Transportation Consortium of South-Central States

Tran-SET

Solving Emerging Transportation Resiliency, Sustainability, and Economic Challenges through the Use of Innovative Materials and Construction Methods: From Research to Implementation

Efficient, Low-cost Bridge Cracking Detection and Quantification Using Deep-learning and UAV Images

Project No. 20STLSU12

Lead University: Louisiana State University

Final Report

September 2021

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Acknowledgements

The authors would like to acknowledge the comments and suggestions provided by the Project Review Committee members

TECHNICAL DOCUMENTATION PAGE

1. Project No.	2. Government Accession No.	3. Recipient's Catalog No.
20STLSU12		

4. Title and Subtitle Efficient, Low-cost Bridge Cracking Detection and Quantification Using Deep-learning and UAV Images	**5. Report Date** Sep. 2021 **6. Performing Organization Code**
7. Author(s) PI: Chao Sun https://orcid.org/0000-0003-3909-0325 Co-PI: Xiangyu Meng https://orcid.org/0000-0003-3381-6690 GRA: Joshua O. Ogbebor https://orcid.org/0000-0001-6996-6493 GRA: Shaopan Guo https://orcid.org/0000- 0002-9443-1527	**8. Performing Organization Report No.**
9. Performing Organization Name and Address Transportation Consortium of South-Central States (Tran-SET) University Transportation Center for Region 6 3319 Patrick F. Taylor Hall, Louisiana State University, Baton Rouge, LA 70803	**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** AWD-003034
12. Sponsoring Agency Name and Address United States of America Department of Transportation Research and Innovative Technology Administration	**13. Type of Report and Period Covered** Final Research Report August 2020 – August 2021 **14. Sponsoring Agency Code**

15. Supplementary Notes

16. Abstract

Many bridges in the State of Louisiana and the United States are working under serious degradation conditions where cracks on bridges threaten structural integrity and public security. To ensure structural integrity and public security, it is required that bridges in the US be inspected and rated every two years. Currently, this biannual assessment is largely implemented using manual visual inspection methods, which is slow and costly. In addition, it is challenging for workers to detect cracks in regions that are hard to reach, e.g., the top part of the bridge tower, cables, mid-span of the bridge girders, and decks. This research develops an efficient low-cost deep learning-based methodology to identify cracks on bridges using computer vision-based techniques and deep learning. The Convolutional Neural Networks (CNN) deep learning method is used to identify cracks from images. In this research, a programmable drone is developed that can fly along a pre-defined trajectory. A large volume of images was collected from local bridges and pavements using drones. The collected images were preprocessed and divided into around forty thousand 256 by 256-pixel sub-images and fed into the CNN model. Data augmentation techniques are applied to increase the number of images in some cases. Parameters of the selected CNN model were optimized to obtain the best configuration. To evaluate the performance of the method, images from a different local bridge were used for testing. Research results show that with the optimized CNN model, cracks in the images can be identified efficiently and accurately. The developed methodology can also category the cracked image as slight, moderate, or severe cracking based on a pre-defined quantification index. The research outcome of this project has the potential to automate crack damage identification of bridge key components in a cost-effective manner. Also, the developed methodology is expected to facilitate crack damage identification for other transportation infrastructures, e.g., pavement and traffic sign structures.

17. Key Words	18. Distribution Statement
Programmable drone, image processing, bridge crack identification, deep learning, CNN models	No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	33	

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

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

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

APPROXIMATE CONVERSIONS FROM SI UNITS

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

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS, ABBREVIATIONS, AND SYMBOLS

AASHO	American Association of State Highway Officials
ASCE	American Society of Civil Engineers
ASHA	Asynchronous successive halving
ANN	Artificial neural networks
DIC	Digital image correlation
DL	Deep learning
FHT	Fast Haar transform
FAA	Federal Aviation Administration
GPS	Global positioning system
ResNet	Residual neural network
RGB	Red, green and blue
ReLU	Rectified Linear Unit
SELU	Scaled exponential Linear Unit
SGD	Stochastic gradient descent
SHA	Successive halving
SHM	Structural health monitoring
STRUM	Spatially tuned robust multifeature
SVM	Support vector machine
UAV	Unmanned aerial vehicle
A	Camera intrinsic matrix
R	Rotation parameter relating world coordinate to camera coordinate
t	Translation parameter between world and camera coordinates
α, β, γ	Scale and skew factors
u_0, v_0	Coordinates of the principal point
s	An arbitrary scale factor
$H_{3\times 3}$	Homography
$[X, Y, Z]$	Coordinate of a 3D point
$[u, v]$	Coordinate of a 2D point

EXECUTIVE SUMMARY

Many bridges in the State of Louisiana and the United States are working under serious degradation conditions where cracks on bridges threaten the structural integrity and public security. To ensure the structural integrity and public security, it is required that bridges in the US be inspected and rated every two years. Currently, this biannual assessment is largely implemented using manual visual inspection methods, which is slow and costly. In addition, it is challenging for workers to detect cracks in regions that are hard to reach, e.g., top part of bridge tower, cables, mid-span of the bridge girders and decks. As unmanned aerial vehicles (UAVs) become more and more popular, researchers started to resort to mages and videos from places which are hard to reach. Especially for bridges, UAVs can quickly fly to the desired locations to take images and videos. Hence, it is promising to integrate the deep learning method with UAV images to develop an automatic crack damage identification method.

This research develops an efficient low-cost deep learning-based methodology to identify cracks on bridges using computer vision-based technique and deep learning. The main objectives of this research are: (1) development of a programmable unmanned aerial vehicle that can fly along desired trajectory; (2) collection of images of target structures using a UAV camera; (3) development of a deep CNN model using collected images and their augmentation; and (4) identification of cracks using the learned deep learning model.

The Convolutional Neural Networks (CNN) deep learning method is used to identify cracks from images. In this research, a programmable drone is developed that can fly along pre-defined trajectory. A large volume of images was collected from local bridges and pavements using drones. The collected images were preprocessed and divided into around forty thousand 256 by 256-pixel sub-images and fed into the CNN model. Data augmentation techniques are applied to increase the number of images in some cases. Parameters of the selected CNN model were optimized to obtain the best configuration. To evaluate the performance of the method, images from a different local bridge were used for testing. Research results show that with the optimized CNN model, cracks in the images can be identified efficiently and accurately. The developed methodology can also category the cracked image as slight, moderate, or severe cracking based on a pre-defined quantification index. The research outcomes of this project on one hand provide a large dataset that can be used to train machine learning models to identify cracking damage. On the other hand, the developed method and the associated optimized CNN models have the potential to automate crack damage identification of bridge key components in a cost-effective manner. Also, the developed methodology is expected to facilitate crack damage identification for other transportation infrastructures, e.g., pavement, highway traffic sign, and traffic signal structures.

1. INTRODUCTION

1.1. Computer Vision-based Crack Damage Identification

Recent advances in computer vision and image processing techniques have provided an automatic visual monitoring system that can capture structural damage via processing the images or videos. This method doesn't require the incorporation of expensive sensors and is less dependent on labor work and experts' experience in comparison with traditional manual inspection methods. Various image processing techniques have been proposed for machine vision purposes including the generative adversarial network *(1)*, convolutional neural network (CNN) *(2)*, seeded region growing algorithm *(3)* and edge detection *(4)*. Recently, vision-based crack identification has been investigated and received more and more research effort.

Digital image correlation (DIC) technique compares changes of digital images at different deformation stages to measure deformation and strain and to detect cracks. However, DIC requires precise camera alignment and reference points of a target surface, which is suitable for lab testing and might not be practical for real life structures. In comparison with to DIC, vision-based crack identification method is more practical and widely accepted due to its advantages of simplicity, noncontact, cost effectiveness, and intuitive interpretation of data *(5-6)*. In recent years, computer vision-based technique is emerging as an effective tool for structural damage identification of a wide range of civil, mechanical and aerospace structures. Although computer vision and image processing techniques have been proposed for crack identification, complicated background information from a real bridge structure is always involved in the images, such as handwriting scripts during human inspection, electrical wires of sensors for health monitoring, and desultory edges of welding joints. All these image background noises will result in errors in the identification of cracks. Therefore, advanced algorithms are required for crack identification based on images with complicated information.

Application of vision-based inspection and monitoring includes deflection measurement *(7-9)*, detection of concrete spalling *(10-11)* and steel corrosion *(12)*. Existing methods to detect cracks from images include the image binarization method *(13)*, the stereo-vision method *(14)* and sequential image processing *(15)*. Abdel et al. *(16)* evaluated the performance of four methods for crack detection of bridges: fast Haar transform (FHT), fast Fourier transform, Sobel and Canny. The authors found that the FHT is the most effective technique in identifying bridge cracks. Prasanna et al. *(17)* developed an automatic crack detection algorithm STRUM (spatially tuned robust multifeature) classifier to detect cracks on concrete structures. It was found that the proposed STRUM can provide accurate crack detection of concrete structures.

To detect cracks in inaccessible areas, robotics and unmanned aerial vehicles (UAVs) were used. Ho et al. *(18)* used cameras mounted on cable climbing robotics, image processing and pattern recognition techniques to detect damage of bridge cables. It was found that the proposed method could be used to detect damage of bridge cables. Zhong et al. *(19)* used a UAV camera to detect cracks on concrete surfaces. Ellenberg et al. *(20)* used UVA camera images to quantify bridge related damaging including deflection, corrosion and cracks. The results indicated that the developed post-processing algorithms were able to extract quantitative information from UAV captured imagery.

Although the combination of UAV cameras and vision-based technique can provide damage information via graphing of inaccessible areas and extensive structures, it is still limited and time consuming to process thousands of target images to extract accurate damage information. Images directly taken using UAV cameras need to be de-noised, standardized and reconstructed for extraction of damage information. To improve image processing efficiency, machine learning techniques have been used and shown effective. In recent years, as an emerging technique, deep learning, which refers to artificial neural networks with many hidden layers for enhanced performance, is spotlighted and shown promising for efficient image processing and damage identification.

Zhao et al. *(21)* developed a traffic surveillance system using deep learning and speeded-up robust features (SURF) to track vehicles and their movements. Zhang et al. *(22)* used convolutional neural network (CNN) deep learning method for road crack detection. To overcome the challenges from real-engineering structures, e.g., lightening and shadow changes, Cha et al. *(23)* developed a CNN-based method for concrete crack detection. It was found that the proposed CNN using Canny and Sobel edge detection methods can find concrete cracks in real structures. Tong et al. *(24)* proposed a CNN-based method for crack length measurement. The authors used k-means clustering analysis to calculate the pre-extract cracks' properties which were used for training and testing. It was found that the accurate crack length recognition can be achieved. Till now, most existing literature on crack identification using vision- and machine learning-based techniques have been validated via laboratory testing. However, the images (crack and intact) used in most existing literature don't include various challenging conditions that widely exist in real-life structures (e.g., human-made markings). In addition, there will be image distortion, lightening, edges and shadow issues when using an UAV camera. All these deficiencies in existing methodologies need to be addressed. In other words, there is a lack of a framework that can implement UAV image sensing and automatic image processing for accurate damage identification of both laboratory and real-engineering structures.

Therefore, an automatic vision- and deep learning based crack detection method will be developed in this research to detect cracks among a large dataset of images recorded under field conditions. One of the key contributions of this project is the development of multiple classes including non-crack objects using training data collected online, which makes the trained deep learning model capable to cover a wide range of field environment. The proposed methodology is envisioned to facilitate the regular inspection of concrete bridges and other aging civil structures and accelerate the assessment of detailed crack distribution without losing accuracy using various cameras and vision devices, such as drones. Specific research activities include: (1) collection of a large volume of images from the Internet with subsequent categorization into five classes (intact surfaces, cracks, multiple joints and edges, single joint or edge, etc.); (2) collection of images of target structures using a UAV camera; (3) development of an image processing method and a deep CNN model using collected images and their augmentation; and (4) identification of cracks using the learned deep learning model.

1.2. Programmable Unmanned Aerial Vehicles (UAVs)

Unmanned aerial vehicles (UAVs) become popular since 2010s. In general, a quadrotor is a type of rotorcraft that uses two pairs of counter-rotating, fix-pitched blades for lift. The use of fixed-pitched blades allows quadrotor propellers to often be connected directly to four individual motors without the need for complicated linkages that control pitch. These motors are then connected in an ‘X’ configuration. To power and control the rotors, a battery and microcontroller are placed near the center of the craft. Changes to the altitude and attitude (the height and orientation with respect to the ground) of the craft are achieved by varying the speed of individual rotors. With such a straightforward design, it is easy to build vehicles that are much smaller than traditional rotorcrafts. The quadcopters for personal entertainment found applications in tourism, light show, etc. Soon, we can reasonably expect autonomous micro aerial vehicles to engage in many critical operations, namely reconnaissance, search-and-rescue, environmental monitoring, security surveillance, inspection, law enforcement, etc. Due to their aerial maneuvering ability, these robots can easily avoid obstacles, explore, create map, and monitor activities in the area. With further progress in energy storage and downscaling, they can operate in spatially restricted outdoor and indoor environments, e.g., forests or urban public areas, while still maintain the lowest disturbance to environment and the inhabitants within. Vision has been one of the key technologies that have been studied for application in navigation, mapping, surveillance and tracking due to its human-like perception capability. Drone technologies could be a potential efficient tool for bridge inspection, cracking detection, and predict structural degradation.

To evaluate the capability of drones in bridge detection, this project investigates the ability of drones to navigate and take videos and images. An extensive search on the existing commercial drones was conducted to gain knowledge of capabilities of drone inspection practices. Based on the knowledge obtained from the technical survey, the budget limitations, and future expandability for broad applications, we built a drone from scratch. Preliminary inspections of bridges were conducted by taking videos. After preliminary data were collected, the drone was controlled to take pictures of cracks on bridges. The images and videos obtained from the inspections were analyzed offline. Our developed drone can take off and land autonomously. It can follow predesigned trajectories by input the GPS location information. Finally, this project covered the merits of bridge inspection using drones, potential challenges, and conclusion, along with future idea to continue the project and extend the functionalities of the fully programmable drone.

1.3. Research Objectives and Tasks

This research project aims to (1) development of a programmable unmanned aerial vehicle that can fly along desired trajectories to take images and videos; (2) collection of large volume of images from local bridges and pavements using a UAV camera; (3) development of a deep CNN model using collected images and their augmentation; and (4) identification of cracks using the learned deep learning model. The major contributions of this research work contain the following four aspects:

(1) A thorough literature review has been conducted on the existing research on aerial imaging, image processing, segmentation, and reconstruction. Also, existing literature on computer vision-based crack damage identification has been reviewed, which is the basis of the proposed method in this project.

(2) A prototype UAV has been assembled and programed based on PX4 autopilot platform from scratch. The developed drone can take off and land autonomously. It can follow predesigned trajectories by input the GPS location information.

(3) A large volume of images (with and without cracks) has been collected from local bridges, buildings, and pavements. The obtained dataset covers images from different bridge components (decks, girders, and piers) with four representative cracking severities: intact, minor cracking,

moderate cracking, and severe cracking. It is noted that images with background noises, such as road boundary marker on were included in the dataset for identification. Also, cracking images with quantified crack width were collected. The collected comprehensive dataset of images and videos can be used to identify cracking damage of a large set of concrete bridges and pavements.

(4) A deep convolutional neural network-based computer vision methodology for efficiently identifying cracks in bridges and pavements has been developed. Four CNN model architectures, ResNet, GoogLeNet, VGG, and AlexNet were selected, and the corresponding parameters were optimized. Python based codes were developed for training, testing, and validation of the CNN models to efficiently identify cracks.

2. OBJECTIVES

The main objectives of this research are:

(a) To assemble and program a prototype UAV that can fly following predesigned trajectories based on PX4 autopilot platform from scratch.

(b) To collect a complete dataset of intact and cracked images (with minor cracking, moderate cracking, and severe cracking) from key components of local bridges, buildings, and pavements; and

(c) To propose a deep convolutional neural network-based computer vision methodology for efficiently identifying cracks in bridges and pavements.

3. LITERATURE REVIEW

3.1. Structural Health Monitoring of Bridges

A large number of bridges in the State of Louisiana and the United States are working under serious degradation conditions where cracking is a major issue that threatens the bridge structural integrity and public security. According to the 2021 ASCE infrastructure report on bridges *(25)*, 42% of all the nation's bridges are at least 50 years old and 7.5% of the bridges are considered structurally deficient. To ensure the structural integrity and public security, it is required that bridges in the US be inspected and rated every two years. Currently, this biannual assessment is largely implemented using manual visual inspection methods, which is slow and costly. In addition, it is challenging for workers to detect cracks in regions that are hard to reach, e.g., top part of bridge tower, cables, mid-span of the bridge girders and decks. It is possible that there will be cracks undetected during inspection, which might cause bridge to collapse when the undetected damage on load-carrying members is beyond the critical level.

To efficiently identify structural damage of bridges, structural health monitoring (SHM) of bridges has been an active research area for the past two decades. Basically, the SHM methods fall into two categories: model-based methods and data-based methods *(26)*. Model-based methods are commonly based on a Finite Element (FE) model representing the structure of interest. The basic idea is that once an initial FE model is created, measured data from the real structure is used to update the structural matrices (mass, stiffness, and damping) such that the updated model can accurately represent the real structure. Based on the updated model, damage detection, localization and quantification can be conducted through further model-updating using measured data from the structure under test. This method has attracted a great deal of effort in research and demonstrated effective for damage identification. Teughels and De Roeck *(27)* used an iterative sensitivity-based FE model updating method for damage detection, where the natural frequency and mode shape discrepancies obtained from ambient testing were minimized. They used damage functions to estimate the stiffness distribution and a trust region strategy to implement the Gauss-Newton method. The FE model updating was validated via data from a real prestressed concrete bridge. It was found that the bridge damage can be identified through updating the Young's and shear modulus. Moaveni et al. *(28)* tested the progressive failure of a full-scale reinforced concrete shear wall building on a shaking table. They used a sensitivity-based FE modeling updating method to detect the damage of the building during experiment. Significant uncertainties were observed in the damage identification results. To quantify the uncertainty, they used meta-modeling and analyzed the variance of five selected parameters. The authors found that the level of confidence of damage identification results depend on the uncertainty of modal parameters and the design of experiments. Although model-based methods are widely used for damage identification, they are challenged by the fact that the inverse problem solving is always computationally intensive and often ill-posed. Hence, this method requires careful regularization *(29, 30)*. In addition, this method is susceptible to the uncertainty of the measurements, the structures, and environmental variations such as temperature change.

In comparison, data-based approaches which are statistical in nature implements damage diagnosis through pattern recognition. Machine learning methods appear as a promising approach for pattern recognition and regression (26, 31-32). For instance, support vector machines (SVM) and artificial neural networks (ANN) exhibit good performance to detect structural damage. Zhang and Sun *(33)* proposed a data-driven method for multi-site structural damage identification using constrained

Independent Component Analysis (cICA). In *(33),* the authors compacted the structural damage information contained in the response into the mixing matrix by enforcing identical independent components to that of intact structures. By doing this, the cICA can significantly reduce the feature dimension and preserve all the valuable information of damage. Through a case study, they found that the mixing matrix elements can well identify multi-site damage, and the proposed method can progressively locate the structural damage. In Ref. *(34),* a machine learning method of multi-label classification was used to detect multi-site structural damage. The authors used multi-label classification method to consider the physical correlation between damage cases. Research results show that the multi-label classification method is better than the traditional multiclass classification and binary classification methods for multi-site damage identification. In *(35)*, Zhang et al. proposed a data-driven method using support vector machines to evaluate bridge scour severity. Using a sensitivity analysis, features from bridge dynamic responses were obtained and scour damage-sensitive features were selected. It was found that the proposed data-driven method can quantify the bridge scour evolution. In addition, Zhang et al. *(36-37)* developed data-driven machine learning methods to evaluate the road roughness using vibration data collected from connected vehicles. A numerical quarter-car model was established to provide required response data of vehicles. Through sensitivity analysis, critical features that were sensitive to road roughness levels were selected and used as input of an artificial neural network (ANN) model. Research results indicated that the proposed method can accurately evaluate the road roughness. While data-driven methods for damage identification have many advantages, the primary challenge is the lack of critical data or data insufficiency corresponding to realistic damage scenarios of critical structures, e.g., tall buildings and long-span bridges. To address this limitation, physics models and knowledge can be incorporated into data-driven methods to improve the performance and reduce the huge amounts of data needed for pure data-driven methods. Recently, physics -informed, -guided, or -constrained methods (These terms are similar) have been under development to combine the merits of data-based and physics-based methods in a wide range of areas, e.g., computational fluid mechanics, wind engineering, and structural health monitoring. Zhang and Sun *(38)* proposed a physics-guided machine learning method that integrates pattern recognition with FE model updating for damage identification. A physics guided neural network (PGNN) was used and a physics-based loss function was developed to quantify the discrepancy between the results from NN model and FE model updating. The authors found that the trained NN model can improve the damage identification results. Modeling uncertainty/modeling error is recognized as a primary challenge for accurate structural identification and damage detection using model updating. To overcome the negative effects caused by modeling uncertainty, Zhang et al. *(39)* proposed a transfer-learning guided Bayesian modeling updating method for damage detection. Pattern recognition was adopted to guide Bayesian model updating and supervise the identification of structural damage. They used domain adaptation to realize transfer learning to bridge the discrepancy between the physical structure and biased numerical models. Research results showed that if modeling errors exist, the transfer learning guided Bayesian modeling updating method outperforms the traditional methods in identifying damage severity.

In recent years, efficient deep learning models were developed and trained to identify images. Recently, due to the rapid development of computer vision techniques and low-cost high quality imaging devices, computer vision based SHM (CV-SHM) is attracting increasing research efforts in the SHM community. In comparison with traditional SHM methods that needs installation of sensors and cable wiring, CV-SHM has the advantage to implement monitoring of target structure over long distance and in a non-contact manner. Portable imaging devices carried by drones, robotics, or vehicles can conduct fast scanning of target structures. Therefore, the cost and labor required in monitoring can be reduced significantly using CV-SHM. Detailed reviews on CV-SHM can be found in Refs. *(40-43)*.

As unmanned aerial vehicles (UAVs) become more and more popular, researchers started to resort to mages and videos from places which are hard to reach. Especially for bridges, UAVs can quickly fly to the desired locations to take images and videos. Hence, it is promising to integrate the deep learning method with UAV images to develop an automatic crack damage identification method. The DIC techniques compare changes of digital images at different deformation stages to measure deformation and strain and to detect cracks. However, DIC requires precise camera alignment and reference points of a target surface, which is suitable for lab testing and might not be practical for real life structures. In comparison with to DIC, vision-based crack identification method is more practical and widely accepted due to its advantages of simplicity, noncontact, cost effectiveness, and intuitive interpretation of data *(5-6)*. In recent years, computer vision-based technique is emerging as an effective tool for structural damage identification of a wide range of civil, mechanical and aerospace structures. Although computer vision and image processing techniques have been proposed for crack identification, complicated background information from a real bridge structure is always involved in the images, such as handwriting scripts during human inspection, electrical wires of sensors for health monitoring, and desultory edges of welding joints. All these image background noises will result in errors in the identification of cracks. Therefore, advanced algorithms are required for crack identification based on images with complicated information.

Application of vision-based inspection and monitoring includes deflection measurement *(7-9)*, detection of concrete spalling *(10-11)* and steel corrosion *(12)*. Choi et al. *(7)* proposed a dynamic displacement vision system to measure response of unapproachable structures using a hand-held video camcorder. They verified the algorithm of the proposed method using static and dynamic testing. It was found that the proposed method can be used to precisely record dynamic displacement of structures during earthquakes. To measure structural response in an arbitrary direction, Park et al. *(8)* developed a motion capture system to measure three dimensional structural displacements. The authors used multiple cameras to measure 2D coordinates of the target and then calculated the 3D coordinates. The proposed method was validated using a reduced 3-story structure and laser displacement sensors. Feng and Feng *(9)* used a single camera to measure the structural displacements at multiple locations. They used the upsampled cross correlation and the orientation code matching template matching techniques. The authors conducted a shaking table test of a 3-story structure model and observed that the single camera can provide accurate displacement when compared with laser displacement sensors. It was also found that the vision sensor can overcome adverse environmental conditions, such as dim light, background disturbance template occlusion.

Existing methods to detect cracks from images include the image binarization method *(13)*, the stereo-vision method *(14)* and sequential image processing *(15)*. Kim et al. *(13)* proposed an image binarization-based method to detect cracks on concrete structures. They optimized the associated parameters of five image binarization methods. Research results indicated that the optimized binarization method can accurately measure the crack width and length. Lecompte et al. *(14)* used two different camera techniques to detect the cracks on the surface of a concrete beam under flexural loading conditions. The authors used the two techniques to measure the displacements at different points on the structural surface and calculated the deformations using the Green-Lagrange strain formula. Research results showed that the proposed two techniques can be used to detect the onset and development of cracks on concrete beam surface. Yamaguchi et al. *(15)* introduced a image-based method for crack detection on concrete surfaces. They proposed an image-based percolation model to extract the continuous texture. Noise reduction was also proposed using the percolation model. The model was validated using precision recall and receiver operating characteristic (ROC). Basically, image-based methods for crack detection fall into two categories: patch-based methods and pixel-based methods. In the patched based methods, a sliding window (the patch) is always used to run across the image to search for the potential cracked sub-region of the original image. In this method, machine learning and deep learning can be used to recognize cracks via cluster analysis (unsupervised learning) or classification (supervised learning). Abdel et al. *(44)* proposed a PCA-based (Principal Component Analysis) method to detect cracks using cluster analysis. To enhance the PCA method, the authors also used raw data features and local region features. They found that PCA with local region features provide the best detection accuracy. Prasanna et al. *(17)* developed an automatic crack detection algorithm STRUM (spatially tuned robust multifeature) classifier to detect cracks in image patches of concrete structures. The STRUM classifier can be selected as SVM (Support Vector Machines), AdaBoost, and Random Forest. It was found that the proposed STRUM can accurately detect cracks of concrete structures and the Random Forest classifier provides the best accuracy.

In pixel-based methods, the original image is directly processed, and the detailed crack morphology is output at pixel level. Edge detection techniques are always used to detect cracks at pixel level. Abdel et al. *(16)* evaluated the performance of four methods for crack detection of bridges: fast Haar transform (FHT), fast Fourier transform, Sobel and Canny. The authors found that the FHT is the most effective technique in identifying bridge cracks. Li et al. *(45)* proposed an integrated image processing method for extracting and segmenting cracks. The authors evaluated the method using collected images from bridges and found that the developed algorithm can accurately and efficiently detect bridge cracks. Yu et al. *(46)* adopted a Sobel detector to detect edges based on which the cracks can be identified. They found that the geometry and patterns of cracks on concrete structures can be accurately detected using the proposed system. Recently, Kim et. al proposed a method to identify concrete cracks using a combination of RGB-D and high-resolution digital cameras *(47)*. Research results showed that the proposed method can well measure crack width regardless of the angle of view.

To detect cracks in inaccessible areas, robotics and unmanned aerial vehicles (UAVs) were used. Xu et al. *(48)* proposed a robot system to inspect stay cables of cable-stayed bridges. Through laboratory and field testing, it was found that the robot can climb the inclined cables smoothly and stably. Ho et al. *(18)* used cameras mounted on cable climbing robotics, image processing and pattern recognition techniques to detect damage of bridge cables. It was found that the proposed method could be used to detect damage of bridge cables. Zhong et al. *(19)* used a UAV camera to detect cracks on concrete surfaces. An 8-rotor UAV was used to collect image data and a non-contact measurement instrument was used to capture the motion characteristics of the UAV in a hovering state without considering wind. The authors determined the minimum safety distance between the UAV and target building. However, the safe working distance in wind conditions and the allowable wind speed in normal operations were not included. Ellenberg et al. *(20)* used UVA camera images to quantify bridge related damaging including deflection, corrosion, and cracks. The results indicated that the developed post-processing algorithms were able to extract quantitative information from UAV captured imagery. To improve computer-vision crack detection drawbacks caused by motion blur or lack of pixel resolution, Bae et al. *(49)* developed a super-resolution crack network. The authors found that the proposed super-resolution crack network outperforms (24% better than) the method using raw images.

Although the combination of UAV cameras and vision-based technique can provide damage information via graphing of inaccessible areas and extensive structures, it is still limited and time consuming to process thousands of target images to extract accurate damage information. Images directly taken using UAV cameras need to be de-noised, standardized and reconstructed for extraction of damage information. To improve image processing efficiency, machine learning techniques have been used and shown effective. In recent years, as an emerging technique, deep learning, which refers to artificial neural networks with many hidden layers for enhanced performance, is spotlighted and shown promising for efficient image processing and damage identification.

Zhao et al. *(21)* developed a traffic surveillance system using deep learning and speeded-up robust features (SURF) to track vehicles and their movements. They used an aerial camera array mounted on an airplane to collect traffic data with a sampling frequency of 1 Hz and a coverage of 25 square miles. Machine learning methods were used to collect traffic data including the speed, density, and volume. It was found that deep learning with speeded up features can accurately (92%) estimate the speed, density, and volume. Pavement cracking detection is important for secure transportation, yet it is challenging because of the inhomogeneity of cracks and the complexity of the background. To address this issue, Zhang et al. *(22)* used convolutional neural network (CNN) deep learning method for road crack detection. Using smart phones, the authors collected 500 images with a size of 3264 by 2448 pixels. The research results indicated that the trained CNN model offered better crack detection results than methods using features extracted based on hand-craft methods. To overcome the challenges from real-engineering structures, e.g., lightening and shadow changes, Cha et al. *(23)* developed a CNN-based method for concrete crack detection. It was found that the proposed CNN using Canny and Sobel edge detection methods can find concrete cracks in real structures. Tong et al. *(24)* proposed a CNN-based method for crack length measurement. The authors used k-means clustering analysis to calculate the pre-extract cracks' properties which were used for training and testing. It was found that the accurate crack length recognition can be achieved. Till now, most existing literature on crack identification using vision- and machine learning-based techniques have been validated via laboratory testing. However, the images (intact and cracked) used in most existing literature don't include various challenging conditions that widely exist in real-life structures (e.g., human-made markings). In addition, there will be image distortion, lightening, edges, and shadow issues when using an UAV camera. Also, most computer vision-based methods focus on crack detection while references on crack quantification are limited. All these deficiencies in existing methodologies need to be addressed. In other words, there is a lack of a framework that can implement UAV image sensing and automatic image processing for accurate damage identification of both laboratory and real-engineering structures.

3.2. Research Motivation

As presented in the literature review, extensive research efforts have been exerted on developing computer vision- and deep learning-based methods to detect crack damage. However, the images (intact and cracked) used in most existing literature don't include various challenging conditions that widely exist in real-life structures (e.g., human-made or road boundary markings). In addition, different references used different CNN model architecture, e.g., VGG-16 *(50)*, AlexNet *(51)*, or GoogLeNet *(52)*, and a comparative study with respect to these architectures is lacked. Also, most computer vision-based methods focus on crack detection while references on crack quantification are limited. All these deficiencies in existing methodologies need to be addressed. In other words, there is a lack of a framework that can implement UAV image sensing and automatic image processing for accurate damage identification of both laboratory and real-engineering structures.

Therefore, it is desirable to develop a comprehensive computer vision- deep learning-based methodology with the required large datasets to perform bridge crack identification (detection and quantification). Furthermore, this developed methodology can provide cracking identification outputs that are compatible with current American Associate of State Highway and Transportation Officials (AASHTO) bridge inspection standard.

4. METHODOLOGY

Figure 1 illustrates the proposed procedure for crack damage detection. As shown in Figure 1, in the first step, a large volume of images with and without cracks will be collected from the Internet and using a UAV. Then these images will be corrected, segmented, and reconstructed to extract features that are closely related to cracks. For example, an image whose original resolution is 2048 x 2048 will be divided into 64 sub-images whose resolution is 256 x 256 using a sliding window. The objective of reconstructing the data structure is to facilitate the location of the crack region as well as to correlate the pixel values with the crack features. The generated image data will form a data library which will be divided into three datasets: one for training, one for testing, and one for validation. Next, a CNN classifier will be trained and validated. Finally, a new cracked image from a local bridge which is not included in the data library will be processed and identified by the trained CNN classifier to test the learned CNN model. Specific research activities to achieve the research goal are described in the following subsections.

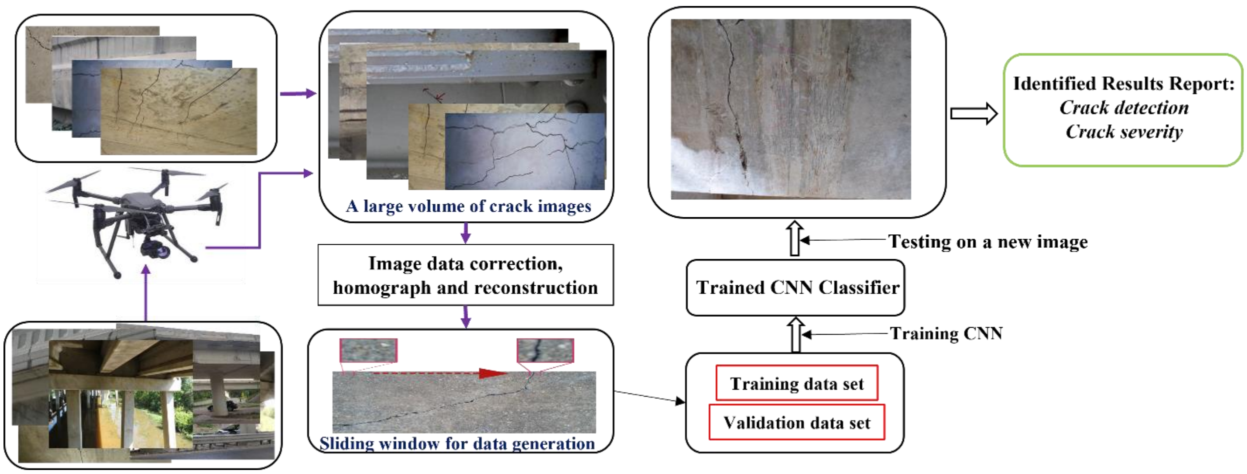

Figure 1. Procedure of CNN deep learning-based crack detection and quantification

4.1. Development of a Prototype Programmable Drone

4.1.1 Market evaluation

Drones are defined as an aircraft without human pilot on board. The flight of drones may operate under remote control by a human operator or fully autonomous. According to the rule of Federal Aviation Administration (FAA), a small drone that is less than 55 pounds can fly for work or business by following the Part 107 guidelines. Drone pilots operating under Part 107 may fly at night, over people, and moving vehicles without a waiver if they meet the requirements defined in the rules. Drones have been extensively studied in controls literature as well as common press. They are being used in mining, construction, aerial photography, search and rescue, movie industry, package delivery, mapping, surveying, farming, animal research, hurricane hunting, and defense.

Existing drones on the market can be classified into two categories: recreational drones and educational drones. We did a survey of existing drones in the market, and the pictures of drones are shown below.

Table 1. Illustration of current drones in the market

	
(a) DJI Inspire 1	(b) DJI Matrices 100
	
(c) Voyager 3	(d) DJI Phanton 3 Pro
	
(e) DJI Phanton 4	(f) Yuneec Typhoon H
	
(g) DJI S900 airframe	(h) Yuneec Typhoon 4k
	
(i) Blade Chroma	(j) Autel Robotics X-Star Premium

There are two reasons why we do not adopt the existing drone from the market. The first reason is due to the budget. Prices range from $1000-$3000. It is well beyond the budget we have in this project. The second reason is that the drones in the market are not fully controllable. The source code is not open, and we are not allowed to do some modification to suit our purposes for applications.

4.1.2. Overall Design

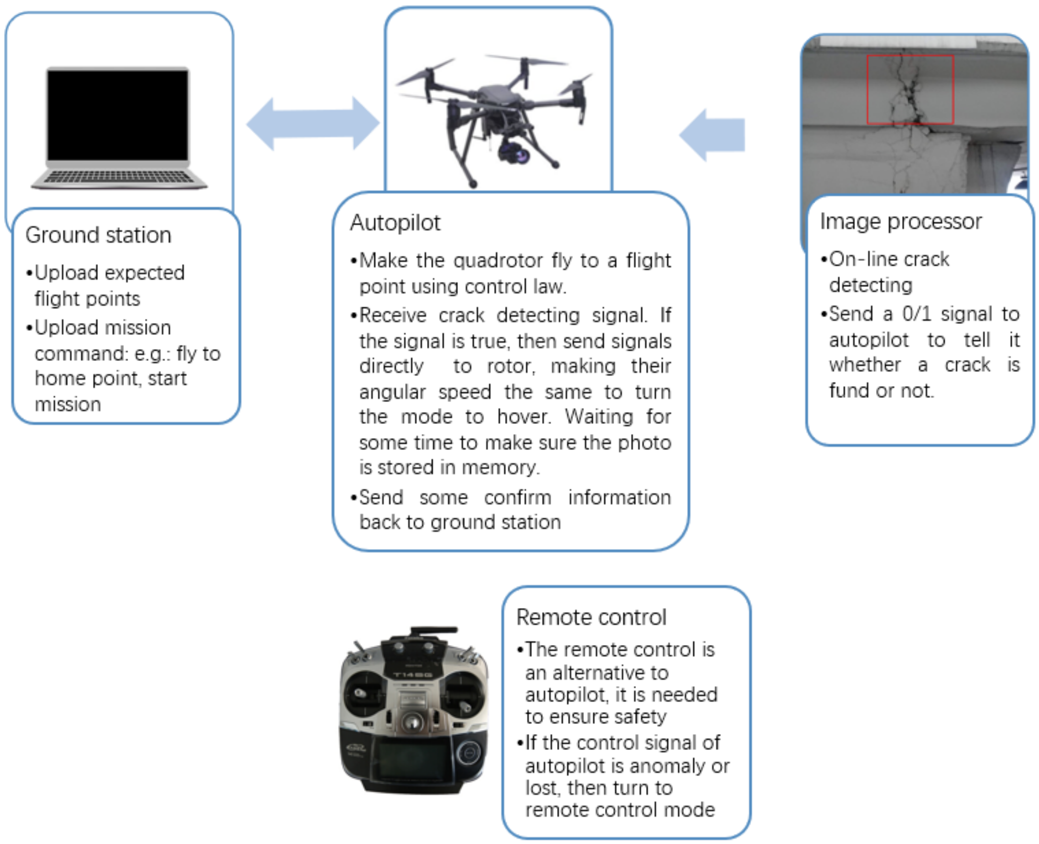

Figure 2. Key technology used in the developed drone

The overall design includes the communication between the drone and ground station (computer), and path planning using the GPS information to design waypoints. The drone first flies to the home point and wait for a command to start. Then the drone will take off from the starting point. Then a trajectory with several waypoints is transmitted from the ground station to the drone. The waypoints are GPS information. Then the drone will visit each way points. At the same time, pictures and videos will be recorded. After completing the visit of all way points, the drone will fly back to the home station. The pictures and videos will be stored the memory card in the drone.

The components are explained below:

- Ground Station: It is the home base. The ground station will update the expected flight points, and communicate with the drone, such as sending take-off and return commands.

- Drone: It is the one performs the inspection task and receive information from the ground station. Cameras will be mounted on the drone to take pictures and videos. The autopilot of the drone control will be discussed below. Due to the budget limitations, we are not able to perform real time crack detection. It can be conducted if additional fundings are available.
- Remote control: It is another way to control the drone. The drone will fly autonomously. The remote control is used to ensure safety. In case, the communication between the ground station and the drone is lost, or some anomaly happens on the drone. The remote control can take over and control the drome fly back to the home station.

4.1.3. Drone Autopilot

The quadrotors use four motors acting as direct power sources for flight and are controlled by varying the lift generated by the four rotors. What we can control in the quadrotors are angular speeds of four rotors. Each rotor has an angular speed and produces a vertical force and a moment to control the position and attitude of quadrotors. Figure 3 shows the controlling principle.

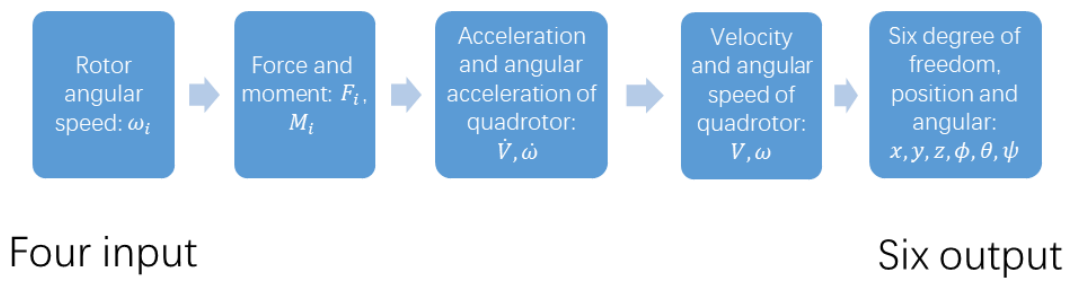

Figure 3. Controlling principle of the drone

4.1.4. Prototype Quadrotor Drone

Figure 4 illustrates the developed prototype of the programmable drone. The key parameter values of the drone are listed in Table 2. Before flying the drone, we need to set the airframe, calibrate compass, gyroscope, and the accelerometer. Figure 5 demonstrates the compass calibration process. During this calibration, the drone is placed in any of the orientations shown in red (incomplete) and held still. Once prompted (the orientation-image turns yellow) rotate the drone around the specified axis in both directions. Once the calibration is complete for the current orientation, the associated image in the corresponding box will turn green. Repeat the calibration process for all orientations. Upon completion of calibrations, the drone is ready to set off.

Figure 4. Developed prototype of the programmable drone

The research team has tested the performance of the developed drone prototype. It can successfully take off and fly along the pre-defined trajectory. Preliminary datasets were gathered using a cell phone camera installed on the drone. Relevant videos can be found via the uploaded YouTube video: https://youtu.be/g_sLGD6dG-Q (positioning model) and https://youtu.be/HEmbD50Kw7I (takeoff mode).

It is noted that this part is an initial exploration of a programmable drone that will be fully used in future research projects and application. In this project, most datasets were gathered using a commercial drone DJ phantom 4, which is described in the following section.

Table 2. Developed prototype drone system performance parameter values

Parameter	Value	Parameter	Value
Takeoff Weight	1100 g	Max flight speed	5 m/s
Dimensions	500×500×225 mm^3	Max tilt angle	35°
Max ascent speed	3 m/s	Max service height	1000 m
Max descent speed	3 m/s	Control Range	3400 m
Max descent speed	12 minutes	Maximum storage	32GB

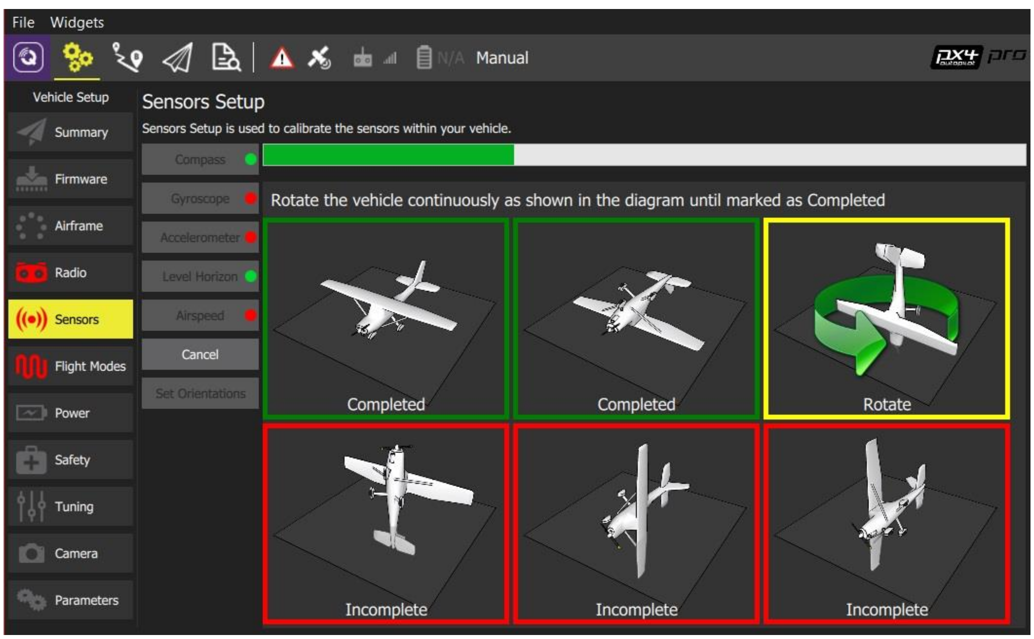

Figure 5. Calibration of the drone

4.2. Data Collection and Processing

4.2.1. Image and video data collection

As mentioned in preceding sections, a large number of datasets covering comprehensive cracking types are essential to achieve desired detection and quantification results using deep learning methods. In this project, the required datasets were mainly collected using a DJ Phantom 4 drone hovering around target bridges, concrete road surface, and asphalt pavements. In addition, images taken during inspection of bridges (Fig. 6 (b) and (c)) using hand-carrying cameras were included in the datasets. Figure 6 shows the three bridges in Louisiana that were used to take images and videos. Figure 6(a) is a I-10 highway bridge across the City Park Lake. It is a steel girder bridge with concrete bent and columns. Figure 6 (b) is a prestressed concrete continuous bridge overpassing a railroad. The main span length is 85 ft. Fig. 6 (c) has a steel main span and the approach span is prestressed concrete continuous bridge. In addition, images and videos of asphalt pavements were collected using the drone hovering over the Touchdown village parking lot in LSU. Through data collection, around 300 raw concrete images (with and without cracks) and 100 raw asphalt images (with and without cracks) were gathered. These images cover representative cracking types, e.g., single crack, multiple cracks, thin or wide cracks, and cracks with human-markers, background noises, or road boundary markers. Figure 6 demonstrates a group of cracked images with representative cracking types.

I-10 Highway bridge across the City Park Lake

Structure Number: 084000153001442 Built in 1997

Structure Number: 084004550530131 Built in 1994

Figure 6. Local bridges used to collect data using drones.

4.2.2. Calibration and homography

Raw images directly taken from the camera are always distorted due to the wide angles of the camera lens. Therefore, distortion calibration will be implemented in the first step to get precise crack assessment. In this research task, the camera calibration algorithm developed by Zhang et al. (53) will be used. The basic idea is described here. Let $m = [u, v]^T$ denote a 2D point and $M = [X, Y, Z]^T$ denote a 3D point. Then define $\widetilde{m} = [u, v, 1]^T$ and $\widetilde{M} = [X, Y, Z, 1]^T$. The relationship between a 3D point M and its 2D projection is:

$$s\widetilde{m} = A[R, t]\widetilde{M},\ A = \begin{matrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{matrix} \qquad [1]$$

where:

s = an arbitrary scale factor;

A = the camera intrinsic matrix;

$[R, t]$ = the rotation and translation parameters that relate the world coordinate system to the camera coordinate system;

α, β, γ are the scale and skew factors; and

u_0, v_0 are the coordinates of the principal point.

Without loss of generality, the model plane is assumed to be on $Z = 0$ of the world coordinate system. Equation 1 can be written as:

$$s\begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = A[r_1, r_2, r_3, t]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = H\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \text{with } H = A[r_1, r_2, t] \qquad [2]$$

where:

$H_{3\times 3}$ is the homography to be determined.

For each image, we can have a linear transformation as shown in Equation 2. With the coordinates of n ($n \geq 3$) images. The homography can be determined via solving the least squares problem. Then the calibration and homography is completed for the selected camera.

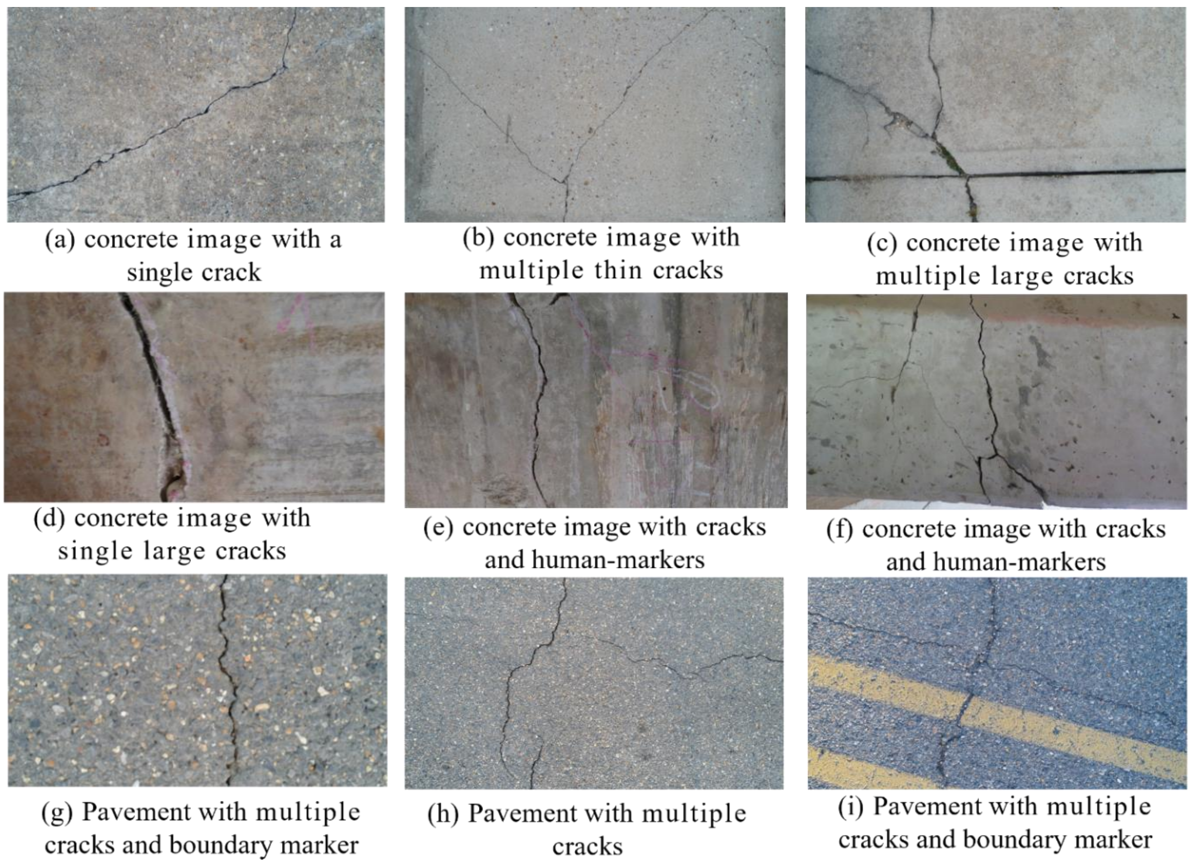

Figure 7. Illustration of collected images with representative cracks and background noises

4.2.3. Crack feature extraction

The collected raw images are RGB (red, green and blue) images in JPG/PNG format, including color information, which increases the difficulty of feature detection and recognition of crack characteristics. To facilitate crack feature extraction, the color information will be changed and images will be converted to grey-scale binary figures in BMP format. Figure 8 shows the procedure converting the RGB crack images to gray-scale binary images. In Figure 8, the original image is not segmented. In this research, the raw image will be segmented into a number of unit sub-figures for the deep learning process. It is noted that the shadows in Figure 8(b) produces fake cracks in the binary image. This issue will be addressed in this research task by finely adjusting the parameters during image conversion.

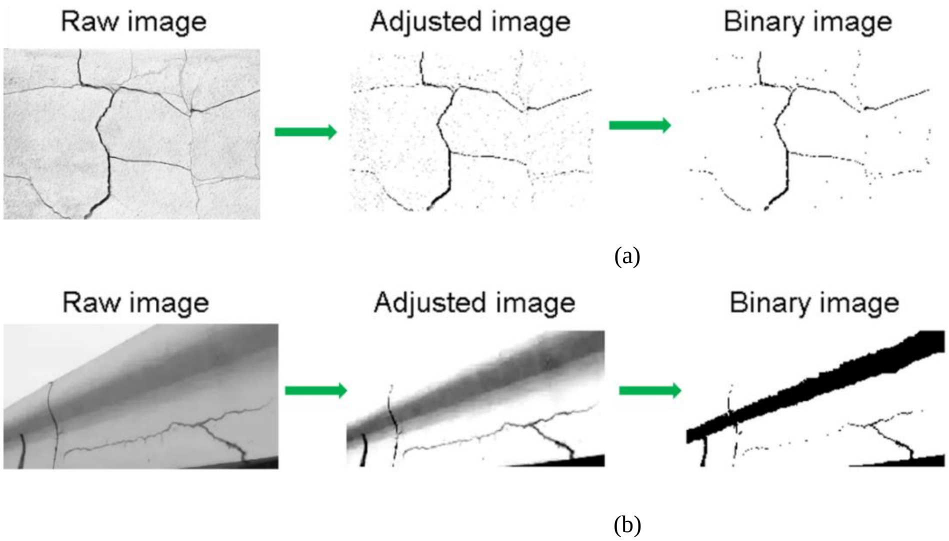

Figure 8. Illustration of crack feature extraction

4.3. Deep Learning-based Crack Damage Identification

Based on the collected datasets, this section uses deep learning models to identify crack images. As mentioned in the literature review, there are two basic methods to detect cracks using images: patch-based and pixel-based. In this project, the patch-based crack detection method is used.

4.3.1. Data preparation: Patch-based crack detection

Patch-based crack detection techniques involve separating the image into sub-images, predicting the presence of cracks on each sub-image and then drawing inference about the crack damage on the full picture. Sub-images were obtained from the images by sliding a window across the images, as shown in Figure 9. In this project, the used slide windows have a dimension of 256 to 380 pixel. The size of the sliding window used is based on the resolution of the images and the distance of the camera from the bridge surface. By using the 256-to-380-pixel window for image pre-processing, care is taken to exclude the images with scratches and surface irregularities, so they won't be identified as cracks by the model. The resulting sub-images were then resized to 256 by 256 pixels to build a database for training and validation.

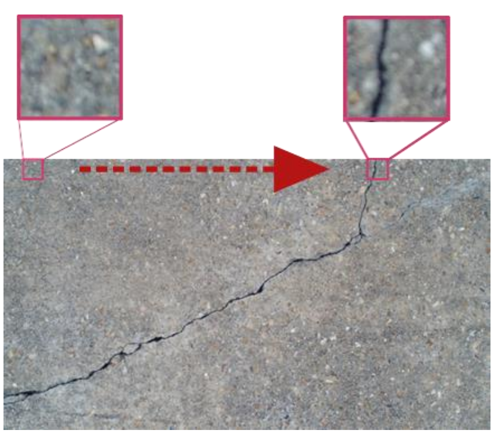

Figure 9. Sliding window for data preparation

4.3.2. Data benchmarking, filtering, and augmentation

- **Data benchmarking and filtering**

The data obtained was screened and filtered. Images that are not clearly cracked, or free from cracks as visually observed by a human, are discarded. As shown in Fig. 10, the filtered images include those with low resolution as a result of poor camera focus when the drone is in motion, images where the surface cracks are visible, and images containing background features and images with ambiguous crack marks. A benchmark for pre-classifying each of the image into a class is then determined for consistency. To build a database for the different classes of the data, the sub-images were scanned for cracks. If there was a conspicuous crack, either narrow, moderate, or severe, the images are classified as 'cracked'. If there were no cracks at all, it was classified as 'non-cracked'.

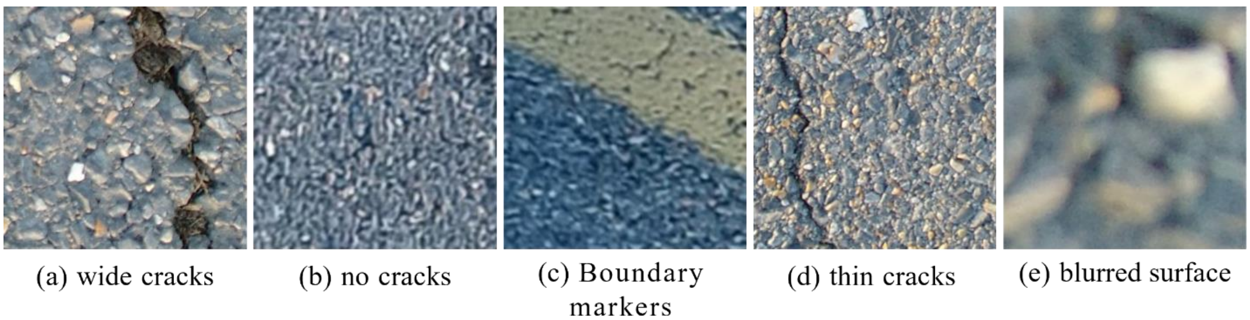

Figure 10. Data benchmarking and filtering with respect to asphalt pavement images

- **Data augmentation**

This data pre-processing procedure involves augmenting the existing dataset with perturbed versions of the existing images. Via flipping and rotating the original image dataset in the 'cracked' classes for both concrete and asphalt images, we can obtain around 10,000 sub-images while the non-cracked concrete and asphalt dataset contains around 20,000 sub-images each. It is noted that application of data augmentation helps to expose the neural network to a wide variety of variations and make it less likely that the neural network recognizes unwanted characteristics in the dataset.

5. ANALYSIS AND FINDINGS

5.1. Results of Hyper-Parameter Optimization

Performance of deep learning models are highly sensitive to its parameters. Hence, optimization of the hyper-parameters for a learning algorithm is of critical importance. A hyper-parameter is a parameter whose value is used to control the learning process. For different datasets, the hyper-parameter needs to be tuned so that the model can optimally learn and properly generalize on the dataset. The optimization of hyper-parameter yields an optimal model which minimizes the loss function on the given data. For the hyper-parameter search, as shown in Table 3, there are different approaches to hyper-parameter optimization. The first is the grid search which selects all the possible combinations of the hyper-parameter configurations in the entire search space. The searching trains the network using each configuration and returns the optimal configuration where the loss is minimal. This approach requires a high computational cost.

In comparison, other approaches can mitigate the computational cost by bypassing the exhaustive search space. For instance, the random search approach randomly selects a combination of hyper-parameters from a discrete or continuous set of values and returns the performance after training. However, there is no guarantee of finding the optimum point. There are many other options to tune hyper-parameters including the Bayesian optimization approach and the reinforcement learning approach *(54,55)*. The early stopping-based search focuses on the promising hyperparameters and, based on some statistical tests, disregards the ones that perform poorly. In this project, the asynchronous successive halving (ASHA) method is used.

Table 3. Approaches for hyper-parameter optimization

Hyper-parameter optimization methods	Description
Random search	Random parameter search and training using all the possible parameters in the set
Bayesian optimization	Probabilistic model determination to map hyperparameters to the objective function and locate the optimum *(56, 57)*
Grid search	Exhaustive parameter search and training using all the possible hyperparameters in a discrete set
Gradient-based optimization	Gradient computation, with respect to hyperparameters and subsequent hyperparameters optimization using gradient descent
Evolutionary optimization	Evolutionary algorithm-based optimization to search the space of hyperparameters
Early stopping-based search	Considers the promising hyper-parameters, and disregards the ones that perform poorly based on statistical tests, successive halving (SHA) Asynchronous successive halving (ASHA) *(58)*
Population-based optimization	Starts training many neural networks in parallel with random hyperparameters and uses information from the population to refine the hyper-parameters *(59)*

Table 4 lists the deep learning model architecture parameters tested using the dataset, and the hyper-parameter search space used in this project. The optimizers considered include the stochastic gradient descent optimizer, SGD, which implements the stochastic gradient descent algorithm, optionally with momentum. The parameters for the optimizer are momentum factor (mm), weight decay (wd), learning rate (lr), and the dampening (dp). The learning rate determines how much the model weights are changed in response to the estimated error each time the model weights are updated. The momentum factor helps accelerate gradients in the right directions, leading to faster convergence. Dampening helps to reduce the step size for higher gradients. The weight decay helps in regularization by adding a small penalty, to the loss function, to keep the weights small, and prevent over fitting.

Table 4. Search space for hyperparameters

Model parameter and hyper-parameter search space	
Model	**ResNet**, **GoogLeNet**, **AlexNet**
Hyper-parameters	
learning rate (lr)	log uniformly between [1e-4 1e-1]
weight decay (wd)	{8e-6,1e-5,3e-5}
eps	1e-8
rho	0.9
betas	0.9, 0.999
batch size	{32,48, 96, 128}
momentum (mm)	{0.6,0.9,1.2}
Dampening (dp)	{0,0.9,0.995}
Model parameters	
Optimizer	Adam(lr, betas, eps), Adadelta(lr, rho, eps) SGD(lr, mm, wd, dp)
Activation function	Rectified Linear Unit (ReLU), leaky ReLU, tanh, Scaled exponential Linear Unit (SELU) & identity

The Adam optimizer implements the Adam algorithm *(60, 61)*. In addition to learning rate and weight decay, the Adam optimizer parameters include betas, coefficients used for computing running averages of gradient and its square, and epsilon, a term added to the denominator to improve numerical stability (eps). The Adadelta optimizer implements Adadelta algorithm *(62)*. In addition to the learning rate, epsilon and weight decay, parameter rho is used for computing a running average of squared gradients.

In this project, the Inception Net, the ResNet, and the AlexNet models are the main deep learning architectures applied to the dataset. Each of these network models are made configurable by either editing the pytorch-based source code or a similar code snippet, and representing it using the pytorch lightning module, whose advantage over pytorch is that it provides a structure for the research. These architectures were chosen because they have unique configurations and performed well on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC15) *(63)*. The Asynchronous Successive Halving Algorithm (ASHA) is used. The hyper-parameter tuning was carried out using pytorch, ray tune, and pytorch lightning modules in Python. Figures 11-13 demonstrate samples of the Python codes developed in this research.

```
from ray.tune.integration.pytorch_lightning import TuneReportCallback, TuneReportCheckpointCallback
import torch, os, sys
import tempfile
from ray import tune
import pytorch_lightning as pl
sys.path.append("..")
from model import residual_net, inception_net, alex_net
from data import datamodule
from pytorch_lightning.plugins import DDPPlugin
from pytorch_lightning.loggers import TensorBoardLogger
```

Figure 11. Code snippet showing modules imported for the optimization

```
def train_fn(config, model_arch, data_dir=os.path.join(os.getcwd(), "Dataset") , num_epochs=60, num_gpus=0, checkpoint_dir=None):
   dm = datamodule.ImgData(num_workers=8, batch_size=config["batch_size"], data_dir=data_dir)
   model = model_name(model_arch)(config,  dm.num_classes, data_dir)
   metrics = {"loss": "val_loss", "acc": "val_accuracy"}
   pl.seed_everything(42, workers=True)
   trainer = pl.Trainer(
      max_epochs=num_epochs,
      gpus=num_gpus,
      logger = TensorBoardLogger(save_dir=tune.get_trial_dir(), name="my_model"),
      #log_every_n_steps=5000,
      progress_bar_refresh_rate=0,
      accelerator='ddp',
      plugins=DDPPlugin(find_unused_parameters=False),
      deterministic=True,
      callbacks=[TuneReportCallback(metrics, on="validation_end")])

   trainer.fit(model, dm)
```

Figure 12. Code snippet showing training function

```
class ImgData(pl.LightningDataModule):
        """ This class handles the Training and Validation dataset for data saved in a folder in the current working directory.
        The Dataset folder is expected to contain images classified subfolders
        The number of subfolders indicates the number of classes while the name of the subfolders are the classnames """
        def __init__(self, num_workers, batch_size, data_dir: str = os.path.join(os.getcwd(), "Dataset")):
                super().__init__()
                self.data_dir = data_dir
                self.num_workers=num_workers
                self.batch_size=batch_size
                self.transform = {
                        'train': transforms.Compose
                                ([
                                 transforms.Resize(size=256),
                                 transforms.ColorJitter(brightness=0.1, contrast=0.1),
                                 #transforms.Normalize(self.mean_nums, self.std_nums),  |
                                 transforms.ToTensor()
                                ]),
                        'val': transforms.Compose
                                ([
                                 transforms.Resize(256),
                                 transforms.ToTensor()
                                ]),
                }
                _, self.classes = self.dataset()
                self.num_classes=len(self.classes)
                self.dims = (3, 256, 256)
```

Figure 13. Code snippet showing data loader transformation

Table 5. Results of hyper-parameter search

Model parameter and hyper-parameter search space			
Model	**Residual network architecture**	**Inception network architecture**	**Alex net**
Model parameters			
Network size/depth	256 × 2 + 512 × 3 ResNet blocks	4 inception blocks	64
Batch size	64	4	relu
Activation function	Relu	leaky relu	
Tuned hyper-parameters			
Optimizer	Ada Delta	Adam	Ada Delta
learning rate	0.0152923	0.000216735	0.0163072
dampening	N/A	0	N/A
betas	N/A	[0.9, 0.999]	N/A
momentum	N/A	N/A	N/A
eps	2.33231e-08	1.7739e-07	2.04677e-06
rho	0.9	N/A	0.9
weight decay	N/A	N/A	N/A
Performance metrics			
Validation Accuracy	99.83%	96.38 %	99.0302 %
Validation loss	0.00652511	0.0978632	0.0378648
at epoch	35	35	35

Figure 11 shows the libraries that are imported to implement the hyper-parameters search. Pytorchlighning, which is a module that runs on pytorch is imported and used for model and dataset handling while Ray tune is imported for ASHA hyper-parameter tuning. TuneReportCallback is imported to pass the metrics, including the training accuracy and validation accuracy from the pytorch lightning-based training to ray tune. TensorBoardlogger is used for logging the experiment progress. As shown in the code snippet in Figure 12, each training process receives a set of hyper-parameter configuration ‘config’ which is then passed into the corresponding deep-learning model modules to create a model and train it with this configuration.

Also, the dataset is handled using the pytorch lightning module. The dataset is made configurable to handle different batch sizes, since the batch size is a parameter in the search-space. Random color jitters are applied to the images in the dataset at every epoch, to improve generalization, as shown in the code snippet in Figure 13. The results are outlined in Table 5.

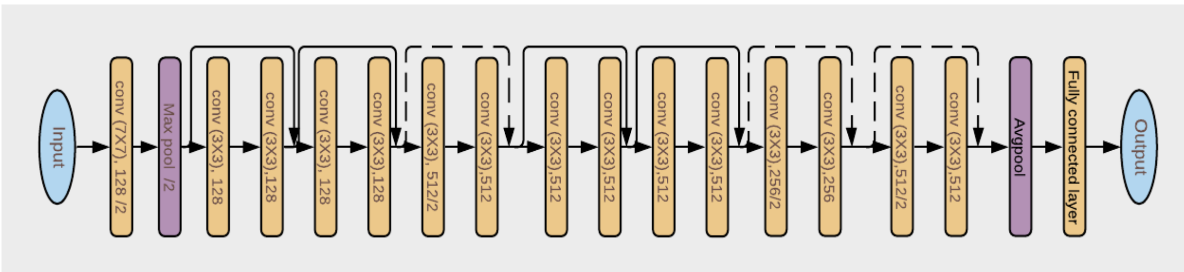

Figure 14. Final residual network structure: dashed lines indicate the change in dimension of input volume

5.2. Results and Performance of the Optimized Deep Learning Models

Via optimization, the optimum ResNet architecture for the dataset is a five-layer network where the first two layers have a size of 256, and the last three of 512. The optimum model with the Inception Network architecture has three layers with a maximum pooling layer. The Alex Network still retains its architecture, and a set of optimum parameters are obtained. The ultimate architecture of the ResNet and the Inception Network are illustrated in Figures 14 and 15, respectively.

To evaluate the performance of the ResNet, AlexNet, and Inception Net models, the validation accuracy of the three optimized models are compared in Figure 16. As observed in Figure 8, the ResNet architecture outperforms the other two models in validation accuracy. The validation accuracy of the ResNet model is generally higher than those of the AlexNet and Inception network at every training epoch, except for some slight dips at some epochs. These dips are related to the ResNet model architecture’s generalization errors that may occur after some training iteration but tend to become smaller as the training progresses. It gives a picture of the model’s approach to the optimum.

Another metric compared is the total number of weighting parameters in the three models. The number of parameters is crucial because it gives an indication of the memory space required to make an inference with the model. The model’s weight would be stored and used for computation and inferences in newer images. More importantly, the number of weight parameters gives a good indication of the computational costs of the model in making inferences. The prediction algorithm divides each new image into several sub-images. A higher number of weights means a higher number of computations for each given image, and a higher computational cost for making inferences for several sub-images in series. The Inception Network has the smallest size of weighting parameters. The residual network has about 260 times more parameters, while the AlexNet has about 800 times more parameters than the Inception Net. A higher validation accuracy will yield a higher generalization, while lower number of parameters will conserve more memory, and computational power. For the project’s optimal crack prediction, the ResNet architecture is chosen.

With the optimized ResNet model, the testing dataset was input into the model for identification. In the first trial, the concrete and asphalt images were mixed in a single dataset for training and validation. Results indicated that the identification accuracy is not high (< 0.7) because the concrete and asphalt images have different features. A mixed dataset will lower the parameter optimization efficiency and accuracy. Hence, two separate datasets: one for concrete and another for asphalt were created to train two separate ResNet learning models, one for concrete and the other for asphalt. The identification results are shown in Figure 18 where representative types of cracks, e.g., thin, medium, and large cracks are covered. In Figure 18, we can find that the optimized ResNet model can well identify different types of cracks in the concrete and asphalt images. Figure 19 illustrates an asphalt pavement image with cracks and yellow boundary markers. We can find that through labeling the cracks and markers in the training dataset, the real cracks can be correctly detected without false identification of the yellow markers as cracks.

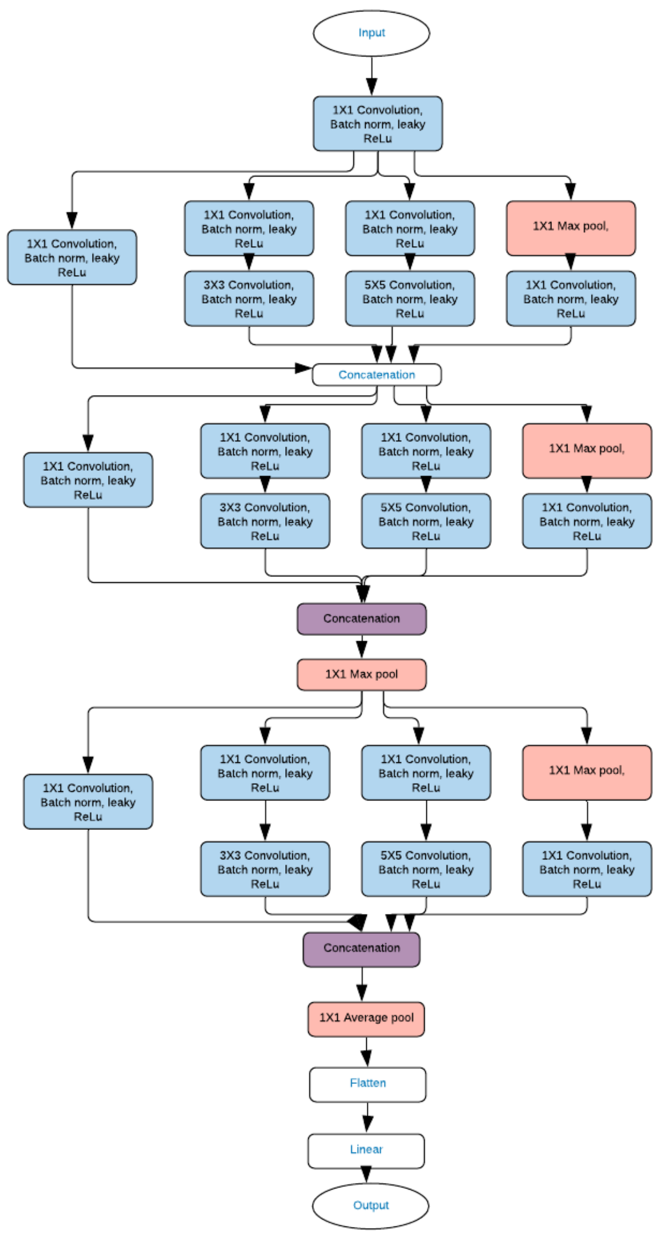

Figure 15. Final inception network structure

Figure 16. Comparison of ResNet, Inception Net, and AlexNet models

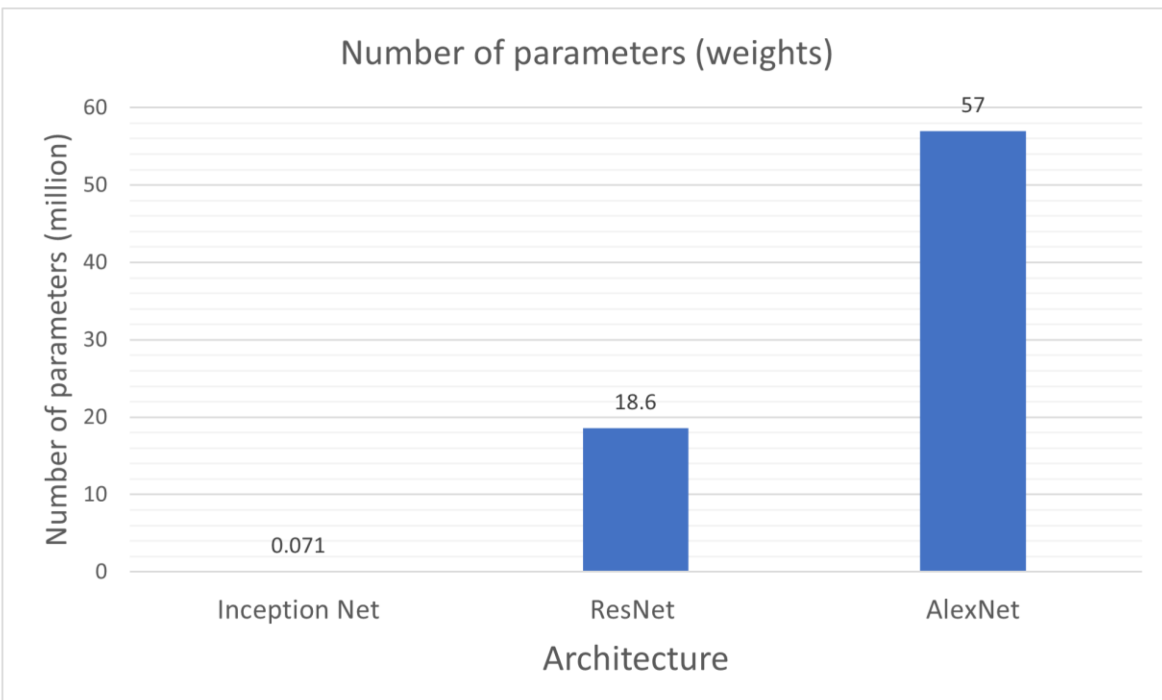

Figure 17. Comparison of number of weighting parameters in the Inception, ResNet, and AlexNet models

(a) Identified cracks in concrete images

(b) Identified cracks in asphalt pavement images

Figure 18. Crack identification using the optimized ResNet model. (a): concrete images with representative types of cracks; (b) asphalt images with representative types of cracks

Figure 19. Detection of asphalt pavement crack without identifying boundary markers as false cracks

6. CONCLUSIONS

This research project develops an automatic crack identification methodology using deep learning techniques and UAV images. The research goal is to offer an efficient, cost-effective inspection method for aging bridges and other transportation infrastructure. The research work has been focused on dataset collection, image processing, and optimization of deep learning models. Both local bridges and asphalt pavements were used to collect large amounts of images and videos. Python-based codes have been developed to implement training, testing, and validation of the deep learning models. Based on the field data and numerical modelling results, the following major conclusions can be drawn:

1) The developed prototype of a programmable drone can be well controlled using the PX4 autopilot platform. Through testing, the developed drone can take off and land autonomously. It can follow predesigned trajectories by inputting the GPS location information. This programmable drone can carry other cameras, e.g., multispectral cameras, LiDARs, and other advanced sensing tools to implement inspection of bridges and other transportation infrastructure

2) A large volume of images and videos (with and without cracks) has been collected from local bridges, buildings, and asphalt pavements using drones. The obtained dataset covers images from different bridge components (decks, girders, and piers) with four representative cracking severities: intact, minor cracking, moderate cracking, and severe cracking. It is noted that images with background noises, such as road boundary markers were included in the dataset for identification. The collected comprehensive dataset of images and videos can be used to train machine learning models to identify cracking damage of a large set of concrete bridges and pavements.

3) A deep convolutional neural network-based computer vision methodology for efficiently identifying cracks in bridges and pavements has been developed. Three CNN model architectures, ResNet, Inception Net, and AlexNet were selected, and the corresponding parameters were optimized. Python based codes were developed for training, testing, and validation of the CNN models to efficiently identify cracks. Through comparison, it is found that the ResNet provides the best identification accuracy and requires an acceptable computational cost. Hence, the ResNet was selected for crack identification in this project.

4) When the collected concrete and asphalt images (with and without cracks) were mixed in one big dataset for training and validation, the inference accuracy rate is unsatisfactory (< 0.7) due to different features of concrete and asphalt images. When two separate datasets, one for concrete and one for asphalt, were used for training and validation, the accuracy rate is high (> 0.96), signaling that crack identification of concrete structures and asphalt pavements needs to be implemented using separate machine learning models.

5) The optimized ResNet deep learning model can well identify representative types of cracks (thin, thick, and multiple cracks) on the concrete and asphalt images. Also, the yellow boundary markers on the asphalt image can be excluded from the identified cracks, indicating that the developed method can get rid of the influence of background noises for crack detection.

REFERENCES

1. I. Goodfellow, J. Pouget Abadie, M. Mirza, et al. Generative Adversarial Nets. Advances in Neural Information Processing Systems (NIPS2014), 2014.

2. A. Krizhevsky, I. Sutskever, G. Hinton. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems (NIPS 2012), 2012.

3. R. Adams, L. Bischof, Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell*., 1994. 16(6): 641-647.

4. M. R. Jahanshahi, J. S. Kelly, S. F. Masri, G. S. Sukhatme. A survey and evaluation of promising approaches for automatic image-based defect detection of bridge structures. *Struct. Infrastruct. Eng*., 2009. 5(6): 455-486.

5. Jahanshahi M, Masri S, Padgett C, et al. An innovative methodology for detection and quantification of cracks through incorporation of depth perception. *Mach Vision Appl*, 2013. 24: 227-241.

6. Koch C, Paal S, Rashidi A, et al. Achievements and challenges in machine vision-based inspection of large concrete structures. *Adv Struct Eng* 2014. 17: 303-318.

7. Choi H.-S., Cheung, J.-H.; Kim, S.-H.; Ahn, J.-H. Structural dynamic displacement vision system using digital image processing. *NDT E Int*. 2011. 44: 597-608.

8. Park, S.W.; Park, H.S.; Kim, J.H.; Adeli, H. 3D displacement measurement model for health monitoring of structures using a motion capture system. *Measurement* 2015. 59: 352–362.

9. Feng D., Feng M.Q. Vision-based multipoint displacement measurement for structural health monitoring. *Struct. Control Heal. Monit*. 2016. 23: 876-890.

10. German, S.; Brilakis, I.; DesRoches, R. Rapid entropy-based detection and properties measurement of concrete spalling with machine vision for post-earthquake safety assessments. *Adv. Eng. Inform*. 2012. 26: 846-858.

11. Dawood, T.; Zhu, Z.; Zayed, T. Machine vision-based model for spalling detection and quantification in subway networks. *Autom. Constr*. 2017. 81:149:160.

12. Jahanshahi, M.R.; Masri, S.F. Parametric Performance Evaluation of Wavelet-Based Corrosion Detection Algorithms for Condition Assessment of Civil Infrastructure Systems. *J. Comput. Civ. Eng*. 2013. 27: 345-357.

13. Kim H, Ahn E, Cho S, et al. Comparative analysis of image binarization methods for crack identification in concrete structures. *Cement Concrete Res* 2017. 99: 53-61.

14. Lecompte, D.; Vantomme, J.; Sol, H. Crack Detection in a Concrete Beam using Two Different Camera Techniques. *Struct. Heal. Monit. Int. J*. 2006. 5: 59-68.

15. Yamaguchi, T.; Shingo, N.; Ryo, S.; Shuji, H. Image-Based Crack Detection for Real Concrete Surfaces. *IEEJ Trans. Electr. Electron. Eng.* 2008. 3: 128-135.

16. Abdel-Qader, I., Abudayyeh, O., Kelly, M.E. Analysis of Edge-Detection Techniques for Crack Identification in Bridges. *J. Comput. Civ. Eng*. 2003. 17:255-263.

17. Prasanna, P., Dana, K.J., Gucunski, N., Basily, B.B., La, H.M., Lim, R.S., Parvardeh, H. Automated Crack Detection on Concrete Bridges. *IEEE Trans. Autom. Sci. Eng.* 2016. 13: 591-599.
18. Ho H. N., Kim K. D., Park Y. S., Lee J. J., An efficient image-based damage detection for cable surface in cable-stayed bridges, *NDT & E International*, 2013. 58: 18-23.
19. Zhong X, Peng X, Yan S, et al. Assessment of the feasibility of detecting concrete cracks in images acquired by unmanned aerial vehicles. *Automat Constr* 2018. 89: 49-57.
20. Ellenberg A, Kontsos A, Moon F, et al. Bridge related damage quantification using unmanned aerial vehicle imagery. *Struct Control Health Monit*, 2016. 23: 1168-1179.
21. Zhao, X.; Dawson, D.; Sarasua, W.A.; Birchfield, S.T. Automated Traffic Surveillance System with Aerial Camera Arrays Imagery: Macroscopic Data Collection with Vehicle Tracking. *J. Comput. Civ. Eng*. 2017.31: 04016072.
22. Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016.
23. Cha, Y.-J.; Choi,W.; Büyüköztürk, O. Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. *Comput. Civ. Infrastruct. Eng.* 2017. 32: 361-378.
24. Tong, Z.; Gao, J.; Han, Z.; Wang, Z. Recognition of asphalt pavement crack length using deep convolutional neural networks. *Road Mater. Pavement Des*. 2018.19: 1334-1349.
25. https://infrastructurereportcard.org/cat-item/bridges/
26. C. Farrar, K. Worden. *Structural health monitoring: A machine learning perspective*, John Wiley & Sons Inc., 2013.
27. A. Teughels, G. De Roeck. Structural damage identification of the highway bridge Z24 by FE model updating. *Journal of Sound and Vibration*, 2004. 278(3): 589- 610.
28. B. Moaveni, Joel P. Conte, Uncertainty and Sensitivity Analysis of Damage Identification Results Obtained Using Finite Element Model Updating, *Computer-Aided Civil and Infrastructure Engineering*, 2009. 24: 320-334.
29. M. Friswell. Damage identification using inverse methods, In: Morassi A, Vestroni F, eds. Dynamic Methods for Damage Detection (CISM Courses and Lectures vol 499). New York, Springer Wien, 2008.
30. K. Worden, M. Friswell. Modal vibration-based damage identification, In: Boller C, Chang F-K, Fujino Y, eds. Encyclopedia of Structural Health Monitoring: John Wiley and Sons, 2009.
31. Ying Y, Garrett JH Jr, Oppenheim IJ, et al. Toward data-driven structural health monitoring: application of machine learning and signal processing to damage detection. *J Comput Civil Eng* 2012. 27(6): 667-680.
32. Worden K and Manson G. The application of machine learning to structural health monitoring. *Philos Trans R Soc* 2007. 365(1851): 515-537.

33. Z. Zhang, C. Sun. Structural Damage Localization via Physics-Guided Machine Learning: A Methodology Integrating Pattern Recognition and Finite Element Model Updating, *Structural Health Monitoring*, DOI: 10.1177/1475921720927488.

34. Z. Zhang, C. Sun. Multi-site Structural Damage Identification Using a Machine Learning Method of Multi-label Classification. *Measurement* 2020. 154: 107473.

35. Z. Zhang, C. Sun. A Numerical Study of Multi-Site Damage Identification: A Data-Driven Method via Constrained Independent Component Analysis, *Structural Control and Health Monitoring* DOI:10.1002/stc.2583.

36. Z. Zhang, C. Sun, C. Li, M. Sun. Vibration based bridge scour evaluation: A data-driven method using support vector machines. *Structural Monitoring and Maintenance*, 2019. 6(2): 125-145.

37. Z. Zhang, C Sun. Transfer-Learning Guided Bayesian Model Updating for Structural Damage Detection Accounting for Modeling Uncertainty, *Mechanical System and Signal Processing*, 2022. 166: 108426.

38. Z. Zhang, C. Sun, R. Bridgelall, M. Sun. Road profile reconstruction and evaluation using connected vehicle responses and wavelet analysis. *Journal of Terramechanics*, 2018. 80： 21-30.

39. Z. Zhang, C. Sun, M. Sun, R. Bridgelall. Application of a Machine Learning Method to Evaluate Road Roughness from Connected Vehicles. *Journal of Transportation Engineering, Part B: Pavements*, 2018. 144(4): 04018043.

40. Feng D and Feng MQ. Computer vision for SHM of civil infrastructure: from dynamic response measurement to damage detection—a review. *Eng Struct* 2018. 156:105-117.

41. Xu Y and Brownjohn JMW. Review of machine-vision based methodologies for displacement measurement in civil structures. *J Civ Struct Heal Monit* 2018.8: 91-110.

42. Spencer BF, Hoskere V and Narazaki Y. Advances in computer vision-based civil infrastructure inspection and monitoring. *Engineering* 2019. 5: 199-222.

43. C. Dong, F. Catbas, A review of computer vision–based structural health monitoring at local and global levels, *Structural Health Monitoring*, 2020. 1-52.

44. Abdel-Qader I, Pashaie-Rad S, Abudayyeh O, et al. PCA-based algorithm for unsupervised bridge crack detection. *Adv Eng Softw* 2006. 37: 771-778.

45. Li G, He S, Ju Y, et al. Long-distance precision inspection method for bridge cracks with image processing. *Autom Constr* 2014. 41: 83-95.

46. Yu S., Jang J. and Han C. Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel. *Autom Constr* 2007. 16: 255-261.

47. Kim H., Lee S., Ahn E., Shin M., Sim S. Crack identification method for concrete structures considering angle of view using RGB-D camera-based sensor fusion, *Structural Health Monitoring*, 2021. 20(2): 500-512.

48. Xu F., Wang X., Wang L. Cable inspection robot for cable-stayed bridges: Design, analysis, and application, *Journal of Field Robotics*, 2011. 28(3): 441-459.

49. Bae H., Jang K., An Yun Kyu, Deep super resolution crack network (SrcNet) for improving computer vision–based automated crack detectability in in situ bridges, *Structural Health Monitoring*, 2020. 20(4): 1428-1442.

50. da Silva WRL and de Lucena DS. Concrete cracks detection based on deep learning image classification. Proceedings, 2018.

51. Dorafshan S, Thomas RJ and Maguire M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Constr Build Mater* 2018. 186: 1031-1045.

52. Jang K, Kim N and An YK. Deep learning–based autonomous concrete crack evaluation through hybrid image scanning. *Struct Heal Monit* 2019. 18: 1722-1737.

53. Zhang Z. A flexible new technique for camera calibration. *IEEE T Pattern Anal Mach Intell* 2010. 22: 1330-1334.

54. P. Neary, Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning, IEEE international conference on cognitive computing (ICCC). IEEE, 2018.

55. J. Rijsdijk, L. Wu, G. Perin, and S. Picek, Reinforcement learning for hyperparameter tuning in deep learning-based side-channel analysis. IACR Cryptol. ePrint Arch., vol. 2021, p. 71, 2021.

56. L. Wu, G. Perin, and S. Picek, I choose you: Automated hyperparameter tuning for deep learning-based side-channel analysis" IACR Cryptol. ePrint Arch., vol. 2020, p. 1293, 2020.

57. M. P. Ranjit, G. Ganapathy, K. Sridhar, and V. Arumugham, Efficient deep learning hyperparameter tuning using cloud infrastructure: Intelligent distributed hyperparameter tuning with bayesian optimization in the cloud, 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), 2019.

58. L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, A system for massively parallel hyperparameter tuning, arXiv preprint arXiv:1810.05934, 2018.

59. M. Jaderberg, V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan et al., Population based training of neural networks, arXiv preprint arXiv:1711.09846, 2017.

60. D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, 2017.

61. I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

62. M. D. Zeiler, Adadelta: An adaptive learning rate method, 2012.

63. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, *International Journal of Computer Vision* (IJCV), 2015. 115(3): 211-252.