



**December 2019**

**Report No. 19-010**

**Charles D. Baker**  
Governor

**Karyn E. Polito**  
Lieutenant Governor

**Stephanie Pollack**  
MassDOT Secretary & CEO

# The Application of Unmanned Aerial Systems in Surface Transportation - Volume II-B: Assessment of Roadway Pavement Condition with UAS

Principal Investigator(s)  
**Dr. Walaa Mogawer**  
University of Massachusetts Dartmouth  
**Dr. Yuanchang Xie**  
University of Massachusetts Lowell



Research and Technology Transfer Section  
MassDOT Office of Transportation Planning



U.S. Department of Transportation  
**Federal Highway Administration**

# Technical Report Document Page

1. Report No. 19-010	2. Government Accession No. n/a	3. Recipient's Catalog No. n/a	
4. Title and Subtitle The Application of Unmanned Aerial Systems In Surface Transportation - Volume II-B: Assessment of Roadway Pavement Condition with UAS		5. Report Date December 2019	
		6. Performing Organization Code 19-010	
7. Author(s) Walaa S Mogawer <sup>1</sup> , Yuanchang Xie <sup>2</sup> , Alexander J. Austerman <sup>1</sup> , Chris Dill <sup>1</sup> , Liming Jiang <sup>2</sup> , and Joshua E. Gittings <sup>2</sup>		8. Performing Organization Report No.	
9. Performing Organization Name and Address 1. University of Massachusetts Dartmouth 285 Old Westport Road, Dartmouth, MA 02747 2. University of Massachusetts, Lowell 220 Pawtucket Street, Lowell, MA 01854		10. Work Unit No. (TRAIS) n/a	
		11. Contract or Grant No.	
12. Sponsoring Agency Name and Address Massachusetts Department of Transportation Office of Transportation Planning Ten Park Plaza, Suite 4150, Boston, MA 02116		13. Type of Report and Period Covered Final Report April 2018 – December 2019	
		14. Sponsoring Agency Code n/a	
15. Supplementary Notes Project Champion – Jeffrey DeCarlo, MassDOT Aeronautics Division			
16. Abstract The purpose of this study was to conduct a literature review of existing studies related to pavement condition analysis using Unmanned Aerial Systems (UAS) and conduct a pilot study to evaluate the applicability of using UAS for pavement condition analysis. Each task was undertaken by a different UMass research team. The literature review conducted by the UMass Dartmouth team suggested that the use of UAS for pavement condition assessment is still in its infancy with little experience and information available in the literature. Currently, no UAS platform can provide pavement condition analysis as currently obtained by traditional automated and manual methods. Photogrammetry for crack detection appears to be the most promising condition assessment technique for potential integration with UAS. The Mass Lowell team reviewed research related to new algorithms for analyzing pavement condition data with a specific focus on deep learning. The results suggested that many deep neural networks models have been developed to detect cracks from images and achieved considerable success, but with limitations. A pilot study was conducted with UAS collected images at the Fitchburg Municipal Airport and analyzed using CrackIT and two deep learning methods. The U-Net model performed very well given a limited training dataset.			
17. Key Words Pavement distress, image processing, deep learning, neural networks, crack detection, classification		18. Distribution Statement unrestricted	
19. Security Classif. (of this report) unclassified	20. Security Classif. (of this page) unclassified	21. No. of Pages 55	22. Price n/a

This page left blank intentionally.

# **The Application of Unmanned Aerial Systems In Surface Transportation - Volume II- C: Evaluation of UAS Highway Speed-Sensing Application**

Prepared By:

Principal Investigator  
**Walaa S. Mogawer**

Senior Research Engineer  
**Alexander J. Austerman**

Research Assistant  
**Chris Dill**  
University of Massachusetts Dartmouth

Co-Principal Investigator  
**Yuanchang Xie**

Research Assistants  
**Liming Jiang and Joshua E. Gittings**  
University of Massachusetts Lowell

Prepared For:  
Massachusetts Department of Transportation  
Office of Transportation Planning  
Ten Park Plaza, Suite 4150  
Boston, MA 02116

December 2019

This page left blank intentionally.

# **Acknowledgments**

This study was undertaken as part of the Massachusetts Department of Transportation Research Program with funding from the Federal Highway Administration State Planning and Research funds. The authors are solely responsible for the accuracy of the facts and data, the validity of the study, and the views presented herein.

The Project Team would like to acknowledge the support of Dr. Jeffrey DeCarlo, Owen Silbaugh, Dr. Scott Uebelhart, Ed Gomez, Thomas Mahoney, Steven Rawding, Andrew Mihaley, Terrence McKenna, Christopher Grazioso, and Samuel Nelson from MassDOT and Scott Ellis from Fitchburg Municipal Airport.

# **Disclaimer**

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

This page left blank intentionally.

# Executive Summary

This study of Assessment of Roadway Pavement Condition with UAS was undertaken as part of the Massachusetts Department of Transportation (MassDOT) Research Program. This program is funded with Federal Highway Administration (FHWA) State Planning and Research (SPR) funds. Through this program, applied research is conducted on topics of importance to the Commonwealth of Massachusetts transportation agencies.

The main objectives of this research are twofold: (1) performing a review of existing Unmanned Aerial System (UAS) technologies for pavement condition survey, and (2) conducting a pilot study to evaluate the applicability of using UAS for pavement condition analysis. The literature review has been focused on both sensors that can be integrated with UAS and advanced algorithms for analyzing pavement distress data. Since most pavement distress data is in the form of images, the review of algorithms are concentrated on pavement image processing methods developed recently. Other than the literature review, several field trips were made to the Fitchburg Municipal Airport (FMA) in Massachusetts to assess the feasibility of using UAS mounted sensors to collect image data for evaluating runway pavement conditions. The collected FMA data was used in conjunction with some online pavement images to evaluate the crack detection capabilities of a well-known Matlab tool and two deep learning methods.

Chapter 1 briefly presents the objectives and background of this research. The literature review results of UAS technologies and data analysis algorithms are detailed in Chapters 2 and 3, respectively. Based on the literature review, the use of UAS for pavement condition assessment is still in its infancy with little experience and information available. For integration with UAS, photogrammetry appears to be the most popular technique, while the use of multi- or hyperspectral imaging is getting increasing attention. The integration of such sensors with UAS for pavement condition assessment has not been thoroughly explored yet, although it does show potential. The literature review also shows that there has been growing interest in applying deep learning methods for detecting pavement distresses from images. Many deep neural networks models have been developed for this purpose and have achieved considerable success. However, most of the methods are to identify and highlight cracked regions with a rectangular box, instead of identifying cracks at the pixel level. There is still a lack of consistent and well-accepted procedures to convert crack detection results into Pavement Condition Index (PCI) values. Additionally, to develop deep neural networks based crack detection algorithms, it is important to have a large database of pavement images with cracks clearly labeled. Compiling such a large database is time consuming, and transfer learning appears to be a possible solution to address this issue.

Increasing a drone's flight altitude will decrease the pavement image quality but reduce costs. A qualitative comparison of the drone images captured at different altitudes at the FMA suggest that flying the drone 50 feet above the runway seems to provide the best balance between efficiency and accuracy. For the camera used in this study, the 50 feet altitude resulted in a 0.16 inches per pixel ground sample distance. Both visual and thermal

cameras were used for data collection at the FMA, and the results suggest that thermal images do not seem to add value compared to images from the regular camera.

The collected pavement images were used to evaluate the crack detection performance of a MATLAB toolbox called CrackIT and two deep learning methods. Overall, the two deep learning methods outperformed the CrackIT toolbox. Among the two deep learning methods, the U-Net has the advantage of generating crack detection results at the pixel level compared to the SSD Mobilenet\_V1 network. Applying deep learning methods requires a large annotated pavement image dataset. To address this issue, an online pavement image database was used in addition to the collected FMA data. The modeling results suggest that U-Net performs well on the FMA testing dataset when trained using the online pavement images in addition to a small sample of FMA images. The trained U-Net model was also applied to some highway pavement images collected using a van-mounted laser scanning system and generated satisfactory crack detection results. Overall, the introduced U-Net is a promising method for image-based pavement crack detection for both airport runways and highways, and demonstrated great robustness and generalization ability.

Going forward, additional research is needed to understand how to determine PCI values from the pixel-level crack detection results, and compare the PCI values with the assessment results of qualified engineers. Also, a programmable procedure needs to be established to generate reliable and consistent PCI outcomes from UAS images. Finally, this research demonstrated the feasibility of collecting pavement images using cameras mounted on rotary wing drones and extracting useful crack information using deep learning methods. In the future, fixed wing drones can be considered to further expand the coverage and speed of UAS based pavement condition data collection.

# Table of Contents

Technical Report Document Page .....	i
Acknowledgments.....	v
Disclaimer .....	v
Executive Summary .....	vii
Table of Contents .....	ix
List of Tables .....	xi
List of Figures .....	xi
List of Acronyms .....	xiii
1.0 Introduction.....	1
1.1 Problem Statement.....	1
1.2 Objectives .....	2
1.3 Report Outline.....	2
2.0 Types of UAS Technology for Pavement Condition Analysis.....	3
2.1 Photogrammetry.....	3
2.2 Light Detection and Ranging (LiDAR) .....	3
2.3 Case Studies of UAS Technology for Pavement Condition Analysis .....	4
2.3.1 Airsight—Germany.....	4
2.3.2 South Dakota.....	4
2.3.3 Massachusetts Institute of Technology .....	4
2.3.4 U.S. Air Force Institute of Technology .....	5
2.3.5 Sidewalk Condition Assessment.....	6
2.4 UAS Platform and Instrumentation Requirements .....	7
2.5 Data Collection and Reduction .....	7
2.6 Cost Savings .....	7
2.7 Benefits of UAS for Pavement Condition Analysis over Traditional Methods .....	7
2.8 Closing Summary of UAS for Pavement Condition Analysis.....	7
3.0 Review of Deep Learning Applications in Pavement Distress Analysis.....	9
3.1 Summary of Gopalakrishnan’s Review Paper .....	9
3.2 MatConvNet.....	11
3.3 CNN for Crack Length Estimation .....	12
3.4 Closing Summary of Deep-Learning Applications in Pavement Distress Analysis.....	12
4.0 Data and Analysis .....	15
4.1 Data Sources .....	15
4.1.1 Fitchburg Municipal Airport (FMA) Data .....	15
4.1.2 Other Data Sources .....	19
4.2 CrackIT for Pavement Crack Detection.....	20
4.3 Deep Neural Networks for Pavement Crack Detection .....	27
4.3.1 Crack Detection .....	27
4.3.2 Crack Segmentation .....	30
4.4 Closing Summary of Data and Analysis.....	36

5.0 Conclusions and Future Work .....	37
6.0 References.....	39

# List of Tables

Table 4.1: Flight plan for the second drone data collection.....	16
Table 4.2: Data used in U-Net modeling .....	31
Table 4.3: Model performance comparison .....	33

# List of Figures

Figure 2.1: Maximum speed of image capture characterized by megapixels of camera.....	6
Figure 4.1: Drone image from the initial data set .....	15
Figure 4.2: Pavement image taken by a cell phone .....	16
Figure 4.3: Pavement image taken by the elevator flight .....	17
Figure 4.4: Ground sample distance and field of view width vs. flight altitude.....	18
Figure 4.5: Pavement image taken by the elevator flight .....	19
Figure 4.6: Sample original and annotated pavement images .....	20
Figure 4.7: Flow diagram of CrackIT detection method .....	20
Figure 4.8: First attempt to apply CrackIT to FMA sample image .....	21
Figure 4.9: First attempt to fine tune CrackIT parameters.....	21
Figure 4.10: L: Sample image from the CrackIT database; R: Sample image from the FMA dataset .....	22
Figure 4.11: Example chunk from FMA image.....	22
Figure 4.12: Results of chunking image into 192 nonoverlapping chunks of 300x300 pixels and applying CrackIT on each chunk .....	22
Figure 4.13: Comparison of image histogram before (L) and after (R) applying multilevel thresholding.....	23
Figure 4.14: Results of adding multilevel thresholding preprocessing step to CrackIT.....	23
Figure 4.15: Results of CrackIT with best smoothing parameters.....	24
Figure 4.16: Error/false detection region .....	24
Figure 4.17: Example of false detection area inspected during field trip .....	25
Figure 4.18: a) Cell phone image of false detection area; b) CrackIT applied on cell phone image using best parameters from processing FMA dataset; c) results of applying CrackIT “out of the box” on cell phone image .....	26
Figure 4.19: Screenshot of utilizing Labelling to draw ground truth bounding boxes.....	28
Figure 4.20: From left to right: Learning rate, mAP, loss for the first attempt .....	28
Figure 4.21: From left to right: Learning rate, mAP, loss for second attempt .....	29
Figure 4.22: From left to right: Learning rate, mAP, loss for third attempt .....	29
Figure 4.23: Two example results of transfer learning: Right subimage of each pair is the ground truth, while left is bounding box predicted by neural network.....	29
Figure 4.24: Example of neuronal image and segmentation.....	30
Figure 4.25: Illustration of how IoU is defined .....	32
Figure 4.26: Prediction results of U-NetCrack500 on Crack500 data .....	34
Figure 4.27: Prediction results of U-NetCrack500 and U-NetCrack500&FMA on FMA data .....	34
Figure 4.28: Prediction results of U-NetCrack500 on highway pavement image data .....	35

Figure 4.29: Application of U-Net to initial FMA data ..... 35

# List of Acronyms

<b>Acronym</b>	<b>Expansion</b>
AASHTO	American Association of State Highway and Transportation Officials
AFCS	Automatic Flight Control System
API	Application Programming Interface
ASTM	American Society for Testing and Materials
CNN	Convolutional Neural Networks
COCO	Common Objects in Context
DIGITS	Deep Learning GPU Training System
DOT	Departments of Transportation
FHWA	Federal Highway Administration
FMA	Fitchburg Municipal Airport
GAPS	German Asphalt Pavement Distress
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
MassDOT	Massachusetts Department of Transportation
MCS	Mission Control System
MIT	Massachusetts Institute of Technology
PCI	Pavement Condition Index
SGD	Stochastic Gradient Descent
SPR	State Planning and Research
SSD	Single Shot MultiBox Detector
SVM	Support Vector Machine
UAD	UMass Amherst Dataset
UAS	Unmanned Aerial Systems

This page left blank intentionally.

# 1.0 Introduction

This study of Task B: Assessment of Roadway Pavement Condition with UAS was undertaken as part of the Massachusetts Department of Transportation (MassDOT) Research Program. This program is funded with Federal Highway Administration (FHWA) State Planning and Research (SPR) funds. Through this program, applied research is conducted on topics of importance to the Commonwealth of Massachusetts transportation agencies.

## 1.1 Problem Statement

---

Paved roads deteriorate over time due to traffic loading and environment. To capture the extent and severity of the deterioration, state agencies and municipalities conduct pavement condition surveys of their roadway network. The data collected during these surveys is converted into indices that are used to guide decision-makers on how to best distribute available funds to keep the roads in service and safe.

The data collected during these condition assessments consist of the distresses present on the roadway (cracking, raveling, weathering, etc.) and roadway profiles. In general, condition surveys are conducted annually or biennially. However, this survey process of evaluating the condition of pavements can be expensive, labor intensive, and time-consuming. Many traditional road evaluation methods utilize measurements taken in situ along with visual examinations and interpretations.

The use of unmanned aerial systems (UAS) offers new potential for pavement managers to conduct condition assessments of roads and airport pavement. UAS have the potential to evaluate large areas in significantly less time compared to the existing manual methods. Findings from the 2016 Phase I research performed for MassDOT (“The State of the Practice of UAS Applications in Transportation” and “Current Counter-Drone Technology Solutions to Shield Airports and Approach and Departure Corridors”) indicate that UAS technology is actively being investigated by a growing number of state Departments of Transportation (DOTs) to creatively improve safety, reduce traffic congestion, and save on costs. According to a March 2016 survey by the American Association of State Highway and Transportation Officials (AASHTO), 33 state DOTs have or are exploring, researching, or the use of UAS to inspect bridges and assist with clearing vehicle crashes, among other innovative applications.

## **1.2 Objectives**

---

The main objective of this research task was to understand the current state and potential future state of practice regarding the use of UAS technology for pavement condition assessment. It attempted to address the following major questions:

- How are other agencies using currently available UAS technology for pavement condition analysis?
- What are the current UAS platform and instrumentation requirements that are needed to perform pavement condition analysis?
- What are the future UAS platform and instrumentation requirements for fully-automated pavement condition analysis?
- What data is collected by UAS and how is the data converted into a usable format?
- How is the UAS data reduced, and how does this data compare to the data collected by the current state-of-the-art conventional and semi-automated methods?
- What are the costs and cost savings involved with the use of UAS technology for pavement condition analysis applications?
- What improvements in safety, time, quality of data, and cost can be expected when comparing traditional (non-UAS) methods, to semi-automated and fully-automated UAS pavement assessment methods?
- If successful, what is the broader impact of using UAS technology for pavement condition analysis and subsequent pavement maintenance, repair, and re-construction decisions?

## **1.3 Report Outline**

---

This research consists of a literature synthesis and a pilot study of using both traditional and deep learning methods for pavement distress analysis. The literature synthesis has been divided into two chapters for clarity. Chapter 2 addresses material related to UAS technology for pavement condition analysis, and Chapter 3 summarizes methods to detect distresses in pavement images. Chapter 4 presents the results of the pilot study and Chapter 5 concludes this research.

## 2.0 Types of UAS Technology for Pavement Condition Analysis

The two main types of data collection system used by UAS for pavement condition assessment are photogrammetry and light detection and ranging (LiDAR). LiDAR is an abbreviation for light detection and ranging.

### 2.1 Photogrammetry

---

Photogrammetry is the oldest and most common type of data collection system used by UAS for pavement condition assessment. This method is a less expensive option than LiDAR or spectral imaging. UAS photogrammetry consists of a series of “bird’s eye view” two-dimensional photos taken from a relatively low altitude and combined into one large 2D photo (1). A three-dimensional point cloud can also be generated by using a special method known as stereo-photogrammetry (1). Photogrammetry is different from aerial photography in that it is used to generate a composite map-like image which allows for distances between objects to be measured (2). New software is constantly being developed with improved algorithms to more accurately and consistently assess the type, severity, and quantity of pavement distresses. Ahmed and Haas (2009) describes the data analysis algorithms by saying they “rely internally on one or more threshold values during processing or may need a pre-processing stage” (1). What this essentially means is that despite the improvements in the algorithms of this software, the process still requires a lot of user input and refining to be reliable (1).

### 2.2 Light Detection and Ranging (LiDAR)

---

The Hartsfield-Jackson Atlanta International Airport has been a large supporter in the use of UAS for runway inspections (2). A presentation put together by the airport, Michael Baker International, and Pond & Company gives a brief introduction into the concepts of data fusion along multiple data collection systems including LiDAR technology (2). The Atlanta airport defines it as a “surveying method that measures the distance to a target by illuminating it with a laser,” offering a secondary method to measure various distances within a project to ensure agreement with photogrammetric data (2). Similar software is found on vehicle-mounted sensors that assess pavement condition; these are used by various public agencies and private businesses to determine road profiles from close range. A benefit that pavement condition assessment could gain from UAS LiDAR technology is that its “data can filter to the ground through the vegetation as long as light can be seen from under the tree canopy” (2). This quality of LiDAR data presents the opportunity to perform data fusion and cross-comparison to compensate for reduced accuracy in photogrammetry due to sub-optimal lighting (2). One downside of the LiDAR onboard equipment is that it “still weighs a lot, thus

making it a challenge to keep the drone under 55 pounds” (2). Smaller and lighter LiDAR equipment are being designed to address this issue.

## **2.3 Case Studies of UAS Technology for Pavement Condition Analysis**

---

### **2.3.1 AirSight—Germany**

The Airbus Operations Division of the Hamburg-Finkenwerder Aerodrome hired a company called AirSight to conduct an UAS inspection of a 130,000-square-foot section of concrete on the aerodrome’s apron (3). Their drone took aerial photographs and was operated on a GPS-based programmed flight path with a minimum distance of 150 meters to the runway and 10 meters to parking aircraft to ensure safety (3). The company’s software then took four days combining all of the photos into what they call a “single georeferenced orthophoto” with a resolution of 2 millimeters per pixel (3). Instead of using an autonomous interpretation of this dataset, several qualified engineers analyzed the single georeferenced orthophoto in AutoCAD and graded it based on a virtual-interpretation of ASTM (American Society for Testing and Materials) D5340 (3). The results were accurate, and the advantages include limited interruption to aerodrome operations, reduced human error or bias, and elimination of software-generated errors (3). This study proves the reliability of qualified human assessment for orthophotographs (3). However, this process can be time-consuming and more expensive than other options. No details were provided about the drone or equipment utilized.

### **2.3.2 South Dakota**

The U.S. Department of Transportation sponsored a 2008 study done by South Dakota State University, whose findings demonstrated the ability of an UAS with a photogrammetric mapping center to assess the condition of an unpaved road (4). The airframe used was an Airstar International Mongoose airframe helicopter powered by a 26cc, single cylinder, Zenoah G260H engine.. The flight controller used was the combination of the Automatic Flight Control System hardware (AFCS) and custom Mission Control System software (MCS). The image sensor used was a UEye 2220c USB video camera with frame size 768x576 with an on board OptiLogic RS-232 laser range finder. No details on the drone flight operations were provided. A GPS and geomagnetic sensor kept track of the drone’s position, velocity and altitude; which allowed for a pre-programmed flight path (4). A computer at ground-level was working in real-time with the UAS computer which “includes camera calibration, integrated sensor orientation, digital 3D road surface model and orthoimage generation, automated feature extraction, and measurement for road condition assessment” (4). The main outcome of this reference was a proposed new strategy for assessing unpaved road conditions.

### **2.3.3 Massachusetts Institute of Technology**

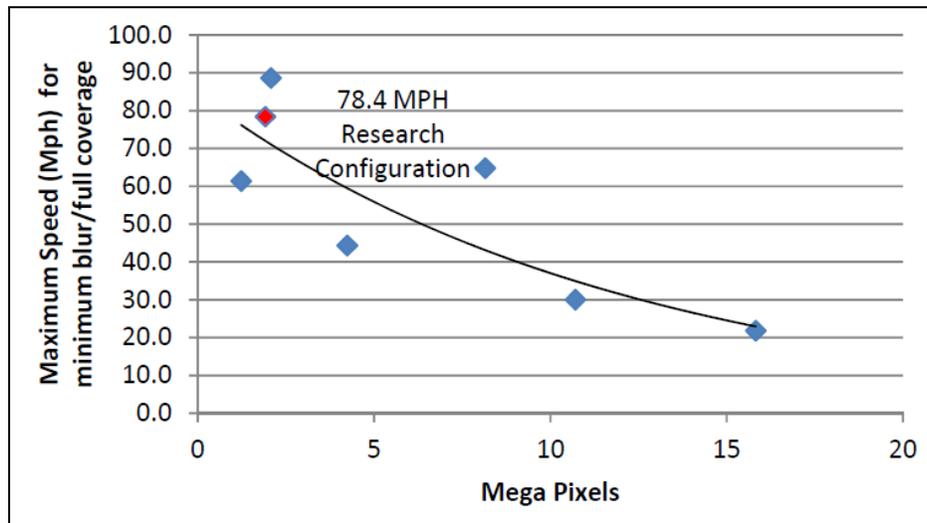
An MIT graduate student focused her 2016 master’s thesis on autonomous pavement distress detection through the use of drones. The experiment was conducted in small areas in Cambridge and Somerville, Massachusetts. A 3DR Solo Drone was used, flying at a speed of

1 meter per second at an altitude of 9 to 12 feet (5). The camera was a GoPro Hero4 (fisheye lens disabled) capturing 24 frames per second in 1080p resolution (5). Python API software was used to analyze the collected photos, while pre-programmed flight paths on Google Maps ran autonomously (5). The program converted images into grayscale, filtered them for smoothness to remove image noise (erosion, dilation, opening, and closing), segmented the data into a binary image to detect variations and inverted it for analysis (6). The next step involved autonomous estimation of contours, combined with elliptical regression (6). Distress geometry is calculated through analysis of texture, circularity, compactness, and size; then a distress classification is assigned (5). Software developers must work alongside pavement experts to tailor the digital image processing so that distress geometry and severity can be correlated to the Pavement Condition Index. This process is most valuable for identifying potholes and is not as useful for the identification of the many other pavement distresses because of the software's generation of a high rate of false positives. (5). An onboard computer was mounted to the drone, but it was "unable to process the pothole classification algorithm fast enough for the tool to be realistically deployed in real time"; running about 10 times slower than would be required for real-time feedback (5). The hardware used in this study was sub-optimal, but the report gives an in-depth explanation of digital image processing.

#### **2.3.4 U.S. Air Force Institute of Technology**

The U.S. Air Force Institute of Technology prepared an extensive report in 2015, which discussed the viability of UAS paired with high-resolution cameras to assess pavement condition. The test area was a 2,000-foot-long airbase runway that displayed a wide range of various types and severities of pavement distresses (7). The size, type and resolution of camera used in this study were all taken into consideration, since higher megapixel cameras operate at slower maximum frame capture rates (7). This means that the UAS must fly slower to capture adequate data if a higher resolution camera is used (7). Furthermore, smaller capacity cameras must take more pictures to equalize the decrease in frame exposure time, which plays a role in determining optimal velocity (7). Another design constraint is battery life, which is directly related to flight time, velocity and quality and quantity of data collected (7). During the course of this study, a graph was created (shown in Figure 2.1) which shows the relationship between maximum flight velocity (while still attaining full coverage) and number of megapixels in a given camera (7). The software interprets the intensity of light values given off by pictures of the pavement and characterizes it within certain thresholds (7). Once these threshold values can be paired with corresponding distress types and severity, a reliable UAS Pavement Condition Index will be more attainable. The software has the capability of identifying pixels that correspond to a crack in the pavement and then assess whether there are any similar pixels within a certain range of pixels next to the already identified "crack-pixel" (7). This feature essentially maps the pixels of interest together, which can be used to identify alligator-cracking and longitudinal cracking (7). The software completes this task by "choosing the smallest distance edge and retaining it only if it connects a new node. Edges connecting already visited nodes are removed. Once all nodes are connected, the algorithm is completed" (7). Unfortunately, when a success rate for this method was graded by the F-Measure method, it displayed only 40 percent accuracy (7). The F-Measure method is a function of the recall and precision rates (both functions of the true positives and false positives and negatives captured by the algorithm) as compared to the

ground-truth image (7). Many different iterations and approaches were taken for analyzing this dataset to maximize accuracy, but spectral imaging may prove to be a useful tool for improving this type of assessment.



Source: Grandsaert, P. Integrating Pavement Crack Detection and Analysis Using Autonomous Unmanned Aerial Vehicle Imagery. United States Airforce Institute of Technology, 2015 (7)

**Figure 2.1: Maximum speed of image capture characterized by megapixels of camera**

### 2.3.5 Sidewalk Condition Assessment

In a 2017 article submitted to the Institute of Physics Conference on Materials Science and Engineering, Ahmet Ersoz headed an experiment using UAS photogrammetry and automated software to assess the condition of a concrete pavement sidewalk. A DJI Inspire 1 Quadcopter operating at different altitudes ranging from 0.5 meters to 3 meters with a speed of 1 to 3 meters per second was used with all photos stored on an onboard SD card (8). A digital image processing software known as an SVM (Support Vector Machine) used specific indexes like aspect ratio and eccentricity; but it still needed “training” from a human operator to hone in its threshold values (8). ASTM D5340 (Standard Test Method for Airport Pavement Condition Index—Surveys). Once the SVM model was properly trained, its assessment was compared to a “boots on the ground” field survey (8). The results were comparable; however, the author admits that the test area was an easy trial course since it did not exhibit a wide range of types and severity of distresses (8). They state that using only photogrammetric data collection leads to dramatically obscured data when there is light interference from shadows or different light angles (sensitive to location, time of day, and season) (8). Ersoz argues that “the main advantage of the system is that it offers a cost-effective solution compared to currently used systems such as a truck-mounted road monitoring system, as the UAS are getting cheaper and easily transportable” (8). He follows to say that the digital image processing algorithms still need significant improvement (8).

## **2.4 UAS Platform and Instrumentation Requirements**

---

Hanson Engineering produced a brief overview on current UAS technology in 2016, which listed quadcopters and fixed wings as the two ideal types of drones (9). They cite two experiments that were conducted to assess the accuracy (compared to traditional surveying methods) of UAS photogrammetry for the purpose of creating a topographical map. The brief overview stated that UAS can be used for accurate mapping as well as for bridge inspections, roadway asset inventories, and airports pavement condition surveys.

## **2.5 Data Collection and Reduction**

---

As noted in the previous case studies, data reduction is widely varied. Currently, automatic data reduction is still in the development stage. Most case studies utilized some manual interpretation of the results.

## **2.6 Cost Savings**

---

No data was available on the cost savings of using UAS for pavement condition analysis.

## **2.7 Benefits of UAS for Pavement Condition Analysis over Traditional Methods**

---

No comprehensive studies have been conducted to compare UAS over traditional pavement condition collection methods.

## **2.8 Closing Summary of UAS for Pavement Condition Analysis**

---

The use of UAS for pavement condition surveys is in its infancy with very little experience and information available. Photogrammetry appears to be the front runner of UAS pavement condition assessment technology, but the use of multi- or hyperspectral imaging is garnering attention in this field of study as it presents more and more benefits when used in conjunction with photogrammetry. LiDAR usage in drones has not been thoroughly explored yet, although it does propose some potential advantages that may be seen at a later time. Understanding how to consistently determine a pavement condition index (PCI) is essential to this process. A better relationship must be established between qualified engineers who can reliably define and determine the PCI of a given section of pavement, and the

programmers who are coding these algorithms; so that a more consistent system may be put into place. The “threshold values” and various assigned distresses need to be better integrated into a method that closely matches ASTM D5340. The “threshold values” and various assigned distresses need to be better integrated into a method for all types of pavements (highway, airport, etc.).

## 3.0 Review of Deep Learning Applications in Pavement Distress Analysis

Due to the various means of obtaining pavement images and the widely varied features associated with pavement distress, classification of pavement distress images presents a significant challenge for computer vision. Recent advances in computer vision and deep learning, however, have led to the development of deep-learning models for detecting distress in road pavement images. The models most recently used have been primarily centered on convolutional neural networks (CNN) (10). In this brief literature review, the current state of the art in CNN-based pavement distress detection is summarized. This review starts with discussing some of the key results noted in a recent review paper (10) on this topic and then examines a few other important papers that are not included in that review. Although these early studies are mostly for detecting the existence of pavement distresses and classifying them, the findings are still useful for developing a model that can detect pavement distresses at the pixel level and ultimately convert them into PCI measures.

### 3.1 Summary of Gopalakrishnan's Review Paper

---

In 2018, Kasthurira Gopalakrishnan published a review paper citing some of the most recent studies in the sub-field of CNN-based pavement distress detection and categorization (10). CNN is a special type of the well-known feedforward neural networks and is widely used in image analysis. Similar to feedforward neural networks, a CNN also consists of an input layer, multiple hidden layers, and an output layer. In feedforward neural networks, neuron in different layers are fully connected, meaning each neuron in a layer is connected to all neurons in the next layer. While for CNN, the hidden layers perform a sequence of convolution and pooling/subsampling operations on the input data, and are not fully connected. The results in some of the key works reviewed by Gopalakrishnan are summarized below.

One paper reviewed by Gopalakrishnan (11) used deep learning to eliminate images of non-interest within a street-view image database. The NVIDIA Deep Learning GPU Training System (DIGITS) was used to perform this work. The author (10, 11) found that this deep-learning model was able to generate results that are 3 percent more accurate than the well-known CrackIT (12) tool.

Zhang et al. (13) developed a CNN model to detect pavement cracks and tested it based on a dataset of 500 images of size 3264x2448 captured using a smartphone. These images were broken up by the authors into 1 million 99x99 RGB images. Of these 1 million images, 640,000 were used for training the CNN, 160,000 were used for five-fold cross validation, and 200,000 were used for testing. This CNN utilized the ReLU activation function and was developed using Caffe. It made use of dropout in between layers in order to avoid overfitting.

The stochastic gradient descent (SGD) method was used for model training with a batch size of 48. The CNN achieved the best performance after 20 epochs of training (10, 13).

Pauly et al. (14) argued that a deeper CNN structure would present better results for pavement crack detection. They used the same 500 images as in Zhang et al. (13) and examined the effects of CNN depth on crack detection. The CNN was developed with four convolutional layers, four subsampling or max pooling layers, and two fully connected layers. The SGD optimization method was employed with a batch size of 48 for model training. The CNN was developed using Keras with a TensorFlow backend. Pauly et al. concluded that the deeper CNN architecture did increase pavement distress detection success, but did not present strong results when tested on images from new locations (10, 14).

Eisenbech et al. (15) developed the open source GAPS (German Asphalt Pavement Distress) dataset, which consists of 1,969 grayscale images. They utilized 1,418 images for training, 51 for validation, and 500 for testing. Initially, they utilized the same CNN that Zhang et al. (13) developed. Later, a deeper network structure was considered which consisted of eight convolutional layers, three subsampling or max pooling layers, and three fully connected layers. They tested several regularization techniques including dropout, batch normalization, and weight decay. The experiments took three months on a computer with two NVIDIA Titan X GPUs. The CNN was developed using Keras with a Theano backend. In general, the CNN achieved higher generalizability when employing the dropout or batch normalization techniques to avoid overfitting (10, 15). This result is not necessarily surprising, but it is important to be noted for future work in the field.

Maeda et al. (16) developed a smartphone App called RoadDamageDetector to detect pavement distress. In their study, they distinguished between eight types of pavement distress. Their work utilized a smartphone mounted on a car dashboard. As the car was driven along the road at approximately 40 km/hr, the phone captured one image each second. These images, which were initially size 600x600, were downsized to 300x300 before being used for training the CNN. In order to develop the CNN, the Single Shot MultiBox Detector (SSD) with Inception V2 and SSD using MobileNet frameworks were utilized. The smartphone App was then developed on a Nexus 5X smartphone. The RoadDamageDetector App achieved a precision of greater than 75 percent (10, 16). The 9,053 images used in this study are also publicly available.

Fan et al. (17) used images from two publicly available databases, the CFD database and the AigleRN database. To train the CNN, they broke up these images into overlapping 27x27 patches surrounding each pixel. These patches served as the input to the CNN. They developed a CNN with four convolutional layers, two subsampling or max pooling layers, and three fully connected layers. All layers used the ReLU activation function except for the output layer which utilized the sigmoid activation function. The Adam optimizer was used with a batch size of 256 for model training, and a dropout rate of .5 was employed along with weight decay to avoid model overfitting. The CNN was trained on the AigleRN database for 43 epochs and on the CFD database for 13 epochs. The CNN was developed with TensorFlow (10, 17). They conducted experiments on each database to confirm the CNN's generalizability. First, they trained the network on the CFD database and tested it on the

AigleRN database. Then, they trained the CNN on the AigleRN database and tested it on the CFD database. Finally, they trained and tested the CNN on a hybrid database of images from both the CFD and AigleRN databases. The results showed good generalizability, especially when the network was trained on the hybrid database. They concluded that training on images with differing pavement conditions helps the CNN to achieve stronger generalizability (17).

Rather than developing a CNN from scratch, Gopalakrishnan et al. (18) applied transfer learning to a pre-trained CNN. They utilized the VGG-16 CNN, which was pre-trained on the ImageNet database. They truncated the VGG-16 CNN, and only used the convolutional layers of the pre-trained CNN but not the fully connected layers. Then, a new fully connected layer was trained on the features recorded by the truncated VGG-16 CNN. They utilized 1,056 images from the FHWA's publicly available pavement performance database. The original images from this database, size 3072x2048, were downsized to 1000x500. The Adam optimizer was used with a batch size of 32. A dropout rate of .5 was also utilized. The CNN was trained for 50 epochs. They developed this transfer learning model using Keras. The best results were generated by a single layer classifier which was trained on the pre-trained VGG-16 CNN (10, 18).

Gopalakrishnan et al. also showed the potential of utilizing pre-trained models with transfer learning for pavement distress detection based on UAS (drone) collected images (10,19). They conducted the analysis using a Keras implementation of the VGG-16 CNN. The size of the images in their study was 224x224. Dropout with a rate of .5 was used to avoid model overfitting, the ReLU activation function was used in the fully connected layer, and the Softmax activation function was used in the output layer. They trained the model for 50 epochs with a batch size of 32. Their results showed that, by truncating the VGG-16 CNN before the fully connected layer and by adding a new fully connected layer, they achieved an accuracy of 89 percent in crack detection (19). The potential for using pre-trained models and transfer learning may be useful for this research.

### **3.2 MatConvNet**

---

Cha et al. (20) reported using MatConvNet, an open source MATLAB toolbox for developing CNN, for the task of crack detection. They utilized 277 images of size 4928x3264 for training and validation of their CNN. These 277 images were broken down into 40,000 images of size 256x256 and were annotated as either containing cracks or not. They then designed a CNN using MatConvNet which they trained and validated on these smaller images. The CNN utilized the ReLU activation function and was trained using SGD with a batch size of 100. Both dropout and batch normalization were employed to avoid overfitting. For training the CNN, 32,000 256x256 images were used, and the other 8,000 were used for validation. The accuracies in the training and validation phases were 98.22 percent achieved at the 51st epoch of training, and 97.95 percent achieved at the 49th epoch of validation.

Their CNN was also tested on 55 images of size 5888x3584 in order to determine if a sliding window technique would allow it to detect cracks in images larger than 256x256. This testing

showed similar results as in training and validating the CNN, with nearly 97 percent accuracy achieved in crack detection. Cha et al. also compared their CNN with the Canny and Sobel edge detectors and found that the CNN significantly outperforms these edge detectors for identifying cracks in pavement images.

### **3.3 CNN for Crack Length Estimation**

---

Tong et al. (21) attempted to use a CNN not only for crack detection, but also for measuring crack length. In their study, the original 8,000 images were broken down into sub-images of size 200x200. Then, the gray-scale range of the images was cut, and gray-level histograms were computed. They found that the crack features in the images had an average gray-level in the range 50-110, while the normal pavement in the images had an average gray-level in the range 110-250. After the above pre-processing steps, Tong et al. developed a CNN which consisted of two convolutional layers, two subsampling or max pooling layers, two fully connected layers, and an output layer (21).

Overall, the CNN was able to identify crack length in an interval of 1 cm (e.g., a 1–2 cm long crack, a 2–3 cm long crack) with an accuracy of 94.35 percent and a mean squared error of .2377. The model performed well for most 1-cm intervals, while the biggest issue was for classifying cracks with length in the ranges of 6-7 cm and 7-8 cm. This was likely due to a lack of images containing cracks of this length.

The CNN was then tested on new images from four highways in central China. Forty images from each highway were acquired and pre-processed in the same way as the images used for developing the CNN. The results show that their CNN has promising generalizability (21).

While the results presented by Tong et al. show the promise of using CNN for crack length identification, one step in their pre-processing method is unclear. That is, they were able to utilize gray-level histograms to distinguish between crack features and non-distressed pavement in their images. However, this result seems to contradict that of Chambon and Moliard (22), who found several years earlier that the histograms of pavement images which contain cracks have only one gray-level peak. Thus, Chambon and Moliard argued that the gray-level histogram of an image could not be used to distinguish a crack from non-distressed pavement (22). At this point, it is unclear why there is this discrepancy in the results of these two studies (21, 22).

### **3.4 Closing Summary of Deep-Learning Applications in Pavement Distress Analysis**

---

This brief review shows that many deep-learning methods have been developed for detecting pavement cracks. While these methods utilize different deep-learning frameworks such as Caffe, TensorFlow, and Theano, it is important to note that the ReLU activation function seems to be the most commonly used. Also, stochastic gradient descent was employed by

most studies for model training, and it seems that dropout between layers is important for boosting CNN generalizability. The successful use of transfer learning models by Gopalakrishnan et al. (18,19) may also be directly relevant to the current UAS and pavement distress detection work at UMass Lowell. Additionally, Tong et al. (21) were able to successfully use a CNN to perform crack length identification. While the study by Tong et al. presents important results for crack length identification using CNN, it is unclear how they were able to achieve such high success by using gray-level histograms to separate crack features from non-cracked pavement, and the generalizability of their model is uncertain.

In summary, this review suggests that CNNs have been widely applied to the task of crack detection and present a promising way forward for pavement distress analysis. Also, the study by Tong et al. (21) shows the potential for CNNs in the task of pavement crack length measurement. Additionally, this review suggests that to develop CNN-based crack length detection algorithms, it is important to have a large database of pavement images with cracks clearly labeled. A key aspect of the work by Gopalakrishnan et al. (18, 19), transfer learning, appears to be an encouraging technique to address the limited labeled pavement distress sample size issue. Overall, the methods reviewed in this report are mostly for identifying and classifying cracked regions, not for directly measuring crack lengths. These methods, however, can be combined with traditional image processing techniques to isolate cracks from cracked regions and to estimate crack lengths.

A very time-consuming step in developing CNN based pavement distress detection models is to label cracks in the training image dataset. The studies reviewed in this report used data from various sources, including smartphones (13, 14, 16), open source image databases (15, 17, 18), and drones (19). It is possible to directly utilize the labeled images in these studies. However, this would restrict the new model to have the same input layer dimension as those models in the previous studies. Another issue with the existing datasets is that some images were taken by smartphones mounted on a dashboard. It is difficult to identify and label small cracks from such images manually. Therefore, the trained CNN models will not be able to identify them either. Lastly, these existing training datasets were labeled for identifying distressed regions (i.e., boxes were drawn around areas with cracks), while the ultimate goal of this research includes estimating crack length. Therefore, the existing datasets are useful, but still need to be further processed (i.e., label cracks only, not regions that have cracks) and expanded to cover additional crack features.

This page left blank intentionally.

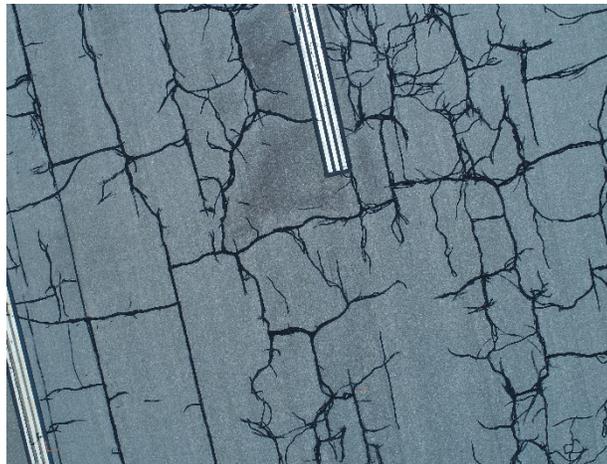
## 4.0 Data and Analysis

Based on the literature review, this research identifies and experiments with two methods for pavement crack detection. One is the MATLAB toolbox CrackIT (23) and the second one is a deep convolutional neural networks approach (24, 25). Both methods have been applied to the pavement image data collected from the Fitchburg Municipal Airport (FMA) in Massachusetts using drones and some additional datasets obtained from the internet. With the kind support of MassDOT Aeronautics Division, the team made two data collection trips to FMA on November 1, 2018 and March 1, 2019 respectively. Prior to these two data collection trips, the team received some pavement images collected by MassDOT Aeronautics Division using drones. For the first data collection trip, the team focused on evaluating the pavement images provided by MassDOT and comparing them with the pavement conditions in the field, and the team concluded that resolution of the received drone images is insufficient to provide useful pavement distress information (see Section 4.1.1). In the March 2019 field trip, MassDOT Aeronautics Division helped the team collect a second set of pavement images using drones. The data and the analysis results are presented below.

### 4.1 Data Sources

---

#### 4.1.1 Fitchburg Municipal Airport (FMA) Data



**Figure 4.1: Drone image from the initial data set**

The team obtained two sets of data from FMA. The first set of data was collected from an altitude of 120 ft and provided by the MassDOT Aeronautics Division. A sample image is presented in Figure 4.1. The team applied CrackIT and a deep learning method to this initial data set. However, the results were unsatisfactory. The team then visited the FMA and took some pictures of the pavement using a cell phone (see Figure 4.2). Clearly, the cell phone image covers a smaller area but is much clearer than the drone photo taken at 120 ft above the ground. The previous unsatisfactory crack detection results may likely be attributed to the

poor image resolution, not necessarily the methods adopted. Therefore, the team decided to collect additional pavement image data instead of trying to fine tune the existing CrackIT and deep learning models.



**Figure 4.2: Pavement image taken by a cell phone**

To investigate how height may affect the image quality, the MassDOT Aeronautics Division helped the team conduct another data collection at FMA. Table 4.1 below shows how the second set of data was collected. The *elevator* flight captured pavement images at a single location but at different altitudes using both visual and thermal cameras. The purpose was to find out how altitude may affect the image quality (e.g., inches on the ground per image pixel). With such information, the team was hoping to identify a threshold altitude, beyond which the collected data may become much less useful. Note that such a threshold may depend on the types of drone and camera being used. Therefore, the conclusion in this study may not be generalized to other cameras and drones. The *coverage* flight was designed to collect additional runway pavement images for crack detection model development and testing.

**Table 4.1: Flight plan for the second drone data collection**

Flight Description	System	Height (ft)	Coverage
Elevator Flight Using both Visual and Thermal Cameras	DJI Matrice 210/XT2 Dual	10-70 (at a 5 ft interval), 80-150 (a 10 ft interval), and 200	Single Point
Coverage Flight Using a Visual Camera	DJI Inspire 2/X4S	32, 40, 50, and 60	Half Runway
	DJI Inspire 2/X4S	70, 90, 120, and 200	Full Runway

Figure 4.3 shows samples of pavement images collected at different altitudes through the elevator flight. It can be seen that once the altitude is greater than 120 ft, the pavement crack details are very difficult to see. This explains why the team’s previous modeling effort was unsuccessful.

Choosing the drone flight altitude needs to consider both the image resolution and the efficiency. The FMA runway is about 150 ft wide. At the 10 ft altitude, it will take a drone 10 runs to cover the entire width of the runway (each run covers a width of 15 ft). While at the 50 ft altitude, the drone just need 2 runs (a width of 78 ft for each run). Figure 4.4 shows



10 ft



40 ft



50 ft



60 ft



70 ft



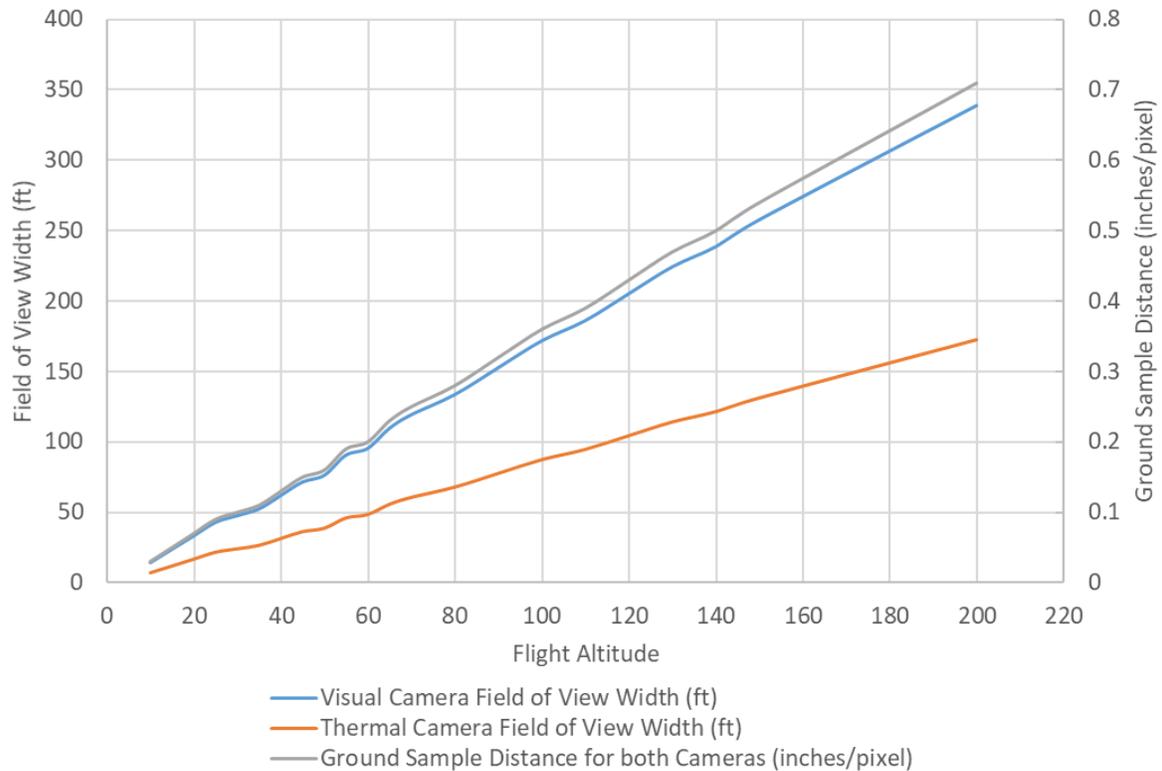
120 ft



150 ft

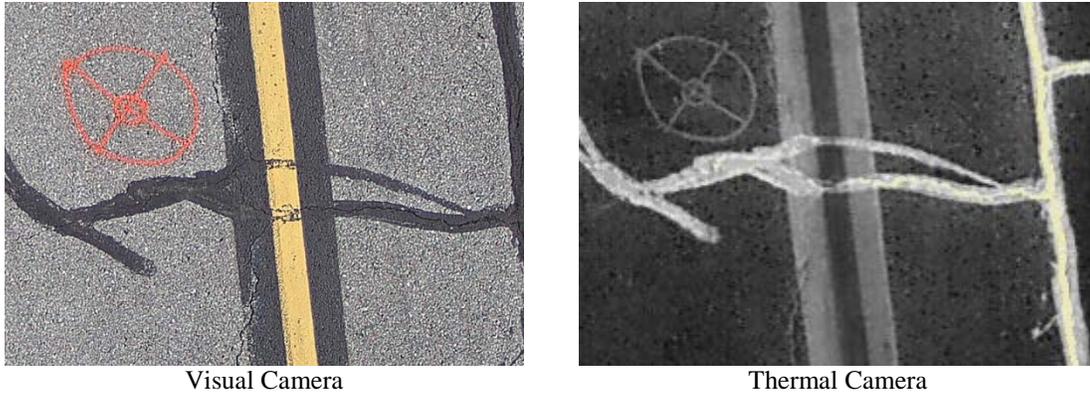
**Figure 4.3: Pavement image taken by the elevator flight**

the relationships between flight altitude and field of view width and ground sample distance measured in inches per pixel (smaller ground sample distances mean better image quality) for the visual and thermal cameras used in this research. Note that both the visual and thermal cameras used in this study have the same ground sample distance performance. Again, depending on the cameras used, this conclusion may be different. However, the overall trend will be the same: higher altitudes will require less flights (i.e., better efficiency) but result in poorer image resolutions. To achieve a balance between image quality and efficiency, the team decided to use the images captured at 50 ft for this research.



**Figure 4.4: Ground sample distance and field of view width vs. flight altitude**

During the elevator flight, the team collected images using both visual and thermal cameras and the results are compared in Figure 4.5 below. In the thermal image, pixels associated with pavement cracks do not appear to exhibit clearer characteristics compared to those in the visual camera image. For deep learning based crack detection models, cracks in the training images have to be marked at the pixel level. This process typically is done manually. Therefore, being able to identify cracks with naked eyes is essential. Given the quality of the collected thermal images, the team decided to only focus on the visual camera data.



**Figure 4.5: Pavement image taken by the elevator flight**

#### 4.1.2 Other Data Sources

The team also obtained additional data from three other sources:

- **UMass Amherst Dataset (UAD):** Prof. Charlie Schweik and his student Ryan Wicks helped the team collect pavement crack images using a MicaSense RedEdge M multispectral camera. The images were taken at the north-west corner of Parking Lot 26 of UMass Amherst campus. A DJI Phantom 4 was used to carry the RedEdge M payload. The camera was mounted on a fixed platform with no gimbal such that the camera was angled forward approximately 10-15 degrees.
- **German Asphalt Pavement Distress (GAPs) Dataset (26):** The GAPs dataset is captured using a mobile mapping system called S.T.I.E.R., which is manufactured and operated by a German engineering company LEHMANN + PARTNER GmbH. The S.T.I.E.R has been designed and certified by the German Federal Highway Research Institute (BASt) and complies with the German Road Monitoring and Assessment (RMA) standards.
- **Crack500 (27, 28):** This dataset is collected by a group at the Temple University using a smartphone. It consists of 500 pavement pictures and each picture is of size 3264 pixels  $\times$  2448 pixels.

The GAPs and Crack500 datasets contain both the original pavement images and the annotated images as shown in Figure 4.6. The annotated images mark cracks at the pixel level and are essential for training/teaching neural networks models how to detect cracks. In Figure 4.6, black pixels in the second column represent cracks. Preparing the annotated images is very time-consuming. Therefore, it is beneficial to have datasets that include both the original and annotated images.

The UMass Amherst dataset contains multispectral images taken at altitudes ranging from 20 feet to 390 feet. Similar to the thermal images collected at the FMA, this Amherst dataset suggests that multispectral images do not seem to reveal more crack information than images captured by visual cameras. Considering the significant efforts needed to annotate the thermal images, the team decided not to include the Amherst data in the modeling process.

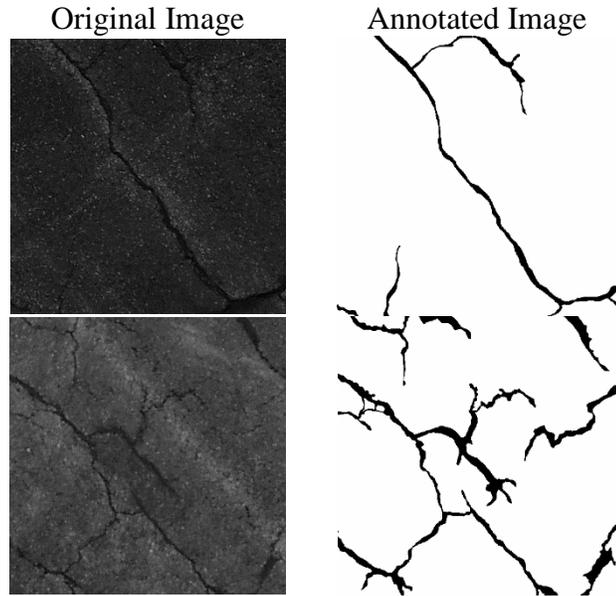


Figure 4.6: Sample original and annotated pavement images

## 4.2 CrackIT for Pavement Crack Detection

The team initially applied the MATLAB-based CrackIT toolbox to the initial FMA dataset. CrackIT provides many functions and routines to aid in the task of automated crack detection. A basic flow diagram describing the CrackIT method is shown in Figure 4.7.

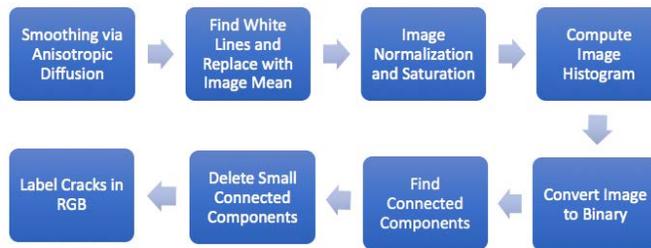


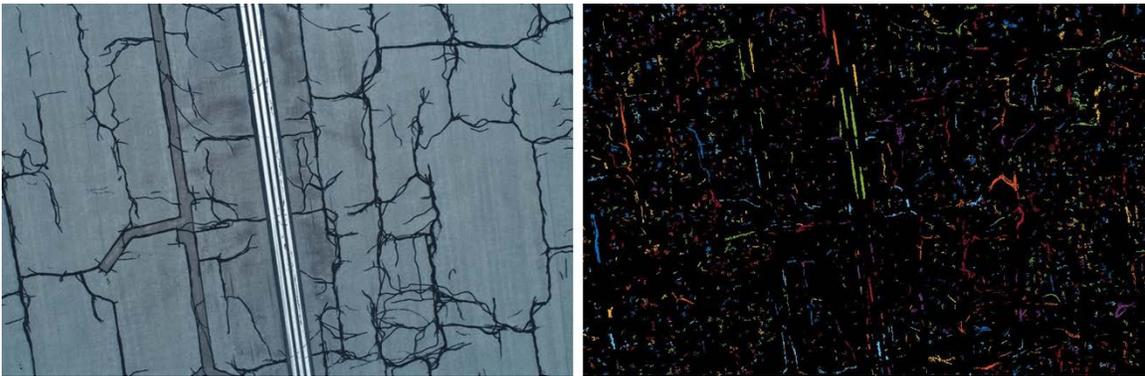
Figure 4.7: Flow diagram of CrackIT detection method

The toolbox only operates on grayscale images, so once an original image from the FMA dataset is read into MATLAB, it is converted to grayscale using the MATLAB function “rgb2gray.” Initially, the toolbox was applied “out of the box” on a sample image from the FMA dataset, and the detected results were poor. On the first attempt (Figure 4.8), CrackIT detected no cracks in the sample image. Only the black background region was returned with no detected cracks.



**Figure 4.8: First attempt to apply CrackIT to FMA sample image**

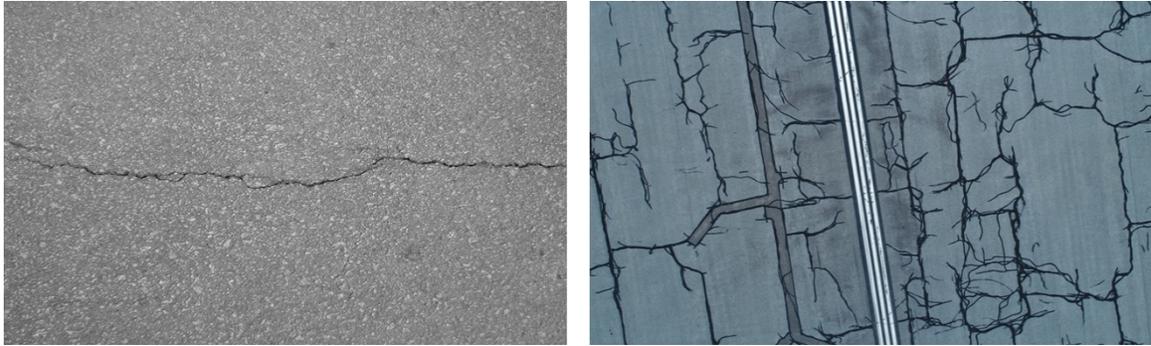
After adjusting some parameters in the CrackIT functions (primarily those used for white line detection, pre-labeling of crack regions, and converting the image to binary), CrackIT returned a very noisy RGB map which contained little meaningful crack detection data shown in Figure 4.9.



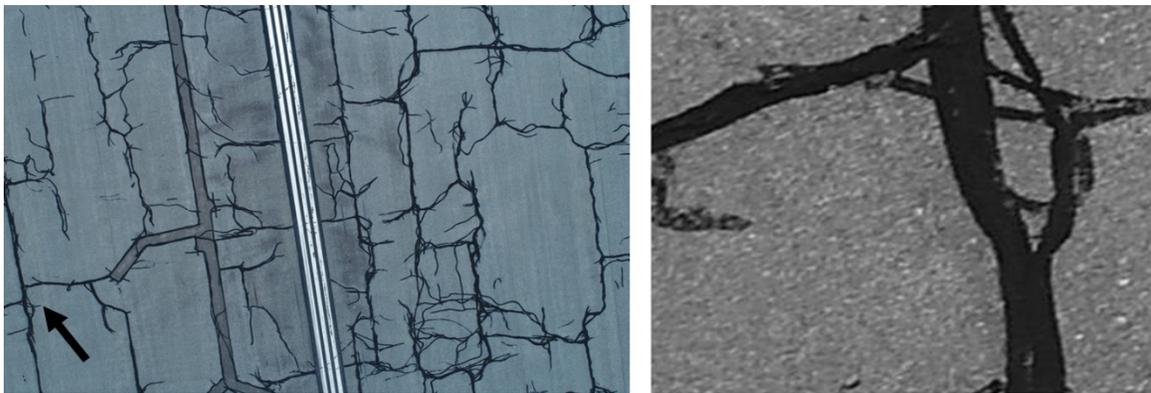
**Figure 4.9: First attempt to fine tune CrackIT parameters**

There are some differences between the images in the CrackIT database and the images in the FMA dataset, such as the overall complexity of the images as well as the presence of many filled cracks in the FMA dataset as shown in Figure 4.10. Additionally, the images in the FMA dataset were captured using a drone, while the images in the CrackIT database were obtained using traditional road survey methods. Thus, it does not seem surprising that CrackIT “out of the box” did not perform well on the FMA data.

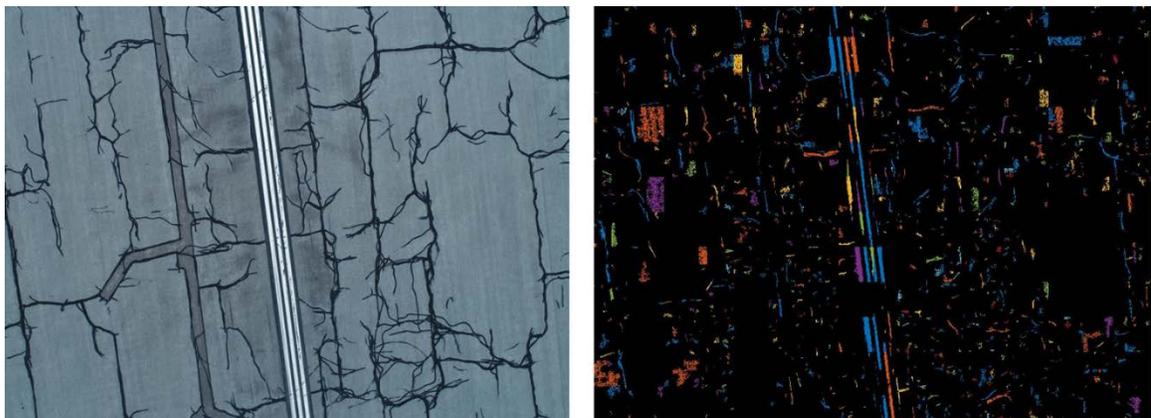
To address this issue, sample images from the FMA dataset were resized and broken down into smaller chunks. The initial size of the images was 3648x4864 pixels. The images were resized to 3600x4800 pixels, and were then broken down into 192 non-overlapping sub-images of size 300x300 pixels as shown in Figure 4.11.



**Figure 4.10: L: Sample image from the CrackIT database; R: Sample image from the FMA dataset**



**Figure 4.11: Example chunk from FMA image**

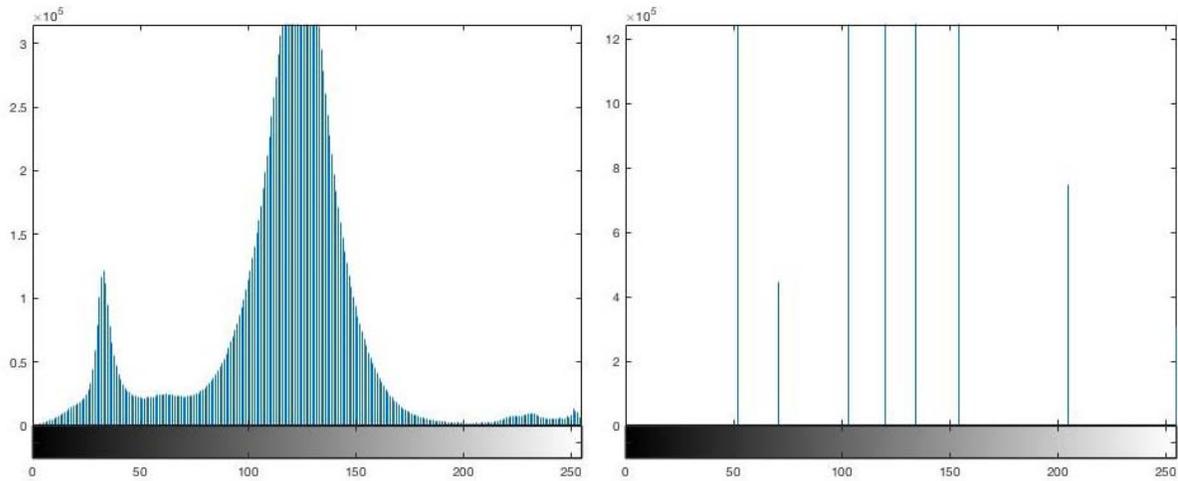


**Figure 4.12: Results of chunking image into 192 nonoverlapping chunks of 300x300 pixels and applying CrackIT on each chunk**

CrackIT was applied on each of these chunks individually, and the chunks were then brought back together to reconstruct a full RGB map of the cracks in the image. This divide-and-conquer approach improved upon the results of applying CrackIT “out of the box” on the full image from the FMA dataset, but the improvement is insignificant. In particular, there were

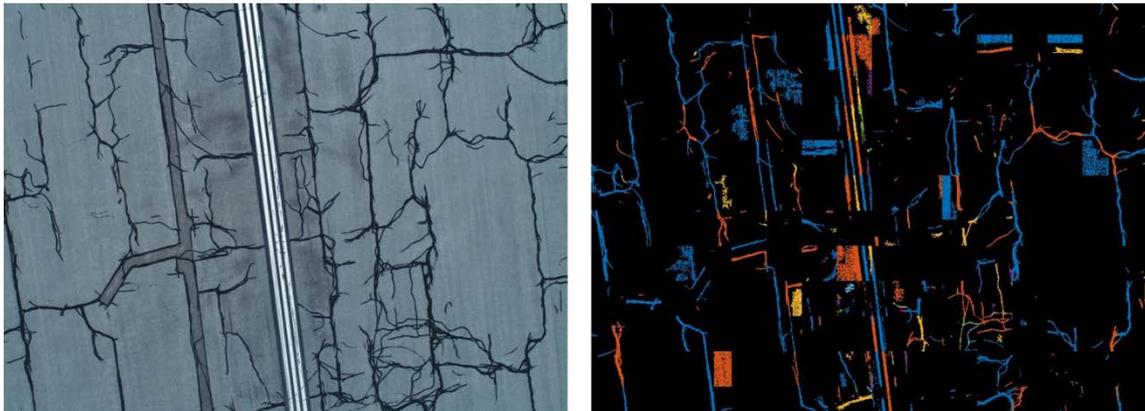
still many regions of cracks which were not detected, as well as substantial issues with noise/error in the RGB crack map as in Figure 4.12.

To further simplify the image which was fed into the CrackIT algorithm, an eight-level thresholding was applied to the image before chunking. This eight-level thresholding, which is done utilizing the MATLAB function “multithresh,” maps each pixel in the image to one of eight possible pixel values, greatly simplifying the distribution of pixels in the image and in some sense de-noising the image as shown in Figure 4.13.



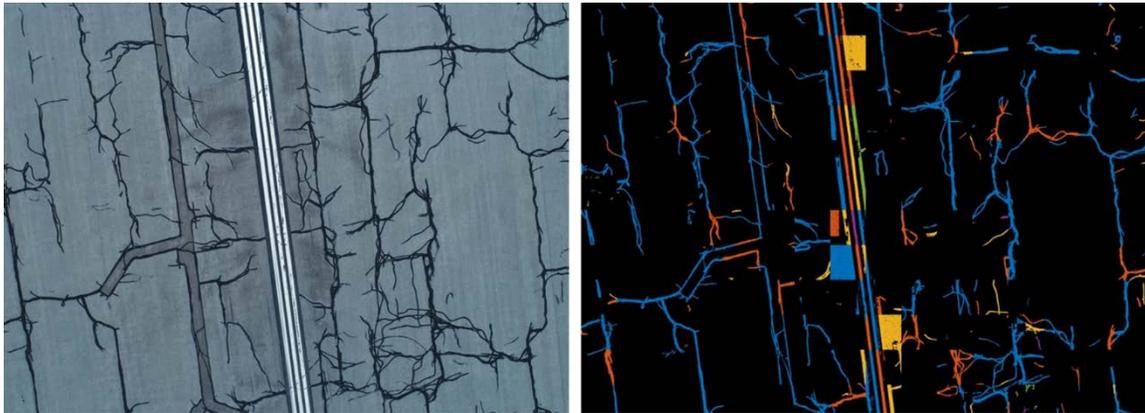
**Figure 4.13: Comparison of image histogram before (L) and after (R) applying multilevel thresholding**

Once this thresholding was complete, the image was again broken into 192 chunks and each chunk was individually processed by the CrackIT algorithm. This eight-level thresholding step improved the results of applying CrackIT on the complex data from the FMA, leading to both increased and improved crack detection as well as a significant reduction in noise in the RGB crack map. A sample of the result is provided in Figure 4.14.

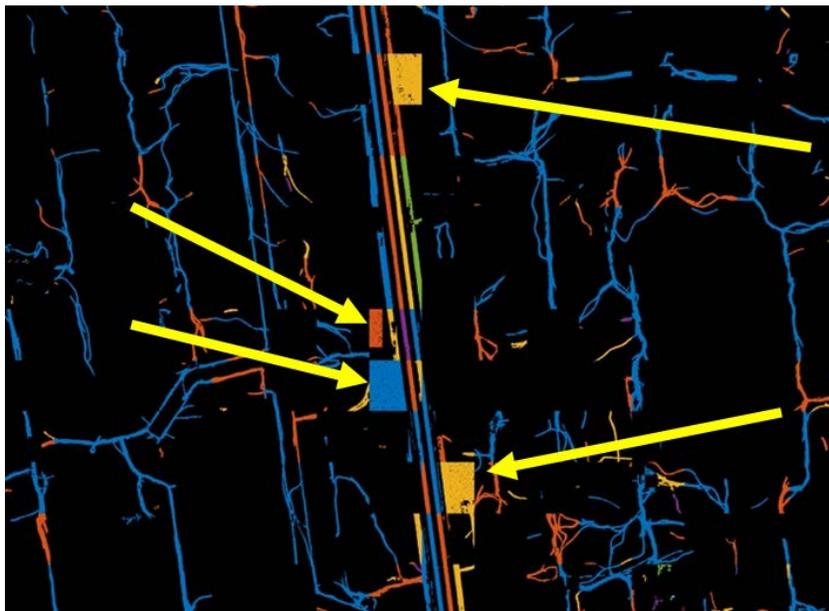


**Figure 4.14: Results of adding multilevel thresholding preprocessing step to CrackIT**

A significant number of smoothing parameters have been tested to further improve the crack detection results. In each of the above trials, the smoothing parameters passed to the CrackIT anisotropic smoothing routine consisted of 4 iterations of smoothing using a conduction coefficient of 60 and a speed of .125. Further testing revealed that 2 iterations of smoothing with a conduction coefficient of 80, and a speed of .125 seemed to improve the crack detection results. Additionally, further testing at larger numbers of iterations with smaller conduction coefficients revealed that 16 iterations of smoothing with a conduction coefficient of 30 and a speed of .125 seemed to improve the results further, at least for some images, and at this point, these are considered to be the best parameters for smoothing. Figure 4.15 shows the results based on the best smoothing parameters.



**Figure 4.15: Results of CrackIT with best smoothing parameters**

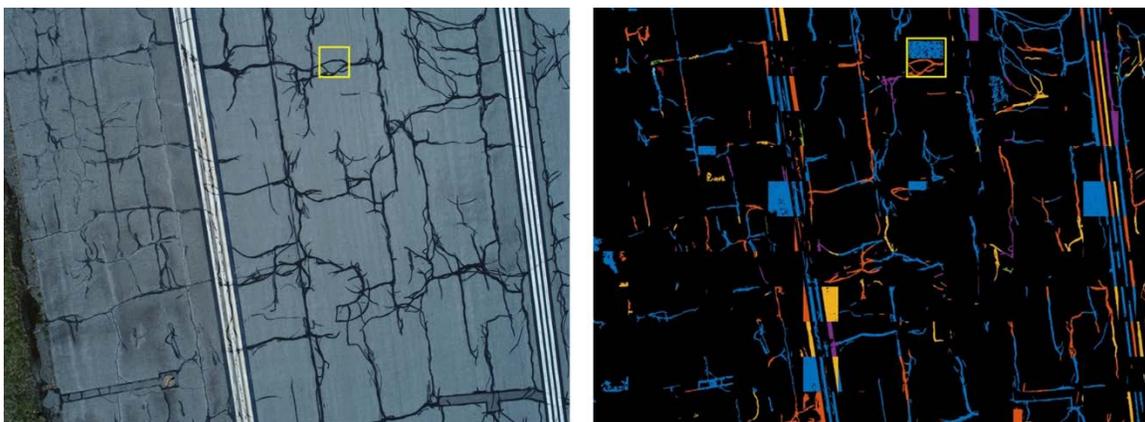


**Figure 4.16: Error/false detection region**

Some issues still remain. The first is the detection of the painted black lines adjacent to the white lines on the runway as crack regions. The second is the detection of false positive/error regions (see those blue, orange, and yellow regions marked in Figure 4.16) in the RGB crack map by the CrackIT algorithm. These false positive regions do not resemble any structures present in the original image, are generally square or rectangular in shape, and fairly densely filled. It is unclear why the algorithm detects these regions as pavement distresses.

Because of the chunking method used in the processing of the FMA images, each of these false positive regions is generally contained only within one 300x300 chunk of the RGB crack detected image. For the step in CrackIT of converting the image to binary, CrackIT provides two possible options which correspond to two different MATLAB methods for converting to binary. If the method used for binary conversion in these regions of false positive is changed from “imextendedmin” to “im2bw” (a change which corresponds to passing a 1 to the CrackIT function “computeBinImg” instead of a 2), these regions of false positive are greatly reduced or eliminated. However, if this method for binary conversion is utilized on every chunk of the original image, then very few cracks are detected and the resultant RGB crack map returns a poorer representation of the cracks in the original images. Thus, at present, there is no consistent way to eliminate these regions of false detection/error.

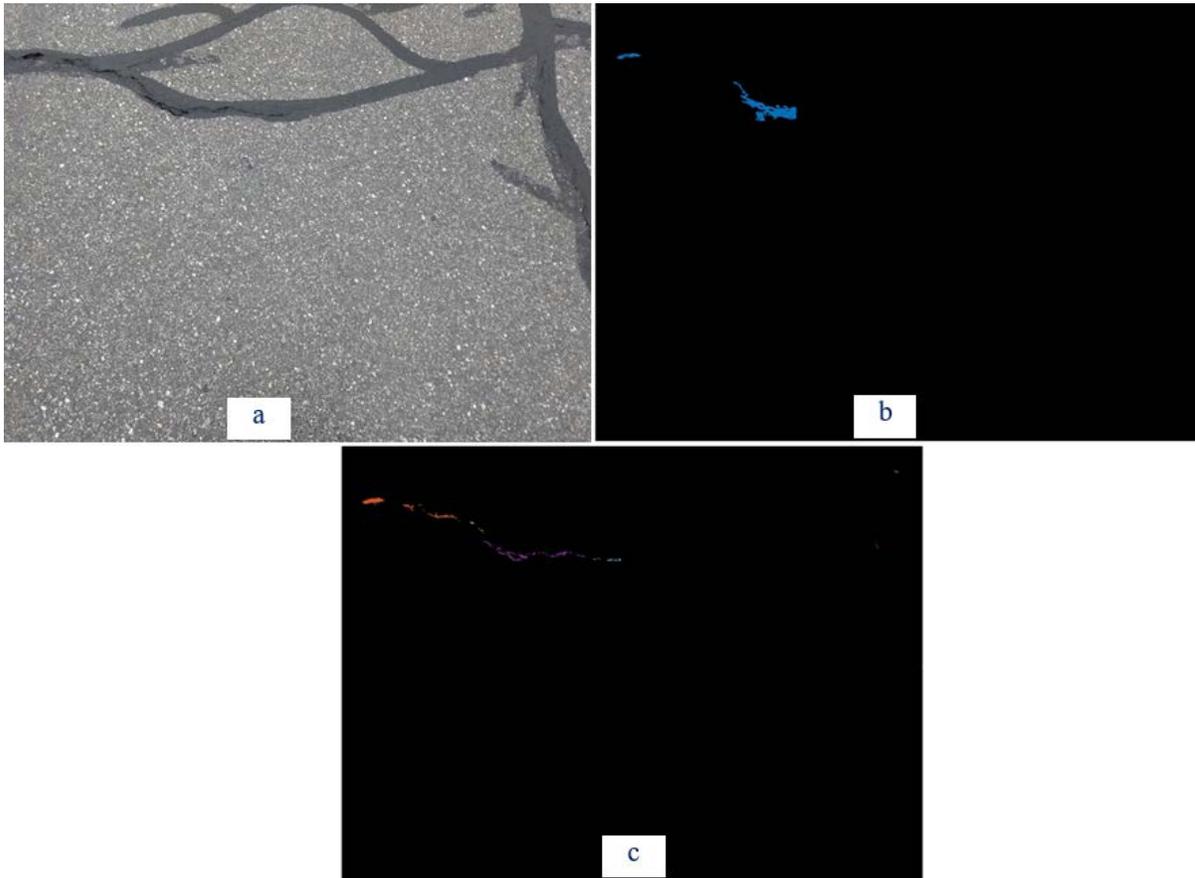
To further investigate these regions of false positive, a survey of several of these regions (one of them is highlighted in Figure 4.17 below) was conducted during a field trip to the FMA. In general, these regions of false positive do not show any signs of square-like distresses. Some of the regions show some signs of pavement wear, but this is inconsistent from region to region. Additionally, images of these regions were captured with cell phones. These cell phone images were processed using CrackIT, both using the parameters found to work well on the FMA dataset and using CrackIT “out of the box.” The results of a sample image are presented in Figure 4.18. CrackIT “out of the box” returned better results on these cell phone images, which is not surprising given that these images more closely resemble those in the CrackIT database.



**Figure 4.17: Example of false detection area inspected during field trip**

From Figure 4.18, it is clear that there are no square-like regions detected in these cell phone images. Upon inspection of the pavement at FMA and comparison with the drone images, it seems that the drone dataset does not present a completely accurate portrayal of the pavement

condition. For example, small, non-filled crack regions which are visible when walking on the pavement at the airport (and would be visible during a traditional pavement survey) are too small to be visible in the drone images from the initial data collection. A lower flyover height might help to improve the results of applying CrackIT to the drone images. Alternative smoothing methods, such as that of (29), have also been tested but have not yet improved the results. With further testing, alternative smoothing methods might also lead to improved results.



**Figure 4.18: a) Cell phone image of false detection area; b) CrackIT applied on cell phone image using best parameters from processing FMA dataset; c) results of applying CrackIT “out of the box” on cell phone image**

Some work on crack depth detection has also been conducted utilizing the Pix4D platform. Once a set of images from a drone flight are loaded into Pix4D, the software reconstructs a map of the terrain. From this reconstructed map, it is possible to calculate volume data for objects in the image. For example, utilizing a sample project provided with Pix4D, it is possible to calculate the volumes of houses and cars in the images. However, volume calculations of crack regions using Pix4D have not been successful, with the software generally computing a volume of 0 for any given small crack region. This difficulty is not surprising, given the large difference in the height of drone (e.g., 50 ft) flyover when compared with crack depth (e.g., 1 inch). Additionally, it is not surprising given the difficulty in visualizing open crack regions in the drone images.

## 4.3 Deep Neural Networks for Pavement Crack Detection

---

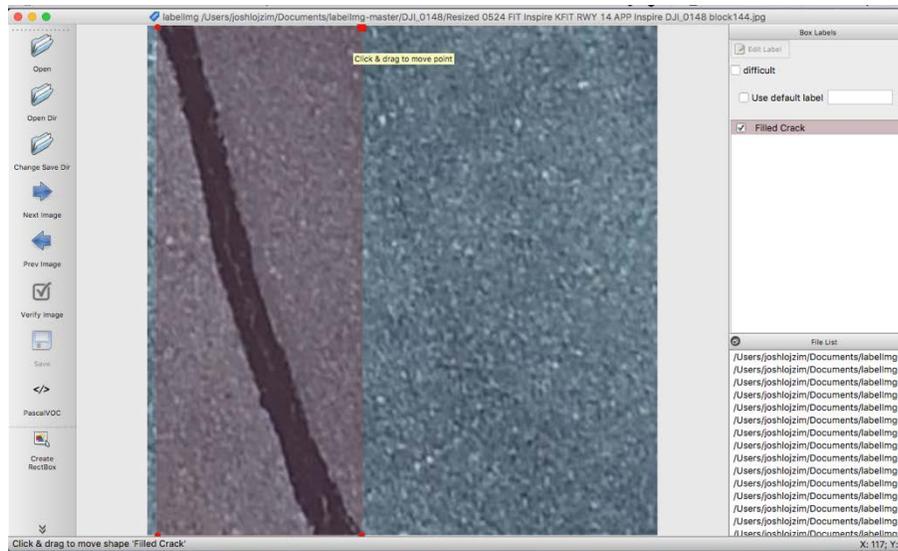
The team also researched two deep neural networks approaches. The first one aims at detecting cracks using the transfer learning technique to avoid the time-consuming image annotation process. The second one is based on a promising deep learning approach called U-Net (24). The detailed modeling processes are described in the following two subsections.

### 4.3.1 Crack Detection

Developing deep learning models for object (e.g., crack) detection usually requires a huge set of training data. To address the lack of training data issue, transfer learning is often adopted, which is an important and useful means of adapting pre-trained deep learning models to new datasets, either when the new dataset is small, when prototyping time is important, or both. In (18) and (19) transfer learning was applied to crack image classification using the VGG16 Convolutional Neural Network (CNN) as the base network. This network was initially trained on the ImageNet Database (31), and was used as the base network for training a crack image classifier. In this present work, transfer learning is applied to the task of crack object detection. The work reported in this subsection is based on the initial dataset collected at FMA.

In order to accomplish this goal, many steps were followed. First, a platform and CNN was selected for transfer learning. In this work, the Tensorflow Object Detection API (32) was selected as the platform for training a new object detector on the initial FMA dataset. This API provided a variety of models which were pre-trained on various massive image databases such as COCO (33). In this work, the SSD Mobilenet\_V1 network pre-trained on COCO was selected as the base network for training the new detector for the FMA data. This network architecture was also utilized for training a crack detector by Maeda et al. (30).

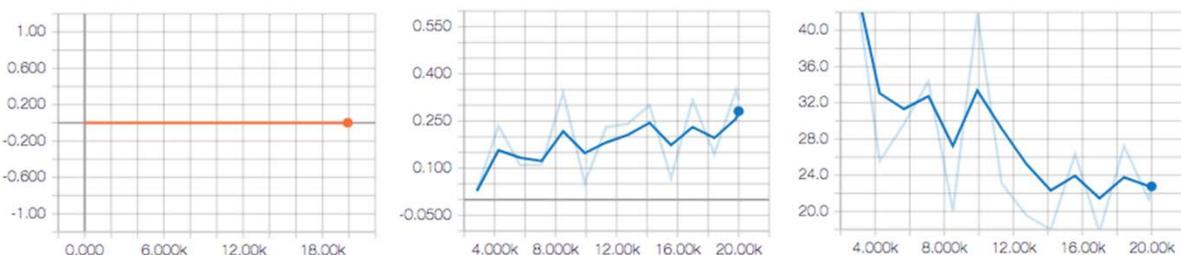
In order to train an object detection network, images were labeled with ground truth bounding boxes which differentiate crack and non-crack regions. Additionally, the SSD Mobilenet network was trained using images of size 300x300 pixels. To prepare a dataset to develop the object detector, three sample images from the initial FMA dataset were resized from 3648x4864 pixels to 3600x4800 pixels, and were then broken into 192 chunks of size 300x300 (just as was done above in the section on CrackIT and shown in Figure 4.11). Each of these 300x300 chunks was then labeled with ground truth bounding boxes using the Python tool LabelImg (34). An example is shown in Figure 4.19. Note that in Figure 4.19, the filled crack area is annotated by a rectangular bounding box. Therefore, utilizing this SSD Mobilenet network model, cracked areas were detected and marked also by a rectangular box instead of at the pixel level as in Figure 4.6.



**Figure 4.19: Screenshot of utilizing LabelImg to draw ground truth bounding boxes**

LabelImg automatically generates xml files which contain the image annotation (bounding box) data. However, to utilize the Tensorflow Object Detection API, the image data must be in TFRecord format. Thus, these xml files of bounding box data were converted to TFRecord files, which could then be utilized for transfer learning. In total, 576 images were used, with 490 (~85%) utilized for training and 86 (~15%) for testing.

The first attempt to develop a model on this data was trained for 20,000 steps, utilized a constant learning rate of .0001, did not use data augmentation in the pre-processing, did not utilize dropout, and eventually attained a mAP (mean average precision) of  $\sim .3176$ .<sup>1</sup> There seemed to be issues with the model overfitting the training data, with the testing loss ultimately at a high value of  $\sim 22$ . The results of this model are shown in Figure 4.20, in which the x-axes are for neural network training iteration, and the y-axes are for learning rate, mAP, and loss function value at each iteration, respectively. The learning rate, mAP, and loss function value do not have units.

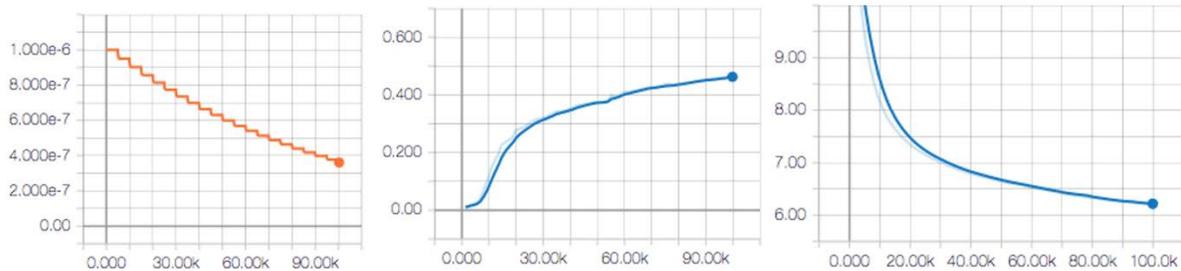


**Figure 4.20: From left to right: Learning rate, mAP, loss for the first attempt**

In order to deal with the issue of overfitting, data augmentation pre-processing steps were added to the training of the model in the second attempt. These included random flipping and cropping of the training data. After 100,000 training steps with an initial learning rate of

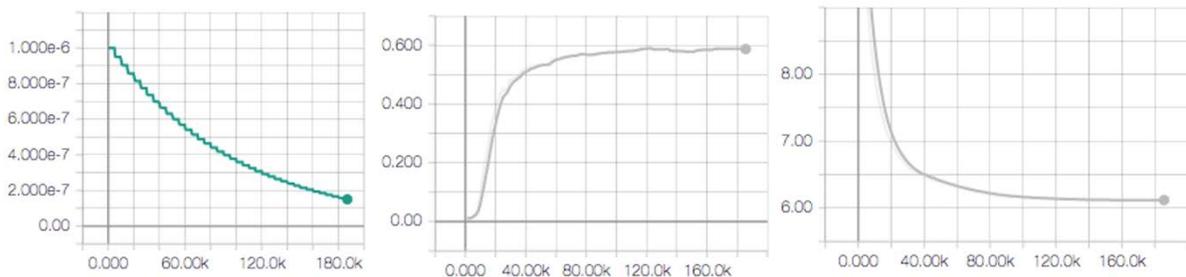
<sup>1</sup> mAP is calculated at an IoU threshold of .5.

.000001, decaying to an ultimate value of  $3.7735e-7$ , and utilizing dropout at a rate of .2, the network achieved a higher mAP of .4653 as in Figure 4.21.

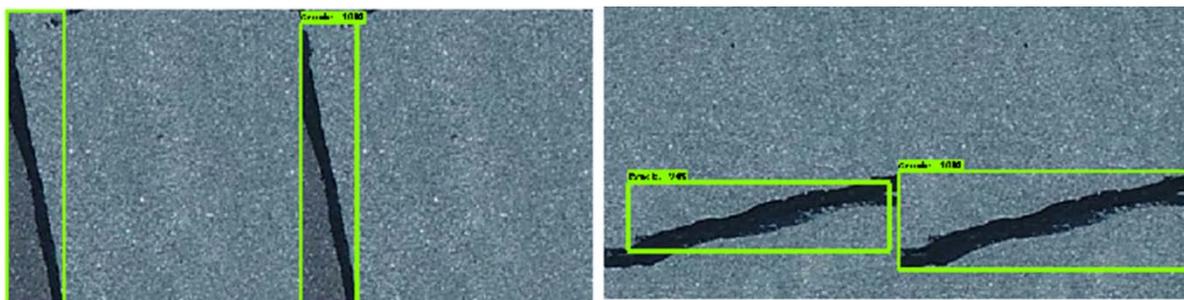


**Figure 4.21: From left to right: Learning rate, mAP, loss for second attempt**

To test the model further, a new random split of the training and testing data was conducted in the third attempt. A model was trained and tested on this new split for ~185,000 steps with a learning rate of .000001, which decayed every 5,000 steps to a final rate of  $1.4989e-7$ . Data augmentation was utilized, as was a dropout rate of .2. Ultimately, a mAP of .5867 was achieved on this split as in Figure 4.22. This was a further improvement over previous results.



**Figure 4.22: From left to right: Learning rate, mAP, loss for third attempt**



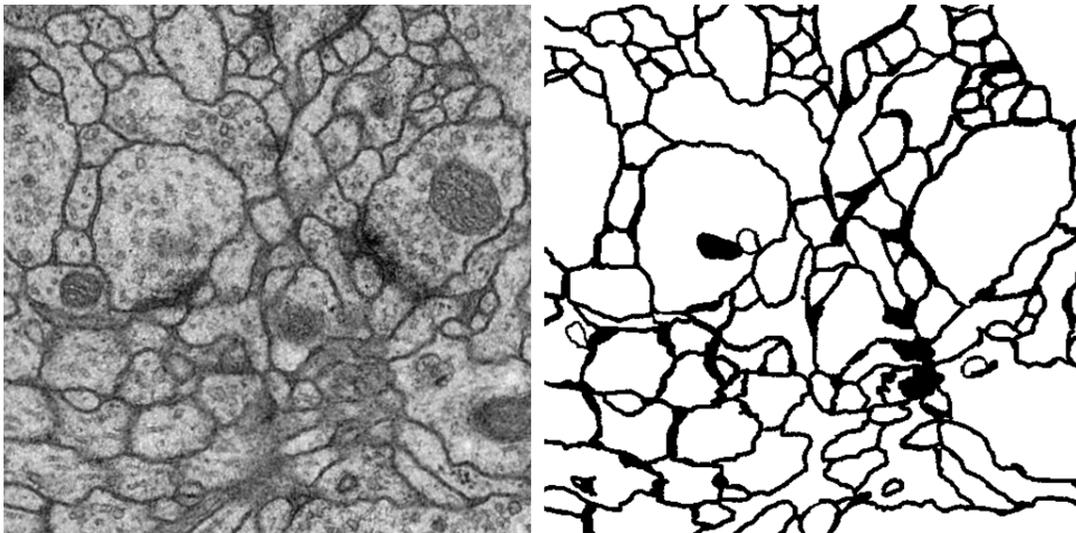
**Figure 4.23: Two example results of transfer learning: Right subimage of each pair is the ground truth, while left is bounding box predicted by neural network.**

One issue for the Tensorflow Object Detection API based crack detector is the detection of small, non-filled crack regions. The performance of deep learning methods depends heavily on the training data. There are significantly more filled crack regions present in the initial FMA dataset than non-filled regions. Therefore, the developed detector is unable to

consistently or accurately detect non-filled cracks. In order to continue to improve the results of the transfer learning object detection method, several potential steps may be taken including generating more ground-truth labeled (i.e., boxed) training data, continued hyperparameter fine-tuning, and possibly utilizing a different base network for transfer learning. Another issue is that the Tensorflow Object Detection API method only boxes the crack regions and does not tell how long or wide those cracks are. To address these issues, a U-Net deep learning model (24) is adopted and described in the next subsection.

### 4.3.2 Crack Segmentation

Given the promising performance of deep neural networks for image segmentation and object detection, this study applied a U-Net model to analyzing pavement images. U-Net is a CNN architecture initially developed for segmentation of neuronal structures in electron microscopic stacks. In its original application, it was utilized to learn a segmentation mask of neuronal boundaries (24) in Figure 4.24.



**Figure 4.24: Example of neuronal image and segmentation**

The task of segmenting neuronal boundaries is similar to the task of segmenting cracks and creating a map of the cracks in a pavement image. The Keras implementation of U-Net (25) was utilized in this work. The team started to experiment with the U-Net model using the initial set of FMA data. Since this U-Net network was designed for grayscale images of size 512x512, all input images were first converted to grayscale. Some of these images were then resized to be broken up into non-overlapping blocks of size 512x512 if needed.

Annotating pavement images for U-Net is at the pixel level, and it is even more time-consuming than labeling images for the Tensorflow Object Detection API in the previous subsection. Therefore, this study utilized the Crack500 dataset that consists of 2,244 already annotated images. In addition, this study selected some drone images obtained at the 50 ft altitude during the second FMA data collection trip. These images (5472 pixels x 3648 pixels) were further divided into smaller pieces (512 pixels x 512 pixels), and 160 of them were randomly chosen. These new images were then annotated manually (as illustrated in

Figure 4.6) and used for model training and testing. Table 4.2 below shows how the input data was separated into training and testing datasets.

**Table 4.2: Data used in U-Net modeling**

Dataset Name	Total Images	Training Images	Testing Images
FMA	160	70	90
Crack500	2,244	1,896	348

#### 4.3.2.1 Measurement of Performance

Crack segmentation is essentially a classification task at the pixel level. Furthermore, it is a difficult classification problem due to the highly imbalanced instances, i.e., much less crack pixels than non-crack pixels. To accurately measure the performance of the proposed model, the following *F1 score*, *Precision*, and *Recall* metrics are presented first, and later an *Intersection over Union* (IoU) is introduced and used in this research.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (1)$$

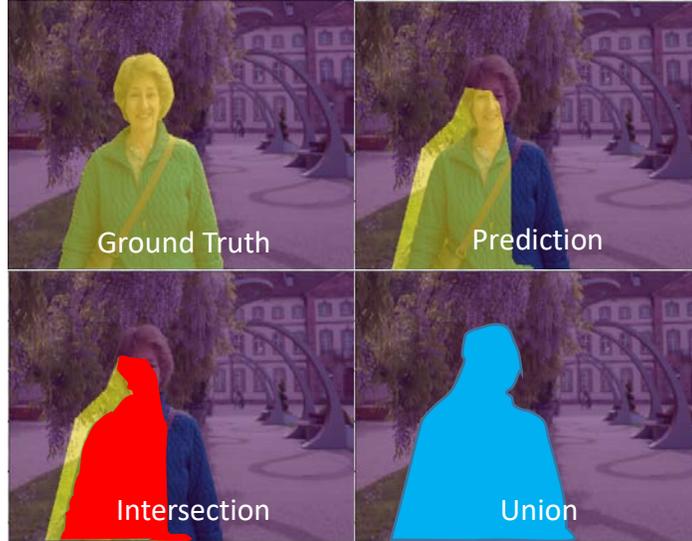
$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Where *TP* denotes *True Positive*, *FP* denotes *False Positive*, and *FN* denotes *False Negative*. *F1* considers both *Precision* and *Recall*, and is calculated as the harmonic average of *Precision* and *Recall*. *F1* score value ranges between 0 and 1. Larger *F1* scores mean better model performances.

Many previous classification studies consider *Precision* as the main performance metric, which can be problematic. *Precision* may work if different outcome categories have approximately the same number of observations. However, pavement crack datasets are often characterized by a highly imbalanced class distribution. Based solely on the *Precision* performance metric, the best-performing model may favor non-crack pixels and generate poor results for crack pixels. Such a model does not provide much useful information for crack detection and may lead to biased if not erroneous conclusions. Therefore, *F1* (see Eq. (1)) is often used in recent studies.

In this research, another intuitive and informative metric, *IoU*, is introduced and adopted. As shown in Figure 4.25 (35), *IoU* is defined as the area of intersection divided by the area of union. Intersection represents the area covered by both the ground truth and prediction, while union includes the area in either the ground truth or prediction. If the prediction matches the ground truth perfectly, then the corresponding *IoU* would be 100%.



**Figure 4.25: Illustration of how IoU is defined**

#### 4.3.2.2 Modelling

Data augmentation was employed to increase the amount of training data for the U-Net model. For hyperparameters of tuning, The Adam optimizer was utilized with a learning rate of .0001. The number of training episodes was set to 1000, and binary cross-entropy was adopted as the loss function. ReLu activation function was used for all layers except for the last one, which utilized the sigmoid function. Additionally, the batch size was set to 5.

Both the Crack500 and FMA datasets were used to train the U-Net model. To find out whether a U-Net model trained on the Crack500 dataset can perform well on the FMA dataset, two U-Net models were generated. The first U-Net<sub>Crack500</sub> model was trained solely based on the Crack500 dataset using 1,896 images (See Table 4.2). The second U-Net<sub>Crack500&FMA</sub> model was trained using 1,896 Crack500 and 70 FMA images.

The U-Net<sub>Crack500</sub> model was initially evaluated based on the Crack500 data using the remaining 348 images set aside for testing, and its performance was very promising. This model was further tested on the FMA data and resulted in less accurate performance, which is not surprising and could be attributed to the differences between the two datasets. The Crack500 dataset was collected using a camera about 7 ft above the ground, while the FMA dataset was collected 50 ft above the ground. Given the limited size of the FMA dataset, training a U-Net model solely based on it is difficult. Therefore, the U-Net<sub>Crack500&FMA</sub> model was developed and tested on the FMA data. The detailed model evaluation results are presented in the following section.

### 4.3.2.3 Results

Table 4.3 presents the performance results of the two U-Net models and compares them with the results reported in a 2019 research paper that also used the Crack500 data. From the evaluation results based on the Crack500 data, U-Net clearly outperforms ( $IoU=0.6$ ) the benchmark FPHBN model (28).

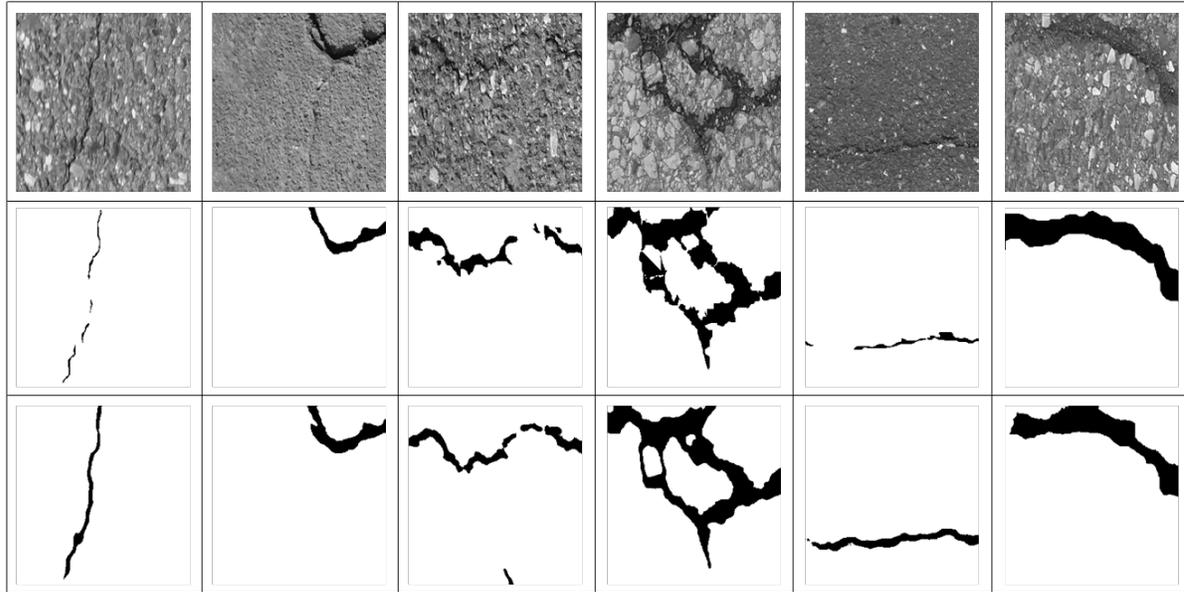
However, the Crack500 data was collected using cell phones, while the FMA data was captured using a drone 50 ft above the runway. Due to the differences between the Crack500 and FMA datasets, applying the U-Net<sub>Crack500</sub> model directly to the FMA data did not generate satisfactory results ( $IoU=0.3$ ). By combining only 70 annotated images from the FMA dataset with the Crack500 data for model training, the U-Net<sub>Crack500&FMA</sub> model was able to produce significantly better performance ( $IoU=0.56$ ) compared to the U-Net<sub>Crack500</sub> model on the FMA testing data.

**Table 4.3: Model performance comparison**

Evaluated based on	Model	Performance ( $IoU$ )
Crack500	Benchmark FPHBN model from an IEEE 2019 paper (28)	0.49
	U-Net <sub>Crack500</sub>	0.6
FMA	U-Net <sub>Crack500</sub>	0.3
	U-Net <sub>Crack500&amp;FMA</sub>	0.56

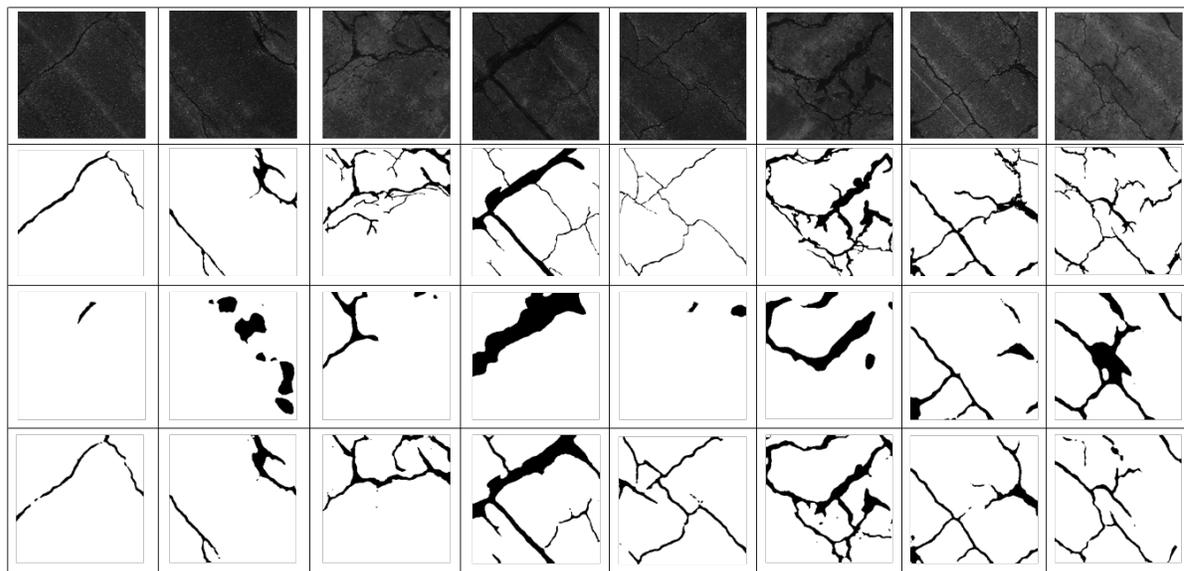
The  $IoU$  results are useful for model comparison, since large values represent better performances. In addition, Figure 4.26 is included to visually illustrate the prediction performances of the U-Net<sub>Crack500</sub> model on some randomly selected Crack500 testing data. Images from the first row are the original pavement data; images in the second row are the ground truth (i.e., true cracks); and images in the last row are the prediction results generated by the U-Net<sub>Crack500</sub> model. Comparing the ground truth and prediction results in Figure 4.26 suggests that the U-Net<sub>Crack500</sub> model performs very well and is able to accurately capture all major cracks.

Similarly, the prediction results of the U-Net<sub>Crack500</sub> and U-Net<sub>Crack500&FMA</sub> models on the FMA testing data are further illustrated in Figure 4.27. A comparison of Figures 4.26 and 4.27 clearly suggests that the Crack500 and FMA datasets are very different. Crack500 data has a higher resolution and differences between cracks and non-cracks can be easily distinguished. On the other hand, FMA data overall is darker and has a lower resolution. Some of the cracks are difficult to identify even manually with human intelligence. In Figure 4.27, the third and fourth rows show the results of the U-Net<sub>Crack500</sub> and U-Net<sub>Crack500&FMA</sub> models, respectively. Compared to the ground truth (i.e., the 2<sup>nd</sup> row), the U-Net<sub>Crack500&FMA</sub> model clearly is able to identify more cracks than the U-Net<sub>Crack500</sub> model. Given the limited time, this research only annotated 160 FMA images. If more annotated FMA images are used in U-Net training, its prediction performance is expected to be further improved.



Note: 1st Row: Original Images; 2nd Row: Ground Truth; 3rd Row: Predictions

**Figure 4.26: Prediction results of U-NetCrack500 on Crack500 data**

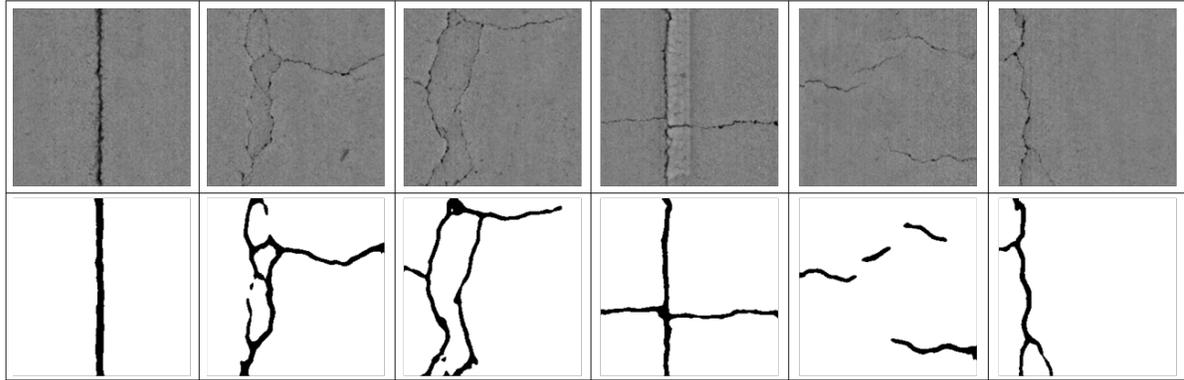


Note: 1<sup>st</sup> Row: Original Images; 2<sup>nd</sup> Row: Ground Truth; 3<sup>rd</sup> Row: U-Net<sub>Crack500</sub> Predictions; 4<sup>th</sup> Row: U-Net<sub>Crack500&FMA</sub> Predictions

**Figure 4.27: Prediction results of U-NetCrack500 and U-NetCrack500&FMA on FMA data**

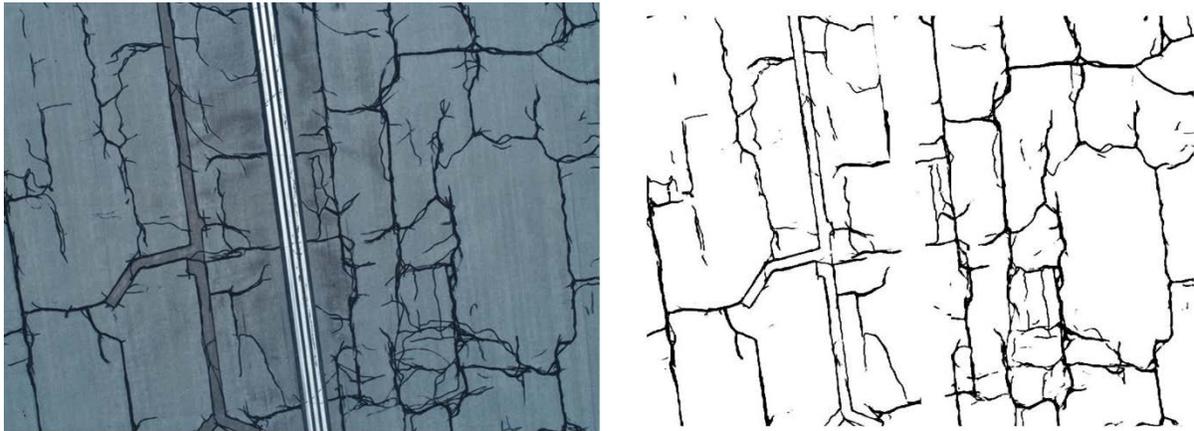
Additionally, the developed U-Net<sub>Crack500</sub> model was also applied to some pavement images generated by a laser pavement scanning system. Such a system is usually mounted on a vehicle (about 2 meters/6.6 ft above the ground) and used by many state departments of transportation to collect highway pavement condition data. The testing results are presented in Figure 4.28. The highway pavement images have not been annotated and the comparison is directly between the raw images and the predicted results. The highway pavement images

are taken about 2 meters/6.6 ft above the ground and are closer to the Crack500 data in terms of resolution than the FMA data. This probably explains why the U-Net<sub>Crack500</sub> model's performance on this dataset is very encouraging, although it is trained on a different set of images. This suggests that the U-Net<sub>Crack500</sub> model has great generalization ability and can potentially be used directly by highway departments on datasets that are similar to the Crack500.



Note: 1<sup>st</sup> Row: Original Images; 2<sup>nd</sup> Row:- Predictions

**Figure 4.28: Prediction results of U-NetCrack500 on highway pavement image data**



**Figure 4.29: Application of U-Net to initial FMA data**

Additionally, the U-Net model has been applied to the first set of FMA images (see Section 4.1.1) taken at 120 feet above the FMA runways. A subset of the images was used to train the U-Net model, which was evaluated on some separate testing images. A sample predicted image is provided in Figure 4.29. Again, the performance of U-Net appears to be very promising.

## **4.4 Closing Summary of Data and Analysis**

This pilot study collected runway pavement images using UAS from Fitchburg Municipal Airport (FMA). These images together with some online pavement image datasets were used to evaluate the crack detection performance of a MATLAB toolbox called CrackIT and two deep learning methods. Overall, the two deep learning methods outperformed the CrackIT toolbox. Both deep learning methods are able to generate satisfactory crack detection results. However, the SSD Mobilenet\_V1 network only highlights the detected cracks with rectangular boxes. On the other hand, the U-Net model can detect cracks at the pixel level, which is an important feature needed for calculating PCI values.

Deep learning methods typically require a huge set of annotated (labelled) training datasets for model development. This can sometimes be a major obstacle for the applications of deep learning models. This pilot study utilized some free pavement image data that has already been annotated together with a small set (70) of annotated pavement images captured at the FMA by a drone, and the trained U-Net model performed well on both the FMA testing data and some images collected using a laser pavement scanning system, suggesting that the develop model has great potential to be used for both airport runways and highway pavements alike.

## 5.0 Conclusions and Future Work

This research consists of two major components: (1) a review of existing studies on pavement conditions analysis and related UAS applications; and (2) evaluation of promising methods for pavement condition data analysis. The literature review results suggest that the use of UAS for pavement condition assessment is still in its infancy with little experience and information available. For integration with UAS, photogrammetry appears to be the most popular technique, while the use of multi- or hyperspectral imaging is getting increasing attention. The integration of LiDAR with UAS for pavement condition assessment has not been thoroughly explored yet, although it does show potential.

Besides the abovementioned sensing technologies, new algorithms for analyzing pavement condition data are also reviewed with a specific focus on deep learning. The results suggest that many deep neural networks models have been developed to detect cracks from images and achieved considerable success. However, most of the methods are designed to identify and highlight cracked regions with a rectangular box, instead of identifying cracks at the pixel level, which is an important step towards calculating pavement's PCI value. To develop deep neural networks based crack detection algorithms, it is important to have a large database of pavement images with cracks clearly labeled. Transfer learning appears to be one effective way to get around this issue.

The research also collects some runway pavement images using UAS from Fitchburg Municipal Airport (FMA). These images are used to evaluate the pavement crack detection performance of a MATLAB toolbox called CrackIT and two deep learning methods. Overall, the two deep learning methods outperform the CrackIT toolbox. Among the two deep learning methods, the U-Net model has the advantage of generating crack detection results at the pixel level. It also performs very well on the FMA data given a limited training dataset.

Going forward, additional research is needed to understand how to determine PCI values from the pixel-level crack detection results, and compare the PCI values with the assessment results of qualified engineers. Also, a programmable procedure needs to be established to generate reliable and consistent PCI outcomes from UAS images. Recently, multi- or hyperspectral imaging has received much attention. The integration of such sensors with UAS for pavement condition assessment could be another interesting research direction. To facilitate highway applications, fixed wing drones can be considered to further expand the coverage and speed of UAS based pavement condition data collection, and their applicability needs to be evaluated.

This page left blank intentionally.

## 6.0 References

1. Fouad, A.M., and C. Haas. Potential of Low-Cost, Close-Range Photogrammetry Toward Unified Automatic Pavement Distress Surveying. Presented at 89th Transportation Research Board Conference, Washington, D.C., 2010. <http://trid.trb.org/view/909660>.
2. Prooyen, C.V., K. Eleam, J. Shivar, and J. Gobbel. Evaluating the Use of Unmanned Aerial Vehicles & LiDAR for FAR Part 139 Inspections, Obstruction Analysis, and Airfield Maintenance. Hartsfield-Jackson Atlanta Airport. Michael Baker International. Pond & Company Architecture. <http://files.constantcontact.com/8c363cd8001/d7374c22-8afe-405c-b4a0-fcccb2b3e496>. Accessed Nov. 19, 2018.
3. Airfield Pavement Inspections Using Drones. AirSight: Hamburg Finkenwerder Aerodrome. 2015. <https://www.airsight.de/projects/item/airfield-pavement-inspections-using-drones/>. Accessed Nov. 19, 2018.
4. Zhang, C. An UAV-based Photogrammetric Mapping System for Road Condition Assessment. Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences ISPRS Congress, Vol. XXXVII, Part B5, 2008, pp. 627-631.
5. Petkova, M. *Deploying Drones for Autonomous Detection of Pavement Distress*. Massachusetts Institution of Technology–School of Architecture and Planning, 2016. <https://dspace.mit.edu/handle/1721.1/106049>. Accessed Nov. 19, 2018.
6. Herold, M. Spectral Characteristics of Asphalt Road Aging and Deterioration: Implications for Remote-Sensing Applications. *Applied Optics OSA*, Vol. 44, No. 20, 2005.
7. Grandsaert, P. *Integrating Pavement Crack Detection and Analysis Using Autonomous Unmanned Aerial Vehicle Imagery*. United States Airforce Institute of Technology, 2015. <http://www.dtic.mil/dtic/tr/fulltext/u2/a615401.pdf>. Accessed Nov. 19, 2018.
8. Ersoz, A., O. Pekcan, and T. Teke. *Crack Identification for Rigid Pavements Using Unmanned Aerial Vehicles*. Institute of Physics: Materials Science and Engineering, 2017. <http://iopscience.iop.org/article/10.1088/1757-899X/236/1/012101>. Accessed Nov. 19, 2018.
9. Comeiro, A. Utilizing Unmanned Aircraft Systems (Drones) For Infrastructure Projects. Hansen Engineering, American Water Resources Association (AWRA) Annual Conference, 2016. <https://www.awra.org/meetings/Orlando2016/doc/powerpoint/Session%2068%20910%20Comerio.pdf>. Accessed Nov. 19, 2018.
10. Gopalakrishnan, K. Deep Learning in Data-Driven Pavement Image Analysis and Automated Distress Detection: A Review. *Data*. Vol. 3, No. 28, July 2018, pp. 1-19. <https://doi.org/10.3390/data3030028>.
11. Some, L. *Automatic Image-Based Road Crack Detection Methods*. KTH Royal Institute of Technology, Stockholm, Sweden, 2016. <https://kth.diva-portal.org/smash/get/diva2:945233/FULLTEXT03.pdf>. Accessed Nov. 19, 2018.

12. Oliveira, H., and P. Correia. CrackIT—An Image Processing Toolbox for Crack Detection and Characterization. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 2014, pp. 798–802. <https://ieeexplore.ieee.org/document/7025160>.
13. Zhang, L., F. Yang, Y. Zhang, and Y. Zhu. Road Crack Detection Using Deep Convolutional Neural Network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016, pp. 3708–3712. <https://ieeexplore.ieee.org/document/7533052>.
14. Pauly, L., H. Peel, S. Luo, D. Hogg, and R. Fuentes. Deeper Networks for Pavement Crack Detection. In Proceedings of the 34th International Symposium on Automation and Robotics in Construction (ISARC 2017), Taipei, Taiwan, 2017, pp. 1–7. <https://doi.org/10.22260/ISARC2017/0066>.
15. Eisenbach, M., R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H. Gross. How to Get Pavement Distress Detection Ready for Deep Learning? A Systematic Approach. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, 2017, pp. 2039–2047. <https://doi.org/10.1109/IJCNN.2017.7966101>.
16. Maeda, H, Y. Sekimoto, T. Seto, T. Kashiyaama, and H. Omata. Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone. Cornell Univ. arXiv, 2018. <https://arxiv.org/abs/1801.09454>. Accessed Nov. 19, 2018.
17. Fan, Z., Y. Wu, J. Lu, and W. Li. Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network. Cornell Univ. arXiv, 2018, <https://arxiv.org/abs/1802.02208>. Accessed Nov. 19, 2018.
18. Gopalakrishnan, K, S. Khaitan, A. Choudhary, and A. Agrawal. Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-based Data-driven Pavement Distress Detection. *Construction and Building Materials*, Vol.157, 2017, pp. 322–330. <https://doi.org/10.1016/j.conbuildmat.2017.09.110>.
19. Gopalakrishnan, K., H. Gholami, A. Vidyadharan, A. Choudhary, and A. Agrawal. Crack Damage Detection in Unmanned Aerial Vehicle Images of Civil Infrastructure Using Pre-Trained Deep Learning Model. *International Journal for Traffic and Transport Engineering*,. Vol. 8, 2018, pp. 1–14. [https://doi.org/10.7708/ijtte.2018.8\(1\).01](https://doi.org/10.7708/ijtte.2018.8(1).01).
20. Cha, Y., W. Choi, and O. Buyukozturk. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, 2017, pp. 1–18. <https://doi.org/10.1111/mice.12263>.
21. Tong, Z., J. Gao, Z. Han, and Z. Wang. Recognition of Asphalt Pavement Crack Length Using Deep Convolutional Neural Networks. *Road Materials and Pavement Design*, Vol. 19, No. 6, 2018, pp. 1334-1349. <https://doi.org/10.1080/14680629.2017.1308265>.
22. Chambon, S., and J. Moliard. Automatic Road Pavement Assessment with Image Processing: Review and Comparison. *International Journal of Geophysics*, Vol. 2011, 2011, pp.1-20. <https://doi.org/10.1155/2011/989354>.
23. Oliveira, H., and Correia, P.L. October. CrackIT—An image processing toolbox for crack detection and characterization. In IEEE International Conference on Image Processing (ICIP), 2014, pp. 798-802.

24. Ronneberger, O., P. Fischer, and T. Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In International Conference on Medical Image Computing and Computer-assisted Intervention, 2015, pp. 234-241.
25. Zhixuhao. Implementation of Deep Learning Framework--Unet, Using Keras. Github Repository, 2018. <https://github.com/zhixuhao/unet>.
26. Eisenbach, M., R. Stricker, D. Seichter, K. Amende, K. Debes, M. Sesselmann, D. Ebersbach, U. Stoeckert, and H.M. Gross. How To Get Pavement Distress Detection Ready for Deep Learning? A systematic approach. In International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2039-2047.
27. Zhang, L., F. Yang, Y.D. Zhang, and Y. J. Zhu. Road Crack Detection Using Deep Convolutional Neural Network. In IEEE International Conference on Image Processing (ICIP), 2016, pp. 3708-3712.
28. Yang, F., L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling. Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection, arXiv preprint, 2019. arXiv:1901.06340.
29. Xu, L., Q. Yan, Y. ang Xia, and J. Jia. Structure Extraction from Texture via Relative Total Variation. *ACM Transactions on Graphics*, Vol. 31, No. 6, 2012. DOI: 10.1145/2366145.2366158.
30. Maeda, H., Y. Sekimoto, T. Seto, T. Kashiya, and H. Omata. Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone. arXiv, 2018. arXiv:1801.09454.
31. IMAGENET. <http://www.image-net.org>. Accessed June 20, 2019.
32. TensorFlow Object Detection API Tutorial. <https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/#>. Accessed June 20, 2019.
33. COCO – Common Objects in Context. <http://cocodataset.org/#home>. Accessed June 20, 2019.
34. LabelImg. <https://github.com/tzutalin/labelImg>. Accessed June 20, 2019.
35. Jordan, J. Evaluating Image Segmentation Models. <https://www.jeremyjordan.me/evaluating-image-segmentation-models/>. Accessed June 20, 2019.