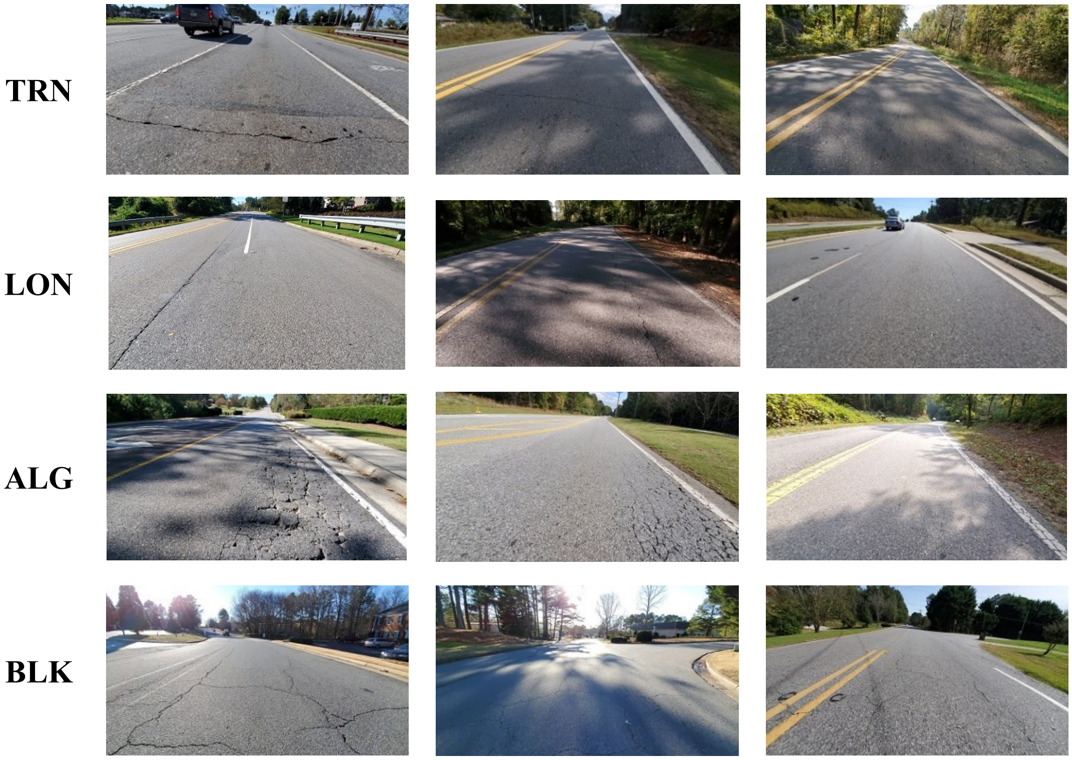
(a) Image selection **(b) Image annotation**

Figure 3-2 RUBIX image selection and annotation

3.3 Classification Module

In the classification module, the images and their corresponding generated localized patches are passed through two-level consecutive hierarchical classifiers to classify the type and severity of defects using the Residual Neural Network (Resnet101) [105]. It uses a residual network that can be extended to a deeper scale with much more hidden layers without suffering from vanishing gradient and optimization problems. The first classifier indicates whether there is a specific type of defect in the image or not. If it classifies the image as an image with a specific defect type, the feature maps extracted from the first-level classifiers are used to localize and segment the defect patches, as will be explained in Section 3.4. Next, the patches extracted by the localization module are passed to the second-level classifier to identify the severities of the defects, which play a key role in pavement condition evaluation. In the first-level classifier for each defect, images are categorized into the two groups: the group that includes the images with a specific type of defect, and the group of images without that defect, but may contain other types of defects. In the first-level classifier, each type of defects is classified by its assigned classifier separately. Thus, four classifiers are used to classify the four defect types.

In the second-level classifier, three defect severity levels are defined (i.e. high, medium, and low severity) for each defect. Figure 3-3(a), (b) and (c) show sample high, medium, and low severity levels for each type. Hence, four classifiers specify the severity of the defects for each defect type.



(a) High severity **(b) Medium severity** **(c) Low severity**

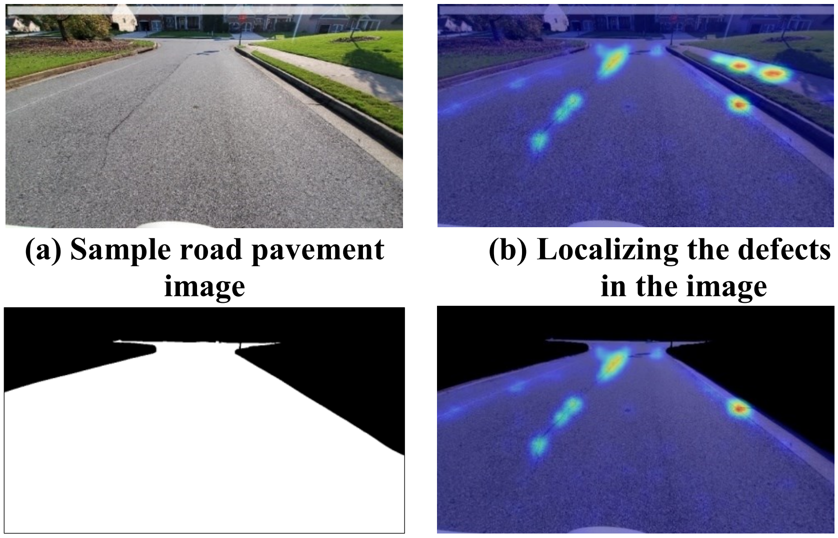
Figure 3-3 Samples of transverse, longitudinal, alligator and block cracks for three severity levels

3.4 Localization Modules

After identifying the defects and their types by the classifiers, feature maps extracted from the first classifiers are segmented to localize the classified defects using two steps. In the first step, the CAM method is used as a weakly supervised method to localize the defects in heatmaps from the feature maps. In the second step, U-Net or the Mask R-CNN are applied as supervised methods to generate the segmented defect images from the heatmaps.

3.4.1 Generation of Defect Heatmap

After classifying the defects, the feature maps from the first classifiers are used to localize the defects in the given images using the CAM method. Although the CAM method extracts discriminative features, which occur mostly within the object region, sometimes it highlights other regions outside of the region of interest (ROI), which is the road area. Thus, in order to handle this challenge, road segmentation is used to reduce the false localizations outside the ROI. Hierarchical Multi-Scale Attention for Semantic Segmentation method presented in [106] is used to segment the road in pavement images [107]. Figure 3-4 (a) shows a sample road pavement image. Figure 3-4 (b) shows localizing the defects in the image. Figure 3-4 (c) shows the road segmented image. Figure 3-4 (d) shows the segmented localized image to reduce false localization.



(a) Sample road pavement image

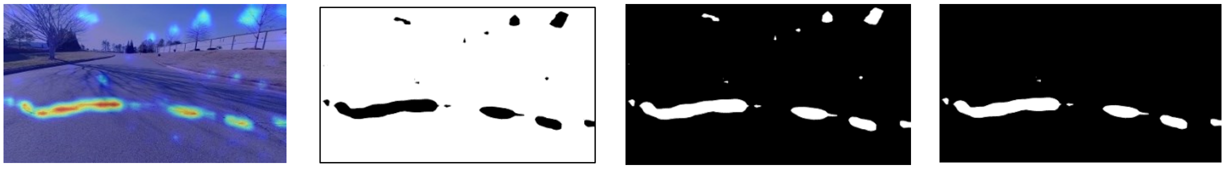
(b) Localizing the defects in the image

(c) Road segmented image

(d) Segmented localized image

Figure 3-4 Reducing false localized defects by using the road segmentation image

To segment the defective area in the localized image, first HSV (Hue, Saturation, Value) color space filtering within the range of (0, 0, 100) to (255, 255, 150) is applied to extract the blue area. Then, the segmented mask is inverted to extract parts that include defects. Finally, road segmentation is used to remove falsely segmented parts outside of the road. Figure 3-5 shows the procedure of CAM result defect segmentation explained above for a sample image. As can be seen in the Figure 3-5 (a), another drawback of CAM method is that it does not detect the whole shapes of objects, especially when the objects are complex, such as pavement defects. This incomplete detection leads to discontinuity in segmented localized images (Figure 3-5 (d)).



(a) Localized defect image with CAM method

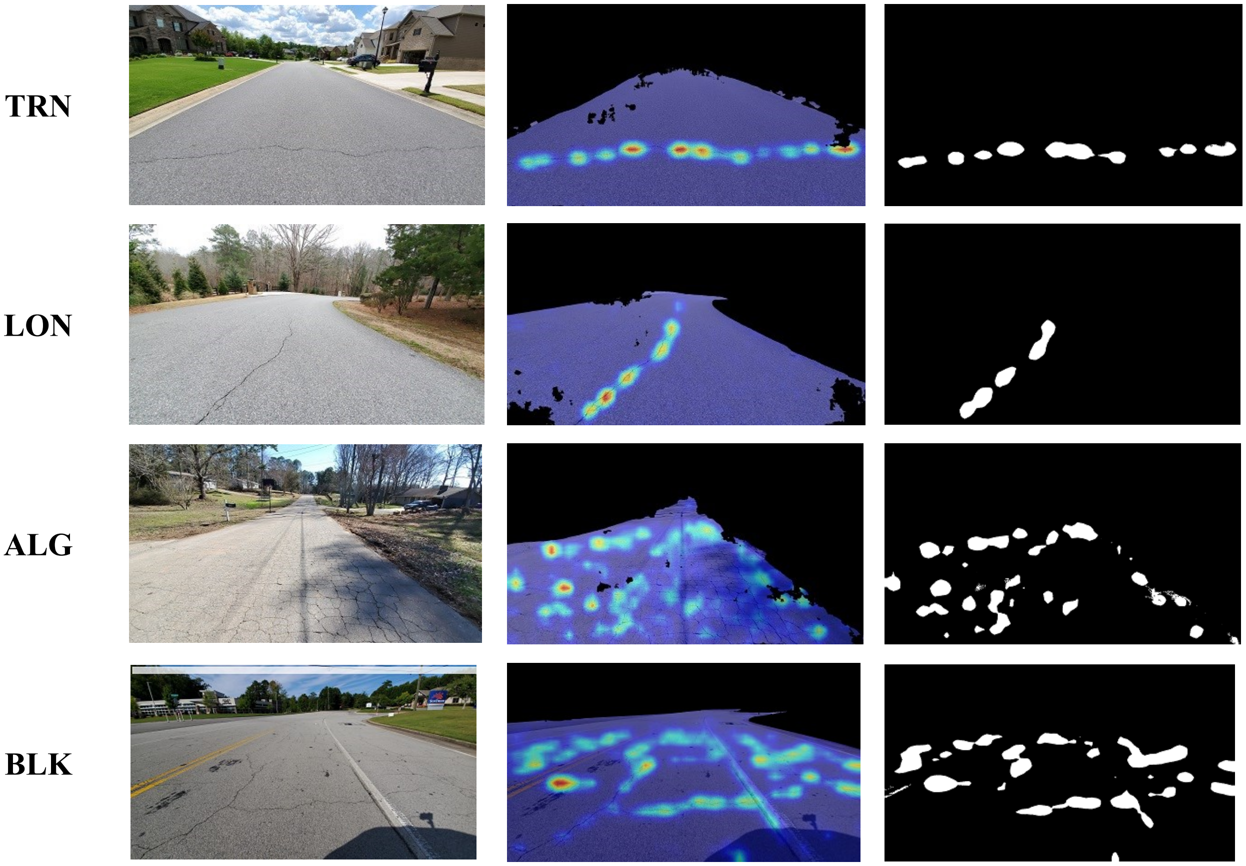
(b) Segmented defect image by applying HSV filtering

(c) Inverted segmented defect image

(d) Removed false segmented parts by road segmentation

Figure 3-5 A sample of CAM result segmentation

This problem is less severe for linear defects, such as longitudinal and transverse cracks, but gets worse in the case of area defects, such as alligator and block cracks. As can be seen in Figure 3-6, the CAM method fails to extract all parts of defects. Figure 3-6(a), (b) and (c) show original images, their corresponding CAM localization results and segmented images for each defect type, respectively**.**



(a) Original image

(b) Localized defect image

(c) Segmented defect image

Figure 3-6 Samples of CAM method localization and segmentation

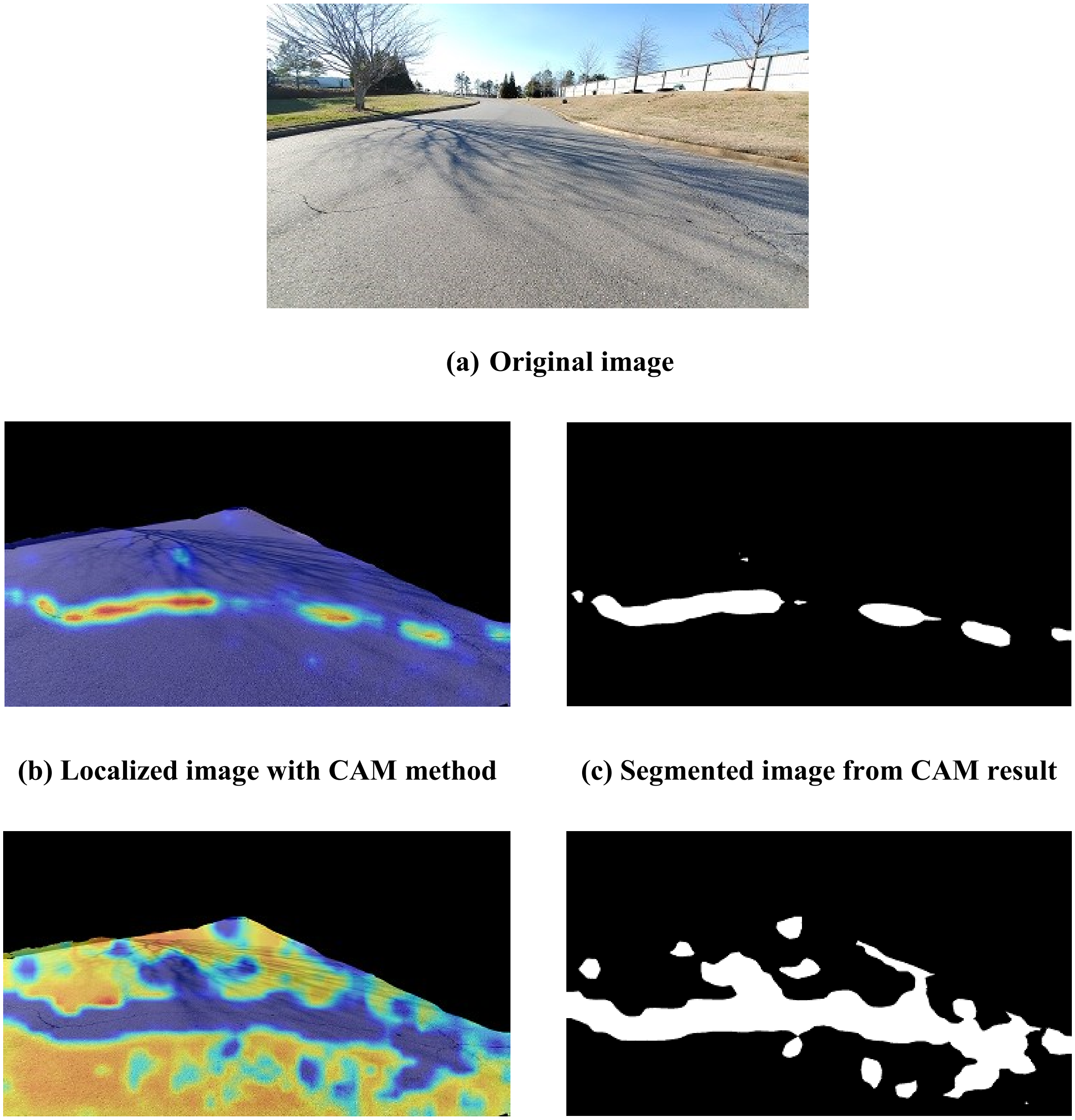
As mentioned in Section 2.4.2, the smoothed CAM approach [109], which serves as an extension of the CAM technique, can also be utilized for object localization. It enhances the images and implements the CAM method on the assembled augmented images several times. This strategy diminishes the noise associated with localization and centers the detector's outcome at the core of the object.

In order to segment the defective areas in the smoothed CAM result, the same range of HSV filtering as mentioned for the CAM method is employed, facilitating the extraction of the blue area. Figure 3-7 shows a sample of CAM, smoothed CAM localization and their corresponding segmented images for a transverse crack. As can be seen in Figure 3-7 (d), the smoothed CAM method has detected all parts of the defects, and its segmentation result (Figure 3-7 (e)) does not suffer from the discontinuity that was common in the results of the CAM method. However, it has two drawbacks: (1) the computation time of the smoothed CAM method is six times longer than the regular CAM method; and (2) its segmentation results include portions of the background and are not accurate. Hence, manual modification is necessary to cleaned them up at some parts. Table 3-2 provides a comparative summary between the CAM and the smoothed CAM methods.

As can be seen in the table, the CAM method is faster than smoothed CAM method and does not include portions of background as the smoothed CAM does. Thus, we decided to use CAM method to localize the defect primarily and solve its discontinuity by adding the supervised segmentation module explained in the next section. Additionally, although the smoothed CAM method comprises some portions of the background, since it does not suffer from discontinuity, by manually eliminating the overlapping results with the background, the smoothed CAM method can be effectively utilized to generate ground truth data for the proposed segmentation modules.

Table 3-2 Comparison of CAM and smoothed CAM method

Aspect	CAM	Smoothed CAM
Speed	Faster	Six times slower
Continuity	Suffers from discontinuity	Dos not have discontinuity problem
Background object	Does not include any part of background alongside the object	Includes some portions of the background alongside of the object



(a) Original image

(b) Localized image with CAM method

(c) Segmented image from CAM result

(d) Localized image with smoothed CAM method

(e) Segmented image from smoothed CAM result

Figure 3-7 A sample of CAM and smoothed CAM localization and their corresponding segmented images for transverse crack

3.4.2 Defect Segmentation

In Section 3.4.1, it was illustrated that the CAM method cannot extract all parts of the defects. To address the aforementioned discontinuity problem in the CAM heatmap segmented images, the proposed framework uses U-Net or Mask R-CNN as supervised deep learning segmentation modules to segment the defect more precisely as unified segmented objects from the CAM heatmaps. U-Net or Mask R-CNN are supervised segmentation modules as explained in Section 2.4.1.2.

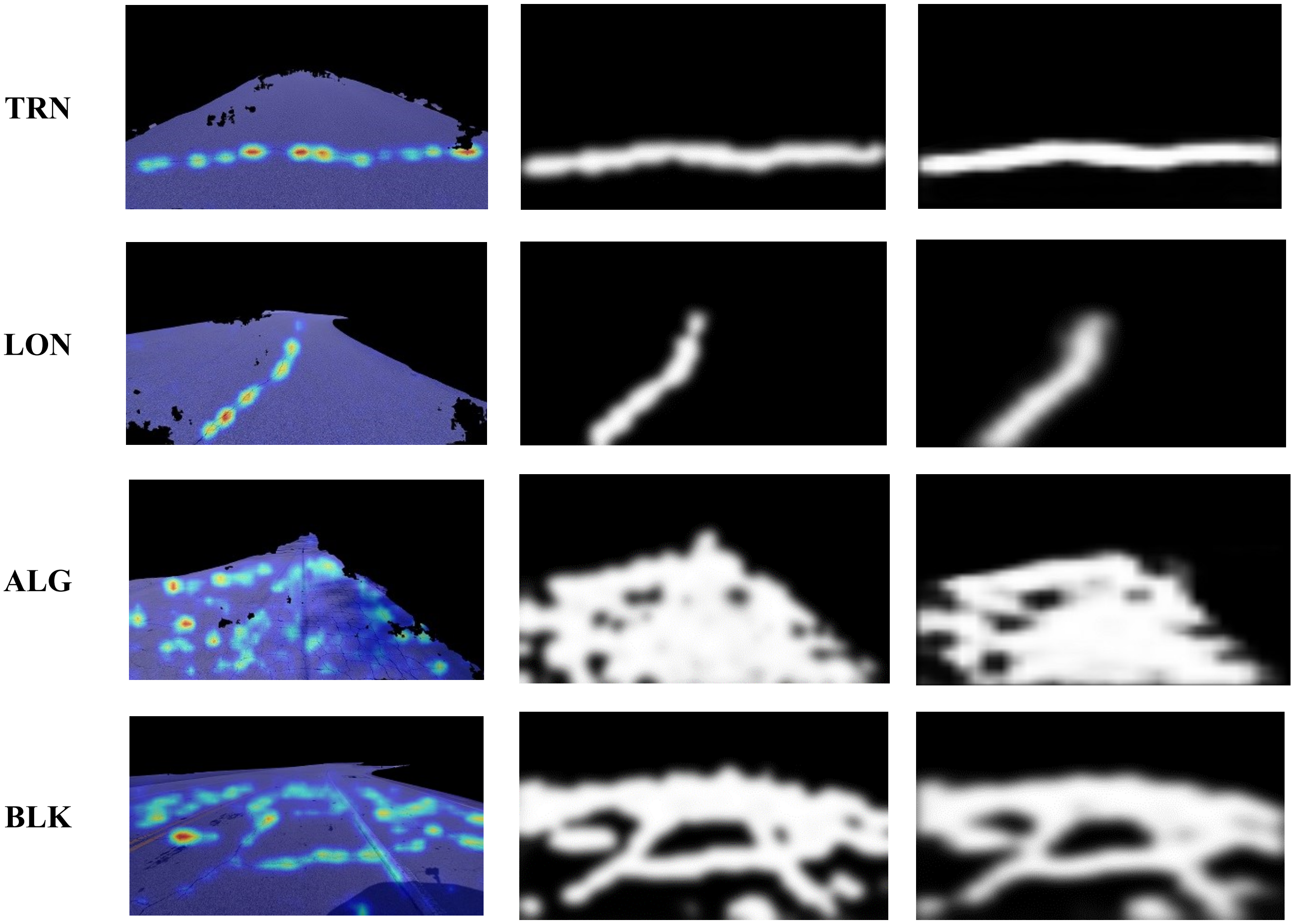
In supervised defect localization methods using segmentation, it is necessary to annotate defects at pixel boundaries on the RGB images. However, this does not allow for specific defect type identification. In the case of instance segmentation, both pixel boundaries and their labels must be annotated for a large number of images with a wide variety of defects on RGB images, which can be a laborious and time-intensive task.

However, as can be seen in Figure 3-1, the proposed method uses a high-level feature images (CAM heatmaps segmented by road segmentation) instead of RGB images to train the segmentation module. As can be seen in Figure 3-7 (a) and Figure 3-7 (b) defects are more noticeable and distinguishable in CAM heatmap compared to the original image. Thus, the segmentation module can learn more efficiently from the pronounced features in the CAM heatmaps instead of RGB images that is used in previous research for the training of the supervised segmentation modules.

The type of defect is classified in the classification module. Therefore, the supervised segmentation module in the proposed framework should be able to segment the defects regardless of their types. It is trained using some defect CAM heatmap results and their corresponding ground truth data. This trained module can then be applied to new datasets without the need for additional training because only the classification module needs to be trained on the new data. Next, feature maps extracted from the classifier, using the CAM method, are then used to segment the defect from the CAM method for new data.

As explained in Section 3.4.1 the smoothed CAM could join the connected defects in the localized image. In other words, the parts of defects disregarded by the CAM method are detected by the smoothed CAM method. However, it has been observed that the segmentation process is somewhat flawed, as it includes parts of the background in conjunction with the actual defects. Consequently, to gather data for training the supervised modules, a semi-automated method is proposed. This method generates segmented results using smoothed CAM, which are subsequently manually adjusted and cleaned up to remove false segmentation parts. These modified results serve as ground truth data for training the segmentation modules as will be explained in Section 3.4.2.1. In the test stage the segmentation module segments the defects from the CAM results.

Figure 3-8 shows samples of generating defect segmentation images from the results of CAM heatmaps for each defect type in the proposed method. Comparing Figure 3-6 and Figure 3-8 demonstrates that in contrast to the CAM method segmentation results, the proposed method does not suffer from the discontinuity problem in defect segmentation.



(a) CAM method feature maps

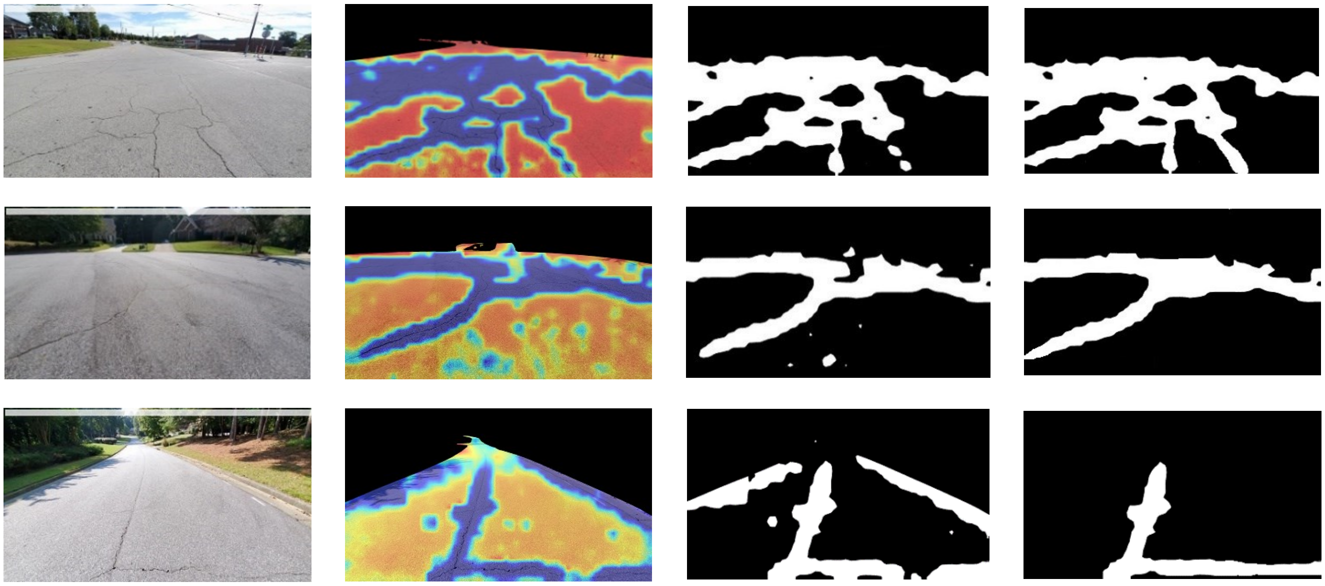
(b) U-Net segmentation results

(c) Mask R-CNN segmentation results

Figure 3-8 Samples of generating defect segmentation images from the results of CAM heatmaps in the proposed method

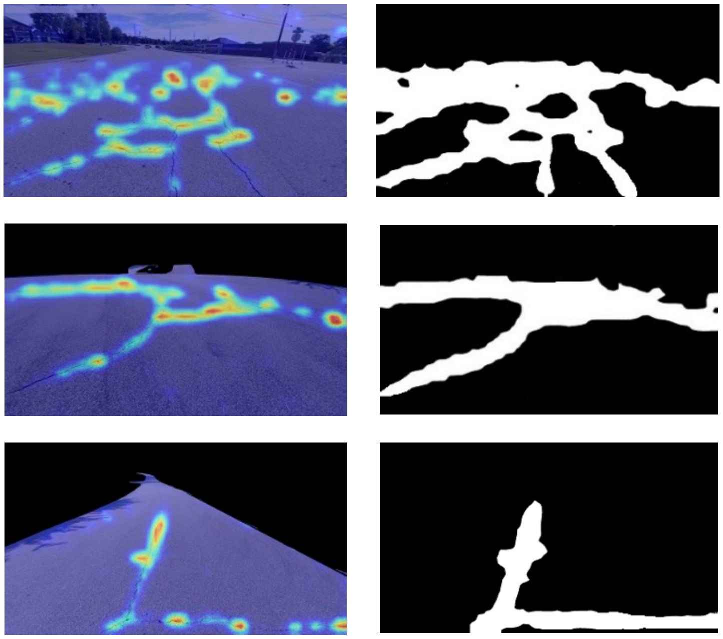
3.4.2.1 Training for Defect Segmentation

Contrary to the process used for training the classification modules, where images are classified into specific defect types and other defect types; Here, first, images are categorized into two groups: images with defects, and images without any defect. Next, the classifiers are trained with these two classes. After that, feature maps from the trained classifier are used with the smoothed CAM method to localize the defects. To generate the segmented ground truth images required for the training of segmentation modules, first, 2,000 best smoothed CAM results were selected. Next, HSV filtering was applied to the smoothed CAM method heatmap results. Segmented ground truth images were generated from the smoothed CAM heatmap results by selecting the pixels with values in the same HSV range explained in Section 3.4.1. Thus, Since the segmented images from the smoothed CAM method include some parts of background, finally, the segmented images were manually adjusted to remove false segmented pixels. Figure 3-9(d) shows the results of manually modification of the segmented images in Figure 3-9(c). These ground truth images and their corresponding CAM heatmap results were used to train U-Net and Mask R-CNN modules. Some samples from the generated ground truth in Figure 3-9 and their corresponding CAM heatmaps are shown in Figure 3-10.



(a) Original image **(b) Localized image with smoothed CAM** **(c) Segmented image by applying HSV filtering** **(d) Manually modified segmented image**

Figure 3-9 Generating ground truth from smoothed CAM heatmap results



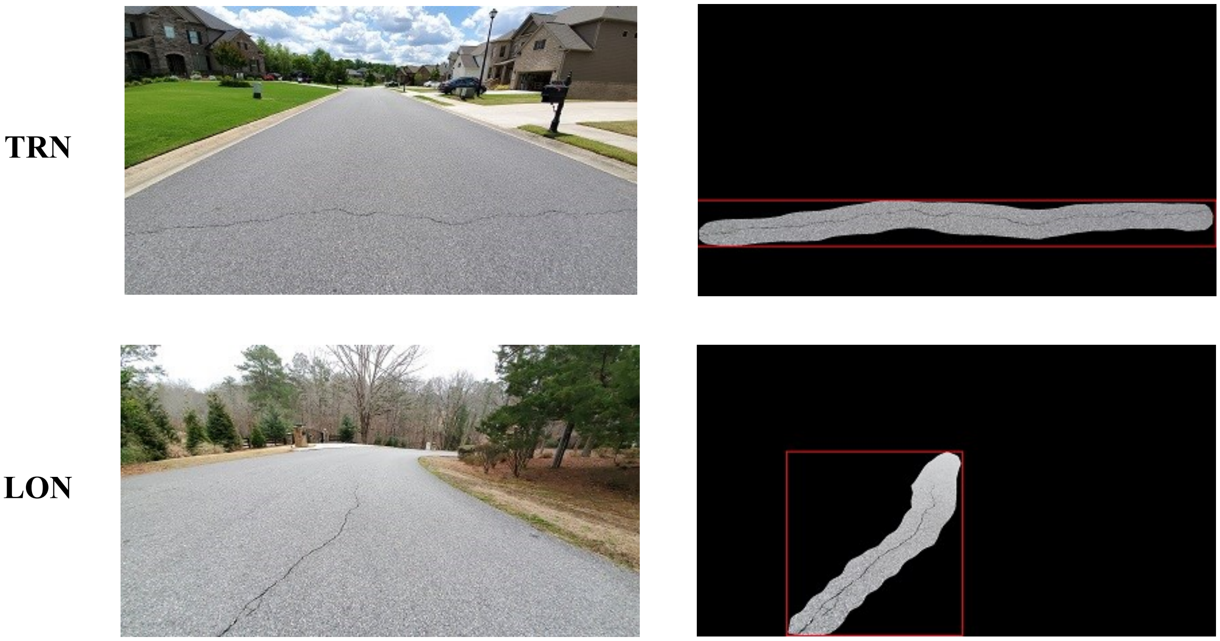
(a) CAM method heatmaps **(b) Corresponding ground truths**

Figure 3-10 CAM heatmap sample images and their corresponding generated ground truths to train segmentation modules

3.5 Linear Crack Width Estimation

As previously explained and as illustrated in Figure 3-10, the ground truth generated from the smoothed CAM method includes sections of the background along with the defect location. Consequently, the segmented results from the proposed method also encounter the same issue specifically for linear cracks, such as longitudinal and transverse cracks.

Figure 3-11 shows two segmented defect patches by the proposed method for transverse and longitudinal crack, respectively. As can be seen in this figure, some portions of the background are segmented around the defects. Given that the estimation of linear crack width is crucial for calculating the PCI score, an innovative method is proposed for determining the crack width from the extracted patches in the next section.



(a) Original image **(b) Segmented patch**

Figure 3-11Two segmented defect patches by the proposed method

Given the proven prowess of the IS method in delineating the boundaries of objects with complex and irregular shapes locally, in this section, a novel method is proposed to identify the maximum crack width in pixels for linear segmented defect patches extracted as explained in Section 3.4 based on IS method [110]. Figure 3-12 shows the block diagram of the proposed method to estimate crack width for linear defects such as longitudinal and transverse cracks.

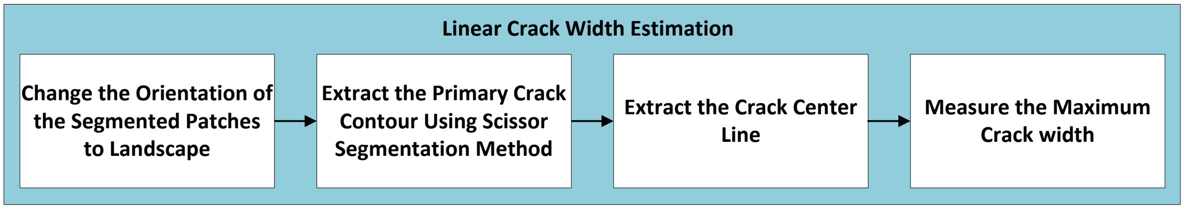
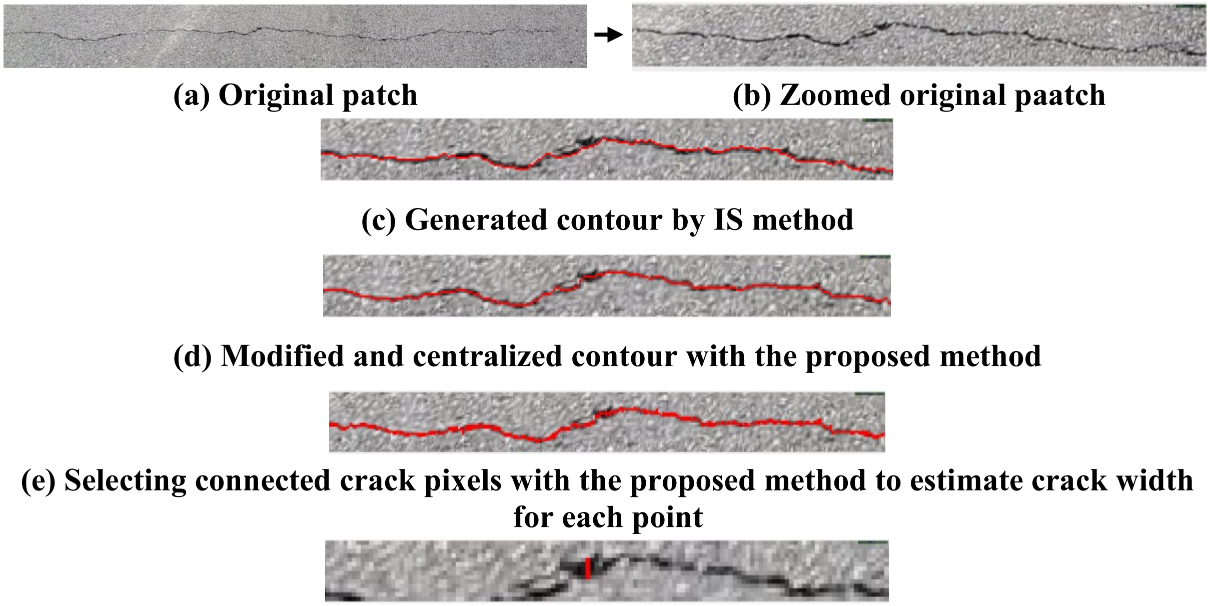

Linear Crack Width Estimation
Change the Orientation of the Segmented Patches to Landscape
Extract the Primary Crack Contour Using Scissor Segmentation Method
Extract the Crack Center Line
Measure the Maximum Crack width

Figure 3-12 Proposed method to estimate linear crack widths

Figure 3-13 shows a sample of crack width estimation. First, the orientation of all extracted segmented patches from Section 3.4 is changed to landscape. Next, primary crack contours are generated from the segmented patches using the IS method. Center points from the beginning and end of the segmented patch is selected as seed points for the IS method.

As can be seen in the Figure 3-13 (c), although the IS method is expected to find the contour of the cracks, in many cases, only one of the edges of the cracks are detached. Therefore, in the third step, the patch is scanned using vertical columns of pixels searching for the darkest pixel near the pixel on the edge. This pixel is considered to be on the centerline of the crack. Figure 3-13 (d) shows the new generated contour which is positioned at the center of the crack.

In the fourth step, for each point, the number of pixels, for which the absolute value of the difference between their intensity and the point's intensity is smaller than a specific threshold, is considered as the crack width value for each point within the same column. Finally, the maximum value derived from these crack width calculations is considered as the estimation of the width of the crack. Figure 3-13 (e) shows the crack width calculation for each point. As can be seen Figure 3-13 (f), maximum crack width value is considered as crack width estimation.



(a) Original patch

(b) Zoomed original paatch

(c) Generated contour by IS method

(d) Modified and centralized contour with the proposed method

(e) Selecting connected crack pixels with the proposed method to estimate crack width for each point

(f) Selecting the maximum crack width value as crack width estimation

Figure 3-13 Sample crack width estimation

By knowing the camera calibration intrinsic parameters, it is possible to convert the crack width in pixel scale to a real-world metric using pinhole camera calculation as shown in Equation (3-1) [118].

The intrinsic parameters include:

1. Focal length (fx, fy): Focal length is the distance from the camera lens center to the image plane or camera sensor. The focal lengths along the X and Y axes can be different, especially if the pixels are not square. For most modern cameras, the pixels are square. Thus, the focal lengths fx and fy are the same in most cameras.
2. Principal point (cx, cy): The principal point is the position where the camera's optical axis intersects the image plane or camera sensor. The optical center is usually near the center of the image, but not always precisely at the center.

In Equation (3-1), Z represents the average distance from the camera to the objects in real-world metric scale. The coordinates (x, y) correspond to the object's position in pixel scale, while (X, Y) denote its position in real-world metric measurements.

$$\begin{gathered} Y = \frac{y - cy}{fy} * Z \\ X = \frac{x - cx}{fx} * Z \end{gathered} \tag{3-1}$$

Assuming y2 and y1 represent the vertical positions of the thickest section of the crack in pixel scale extracted by the proposed method, by inserting them into Equation (3-1), and taking the difference, we arrive at Equation (3-2), which calculates the crack width.

$$Actual\ crack\ width = Y2 - Y1 = \frac{y2 - y1}{fy} * Z = \frac{Pixel\ crack\ width}{fy} * Z \tag{3-2}$$

3.6 Summary

In this chapter, we have delved into the intricacies of the proposed framework. We have provided comprehensive explanation of both the classification and localization modules, shedding light on their corresponding submodules in detail. Furthermore, we have also illuminated how the hybrid proposed method integrates both weakly-supervised and supervised techniques. Additionally, we proposed an innovative method to estimate the linear crack width estimation, which, can be used for calculating PCI. Looking ahead, the following chapter will be dedicated to meticulously evaluating the proposed framework.

Chapter 4: Case Study

4.1 Introduction

This chapter delineates the outcomes obtained from experiments performed using the proposed framework. It also offers a comparative analysis, putting the results of the proposed method up against those from the supervised methods, showcasing both sets of results for an informed evaluation.

4.2 Implementation

In this chapter, the proposed framework is evaluated. Images used in the case study were collected from Forsyth County in Georgia, USA, for suburban and country roads. Images were captured with smartphones mounted on the vehicle hood.

Total number of 34,714 images and 32,738 images were annotated for defect type and severity type classification, respectively. 700 images including four defect types, were also annotated with polygon shapes using RUBIX for defect localization. To ensure robust model training and unbiased performance assessment, the dataset was randomly partitioned, with approximately 80% allocated for training purposes and the remaining 20% designated for testing. This division ensures that the models are thoroughly trained while also maintaining an separate set for validation, thereby enabling an accurate measure of their performance. Notably, all training procedures were carried out in parallel using multiple Graphics Processing Units (GPUs), thereby accelerating the computation speed significantly.

The Resnet with 101 deep layers (Resnet101) is used as a classifier in the two-level classifier module as explained in the Section 3.3. Binary cross entropy and SoftMax cross entropy are used for the first-level and the second-classifier as loss functions, respectively. Stochastic Gradient Descent (SGD) is used as an optimizer. Resnet 101 Feature Pyramid Network (FPN) backbone was used as a feature extractor in the Mask R-CNN .The Pytorch implementation of U-Net model for high definition images was used [111].

4.3 The Proposed Method Experimental Results

4.3.1 Results of Classification

The high accuracy results of the first-level classifier, second-level classifier as well as the total number of images and patches for each defect type are shown Table 4-1, and Table 4-2 respectively.

Figure 4-1and Figure 4-2 also show the confusion matrices of the first-level and the second-level classifiers on the validation set. Although all accuracies are high, considering first and the second-level classifier, linear cracks, such as transverse and longitudinal cracks showed more promising results compared to the area defects, such as block and alligator cracks in the second classifier.

Table 4-1 Results of first-level classifiers

Defect		With Defect	Without Defect	Overall
TRN	Accuracy	0.98	**0.99**	**0.99**
	Number of images	4,168	5,986	10,154
LON	Accuracy	0.97	0.98	0.98
	Number of images	4,115	6,020	10,135
ALG	Accuracy	**0.99**	**0.99**	**0.99**
	Number of images	3,336	3,877	7,213
BLK	Accuracy	**0.99**	**0.99**	**0.99**
	Number of images	3,331	3,881	7,212

(a) TRN

	Without Defect	With Defect
Without Defect	1,185	12
With Defect	13	821

(b) LON

	Without Defect	With Defect
Without Defect	1,182	22
With Defect	24	799

(c) ALG

	Without Defect	With Defect
Without Defect	771	5
With Defect	3	664

(d) BLK

	Without Defect	With Defect
Without Defect	773	3
With Defect	3	664

Figure 4-1 Confusion matrix of first-level classifiers

Table 4-2 Results of second-level classifiers

Defect		High	Medium	Low	Overall
TRN	Accuracy	**0.98**	**0.98**	**0.98**	**0.98**
	Number of patches	2,815	2,845	2,840	8500
LON	Accuracy	**0.98**	**0.98**	0.97	**0.98**
	Number of patches	2,812	2,816	2,800	8,428
ALG	Accuracy	0.97	0.90	0.92	0.93
	Number of patches	2,710	2,775	2,810	8,295
BLK	Accuracy	0.96	0.96	0.94	0.96
	Number of patches	2,805	2,010	2,700	7,515

(a) TRN

	High	Medium	Low
High	556	5	2
Medium	3	558	8
Low	1	6	561

(b) LON

	High	Medium	Low
High	555	2	6
Medium	2	547	9
Low	8	6	539

(c) ALG

	High	Medium	Low
High	529	8	5
Medium	12	507	36
Low	9	26	527

(d) BLK

	High	Medium	Low
High	547	7	7
Medium	5	388	9
Low	10	15	515

Figure 4-2 Confusion matrix of second-level classifiers

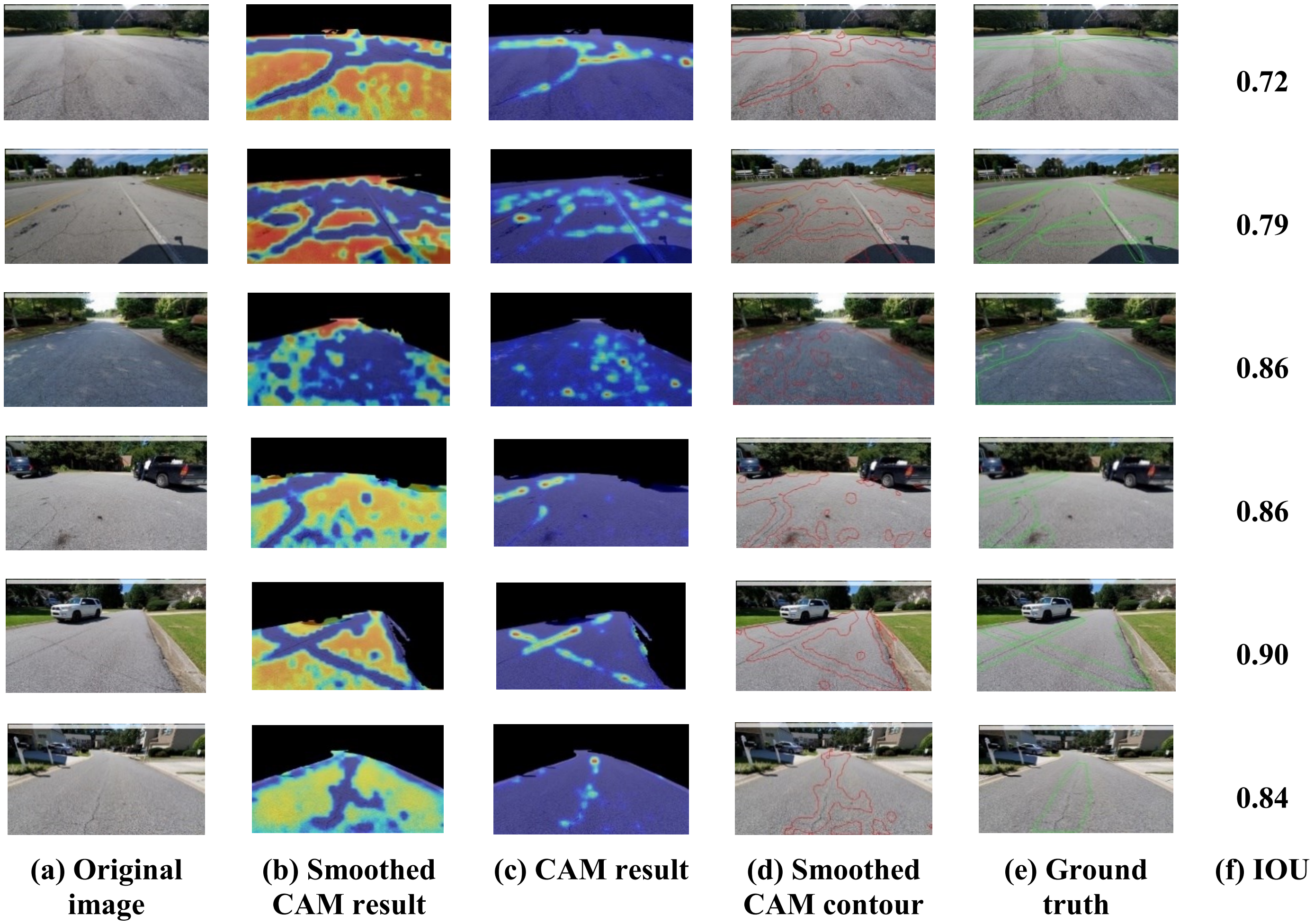
4.3.2 Results of Localization

To assess the segmentation results from the smoothed CAM method, we annotated 100 images including defects and calculated the Intersection Over Union (IOU) between the images segmented by the smoothed CAM method and those from the ground truth. The evaluation yielded an average IOU of 0.81, indicating a strong correlation and demonstrating a commendable performance of the method. Figure 4-3 shows some validation results of the smoothed CAM method and their IOU results.

As mentioned earlier, 700 images were annotated for defect localization; 200 images from which were randomly selected to evaluate the proposed method. The rest will be used to train the supervised methods to compare their results with the proposed method. Some annotated samples are shown in Figure 4-4.

The number of defects for each type in the 700 images is shown in Table 4-3. Since the transverse and the longitudinal cracks are more common compared to the alligator and block cracks, their numbers are greater than the others.

After annotating the images, binary masks were generated from the polygon annotations as ground truth segmentation masks. Samples of annotated images are shown in Figure 4-5.



(a) Original image (b) Smoothed CAM result (c) CAM result (d) Smoothed CAM contour (e) Ground truth (f) IOU

Figure 4-3 Validation of some segmented defect samples from the smoothed CAM method

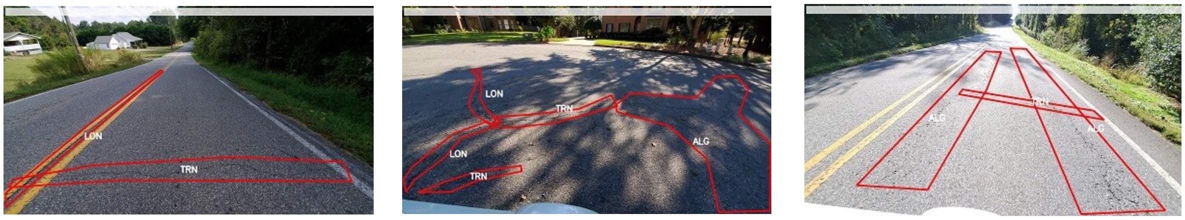


Figure 4-4 Samples of annotated images

Table 4-3 Number of defects in annotated 700 images for each defect type.

Defect Type	Number of defects
TRN	657
LON	585
ALG	240
BLK	87
Total	1569

Next, the masks generated by U-Net and Mask R-CNN (Segmentation module) from the proposed framework were compared with their corresponding ground truth in terms of confidence. Finally, precision, recall and F1-score were calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4-1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4-2}$$

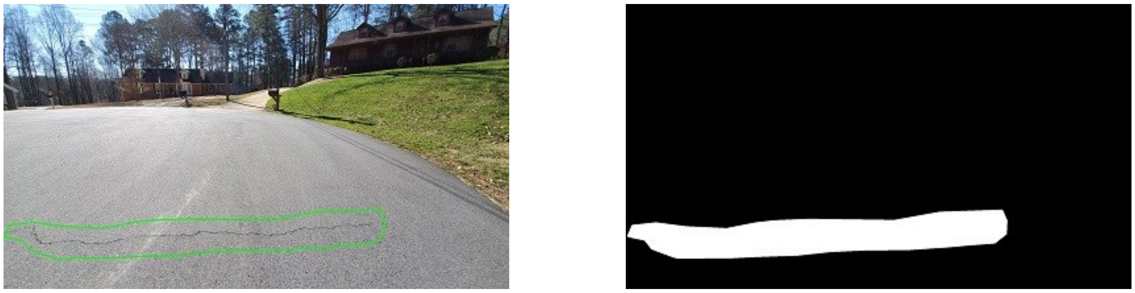
$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} \tag{4-3}$$

TP (True Positive), FP (False Positive) and FN (False Negative) for instance segmentation are defined in Table 4-4.

Table 4-4 TP, FP and FN explanation for the proposed method and instance segmentation

TP	Number of pixels predicted as object correctly
FP	Number of pixels predicted as object wrongly
FN	Number of pixels predicted as background wrongly

Precision refers to the ability of the model to correctly identify true positive cracks from all the predicted positive crack segments. A high precision model will correctly identify most of the true positive crack segments in the image, while also minimizing the number of false positive crack segments. In contrast, recall measures the completeness of true predictions. A model with high recall in pavement crack detection can accurately identify a large proportion of true positive crack segments present in the image, with minimal false negative crack segments. In other words, the higher the recall value the model has, the less missing detected defect it has. This implies that the model can efficiently identify almost all the cracks, regardless of their size and shape, which are present in the image. Thus, to achieve a trade-off between precision and recall, another metric is defined which is F1-score. F1-score is a metric that combines precision and recall. In other words, it can measure the ability of the model to balance between precision and recall. Specifically, it can be regarded as a metric to estimate overall accuracy of the model.



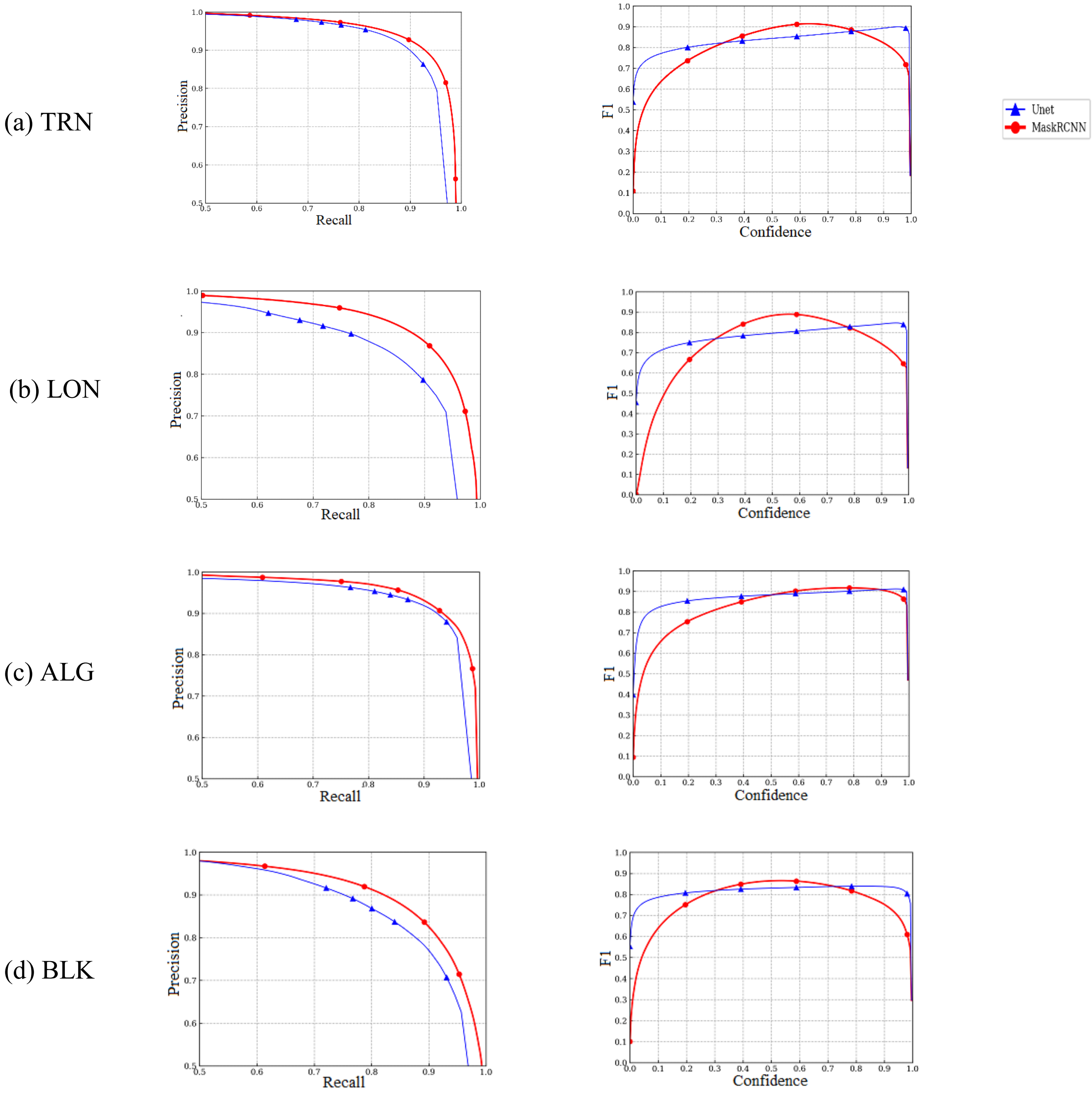
(a) Defect annotation in polygon shape **(b) Corresponding generated ground truth mask.**

Figure 4-5 Sample of defect annotation and its corresponding generated ground truth mask

Table 4-5 The proposed method localization experimental results

Defect	Method	Precision (0.5)	Recall (0.5)	F1 (0.5)	mAP
TRN	Mask R-CNN	0.95	**0.84**	**0.89**	0.81
	U-Net	**0.97**	0.75	0.84	**0.94**
LON	Mask R-CNN	**0.92**	**0.85**	**0.88**	0.73
	U-Net	**0.92**	0.70	0.80	**0.88**
ALG	Mask R-CNN	**0.97**	0.81	**0.88**	0.90
	U-Net	0.95	**0.83**	**0.88**	**0.93**
BLK	Mask R-CNN	**0.88**	**0.85**	**0.86**	0.72
	U-Net	**0.88**	0.79	0.83	**0.83**

The experimental results of the proposed method for each defect type are shown in Table 4-5. As can be seen in this table, in terms of precision values, Mask R-CNN and U-Net have similar performance. However, regarding the recall and F1-score values, Mask R-CNN outperforms U-Net, which means that U-Net has more missing detected defects compared to Mask R-CNN. In contrast, U-Net mean average precision (mAP) values are better than those of Mask R-CNN; that means regarding all confidences, U-Net shows better precision compared to Mask R-CNN. Figure 4-6 shows precision-recall, and F1-score curves of the proposed method localization for each defect type. As can be seen in this figure, in general, Mask R-CNN outperforms U-Net in localizing the defects. Additionally, according to the results, substantially, transverse crack showed superior localization results among the all defect types. The proposed experimental results for some samples are shown in Figure 4-7, Figure 4-8, Figure 4-9 and Figure 4-10 for transverse, longitudinal, alligator and block cracks, respectively.



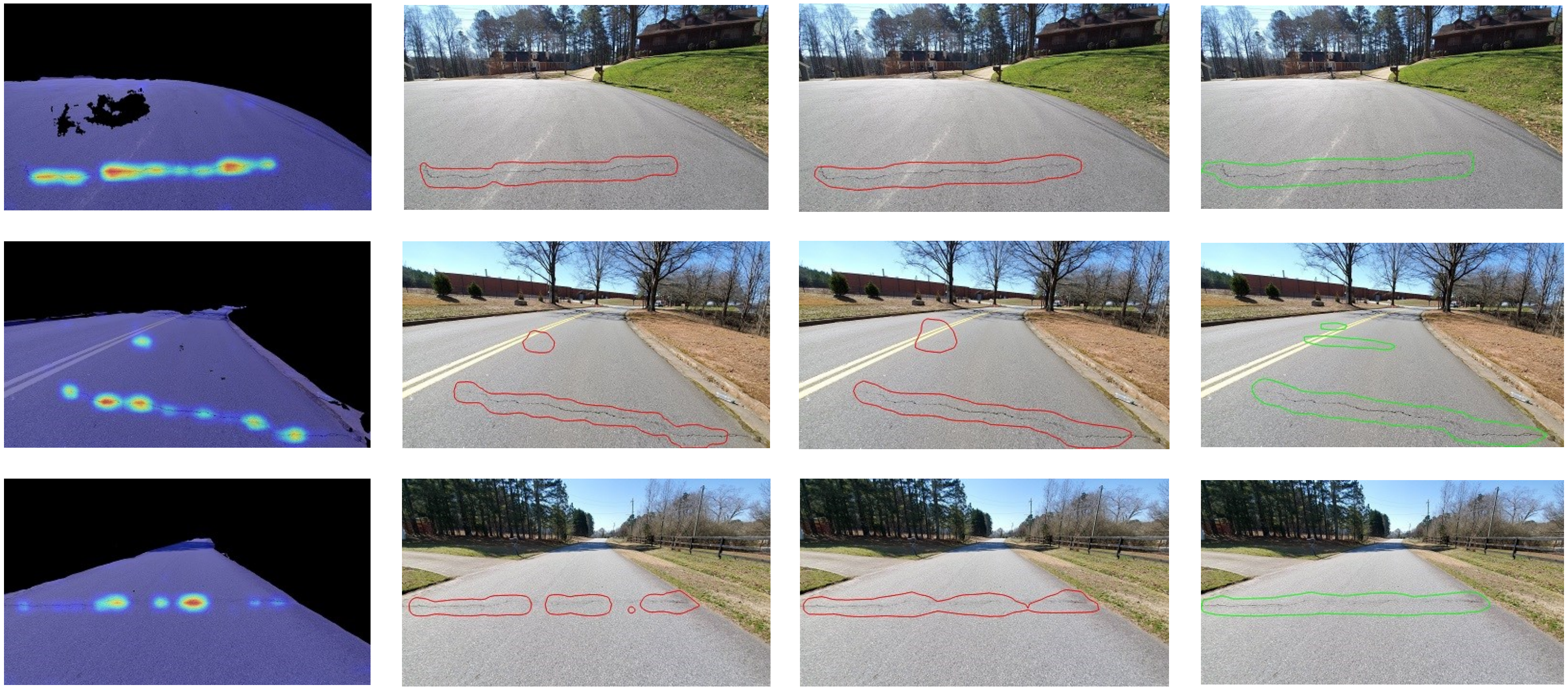
(a) TRN

(b) LON

(c) ALG

(d) BLK

Figure 4-6 Proposed method localization Precision-Recall and F1-score curves for each defect



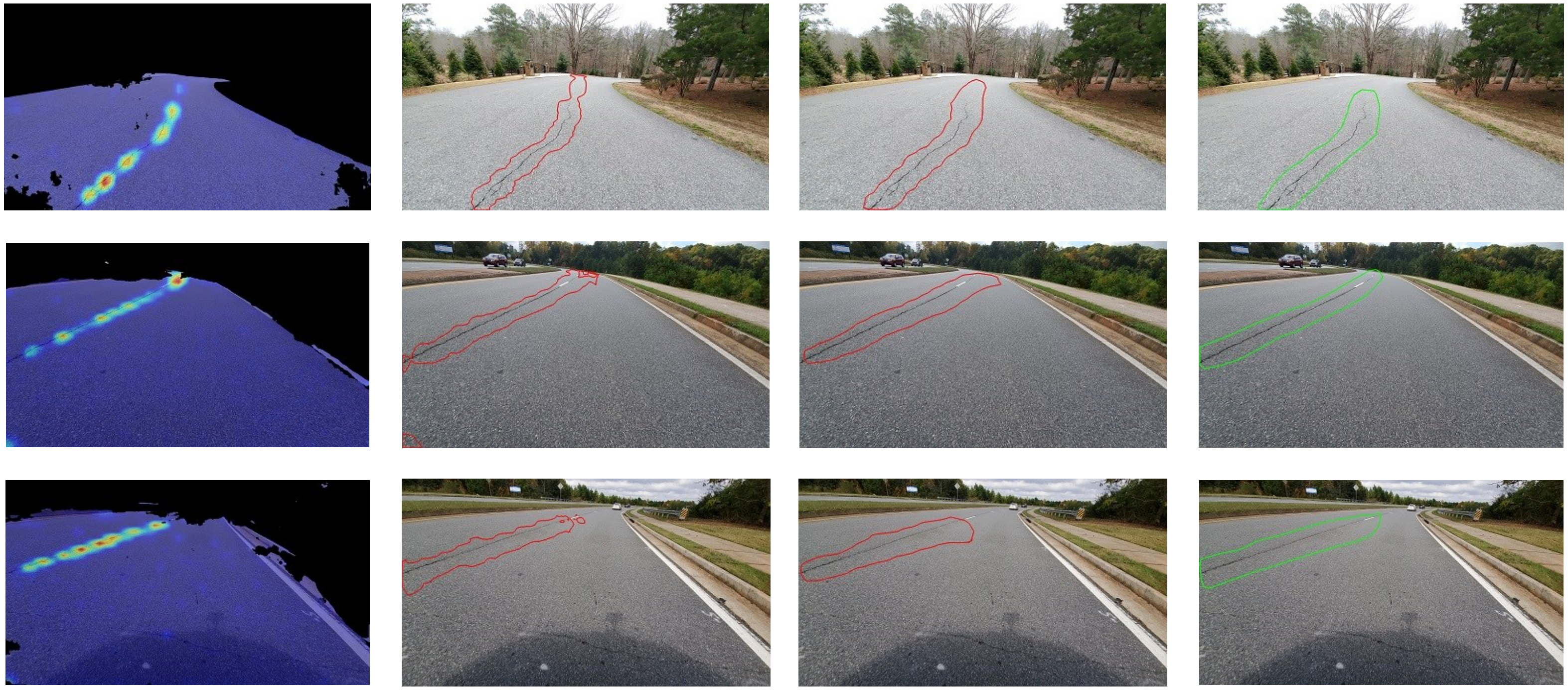
(a) CAM heatmaps

(b) Proposed method (U-Net)

(c) Proposed method (Mask R-CNN)

(d) Ground truth

Figure 4-7 The proposed method results for some transverse defect samples



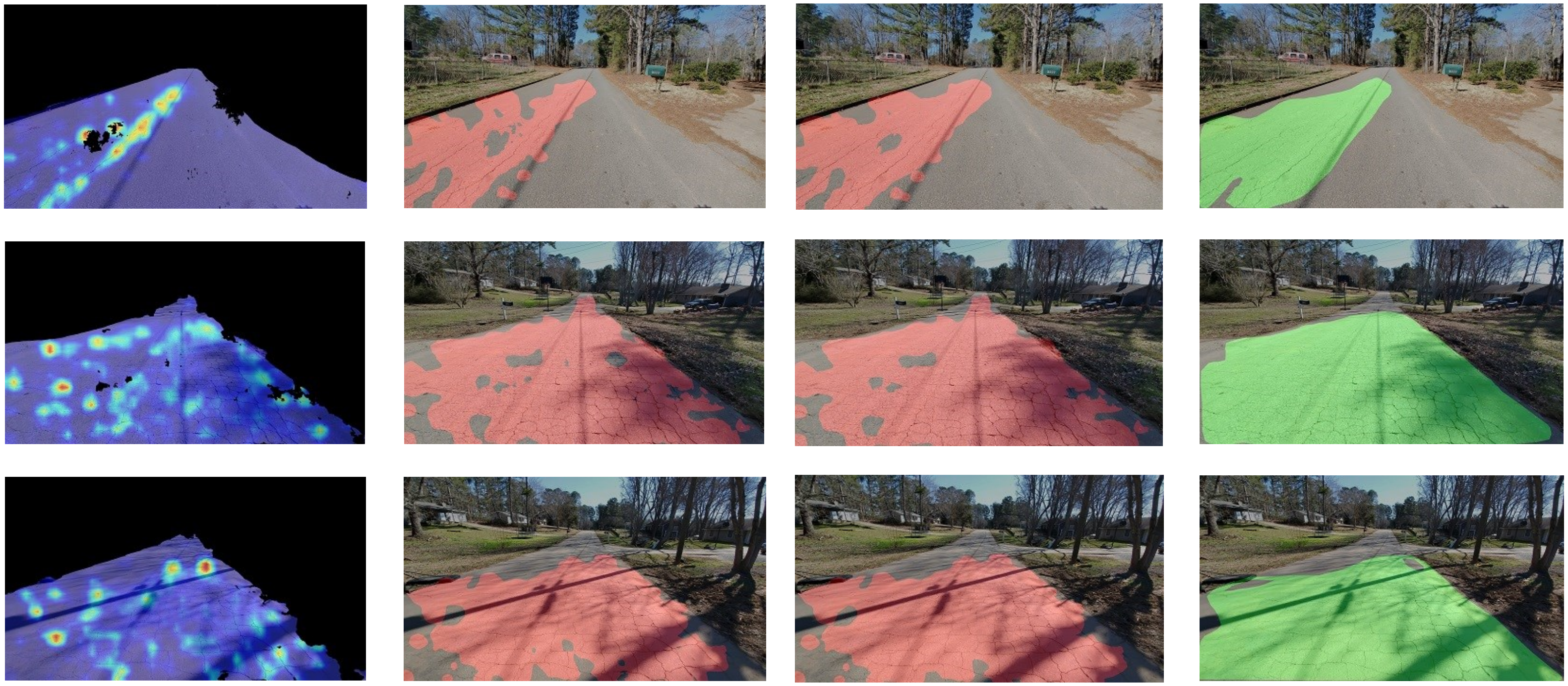
(a) CAM heatmaps

(b) Proposed method (U-Net)

(c) Proposed method (Mask R-CNN)

(d) Ground truth

Figure 4-8 The proposed method results for some longitudinal defect samples



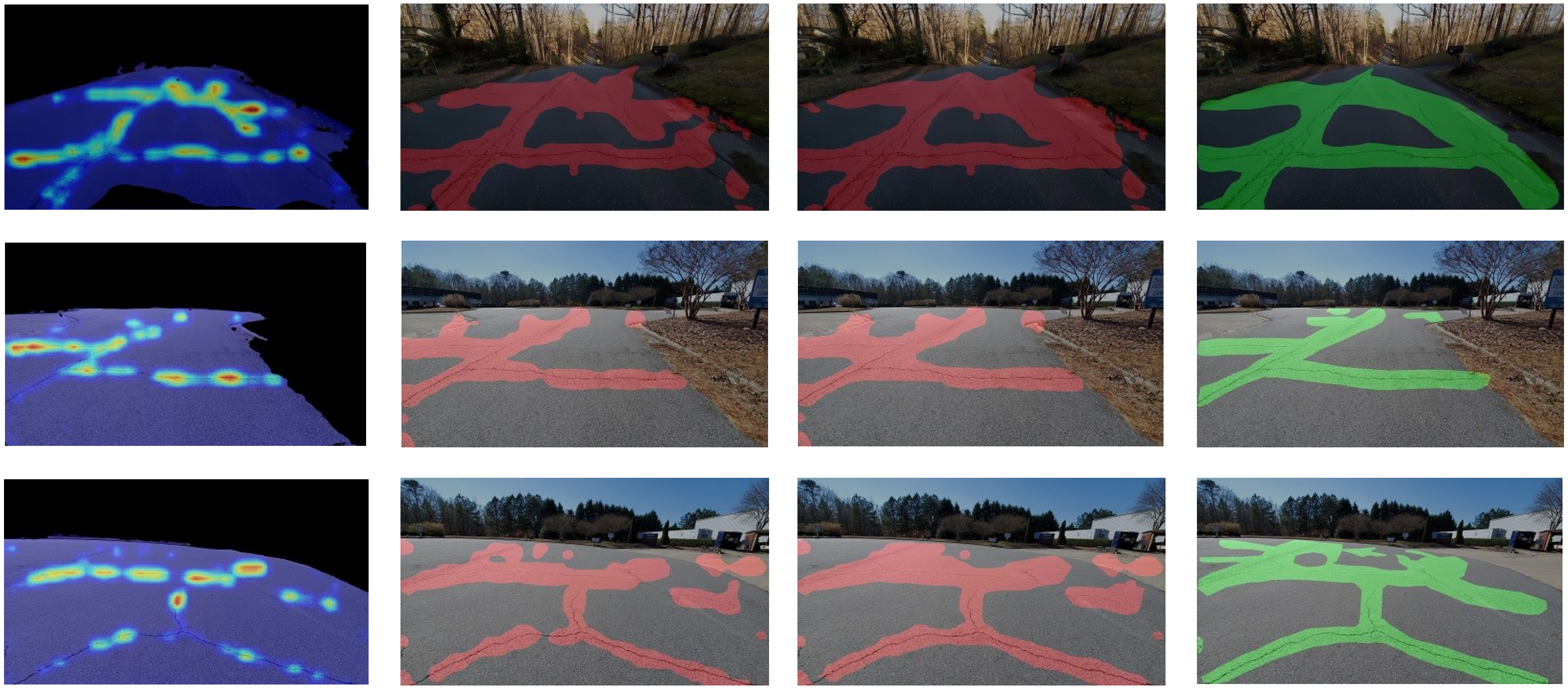
(a) CAM heatmaps

(b) Proposed method (U-Net)

(c) Proposed method (Mask R-CNN)

(d) Ground truth

Figure 4-9 The proposed method results for some alligator defect samples



(a) CAM heatmaps

(b) Proposed method (U-Net)

(c) Proposed method (Mask R-CNN)

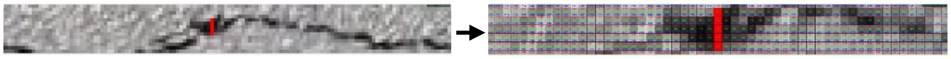
(d) Ground truth

Figure 4-10 The proposed method results for some block defect samples

4.3.3 Crack Width Estimation Calculation

As explained in Section 3.5, an innovative method is proposed to estimate the linear crack width from the segmented patches extracted from the localization module that is essential to calculate PCI. First, the orientation of all segmented patches extracted from Section 3.4 is adjusted to landscape. Following this, the primary contour of crack is generated using the IS algorithm. Subsequently, the contour from the proposed method is centralized, aligning it with the crack's center. Finally, the crack width in each part of the contour is calculated and the largest crack value is selected as crack patch width. Additionally, using the intrinsic parameters obtained from camera calibration, it is possible to transform the crack width from pixel dimensions to real-world metrics through the pinhole camera model using Equation (3-2).

The experimental results of estimating the crack width for an example of crack are shown in Figure 4-11. For this example, assuming fy = 525, z = 300 cm, and the crack width extracted by the proposed method measures 5 pixels, the actual crack width is 2.85 cm.



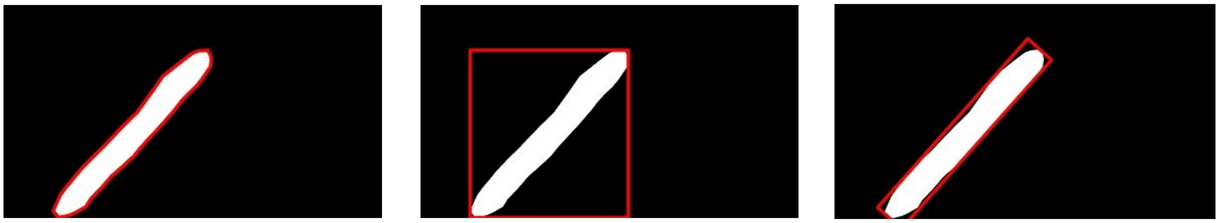
(a) Crack width calculation

(b) Crack width calculation (zoomed area)

Figure 4-11 An example of crack width estimation

4.4 Comparison Between the Hybrid Method and Supervised Methods

In this section, the supervised methods such as instance segmentation, axis-aligned object detection and the oriented bounding box object detection performance in defect localization are compared with the proposed method. The mentioned annotated images in the previous section were used to train the supervised modules, and the same 200 test images were used to evaluate their performance and compare them with the proposed method. Mask R-CNN, YOLO5 (AABB) (Axis-aligned Bounding Boxes), YOLO5 (OBB) (Oriented Bounding Boxes) were use as supervised methods as instance segmentation, axis-aligned object detection and oriented object detection. YOLOv5x X-Large architecture [114] was used to train for both axis-aligned and oriented object detection. Mask CNN architecture with the backbone of Resnet-50 was used as Instance Segmentation. To provide the dataset to train YOLO5 (AABB) and YOLO5 (OBB), annotated polygon shapes should be converted to axis-aligned and oriented bounding boxes. As illustrated in Figure 4-12, generated regular rectangle and minimum area-oriented rectangle that enclosed the polygon are selected as axis-aligned and oriented bounding boxes, respectively, to train YOLO5 (AABB) and YOLO5 (OBB) object detection models.



(a) Polygon **(b) Generated axis-aligned bounding box** **(c) Generated oriented bounding box**

Figure 4-12 Generating axis-aligned and oriented bounding box from the polygon

Figure 4-14 shows some samples of generated axis-aligned and oriented bounding boxes from annotated polygon shapes. To evaluate the instance segmentation model, the same method is used as described in Equations (4-1), (4-2), (4-3) and Table 4-4 in Section 3.6.2 for the proposed method. To evaluate object detection methods, intersection over union (IOU) metric is selected to calculate TP, FP and FN as shown in Figure 4-13 and Table 4-6. Figure 4-15 shows some defect localization results for the three supervised methods. As can be seen in this figure, instance segmentation localized more defects compared to the other two object detection methods. Similarly, Among the object detection methods, axis-aligned object detection method performs better in localizing the defects compared to the oriented object detection. For instance, as Figure 4-15 (c) shows, YOLO5 (OBB) was unable to detect the block defect sample.

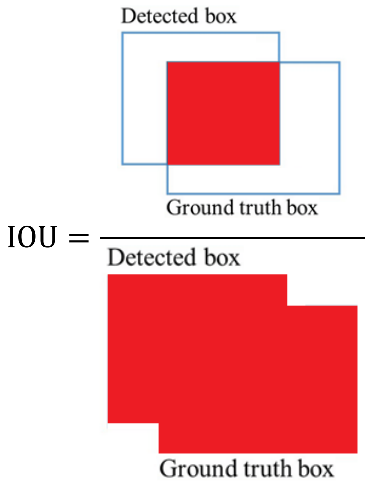
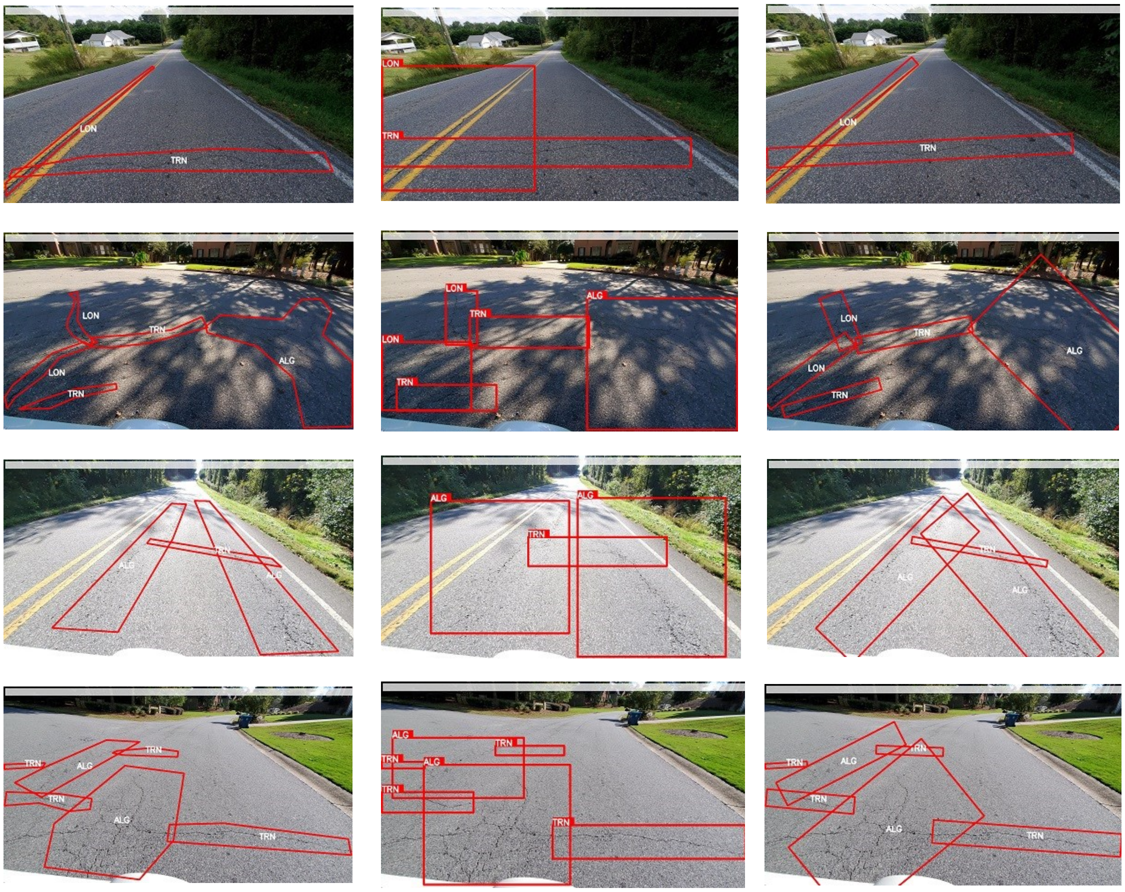


Figure 4-13 IOU calculation



(a) Polygons

(b) Axis-aligned Bounding Boxes

(c) Oriented Bounding Boxes

Figure 4-14 Some annotated sample images in polygon shapes and their converted corresponding shapes to the axis-aligned and oriented bounding boxes shape

Table 4-6 TP, FP and FN explanation for object detection

TP	Correct detection with IOU > threshold
FP	Wrong detection with IOU < threshold
FN	Ground truth not detected

The confusion matrices of two objection detection methods are shown in Table 4-7 and Table 4-8.

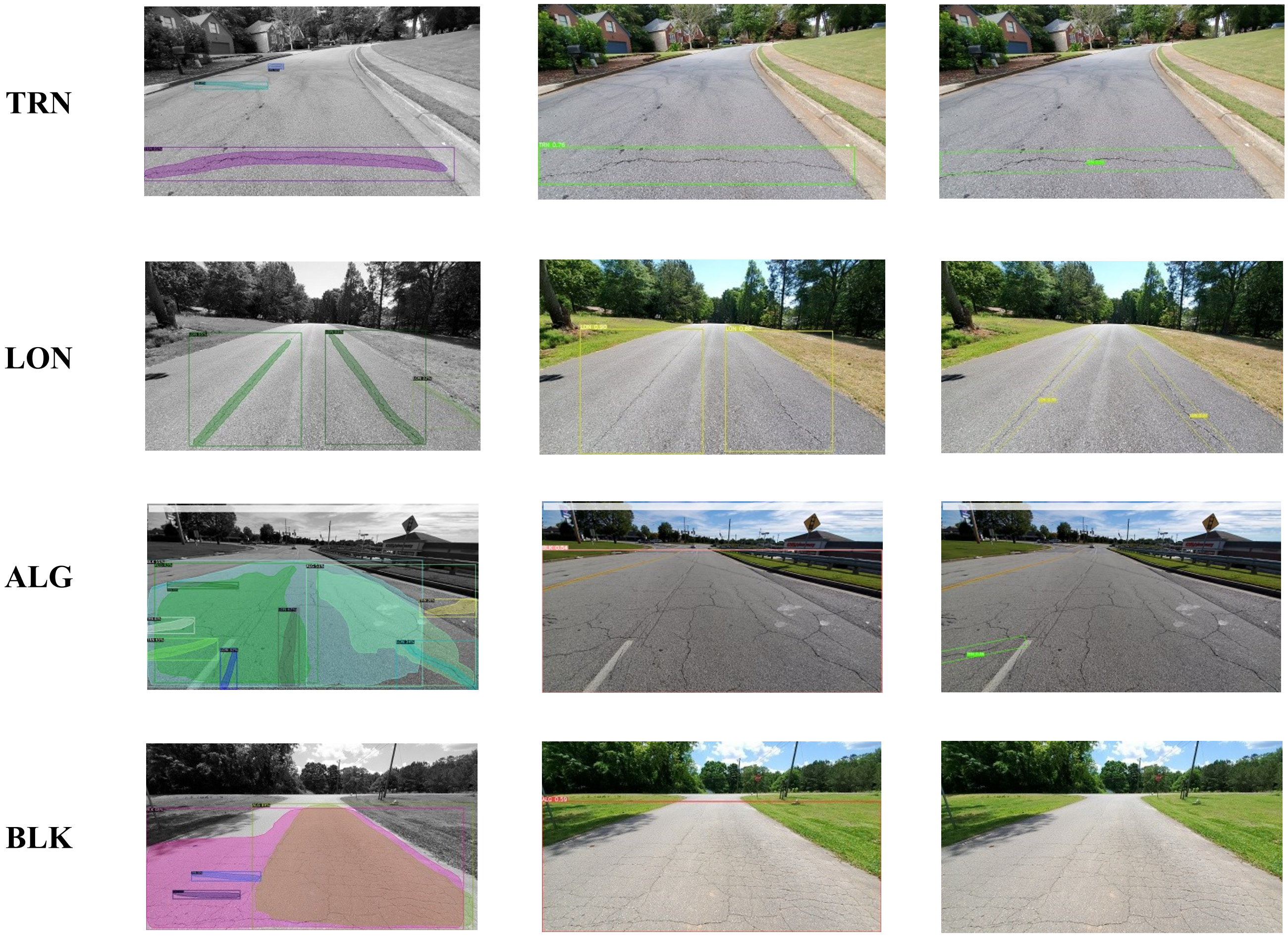
Table 4-7 Axis-aligned object detection confusion matrix

		TRN	**LON**	**ALG**	**BLK**
Prediction	**TRN**	0.64	0	0	0
	LON	0	0.88	0.07	0.14
	ALG	0	0	0.79	0
	BLK	0	0	0.07	0.43
	Background	0.36	0.12	0.07	0.43

Table 4-8 Oriented object detection confusion matrix.

		TRN	**LON**	**ALG**	**BLK**
Prediction	**TRN**	0.26	0	0	0
	LON	0	0.71	0	0
	ALG	0	0	0.07	0
	BLK	0	0	0	0.25
	Background	0.74	0.29	0.93	0.75

As can be seen in Table 4-7 and Table 4-8, and as illustrated in Figure 4-16, although the oriented bounding boxes comprise less part of the background compared to the axis-aligned ones, it did not show as good results as axis-aligned object detection method because of two reasons: (1) oriented object detection model has one more parameter called rectangle angle compared to the axis-aligned object detection method. Hence, it needs more annotated data in various angles to be generalized for defect localization. (2) most of generated oriented bounding boxes exceed the image margins. As aforementioned, oriented bounding boxes are generated from annotated polygon shapes. However, as can be seen in Figure 4-17(b), some of them exceed the image margins and should be modified to be located inside the image margins. Figure 4-17(c) shows some samples of modified bounding boxes. Although, this modification solves the mentioned problem, it causes another issue. As illustrated in Figure 4-17(c), with the new generated oriented bounding boxes, some parts of defects are excluded from the bounding boxes and regarded as background that baffles the deep learning model to distinguish between the defects and the background.



(a) Instance segmentation (Mask R-CNN)

(b) Object detection YOLO5 (AABB)

(c) Object detection YOLO5 (OBB)

Figure 4-15 Supervised methods' visualized defect localization result

Figure 4-18 shows the spatial distributions of defects in the images. As illustrated in this figure, most of the defects are located at the lower part of images next to the image margin due to the camera's perspective field of view and should be modified. Thus, that is another reason that oriented object detection did not show as good results as axis-aligned object detection model.

In contrast, the aforementioned issues for object detection methods are not the case for the instance segmentation. That is because it uses annotated polygon shapes, which possess minimum common area parts with background. Thus, it is expected that instance segmentation outperforms bounding box defect localization methods. Experimental results of all methods and their precision-recall and F1-score curves are shown in Table 4-9 and Figure 4-19. Considering the transverse, longitudinal and alligator cracks results, instance segmentation generates the highest precision, and mean average precision compared to the object detection methods.

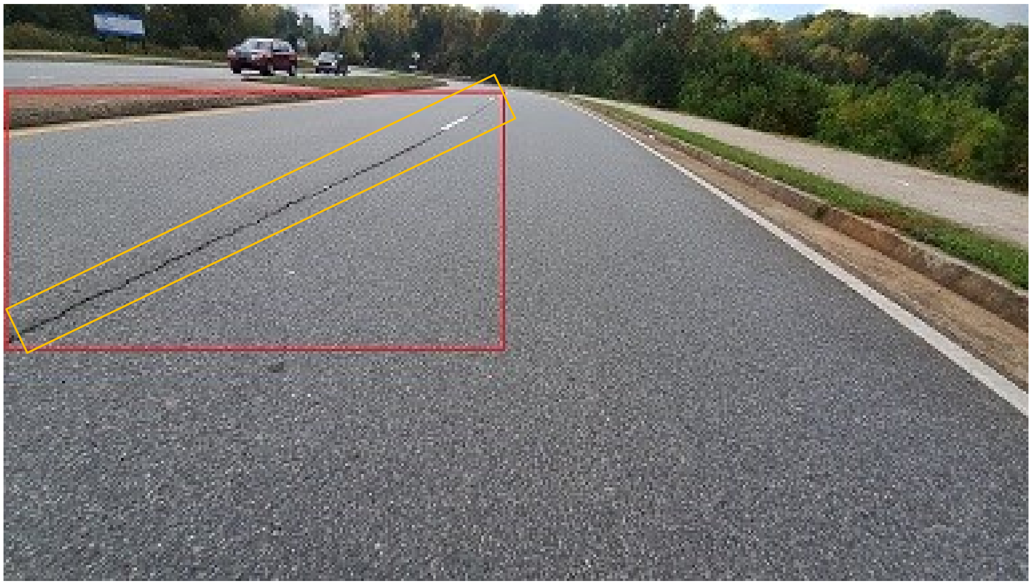
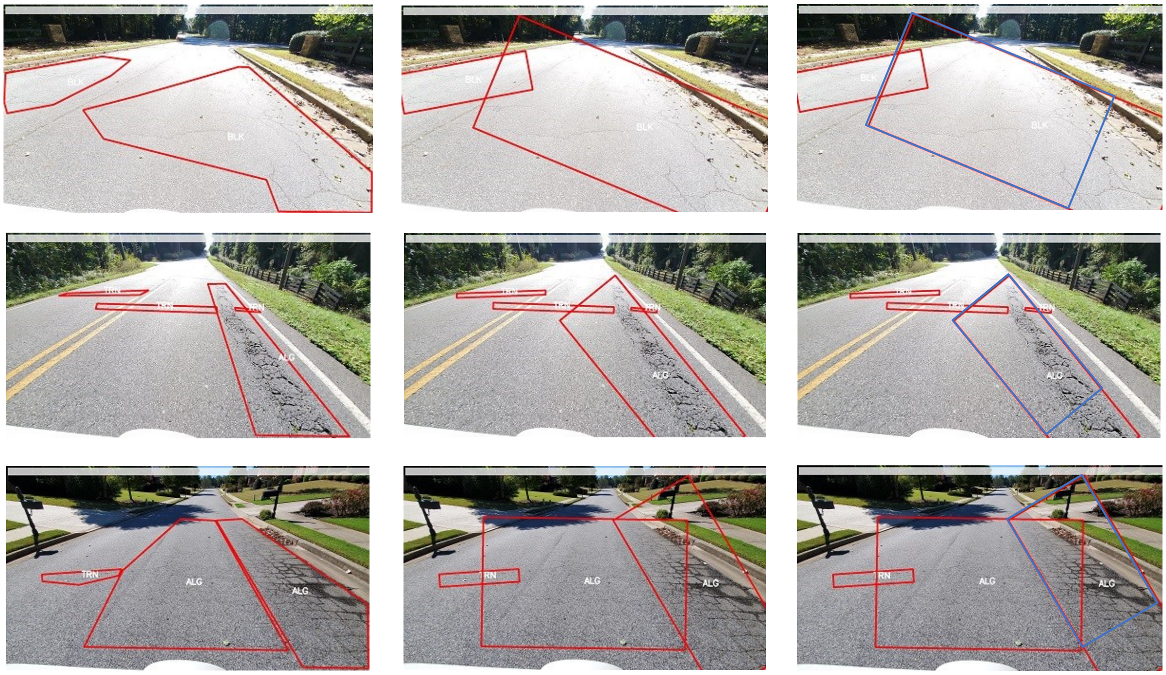


Figure 4-16 Comparing axis-aligned and oriented bounding boxes. Oriented bounding box comprises less part of background than axis-aligned bounding box

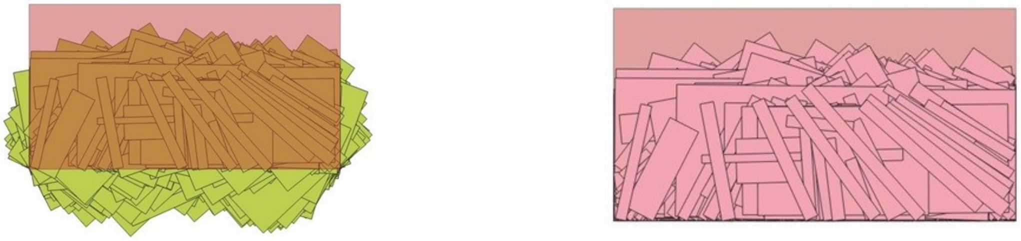


(a) Annotated polygon shapes by RUBIX

(b) Generated oriented bounding boxes from the polygons by RUBIX

(c) Modified oriented bounding boxes by RUBIX

Figure 4-17 Samples of oriented bounding box generation and modification



(a) Generated oriented bounding boxes spatial distribution for all defects

(b) Modified oriented bounding boxes distribution for all defects

Figure 4-18 Oriented bounding box distribution before and after modification for all defects

However, when it comes to the recall values, its values are equal to or less than the axis-aligned objection detection method, especially for longitudinal cracks, which means it has more missing defects compared to the other one. Thus, axis-aligned object detection method outperforms instance segmentation method, especially in linear crack localization, such as longitudinal and transverse cracks. This can be explained because the bounding boxes comprise more areas around the objects than the polygon shapes due to their coherent characteristic. Additionally, axis-aligned object detection offers superior performance to oriented one for longitudinal, transverse and alligator crack.

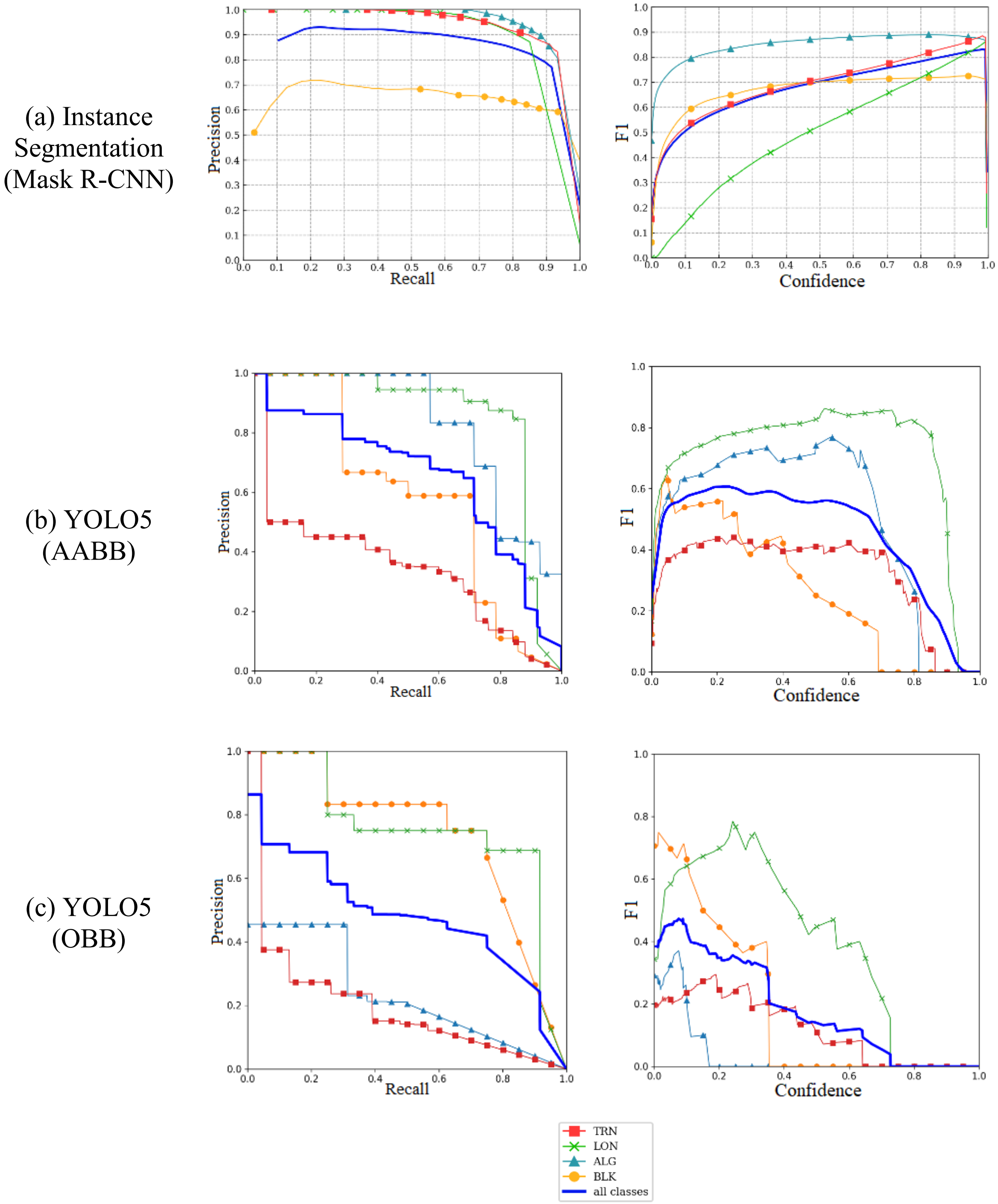
The aforementioned conclusions for the three mentioned defects cannot be generalized to the block crack owing to some reasons. First, as can be seen in Table 4-3, there are a much smaller number of block cracks in the annotated dataset compared to other defect types, which makes it hard for the model to be generalized. Additionally, due to block non-uniform features, they have a wide variety of complex shapes that makes it difficult to distinguish them from the background. For instance, Figure 4-20 shows some samples of each defect type. As illustrated in this figure, linear defects such as longitudinal and transverse cracks are located mostly at roughly straight-line shapes in a vertical and horizontal directions, respectively. Alligator cracks have also roughly the same pattern in all images. However, block cracks, due to their non-uniform structure, comprise more various shapes compared to the other type of defects. Thus, the model needs a greater number of defects in more comprehensive various defect shapes to learn how to distinguish them from the background.

Additionally, as observed in Table 4-3, the relatively small number of defects present in the training images results in a low recall value for the supervised modules. This indicates that these modules may not be adequately generalized to handle unseen data.

Comparing the proposed method alongside other supervised methods, the proposed method outperforms supervised methods, considering all values. Except for longitudinal cracks, the proposed method's precision and mean average precision are approximately equal to or more than the instance segmentation corresponding values. Even for longitudinal cracks, the difference is small between the proposed method precision value (0.92) and instance segmentation precision value (1.00). With respect to recall results, even for longitudinal crack, the recall value is a little bit smaller than axis-aligned object detection, regarding other recall and F1-score values, the proposed method offers superior performance to other methods. Some visualized defect localizations from all methods for transverse, longitudinal, alligator and block cracks are shown in Figure 4-21, Figure 4-22, Figure 4-23 and Figure 4-24, respectively.

Table 4-9 Comparing the proposed method with supervised methods

Defect	Method	Type	Precision	Recall	F1	mAP
TRN	Instance segmentation	Mask R-CNN	**0.99**	0.56	0.71	**0.95**
	Object detection	YOLO5 (AABB)	0.33	0.64	0.44	0.34
	Object detection	YOLO5 (OBB)	0.15	0.39	0.22	0.20
	Proposed Method	Mask R-CNN	0.95	**0.84**	**0.89**	0.81
	Proposed Method	U-Net	**0.97**	0.75	0.84	**0.94**
LON	Instance segmentation	Mask R-CNN	**1.00**	0.36	0.52	**0.98**
	Object detection	YOLO5 (AABB)	0.69	**0.91**	0.78	0.85
	Object detection	YOLO5 (OBB)	0.48	0.90	0.63	0.75
	Proposed Method	Mask R-CNN	**0.92**	**0.85**	**0.88**	0.73
	Proposed Method	U-Net	**0.92**	0.70	0.80	**0.88**
ALG	Instance segmentation	Mask R-CNN	**0.95**	0.81	0.87	0.90
	Object detection	YOLO5 (AABB)	0.61	0.79	0.69	0.82
	Object detection	YOLO5 (OBB)	0.45	0.31	0.37	0.24
	Proposed Method	Mask R-CNN	**0.97**	0.81	**0.88**	0.90
	Proposed Method	U-Net	0.95	**0.83**	**0.88**	**0.93**
BLK	Instance segmentation	Mask R-CNN	0.64	0.78	0.70	0.61
	Object detection	YOLO5 (AABB)	0.64	0.50	0.56	0.58
	Object detection	YOLO5 (OBB)	0.74	0.63	0.68	0.74
	Proposed Method	Mask R-CNN	**0.88**	**0.85**	**0.86**	0.72
	Proposed Method	U-Net	**0.88**	0.79	0.83	**0.83**



(a) Instance Segmentation (Mask R-CNN)

(b) YOLO5 (AABB)

(c) YOLO5 (OBB)

Figure 4-19 Supervised methods localization Precision-Recall and F1-score curves

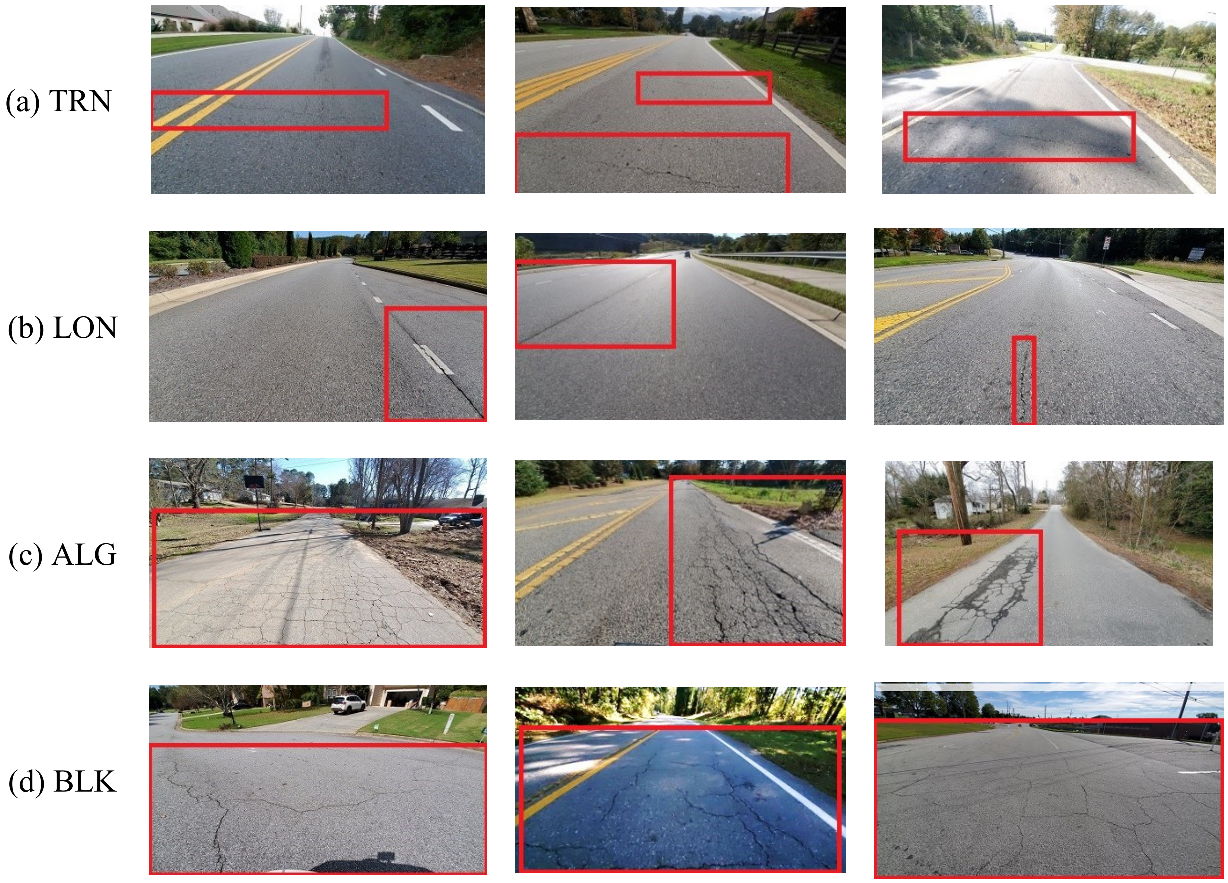
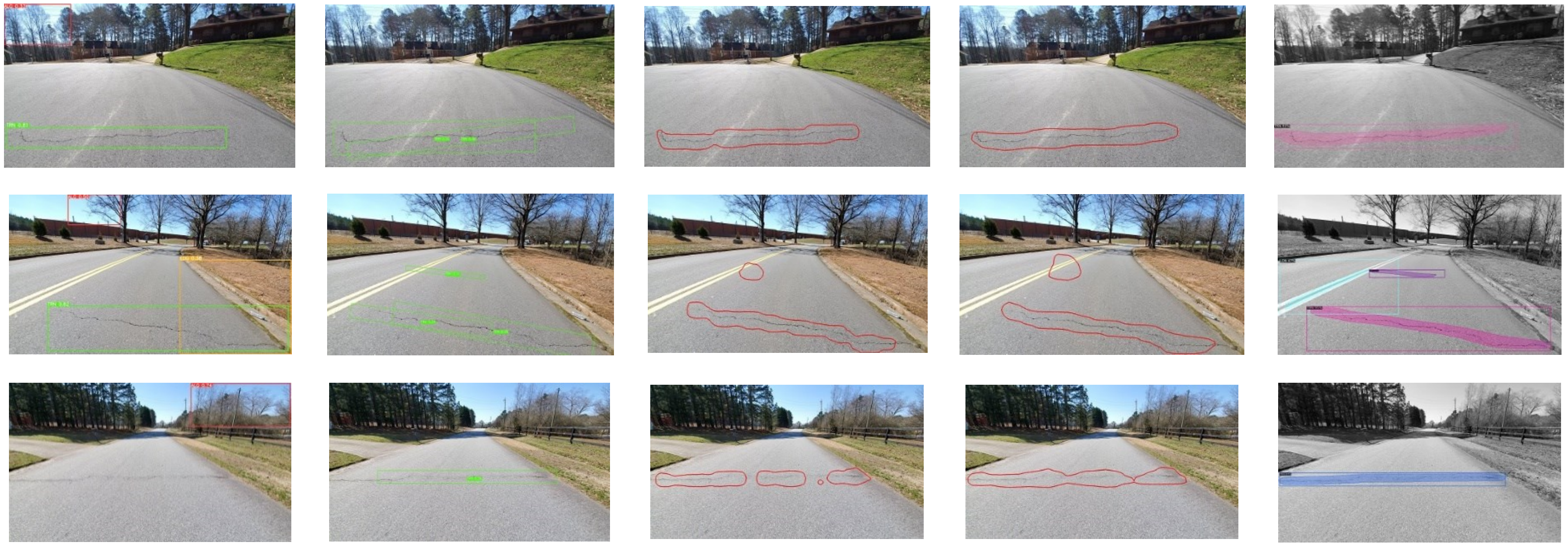


Figure 4-20 Samples of various shapes of defects for each type

In Figure 4-21, Figure 4-22, Figure 4-23 and Figure 4-24, since area cracks cannot be visualized by contours their masks are visualized instead of contours. Additionally, in these figures, there are instances where a defect is not detected, predominantly when using YOLO5 (OBB).



(a) YOLO5(AABB)

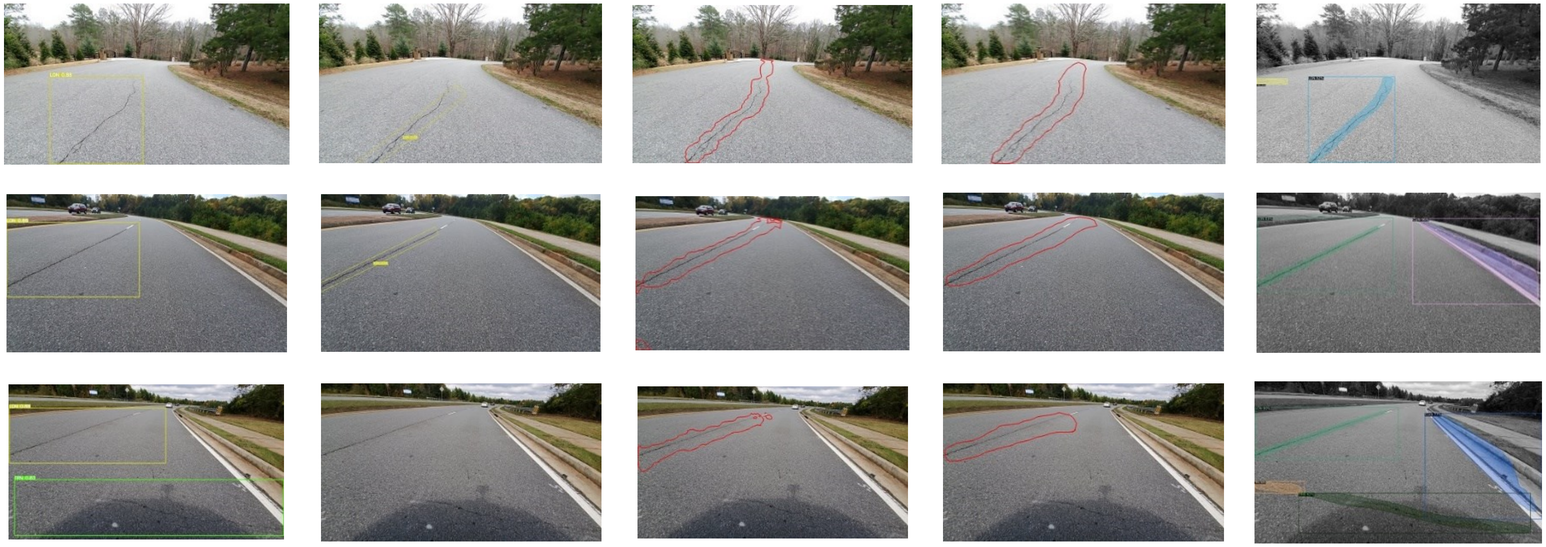
(b) YOLO5(OBB)

(c) Proposed method (U-Net)

(d) Proposed method (Mask R-CNN)

(e) Instance segmentation

Figure 4-21 Transverse crack sample results from all methods



(a) YOLO5(AABB)

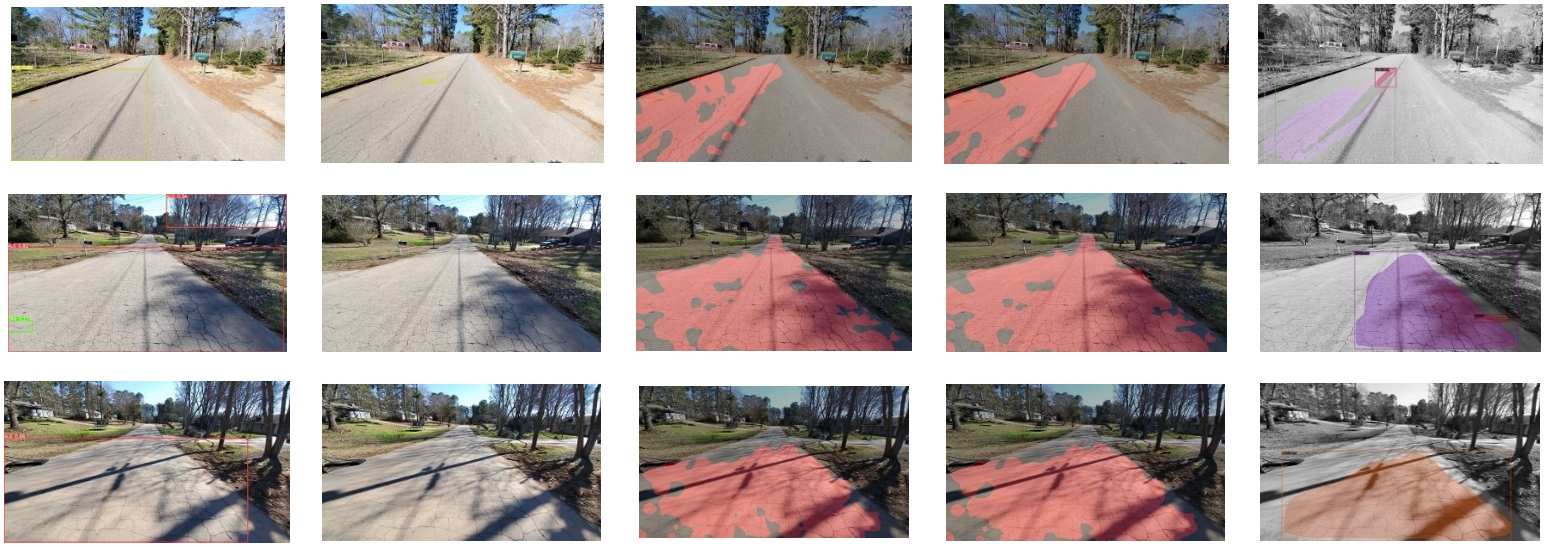
(b) YOLO5(OBB)

(c) Proposed method (U-Net)

(d) Proposed method (Mask R-

(e) Instance segmentation

Figure 4-22 Longitudinal crack sample results from all methods



(a) YOLO5(AABB)

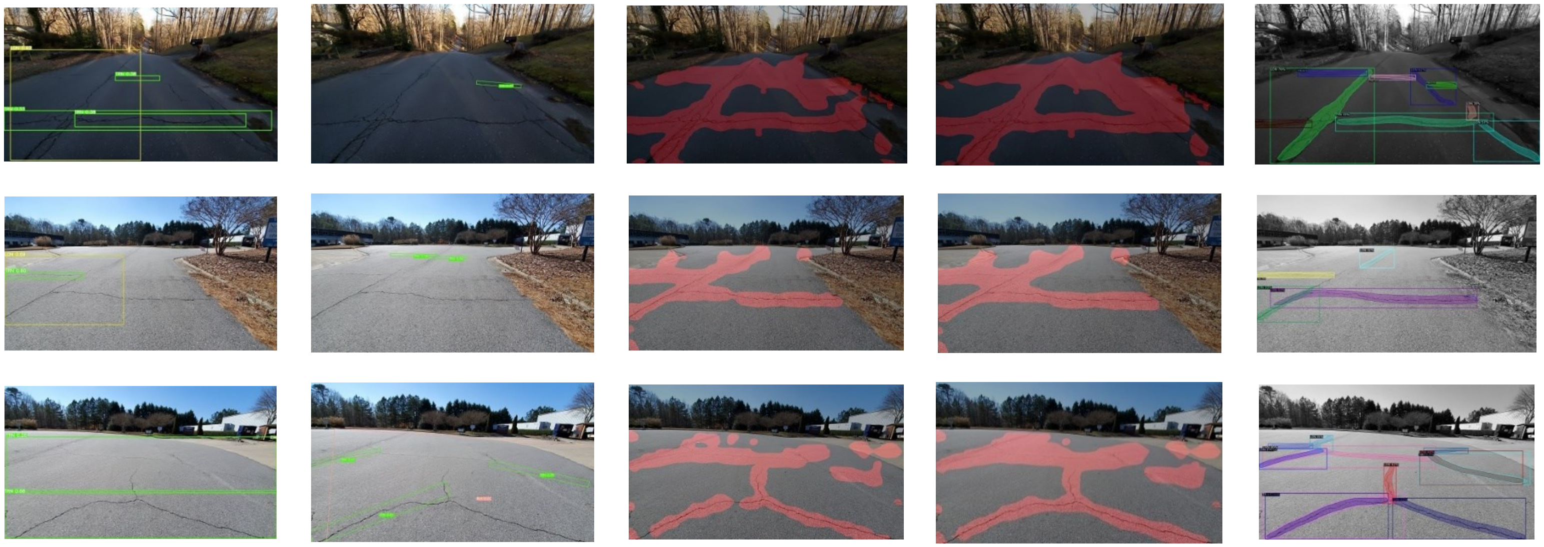
(b) YOLO5(OBB)

(c) Proposed method (U-Net)

(d) Proposed method (Mask R-

(e) Instance Segmentation

Figure 4-23 Alligator crack sample results from all methods



(a) YOLO5(AABB) (b) YOLO5(OBB) (c) Proposed method (U-Net) (d) Proposed method (Mask R-CNN) (e) Instance segmentation

Figure 4-24 Block crack sample results for all method

4.5 Summary and Conclusions

In this chapter, we conducted a comprehensive analysis of the proposed hybrid framework, detailing each individual module. The uniqueness of the method lies in its ability to not only localize and categorize pavement defects, but also to evaluate the extent of their severity utilizing both supervised and weakly supervised techniques. Moreover, the performance of the proposed method was assessed in comparison with supervised methods, such as the widely used instance segmentation and axis-aligned and oriented bounding box object detection techniques. The comparative analysis revealed that the performance of the proposed method surpasses these established techniques in the most cases.

Chapter 5: Summary, Contributions and Future work

5.1 Summary of Research

The primary objective of this research is to create a sophisticated framework for the classification of pavement defect types and severity levels, as well as the localization of defects, using mainly weakly supervised deep learning-based approaches. This framework leverages images obtained from passive Charge-Coupled Device (CCD) cameras and analyzes the defects as the basis for automating the evaluation of pavement conditions. The proposed framework integrates with a Geographic Information System (GIS)-based inspection and asset management system, named RUBIX. This integration allows for comprehensive monitoring and evaluation of the pavement's condition. It enables tracking of pavement conditions and provides insights necessary for effective asset management and maintenance planning.

Chapter 2 undertook a comprehensive examination of existing literature focused on pavement crack detection, localization, and severity estimation techniques. It critically evaluated the constraints of traditional image processing and computer vision methods, namely their struggle with large-scale datasets and inferior adaptability compared to deep learning approaches. Consequently, a deep learning-based method was chosen, considering defect type classification, and localization through integrating weakly supervised and supervised techniques.

Chapter 3 delved into the details of the proposed hybrid method. The proposed framework presents an innovative combination of deep learning-based classification and localization modules to detect, classify defects and estimate their severities, all seamlessly integrated with a GIS-based inspection and asset management system called RUBIX. Leveraging the strengths of both supervised and weakly supervised methods, this hybrid approach significantly improves defect localization. To streamline the data provisioning process for training the segmentation module in the localization section, a semi-automated method has also been proposed. In addition, an innovative approach based on IS method was proposed to estimate the linear crack width from the patches extracted by the localization section.

Chapter 4 conducted a comprehensive analysis of the proposed hybrid framework. Furthermore, it evaluated the effectiveness of the proposed framework against established supervised techniques like instance segmentation, axis-aligned, and oriented bounding box object detection. The outcomes of these comparative assessments affirmed the superior performance of the proposed framework compared to the other methods.

5.2 Research Contributions

The following are the main contributions of this study:

1. A GIS-based framework, integrated with deep learning modules, is designed to automate the evaluation of road pavement conditions by detecting pavement defects.
2. Resnet is applied to classify the types and severity of defects. The severity estimation is based on the patches extracted from the results of the segmentation part of the localization module.
3. Image labels are used, not only to classify the defect types and their severities, but also to localize them primarily using weakly supervised method (i.e. CAM).
4. The discontinuity in defects segmented by the weakly supervised method is addressed by using the supervised segmentation modules. The supervised component of the proposed method requires training only once. There is no need to retrain this supervised part for new

data - a process that is typically time-consuming and complex task in instance segmentation methods. Consequently, the framework can be readily generalized to new data by simply fine-tuning the classifier modules, thereby enhancing its efficiency and versatility. Therefore, the innovative method put forward in this study synthesizes the strengths of both supervised and weakly supervised techniques. In doing so, it forms a hybrid approach that brings together the precision of supervised learning with the flexibility of weakly supervised learning.

5. Given that producing ground truth for training the supervised modules can be a time-consuming and labor-intensive process, a semi-automated solution using smoothed CAM was suggested to simplify this task.
6. An innovative technique was proposed to estimate linear crack widths, leveraging segmented patches obtained from the localization module. This method provides valuable data that can be later employed in the computation of the Pavement Condition Index (PCI).
7. It was demonstrated that the proposed framework outperforms the supervised localization methods (i.e. instance segmentation and object detection), in most of the cases. While the precision values of the proposed method are either equivalent to or slightly lower than those of the instance segmentation method, the recall values for the proposed method (0.85) surpass those of supervised methods in all cases, except for longitudinal cracks, which means the proposed framework has lower number of missing defect defection compared to the other methods. Similarly, the F1-score of the proposed method (0.88) exceeds those of all the considered three supervised methods. The proposed method generates higher recall (0.85) compared to the supervised methods. Although the precisions for linear cracks are slightly smaller than the instance segmentation method, its precisions for area cracks are higher.

5.3 Limitation and Future Work

While this work has made significant strides, there remain some areas of improvement to be considered for future endeavors. Each challenge is followed by a proposed solution and potential avenues for future research.

1. A primary objective for future work is to expand the range of defect types recognized in the classification module. By broadening the spectrum of identified anomalies, it will enable a more comprehensive and nuanced analysis of road pavement conditions.
2. Furthermore, enhancing the precision of the dataset utilized for training the segmentation component represents another key direction for future work. In the proposed semi-automated process previously implemented, we utilized smoothed CAM results which, while effective, still incorporate some background components alongside the defects. To rectify this, future efforts will focus on generating and providing more precise annotations to further refine the accuracy of the segmentation training process. The expectation is that this will significantly improve the localization results, rendering the model more adept at distinguishing defect regions from the background.
3. Another avenue for future work involves leveraging the detailed defect data generated by the framework to compute the PCI. Integrating PCI calculations with our framework could potentially add another layer of usefulness to our analysis. By translating the specific defect information (e.g. width, length, etc.) into a holistic, numerical condition score, the system would provide an actionable snapshot of the pavement’s health.

REFERENCES

[1] N. S. P. Peraka and K. P. Biligiri, “Pavement asset management systems and technologies: A review,” *Autom. Constr.*, vol. 119, 103336, 2020.

[2] B. Becerik-Gerber, S. F. Masri, and M. R. Jahanshahi, “An inexpensive vision-based approach for the autonomous detection, localization, and quantification of pavement defects,” *in Transportation Research Record (TRR)*, 2015.

[3] K. T. Chang, J. R. Chang, and J. K. Liu, “Detection of pavement distresses using 3D laser scanning technology,” in *Computing in civil engineering*, 2005, pp. 1–11.

[4] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, “A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure,” *Adv. Eng. Inform.*, vol. 29, no. 2, pp. 196–210, 2015.

[5] S. Mathavan, K. Kamal, and M. Rahman, “A review of three-dimensional imaging technologies for pavement distress detection and measurements,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2353–2362, 2015.

[6] “Practice for Roads and Parking Lots Pavement Condition Index Surveys”, doi: 10.1520/d6433-20 in *ASTM international*.

[7] J. Chaki and N. Dey, *A Beginner’s Guide to Image Preprocessing Techniques*. Boca Raton: CRC Press, 2018. doi: 10.1201/9780429441134.

[8] A. M. Khan, “Image Segmentation Methods: A Comparative Study,” in *International Journal of Soft Computing and Engineering* vol. 3, no. 4, 2013.

[9] J. Chaki and N. Dey, *A beginner’s guide to image shape feature extraction techniques*. CRC Press, 2019.

[10] H. Zakeri, F. M. Nejad, and A. Fahimifar, “Image based techniques for crack detection, classification and quantification in asphalt pavement: a review,” *Arch. Comput. Methods Eng.*, vol. 24, no. 4, pp. 935–977, 2017.

[11] H.-D. Cheng and M. Miyojim, “Automatic pavement distress detection system,” *Inf. Sci.*, vol. 108, no. 1–4, pp. 219–240, 1998.

[12] H. Zhao, G. Qin, and X. Wang, “Improvement of canny algorithm based on pavement edge detection,” in *2010 3rd International Congress on Image and Signal Processing*, IEEE, 2010, pp. 964–967.

[13] I. Abdel-Qader, O. Abudayyeh, and M. E. Kelly, “Analysis of Edge-Detection Techniques for Crack Identification in Bridges,” *J. Comput. Civ. Eng.*, vol. 17, no. 4, pp. 255–263, Oct. 2003, doi: 10.1061/(ASCE)0887-3801(2003)17:4(255).

[14] H. Lokeshwor, L. K. Das, and S. Goel, “Robust method for automated segmentation of frames with/without distress from road surface video clips,” *J. Transp. Eng.*, vol. 140, no. 1, pp. 31–41, 2014.

[15] S. Chambon and J.-M. Moliard, “Automatic road pavement assessment with image processing: review and comparison,” *Int. J. Geophys.*, vol. 2011, 2011.

[16] J. A. Acosta, J. L. Figueroa, and R. L. Mullen, “Low-Cost Video Image Processing System for Evaluating Pavement Surface Distress,” *Transp. Res. Rec.*, no. 1348, 1992, Accessed: Apr. 05, 2023. [Online]. Available: https://trid.trb.org/view/370750

[17] C. Chen, H. Seo, C. H. Jun, and Y. Zhao, “Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM,” *Int. J. Pavement Eng.*, vol. 23, no. 9, pp. 3274–3283, Jul. 2022, doi: 10.1080/10298436.2021.1888092.

[18] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen, “Automatic Road Crack Detection Using Random Structured Forests,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3434–3445, Dec. 2016, doi: 10.1109/TITS.2016.2552248.

[19] N. Safaei, O. Smadi, A. Masoud, and B. Safaei, “An Automatic Image Processing Algorithm Based on Crack Pixel Density for Pavement Crack Detection and Classification,” International Journal of Pavement Research and Technology, vol. 15, no. 1, pp. 159–172, Jun. 2021, doi: 10.1007/s42947-021-00006-4 .

[20] N. Sholevar, A. Golroo, and S. R. Esfahani, “Machine learning techniques for pavement condition evaluation,” *Autom. Constr.*, vol. 136, 104190, Apr. 2022, doi: 10.1016/j.autcon.2022.104190.

[21] M.-T. Cao, K.-T. Chang, N.-M. Nguyen, V.-D. Tran, X.-L. Tran, and N.-D. Hoang, “Image Processing Based Automatic Detection of Asphalt Pavement Rutting Using a Novel Metaheuristic Optimized Machine Learning Approach,” May 2021, doi: 10.21203/rs.3.rs-334982/v1.

[22] Y.-A. Hsieh and Y. J. Tsai, “Machine Learning for Crack Detection: Review and Model Performance Comparison,” *J. Comput. Civ. Eng.*, vol. 34, no. 5, 04020038, Sep. 2020, doi: 10.1061/(ASCE)CP.1943-5487.0000918.

[23] S. E. Park, S.-H. Eem, and H. Jeon, “Concrete crack detection and quantification using deep learning and structured light,” *Constr. Build. Mater.*, vol. 252, 119096, Aug. 2020, doi: 10.1016/j.conbuildmat.2020.119096.

[24] P. del Río-Barral, M. Soilán, S. M. González-Collazo, and P. Arias, “Pavement Crack Detection and Clustering via Region-Growing Algorithm from 3D MLS Point Clouds,” *Remote Sens.*, vol. 14, no. 22, Art. no. 22, Jan. 2022, doi: 10.3390/rs14225866.

[25] M. Zhong, L. Sui, Z. Wang, and D. Hu, “Pavement Crack Detection from Mobile Laser Scanning Point Clouds Using a Time Grid,” *Sensors*, vol. 20, no. 15, Art. no. 15, Jan. 2020, doi: 10.3390/s20154198.

[26] Z. Du, J. Yuan, F. Xiao, and C. Hettiarachchi, “Application of image technology on pavement distress detection: A review,” *Measurement*, vol. 184, 109900, Nov. 2021, doi: 10.1016/j.measurement.2021.109900.

[27] Kamal, K., Mathavan, S., Zafar, T., Moazzam, I., Ali, A., Ahmad, S.U. and Rahman, M., “Performance assessment of Kinect as a sensor for pothole imaging and metrology,” International Journal of Pavement Engineering, vol. 19, no. 7, pp. 565–576, Jun. 2016, doi: 10.1080/10298436.2016.1187730.

[28] T. Cao, Y. Wang, and S. Liu, “Pavement Crack Detection Based on 3D Edge Representation and Data Communication With Digital Twins,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–10, 2022, doi: 10.1109/TITS.2022.3194013.

[29] Feng, Huifang, Wen Li, Zhipeng Luo, Yiping Chen, Sarah Narges Fatholahi, Ming Cheng, Cheng Wang, José Marcato Junior, and Jonathan Li., “GCN-Based Pavement Crack Detection Using Mobile LiDAR Point Clouds,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 11052–11061, Aug. 2022, doi: 10.1109/tits.2021.3099023.

[30] M. Nasrollahi, N. Bolourian, and A. Hammad, “Concrete surface defect detection using deep neural network based on lidar scanning,” in *Proceedings of the CSCE Annual Conference, Laval, Greater Montreal, QC, Canada*, 2019, pp. 12–15.

[31] H. Feng, L. Ma, Y. Yu, Y. Chen, and J. Li, “SCL-GCN: Stratified Contrastive Learning Graph Convolution Network for pavement crack detection from mobile LiDAR point clouds,” *Int. J. Appl. Earth Obs. Geoinformation*, vol. 118, 103248, Apr. 2023, doi: 10.1016/j.jag.2023.103248.

[32] H. Zhou, B. Gao, and W. Wu, “Automatic Crack Detection and Quantification for Tunnel Lining Surface from 3D Terrestrial LiDAR Data,” *J. Eng. Res.*, Sep. 2022, doi: 10.36909/jer.18459.

[33] R. Ravi, D. Bullock, and A. Habib, “Pavement Distress and Debris Detection using a Mobile Mapping System with 2D Profiler LiDAR,” *Transp. Res. Rec.*, vol. 2675, no. 9, pp. 428–438, Sep. 2021, doi: 10.1177/03611981211002529.

[34] C. Lin, G. Sun, L. Tan, B. Gong, and D. Wu, “Mobile LiDAR Deployment Optimization: Towards Application for Pavement Marking Stained and Worn Detection,” *IEEE Sens. J.*, vol. 22, no. 4, pp. 3270–3280, Feb. 2022, doi: 10.1109/JSEN.2022.3140312.

[35] Z. Li, C. Cheng, M.-P. Kwan, X. Tong, and S. Tian, “Identifying Asphalt Pavement Distress Using UAV LiDAR Point Cloud Data and Random Forest Classification,” ISPRS International Journal of Geo-Information, vol. 8, no. 1, 39, Jan. 2019, doi: 10.3390/ijgi8010039.

[36] T. Wen, H. Lang, S. Ding, J. J. Lu, and Y. Xing, “PCDNet: Seed Operation–Based Deep Learning Model for Pavement Crack Detection on 3D Asphalt Surface,” *J. Transp. Eng. Part B Pavements*, vol. 148, no. 2, 04022023, Jun. 2022, doi: 10.1061/JPEODX.0000367.

[37] Fei, Yue, Kelvin CP Wang, Allen Zhang, Cheng Chen, Joshua Q. Li, Yang Liu, Guangwei Yang, and Baoxian Li. “Pixel-Level Cracking Detection on 3D Asphalt Pavement Images Through Deep-Learning- Based CrackNet-V,” IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 1, pp. 273–284, Jan. 2020, doi: 10.1109/tits.2019.2891167.

[38] X. Chen, J. Li, S. Huang, H. Cui, P. Liu, and Q. Sun, “An Automatic Concrete Crack-Detection Method Fusing Point Clouds and Images Based on Improved Otsu’s Algorithm,” *Sensors*, vol. 21, no. 5, Art. no. 5, Jan. 2021, doi: 10.3390/s21051581.

[39] P. Asadi, H. Mehrabi, A. Asadi, and M. Ahmadi, “Deep Convolutional Neural Networks for Pavement Crack Detection using an Inexpensive Global Shutter RGB-D Sensor and ARM-Based Single-Board Computer,” *Transp. Res. Rec.*, vol. 2675, no. 9, pp. 885–897, Sep. 2021, doi: 10.1177/03611981211004974.

[40] R. Ghosh and O. Smadi, “Automated Detection and Classification of Pavement Distresses using 3D Pavement Surface Images and Deep Learning,” *Transp. Res. Rec.*, vol. 2675, no. 9, pp. 1359–1374, Sep. 2021, doi: 10.1177/03611981211007481.

[41] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, “Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6063–6081, Jul. 2022, doi: 10.1109/TITS.2021.3076844.

[42] N. Kheradmandi and V. Mehranfar, “A critical review and comparative study on image segmentation-based techniques for pavement crack detection,” *Constr. Build. Mater.*, vol. 321, 126162, Feb. 2022, doi: 10.1016/j.conbuildmat.2021.126162.

[43] Y. Zhou, X. Guo, F. Hou, and J. Wu, “Review of Intelligent Road Defects Detection Technology,” Sustainability, vol. 14, no. 10, 6306, May 2022, doi: 10.3390/su14106306.

[44] H. Majidifard, Y. Adu-Gyamfi, and W. G. Buttlar, “Deep machine learning approach to develop a new asphalt pavement condition index,” *Constr. Build. Mater.*, vol. 247, 118513, 2020.

[45] J. Ren, G. Zhao, Y. Ma, D. Zhao, T. Liu, and J. Yan, “Automatic Pavement Crack Detection Fusing Attention Mechanism,” Electronics, vol. 11, no. 21, 3622, Nov. 2022, doi: 10.3390/electronics11213622.

[46] F.-J. Du and S.-J. Jiao, “Improvement of Lightweight Convolutional Neural Network Model Based on YOLO Algorithm and Its Research in Pavement Defect Detection,” Sensors, vol. 22, no. 9, p. 3537, May 2022, doi: 10.3390/s22093537.

[47] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, and H. Kang, “Pavement distress detection and classification based on YOLO network,” International Journal of Pavement Engineering, vol. 22, no. 13, pp. 1659–1672, Jan. 2020, doi: 10.1080/10298436.2020.1714047.

[48] Y. K. Yik, N. E. Alias, Y. Yusof, and S. Isaak, "A Real-time Pothole Detection Based on Deep Learning Approach," *J. Phys. Conf. Ser.*, vol. 1828, no. 1, 012001, Feb. 2021, doi: 10.1088/1742-6596/1828/1/012001.

[49] Y. Zhang, J. Huang, and F. Cai, "On Bridge Surface Crack Detection Based on an Improved YOLO v3 Algorithm," *IFAC-Pap.*, vol. 53, no. 2, pp. 8205–8210, Jan. 2020, doi: 10.1016/j.ifacol.2020.12.1994.

[50] J. Liu, X. Yang, and V. C. S. Lee, "Automated pavement crack detection using region-based convolutional neural network," in *Functional Pavements*, CRC Press, 2020.

[51] S. A. Hassan, S. H. Han, and S. Y. Shin, "Real-time Road Cracks Detection based on Improved Deep Convolutional Neural Network," in *2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, Aug. 2020, pp. 1–4. doi: 10.1109/CCECE47787.2020.9255771.

[52] H. Deng, J. Cheng, T. Liu, B. Cheng, and Z. Sun, "Research on Iron Surface Crack Detection Algorithm Based on Improved YOLOv4 Network," *J. Phys. Conf. Ser.*, vol. 1631, no. 1, 012081, Sep. 2020, doi: 10.1088/1742-6596/1631/1/012081.

[53] D. Jeong, "Road Damage Detection Using YOLO with Smartphone Images," in *2020 IEEE International Conference on Big Data (Big Data)*, Dec. 2020, pp. 5559–5562. doi: 10.1109/BigData50022.2020.9377847.

[54] Z. Deyin, W. Penghui, T. Mingwei, C. Conghan, W. Li, and H. Wenxuan, "Investigation of Aircraft Surface Defects Detection Based on YOLO Neural Network," in *2020 7th International Conference on Information Science and Control Engineering (ICISCE)*, Dec. 2020, pp. 781–785. doi: 10.1109/ICISCE50968.2020.00165.

[55] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788. Accessed: Mar. 19, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[56] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data," Remote Sensing, vol. 13, no. 1, Dec. 2020, doi: 10.3390/rs13010089.

[57] Xu, X., Zhao, M., Shi, P., Ren, R., He, X., Wei, X. and Yang, H., "Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN," Sensors, vol. 22, no. 3, Feb. 2022, doi: 10.3390/s22031215.

[58] R. Li, J. Yu, F. Li, R. Yang, Y. Wang, and Z. Peng, "Automatic bridge crack detection using Unmanned aerial vehicle and Faster R-CNN," *Constr. Build. Mater.*, vol. 362, 129659, Jan. 2023, doi: 10.1016/j.conbuildmat.2022.129659.

[59] K. Hacıefendioğlu and H. B. Başağa, “Concrete Road Crack Detection Using Deep Learning-Based Faster R-CNN Method,” Iranian Journal of Science and Technology, Transactions of Civil Engineering, vol. 46, no. 2, pp. 1621–1633, Jun. 2021, doi: 10.1007/s40996-021-00671-2.

[60] E. Ibragimov, H.-J. Lee, J.-J. Lee, and N. Kim, “Automated pavement distress detection using region based convolutional neural networks,” *Int. J. Pavement Eng.*, vol. 23, no. 6, pp. 1981–1992, May 2022, doi: 10.1080/10298436.2020.1833204.

[61] C. Gou, B. Peng, T. Li, and Z. Gao, “Pavement Crack Detection Based on the Improved Faster-RCNN,” in *2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Nov. 2019, pp. 962–967. doi: 10.1109/ISKE47853.2019.9170456.

[62] H. Alzraiee, A. Leal Ruiz, and R. Sprotte, “Detecting of Pavement Marking Defects Using Faster R-CNN,” *J. Perform. Constr. Facil.*, vol. 35, no. 4, 04021035, Aug. 2021, doi: 10.1061/(ASCE)CF.1943-5509.0001606.

[63] J. Zhai, Z. Sun, J. Huyan, W. Li, and H. Yang, “Feature representation improved Faster R-CNN model for high-efficiency pavement crack detection,” *Can. J. Civ. Eng.*, vol. 50, no. 2, pp. 114–125, Feb. 2023, doi: 10.1139/cjce-2022-0137.

[64] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 91–99, 2015.

[65] L. Attard, C. J. Debono, G. Valentino, M. Di Castro, A. Masi, and L. Scibile, “Automatic Crack Detection using Mask R-CNN,” in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Sep. 2019, pp. 152–157. doi: 10.1109/ISPA.2019.8868619.

[66] T. S. Tran, V. P. Tran, H. J. Lee, J. M. Flores, and V. P. Le, “A two-step sequential automated crack detection and severity classification process for asphalt pavements,” *Int. J. Pavement Eng.*, vol. 23, no. 6, pp. 2019–2033, May 2022, doi: 10.1080/10298436.2020.1836561.

[67] Wang, P., Wang, C., Liu, H., Liang, M., Zheng, W., Wang, H., Zhu, S., Zhong, G. and Liu, “Research on Automatic Pavement Crack Recognition Based on the Mask R-CNN Model,” *Coatings*, vol. 13, no. 2, Art. no. 2, Feb. 2023, doi: 10.3390/coatings13020430.

[68] Dong, J., Liu, J., Fang, H., Zhang, J., Hu, H. and Ma, D., “Intelligent Segmentation and Measurement Model for Asphalt Road Cracks Based on modified Mask R-CNN Algorithm,” *Comput. Model. Eng. Sci.*, vol. 128, no. 2, pp. 541–564, Aug. 2021, doi: 10.32604/cmes.2021.015875.

[69] Y. Zhang, B. Chen, J. Wang, J. Li, and X. Sun, “APLCNet: Automatic Pixel-Level Crack Detection Network Based on Instance Segmentation,” *IEEE Access*, vol. 8, pp. 199159–199170, 2020, doi: 10.1109/ACCESS.2020.3033661.

[70] Liu, Z., Yeoh, J.K., Gu, X., Dong, Q., Chen, Y., Wu, W., Wang, L. and Wang, D., “Automatic pixel-level detection of vertical cracks in asphalt pavement based on GPR investigation and improved mask R-CNN,” *Autom. Constr.*, vol. 146, 104689, Feb. 2023, doi: 10.1016/j.autcon.2022.104689.

[71] J. Zhai, Z. Sun, J. Huyan, H. Yang, and W. Li, “Automatic pavement crack detection using multimodal features fusion deep neural network,” *Int. J. Pavement Eng.*, vol. 0, no. 0, pp. 1–14, Jun. 2022, doi: 10.1080/10298436.2022.2086692.

[72] Xiao, L., Li, W., Deng, N., Yuan, B., Bi, Y., Cui, Y. and Cui, X., “Automatic Pavement Crack Identification Based on an Improved C-Mask Region-Based Convolutional Neural Network Model,” *Transp. Res. Rec.*, vol. 2677, no. 3, pp. 1194–1216, Mar. 2023, doi: 10.1177/03611981221122778.

[73] Y. Yang, Z. Zhao, L. Su, Y. Zhou, and H. Li, “Research on Pavement Crack Detection Algorithm based on Deep Residual Unet Neural Network,” *J. Phys. Conf. Ser.*, vol. 2278, no. 1, 012020, May 2022, doi: 10.1088/1742-6596/2278/1/012020.

[74] L. Zhang, J. Shen, and B. Zhu, “A research on an improved Unet-based concrete crack detection algorithm,” *Struct. Health Monit.*, vol. 20, no. 4, pp. 1864–1879, Jul. 2021, doi: 10.1177/1475921720940068.

[75] J. Huyan, W. Li, S. Tighe, Z. Xu, and J. Zhai, “CrackU-net: A novel deep convolutional neural network for pixelwise pavement crack detection,” *Struct. Control Health Monit.*, vol. 27, no. 8, e2551, 2020, doi: 10.1002/stc.2551.

[76] L. Fan, H. Zhao, Y. Li, S. Li, R. Zhou, and W. Chu, “RAO-UNet: a residual attention and octave UNet for road crack detection via balance loss,” *IET Intell. Transp. Syst.*, vol. 16, no. 3, pp. 332–343, 2022, doi: 10.1049/itr2.12146.

[77] Chen, T., Cai, Z., Zhao, X., Chen, C., Liang, X., Zou, T. and Wang, P., “Pavement crack detection and recognition using the architecture of segNet,” *J. Ind. Inf. Integr.*, vol. 18, 100144, 2020.

[78] G. Li, Q. Liu, W. Ren, W. Qiao, B. Ma, and J. Wan, “Automatic recognition and analysis system of asphalt pavement cracks using interleaved low-rank group convolution hybrid deep network and SegNet fusing dense condition random field,” *Measurement*, vol. 170, p. 108693, Jan. 2021, doi: 10.1016/j.measurement.2020.108693.

[79] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” presented at the Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969. Accessed: Jan. 27, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html

[80] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds.,

in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[81] F. Liu and L. Wang, “UNet-based model for crack detection integrating visual explanations,” *Constr. Build. Mater.*, vol. 322, 126265, Mar. 2022, doi: 10.1016/j.conbuildmat.2021.126265.

[82] L. Wang, X. MA, and Y. Ye, “Computer vision-based Road Crack Detection Using an Improved I-UNet Convolutional Networks,” in *2020 Chinese Control And Decision Conference (CCDC)*, Aug. 2020, pp. 539–543. doi: 10.1109/CCDC49329.2020.9164476.

[83] G. Yu, J. Dong, Y. Wang, and X. Zhou, “RUC-Net: A Residual-Unet-Based Convolutional Neural Network for Pixel-Level Pavement Crack Segmentation,” *Sensors*, vol. 23, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/s23010053.

[84] C. Han, T. Ma, J. Huyan, X. Huang, and Y. Zhang, “CrackW-Net: A Novel Pavement Crack Image Segmentation Convolutional Neural Network,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22135–22144, Nov. 2022, doi: 10.1109/TITS.2021.3095507.

[85] J. Ha, D. Kim, and M. Kim, “Assessing severity of road cracks using deep learning-based segmentation and detection,” The Journal of Supercomputing, vol. 78, no. 16, pp. 17721–17735, May 2022, doi: 10.1007/s11227-022-04560-x.

[86] F. Liu, J. Liu, and L. Wang, “Asphalt Pavement Crack Detection Based on Convolutional Neural Network and Infrared Thermography,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22145–22155, Nov. 2022, doi: 10.1109/TITS.2022.3142393.

[87] A. Issa, H. Samaneh, and M. Ghanim, “Predicting pavement condition index using artificial neural networks approach,” *Ain Shams Eng. J.*, vol. 13, no. 1, 101490, Jan. 2022, doi: 10.1016/j.asej.2021.04.033.

[88] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, “Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection,” *Constr. Build. Mater.*, vol. 157, pp. 322–330, Dec. 2017, doi: 10.1016/j.conbuildmat.2017.09.110.

[89] S. Park, S. Bang, H. Kim, and H. Kim, “Patch-Based Crack Detection in Black Box Images Using Convolutional Neural Networks,” *J. Comput. Civ. Eng.*, vol. 33, no. 3, 04019017, May 2019, doi: 10.1061/(ASCE)CP.1943-5487.0000831.

[90] S. Li and X. Zhao, “Image-Based Concrete Crack Detection Using Convolutional Neural Network and Exhaustive Search Technique,” *Adv. Civ. Eng.*, vol. 2019, e6520620, Apr. 2019, doi: 10.1155/2019/6520620.

[91] K. Chen, A. Yadav, A. Khan, Y. Meng, and K. Zhu, “Improved Crack Detection and Recognition Based on Convolutional Neural Network,” *Model. Simul. Eng.*, vol. 2019, e8796743, Oct. 2019, doi: 10.1155/2019/8796743.

[92] B. Li, K. C. P. Wang, A. Zhang, E. Yang, and G. Wang, “Automatic classification of pavement crack using deep convolutional neural network,” *Int. J. Pavement Eng.*, vol. 21, no. 4, pp. 457–463, Mar. 2020, doi: 10.1080/10298436.2018.1485917.

[93] Schmugge, S.J., Rice, L., Nguyen, N.R., Lindberg, J., Grizzi, R., Joffe, C. and Shin, M.C., “Detection of cracks in nuclear power plant using spatial-temporal grouping of local patches,” in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2016, pp. 1–7. doi: 10.1109/WACV.2016.7477601.

[94] A. Dipankar and S. K. Suman, “Pavement crack detection based on a deep learning approach and visualisation by using GIS,” *Int. J. Pavement Eng.*, vol. 24, no. 1, 2173754, Dec. 2023, doi: 10.1080/10298436.2023.2173754.

[95] J. C. H. Ong, M.-Z. P. Ismadi, and X. Wang, “A hybrid method for pavement crack width measurement,” *Measurement*, vol. 197, 111260, Jun. 2022, doi: 10.1016/j.measurement.2022.111260.

[96] Fan, Z., Li, C., Chen, Y., Di Mascio, P., Chen, X., Zhu, G. and Loprencipe, G., “Ensemble of Deep Convolutional Neural Networks for Automatic Pavement Crack Detection and Measurement,” *Coatings*, vol. 10, no. 2, Art. no. 2, Feb. 2020, doi: 10.3390/coatings10020152.

[97] L. Gao, Y. Yu, Y. Hao Ren, and P. Lu, “Detection of Pavement Maintenance Treatments using Deep-Learning Network,” *Transp. Res. Rec.*, vol. 2675, no. 9, pp. 1434–1443, Sep. 2021, doi: 10.1177/03611981211007846.

[98] H. Zhang, Z. Qian, Y. Tan, Y. Xie, and M. Li, “Investigation of pavement crack detection based on deep learning method using weakly supervised instance segmentation framework,” *Constr. Build. Mater.*, vol. 358, 129117, Dec. 2022, doi: 10.1016/j.conbuildmat.2022.129117.

[99] Y. Inoue and H. Nagayoshi, “Crack Detection as a Weakly-Supervised Problem: Towards Achieving Less Annotation-Intensive Crack Detectors,” in *2020 25th International Conference on Pattern Recognition (ICPR)*, Jan. 2021, pp. 65–72. doi: 10.1109/ICPR48806.2021.9412041.

[100] Y. Tang, Y. Qian, and E. Yang, “Weakly supervised convolutional neural network for pavement crack segmentation,” *Intell. Transp. Infrastruct.*, vol. 1, p. liac013, Sep. 2022, doi: 10.1093/iti/liac013.

[101] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2921–2929. Accessed: Jul. 10, 2022. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/Zhou_Learning_Deep_Features_CVPR_2016_paper.html

[102] Z. Dong, J. Wang, B. Cui, D. Wang, and X. Wang, “Patch-based weakly supervised semantic segmentation network for crack detection,” *Constr. Build. Mater.*, vol. 258, 120291, Oct. 2020, doi: 10.1016/j.conbuildmat.2020.120291.

[103] S. Huang, W. Tang, G. Huang, L. Huangfu, and D. Yang, “Weakly Supervised Patch Label Inference Networks for Efficient Pavement Distress Detection and Recognition in the Wild,” *IEEE Trans. Intell. Transp. Syst.*, pp. 1–13, 2023, doi: 10.1109/TITS.2023.3245192.

[104] Z. Al-Huda, B. Peng, R. N. A. Algburi, S. Alfasly, and T. Li, “Weakly supervised pavement crack semantic segmentation based on multi-scale object localization and incremental annotation refinement,” *Appl. Intell.*, Oct. 2022, doi: 10.1007/s10489-022-04212-w.

[105] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. Accessed: Jan. 26, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[106] A. Tao, K. Sapra, and B. Catanzaro, “Hierarchical Multi-Scale Attention for Semantic Segmentation.” arXiv, May 21, 2020. doi: 10.48550/arXiv.2005.10821.

[107] “NVIDIA/semantic-segmentation.” NVIDIA Corporation, Jan. 24, 2023. Accessed: Jan. 27, 2023. [Online]. Available: https://github.com/NVIDIA/semantic-segmentation

[108] O. Ronneberger, “Invited Talk: U-Net Convolutional Networks for Biomedical Image Segmentation,” Bildverarbeitung für die Medizin 2017, pp. 3–3, 2017, doi: 10.1007/978-3-662-54345-0_3.

[109] J. Gildenblat, “Class Activation Map methods implemented in Pytorch.” Jul. 25, 2022. Accessed: Jul. 25, 2022. [Online]. Available: https://github.com/jacobgil/pytorch-grad-cam

[110] E. N. Mortensen and W. A. Barrett, “Interactive Segmentation with Intelligent Scissors,” *Graph. Models Image Process.*, vol. 60, no. 5, pp. 349–384, Sep. 1998, doi: 10.1006/gmip.1998.0480.

[111] “U-Net: Semantic segmentation with PyTorch.” Mar. 18, 2023. Accessed: Mar. 18, 2023. [Online]. Available: https://github.com/milesial/Pytorch-UNet

[112] X. Yang and J. Yan, “Arbitrary-Oriented Object Detection with Circular Smooth Label,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., Cham: Springer International Publishing, 2020, pp. 677–694. doi: 10.1007/978-3-030-58598-3_40.

[113] “Yolov5 for Oriented Object Detection.” Mar. 17, 2022. Accessed: Mar. 17, 2022. [Online]. Available: https://github.com/hukaixuan19970627/yolov5_obb

[114] G. Jocher, “YOLOv5 by Ultralytics.” May 2020. doi: 10.5281/zenodo.3908559.

[115] J. C. H. Ong, M.-Z. P. Ismadi, and X. Wang, “A hybrid method for pavement crack width measurement,” Measurement, vol. 197, 111260, Jun. 2022, doi: 10.1016/j.measurement.2022.111260.

[116] W. Wang, A. Zhang, K. C. Wang, A. F. Braham, and S. Qiu, “Pavement crack width measurement based on Laplace’s equation for continuity and unambiguity,” Comput.-Aided Civ. Infrastruct. Eng., vol. 33, no. 2, pp. 110–123, 2018.

[117] “Rival Solutions – Road Evaluation Simplified & Mobile.” http://www.rivalsolutions.com/rubix/#contact (accessed Jul. 28, 2023).

[118] R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”, *Cambridge University Press*, 2003.

Appendix A: List of Publications

Amir Jamali, Claude Laflamme, Rob Huber, Amin Hammad. Pavement Defect Classification and Localization Using Weakly Supervised Deep Learning. *Creative Construction Conference (2023 June).*

Amir Jamali, Claude Laflamme, Rob Huber, Amin Hammad. Pavement Defect Classification and Localization Using Hybrid Weakly Supervised and Supervised Deep Learning and GIS. *Journal of Computing in Civil Engineering (to be submitted).*