TriDurLE

National Center for Transportation
Infrastructure Durability & Life-Extension

Project ID: 2022-SDSU-01

AI and Smart Phone Based Damage Detections for Concrete Bridge Decks

Final Report

by

Abir Hadi[b], Ammad Khan[a], Kwanghee Won[b], Mostafa Tazarv[a]

Departments of Civil and Environmental Engineering[a] and Computer Science[b]
South Dakota State University
Brookings, South Dakota

for

National University Transportation Center TriDurLE
Department of Civil & Environmental Engineering
405 Spokane Street PO Box 642910
Washington State University Pullman, WA 99164-2910

August 31, 2024

ACKNOWLEDGEMENTS

The work presented herein was conducted with joint support from the Alaska Department of Transportation and Public Facilities (Alaska DOT&PF), the National Center for Transportation Infrastructure Durability and Life-Extension (TriDurLE), a University Transportation Center (UTC) funded by the U.S. Department of Transportation. The contents of this document reflect the views of the authors, who are responsible for the facts and accuracy of the information presented. This document does not constitute a standard, specification, or regulation.

The support and constructive feedback from the project advisory panel, Ben Fetterhoff, Jesse Escamilla, David McAdoo, Jared Jones, and Elmer Marx of Alaska DOT&PF are greatly appreciated. The authors are indebted to Joshua Olsen of the South Dakota Department of Transportation (SDDOT) for the support and technical input while inspecting the bridges. The research team is thankful to graduate student Andrew Grismer and undergraduate student Rylan Lipetzky for their assistance and efforts.

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. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

ABSTRACT

Regular maintenance of transportation infrastructure, particularly bridges, is essential for extending their service life and reducing long-term repair costs. Transportation agencies such as the Federal Highway Administration (FHWA) emphasize the importance of frequent bridge inspections and recommend inspection schedules and methods. Currently, routine bridge inspections are carried out every two years usually using nondestructive methods. The Alaska Department of Transportation and Public Facilities (DOT&PF) is responsible for condition assessment of approximately 1,000 bridges in the state. Of which, 44% are in good condition, 49% are in fair condition, and 7% are in poor condition according to the 2023 National Bridge Inventory (NBI) data. Alaska DOT&PF engineers inspect approximately 500 bridges each year. The inspector must complete both an NBI inspection and an element-level inspection per bridge.

Nationwide, most steps of bridge inspections such as data collection, defect quantification, and damage reporting are done manually. Furthermore, each of these steps is time consuming, error prone, and hard to repeat in the following inspections due to the manual nature of these activities. For example, the "deck delamination mapping" requires manual detection and measurement of delaminated concrete, patched areas, exposed reinforcing steel bars, and spalling. Such measurements often require traffic control for the safety of inspection crew, and the findings might not be consistent if done by different inspectors.

Computer vision, a field of artificial intelligence (AI) that can analyze scenes at the human level performance, can speed up defect identification and quantification only using images of bridge elements. Furthermore, this AI and other tools can be utilized to expedite and unify reporting. The main goal of this project was to develop practical AI tools that help inspectors with defect detection and quantification, and to facilitate the inspection reporting following DOT&PF requirements. To achieve the project goal, a few bridge elements were targeted for further investigation, inspection databases including photographs of the selected elements with/without damage were compiled, and computer vision tools were developed for the selected elements to recognize the element defects, quantify the defect, and produce a report following standard practices.

A comprehensive literature review on the current scientific trend and AI tools for delamination detection, bridge map generation, and damage state estimation was performed. Then, a new definition of condition states for delamination of concrete bridge decks suitable for computer programming was proposed. Subsequently, more than 45 bridges were inspected, and a comprehensive dataset of inspector annotated delamination was compiled using LiDAR and conventional cameras. Finally, an AI-based tool that can detect, quantify, and map the inspector's annotations, and produce a delamination map of whole bridge deck including their condition states was developed. Overall, the accuracy of the top AI model in finding inspector-marked delamination over the entire length of 18 bridge decks was 99.5%.

TABLE OF CONTENTS

ABBREVIATIONS

AASHTO	American Association of State Highway and Transportation Officials
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CV	Computer Vision
CVT	Computer Vision Tool
DCNN	Deep Convolutional Neural Network
DNN	Deep Neural Network
DOT	Department of Transportation
DSLR	Digital Single-Lens Reflex (Camera)
FCN	Fully Convolutional Networks
FHWA	Federal Highway Administration
ft	Feet
GAN	Generative Adversarial Networks
IMU	Inertial Measurement Unit
in.	Inch
LiDAR	Light Detection and Ranging
kip	1000 pounds
ksi	kip per square inch
NDE	Nondestructive Evaluation
RGB	Red-Green-Blue (Images)
SDSU	South Dakota State University
SfM	Structure from Motion
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

ES.1 Introduction

Regular maintenance of transportation infrastructure such as bridges is crucial in extending their service while reducing long term repair costs. Transportation agencies such as the Federal Highway Administration (FHWA) seek emerging technologies and innovative programs to enhance asset management and maintenance. Furthermore, timely inspections of bridges by trained personnel are mandated per national and state specifications.

The Alaska Department of Transportation and Public Facilities (DOT&PF) is responsible for condition assessment of approximately 1,000 bridges in the state. Of which, approximately 44% are in good condition, 49% are in fair condition, and 7% are rated poor based on the 2023 National Bridge Inventory (NBI) data (Infobridge, 2024).

Alaska DOT&PF engineers inspect approximately 500 bridges each year. Alaska Bridges and Structures Manual (2017) specifies different types of inspections such as inventory, routine, in-depth, and damage. In general, visual inspection is the common practice for inventory and routine inspections and is combined with other tools such as non-destructive evaluation (NDE) in other inspections for enhanced assessment. The inspector must complete both an NBI inspection (following the FHWA Recording and Coding Guide, 1995) and an element level inspection (based on the AASHTO MBEI, 2019) per bridge.

Nationwide, most steps of bridge inspections such as data collection, defect quantification, and damage reporting are done manually. Furthermore, each of these steps is time consuming, error prone, and hard to repeat in the following inspections due to the manual nature of these activities. For example, the “deck delamination mapping” requires manual detection and measurement of delaminated concrete, patched areas, exposed reinforcing steel bars, and spalling. Such measurements often require traffic control for the safety of inspection crew, and the findings might not be consistent if done by different inspectors.

Computer vision, a field of artificial intelligence (AI) that can analyze scenes at the human level performance, can speed up defect identification and quantification only using images of bridge elements. Furthermore, this and other AI tools can be utilized to expedite and unify reporting. Even though the academic and industry level developments are interesting and on the right track for field deployment, they usually lack practicality and are overwhelming for everyday users. For example, the academic-level computer programs are not as user-friendly as commercial software, and the industry-level software packages are so enlarged targeting multiple assets preventing them from being practical. In summary, current tools do not fully address the needs of DOTs for routine bridge inspections.

ES.2 Objectives

The main goal of this project was to develop practical AI tools that help inspectors with defect detection and quantification, and to facilitate the inspection reporting following NBI and MBEI requirements for DOT&PF. To achieve the project goal, a few bridge elements were targeted for further investigation, inspection databases including photographs of the selected elements with and without damage were compiled, and computer vision tools were developed for the selected elements to recognize the element defects, quantify the defect per NBI/MBEI, and produce a report following the DOT standard practices.

The present report includes a summary of the activities that were focused only on concrete bridge decks to quantify delamination marked by inspectors. Specifically, the report presents a novel approach to quantify and map bridge deck delamination using iPhone’s LiDAR camera, computer vision, and deep learning networks. First, a comprehensive literature review of the current scientific trend and AI tools is presented on delamination detection, bridge map generation, and damage state estimation. Then, a new definition of damage states for delamination of concrete bridge decks suitable for computer programming was proposed. Subsequently, more than 40 bridges were inspected, and a comprehensive dataset of inspector annotated delamination was compiled using LiDAR and conventional cameras. Finally, an AI-based tool that can detect, quantify, and map the inspector’s annotations, and produce a delamination map of whole bridge deck including their condition states was developed. Other bridge elements, various data collection modalities, and autonomous damage map generation are currently under investigation by the research team.

ES.3 Literature Review

Transportation infrastructure inspection is a mandated routinely scheduled job by the federal government. The inspection result of assets enables the authority to maintain and forecast possible repair and reconditioning. One of the key assets in the federal and state highway authorities is bridges connecting roadways. A brief review of recent trends on bridge assessments using non-destructive techniques and computer vision assisted evaluation of bridge conditions is discussed. The focus was on assessment of bridges with concrete decks, concrete decks with asphalt overlay and closely related structures.

ES.3.1 Nondestructive Evaluation Techniques for Bridge Decks

The structural integrity and longevity of bridge decks are critical for transportation agencies. To maintain and ensure their safety and without causing any damage, NDE techniques are employed. These methods allow for the assessment of concrete quality, detection of internal defects, and evaluation of corrosion in steel reinforcements. NDE techniques, such as chain drag, half-cell potential, impact echo, and electrical resistivity, have become invaluable tools in bridge maintenance, offering precise, reliable, and efficient means for early detection and preventive maintenance.

Chain drag method is the simplest yet effective technique for detecting delamination in which a steel chain is dragged on concrete. The areas with a different sound compared with that of solid concrete are usually damaged and delaminated.

The half-cell potential (HCP) method is widely used for detecting corrosion in reinforced concrete structures. In HCP method, an electrode is attached to the steel bars and another electrode is held by the inspector and places the electrode at different intervals on the bridge deck. The potential difference between the electrodes varies based on the presence of corrosion in reinforcing steel bars in concrete (Gucunski and Nazarian, 2010). Using the collected data, a map is generated to visually present the damage of the entire bridge deck. ASTM C876-22b (2022) is utilized to perform HCP and the data processing needs expert knowledge. Further, recent advancements focus on integrating this technique with other methods to enhance diagnostic accuracy.

Impact echo (IE) method utilizes acoustic waves to assess the condition of bridge decks and identifies delamination in concrete. P-waves are generated which travel through concrete. If they encounter different materials or air pockets, they reflect energy which is received by the device. Recent innovations include the use of advanced signal processing algorithms and machine learning to better interpret results. Portable and user-friendly devices are now available, allowing for more frequent and widespread use in the field.

Electrical resistivity (ER) measurement is used to assess the moisture content and corrosion in concrete. The ingression of impurities and presence of air void in concrete enhances the electric resistance of concrete. The presence of voids filled with moisture lowers the concrete electric resistance (Khudhair and Gucunski, 2023). Modern trends emphasize the development of non-invasive, high-resolution resistivity mapping techniques.

ES.3.2 Computer Vision for Bridge Element Inspection

With a widespread use of computer vision techniques, the expert communities are exploring its application in bridge inspection tasks. Matsumoto (2014) used line sensor camera and infrared thermography to detect bridge deck defects and delamination. Their proposed framework for bridge inspection and repair consists of three phases. In Phase 1, the deck is scanned using video imagery. Phase 2 involves prioritizing the areas of the deck that are distressed, determining their condition state according to the National Bridge Inventory (NBI) scale of 1 to 4, and planning necessary repairs or rehabilitation. In Phase 3, selected spans of the bridge are scanned from underneath to further assess and address the identified issues.

Won and Sim (2020) detected transverse cracks on bridge decks using optical sensors. They used a three-dimensional (3D) computer vision-based localization and mapping technique by utilizing stereovision setup. The proposed framework utilizes multi-view image sequences with a Bag-of-Words representation, eliminating the need for traditional GPS or odometer data for localization. For crack detection, the system uses crack hierarchy and circular histograms instead of conventional methods that rely on-line segments and intensities. Additionally, it creates a comprehensive transverse crack database based on bespoke ontologies and schemas, covering the entire bridge deck. These advancements aim to enable faster deployment and achieve higher accuracy in large-scale crack mapping and measurement. They collected a dataset of 2,638 image patches, consisting of 1,262 from cracked areas and 1,376 from non-crack (background) areas of the bridge. Two classification approaches were applied to these patches: a conventional method and a crack pixel-based representation approach. A support vector machine (SVM) classifier with a radial basis function kernel was trained using the data, with parameters chosen empirically.

Perry et al. (2020) used Structure-from-Motion (SfM) to generate 3D reconstructed model of bridges and to identify target elements. Then they used image processing techniques such as the black hat transform with the Canny edge detection to detect bridge defects. The black hat transform is a morphological operation used in image processing to highlight dark features on a bright background. Specifically, it emphasizes the portions of an image that are smaller than the structuring element used in the transform and darker than their surroundings. The authors also proposed a method to track growth of defects through image comparison.

Ali et al. (2021) used computer vision techniques with a faster region-based convolutional neural network (R-CNN) damage detection framework on data collected using drones for GPS denied locations to identify structural damage related to loosened bolts and steel corrosion. The process begins with identifying the structure to be inspected, followed by deploying an autonomous unmanned aerial vehicle (UAV) equipped with a proportional-integral-derivative (PID) controller for flight stability, an ultrasonic beacon system for precise positioning, and an action camera for capturing visual data. The data is transmitted in real-time using a real-time streaming protocol. A modified faster R-CNN machine learning model, consisting of a region proposal network (RPN), object proposals, and a Fast R-CNN, is employed for image analysis. The model identifies and highlights structural damage. The ground station processes and analyzes the collected data, completing the inspection process. This integration of robotics, computer vision, and machine learning enables efficient and safe autonomous inspection of structures.

Tazarv et al. (2022) used computer vision techniques for inspection and condition assessment of RC bridge columns after earthquakes to automatically determine their conditions. They used deep learning based cascaded detection of deficiencies in their framework using Mask R-CNN. The cascaded detection framework begins with Stage 1, which detects columns in images. If columns are not detected, the user is prompted to find them manually. Once columns are identified, the process moves to Stage 2 for detecting longitudinal bars, followed by Stage 3 for detecting transverse bars. If both bars are detected, the system reports damage state 4 or 5. Stage 4 involves spalling detection; if spalling is found, it results to damage state 3 or 4. If not, it continues to Stage 5 for crack detection, leading to damage state 1 or 2. Their results indicate high performance across all components.

Dunphy et al. (2022) developed a multiclass damage detection framework for concrete structures using a combination of generative adversarial networks (GANs) and CNNs. It begins with a multiclass image database of various concrete conditions, such as undamaged, cracked, construction joint, pitting, and spalling. The database was split into labeled and unlabeled datasets. GAN training was performed on unlabeled data, producing a trained generator and discriminator network. The trained discriminator was then used for transfer learning to initialize the weights and biases of the CNN. This CNN was further trained using the labeled data. Finally, hyper-training was applied to the fully connected layer of the CNN to optimize it for image classification tasks, leading to accurate identification of different types of concrete damage.

Jiang et al. (2023) developed a framework using object detection models to identify steel bolts and nuts, as well as their condition, including corrosion and tightness. The proposed method for automatic bolt inspection was a vision-based system leveraging image processing and deep learning algorithms with images captured by UAV. It consisted of three main components: bolt image acquisition, data preprocessing, and bolt damage identification. The image acquisition phase involved using a modified UAV to capture images of bolts from various angles, including the bridge’s sides, bottom, and cable clamps. Data preprocessing addresses two key issues: image motion blur due to the UAV’s high speed and variations in bolt pixel size caused by differing object distances. To tackle these, an image motion deblurring method using inverse filtering and optical flow was employed to correct motion blur, and an adaptive scale segmentation method with multi-scale template matching and enhanced super-resolution generative adversarial networks (ESRGANs) ensured uniform bolt image sizes of 240 × 240 pixels. Finally, a two-stage bolt damage identification process was used to detect and classify bolt conditions.

Jiang et al. (2023) proposed a three-stage framework for bridge condition assessment and damage localization using UAV collected data. The authors used the YOLOv3 object detection network to detect several types of damage such as cracks, corrosion, and spalling. During data collection, the authors also used visual simultaneous localization and mapping (VSLAM) techniques to localize each damage on bridges by retrieving camera pose with respect to reference world coordinates. To achieve this, the authors used stereo-inertial SLAM that used VSLAM with Inertial Measurement Unit (IMU) sensor fusion to increase the robustness of the system. Integration of IMU with VSLAM made the process more robust because of a better estimation of the camera poses for each image. Their framework was tested on a long-span river bridge, offering a more rigorous evaluation environment than standard indoor tests due to factors like strong winds and a large detection area.

Kao et al. (2023) proposed a framework to quantify cracks in bridges using UAV images and deep learning-based object detection models. They used planar markers to retrieve scales of images then cracks. To retrieve the crack boundaries, they used a local thresholding method to create a binarized image. Following that, the authors employed a crack width measurement technique in reference to the planar marker. This model performed well in identifying cracks even in images with uneven lighting and complex backgrounds. The study also demonstrated that the overall crack measurement accuracy exceeded 0.22 mm. Two edge detection methods evaluated in the study showed similar performance;

however, the Canny edge detector's results varied with different thresholds, leading to greater discrepancies between measured and actual crack widths. In contrast, the morphological edge detector, which does not rely on thresholds, provided crack edges more consistent with the true dimensions.

Mirzazade et al. (2023) used a 3D reconstruction approach SfM to detect and segment bridge damages based on U-Net, a popular deep learning model for semantic segmentation. SfM is a computer vision technique that reconstructs 3D structures from 2D image (only width and height) sequences taken from different viewpoints. It estimates camera positions and the 3D coordinates of scene points by analyzing the motion and structure captured in the overlapping images. They also collected Light Detection and Ranging (LiDAR) scans of structures to verify the integrity of the 3D reconstruction using SfM.

ES.4 Proposed Condition States for Concrete Decks

Delamination is a state of separation of layers within the concrete bridge deck. This leads to loss of structural integrity of the deck and increase maintenance cost. Detection and forecasting the growth of the delamination is inherently difficult because of the lack of visual perception. ASSHTO (MBEI, 2019) defines the condition state of such deficiencies (e.g., **Table ES.1** for bridge decks)

Table ES.1 – AASHTO Condition States for Concrete Decks (AASHTO MBEI, 2019)

Defects	CS 1	CS 2	CS 3	CS 4
	Good	**Fair**	**Poor**	**Severe**
Delamination/Spall/Patched Area (1080)	None.	Delaminated. Spall 1in. or less deep or 6in. or less in diameter. Patched area that is sound.	Spall greater than 1in. deep or greater than 6in. diameter. Patched area that is unsound or showing distress. Does not warrant structural review.	The condition warrants a structural review to determine the effect on strength or serviceability of the element or bridge; OR a structural review has been completed and the defects impact strength or serviceability of the element or bridge.
Exposed Rebar (1090)	None.	Present without measurable section loss.	Present with measurable section loss but does not warrant structural review.	
Efflorescence/Rust Staining (1120)	None.	Surface white without build-up or leaching without rust staining.	Heavy build-up with rust staining.	
Cracking (RC) (1130)	Insignificant cracks or moderate width cracks that have been sealed.	Unsealed moderate width cracks or unsealed moderate pattern (map) cracking.	Wide cracks or heavy pattern (map) cracking.	
Abrasion/Wear (PSC/RC) (1190)	No abrasion or wearing.	Abrasion or wearing has exposed coarse aggregate but the aggregate remains secure in the concrete.	Coarse aggregate is loose or has popped out of the concrete matrix due to abrasion or wear.	

Any defects that are expected to be found using computer tools need to be quantitative but not qualitative. AASHTO definitions are not measurable at all four levels (e.g., CS-4). To this end, **Table ES.2** presents alternative definitions of condition states for concrete decks specifically for delamination, spalling, and patching. The new definitions are measurable and programmable. Furthermore, to minimize variations from AASHTO, the proposed definitions are kept at four levels. Comparing with the AASHTO definitions, some parameters, such as the depth of defects, were removed in the new definitions since they are challenging to estimate using computer vision techniques.

Table ES.2 – Proposed Condition States for Delamination, Spalling, and Patching in Concrete Decks

Defects	CS 1	CS 2	CS 3	CS 4
	Good	**Fair**	**Poor**	**Severe**
Delamination, Spall, or Patched Area (1080)	Defected area is less than or equal to 7 in^2 (45 cm^2)	Defected area is greater than 7 in^2 (45 cm^2) but less than 30 in^2 (195 cm^2)	Defected area is greater than 30 in^2 (195 cm^2) but less than 110 in^2 (710 cm^2)	Defected area is greater than 110 in^2 (710 cm^2)

ES.5 Bridge Deck Deficiency Database

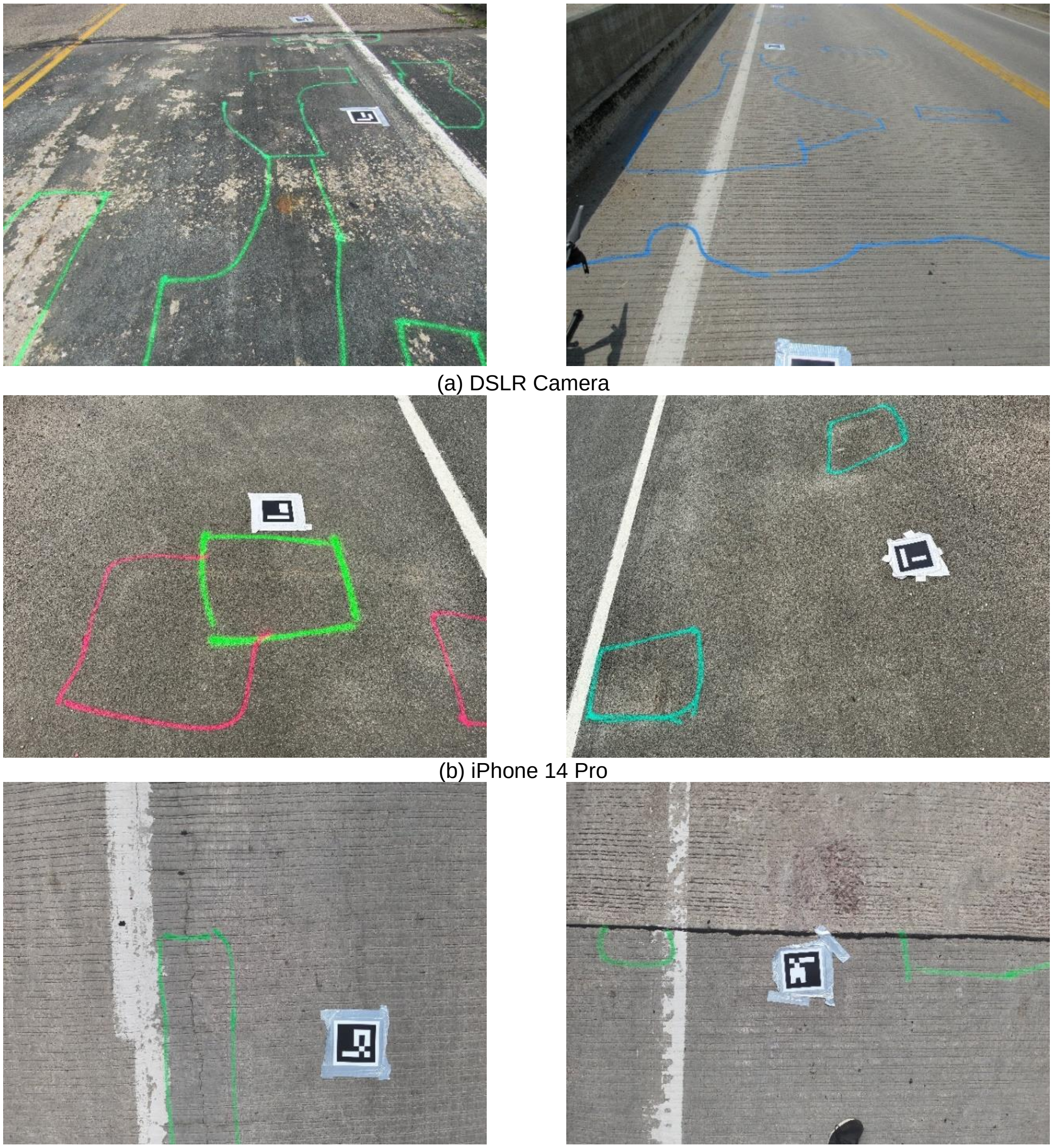
There are several existing datasets specific to bridge health assessment including bridge elements such as columns, decks, and overlays. Specifically designed for crack detection on concrete surfaces, the CrackForest dataset by Shi et al (2016) includes annotated images of cracks on various concrete structures, including bridges. A derivative of the Common Objects in Context (COCO) dataset by Bianchi et al. (2020), was tailored for bridge inspection tasks, which includes annotated images of bridges highlighting various structural elements and potential defects. Dorafshan et al. (2021) provided a dataset containing images of concrete surfaces, including bridges, annotated for different types of defects such as cracks and spalling. The Road Damage Detection (RDD) dataset by Arya et al. (2022) was for the road surface but included images of bridge decks. The dataset is useful for detecting surface defects such as potholes and cracks. **Table ES.3** presents a summary of past studies related to image datasets.

Table ES.3 – Summary of Past Studies on Bridge Related Image Datasets

Reference	Dataset	Categories	Data Type	Size	Application
Shi et al. (2016)	CrackForest	Cracks	RGB Image	10,000	Detect and analyze cracks on the road surface
Zhang et al. (2016)	CCIC	Cracks	RGB Image	40,000	Detect cracks on concrete surfaces
Rubio et al. (2019)	--	Exposed Rebar, Delamination	RGB Images	734	Detect and quantify exposed rebars and delamination
Zou et al. (2019)	DeepCrack	Cracks	RGB Image	35,100	Detect fine cracks on various road surfaces
Bianchi et al. (2020)	COCO Bridge Damage	Bearing, Cover-plates, Gusset-plates, Stiffeners	RGB Image	774	Detect bearing, cover-plate, gusset-plate, stiffeners and evaluate their conditions
Dorafsan et al. (2021)	SDNET2021	Delamination	Infrared Thermography, Impact Echo, Ground Penetrating Radar	1,936	Detect delamination and reinforcement corrosion
Arya et al. (2022)	RDD2022	Cracks and Potholes	RGB Image	47,420	Detect and classify road damage types, such as cracks, potholes, and surface wear
Tazarv et al. (2022)	RC Column Damage	Cracks, Spalling, Exposed Rebars	RGB Image	3,036	Detect post-earthquake damages of RC bridge columns and determine their damage states

The available datasets do not explicitly include delamination data and lack thermal image data, which may be used for identifying internal defects like delamination. Further, the existing datasets do not include annotations from experts, such as the Department of Transportation (DOT) inspectors, for delamination to be used as ground truths. Therefore, collecting new data was essential in the present study due to the limitations of the existing datasets.

To collect new data for bridge decks, the research team accompanied with inspectors from the South Dakota Department of Transportation (SDDOT) inspected more than 45 bridges in South Dakota in the summers of 2023 and 2024. The inspectors used the chain drag method to evaluate the bridge decks for delamination and annotated them with paints. The research team then collected RGB images using digital single-lens reflex (DSLR) cameras, iPhone 14 Pro, and DJI drones (Matrice 200, and Mavic 3T) before and after the annotations. Furthermore, PolyCam, a mobile application utilizing iPhone LiDAR camera, was used to scan the bride decks to record three-dimensional (3D) meshes of each deck. RGB images taken by iPhone 14 Pro had a resolution of 8064 by 6048 pixels. Further, the RGB images obtain by the DJI Mavic 3T and the DSLR camera had a resolution of 4000 by 3000 and 4608 by 3456 pixels, respectively. The images taken by the drones were taken approximately at a height of 3 meters from the deck surface. For the images taken by the other cameras, the height varied between 1 to 2 meters. **Figure ES.1** shows a few samples from each of the data collection devices. In total, 1918 RGB images were collected from 18 bridges that had delamination detected and annotated by the inspector using chain drag. Furthermore, the deck of 18 bridges with delamination were LiDAR scanned and data was processed per bridge. PolyCam utilizes the integration of LiDAR data and photogrammetry to enhance the precision and detail of 3D models generated using iPhones that are equipped with LiDAR sensors, such as the iPhone 12 Pro and subsequent models. The process initiates with the iPhone's LiDAR sensor, which emits light pulses to measure the time taken for the reflections to return from various objects, thereby generating a highly accurate depth map of the scene. Concurrently, multiple images of the scene from various angles are also captured using the iPhone's camera. Photogrammetry algorithms subsequently analyze these visual inputs to discern the structural, textural, and detailed attributes of the scene. The integrated data then undergoes further processing to produce a coherent 3D model of the target object.



(a) DSLR Camera

(b) iPhone 14 Pro

(c) DJI Mavic 3T

Figure ES.1 – Sample RGB Images from Inspector Annotation Dataset

ES.6 Bridge Deck Damage State Detection Software

In the present project, the research team developed a computer vision-based bridge deficiency detection, quantification, and condition state estimation software that is focused on delamination in concrete bridge decks. **Figure ES.2** shows a high-level architecture of the software. The software has several interconnected modules to allow data collection and analyses using different devices and methods. Images can be acquired using different sources (DSLR, iPhone cameras, or cameras on UAVs). Further, 3D maps of bridge decks are obtained from the iPhone's LiDAR camera. Subsequently, the image dataset was used in deep learning models. For each bridge, a point-cloud map is obtained using an iPhone and then run through the software to generate a 2D map of the deck. The software then detects local damages (delamination) using the trained neural network and pinpoints the damage on the global map of the deck. The findings are then summarized. In the following sections, a brief introduction of computer vision tasks is presented first, then the components of the software are discussed.

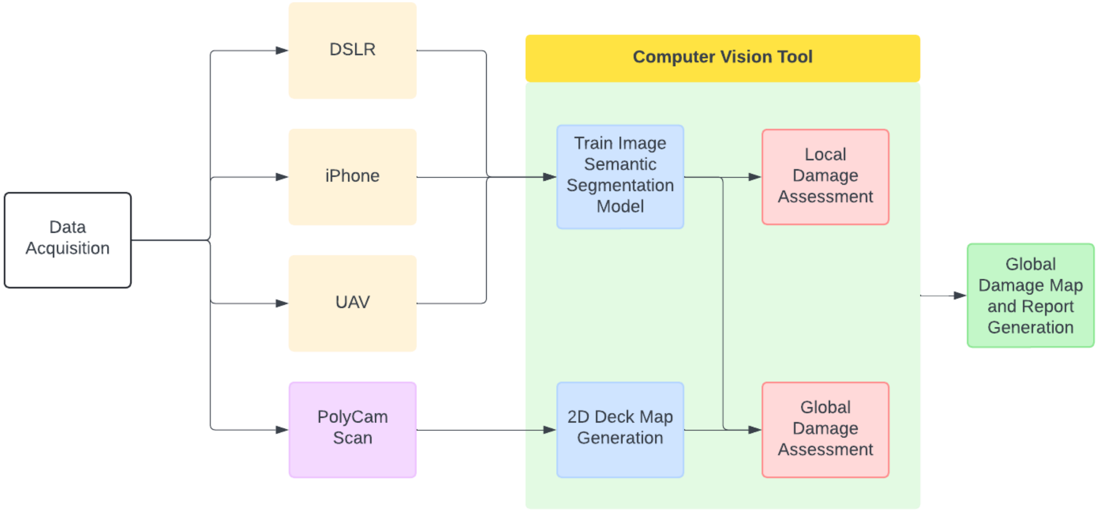


Figure ES.2 – Proposed Computer Vision Framework for Delamination Detection of Bridge Decks

To detect the damages on bridge deck, such as delamination, patches, and inspector annotations to quantify the data, deep learning models may be trained to learn the tasks from existing datasets. This process requires data preparation, training on the dataset, domain adaptation and performance evaluation of domain adapted model on the collected data. The model learns from the prepared data to classify each pixel and compare its prediction with ground. The outcome of the prediction determines how much the model has learnt and needs to be updated to make performance better.

The training set was color augmented in the hue, saturation, and value (HSV) color space to generate different variations of the inspector annotated delamination spots. This augmentation was applied randomly with a probability of 50%. Different geometric augmentation such as scaling and rotations were also applied. Scaling ranged from 0.5 to 1 of the training sample with a probability of 50% to be applied. A rotation of -45 degree to 45 degrees was applied to increase the number samples. The dataset was then split into a training and testing set using the industry standard 0.9:0.1 ratio. The training set was further split into a train and validation split of 0.9:0.1 ratio. **Table ES.4** presented the sample size for the inspector annotation dataset during training and testing. Note that the training set of 1,553 was increased to 70,000 after performing the above-mentioned augmentations.

Table ES.4 – Dataset Distribution

Set	No. of Samples
Train	1553
Validation	172
Test	192

Three different neural networks (U-Net, Attention U-Net, Recurrent Residual U-Net) were selected with different hyperparameters, but the same loss function (the Dice function) was used in all models. **Table ES.5** lists the best hyperparameters of the models during training on the dataset.

Table ES.5 – Hyperparameters of Evaluated Deep Learning Models During Training

Model	Batch Size	Learning Rate	Weight Decay	Momentum
U-Net	5	0.0001	1×10^{-8}	0.999
Attention U-Net	2	0.0005	1×10^{-8}	0.99
Recurrent Residual U-Net	4	0.0001	1×10^{-8}	0.99

The model learns from the prepared data to classify each pixel and compares its predictions with benchmark data, which is called a ground truth. The outcome of the prediction determines how much the model has learned or needs to be updated and enhanced. **Figure. ES.3** shows the proposed workflow for delamination detection of concrete bridge decks using inspector marks and its quantification process.

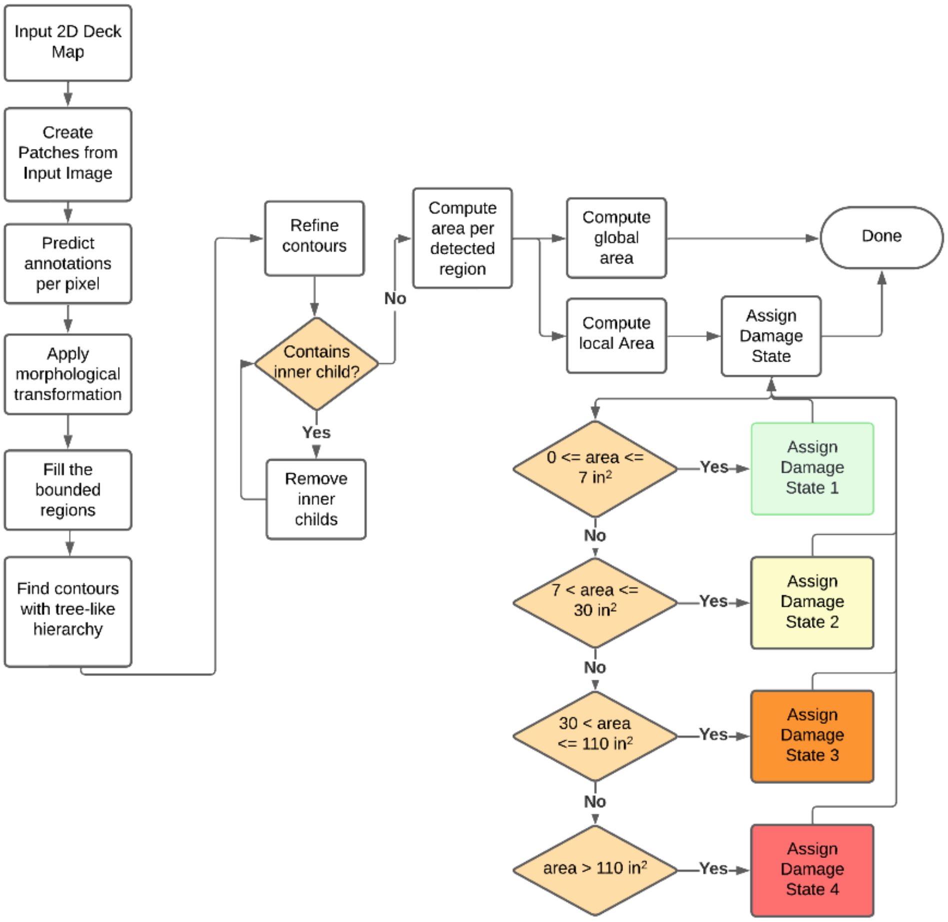
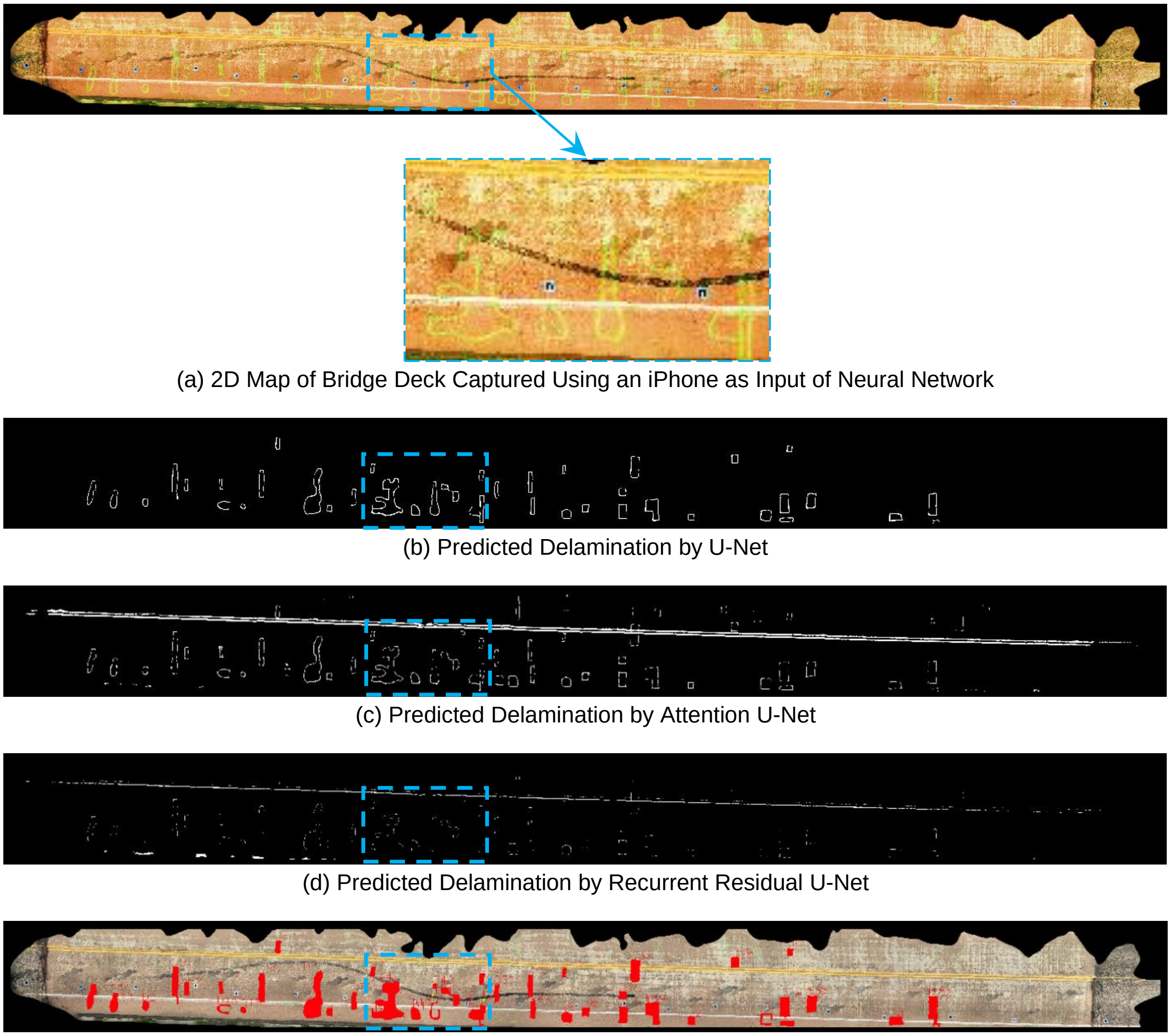


Figure ES.3 – AI-based Delamination Detection Flowchart Using LiDAR and Inspector Annotations

Once a 2D map of the bridge (either lane by lane or whole width of the bridge) is obtained, it is forwarded to the neural network for semantic segmentation and quantification of the inspector annotations. A delamination analysis at the global scale (e.g., area-to-area at the full length of the bridge) provides a complete assessment of the entire processes and methods of autonomous delamination detection in real applications. **Figure ES.4** shows a sample of delamination assessment for a bridge on its full length, which was 75.4 meters.



(a) 2D Map of Bridge Deck Captured Using an iPhone as Input of Neural Network

(b) Predicted Delamination by U-Net

(c) Predicted Delamination by Attention U-Net

(d) Predicted Delamination by Recurrent Residual U-Net

(e) Condition States of Delaminated Regions Based on U-Net shown in (b)

Figure ES.4 – Delamination Assessment of a 75.4-m Bridge

Figure ES.4a shows the input image, a 2D map of the bridge deck including the marked delamination by the inspector (the enclosed areas with green paint), to the neural network. This input image is then processed to extract patches with a size of 512-by-512 pixels with a stride of 256. This makes each patch have a 50% overlap with the adjacent ones. **Figures ES.4b-d** show the predicted pixels by the three neural networks as the delamination boundaries (finding the green markups automatically). The findings are then used to compute the delamination condition state for individual zone using a contour analysis on the post-processed prediction map. The predicted delaminated pixels are then connected using morphological transformations and passed through contour analysis. After the completion of the contour analysis, which was done to enclose each delamination and to obtain its shape, each of the delaminated areas is then color coded based on the severity of the delamination. The scale information was retrieved from the mesh data. **Table ES.6** presents a summary of global (area to area) evaluation results. The U-Net and Attention U-Net models performed well on the test set while Recurrent Residual U-Net showed an inferior performance on the test and deck map sets. Overall, U-Net performed better than the other two models and is recommended for field applications.

Table ES.6 – Summary of Neural Network Evaluation Results

Model	Deck Map (Global) Performance (Area Level)
U-Net	99.5%
Attention U-Net	84.5%
Recurrent Residual U-Net	< 5%

ES.7 Summary and Conclusions

Regular maintenance of transportation infrastructure such as bridges is crucial in extending their service while reducing long term repair cost. Transportation agencies such as the Federal Highway Administration (FHWA) seek emerging technologies and innovative programs to enhance asset management and maintenance. Furthermore, timely inspections of bridges by trained personnel are mandated per national and state specifications. The Alaska Department of Transportation and Public Facilities (DOT&PF) engineers inspect approximately 500 bridges each year. Of which, approximately 44% are in good condition, 49% are in fair condition, and 7% are rated poor based on the 2023 National Bridge Inventory (NBI) data.

In general, visual inspection is the common practice for inventory and routine inspections and is combined with other tools such as non-destructive evaluation (NDE) in other inspections for enhanced assessment. The inspector must complete both an NBI inspection and an element level inspection for each bridge. Nationwide, most steps of bridge inspections such as data collection, defect quantification, and damage reporting are done manually. Furthermore, each of these steps is time consuming, error prone, and hard to repeat in the following inspections due to the manual nature of these activities. Such measurements often require traffic control for the safety of inspection crew, and the findings might not be consistent if done by different inspectors.

Computer vision, a field of artificial intelligence (AI) that can analyze scenes at the human level performance, can speed up defect identification and quantification only using images of bridge elements. Progress in computer aided analysis of visual imagery has initiated widespread use of computer vision tools in various tasks, including but not limited to object classification, categorization, detection and generation by learning from provided datasets for specific tasks. Even though the academic and industry level developments are interesting and on the right track for field deployment, they usually lack practicality and are overwhelming for everyday users. For example, the academic-level computer programs are not as user-friendly as commercial software, and the industry-level software packages are so enlarged targeting multiple assets preventing them from being practical. In summary, current tools do not fully address the needs of DOTs for routine bridge inspections.

The primary objective of this project was to develop practical AI tools that help inspectors with defect detection and measurement and facilitate the inspection reporting following standard practices. To achieve the project goals, a comprehensive literature review of the current scientific trend and AI tools was presented on delamination detection, bridge map generation, and damage state estimation. Then, a new definition of damage states for delamination of concrete bridge decks suitable for computer programming was proposed. Subsequently, more than 40 bridges were inspected, and a comprehensive dataset of inspector annotated delamination was compiled using LiDAR and conventional cameras. Finally, an AI-based tool that can detect, quantify, and map the inspector's annotations, and produce a delamination map of whole bridge deck including their condition states was developed.

The following conclusions can be drawn from this computer vision-based study on bridge inspections:

- A new condition state definition was developed for delamination of concrete bridge decks. The new definition enhances the physical interpretation of existing criteria in AASHTO by using definitive and quantitative measures at all four levels that are suitable for computer programming and automated analysis. The new definition ensures greater alignment between manual assessments and computer-based analyses, improving the accuracy and consistency of damage evaluations.
- A new image-LiDAR dataset of inspector annotated delamination was compiled after inspecting more than 40 bridges. The images are particularly applicable in training neural network models to automatically detect and measure inspector-annotated delaminated areas. This dataset contains 1,918 RGB photographs of inspector annotations and 18 LiDAR scans of bridge decks.
- Three computer vision models for image segmentation (U-Net, Attention U-Net, and Recurrent Residual U-Net) were evaluated following through hyper-parameter tuning, image preprocessing, and augmentation. The color and geometric augmentations of a somewhat small training dataset expanded the dataset to 70,000 to improve the generalization and accuracy.
- A method to generate high-quality bridge deck maps was developed to extract 2D bridge deck maps from reconstructed 3D mesh data. This allows high accuracy in quantification of the annotated area by inspectors.
- Two-level assessment of the detection of delaminated area in bridge decks were conducted. The local delamination assessment, which evaluates pixel-to-pixel correspondences, was conducted to measure the quality of the granular prediction. The pixel-to-pixel level accuracy was less than 75% for the best model. The global delamination assessment (area-to-area) exhibited, on the other hand, a near-to-perfect performance with 99.5% accuracy over 18 bridges. Furthermore, the global assessment confirmed that regardless of the quality of the local assessment, the framework can quantify the area inside the delaminated spot with high accuracy.

Overall, the proposed smart-phone and computer vision based study can expediate bridge deck inspections for delamination by automating the entire process of data collection using iPhone, analysis of point-clouds, and damage identification and condition state estimation using neural networks.

CHAPTER 1. INTRODUCTION

1.1 Introduction

Regular maintenance of transportation infrastructure such as bridges is crucial in extending their service while reducing long term repair cost. Transportation agencies such as the Federal Highway Administration (FHWA) seek emerging technologies and innovative programs to enhance asset management and maintenance. Furthermore, timely inspections of bridges by trained personnel are mandated per national and state specifications.

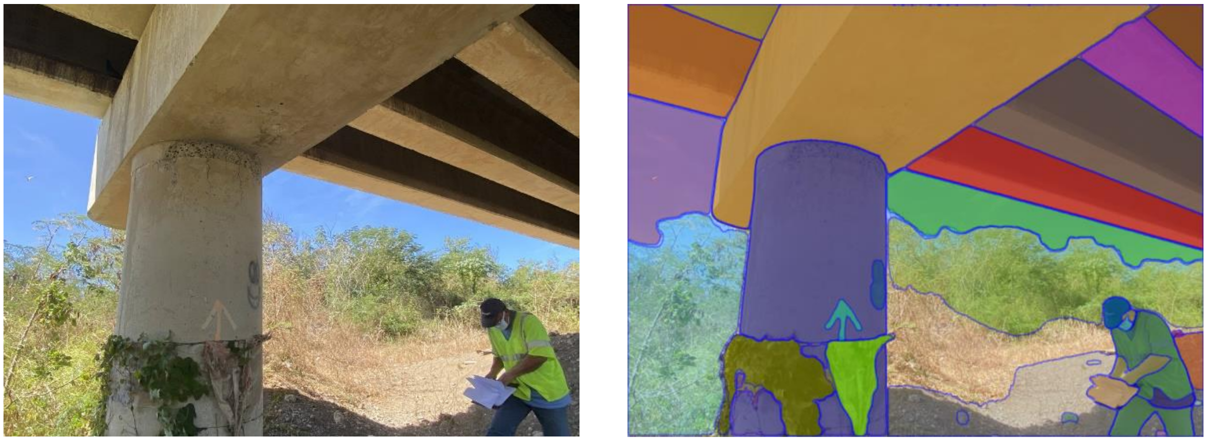
The Alaska Department of Transportation and Public Facilities (DOT&PF) is responsible for condition assessment of approximately 1,000 bridges in the state. Of which, approximately 44% are in good condition, 49% are in fair condition, and 7% are rated poor based on the 2023 National Bridge Inventory (NBI) data (Infobridge, 2024).

Alaska DOT&PF engineers inspect approximately 500 bridges each year. Alaska Bridges and Structures Manual (2017) specifies different types of inspections such as inventory, routine, in-depth, and damage. In general, visual inspection is the common practice for inventory and routine inspections and is combined with other tools such as non-destructive evaluation (NDE) in other inspections for enhanced assessment. The inspector must complete both an NBI inspection (following the FHWA Recording and Coding Guide, 1995) and an element level inspection (based on the AASHTO MBEI, 2019) per bridge.

Nationwide, most steps of bridge inspections such as data collection, defect quantification, and damage reporting are done manually. Furthermore, each of these steps is time consuming, error prone, and hard to repeat in the following inspections due to the manual nature of these activities. For example, the "deck delamination mapping" requires manual detection and measurement of delaminated concrete, patched areas, exposed reinforcing steel bars, and spalling. Such measurements often require traffic control for the safety of inspection crew, and the findings might not be consistent if done by different inspectors.

Computer vision, a field of artificial intelligence (AI) that can analyze scenes at the human level performance, can speed up defect identification and quantification only using images of bridge elements. **Figure. 1.1** shows sample of an image segmentation using the Segment Anything model (Kirillov et al., 2023), where all pixels associated with bridge elements, and also other elements of the scene, are grouped and color coded.

Furthermore, this and other AI tools can be utilized to expedite and unify reporting. Even though the academic and industry level developments are interesting and on the right track for field deployment, they usually lack practicality and are overwhelming for everyday users. For example, the academic-level computer programs are not as user-friendly as commercial software, and the industry-level software packages are so enlarged targeting multiple assets preventing them from being practical. In summary, current tools do not fully address the needs of DOTs for routine bridge inspections.



(a) Original Image (b) Segmented Image

Figure 1.1 – Sample of Computer Vision Application

1.2 Objectives and Scope

The main goal of this project was to develop practical AI tools that help inspectors with defect detection and quantification, and to facilitate the inspection reporting following NBI and MBEI requirements for DOT&PF. To achieve the project goal, a few bridge elements were targeted for further investigation, inspection databases including photographs of the selected elements with and without damage were compiled, and computer vision tools were developed for the selected elements to recognize the element defects, quantify the defect per NBI/MBEI, and produce a report following the DOT standard practices.

The present report includes a summary of the activities that were focused only on concrete bridge decks to quantify delamination marked by inspectors. Specifically, the report presents a novel approach to quantify and map bridge deck delamination using iPhone's LiDAR camera, computer vision, and deep learning networks. First, a comprehensive literature review of the current scientific trend and AI tools is presented on delamination detection, bridge map generation, and damage state estimation. Then, a new definition of damage states for delamination of concrete bridge decks suitable for computer programming was proposed. Subsequently, more than 40 bridges were inspected, and a comprehensive dataset of inspector annotated delamination was compiled using LiDAR and conventional cameras. Finally, an AI-based tool that can detect, quantify, and map the inspector's annotations, and produce a delamination map of whole bridge deck including their condition states was developed.

Other bridge elements, various data collection modalities, and autonomous damage map generation are currently under investigation by the research team.

1.3 Expected Contributions

The main outcome of this project is an inspection software for concrete bridge decks to detect and map delamination using iPhone and AI. The main products of the projects are: (1) a comprehensive inspector annotation dataset for concrete bridge deck delamination, (2) a computer vision tool to process 3D meshes obtained by iPhone using PolyCam, an augmented reality-based iOS (Apple), application which further processes the 3D mesh into a 2D map of the bridge deck, (3) and a deep-learning tool that can detect delamination marked by inspector and estimate its condition state.

1.4 Document Outline

As part of cover materials, an executive summary was presented in a chapter with the same name. Chapter 1 presents an introduction to the study and the scope of the work and lists the expected contributions. Chapter 2 discusses current literature review on non-destructive techniques for bridge deck inspection and computer vision tool development trends for bridge health assessment. Chapter 3 discusses AASHTO guidelines and condition state definitions for delamination and proposes new condition states suited for computer programming. Chapter 4 introduces a comprehensive image dataset collected for delamination. Chapter 5 provides details of the proposed framework, AI tool development, and a summary of the results on real world data outside of the collected dataset. Finally, Chapter 6 provides a summary and conclusion of the study.

CHAPTER 2. LITERATURE REVIEW

2.1 Introduction

Transportation infrastructure routine inspection is a scheduled activity mandated by federal and state agencies to collect various types of data such as overall and element-level defects. The outcomes of state-wide inspections enable the authority to maintain and forecast repair needs and costs. Bridges are usually inspected every two years and inspection data is collected and stored in state and federal repositories. Part of such data for all 50 states of the US is publicly available (Infobridge, 2024) for research and development.

In this chapter, a review of recent trends on bridge assessments using nondestructive evaluation (NDE) and other cutting-edge technologies such as artificial intelligence (AI) and computer vision is discussed. The review is mostly focused on concrete bridge decks.

2.2 Bridge Element Inspection and Assessment

Bridge elements with a four-level condition state are defined in the AASHTO Manual for Bridge Element Inspection (AASHTO MBEI, 2019). Some key elements are discussed below.

2.2.1 Bridge Decks

Bridge deck inspection guideline focuses on deck elements such as surface, overlays, and protective coatings. The deck inspection specifically focuses on wearing surfaces such as flexible overlays, epoxy overlays, and concrete protective coating. **Table 2.1** presents the AASHTO definition of condition states for concrete decks.

Table 2.1 – AASHTO Condition States for Concrete Decks (AASHTO MBEI, 2019)

Defects	CS 1	CS 2	CS 3	CS 4
	Good	**Fair**	**Poor**	**Severe**
Delamination/Spall/Patched Area (1080)	None.	Delaminated. Spall 1 in. or less deep or 6 in. or less in diameter. Patched area that is sound.	Spall greater than 1 in. deep or greater than 6 in. diameter. Patched area that is unsound or showing distress. Does not warrant structural review.	The condition warrants a structural review to determine the effect on strength or serviceability of the element or bridge; OR a structural review has been completed and the defects impact strength or serviceability of the element or bridge.
Exposed Rebar (1090)	None.	Present without measurable section loss.	Present with measurable section loss but does not warrant structural review.	
Efflorescence/Rust Staining (1120)	None.	Surface white without build-up or leaching without rust staining.	Heavy build-up with rust staining.	
Cracking (RC) (1130)	Insignificant cracks or moderate width cracks that have been sealed.	Unsealed moderate width cracks or unsealed moderate pattern (map) cracking.	Wide cracks or heavy pattern (map) cracking.	
Abrasion/Wear (PSC/RC) (1190)	No abrasion or wearing.	Abrasion or wearing has exposed coarse aggregate but the aggregate remains secure in the concrete.	Coarse aggregate is loose or has popped out of the concrete matrix due to abrasion or wear.	

2.2.2 Superstructure Elements

Superstructure elements such as girders are considered part of the main load-bearing components of a bridge and are inspected for cracks, corrosion, and loss of protective coating. **Table 2.2** presents the AASHTO condition states for prestressed concrete girders and **Table 2.3** presents such definitions for steel girders.

Table 2.2 – AASHTO Condition States for Prestressed Concrete Girders (AASHTO MBEI, 2019)

Defects	CS 1	CS 2	CS 3	CS 4
	Good	Fair	Poor	Severe
Delamination/Spalls/ Patch Areas (1080)	None.	Delaminated. Spall 1 in. or less deep or 6 in. or less in diameter. Patched area that is sound.	Spall greater than 1 in. deep or greater than 6 in. diameter. Patched area that is unsound or showing distress. Does not warrant structural review.	The condition warrants a structural review to determine the effect on strength or serviceability of the element or bridge; OR a structural review has been completed and the defects impact strength or serviceability of the element or bridge.
Exposed Rebar (1090)	None.	Present without measurable section loss.	Present with measurable section loss but does not warrant structural review.	
Exposed Prestressing (1100)	None.	Present without section loss.	Present with section loss but does not warrant structural review.	
Efflorescence/Rust Staining (1120)	None.	Surface white without build-up or leaching without rust staining.	Heavy build-up with rust staining.	
Cracking (PSC) (1110)	Insignificant cracks or moderate-width cracks that have been sealed.	Unsealed moderate width cracks or unsealed moderate pattern (map) cracking.	Wide cracks or heavy pattern (map) cracking.	
Abrasion/Wear (PSC/RC) (1190)	No abrasion or wearing.	Abrasion or wearing has exposed coarse aggregate but the aggregate remains secure in the concrete.	Coarse aggregate is loose or has popped out of the concrete matrix due to abrasion or wear.	
Distortion (1900)	None.	Distortion does not require mitigation or mitigated distortion.	Distortion that requires mitigation that has not been addressed but does not warrant structural review.	
Settlement (4000)	None.	Exists within tolerable limits or no observed structural distress.	Exceeds tolerable limits but does not warrant structural review.	
Scour (6000)	None.	Exists within tolerable limits or has been arrested with effective countermeasures.	Exceeds tolerable limits but is less than the critical limits determined by scour evaluation and does not warrant structural review.	
Damage (7000)	Not applicable.	The element has impact damage. The specific damage caused by the impact has been captured in CS 2 under the appropriate material defect entry.	The element has impact damage. The specific damage caused by the impact has been captured in CS 3 under the appropriate material defect entry.	The element has impact damage. The specific damage caused by the impact has been captured in CS 4 under the appropriate material defect entry.

Table 2.3 – AASHTO Condition States for Steel Girders (AASHTO MBEI, 2019)

Defects	CS 1	CS 2	CS 3	CS 4
	Good	Fair	Poor	Severe
Corrosion (1000)	None.	Freckled rust. Corrosion of the steel has initiated.	Section loss is evident, or pack rust is present but does not warrant structural review.	The condition warrants a structural review to determine the effect on strength or serviceability of the element or bridge; OR a structural review has been completed and the defects impact strength or serviceability of the element or bridge.
Cracking (1010)	None.	Crack that has self-arrested or has been arrested with effective arrest holes, doubling plates, or similar.	Identified crack that is not arrested but does not warrant structural review.	
Connection (1020)	Connection is in place and functioning as intended.	Loose fasteners or pack rust without distortion is present but the connection is in place and functioning as intended.	Missing bolts, rivets, or fasteners; broken welds; or pack rust with distortion but does not warrant a structural review.	
Distortion (1900)	None.	Distortion not requiring mitigation or mitigated distortion.	Distortion that requires mitigation that has not been addressed but does not warrant structural review.	
Settlement (4000)	None.	Exists within tolerable limits or arrested with no observed structural distress.	Exceeds tolerable limits but does not warrant structural review.	
Scour (6000)	None.	Exists within tolerable limits or has been arrested with effective countermeasures.	Exceeds tolerable limits but is less than the critical limits determined by scour evaluation and does not warrant structural review.	
Damage (7000)	Not applicable.	The element has impact damage. The specific damage caused by the impact has been captured in CS 2 under the appropriate material defect entry.	The element has impact damage. The specific damage caused by the impact has been captured in CS 3 under the appropriate material defect entry.	The element has impact damage. The specific damage caused by the impact has been captured in CS 4 under the appropriate material defect entry.

2.2.3 Substructure Elements

Substructure elements include columns, abutments, joints, and bearings. These elements support the structure between spans, connect multiple spans, and allow controlled movement and load transfer between connected components and both ends of the bridge.

2.3 Nondestructive Evaluation Techniques for Bridge Decks

The structural integrity and longevity of bridge decks are critical for transportation agencies. To maintain and ensure their safety and without causing any damage, NDE techniques are employed. These methods allow for the assessment of concrete quality, detection of internal defects, and evaluation of corrosion in steel reinforcements. NDE techniques, such as chain drag, half-cell potential, impact echo, and electrical resistivity, have become invaluable tools in bridge maintenance, offering precise, reliable, and efficient means for early defect detection and planning preventive maintenance. This section explores the principles, advancements, and applications of NDE methods in bridge deck inspections.

2.3.1 Chain Drag

Chain drag method is the simplest yet effective technique for detecting delamination in concrete elements in which a steel chain is dragged on concrete (**Figure. 2.1**). The areas with a different sound compared with that of a solid concrete are usually damaged and delaminated.

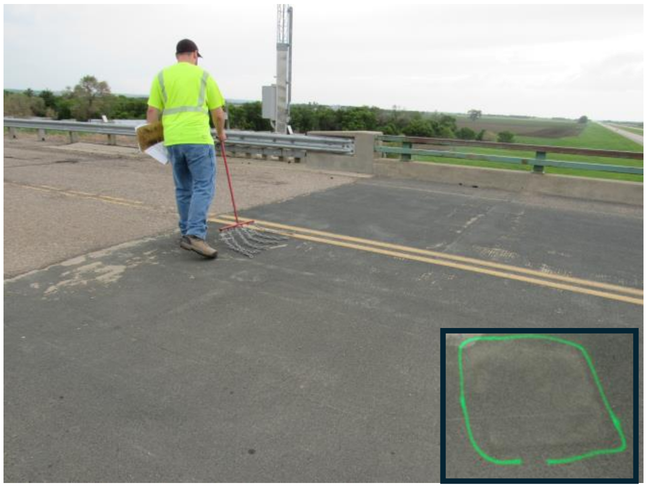


Figure 2.1 – Chain Drag for Delamination Detection in Concrete Decks

2.3.2 Half-Cell Potential (HCP)

The half-cell potential method is widely used for detecting corrosion in reinforced concrete structures. In HCP method, an electrode is attached to steel bars and another electrode is held by the inspector. The electrodes are placed at different intervals on the bridge deck. The potential difference between the electrodes varies based on the presence of corrosion in reinforcing steel bars in concrete (Gucunski and Nazarian, 2010). Using the collected data a map is generated to visually present the damage state of the entire bridge deck. ASTM C876-22b (2022) is utilized to perform HCP. Data processing usually needs expert knowledge. Recent advancements have focused on integrating this technique with other methods to enhance diagnostic accuracy.

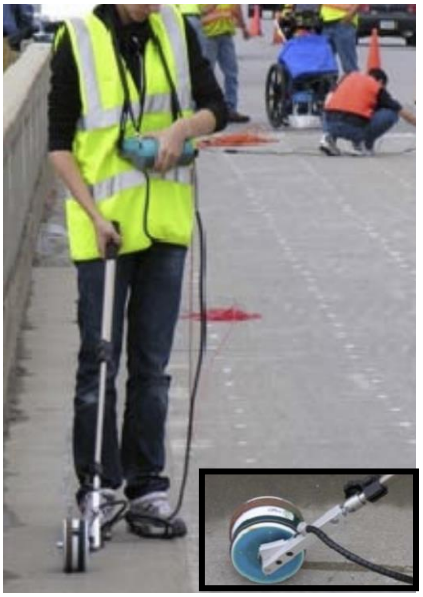


Figure 2.2 – Half-Cell Potential for Delamination Detection in Concrete Decks (Gucunski & Nazarian, 2010)

2.3.3 Impact Echo (IE)

Impact echo method utilizes acoustic waves to assess the condition of the bridge deck and identifies delamination in concrete. P-waves are generated which travel through the concrete. If they encounter different materials or air pockets, they reflect energy which is received by the device. Recent innovations include the use of advanced signal processing algorithms and machine learning to better interpret results. Portable and user-friendly devices are now available, allowing for more frequent and widespread use in the field.

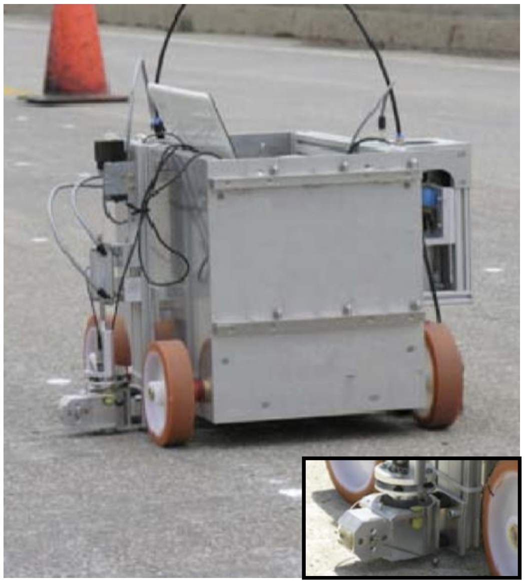


Figure 2.3 – Impact Echo for Delamination Detection in Concrete Decks (Gucunski & Nazarian, 2010)

2.3.4 Electric Resistivity (ER)

Electrical resistivity measurement is used to assess the moisture content and corrosion in concrete. The ingression of impurities and presence of air void in concrete enhances the electric resistance of concrete. The presence of voids filled with moisture lowers the concrete electric resistance (Khudhair and Gucunski, 2023). Modern trends emphasize the development of non-invasive, high-resolution resistivity mapping techniques. These advancements provide detailed insights into the condition of bridge decks, aiding in preventive maintenance.

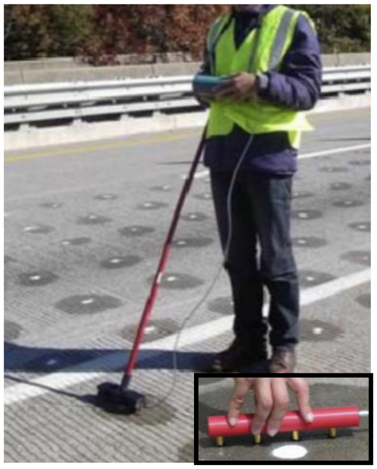


Figure 2.4 – Electric Resistivity for Delamination Detection in Concrete Decks (Gucunski & Nazarian, 2010)

2.4 Computer Vision for Bridge Element Inspection

With a widespread use of computer vision techniques, the expert communities are exploring its application in bridge inspection tasks. Matsumoto (2014) used line sensor camera and infrared thermography to detect bridge deck defects and delamination. The framework (**Fig. 2.5**) for bridge inspection and repair consists of three phases. In Phase 1, the deck is scanned using video imagery. Phase 2 involves prioritizing the areas of the deck that are distressed, determining their condition state according to the National Bridge Inventory (NBI) scale of 1 to 4, and planning necessary repairs or rehabilitation. In Phase 3, selected spans of the bridge are scanned from underneath to further assess and address the identified issues.

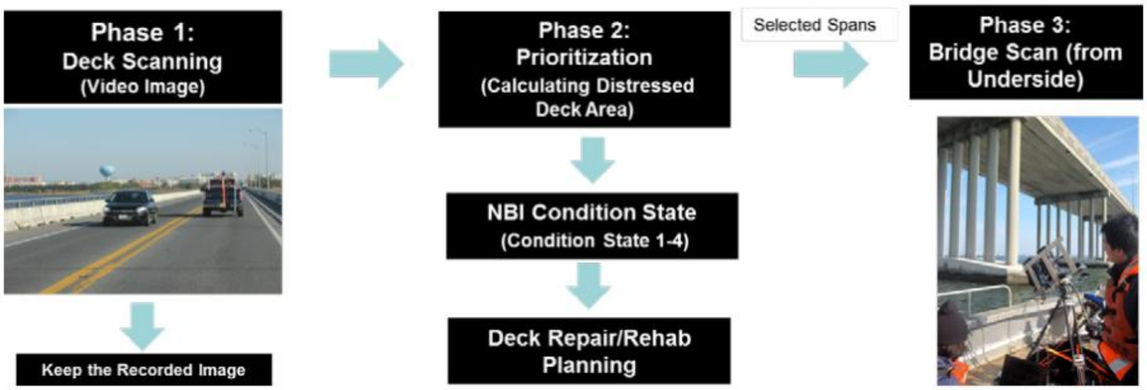
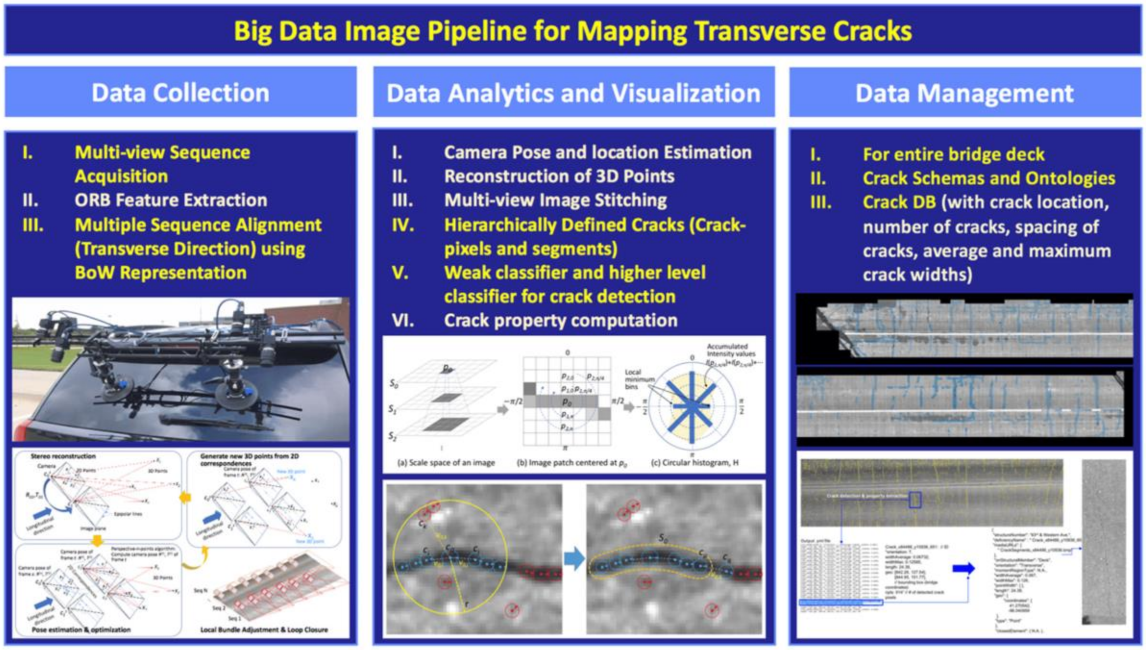


Figure 2.5 – Deck Damage Detection Framework Using Video and Infrared Thermography (Matsumoto et al., 2014)

Won and Sim (2020) detected transverse cracks on bridge decks using optical sensors. They used a three-dimensional (3D) (including height, width, and depth information) computer vision-based localization and mapping technique by utilizing stereovision setup. The proposed framework (**Fig. 2.6**) performs crack detection and localization in bridge deck inspections by employing several innovative techniques. It utilizes multi-view image sequences with a Bag-of-Words representation, eliminating the need for traditional Global Positioning System (GPS) or odometer data for localization. For crack detection, the system uses crack hierarchy and circular histograms instead of conventional methods that rely on-line segments and intensities. Additionally, it creates a comprehensive transverse crack database based on bespoke ontologies and schemas, covering the entire bridge deck. These advancements aimed at faster deployment and achieving higher accuracy in large-scale crack mapping and measurement. They collected a dataset of 2,638 image patches, consisting of 1,262 from cracked areas and 1,376 from non-crack (background) areas of the bridge. Two classification approaches were applied to these patches: a conventional method and a crack pixel-based representation approach. A support vector machine (SVM) classifier with a radial basis function kernel was trained using the data, with parameters chosen empirically. Performance, evaluated via 10-fold cross-validation, showed that the conventional method achieved an accuracy of 87.2%, while the crack pixel-based approach achieved a higher accuracy of 95.5%. The results indicate that the crack pixel and segment-based methods offer a more effective description and identification of cracks compared to the conventional approach.



Big Data Image Pipeline for Mapping Transverse Cracks

Data Collection

I. Multi-view Sequence Acquisition
II. ORB Feature Extraction
III. Multiple Sequence Alignment (Transverse Direction) using BoW Representation

Data Analytics and Visualization

I. Camera Pose and location Estimation
II. Reconstruction of 3D Points
III. Multi-view Image Stitching
IV. Hierarchically Defined Cracks (Crack-pixels and segments)
V. Weak classifier and higher level classifier for crack detection
VI. Crack property computation

Data Management

I. For entire bridge deck
II. Crack Schemas and Ontologies
III. Crack DB (with crack location, number of cracks, spacing of cracks, average and maximum crack widths)

Stereo reconstruction
Generate new 3D points from 2D correspondences
Pose estimation & optimization
Local Bundle Adjustment & Loop Closure
(a) Scale space of an image
(b) Image patch centered at p_0
(c) Circular histogram, H

Figure 2.6 – Computer Vision-based Localization and Mapping of Bridge Deck Cracks (Won and Sim, 2020)

Perry et al. (2020) used Structure-from-Motion (SfM) to generate 3D reconstructed model of bridges and to identify target elements. Then they used image processing techniques such as the black hat transform with the Canny edge detection to detect bridge defects. The black hat transform is a morphological operation used in image processing to highlight dark features on a bright background. Specifically, it emphasizes the portions of an image that are smaller than the structuring element used in the transform and darker than their surroundings. The authors also proposed a method to track growth of defects through image comparison (**Fig. 2.7**).

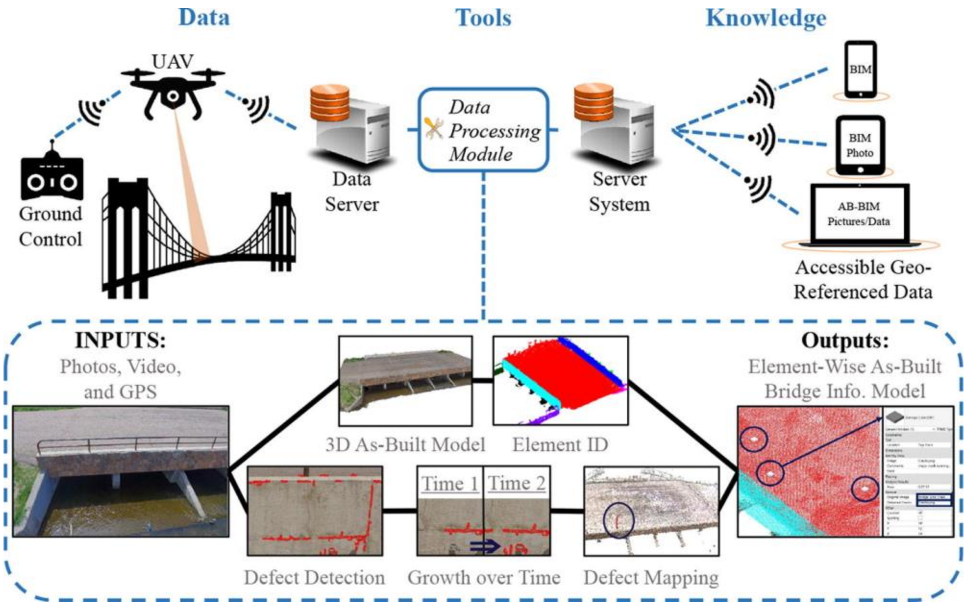


Figure 2.7 – 3D Reconstruction and Defect Detection Framework (Perry et al., 2020)

Ali et al. (2021) used computer vision techniques with a faster region-based convolutional neural network (R-CNN) damage detection framework (**Fig. 2.8**) on data collected using drones for GPS denied locations to identify structural damage related to loosened bolts and steel corrosion. The process begins with identifying the structure to be inspected, followed by deploying an autonomous unmanned aerial vehicle (UAV) equipped with a proportional-integral-derivative (PID) controller for flight stability, an ultrasonic beacon system for precise positioning, and an action camera for capturing visual data. The data was transmitted in real-time using a real-time streaming protocol. A modified Faster R-CNN machine learning model, consisting of a Region Proposal Network (RPN), object proposals, and a Fast R-CNN, was employed for image analysis. The model identifies and highlights structural damage and the ground station processes and analyzes the collected data, completing the inspection process. This integration of robotics, computer vision, and machine learning enables efficient and safe autonomous inspection of structures.

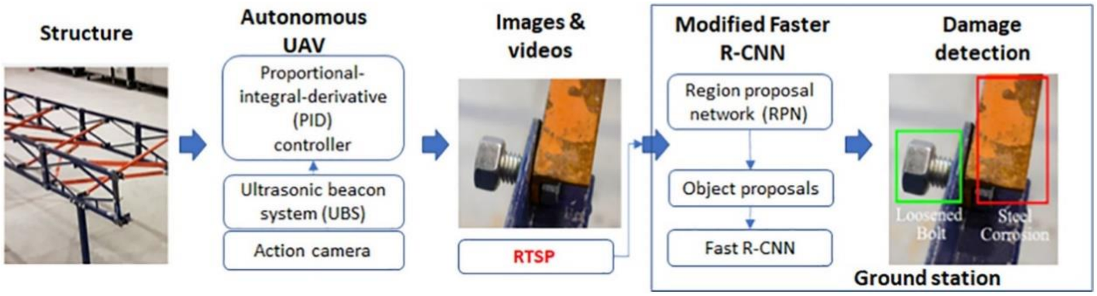


Figure 2.8 – Loosened Bolt and Corrosion Detection Framework (Ali et al., 2021)

Comparing the performance of ZF-Net (Zeiler et al. 2014) and ResNet-101 within the autonomous structural inspection framework, Ali et al. (2021) found that ResNet-101 is more accurate. ZF-Net is an improved version of AlexNet whose parameters where visually examined by reconstructing the learned feature vectors using deconvolution and some parameters (such as convolutional kernel) were tuned. For steel corrosion detection, ResNet-101 achieved an average precision (AP) of 93.31% and a mean intersection-over-union (IoU) of 90.44%, surpassing ZF-Net's AP of 90.62% and IoU of 88.72%. In detecting steel cracks, ResNet-101 scored an AP of 93.07% and an IoU of 92.24%, whereas ZF-Net obtained an AP and IoU of 85.27% each. For loosened bolts, ResNet-101 also excelled with an AP of 92.95% and an IoU of 93.79%, compared with the ZF-Net's AP of 88.64% and IoU of 88.64%. The overall mean performance demonstrates that ResNet-101 had a higher mean AP (93.11%) and mean IoU (92.16%) compared with ZF-Net, which had a mean AP of 88.17% and mean IoU of 87.54%.

Tazarv et al. (2022) used computer vision techniques for inspection and condition assessment of RC bridge columns after earthquakes to automatically determine their conditions. They used deep learning based cascaded detection of deficiencies in their framework using Mask R-CNN. This cascaded detection framework (**Fig. 2.9)** for structural damage assessment begins with Stage 1, which detects columns in images. If columns are not detected, the user is prompted to find them manually. Once columns are identified, the process moves to Stage 2 for detecting longitudinal bars, followed by Stage 3 for detecting transverse bars. If both bars are detected, the system reports damage state 4 or 5. Stage 4 involves spalling detection; if spalling is found, it results to damage state 3 or 4. If not, it continues to Stage 5 for crack detection, leading to damage state 1 or 2.

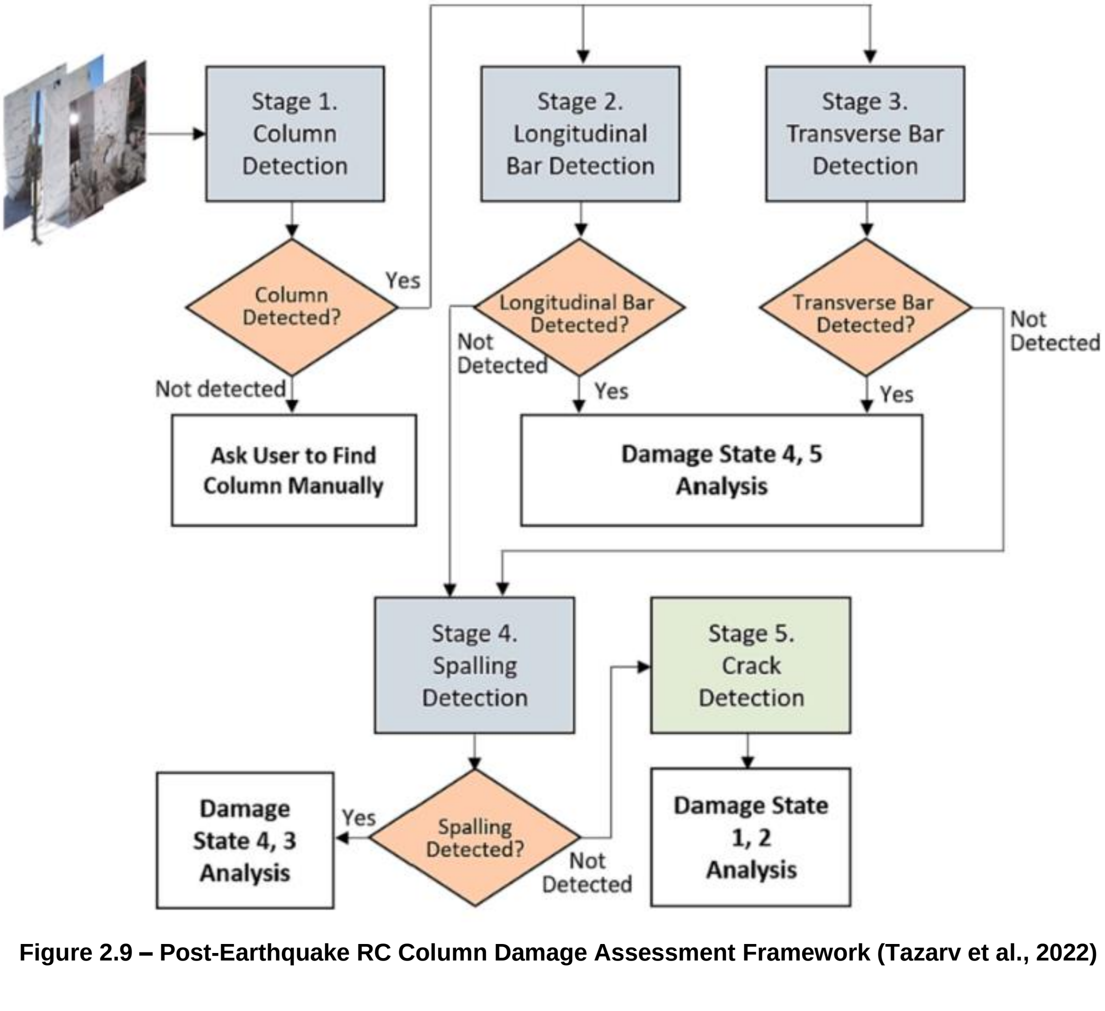


Figure 2.9 – Post-Earthquake RC Column Damage Assessment Framework (Tazarv et al., 2022)

The results of the study by Tazarv et al. (2022) indicate a high performance across all components. For columns, with 41 instances, the localization precision was 90.13% and its recall was 90.91%, while the segmentation precision was 88.90% and the recall was 89.23%. Spalled areas, with 72 instances, showed a localization precision of 95.28% and a recall of 95.88%, with a segmentation precision at 93.97% and a recall at 88.71%. Transverse bars, with 56 instances, had a localization precision and recall of 95.27% and 95.82%, respectively, and a segmentation precision and recall of 92.71% and 93.14%, respectively. Longitudinal bars, with 31 instances, achieved a localization precision of 92.31% and a recall of 92.79%, and a segmentation precision of 91.83% and a recall of 92.17%. The average metrics across all components were 93.24% for the localization precision, 93.85% for the localization recall, 91.10% for the segmentation precision, and 90.81% for the segmentation recall, demonstrating the system's overall effectiveness in deficiency detection.

Dunphy et al. (2022) developed a multiclass damage detection framework for concrete structures (**Fig. 2.10**) using a combination of Generative Adversarial Networks (GANs) and CNNs. It begins with a multiclass image database of various concrete conditions, such as undamaged, cracked, construction joint, pitting, and spalling. The database was split into labeled and unlabeled datasets. GAN training was performed on unlabeled data, producing a trained generator and discriminator network. The trained discriminator was then used for transfer learning to initialize the weights and biases of the CNN. This CNN was further trained using the labeled data. Finally, hypertraining was applied to the fully connected layers of CNN to optimize it for image classification tasks, leading to accurate identification of different types of concrete damage.

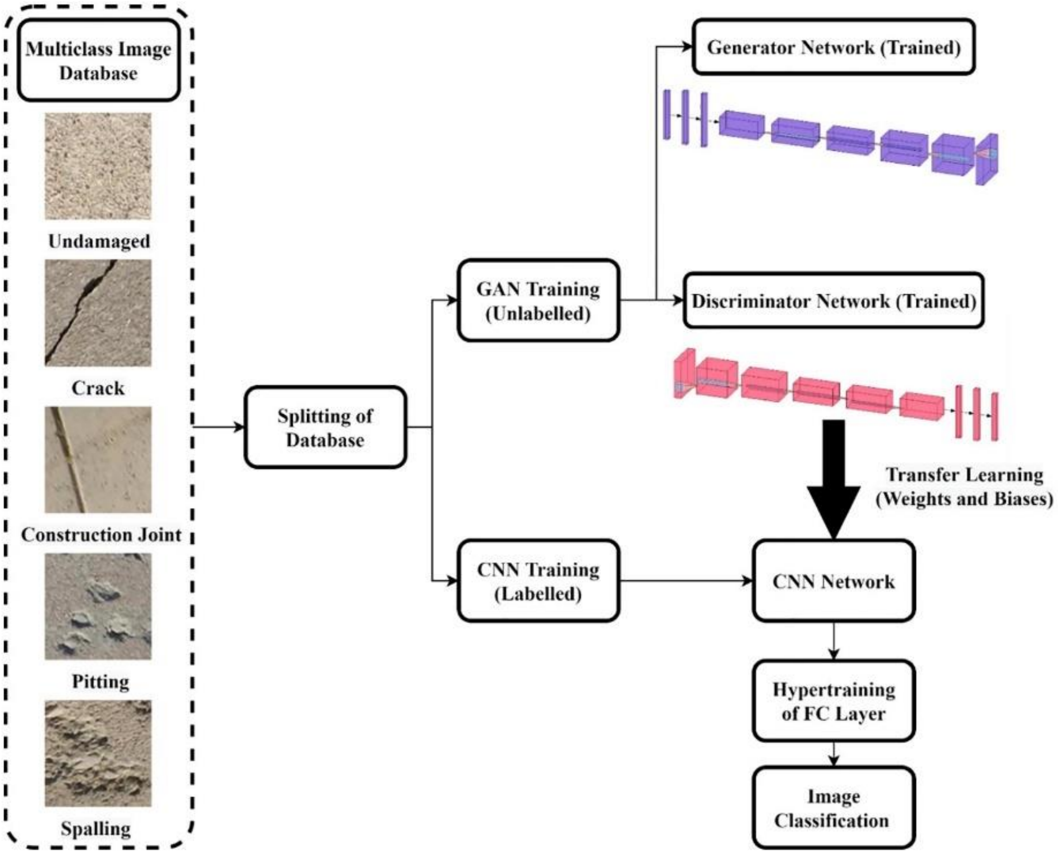

Multiclass Image Database
Undamaged
Crack
Construction Joint
Pitting
Spalling
Splitting of Database
GAN Training (Unlabelled)
CNN Training (Labelled)
Generator Network (Trained)
Discriminator Network (Trained)
Transfer Learning (Weights and Biases)
CNN Network
Hypertraining of FC Layer
Image Classification

Figure 2.10 – GAN-based Transfer Learning Damage Detection Framework (Dunphy et al., 2022)

Jiang et al. (2023) developed a framework (**Fig. 2.11**) using object detection models to identify steel bolts and nuts, as well as their condition, including corrosion and tightness. The proposed method for automatic bridge bolt inspection was a vision-based system leveraging image processing and deep learning algorithms with images captured by UAV. It consisted of three main components: bolt image acquisition, data preprocessing, and bolt damage identification. The image acquisition phase involved using a modified UAV to capture images of bolts from various angles, including the bridge's sides, bottom, and cable clamps. Data preprocessing addressed two key issues: image motion blur due to the UAV's high speed and variations in bolt pixel size caused by differing object distances. To tackle these, an image motion deblurring method using inverse filtering and optical flow was employed to correct motion blur, and an adaptive scale segmentation method with multi-scale template matching and Enhanced Super-Resolution Generative Adversarial Networks (ESRGAN) ensured uniform bolt image sizes of 240 × 240 pixels. Finally, a two-stage bolt damage identification process was used to detect and classify bolt conditions. YOLOv5 network was utilized for bolt detection and segmentation, while EfficientNet classified the bolts into categories such as normal, corroded, or loose. This method showed a high accuracy. For blurred images, it achieved an accuracy of 98.6% with 102 true positives (TP), 36 true negatives (TN), 2 false positives (FP), and no false negatives (FN). For non-blurred images, it attained a 99.6% accuracy with 169 TP, 96 TN, 1 FP, and no FN. The second method, combining preprocessing with a single network, had a lower accuracy of 82.9% for the blurred images (91 TP, 25 TN, 15 FP, 9 FN) and 88.4% for the non-blurred images (150 TP, 86 TN, 19 FP, 11 FN). The third method, using only a single network, had the lowest accuracy, 65.5% for the blurred images (79 TP, 18 TN, 29 FP, 22 FN) and 83.5% for the non-blurred images (143 TP, 79 TN, 27 FP, 17 FN).

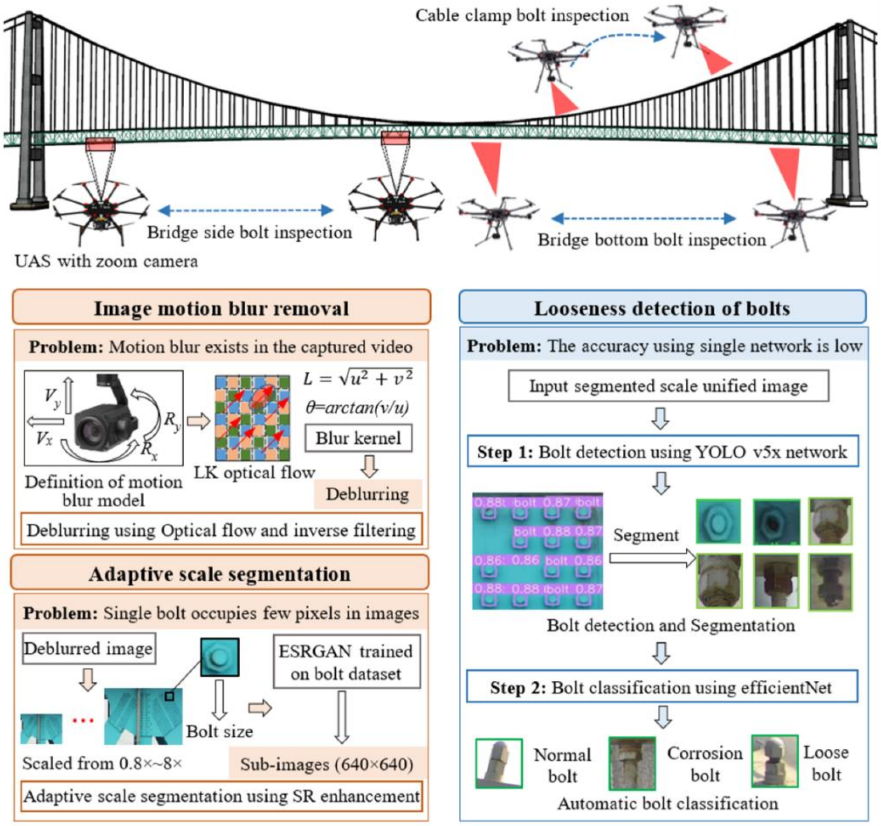

Cable clamp bolt inspection
Bridge side bolt inspection
Bridge bottom bolt inspection
UAS with zoom camera
Image motion blur removal
Problem: Motion blur exists in the captured video
V_y
V_x
R_y
R_x
$L = \sqrt{u^2 + v^2}$
θ=arctan(v/u)
Blur kernel
Definition of motion blur model
LK optical flow
Deblurring
Deblurring using Optical flow and inverse filtering
Adaptive scale segmentation
Problem: Single bolt occupies few pixels in images
Deblurred image
ESRGAN trained on bolt dataset
Bolt size
Scaled from 0.8×~8×
Sub-images (640×640)
Adaptive scale segmentation using SR enhancement
Looseness detection of bolts
Problem: The accuracy using single network is low
Input segmented scale unified image
Step 1: Bolt detection using YOLO v5x network
Segment
Bolt detection and Segmentation
Step 2: Bolt classification using efficientNet
Normal bolt
Corrosion bolt
Loose bolt
Automatic bolt classification

Figure 2.11 – Bolt Detection and Damage Identification Framework (Jiang et al., 2023)

Jiang et al. (2023) proposed a three-stage framework for bridge condition assessment and damage localization using UAV collected data as shown in **Fig. 2.12**. The authors used the YOLOv3 object detection network to detect several types of damage such as cracks, corrosion, and spalling. During data collection, the authors also used visual simultaneous localization and mapping (VSLAM) techniques to localize each damage on bridges by retrieving camera pose with respect to reference world coordinates (**Fig. 2.13**). To achieve this, the authors used stereo-inertial SLAM that used VSLAM with Inertial Measurement Unit (IMU) sensor fusion to increase the robustness of the system. Integration of IMU with VSLAM made the process more robust because of a better estimation of the camera poses for each image.

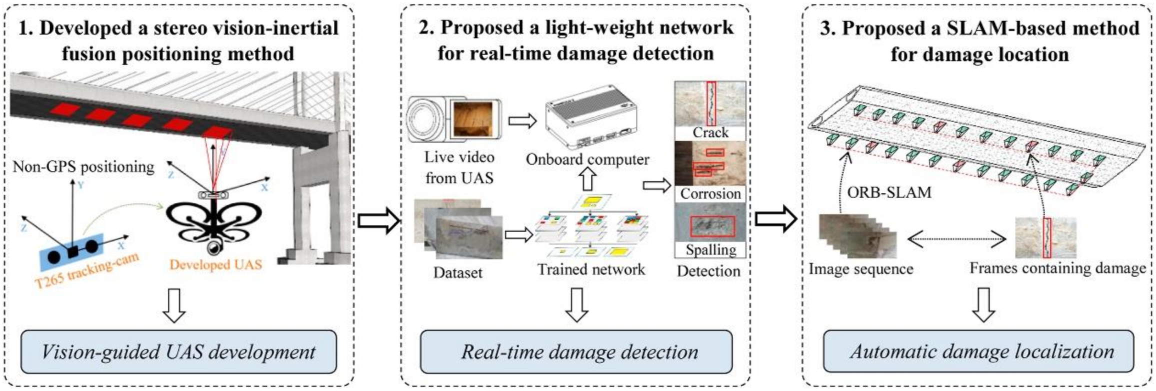

1. Developed a stereo vision-inertial fusion positioning method
Non-GPS positioning
T265 tracking-cam
Developed UAS
Vision-guided UAS development
2. Proposed a light-weight network for real-time damage detection
Live video from UAS
Onboard computer
Dataset
Trained network
Crack
Corrosion
Spalling
Detection
Real-time damage detection
3. Proposed a SLAM-based method for damage location
ORB-SLAM
Image sequence
Frames containing damage
Automatic damage localization

Figure 2.12 – Damage Detection and Localization Framework (Jiang et al., 2023)

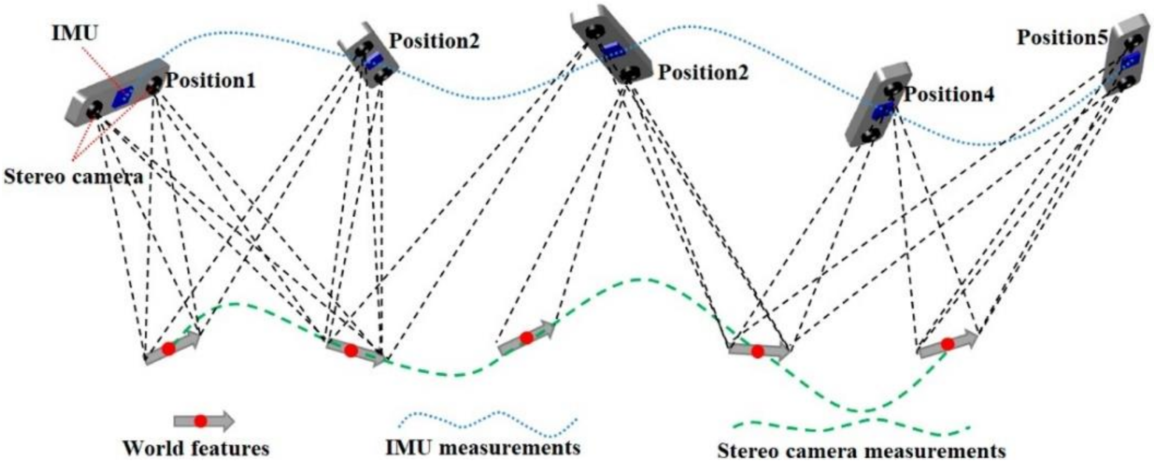


Figure 2.13 – A Damage Localization Technique by Jiang et al. (2023)

Their framework was tested on a long-span river bridge, offering a more rigorous evaluation environment than standard indoor tests due to factors like strong winds and a large detection area. Manual identification of UAV video images served as the ground truth for assessing the proposed real-time damage detection method. The results revealed a classification accuracy of 92.8% and an IoU of 80.7% between manually labeled and real-time detected damage.

Kao et al. (2023) proposed a framework to extract and quantify cracks in bridges using UAV images and deep learning-based object detection models (**Fig. 2.14**). They used planar markers to retrieve scales of images then cracks. To retrieve the crack boundaries, the authors used a local thresholding method to create a binarized image. Following that, the authors employed a crack width measurement technique in reference to the planar marker. Their YOLOv4 deep learning model was trained for crack detection and achieved an accuracy of 92%. This model performed well in identifying cracks even in images with uneven lighting and complex backgrounds. The study also demonstrated that the overall crack measurement accuracy exceeded 0.22 mm. Two edge detection methods evaluated in the study showed similar performance; however, the Canny edge detector's results varied with different thresholds, leading to greater discrepancies between measured and actual crack widths. In contrast, the morphological edge detector, which does not rely on thresholds, provided crack edges more consistent with the true dimensions.

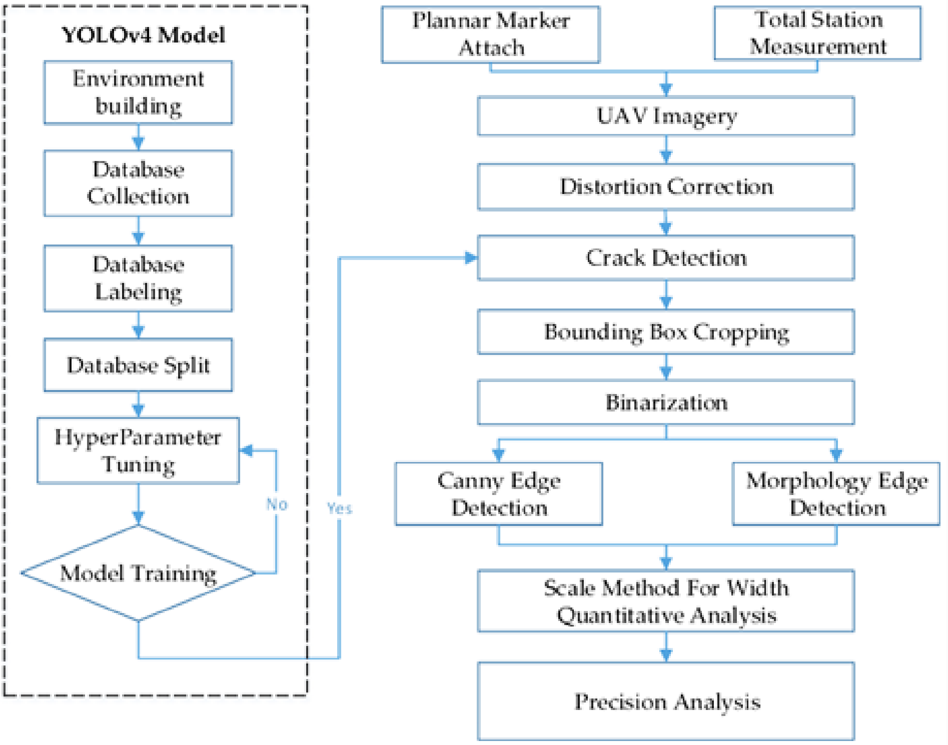


Figure 2.14 – Crack Detection and Quantification Framework (Kao et al., 2023)

Mirzazade et al. (2023) used a 3D reconstruction approach (**Fig. 2.15**) using SfM to detect and segment bridge damages based on U-Net, a popular deep learning model for semantic segmentation. SfM is a computer vision technique that reconstructs 3D structures from 2D image (only width and height) sequences taken from different viewpoints. It estimates camera positions and the 3D coordinates of scene points by analyzing the motion and structure captured in the overlapping images. They also collected Light Detection and Ranging (LiDAR) scans of structures to verify the integrity of the 3D reconstruction using SfM.

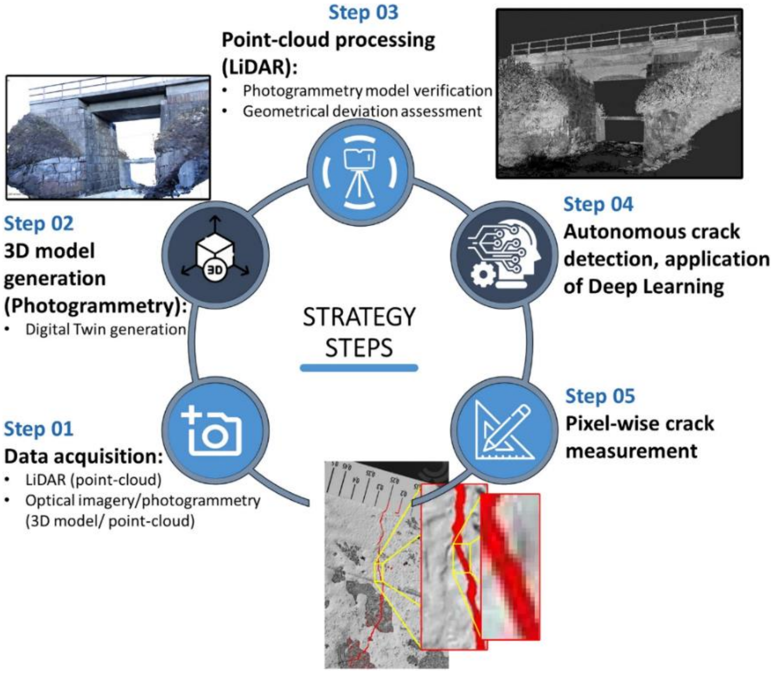

Step 03
Point-cloud processing (LiDAR):
• Photogrammetry model verification
• Geometrical deviation assessment
Step 02
3D model generation (Photogrammetry):
• Digital Twin generation
Step 04
Autonomous crack detection, application of Deep Learning
STRATEGY STEPS
Step 01
Data acquisition:
• LiDAR (point-cloud)
• Optical imagery/photogrammetry (3D model/ point-cloud)
Step 05
Pixel-wise crack measurement
3D

Figure 2.15 – 3D Reconstruction and Crack Detection Framework (Mirzazade et al., 2023)

CHAPTER 3. CONDITION STATES FOR SELECT BRIDGE ELEMENTS

3.1 Introduction

The AASHTO Manual for Bridge Elements Inspection (MBEI, 2019) provides a four-level condition states for bridge elements. Other federal and state guidelines exist; however, most transportation agencies are moving toward using the AASHTO manual for consistency and reporting to the federal government.

As discussed in the previous chapters, this report is focused on concrete bridge decks and how delamination detection and reporting can be expediated using new technologies. In this chapter, the AASHTO specified condition states for concrete decks are presented first then the definition is modified to be quantitative best suited for computer programming.

3.2 AASHTO Condition States for Concrete Decks

Table 3.1 (the same as **Table 2.1**) presents different damage types seen for concrete bridge decks and includes their four-stage condition definitions. Except delamination, other deck defects are on the surface and visible. Nevertheless, concrete delamination is a sub-surface damage, which can only be detected using a nondestructive method or concrete removal. Delamination is a state of separation of layers within concrete and may lead to a loss of structural integrity for bridge decks and an increase in maintenance costs.

Table 3.1 – AASHTO Condition States for Concrete Decks (AASHTO MBEI, 2019)

Defects	CS 1	CS 2	CS 3	CS 4
	Good	**Fair**	**Poor**	**Severe**
Delamination/Spall/Patched Area (1080)	None.	Delaminated. Spall 1in. or less deep or 6in. or less in diameter. Patched area that is sound.	Spall greater than 1in. deep or greater than 6in. diameter. Patched area that is unsound or showing distress. Does not warrant structural review.	The condition warrants a structural review to determine the effect on strength or serviceability of the element or bridge; OR a structural review has been completed and the defects impact strength or serviceability of the element or bridge.
Exposed Rebar (1090)	None.	Present without measurable section loss.	Present with measurable section loss but does not warrant structural review.	
Efflorescence/Rust Staining (1120)	None.	Surface white without build-up or leaching without rust staining.	Heavy build-up with rust staining.	
Cracking (RC) (1130)	Insignificant cracks or moderate width cracks that have been sealed.	Unsealed moderate width cracks or unsealed moderate pattern (map) cracking.	Wide cracks or heavy pattern (map) cracking.	
Abrasion/Wear (PSC/RC) (1190)	No abrasion or wearing.	Abrasion or wearing has exposed coarse aggregate but the aggregate remains secure in the concrete.	Coarse aggregate is loose or has popped out of the concrete matrix due to abrasion or wear.	

It is clear that the AASHTO definitions of condition states are mostly qualitative with no specific definitions for condition state 4. Alternative and measurable definitions are required, which can be used in computer programming.

3.3 Proposed Condition States for Concrete Decks

As discussed above, any defects that are expected to be found using computer tools need to be quantitative not qualitative. To this end, **Table 3.2** presents alternative definitions of condition states for concrete decks specifically for delamination, spalling, and patching. The new definitions are measurable and programmable. Furthermore, to minimize variations from AASHTO, the proposed definitions are kept at four levels. Very small delaminated regions are usually hard to find in practice thus a limit of 7 in^2 (45 cm^2) was proposed for CS-1, which is indeed insignificant compared with the area of the bridge deck. The proposed area for CS-2 follows that of the AASHTO manual at this level. Based on the inspection of more than 45 bridges with experienced inspectors, the research team noticed that a 12 in. (30 cm) by 12 in. (30 cm) area was the smallest region inspectors annotated as delamination. With a 30% reduction, the research team proposed a cap of 110 in^2 (710 cm^2) in CS-3. Any areas greater than that is marked as CS-4. It is clear that these limits are subjective; however, the user can adjust them as needed.

Furthermore, comparing with the AASHTO definitions, some parameters, such as the depth of defects, were removed in the new definitions since they are challenging to estimate using computer vision techniques. For example, there are usually large errors in the direction perpendicular to the plane of view of cameras thus the depth estimate is less accurate using computer vision methods. Note that other types of defects were not quantified herein since they were beyond the scope of this project.

Table 3.2 – Proposed Condition States for Delamination, Spalling, and Patching in Concrete Decks

Defects	CS 1	CS 2	CS 3	CS 4
	Good	**Fair**	**Poor**	**Severe**
Delamination, Spall, or Patched Area (1080)	Defected area is less than or equal to 7 in^2 (45 cm^2)	Defected area is greater than 7 in^2 (45 cm^2) but less than 30 in^2 (195 cm^2)	Defected area is greater than 30 in^2 (195 cm^2) but less than 110 in^2 (710 cm^2)	Defected area is greater than 110 in^2 (710 cm^2)

CHAPTER 4. BRIDGE DECK INSPECTION DATABASE

4.1 Introduction

Large data is needed for a successful training of neural networks that are sufficiently accurate in performing the desired tasks. For example, if crack detection in images is desired, thousands and sometimes millions of images with and without cracks are required. Most civil engineering related deep learning studies do not have access to such large data; however, data augmentation approaches and pre-trained networks can be utilized to overcome such data scarcity. A handful of datasets targeting specific bridge elements have been compiled in some recent studies. These datasets include mono-modality, such as red-green-blue (RGB) images or thermal images, and multimodality such as RGB and thermal images combined. In this chapter, a few key past studies collecting bridge related data are reviewed first. Subsequently, the data collection of the present study is discussed.

4.2 Existing Image Databases

There are several existing datasets specific to bridge health assessment including bridge elements such as columns, decks, and overlays. Specifically designed for crack detection on concrete surfaces, the CrackForest dataset by Shi et al (2016) includes annotated images of cracks on various concrete structures, including bridges. A derivative of the Common Objects in Context (COCO) dataset by Bianchi et al. (2020), was tailored for bridge inspection tasks, which includes annotated images of bridges highlighting various structural elements and potential defects. Dorafshan et al. (2021) provided a dataset containing images of concrete surfaces, including bridges, annotated for different types of defects such as cracks and spalling. The Road Damage Detection (RDD) dataset by Arya et al. (2022) was for the road surface but included images of bridge decks. The dataset is useful for detecting surface defects such as potholes and cracks. **Table 4.1** presents a summary of past studies related to image datasets.

Table 4.1 – Summary of Past Studies on Bridge Related Image Datasets

Reference	Dataset	Categories	Data Type	Size	Application
Shi et al. (2016)	CrackForest	Cracks	RGB Image	10,000	Detect and analyze cracks on the road surface
Zhang et al. (2016)	CCIC	Cracks	RGB Image	40,000	Detect cracks on concrete surfaces
Rubio et al. (2019)	--	Exposed Rebar, Delamination	RGB Images	734	Detect and quantify exposed rebars and delamination
Zou et al. (2019)	DeepCrack	Cracks	RGB Image	35,100	Detect fine cracks on various road surfaces
Bianchi et al. (2020)	COCO Bridge Damage	Bearing, Cover-plates, Gusset-plates, Stiffeners	RGB Image	774	Detect bearing, cover-plate, gusset-plate, stiffeners and evaluate their conditions
Dorafsan et al. (2021)	SDNET2021	Delamination	Infrared Thermography, Impact Echo, Ground Penetrating Radar	1,936	Detect delamination and reinforcement corrosion
Arya et al. (2022)	RDD2022	Cracks and Potholes	RGB Image	47,420	Detect and classify road damage types, such as cracks, potholes, and surface wear
Tazarv et al. (2022)	RC Column Damage	Cracks, Spalling, Exposed Rebars	RGB Image	3,036	Detect post-earthquake damages of RC bridge columns and determine their damage states

The available datasets do not explicitly include delamination data, which is crucial for training and validating models. Furthermore, the current datasets lack thermal image data, which may be used for identifying internal defects like delamination. Thermal imaging provides a unique perspective by highlighting temperature variations that may indicate underlying structural issues. For instance, delaminated areas often exhibit different thermal properties compared to intact regions, making thermal images a powerful tool for nondestructive evaluation. Further, the existing datasets do not include annotations from experts, such as the Department of Transportation (DOT) inspectors, for delamination to be used as ground truth. Expert annotations are essential for creating reliable and accurate training datasets. These annotations serve as the benchmark for evaluating the performance of models and ensuring that they are learning to identify defects correctly.

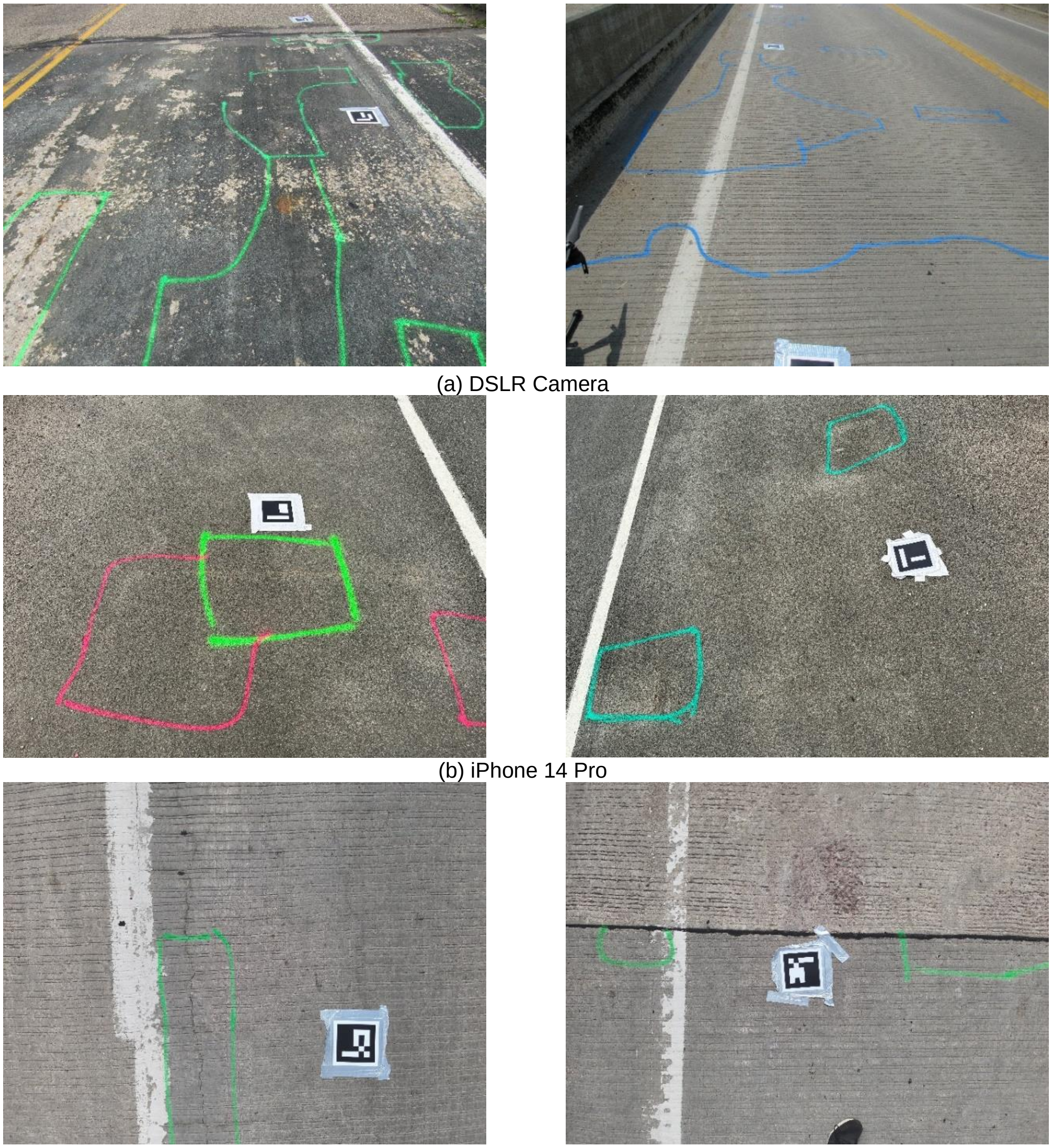
In summary, collecting new data was essential in the present study due to the limitations of the existing datasets. The current datasets do not explicitly include delamination data, lack thermal data, and do not provide expert annotations for delamination. By addressing these gaps, we can create a more comprehensive and representative dataset that significantly improves the training and validation of models. This, in turn, enhances the accuracy, reliability, and robustness of detection systems, ensuring that they are better equipped to handle real-world scenarios and critical applications.

4.3 New Image Dataset for Concrete Bridge Decks

To collect new data for bridge decks, the research team accompanied with inspectors from the South Dakota Department of Transportation (SDDOT) inspected more than 45 bridges in South Dakota in the summers of 2023 and 2024. The inspectors used the chain drag method, a nondestructive technique, to evaluate the bridge decks for delamination and annotated them with paints. The research team then collected RGB images using digital single-lens reflex (DSLR) cameras, iPhone 14 Pro, and DJI drones (Matrice 200, and Mavic 3T) before and after the annotations. Furthermore, PolyCam, a mobile application utilizing iPhone LiDAR camera, was used to scan the bride decks to record three-dimensional (3D) meshes of each deck.

RGB images taken by iPhone 14 Pro had a resolution of 8064 by 6048 pixels. Further, the RGB images obtain by the DJI Mavic 3T and the DSLR camera had a resolution of 4000 by 3000 and 4608 by 3456 pixels, respectively. The images taken by the drones were taken approximately at a height of 3 meters from the deck surface. For the images taken by the other cameras, the height varied between 1 to 2 meters. **Figure. 4.1** shows a few samples from each of the data collection devices. In total, 1918 RGB images were collected from 18 bridges that had delamination detected and annotated by the inspector using chain drag. Furthermore, the deck of 18 bridges with delamination were LiDAR scanned and data was processed per bridge.

PolyCam utilizes the integration of LiDAR data and photogrammetry to enhance the precision and detail of 3D models generated using iPhones that are equipped with LiDAR sensors, such as the iPhone 12 Pro and subsequent models. The process initiates with the iPhone’s LiDAR sensor, which emits light pulses to measure the time taken for the reflections to return from various objects, thereby generating a highly accurate depth map of the scene. Concurrently, users capture multiple images or videos of the object or scene from various angles using the iPhone’s camera. Photogrammetry algorithms subsequently analyze these visual inputs to discern the structural, textural, and detailed attributes of the scene. The integrated data then undergoes further processing to produce a coherent 3D model of the target object.



(a) DSLR Camera

(b) iPhone 14 Pro

(c) DJI Mavic 3T

Figure 4.1 – Sample RGB Images from Inspector Annotation Dataset

In the following chapter, a method to generate a 2D map of the whole bridge deck using the abovementioned 3D maps is discussed. Furthermore, it is discussed how the RGB images taken by various devices are used to train and evaluate semantic segmentation models to quantify the inspector annotated assessments for report generation.

CHAPTER 5. COMPUTER VISION AIDED BRIDGE INSPECTION AND REPORTING SOFTWARE

5.1 Introduction

Computer vision based civil infrastructure inspection tools have been developed over the years to automate bridge inspection tasks such as detection of cracks, spalling, and bar corrosion (Koch et. al, 2015). Some studies were focused on a specific damage type (e.g., crack detection of bridge components, Prasanna et al., 2012) and a few explored multiple deficiencies (e.g., Oh et al., 2009; Lee et al., 2014) to produce more general inspection tools. The recent advancement in image processing and deep learning provides new pathways to design and deploy cost effective and efficient inspection tools than conventional inspections, which are mostly manual. Especially, solving computer vision tasks using deep convolutional neural networks (DCNN) has gained superior traction in recent years.

5.2 Proposed Computer Vision Tools for Delamination Detection

In the present project, the research team has developed a computer vision based bridge deficiency detection, quantification, and condition state estimation software that is focused on delamination in concrete bridge decks. **Figure. 5.1** shows a high-level architecture of the software. The software has several interconnected modules to allow data collection and analyses using different devices and methods. Images can be acquired using different sources (digital single-lens reflex cameras (DSLR), iPhone cameras, or cameras on unmanned aerial vehicles (UAVs)). Further, 3D maps of bridge decks are obtained from the iPhone's LiDAR camera. Subsequently, the image dataset (as discussed in **Ch. 4**) was used in deep learning models. For each bridge, a point-cloud map is obtained using an iPhone and then run through the software to generate a 2D map of the deck. The software then detects local damages (delamination) using a trained neural network and pinpoints the damage on the global map of the deck. The findings are then summarized. In the following sections, a brief introduction of computer vision tasks is presented first, then the components of the software are discussed.

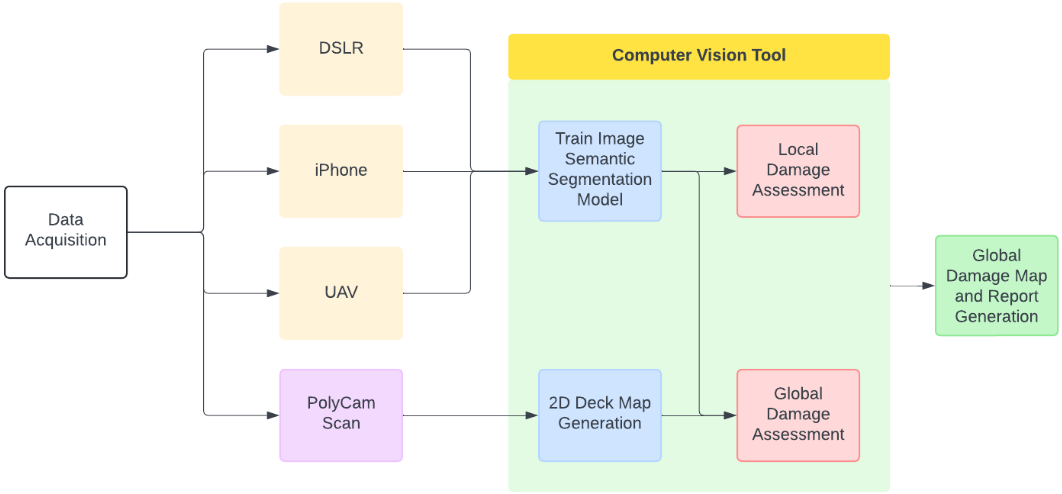


Figure 5.1 – Proposed Computer Vision Framework for Delamination Detection of Bridge Decks

5.3 Computer Vision Background

5.3.1 Stereovision and World Mapping

Stereovision, also known as stereoscopic vision or stereo imaging, is a technique used in computer vision and robotics to perceive depth by utilizing two or more images of the same scene taken from different viewpoints. Stereovision enables machines to reconstruct the 3D structure of their environment. **Figure 5.2** shows a typical stereo system. The fundamental principle behind stereovision is triangulation. When two cameras, separated by a known baseline distance, capture images of the same scene, corresponding points in the images can be identified. The disparity, or difference in position, of these projected 2D points between the images is inversely proportional to their distance from the cameras. By calculating the disparity, the depth information can be extracted, allowing for the reconstruction of a 2.5D (depth from a single direction) scene.

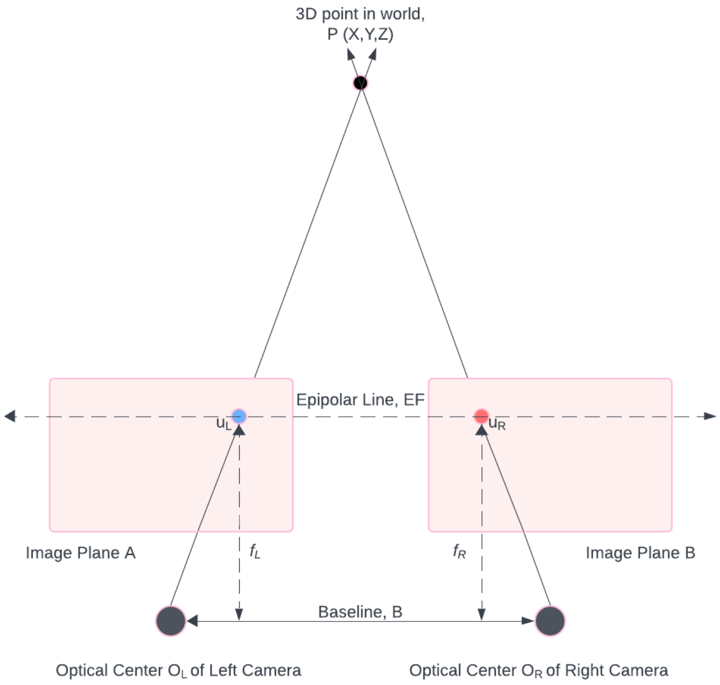


Figure 5.2 – Depth Estimation from Stereo Camera Setup

In the above figure (**Fig. 5.2**), two cameras are perfectly parallel, meaning that there is no relative rotation between the cameras. The baseline of, B, between the cameras determines the distance between the optical centers of the system. f_A and f_B are the focal lengths of the left and right cameras, respectively. Both the cameras are looking at a real-world point, P, with a coordinate of (X, Y, Z). The projected image coordinate of P in the left camera is u_L and in the right camera is u_R. The projected image coordinates in the image plan can be estimated using:

$$u_L = f_L \frac{X}{Z} \tag{5.1}$$

$$u_R = f_R \frac{X - B}{Z} \tag{5.2}$$

The distance between the two projected points is known as disparity and is given by,

$$disparity = u_L - u_R = \left(\frac{f_L + f_R}{2}\right)\frac{B}{Z} \tag{5.3}$$

The depth information of the real-world point, P, which is the distance between the center of baseline and P is modeled by,

$$depth = \left(\frac{f_L + f_R}{2}\right)\frac{B}{disparity} \tag{5.4}$$

The actual process of stereovision setup requires more complex estimation because the camera lenses always induce distortion to the acquired image. Before computing the depth information, it is required to undistort both the images using distortion coefficients and intrinsic parameters of the cameras. The process of estimating the camera matrix and the distortion coefficients is known as camera calibration. The projection of a real-world point to the image plane (both homogenous coordinates) is represented in terms of a linear transformation using the camera matrix as:

$$\begin{bmatrix} wu \\ wv \\ w \end{bmatrix} = K\ [R \mid T] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = P \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{5.5}$$

where, P is known as the camera matrix and is composed of the intrinsic and extrinsic parameters, *K* includes the intrinsic parameters, *R* is a 3-by-3 rotation matrix, and *T* is the translation vector. The intrinsic parameters contain the information about focal lengths in the X and Y directions and the coordinates of the principal point in both directions as well. The matrix is given by:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.6}$$

where, *f*x and *f*y are the focal lengths in the X and Y directions. The principal point is given by c_x and c_y.

The most common distortion type for cameras is the radial distortion. The distortion of the image coordinates in terms of both the distortion types are given by,

$$x_{radial} = x\,(1 + k_1 r^2 + k_2 r^4 + k_3\, r^6\,) \tag{5.7}$$

The most common method of estimating these parameters is through a 2D planer pattern to compute the calibration parameters. The reason this method is popular is that it simplifies the calibration process. Since the pattern is flat and lies on a single plane (with no Z-axis components), it eliminates the complexity of dealing with depth information or 3D spatial relationships. This makes it easier to compute calibration parameters, such as intrinsic camera parameters (such as focal length and optical center) and extrinsic parameters (such as the camera's position and orientation relative to the pattern). By focusing only on the 2D plane, the process becomes less computationally intensive and more straightforward, as the calibration only needs to account for distortions and errors in the 2D image plane without additional complications from depth. So, the projection equation reduces to:

$$x_i = PX_i; \qquad Z_i = 0 \tag{5.8}$$

Here, X_i is the 3D point with a coordinate of (X_i, Y_i, 0) and x_i is the corresponding 2D point in image plane (u_i, v_i).

World mapping, in the context of computer vision and robotics, refers to the process of creating a comprehensive, 3D representation of an environment. This map serves as a crucial reference for navigation, planning, and interaction within the environment. World mapping often leverages data from various sensors, including stereoscopic cameras, LiDAR, and GPS, to build an accurate and detailed map. Key components of a world mapping framework include:

- **Camera Calibration:** Calibrate both cameras to determine their intrinsic parameters (camera matrix, distortion coefficients) and the extrinsic parameters (rotation and translation between the cameras)
- **Image Rectification:** A technique to rectify the images to align them as if they were captured by a camera pair with parallel optical axes. This simplifies the correspondence problem to a 1D search along horizontal lines.
- **Disparity Map Generation:** Compute the disparity map, which measures the pixel-wise difference between the corresponding points in the left and right rectified images.
- **Point Cloud Generation:** Convert the disparity map to a point cloud representing the 3D coordinates of each pixel.
- **3D Reconstruction:** Construct a 3D mesh from the point cloud by connecting the points to form a surface.

Both stereovision and world mapping are pivotal technologies in the field of computer vision and robotics, providing machines with the ability to perceive, understand, and interact with their surroundings in three dimensions. By mimicking human depth perception and creating detailed maps of the environment, these technologies drive advancements in autonomy, navigation, and interaction across various industries, enhancing both functionality and safety.

5.3.2 Deep Convolutional Neural Networks (DCNNs)

DCNNs, often simply referred to as convolutional neural networks (CNNs), have revolutionized the field of computer vision and become the cornerstone of many modern image recognition and processing systems. DCNNs are often used as latent feature extractors that are more robust than handcrafted filters for feature extraction. Convolutional layers in deep networks can learn hierarchical features from low-level edges to high-level object parts. This multi-scale feature extraction is crucial for semantic segmentation, as it helps in identifying both small and large objects in an image. Architectures like U-Net (Ronnenberger et al., 2015) use encoder-decoder structures to capture fine-grained details. The encoder compresses the image into a lower-dimensional feature space, and the decoder reconstructs the spatial details from these features. This section provides details on the building blocks of encoders in deep convolutional networks.

The concept of convolutional networks dates to the late 1980s and early 1990s with the work of LeCun et al. (1989), who developed the LeNet architecture for handwritten digit recognition (LeCun et al., 1998). However, the true potential of CNNs was realized in 2012 when Krizhevsky et al. (2012) introduced the AlexNet architecture, which significantly outperformed previous methods in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). This breakthrough demonstrated the power of deep learning and large-scale datasets, sparking widespread interest and research in the area.

Convolutional layers are the fundamental building blocks of CNNs, designed to process grid-like data such as images. The primary function of a convolutional layer is to detect local patterns within an image, such as edges, textures, or more complex features, by applying a set of learnable filters (also known as kernels) across the input data.

Each filter in a convolutional layer slide over the input image to perform a dot product between the filter's weights and a local region of the input image. This operation is mathematically expressed as:

$$(X * W)_{i,j} = \sum_{m} \sum_{n} X_{i+m,j+n} W_{m,n} + b \tag{5.9}$$

Here, X represents the input image, W denotes the filter, i and j are the coordinates of the output feature map, m and n iterate over the filter dimensions, and b is the bias term. The result of this dot product is a single value that forms part of the output feature map, capturing the presence of the pattern encoded by the filter at that specific location.

The convolution operation is repeated as the filter moves across the entire input image, producing a 2D activation map (feature map) for each filter. The depth of the output from a convolutional layer is determined by the number of filters used, with each filter detecting different features within the image.

Pooling layers, typically placed after convolutional layers, are used to reduce the spatial dimensions of the feature maps while retaining the most important information. This process, known as downsampling or subsampling, helps to decrease the computational load, memory usage, and overfitting.

A common type of pooling operation is max pooling, which selects the maximum value within a defined window of the input feature map. The mathematical operation for max pooling can be described as:

$$Y_{i,j} = \max_{0 \le m < p, 0 \le n < q} \left(X_{s \cdot i+m, s \cdot j+n} \right) \tag{5.10}$$

where $Y_{i,j}$ is the output of the pooling layer, X is the input feature map, $p \times q$ is the size of the pooling window and s is the stride, indicating the step size with which the pooling window moves across the input feature map.

Convolutional and pooling layers work together in CNNs to progressively abstract higher-level features from the raw input image. Initially, convolutional layers capture low-level features such as edges and textures. As the data passes through multiple layers of convolutions and pooling, the network learns increasingly complex features, ultimately leading to high-level representations that are useful for tasks such as classification, detection, and segmentation.

For example, after the initial convolutional and pooling layers have processed an image of a cat, the network might detect edges and textures corresponding to fur patterns and contours. Deeper layers could then recognize parts of the cat such as ears, eyes, and whiskers. Finally, the fully connected layers use these high-level features to make a prediction about the presence of a cat in the image.

In conclusion, convolutional layers identify local patterns through learnable filter-based operations, while pooling layers reduce spatial dimensions and retain important information. Together, they enable CNNs to build hierarchical feature representations that are essential for effectively analyzing and interpreting image data.

5.3.3 Semantic Segmentation of Images

Semantic segmentation is a computer vision task that involves partitioning an image into semantically meaningful regions, with each pixel assigned a class label corresponding to the object or region it represents. This task is crucial for understanding the detailed structure and composition of images, making it essential for applications in autonomous driving, medical imaging, and scene understanding.

Semantic segmentation goes beyond traditional image classification and object detection by providing a dense prediction, where every pixel in an image is labeled. The goal is to create a precise and comprehensive map of the objects and regions within an image.

The introduction of fully convolutional networks (FCNs) by Long et al. (2015) marked a significant advancement in semantic segmentation. FCNs replace the fully connected layers of traditional CNNs with convolutional layers that produce spatial outputs, enabling end-to-end learning and prediction of segmentation masks.

In a fully convolutional network, the input image X is processed through several convolutional and pooling layers, resulting in a feature map. Let $f(X, W)$ represent the output of the network given the input X and parameters W. For semantic segmentation, the output is a dense prediction with the same spatial dimensions as the input image but with a depth corresponding to the number of classes.

Mathematically, the output feature map $f(X, W)$ can be expressed as:

$$f(X\colon W) = Conv_n\big(Pool_{n-1}(\ldots Conv_2(Pool_1(Conv_1(X_1, W_1), W_2), \ldots), W_n)\big) \tag{5.11}$$

where, $Conv_i$ and $Pool_i$ denote the convolutional and pooling layers at the *i*-th stage, respectively, and W_i represents the weights of the *i*-th convolutional layer.

To obtain pixel-level predictions, the spatial resolution of the feature maps needs to be restored to match the input image. This is achieved using upsampling techniques such as deconvolution (transposed convolution) or bilinear interpolation. The deconvolution operation can be mathematically represented as:

$$Y_{i,j} = \sum_m \sum_n X_{\lfloor \frac{i+m}{s} \rfloor, \lfloor \frac{j+n}{s} \rfloor} W_{m,n} \tag{5.12}$$

where, *Y* is the upsampled output, *X* is the input feature map, *W* is the filter, *s* is the stride, and *m* and *n* iterate over the filter dimensions.

In practice, upsampling layers are combined with skip connections from earlier layers in the network to refine the segmentation results and recover spatial details lost during downsampling. This approach is exemplified in architectures like U-Net (Ronneberger et al., 2015), which uses symmetric downsampling and upsampling paths with skip connections to merge high-resolution features from the encoder with upsampled features in the decoder.

Training a semantic segmentation model involves minimizing a loss function that measures the discrepancy between the predicted segmentation map and the ground truth. A common choice is a pixel-wise cross-entropy loss, defined as:

$$\mathcal{L}_{CE} = -\sum_{i,j}\sum_{c} t_{i,j,c}\log(p_{i,j,c}) \tag{5.13}$$

where, $t_{i,j,c}$ is the ground truth binary indicator for class c at pixel (i,j), and $p_{i,j,c}$ is the predicted probability for class c at pixel (i,j).

Another loss function used is the Dice coefficient loss, which measures the overlap between the predicted and ground truth masks:

$$\mathcal{L}_{Dice} = 1 - \frac{2\sum_{i,j} p_{i,j}t_{i,j}}{\sum_{i,j} p_{i,j}^2 + \sum_{i,j} t_{i,j}^2} \tag{5.14}$$

where, $p_{i,j}$ and $t_{i,j}$ are the predicted and ground truth binary values at pixel (i,j), respectively. In the present research, the Dice loss was used to train the models.

5.4 AI Based Delamination Detection for Bridge Decks

As discussed in the previous sections, detection of delamination using paint marks by inspectors is the focus of this report. Note inspectors can use chain drag or other methods to detect delamination. In the proposed method, the inspector needs to scan the bridge deck with an iPhone or iPad equipped with a LiDAR camera. Subsequently, the AI-based software developed in this project analyzes the point-cloud maps of bridge decks to detect paint marks (delamination). Note that other novel methods such as the use of drones and thermal imaging are under investigation by the research team and are beyond the scope of this report.

To detect delamination of concrete bridge decks using inspector annotations, deep learning models were trained on large datasets to learn the desired tasks. In general, such processes require data preparation, training on the target dataset, domain adaptation, and performance evaluation of the domain adapted model on the collected data. The model learns from the prepared data to classify each pixel and compares its predictions with benchmark data, which is called a ground truth. The outcome of the prediction determines how much the model has learned or needs to be updated and enhanced. **Figure. 5.3** shows the proposed workflow for delamination detection of concrete bridge decks using inspector marks and its quantification process, which will be briefly described next.

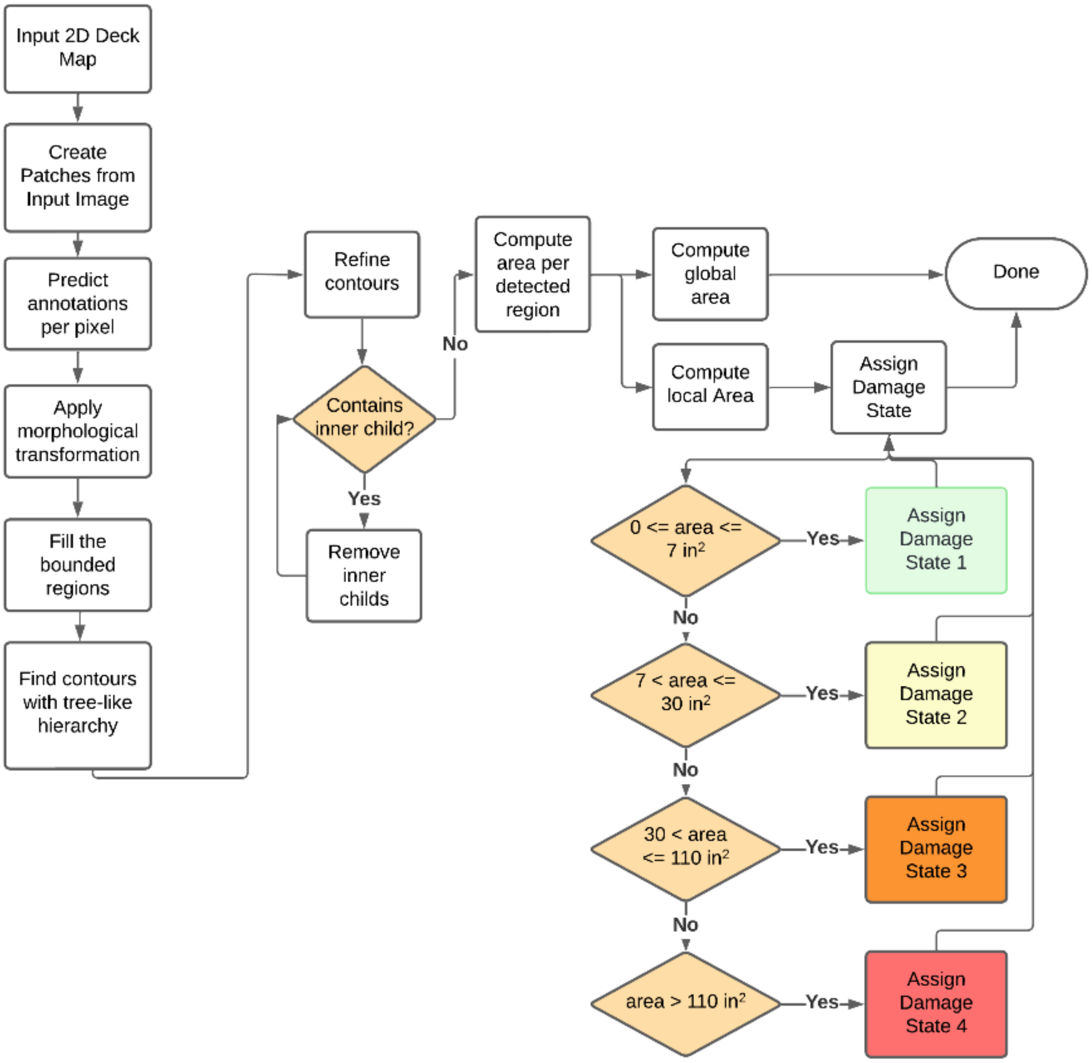
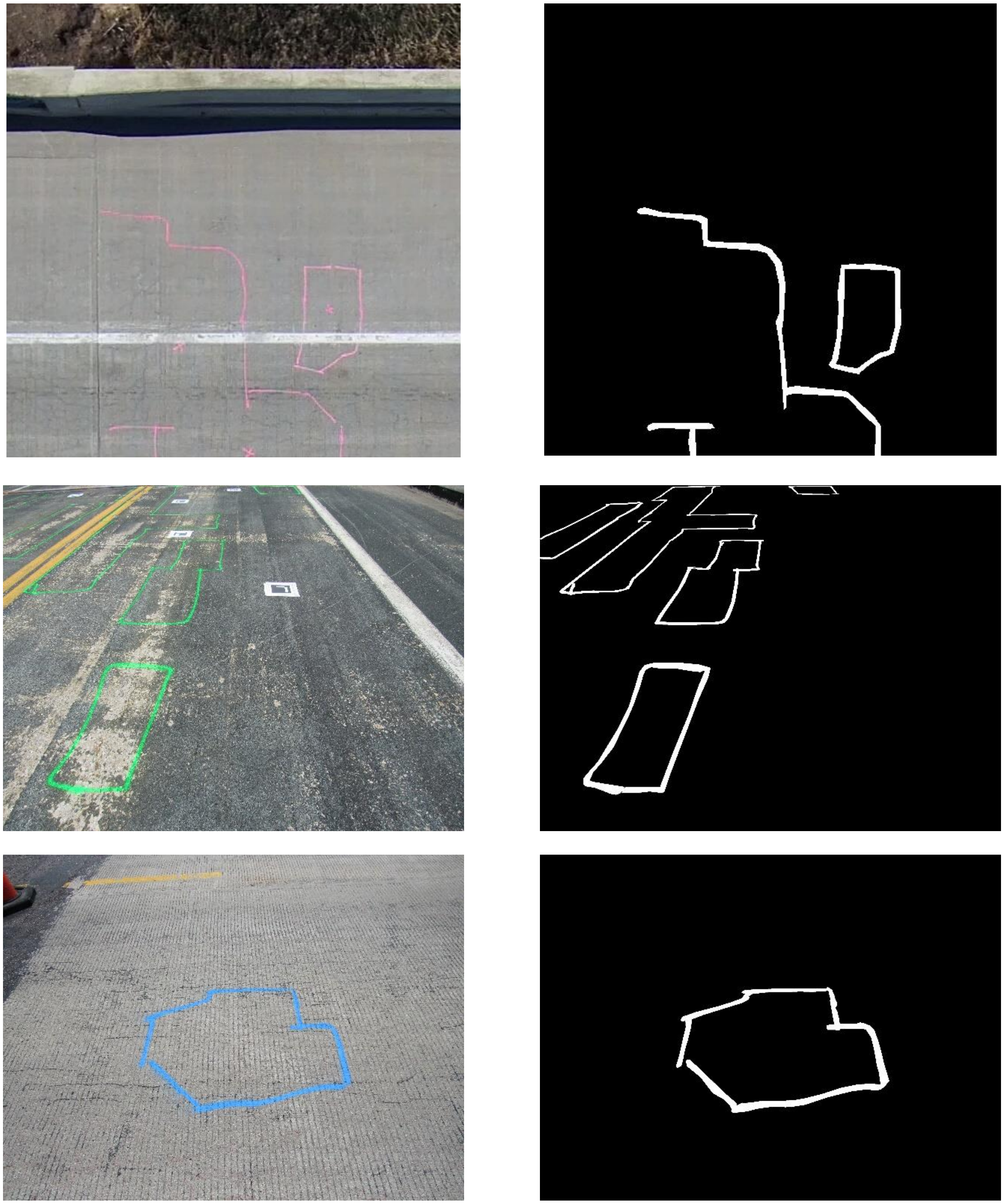


Figure 5.3 – AI-based Delamination Detection Flowchart Using LiDAR and Inspector Annotations

5.4.1 Data Preparation

The performance of any deep learning model lies on the quality and volume of the data used for training and validation. Deep learning models inherently learn better on huge numbers of data points that makes it better in generalizing the probability distribution. This data dependency requires the training data to be as accurate as possible to the real application scenario. The dataset used to train neural networks of the present study was the one discussed in **Ch. 4**. In summary, the inspector annotated database had 1918 images taken from various devices.

To better generalize features when the number of datapoints is limited (e.g., thousands instead of millions), several data augmentation techniques can be utilized to further diversify the dataset. In the present study, the training set was color augmented in the hue, saturation, and value (HSV) color space to generate different variations of the paint colors when the inspector annotates a delamination. This augmentation was applied randomly with a probability of 50%. Furthermore, different geometric augmentations such as scaling and rotation were applied. Scaling ranged from 0.5 to 1 of the training sample with a probability of 50%. A rotation of -45 degree to +45 degree was also applied to increase the number of samples. **Figure. 5.4** shows samples of raw and labeled data that were used in the model training.



Original Image

Labeled Image

Figure 5.4 – Samples of Data and Labels Used in Training

The final dataset was split into a training and testing set using the industry standard of 90% for training and 10% for testing. The training set was further split into train and validation sets, 90% training and 10% for validation. **Table 5.1** presents a summary of the inspector annotation datasets used in training, validation, and testing of the network. Note that the training set of 1,553 was increased to 70,000 after performing the above-mentioned augmentations.

Table 5.1 – Dataset Distribution Used in Neural Network

Set	Number of Samples
Training	1553
Validation	172
Testing	192

5.4.2 Selected Neural Networks

A deep learning network, U-Net (Ronnenberger et. al., 2015), was selected for semantic segmentation tasks of the present project to find the inspector annotations for delamination. U-Net (**Fig. 5.5**) is composed of one encoding path, a bottleneck channel, and a decoding path to classify the pixels of images. This forms a U-shaped structure of the network. The encoding path involves a series of convolutional and max pooling layers, which progressively reduce the spatial dimensions of the input image while extracting hierarchical features. These layers act as an encoder, capturing context and high-level representations from the input image. The decoding path consists of upsampling and concatenation operations followed by convolutional layers. These layers gradually upsample the feature maps back to the original input resolution while incorporating high-resolution features from the contracting path through skip connections. The skip connections enable precise localization and refinement of segmentation boundaries by preserving fine-grained spatial information.

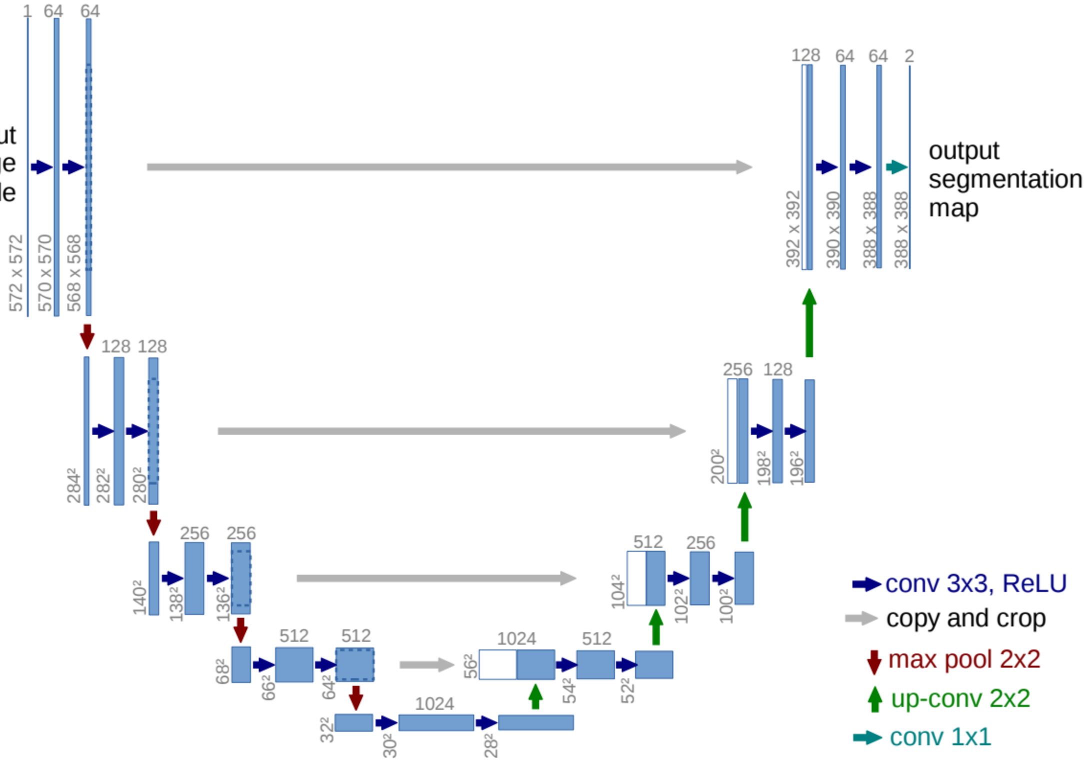


Figure 5.5 – U-Net Developed by Ronnenberger et al. (2015)

There are various versions of U-Net designed for different tasks and applications. In the present project, three variants of the U-Net model were explored for delamination detection: U-Net by Ronnenberger et. al. (2015), Attention U-Net by Oktay et al. (2018), and Recurrent Residual U-Net by Alom et al. (2018).

The Attention U-Net extends the model by introducing attention gates as filters on the outputs of the skip connections as shown in **Fig. 5.6**. The structure of Attention U-Net retains the symmetric encoder-decoder design of the original U-Net. In the encoder path, convolutional layers with ReLU activations and max-pooling operations are used to extract hierarchical features and downsample the input image. The decoder path employs upsampling and convolutional layers to reconstruct the image dimensions and refine the segmentation output. Attention gates are integrated into the skip connections between corresponding layers in the encoder and decoder paths. These gates learn to weigh the importance of different features dynamically, allowing the network to emphasize salient regions and ignore background noise.

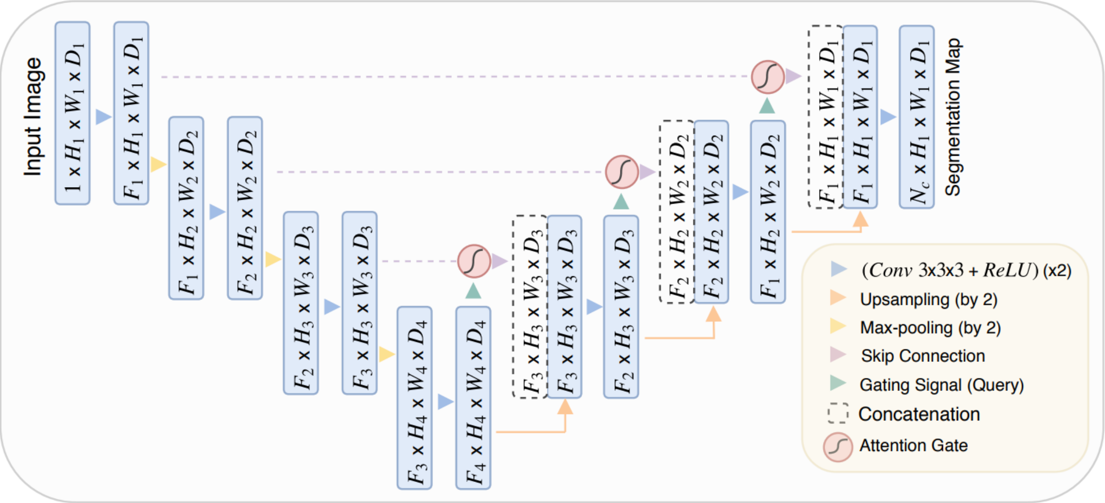

Input Image
$1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_2 \times W_2 \times D_2$
$F_2 \times H_2 \times W_2 \times D_2$
$F_2 \times H_3 \times W_3 \times D_3$
$F_3 \times H_3 \times W_3 \times D_3$
$F_3 \times H_4 \times W_4 \times D_4$
$F_4 \times H_4 \times W_4 \times D_4$
$F_3 \times H_3 \times W_3 \times D_3$
$F_3 \times H_3 \times W_3 \times D_3$
$F_2 \times H_3 \times W_3 \times D_3$
$F_2 \times H_2 \times W_2 \times D_2$
$F_2 \times H_2 \times W_2 \times D_2$
$F_1 \times H_2 \times W_2 \times D_2$
$F_1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_1 \times W_1 \times D_1$
$N_c \times H_1 \times W_1 \times D_1$
Segmentation Map
$(Conv$ 3x3x3 + $ReLU)$ (x2)
Upsampling (by 2)
Max-pooling (by 2)
Skip Connection
Gating Signal (Query)
Concatenation
Attention Gate

Figure 5.6 – Attention U-Net Framework Developed by Oktay et al. (2018)

The Recurrent Residual U-Net model extends the U-Net by replacing the last convolution block of each layer with a recurrent convolution block with residual connection (**Fig. 5.7**). The model is better in capturing long-range spatial dependencies and contextual information because of the residual layers, which add temporal connections to the network. Recurrent feature processing allows the network to iteratively improve its comprehension of the input data. The residual connections offer shortcut paths that avoid one or more layers, which makes it easier to train deeper networks. This expedites convergence and aids in the solution of the vanishing gradient issue. Remaining connections are added to the encoder and decoder paths in Recurrent Residual U-Net, which enhances feature propagation and network stability.

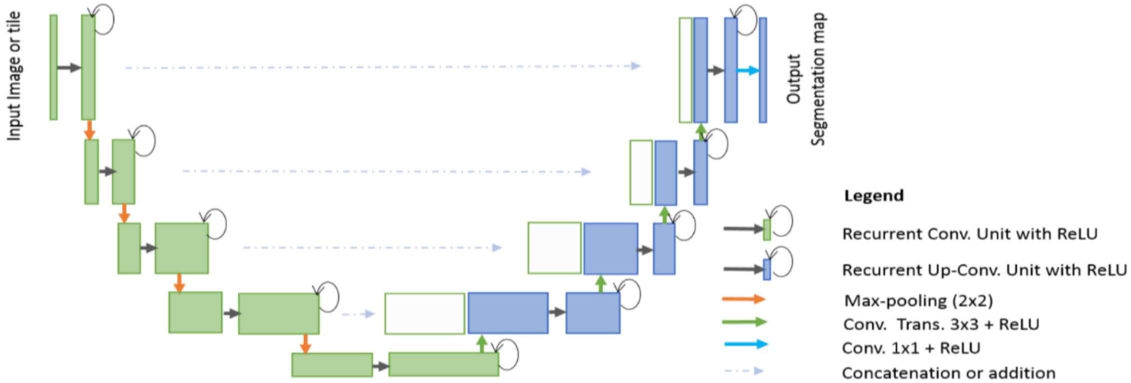


Figure 5.7 – Recurrent Residual U-Net Framework Developed by Alom et al. (2018)

5.4.3 Model Training

For these three different models, different hyperparameters were used but with the same loss function, the Dice function. **Table 5.2** lists the best hyperparameters of the models during training on the dataset.

Table 5.2 – Hyperparameters of Evaluated Deep Learning Models During Training

Model	Batch Size	Learning Rate	Weight Decay	Momentum
U-Net	5	0.0001	1×10^{-8}	0.999
Attention U-Net	2	0.0005	1×10^{-8}	0.99
Recurrent Residual U-Net	4	0.0001	1×10^{-8}	0.99

The models are trained on the training set of the annotated dataset with each epoch having a validation stage on the validation dataset. Each validation stage targets to save the best model parameters on the validation set. **Figure. 5.8** shows the trend of the validation score of the U-Net model. The best model had a score of 0.7375.

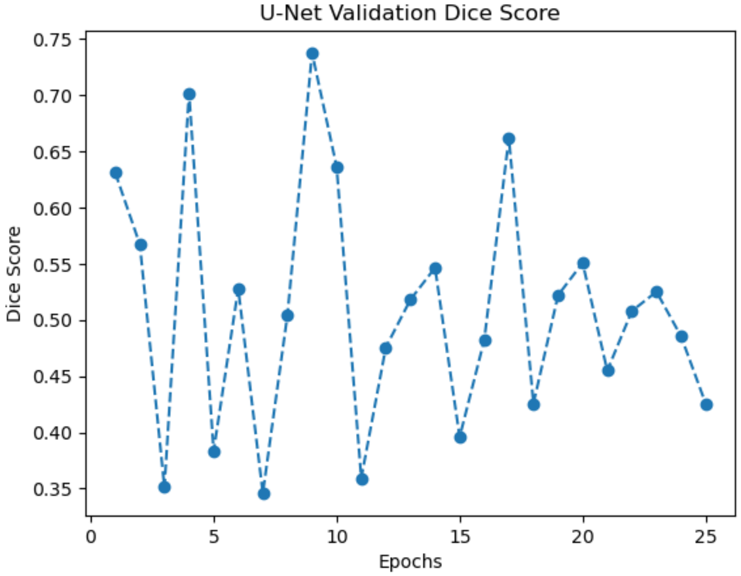


Figure 5.8 – U-Net Validation Score Trend during Training

Figure. 5.9 shows the trend of the validation score of the Attention U-Net model in which the best model had a score of 0.6265.

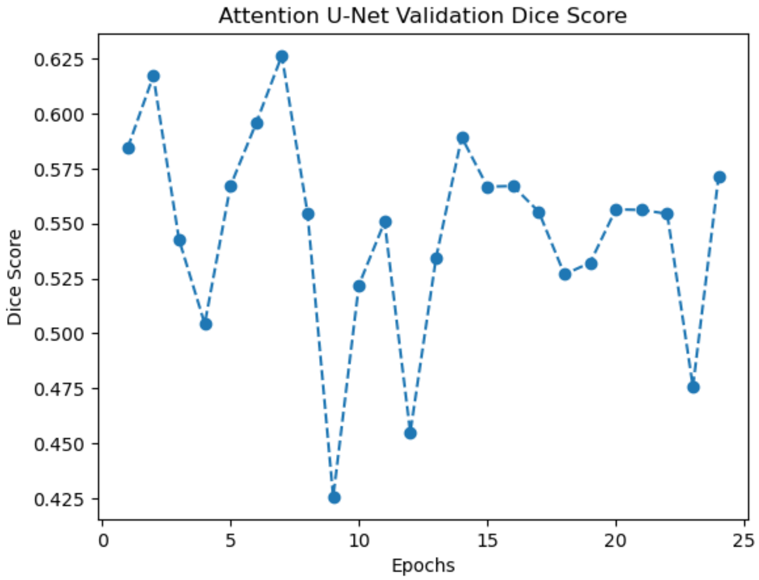


Figure 5.9 – Attention U-Net Validation Score Trend during Training

Figure. 5.10 shows the trend of the validation score of the Recurrent Residual U-Net model where the best model showed a score of 0.3125.

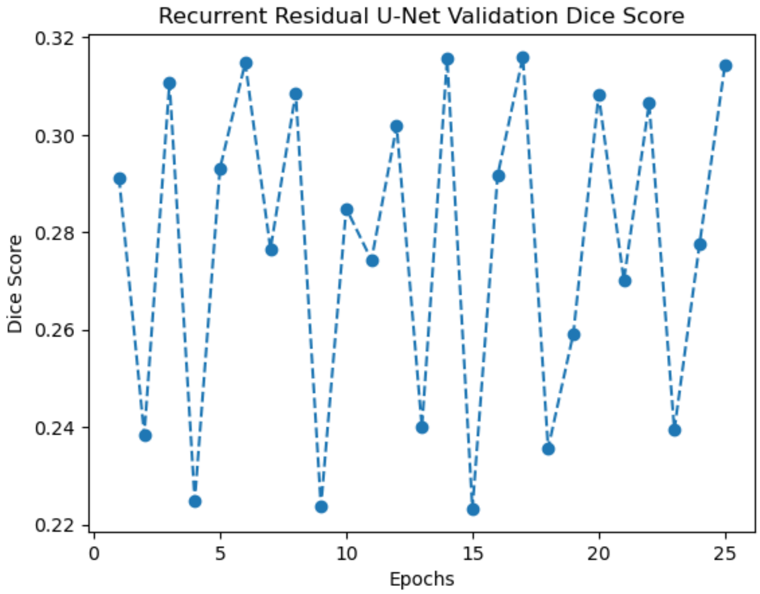


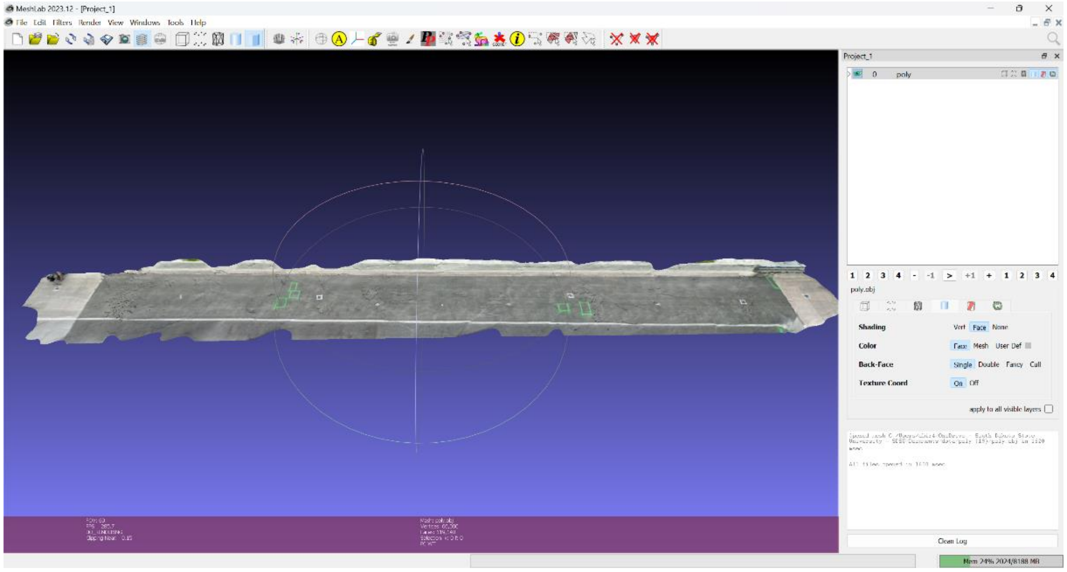
Figure 5.10 – Recurrent Residual U-Net Validation Score Trend during Training

The U-Net model achieved a validation score of 0.7375, indicating its effectiveness in segmenting the dataset. As shown in **Fig. 5.8**, the validation score of the U-Net model demonstrates a consistent improvement, culminating in its best score. The Attention U-Net and Recurrent Residual U-Net models both achieved a lower validation score. Despite the additional mechanisms designed to enhance segmentation performance in these models, such as attention mechanisms and recurrent residual connections, they did not surpass U-Net in terms of validation scores.

The results highlight that while advanced architectures like Attention U-Net and Recurrent Residual U-Net incorporate sophisticated mechanisms aimed at improving segmentation, the classical U-Net model remains highly competitive and outperformed the other models in this study. These findings suggest that the choice of model architecture should consider the specific dataset and task requirements, as more complex models do not necessarily guarantee a better performance. Further research and experimentation with different configurations and training strategies may help to improve the performance of these advanced models.

5.5 Bridge Deck Map Generation

Bridge deck maps were generated from 3D mesh scans recorded by an iPhone and an application that uses the phone's LiDAR sensor. In the present study, PolyCam was used to collect the point cloud data. At each bridge site, the bridge was scanned lane-by-lane to produce a global 3D map of the bridge deck. **Figure. 5.11** shows a sample of the recorded 3D mesh in a file viewer called "MeshLab".



Shading: Vert [✓ Face] None
Color: [✓ Face] Mesh User Def
Back-Face: [✓ Single] Double Fancy Cull
Texture Coord: [✓ On] Off
apply to all visible layers □

Figure 5.11 – A Sample of PolyCam Generated 3D Mesh Viewed in MeshLab

Subsequently, the 3D scan recorder with an extension of OBJ including its texture map was processed to obtain a 2D image of the bridge deck by mapping all 3D points to a reference plane, which was the bridge deck surface in the present study. **Figure. 5.12** shows a sample 2D map of a bridge deck extracted from its 3D map (**Fig. 5.11**).

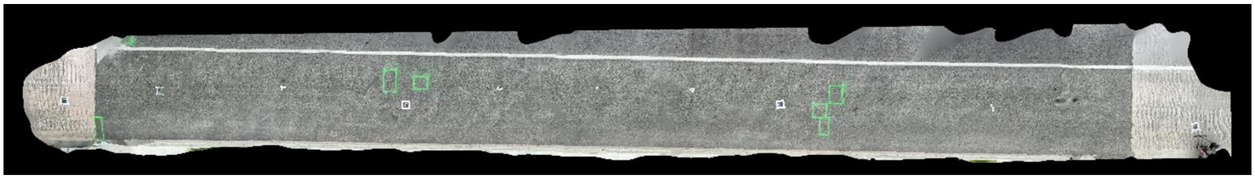


Figure 5.12 – 2D Maps of Bridge Decks Developed Using PolyCam 3D Mesh Scans

To obtain such a 2D image from its 3D mesh data, each of the faces of the mesh (triangles describing the planes) was processed independently to compute its projected transformations and to extract the pixel information given by the texture map. From these 3D mesh scans, an intricate process of extracting 2D maps was followed as shown in **Fig. 5.13**. The process begins with loading the mesh object exported from the PolyCam application and then parsing the object file to load the vertices, textures, and segment information. Then, the histogram of the depth of the vertices is computed to clip the vertices that are too far from the reference plane, which is the bridge deck surface. After getting vertices that fell on the deck, a Principal Component Analysis (PCA) is carried out to find the best plane that contains those vertices. Subsequently, the vertices are projected on the reference plane. Finally, the textures are extracted from the texture map given by the *uv* coordinates of the mesh descriptor.

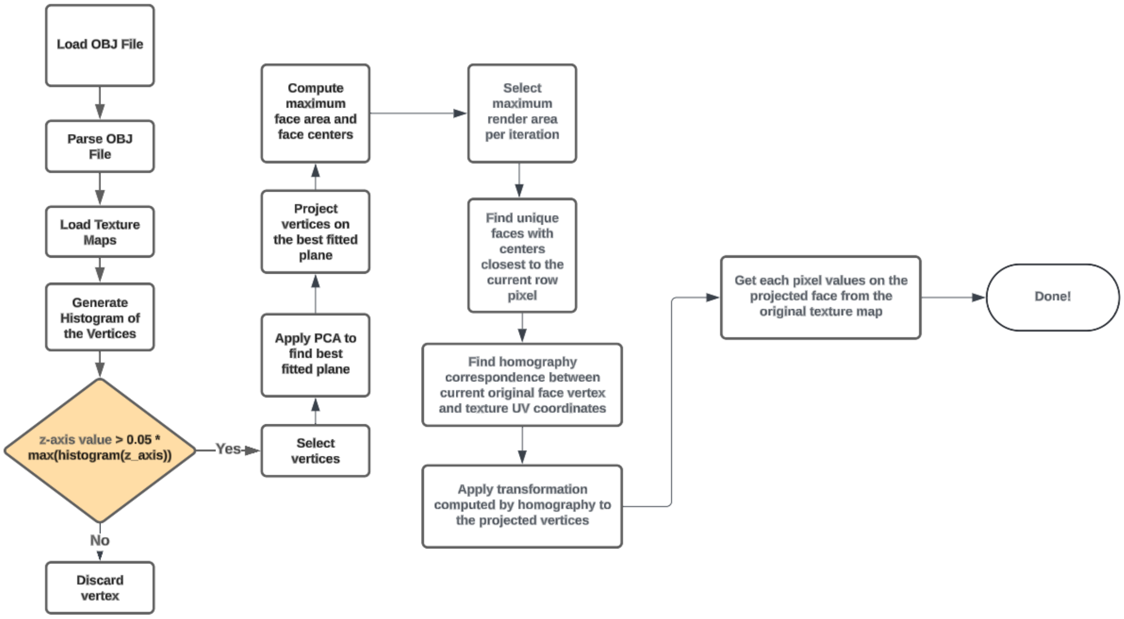

Load OBJ File
Parse OBJ File
Load Texture Maps
Generate Histogram of the Vertices
z-axis value > 0.05 * max(histogram(z_axis))
Yes
No
Discard vertex
Select vertices
Apply PCA to find best fitted plane
Project vertices on the best fitted plane
Compute maximum face area and face centers
Select maximum render area per iteration
Find unique faces with centers closest to the current row pixel
Find homography correspondence between current original face vertex and texture UV coordinates
Apply transformation computed by homography to the projected vertices
Get each pixel values on the projected face from the original texture map
Done!

Figure 5.13 – Flowchart for Bridge Deck 2D Map Extraction from Point Cloud

Once a 2D map of the bridge (either lane by lane or whole width of the bridge) is obtained, it is forwarded to the neural network for semantic segmentation and quantification of the inspector annotations.

5.6 Detection and Quantification of Delamination

For the detection of delamination, the best model, U-Net, from the three candidates discussed in **Sec. 5.4.3** was selected for further investigation and validation. As discussed before, this model outperformed the other two models by higher margins for this specific task.

5.6.1 Detection of Delamination

The detection of delamination is performed on the 2D map of the bridge deck generated from the previous stage. Since the generated map is a very high-resolution image and can cover the whole length of the bridge, the image was split into small patches to reduce computational efforts by generating a finer segmentation of the target pixels. The trained U-Net models look for the pixels that fall within the annotation zone (delamination found by inspectors). The annotation by the inspector ranges through different highlighting color spectrum. Usually, the region is marked with a water-based color spray. The model looks through each pixel of the input data to determine whether this is part of the annotation region or not. Once the entire bridge deck image is processed, the predicted masks are used to find each independent bounding region using a contour analysis. All the contours form a tree structure from which each child contour can be processed to determine local deficiencies per delaminated area and cumulative global deficiencies by adding them together. Each local deficiency is further analyzed using the proposed condition states by their area. The following algorithm describes the process of computing the local area in pixels for each delamination:

Table 5.3 – Algorithm to Compute Delaminated Area

```
Input: Predicted Semantic Segmentation Map,  map
Output: local_damage_area, refined_contours

# Close any open polygon applying a 5x5 dilation kernel
processed_map := Apply Morphological Extension on map
hierarchy, contours := DetectContours (preprocessed_map)

# Refine Contours and Compute Local Damage Area
local_damage_area := []
foreach contour in contours:
    refined_contour := []
    foreach point in contour:
        refined_point := Find Coordinate of the Closest Pixel with Value == 1
        refined_contour := Append(refined_contour, refined_point)
    end
    refined_contours := Append(refined_contours, refined_contour)
    refined_area := PolygonArea(refined_contours)
    local_damage_area := Append(local_damage_area, refined_area)
end
```

5.6.2 Quantification of Condition States for Delamination

Following the abovementioned method, the predicted delaminated areas are computed first in the pixel unit then are scaled to actual sizes (e.g., SI units) using the point cloud data. The quantification of the condition states for delamination on concrete decks was first presented in **Ch. 3**. A global flowchart of the software with condition state estimation was discussed earlier in **Sec. 5.4**. However, **Fig. 5.14** shows a close-up of the flowchart detailing the steps taken to estimate the delamination condition states for bridge decks.

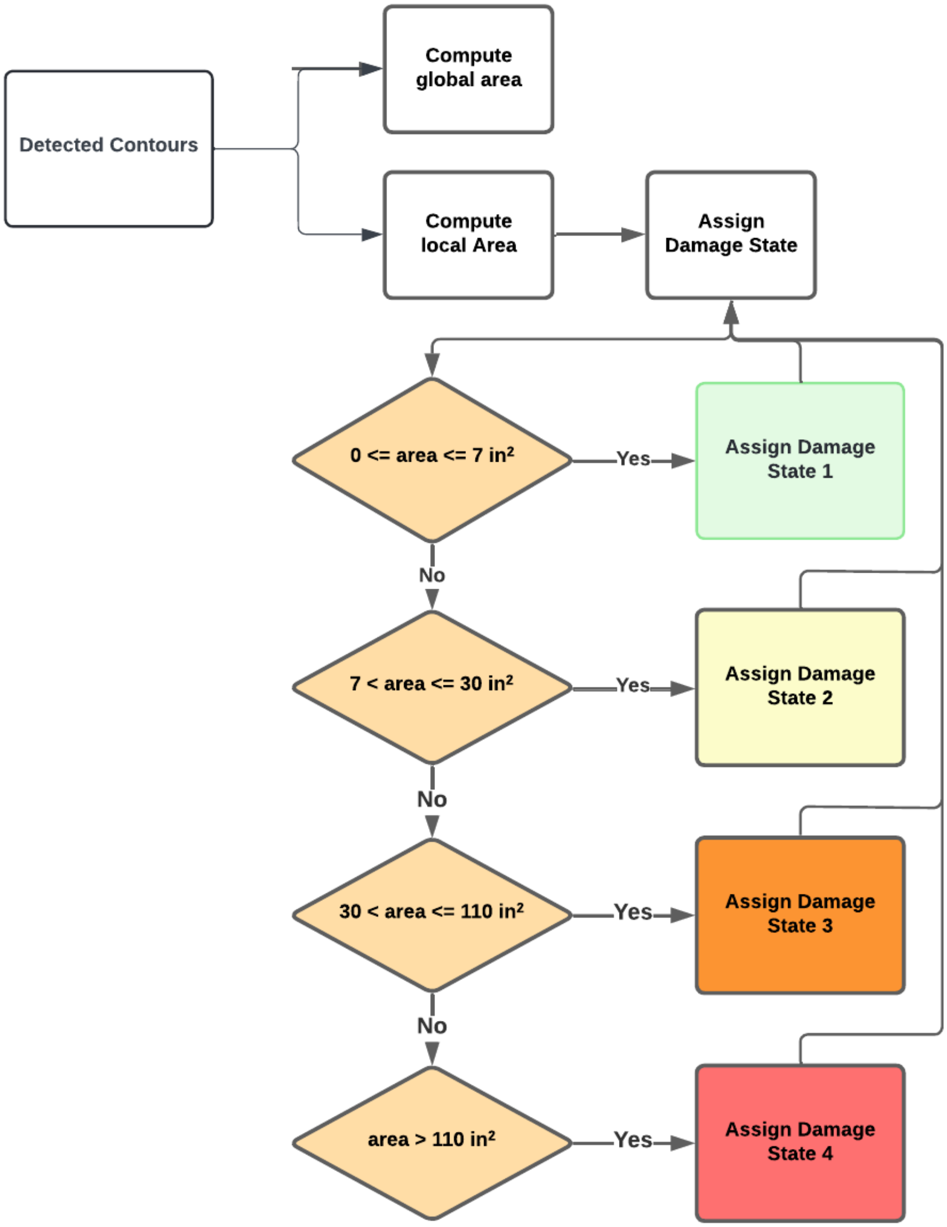


Figure 5.14 – Quantification of Condition States for Delamination

After computing the area of each delaminated region, the following algorithm was used to determine and label the condition state of each defect. The condition states are color coded according to AASHTO MBEI (2019) and then delaminated areas are overlaid on the top of the deck 2D map to form a complete damage map of the bridge deck.

Table 5.4 – Algorithm to Determine Condition State and Generate Damage Map

```
Input: refined_contours, local_damage_area, area_in_pixels, area_in_metric
Output: damage_state_per_delam

# Define conversion resolution from pixels to metric unit
resolution := area_in_inches / area_in_pixels

# Determine Damage States
damage_state_per_delam := []
foreach area in local_damage_area:
    unit_area := area * resolution
    # Damage State 1
    if $unit\_area \le 7$:
        damage_state_per_delam := Append(damage_state_per_delam, 1)
    elif $unit\_area > 7$ and $unit\_area \le 30$:
        damage_state_per_delam := Append(damage_state_per_delam, 2)
    elif $unit\_area > 30$ and $unit\_area \le 110$:
        damage_state_per_delam := Append(damage_state_per_delam, 3)
    else
        damage_state_per_delam := Append(damage_state_per_delam, 4)
    endif
end
```

5.7 Results of AI-Based Delamination Detection

The performance of the models was evaluated on the test set of the dataset (individual images) as well as the complete 2D deck maps of actual bridges, which were scanned with an iPhone and the maps were developed using the methods discussed in **Sec. 5.5**. Since each image of the testing set has one or a few delamination, such delamination can be assumed a "local" defect. However, a bridge deck might have many delaminated areas which must be evaluated. For ease of discussion, the deck-level delamination assessment is called "global" assessment. In other words, a local assessment is to determine the accuracy of the CNN model at the pixel level while the global assessment is to compare the predicted and actual damages at the area level. The former was used during training and the latter was used during testing of the networks. The accuracy at both levels is discussed herein. Note that a pixel-to-pixel assessment was needed during the training of the models and to select the best model for delamination task.

5.7.1 Local Assessment of Neural Networks for Delamination

The local delamination was assessed on the patches of images with and without delamination. The Dice score can be used to evaluate such local performances. The Dice score is calculated using two other indices named "precision" and "recall". The precision tells us about the ratio between the detected true positive (TP) instances and the sum of true positives and false positive (FP) instances while the recall is the ratio of detected true positives and the sum of true positives and false negatives (FN) as:

$$Precision = \frac{TP}{TP + FP} \tag{5.15}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.16}$$

In the present study, TP refers to the pixels in the ground truth image that are correctly identified as part of a delaminated area, which has been marked by inspectors during the inspection process. These are the pixels that both the inspectors and the model agree belong to a delaminated region, meaning the model has successfully detected the defect in the correct location. FP, on the other hand, represents the pixels that do not belong to a delaminated region in the ground truth image but have been incorrectly classified by the model as delaminated. These are the pixels where the model mistakenly predicts delamination when none exists according to the inspector's annotation. FN is the number of pixels that are wrongly categorized as background or another object when they belong to the target object or region. Precision measures the accuracy of the positive predictions made by the model while recall measures the ability of the model to find all the relevant pixels of the target object or region in the image. Subsequently, the Dice score (also known as the F-1 score) of the testing set is computed using the following equation:

$$Dice = \frac{2 \cdot Precision \ \cdot Recall}{Precision + Recall} \tag{5.17}$$

These local metrics were mostly assessed during the training of the neural models. **Table 5.5** presents a summary of the assessment for local (pixel-to-pixel) delamination. Furthermore, **Fig. 5.15** shows samples of the results at local views.

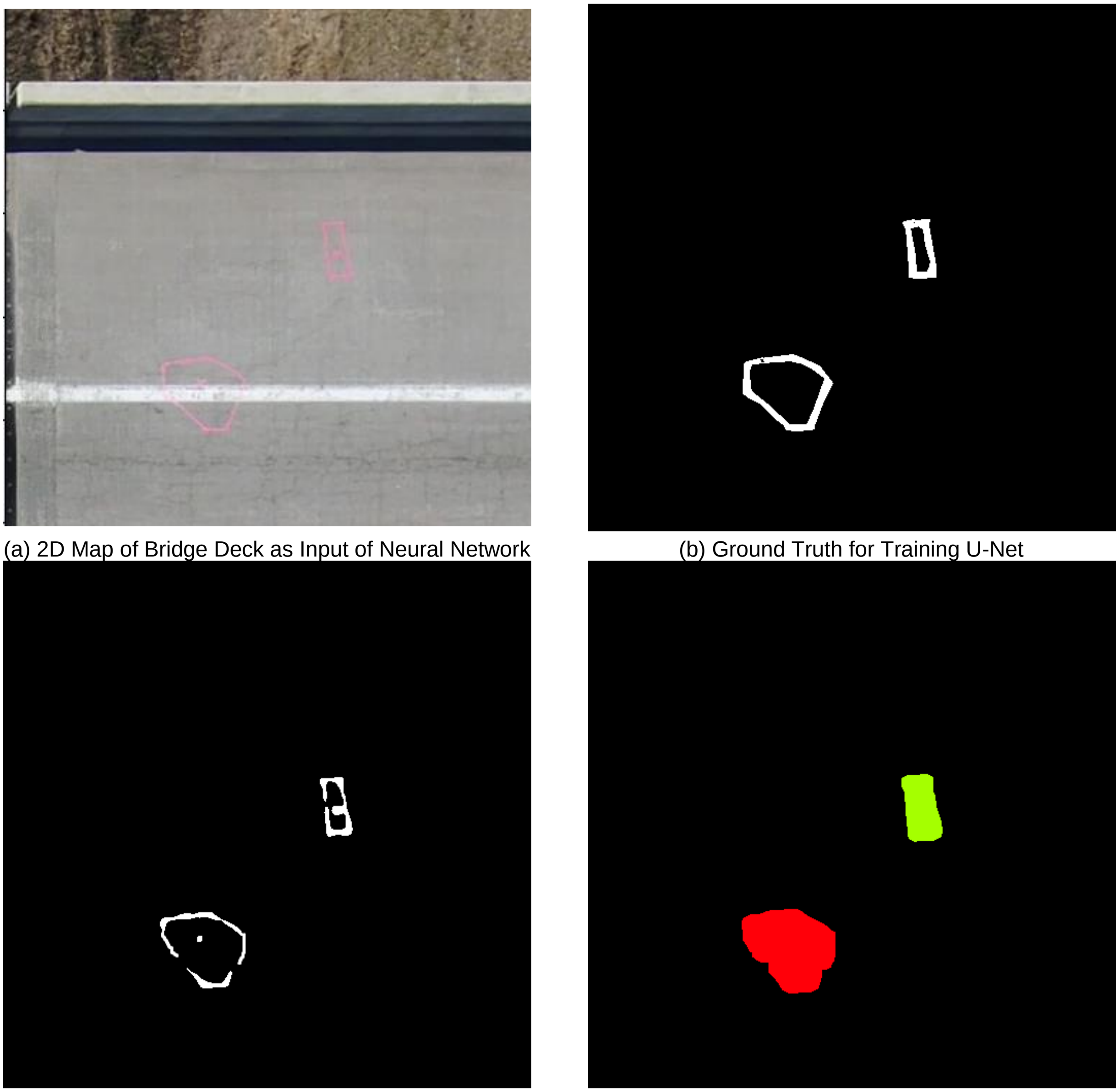
Table 5.5 – Local Assessment of Neural Networks for Delamination of Bridge Decks

Model	**Test Set Performance** **(Dice Score at Pixel Level)**
U-Net	73.75%
Attention U-Net	62.65%
Recurrent Residual U-Net	31.25%

CNNs work within a local area defined by a convolutional kernel to capture spatial hierarchies. This technique allows CNNs to learn features at different levels of abstraction, from low-level features (e.g., edges and textures) to high-level features (e.g., objects and patterns). While generating a segmentation mask for each pixel, the model determines the probability of that pixel belonging to a particular class based on its learned features and context from surrounding pixels. However, due to the complex nature of real-world images and the limitations of the model in perfectly capturing all nuances, it is challenging to achieve a complete pixel segmentation. Even the state-of-the-art models have limitations in fully capturing all possible variations of unseen data. Sometimes, CNN models may be overfitted during training thus they fail to perform well on new, unseen examples. There are also practical limitations on the size and complexity of the models that can be used, due to computational resources and the need for real-time performance in many applications.

The performance assessment of the CNN models in the present study was done at the local and global levels using different approaches. In the local assessment, a pixel-level assessment was done while in the global assessment the identified pixels were analyzed to enclose the damaged areas thus the area was compared with the area, instead of finding just the pixels that represent the inspector annotations at the local level. It was found, as further discussed next, that the global assessment shows a better accuracy than that of the local assessment. This is because finding the damaged area using all those pixels that are found by CNN is less sensitive to the pixels that are missed, versus a pixel-to-pixel comparison. **Figure 5.15** better shows why the global assessments are more accurate than the local estimates. **Figure 5.15a** shows a slice of the bridge deck that has delamination annotated by the inspector. **Figure 5.15b** shows the ground truth mask for the annotation. **Figure 5.15c** shows the pixel-wise prediction by the model where the model did not detect all the pixels which were part of the delaminated area. After performing a contour analysis, the boundary of the damaged region was found followed by a condition state analysis as shown in **Fig. 5.15d**. It is clear that when comparing the predicted delaminated area (**Fig. 5.15d**) with the actual delaminated area (**Fig. 5.15b**), the number of missed pixels was not important thus a global assessment is generally more accurate than a pixel-to-pixel assessment (compare **Fig. 5.15c** with **Fig.**

5.15b). Note that the accuracy presented in **Table 5.5** was at the pixel level not the area level, which is discussed next.



(a) 2D Map of Bridge Deck as Input of Neural Network

(b) Ground Truth for Training U-Net

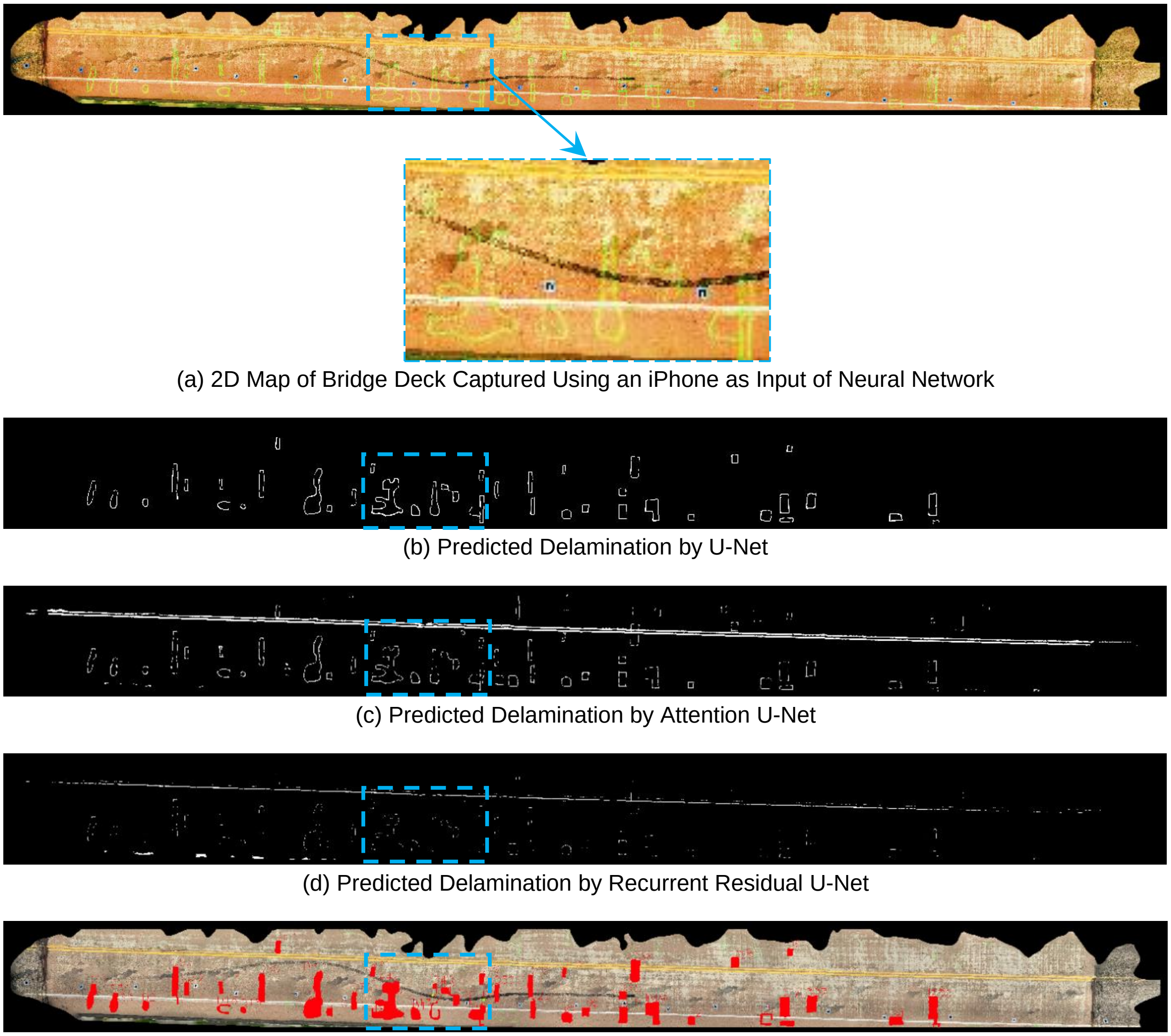
(c) Predicted Delamination by U-Net

(d) Contour Analysis and Condition State Evaluation on Predicted Mask

Figure. 5.15 – Samples of Local Assessments of Neural Networks for Delamination

5.7.2 Global Assessment of Neural Networks for Delamination

A delamination analysis at the global scale (e.g., area-to-area at the full length of the bridge) provides a complete assessment of the entire processes and methods of autonomous delamination detection in real applications. **Figure 5.16** shows one sample of delamination assessment for a bridge on its full length, which was 75.4 meters (248 ft).



(a) 2D Map of Bridge Deck Captured Using an iPhone as Input of Neural Network

(b) Predicted Delamination by U-Net

(c) Predicted Delamination by Attention U-Net

(d) Predicted Delamination by Recurrent Residual U-Net

(e) Condition States of Delaminated Regions Based on U-Net shown in (b)

Figure 5.16 – Delamination Assessment of a 75.4-m (248-ft) Bridge

Figure 5.16a shows the input image, a 2D map of the bridge deck including the marked delamination by the inspector (the enclosed areas with green paint), to the neural network. This input image is then processed to extract patches with a size of 512-by-512 pixels with a stride of 256. This makes each patch to have a 50% overlap with the adjacent ones.

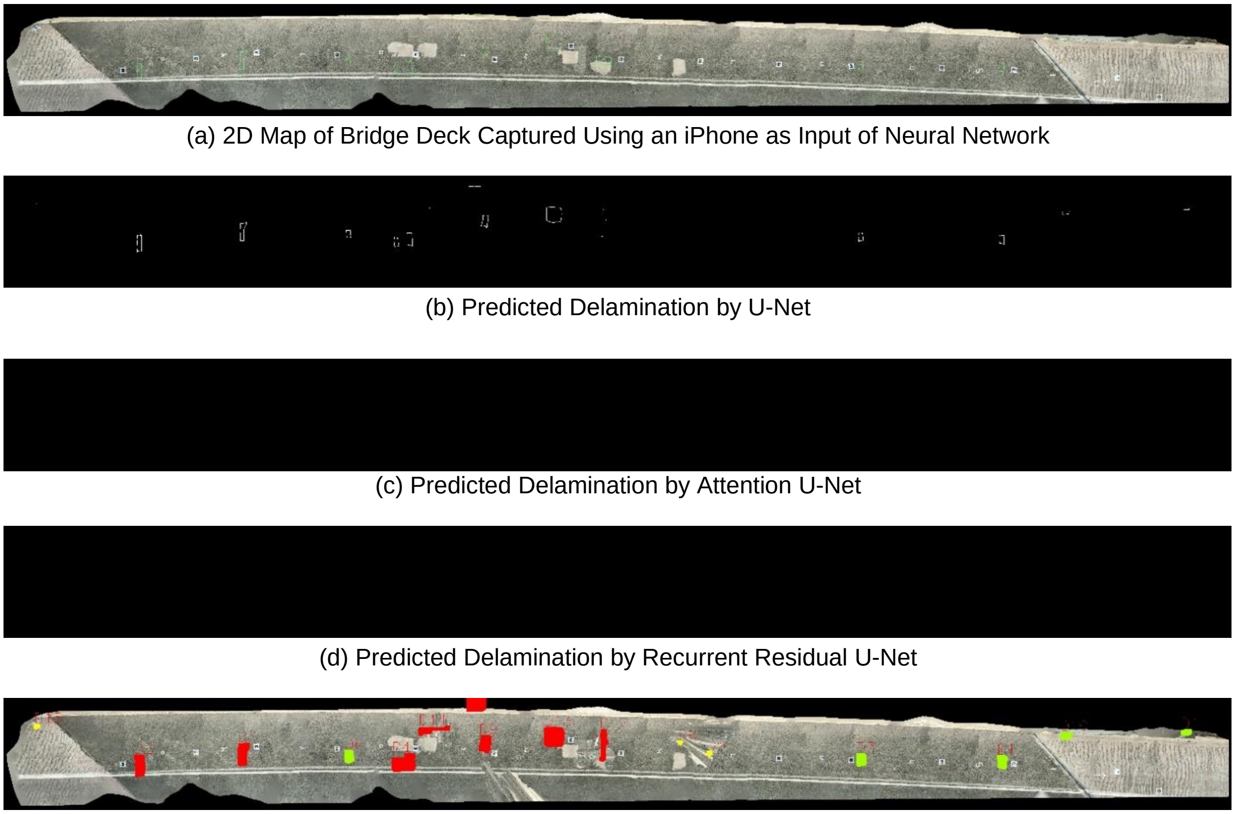
Figures 5.16b-d show the predicted pixels by the three neural networks as the delamination boundaries (finding the green markups automatically). The findings are then used to compute the delamination condition state for individual zone using a contour analysis on the post-processed prediction map. The predicted delaminated pixels are then connected using morphological transformations and passed through the contour analysis. After the completion of the contour analysis, which was done to enclose each delamination and to obtain its shape, each of the delaminated areas is then color coded based on the severity of the delamination (**Sec. 3.3**). The scale information was retrieved from the mesh (point cloud) data.

A similar evaluation was performed on 18 different bridges to evaluate the framework. **Table 5.6** presents a summary of the findings at the global level for these 18 bridges. The accuracy of the U-Net model was extremely high (with only 1% error) for these real-world applications.

Table 5.6 – Global Assessment of Neural Networks for Delamination of Bridge Decks

Model	**Deck Map Performance (Area)**
U-Net	99.5%
Attention U-Net	84.5%
Recurrent Residual U-Net	< 5%

Figures 5.17 and **5.18** show the results of similar area-level assessments for two more bridge decks, one curved and another skewed. It can be seen that in all these cases, U-Net showed a better performance compared with the other two models.



(a) 2D Map of Bridge Deck Captured Using an iPhone as Input of Neural Network

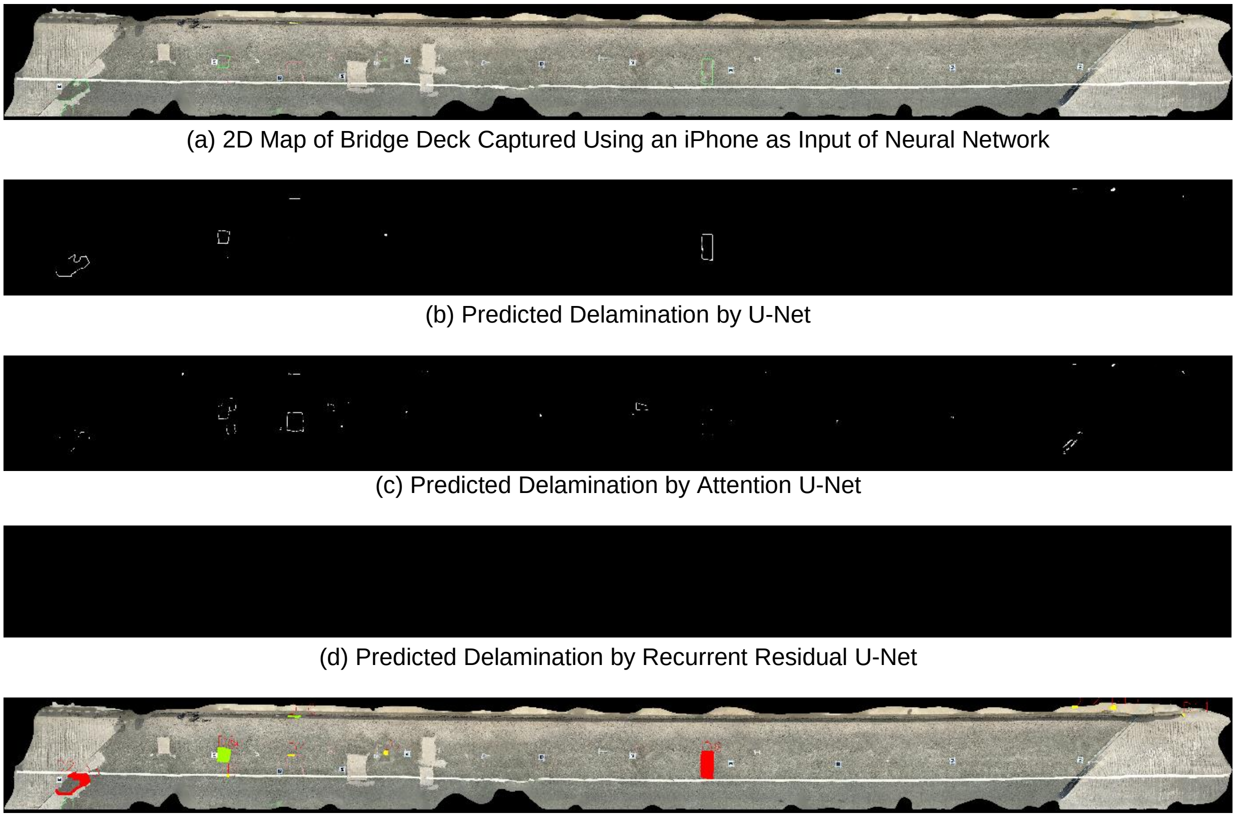
(b) Predicted Delamination by U-Net

(c) Predicted Delamination by Attention U-Net

(d) Predicted Delamination by Recurrent Residual U-Net

(e) Condition States of Delaminated Regions Based on U-Net shown in (b)

Figure 5.17 – Delamination Assessment of a 52.7-m (173-ft) Curved Bridge



(a) 2D Map of Bridge Deck Captured Using an iPhone as Input of Neural Network

(b) Predicted Delamination by U-Net

(c) Predicted Delamination by Attention U-Net

(d) Predicted Delamination by Recurrent Residual U-Net

(e) Condition States of Delaminated Regions Based on U-Net shown in (b)

Figure 5.18 – Delamination Assessment of a 52.4-m (172-ft) Skewed Bridge

5.7.3 Summary of AI-Based Delamination Prediction Results

Table 5.7 presents a summary of the local (pixel to pixel) and global (area to area) evaluations. The U-Net and Attention U-Net models performed well on the test set while Recurrent Residual U-Net showed an inferior performance on the test and deck map sets. Overall, U-Net performed better than the other two models both on the local and global scales and is recommended for field applications.

Table 5.7 – Summary of Neural Network Evaluation Results

Model	**Test Set (Local) Performance (Dice Score at Pixel Level)**	**Deck Map (Global) Performance (Area Level)**
U-Net	73.75%	99.5%
Attention U-Net	62.65%	84.5%
Recurrent Residual U-Net	31.25%	< 5%

5.8 Summary

This chapter presented a pipeline to automate delamination detection for concrete bridge decks that utilizes iPhone’s LiDAR sensor, point clouds, and computer vision and deep learning techniques. The deep learning models were trained with an early stopping criterion to prevent overfitting, and the dataset was diversified using geometric and color augmentation techniques. The models were trained on 1,553 samples, which then was increased to 70,000 after the augmentation. The local performance was assessed on a dataset including 192 samples. Furthermore, to evaluate the accuracy of the neural models on the real-world data, 18 bridges were assessed. For local detection of delamination (a pixel-to-pixel comparison), U-Net showed 73.75% overlap with the ground truth data. At the global level (an area-to-area comparison), U-Net exhibited an accuracy of 99.5%. Overall, U-Net was found to be the best tool for this task with the highest performance at the local and global levels.

CHAPTER 6. SUMMARY AND CONCLUSIONS

6.1 Summary

Regular maintenance of transportation infrastructure such as bridges is crucial in extending their service life while reducing long term repair costs. Transportation agencies such as the Federal Highway Administration (FHWA) seek emerging technologies and innovative programs to enhance asset management and maintenance. Furthermore, timely inspections of bridges by trained personnel are mandated per national and state specifications. The Alaska Department of Transportation and Public Facilities (DOT&PF) engineers inspect approximately 1000 bridges every two years. Of which, approximately 44% are in good condition, 49% are in fair condition, and 7% are rated poor based on the 2023 National Bridge Inventory (NBI) data.

In general, visual inspection is the common practice for inventory and routine inspections and is combined with other tools such as nondestructive evaluation (NDE) in other inspections for enhanced assessment. The inspector must complete both an NBI inspection and an element level inspection for each bridge. Nationwide, most steps of bridge inspections such as data collection, defect quantification, and damage reporting are done manually. Furthermore, each of these steps is time consuming, error prone, and hard to repeat in the following inspections due to the manual nature of these activities. Such measurements often require traffic control for the safety of inspection crew, and the findings might not be consistent if done by different inspectors.

Computer vision, a field of artificial intelligence (AI) that can analyze scenes at the human level performance, can speed up defect identification and quantification only using images of bridge elements. Progress in computer aided analysis of visual imagery has initiated widespread use of computer vision tools in various tasks, including but not limited to object classification, categorization, detection and generation by learning from provided datasets for specific tasks. Even though the academic and industry level developments are interesting and on the right track for field deployment, they usually lack practicality and are overwhelming for everyday users. For example, the academic-level computer programs are not as user-friendly as commercial software, and the industry-level software packages are so enlarged targeting multiple assets preventing them from being practical. In summary, current tools do not fully address the needs of DOTs for routine bridge inspections.

The primary objective of this project was to develop practical AI tools that help inspectors with defect detection and measurement and facilitate the inspection reporting following standard practices. To achieve the project goals, a comprehensive literature review of the current scientific trend and AI tools was performed on delamination detection, bridge map generation, and damage state estimation. Then, a new definition of condition states for delamination of concrete bridge decks suitable for computer programming was proposed. Subsequently, more than 45 bridges were inspected, and a comprehensive dataset of inspector annotated delamination was compiled using LiDAR and conventional cameras. Finally, an AI-based tool that can detect, quantify, and map the inspector’s annotations, and produce a delamination map of whole bridge deck including their condition states was developed.

6.2 Conclusions

The following conclusions can be drawn from this computer vision-based study on bridge inspections:

- A new condition state definition was developed for delamination of concrete bridge decks. The new definition enhances the physical interpretation of existing criteria in AASHTO by using definitive and quantitative measures at all four levels that are suitable for computer programming and automated analysis. The new definition ensures greater alignment between manual assessments and computer-based analyses, improving the accuracy and consistency of damage evaluations.
- A new image-LiDAR dataset of inspector annotated delamination was compiled after inspecting more than 45 bridges. The images are particularly applicable in training neural network models to automatically detect and measure inspector-annotated delaminated areas. This dataset contains 1,918 RGB photographs of inspector annotations and 18 LiDAR scans of bridge decks.
- Three computer vision models for image segmentation (U-Net, Attention U-Net, and Recurrent Residual U-Net) were evaluated following through hyper-parameter tuning, image preprocessing, and augmentation. The color and geometric augmentations of a somewhat small training dataset expanded the dataset to 70,000 to improve the generalization and accuracy.
- A method to generate high-quality bridge deck maps was developed to extract 2D bridge deck maps from reconstructed 3D mesh data. This allows high accuracy in quantification of the annotated area by inspectors.
- Two-level assessment of the detection of delaminated area in bridge decks were conducted. The local delamination assessment, which evaluates pixel-to-pixel correspondences, was conducted to measure the quality of the granular prediction. The pixel-to-pixel level accuracy was less than 75% for the best model. The global delamination assessment (area-to-area) exhibited, on the other hand, a near-to-perfect performance with 99.5% accuracy over 18 bridges. Furthermore, the global assessment confirmed that regardless of the quality of the local assessment, the framework can quantify the area inside the delaminated spot with high accuracy.

Overall, the proposed smart-phone and computer vision based study can expediate bridge deck inspections for delamination by automating the entire process of data collection using iPhone, analysis of point-clouds, and damage identification and condition state estimation using neural networks.

6.3 Future Works

The present work was on the framework of generating and quantifying delamination using inspector annotations. The tool can be expanded in future by including:

- Automated detection of spalling, patched areas, and other damage types in bridge decks.
- Inspection of substructure elements such as girders for different deficiencies.
- More sophisticated and computationally efficient deep learning model designed specifically for bridge damage detection to reduce top of the line hardware dependency.
- Producing a generic desktop-based application that can be extended in a cross-platform system.

REFERENCES

AASHTO MBEI. (2019). "Manual for Bridge Element Inspection, 2nd Edition, with 2022 Interim Revisions," American Association of State Highway and Transportation Officials (AASHTO), 126 pp.

Alaska Bridges and Structures Manual. (2017). Alaska Department of Transportation and Public Facilities, Juneau, AK, 386 pp.

Ali, R., Kang, D., Suh, G., & Cha, Y. J. (2021). Real-time multiple damage mapping using autonomous UAV and deep faster region-based neural networks for GPS-denied structures. Automation in Construction, 130, 103831.

Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., & Asari, V. K. (2018). Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. arXiv preprint arXiv:1802.06955.

Arya, D., Maeda, H., Ghosh, S. K., Toshniwal, D., & Sekimoto, Y. (2022). RDD2022: A multi‐national image dataset for automatic road damage detection. Geoscience Data Journal. assessment of standard RC bridge columns using computer vision and seismic analyses. Engineering Structures, 272, 115002.

ASTM (American Society for Testing and Materials) C876 − 22b., 2022. Standard test method for corrosion potentials of uncoated reinforcing steel in concrete. ASTM International: West Conshohocken, PA, USA. DOI: 10.1520/C0876-22B

Bianchi, E., Abbott, A. L., Tokekar, P., & Hebdon, M. (2021). COCO-bridge: Structural detail data set for bridge inspections. Journal of Computing in Civil Engineering, 35(3), 04021003.

Dong, C. Z., & Catbas, F. N. (2021). A review of computer vision–based structural health monitoring at local and global levels. Structural Health Monitoring, 20(2), 692-743.

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. Data in brief, 21, 1664-1668.

Dunphy, K., Sadhu, A., & Wang, J. (2022). Multiclass damage detection in concrete structures using a transfer learning-based generative adversarial networks. Structural Control and Health Monitoring, 29(11), e3079.

FHWA. (1995). "Recording and Coding Guide - Federal Highway Administration", Federal Highway Administration, 124 pp.

Gucunski, N., & Nazarian, S. (2010). Material characterization and condition assessment of reinforced concrete bridge decks by complementary NDE technologies. In Structures Congress 2010 (pp. 429-439). https://doi.org/10.1061/41130(369)40

Infobridge. (2024). Federal Highway Administration, Retrieved on Jan. 15, 2024: <https://infobridge.fhwa.dot.gov/>.

Jahanshahi, M. R., Masri, S. F., Padgett, C. W., & Sukhatme, G. S. (2013). An innovative methodology for detection and quantification of cracks through incorporation of depth perception. Machine vision and applications, 24, 227-241.

Jiang, S., Cheng, Y., & Zhang, J. (2023). Vision-guided unmanned aerial system for rapid multiple-type damage detection and localization. Structural Health Monitoring, 22(1), 319-337.

Jiang, S., Zhang, J., Wang, W., & Wang, Y. (2023). Automatic inspection of bridge bolts using unmanned aerial vision and adaptive scale unification-based deep learning. Remote Sensing, 15(2), 328.

Kao, S. P., Chang, Y. C., & Wang, F. L. (2023). Combining the YOLOv4 deep learning model with UAV imagery processing technology in the extraction and quantization of cracks in bridges. Sensors, 23(5), 2572.

Khudhair, M. and Gucunski, N., 2023. Multi-NDE Technology Approach to Improve Interpretation of Corrosion in Concrete Bridge Decks Based on Electrical Resistivity Measurements. Sensors, 23(19), p.8052. https://doi.org/10.3390/s23198052

Kim, Y. W., & Krishna, A. V. (2020). A study on the effect of Canny edge detection on downscaled images. Pattern Recognition and Image Analysis, 30, 372-381.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y. and Dollár, P. (2023). Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4015-4026.

Koch, C., Georgieva, K., Kasireddy, V., Akinci, B., & Fieguth, P. (2015). A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure. Advanced engineering informatics, 29(2), 196-210.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), 541-551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Lee, S., Kalos, N., & Shin, D. H. (2014). Non-destructive testing methods in the US for bridge inspection and maintenance. KSCE Journal of Civil Engineering, 18, 1322-1331.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road damage detection and classification using deep neural networks with smartphone images. Computer ‐ Aided Civil and Infrastructure Engineering, 33(12), 1127-1141.

Matsumoto, M., Mitani, K., & Catabas, F. N. (2014). Non-destructive bridge deck assessment using image processing and infrared thermography. In Transportation Research Board 93rd Annual Meeting.

Mirzazade, A., Popescu, C., Gonzalez-Libreros, J., Blanksvärd, T., Täljsten, B., & Sas, G. (2023). Semi-autonomous inspection for concrete structures using digital models and a hybrid approach based on deep learning and photogrammetry. Journal of Civil Structural Health Monitoring, 13(8), 1633-1652.

Oh, J. K., Jang, G., Oh, S., Lee, J. H., Yi, B. J., Moon, Y. S., ... & Choi, Y. (2009). Bridge inspection robot system with machine vision. Automation in Construction, 18(7), 929-941.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., ... & Rueckert, D. (2018). Attention u-net: Learning where to look for the pancreas. arXiv preprint arXiv:1804.03999.

Polycam. (n.d.). Polycam - LiDAR & 3D scanner for iPhone & Android. https://poly.cam/

Prasanna, P., Dana, K., Gucunski, N., & Basily, B. (2012, April). Computer-vision based crack detection and analysis. In Sensors and smart structures technologies for civil, mechanical, and aerospace systems 2012 (Vol. 8345, pp. 1143-1148). SPIE.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18 (pp. 234-241). Springer International Publishing.

Rubio, J. J., Kashiwa, T., Laiteerapong, T., Deng, W., Nagai, K., Escalera, S., ... & Prendinger, H. (2019). Multi-class structural damage segmentation using fully convolutional networks. Computers in Industry, 112, 103121.

Shi, Y., Cui, L., Qi, Z., Meng, F., & Chen, Z. (2016). Automatic road crack detection using random structured forests. IEEE Transactions on Intelligent Transportation Systems, 17(12), 3434-3445.

Tazarv, M., Won, K., Jang, Y., Hart, K. & Greeneway, E. (2022). Post-earthquake serviceability assessment of standard RC bridge columns using computer vision and seismic analyses. Engineering Structures, 272, p.115002.

Wiggenhauser, H. (2008). Advanced NDT methods for the assessment of concrete structures. In Concrete Repair, Rehabilitation and Retrofitting II (pp. 37-48). CRC Press. http://dx.doi.org/10.1201/9781439828403.ch3

Won, K., & Sim, C. (2020). Automated transverse crack mapping system with optical sensors and big data analytics. Sensors, 20(7), 1838.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13 (pp. 818-833). Springer International Publishing.

Zhang, C., Chang, C. C., & Jamshidi, M. (2020). Concrete bridge surface damage detection using a single‐stage detector. Computer‐Aided Civil and Infrastructure Engineering, 35(4), 389-409.

Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016, September). Road crack detection using deep convolutional neural network. In 2016 IEEE international conference on image processing (ICIP) (pp. 3708-3712). IEEE.

Zhu, Z., German, S., & Brilakis, I. (2010). Detection of large-scale concrete columns for automated bridge inspection. Automation in construction, 19(8), 1047-1055.

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., & Wang, S. (2018). Deepcrack: Learning hierarchical convolutional features for crack detection. IEEE transactions on image processing, 28(3), 1498-1512.