Post Print Version : Doi: https://doi.org/10.1177/1475921720932384

Instant bridge visual inspection using a UAV by image capturing and geo-tagging system and deep convolutional neural network

Muhammad Rakeh Saleem

Graduate student

Department of Civil and Environmental Engineering, Chung-Ang University

Email ID: rakeh@cau.ac.kr

Jong-Woong Park* (corresponding author)

Assistant Professor

Department of Civil and Environmental Engineering, Chung-Ang University

84 Heukseok-ro, Heukseok-dong, Dongjak-gu, Seoul 06974, South Korea.

Email ID: jongwoong@cau.ac.kr

Jin-Hwan Lee

Graduate student

Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology

Email ID: archi_tensai@kaist.ac.kr

Hyung-Jo Jung

Professor

Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology

Email ID: hjung@kaist.ac.kr

Muhammad Zohaib Sarwar

PhD Candidate

Department of Structural Engineering, Norwegian University of Science and Technology

Email ID: muhammad.z.sarwar@ntnu.no

Abstract

The structural condition of bridges is generally assessed using manual visual inspection. However, this approach consumes labor, time, and capital, and produces subjective results. Therefore, industries today are using automated visual inspection approaches, which quantify and localize damages such as cracks using robots and computer vision. This paper proposes an instant damage identification and localization approach that uses an image capturing and geo-tagging (ICGT) system and deep convolutional neural network (Deep CNN) for crack detection. The ICGT allows the geo-tagging of three-dimensional coordinates and camera pose data with bridge inspection images; the Deep CNN is trained for automated crack identification. The damages extracted by the CNN are instantly transformed into a global bridge damage map, with georeferencing data acquired using the ICGT. This method is experimentally validated through a lab-scale test on a wall and a field test on a bridge to demonstrate the performance of the instant damage map.

Keywords

Visual inspection, damage detection, image capturing and geo-tagging system, crack map, homography, structural health monitoring

Introduction

Bridges are continuously subjected to a combined effect of material aging, structural degradation caused by moving traffic, and environmental loads. Design and construction defects further deteriorate bridges and result in the loss of load carrying capacity [1]. Visual inspection is the primary method used for routine inspections to determine the status of a bridge. This approach uses non-destructive evaluation (NDE) techniques. However, visual inspection is laborious and time consuming, and introduces subjectivity in the results [2]–[5].

To overcome these limitations, researchers have introduced automated visual inspection systems for bridge health monitoring that utilize computer vision techniques [6]–[11]. Automated visual inspection is implemented by first capturing panoramic images according to a standardized procedure and performing automatic image processing to extract the damage locations and quantities. The images are generated by stitching inspection images, which is followed by damage extraction. The extracted damages are overlaid for the bridge inspection report. Researchers have used various robotic vehicles for automated visual inspection over the past two decades [12]. The ubiquitous bridge inspection robot system (U-BIROS) was developed [13] to scan bridges and generate a crack map using a robotic arm equipped with cameras. An autonomous robotic system for high-efficiency bridge deck inspection was proposed with several NDE sensors. The robotic crack inspection and mapping (ROCIM) [14] method was proposed for autonomous inspection using a visual mounted camera and integrated edge detector software. A wall-climbing robot was developed for monitoring reinforced concrete and detect corrosion to perform inspection using impact-echo and metric learning support vector machine (SVM) [15].

In the last two decades, computer-vision based methods are largely utilized in structural health monitoring (SHM) [16]–[19] of large scale concrete structures to facilitate automation and expedite inspection process. Consequently, a progression of SHM applications is seen in wide variety of concrete structures such as dam [20], road tunnels [21], wind turbine [22], [23], and others. In recent years, many researchers used unmanned aerial vehicles (UAVs) in visual inspection for its mobility and flexibility. An automated vision-based crack detection method was investigated for bridge damage detection using a convolution neural network (CNN) [24]; the study used three-dimensional dense point-cloud data to generate a condition-aware model for structure inspection with semantic segmentation. In another study, an autonomous UAV inspection method with a deep learning neural network and damage localization using geo-tagging was proposed [25]. Table 1 shows a detailed survey on state-of-the-art SHM methods however, almost all applications used computer vision for bridge inspection.

Table 1. Survey on state-of-the-art vision-based robotic systems for bridge inspection.

Classification	Literature	Proposed Methodology	Applications
UAV	Jang et al. [26]	Hybrid images combining vision and infrared thermography through CNN	UAV-based inspection of concrete cracks
	Akbar et al. [27]	SURF-based detection and image stitching using UAV	UAVs for reliable SHM on structural sites
	Lei et al. [28]	Vision-based inspection using SVM-support crack inspection	Low-cost UAV for efficient crack inspection
	Song et al. [29]	Integration of visual inertial state and three-dimensional LIDAR into graph structured SLAM	UAV-based inspection under bridges
	Brooks et al. [30]	Vision-based image processing to reconstruct three-dimensional sites and identify damage	Traffic monitoring and structural health inspection
Unmanned ground vehicle (UGV)	Prasanna et al. [7]	Machine learning-based robust, spatially tuned multifeatured classifier	Automated detection of cracks in concrete bridge
	La et al. [31]	High-efficiency bridge inspection using NDE sensors	Autonomous robotic system for bridge deck inspection
	Gibb et al. [32]	Autonomous system capable of quasi-real-time data processing using GPR	Robotic system for inspection of infrastructures
	Firoozi Yeganeh Sayna [33]	In-depth measurement using Kinect RGB-D camera combined with three-dimensional point cloud autonomous stitching	Autonomous quantification of rut depths using RDG–depth sensor.
Wall climbing robots	Jiang and Zhang [34]	Deep CNN combined with image post-processing	Applied on three-story building for crack detection.
	Li et al. [15]	Impact-echo inspection combined with metric learning SVM	Applied to bridge deck inspection with visible cracks
	Wang and Kawamura [35]	Automated sensing system using GMR sensor array	Applied for crack and corrosion inspection in steel bridges
	La et al. [36]	Image stitching and three-dimensional map construction using ICP algorithm	Three-dimensional structure inspection and crack detection for steel structures and bridges

SURF: speeded-up robust features; SLAM: simultaneous localization and mapping; GPR: ground penetrating radar;

GMR: giant magnetoresistance; ICP: iterative closest point.

While several studies have focused on developing image processing and machine learning techniques for automated damage identification, few have considered panoramic image stitching for bridge inspection [37], [38]. Moreover, when it comes to the inspection of concrete and steel surfaces, where distinctive image features are rarely found, handling the acquired images is a challenge. Instead of conducting image stitching, which finds correspondences between images, direct image stitching can be implemented using georeferenced images with the global positioning system (GPS) and inertia measurement unit (IMU), which provide six degrees-of-freedom (DoF) information where each image can be directly mapped for panoramic image stitching. For bridge inspection needs, a georeferencing system with a camera that records GPS, IMU, as well as light detection and ranging (LiDAR), which measures distance between structure and the camera, can be used.

In this study, we develop an image capturing and geo-tagging (ICGT) system that synchronizes GPS, IMU, and 1D-LiDAR with a camera installed on a UAV to generate georeferenced images of bridges, and uses a deep convolutional neural network (Deep CNN) for instant and automated damage detection and localization. The Deep CNN identifies damages in the georeferenced images acquired by the ICGT, and the extracted damages are projected to create a panoramic damage map using direct image stitching. This method comprises three parts: the (1) development of the ICGT platform; (2) development of a CNN-based crack detection algorithm incorporating image slicing and multistage training; and (3) direct projection for global bridge damage map (GBDM). It is noteworthy that several types of damage can occur in a bridge, but this study only focuses on crack detection for validation. The main contribution of this study is a framework for instant damage identification using a UAV incorporating Deep CNN. The proposed method is experimentally validated with a lab-scale test and a field experiment.

The remainder of the paper is organized as follows; a) framework of the proposed method detailing the overall procedure; b) development of the ICGT, with explanation of the system design, hardware development of the ICGT platform, and software framework; c) crack detection using deep CNN, describing the architecture of the CNN, database generation and training setup, parametric study for probability threshold, and evaluation of crack detection algorithm; d) GBDM, which proposes the integration of georeferenced data acquired by the ICGT for direct stitching of damages extracted by deep the CNN; e) experimental validation, detailing the lab-scale and field-scale tests and discussing the results; and, f) conclusions, which briefly articulate the overall summary and findings of the study.

Framework of the proposed method

The overall scheme of the proposed method is shown in Figure 1, which is subdivided into three stages. The first stage involves the development of the ICGT, which controls the camera shutter, and acquires GPS, IMU, and 1D-LiDAR data for tagging the location and the distance between the structure and the camera. To acquire georeferencing data from images simultaneously, a fast and low-cost ICGT was designed that is capable of handling middleware tasks efficiently. The ICGT module is application independent—it can be mounted either on a UAV or UGV for autonomous inspection or on a fixed platform for acquiring data. The second stage is the crack damage identification for offline inspection images using the Deep CNN; an image and masked based (Mask R-CNN) method is employed, owing to its GPU-accelerated processing for instant crack detection and damage identification. Lastly, the Mask R-CNN results of each image are extracted based on its geotagged position. The images are directly projected on to a global coordinate to generate a GBDM.

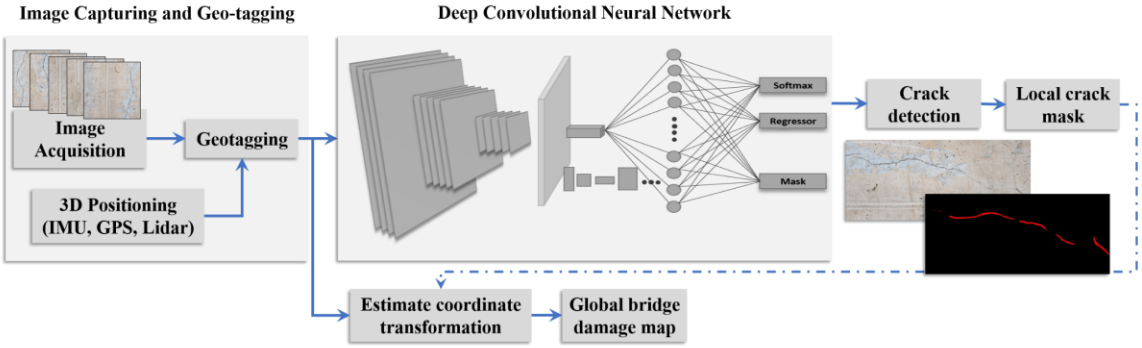


Image Capturing and Geo-tagging
Deep Convolutional Neural Network
Image Acquisition
Geotagging
3D Positioning (IMU, GPS, Lidar)
Softmax
Regressor
Mask
Crack detection
Local crack mask
Estimate coordinate transformation
Global bridge damage map

Figure 1. Overall scheme of proposed method

Development of an Image Capturing and Geo-Tagging System for UAVs

System design

A low-level block diagram of the ICGT is shown in Figure 2. The block diagram is composed of two units: (1) a UAV that feeds three-dimensional position data, such as GPS data, and three-dimensional LiDAR data to the ICGT; (2) an ICGT that controls the camera shutter and acquires sensor data such as that from the IMU and 1D-LiDAR. Communication between the UAV and ICGT is handled by a robot operating system (ROS). The ICGT comprises of a Teensy 3.6 microcontroller for handling low-level operations related to sensor initialization, data acquisition, and data syncing and storage. Teensy communicates with the UAV-master unit via the UART serial port, and IMU and 1D-LiDAR through the I2C interface. The camera unit is connected via general-purpose input/output pins to a multi-function universal serial bus port for program-controlled image capturing.

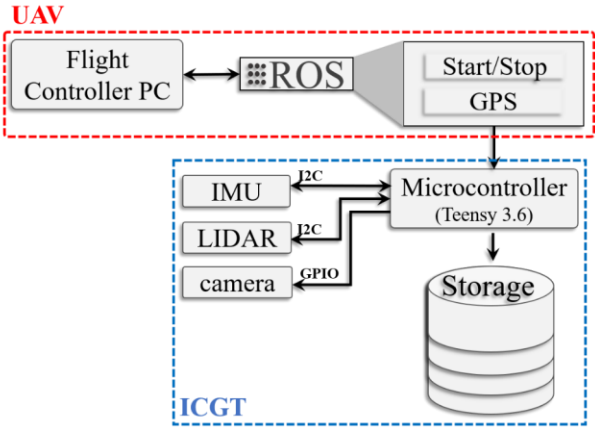


Figure 2. Low-level block diagram of ICGT design

Hardware development

Figure 3(a) shows the hardware for the ICGT comprised of Teensy 3.6, an IMU, a 1D-LiDAR, and a real-time clock (RTC). Teensy 3.6 has a 180 MHz ARM Cortex-M4 processor capable of performing tasks with high processing requirements. The IMU selected for the development was BNO055 IMU, integrating a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope, a triaxial magnetometer, and a 32-bit cortex M0+ microcontroller running sensor fusion algorithm for all three sensors. The accelerometer and gyroscope are less susceptible to external disturbances when running IMU fusion mode, as a result there is no significant drift and the offset is negligible [39]. 1D-LiDAR Lite v3 was used for a heading distance with a range of 40 m and accuracy of ±2.5 cm. Sony Alpha-9 (with 24.2-megapixel, 35-mm, full-frame stacked CMOS sensor) was chosen as the camera. The camera can fulfil the large buffer size and high-speed continuous shooting requirements of long-duration aerial inspections without interruptions. The actualization of the ICGT on top of the camera unit is shown in Figure 3(b).

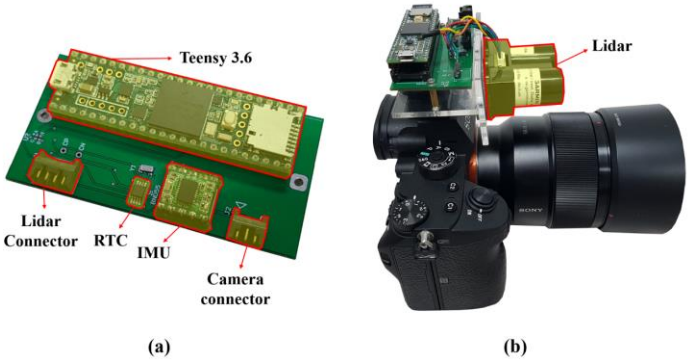


Figure 3. Development of ICGT: (a) ICGT hardware; (b) ICGT integrated with a camera

Software framework

The software framework of the ICGT is shown in Figure 4. The figure describes the overall process of the control software. Upon initialization, the master unit (i.e., UAV) and the slave unit (i.e., ICGT) are initialized, and all sensors are set up. The ICGT waits for the "Start/Stop" command to initiate the inspection through a middleware rosservice. Rosservice is a command-line tool within the ROS core toolchain and is used to commence or cease the ICGT from a remote station during the inspection. When the master unit transmits the "Start" command, the slave unit will perform two tasks simultaneously: (1) control the camera shutter for 500 ms to increase the focus and for 2 s to continuously shoot at 5 fps, (2) acquire data from the IMU, 1D-LiDAR, and GPS sensors. During the 500 ms of focusing, the sensor data are written to the SD card, and the camera buffer is cleared for the next 2 s of continuous shooting. Note that these two tasks run simultaneously at 5 fps, so that data between the images and the sensor data are synchronized. The procedure follows a cyclic routine continuously until the master unit gives the "Stop" command.

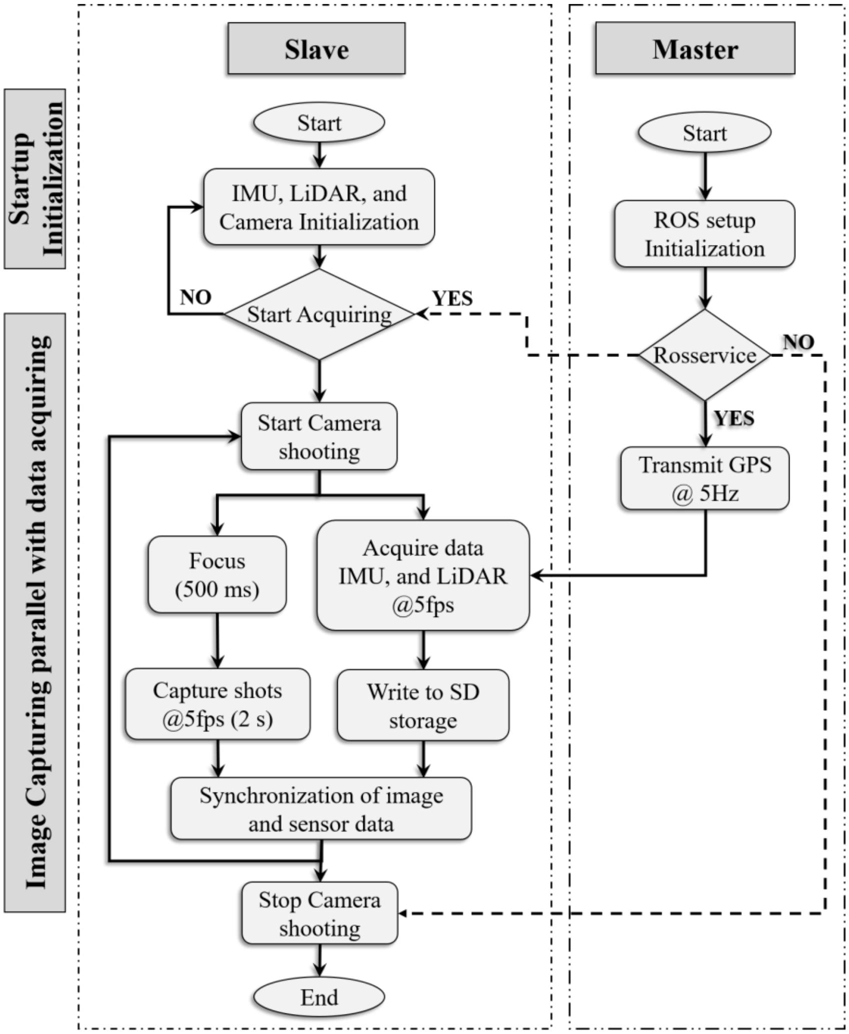


Figure 4. Software flowchart of the proposed system

Crack Detection using Deep Convolutional Neural Network

Image Slicing with Mask R-CNN Network

There have been several advancements in the cross-discipline of computer vision and deep learning, especially with respect to CNNs. As a sophisticated classification method, CNNs have been applied in structural inspection for object identification and localization, and subsequently a variety of CNN models have been proposed. Most object detection models such as Faster R-CNN [40], YOLO [41], and SSD [42] have achieved fast and accurate object detection. To attain the maximum possible crack detection efficiency, this study adopts a recent deep learning-based object detection method—the mask and region-based convolutional neural network [43].

In this study, we developed an image slicing-based crack detection method that combines mask-based region detection (Figure 5). This method is used to slice high-resolution images into small patches for fine crack detection and reduce the memory load for processing. A 2K resolution image is fed as the input, which is sliced down to a 512 × 512 image to preserve the low-level features that are passed through crack detection block. The outputs of these small patches are spliced to reconstruct the original 2K image with higher detection accuracy. The low-resolution images provide better accuracy in detecting minor cracks with less inference time, while a high-resolution image would require a longer inference time. The detection results are extracted in the form of binary mask vectors and are spliced together to reconstruct the original input images. The proposed method yields better results than the general procedure and is more reliable in terms of fine crack detection, which is critical in civil infrastructure inspection.

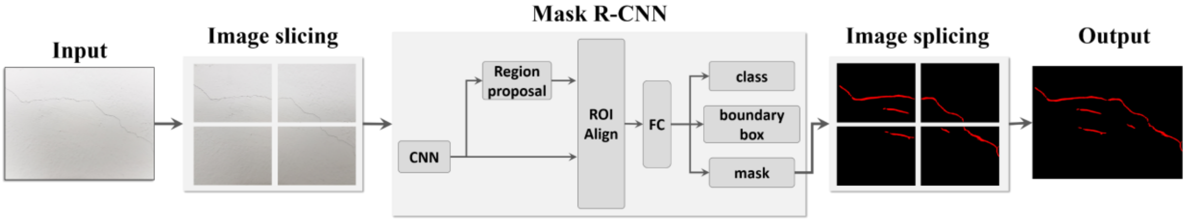


Figure 5. Network architecture of Mask R-CNN with image slicing and splicing

Database generation

Thus far, CNNs have been used to detect damages such as cracks, corrosions, and spalling from images or video, but few crack datasets are available. Most images are either taken by hand-held cameras or on-board cameras on vehicles (UGVs and UAVs). Therefore, the obtained images could vary with respect to field of view (FOV), with equal chances for blur and noise interference to occur. To address these problems and enhance detection performance, the dataset must have a large number of images with mixed variations of images. The proposed method uses a custom database with images acquired from two sources; (1) an open-source dataset called SDNET2018 [44] and (2) Google databank using ImageScraper [45]. In addition, 5% of the added images contain noise, which were originally part of SDNET2018, to handle occluded images in the field. To validate the database and compare its performance against a benchmark, a second dataset called crack forest dataset (CFD) was used. Details of the two datasets with their corresponding image characteristics and targets structures are listed in Table 2.

Table 2. Datasets used for crack classification.

Name	Source	No. of Images	Train: Test	Resolution	Target structure	Noise	Location
Dataset-1	SDNET 2018* + Google	1073	853:220	~256 × 256 1920 × 1080	54 bridge decks 72 walls, 104 pavements	Yes Blurring was added by authors	Utah State University, Logan, Utah, USA
Dataset-2	CFD	329	263: 66	480 × 320	Road surface	No	Urban road surface, Beijing, China

*Note: SDNET2018 is large database consisting of 56000 images, from which only images of bridge decks with concrete cracks were used.

Proposed training configuration

The collected images of bridges with cracks were annotated manually using the VGG image annotator [46]—an online tool for defining and labeling images. Due to the size limitation of datasets, the data were augmented during the training to prevent overfitting. For training and validation, data were divided in an 80:20 ratio, respectively, and the model was initialized with weights of a pretrained model called ResNet50 [47], to reduce training time and computation complexity. We optimized the hyperparameters and compared them with those of the general configuration by adopting a multistage scheduled learning for performance comparison, as shown in Table 3. The progressive learning rate with the multistage training procedure helped the model learn expeditiously without failing in the initial training itself. To time the models, we used a server with 24×Intel Core i9-7920X CPU, 4×NVIDIA GeForce RTX 2080Ti GPU, and 128 GB RAM. Since the training requires a large memory size, model files with increasing training and validation accuracies were saved, and those with low accuracy were discarded. This helped to reduce the memory allocation. The optimized model with increased accuracy was used for damage detection.

Table 3. General configuration and proposed training parameters

Hyperparameters	General configuration	Proposed configuration	
Batch size	1	4	
Steps per epoch	500	1000	
Weight decay	0.01	0.001	
Multi-stage	✗	✓	
		Stage-1	Stage-2
Layers	Heads	heads	all
No. of epoch	100	100	200
Validation steps	50	50	200
Learning rate	0.001	0.001	0.0001

Parametric Study of Probability Threshold

The baseline (ground truth) for crack detection can differ with the environmental conditions and inspection procedures to be followed for a certain inspection site. For cases where precision is more important than recall, the intersection-over-union (IoU) threshold must be high, and vice versa. Generally, recall is considered critical to the inspection of infrastructures, in order to not miss any possible source (crack) of failure. The evaluation of a model can be performed based on the true positives and true negatives extracted from a confusion matrix, which describes the actual and predicted score. To evaluate the proposed model, precision, recall, accuracy, and F1-score were considered as the performance measures [Eq. (1–4)].

$$Precision = \frac{TP}{TP + FP} \qquad (1),$$

$$Recall = \frac{TP}{TP + FN} \qquad (2),$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (3),$$

$$F1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \qquad (4),$$

where TP, TN, FP, and FN are the true positive, true negative, false positive, and false negative, respectively. TP, TN and FP, FN are defined mask-wise with respect to the ground truths, as in previous studies [43]. A validation dataset was considered to calculate the parameters, and the threshold for IoU was set as 0.5. In addition, the mean average precision (mAP) of images was calculated based on the COCO evaluation metrics in the IoU range of 0.5–0.95.

Evaluation of Crack Detection Algorithm

The crack detection algorithm can be evaluated by comparing the training results using the general and proposed configurations for the individual evaluation metrics. To evaluate the performance of the trained model, recall must be chosen as the primary indicator because it is the key to identifying the cause of damage. Table 4 lists the evaluation results for two different training configurations in dataset-1 and dataset-2. From the performance measures, the trained model with dataset-1 showed better performance compared to dataset-2. This is due to the large number of images used for training dataset-1, while dataset-2 was limited and suffered from overfitting. This shows that increasing the volume of data in a database will benefit the training process. The performance of the proposed configuration was tested on validation dataset-1 containing 220 images and dataset-2 containing 66 images, and the results are listed in Table 4.

Table 4. Evaluation metrics for custom dataset vs. crack forest dataset.

Training Configuration	Source	Accuracy (%)	Precision (%)	F1-score (%)	mAP (%)	Recall (%)
General	Dataset-1	54.93	57.68	55.30	39.27	53.12
	Dataset-2	63.93	56.00	61.24	31.73	61.24
Proposed	Dataset-1	**77.12**	**78.27**	**79.90**	**77.50**	**81.60**
	Dataset-2	53.70	48.56	59.43	34.23	76.57

*Note: The bold font indicates improved results.

To demonstrate the working of the crack predictions model, i.e., the applicability of the trained model to an on-site environment, a set of sample images was obtained from concrete structures in Chung-ang University, Seoul, South Korea, with smartphone camera (Samsung Galaxy S8). The images were tested with dataset-1 trained model selected from the proposed configuration, owing to its high validation accuracy and low validation loss. Meanwhile, the normal detection method, i.e., excluding image slicing, and the proposed detection method, i.e., including image slicing, were compared using the test images. Figure 6 shows the input images, their ground truth labels, crack detection without image slicing, and the proposed image slicing method using Mask R-CNN for single and multiple cracks. Note that these images vary from a plane to complex background and from single crack to multiple connected cracks. Image slicing helps in more accurate damage detection from high-resolution images as well as textured images. The description of cracks for *Figure 6 is* as follows; (a) crack on a textured wall with crack-like lines in the background; (b) small crack on a shiny steel-aluminum surface; (c) multiple cracks on a concrete wall with a thin hair-line crack; and, (d) a bridge pier with concrete voids. The crack damage results showed promising results with image slicing for crack detection when compared with normal image detection.

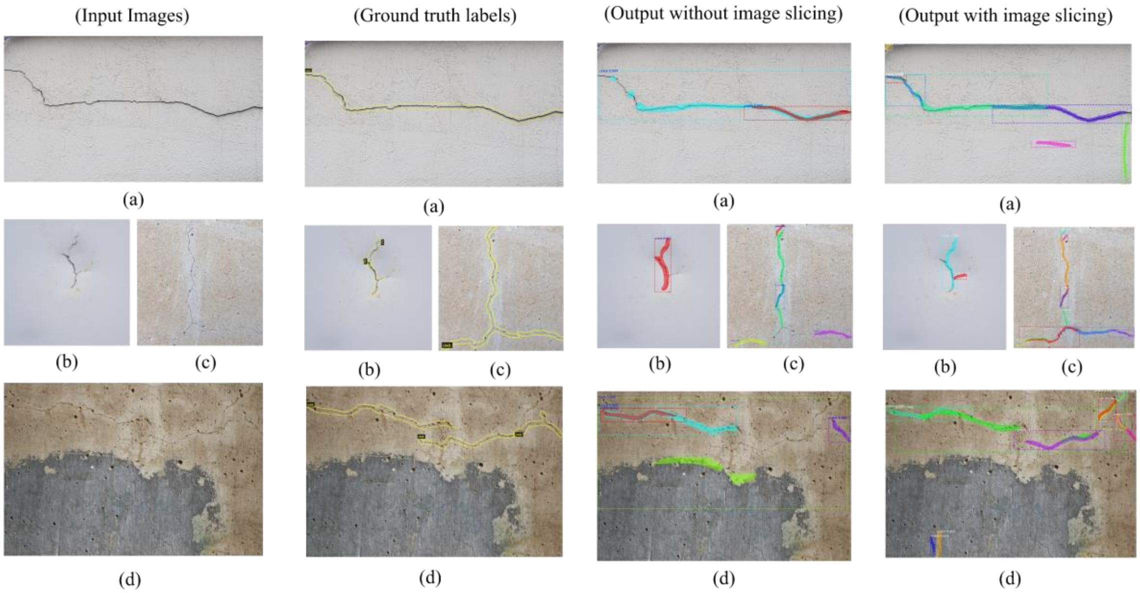


Figure 6. Mask R-CNN prediction results compared to actual images

Discussion

From Table 4, it is evident that the proposed training setup achieves better results than the general training method. This training configuration yields improved recall and precision with a batch size of 4, a weight decay of 0.001, and a scheduled learning momentum of 0.001 that increases to 0.0001 once the weights are learned to avoid overfitting. The empirical results show that increasing the batch size with low weight decay caused overfitting with no significant improvement in the overall accuracy or recall. The optimized model achieved a recall of 81.6% and a mAP of 77.50% when compared with database-2 which achieves a recall of 76.57%. Additionally, the trained model outperformed a previous study that also used Mask R-CNN to achieve a recall of 76.15% [48]. In addition, Figure 6 depicts the damage detection results from the image slicing method for high-resolution images. The method works well on images with fine to long cracks as well as crack-like texture and a redundant background.

Global Bridge Damage Map

To generate an inspection report and gather information about the overall condition of the bridge, it is important to have global knowledge of bridge damage. To validate the structural integrity, a post-processing method was applied to generate a GBDM after inspection. In this regard, damage indicators collected through a conjoint Mask R-CNN and the ICGT are projected on to a global coordinate plane.

Camera Calibration and Transformation

Image distortion is common in wide-angle lens cameras, and therefore the camera must be calibrated for precise crack detection. In this study, we adopted the camera calibration algorithm developed by Jean-Yves Bouguet [49]. The algorithm finds the extrinsic and intrinsic parameters by detecting checkerboard pattern and using the pin-hole model [50] to map the crack locations on to a two-dimensional plane. The equation defining the relationship between a two-dimensional point projected on to an image plane *(x,y)* for a three-dimensional point and the world coordinates

[U, V, W] is shown in Equation (5).

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [K] \cdot [R|T] \cdot \begin{bmatrix} u \\ v \\ w \\ 1 \end{bmatrix} \qquad (5),$$

where K is the camera intrinsic matrix that defines the camera focal length and the principle points, and s is the skew coefficient. *[R|T]* denotes the camera extrinsic matrix that corresponds to rotation and translation. If R_b is the given baseline rotation vector and R_i is the given rotation matrix, then R_{new} rotation can be measured using the relationship defined in Equation (6).

$$R_{new} = R_b^T R_i \qquad (6),$$

Image Projection

A digital image captured by a camera is a two-dimensional projection of a three-dimensional real-world object, and the task of image analysis depends on the projection of that two-dimensional image. Image projection is classified into two categories—perspective projection, in which parallel three-dimensional lines are not necessarily projected on to parallel lines in the images, and orthographic projection, which are parallel projections of the object on to the projection plane. To project a two-dimensional image on to a three-dimensional coordinate system and calculate the relative 6-DoF position of an image, the transform geometry must be known. The projection of image points on to the world coordinate points can be calculated using Equation (7) for two-dimensional image points. The outputs are the world points that map the image points on to the world points using an input rotation matrix, a translation matrix, and the camera parameters.

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = R^{-1}\left(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} K^{-1} - t\right) \qquad (7),$$

To create a GBDM, image projection should be applied to crack masks extracted from the CNN using georeferenced data collected from the ICGT. Figure 7(a) compares the ideal image plane with the actual transformed plane with no change in the heading (z-axis) or height (y-axis). Figure 7(b) and (c) shows the translated CNN masks of the captured images. For simplicity, we assumed the first image as the origin and calculated the rotation and translation with respect to it. Figure 7(d) shows the spliced mask vector image with coordinate transformation for two images.

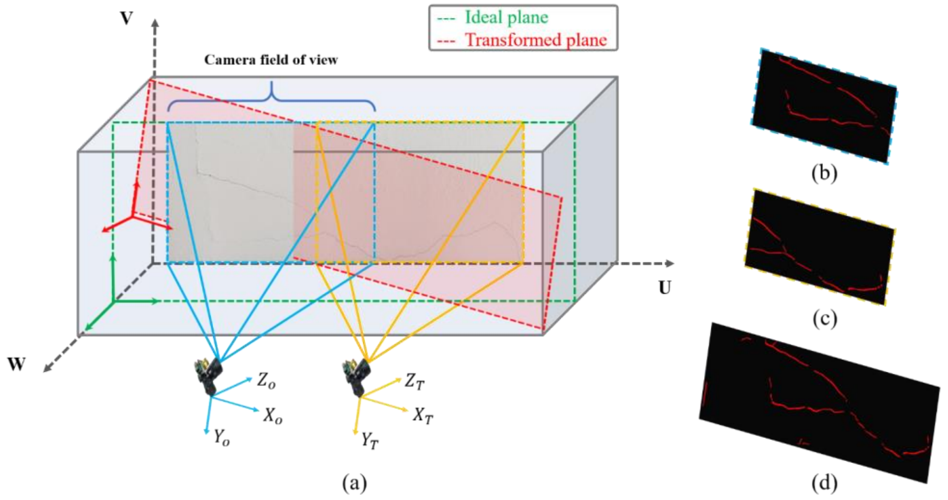


Figure 7. Image projection (a) Referenced ideal plane and transformed plane (b, c) Transformed crack masks (d) Spliced masks with coordinate transformation

In this figure, (*U, V, W)* correspond to the world points, (X_o, Y_o, Z_o) are the initial points of the camera location marked as the origin, and (X_T, Y_T, Z_T) are the translated camera points on the world coordinate plane. Point *X* denotes the translation along the x-axis, *Y* is the altitude of the camera, and *Z* is the heading that remains constant.

Implementation of the Global Bridge Damage Map

The global crack map can not only be used for quantitative analysis, but also for visually assessing the global crack pattern. An offline post-processing procedure was conducted to generate a two-dimensional bridge damage map using the transformation matrix, as it preserves the orientation information for all images. For efficient crack localization, we adopted image slicing to divide large-scale images to small-scale ones. After detection, we spliced them together and created a global bridge map. The details of the post-processing method are shown in Algorithm 1, providing the detailed flow of offline bridge damage map generation.

Algorithm 1: *Vision-based crack localization and mapping*

```
Input: Original RGB color image, In=(X×Y×3)
Output: Spliced image, Out=(X×Y×3)
Step 1: Get Aerial Images with respective Georeferencing Information.
    Load images, lidar and IMU measurements;
Step 2: Crack Feature Extraction.
    Perform mask R-CNN based crack detection;
    Reduce memory load by extracting binary masks;
    Save mask based binary images;
Step 3: Splice Back to Original Size.
    Add sliced patches back to regenerate original image;
    Join binary mask to reconstruct image of original size;
```

```
Step 4: Apply Coordinate Transformation.
    Calculate rotation and transformation matrices from IMU and LIDAR data;
    Compute World coordinates by applying coordinate transformation;
    Perform Image alignment;
Step 5: Global Map Generation.
    Project transformed masks on stitched bridge panorama;
    Splice each mask to bridge panorama w.r.t its localization and image transformation;
```

Experiments and Result Analyses

To evaluate the performance of the ICGT and CNN-based damage detection, a lab-scale experiment was conducted to validate indoor localization and damage identification, and a field-scale experiment was conducted for bridge structure inspection, as detailed in the following subsections.

Lab-scale Validation: Experimental Setup

For indoor experimentation of crack damage detection, the ICGT system was mounted on a tripod stand. The proposed system can be easily configured, and the mounted hardware can be attached in a fixed position or translated manually. The experimental setup is shown in Figure 8(a), where the ICGT is attached on top of the hot shoe of the Sony Alpha9 camera. The ICGT was placed approximately 1.2 m from the ground and 1 m from the target wall. The test was carried out in the corridor beside the lab at the College of Engineering, Chung-Ang University. The tests confirmed the ICGT can capture images with position information in indoor testing. Figure 8(b) shows the extracted crack, and the two images with 50% overlap were stitched together to understand the view of the camera.

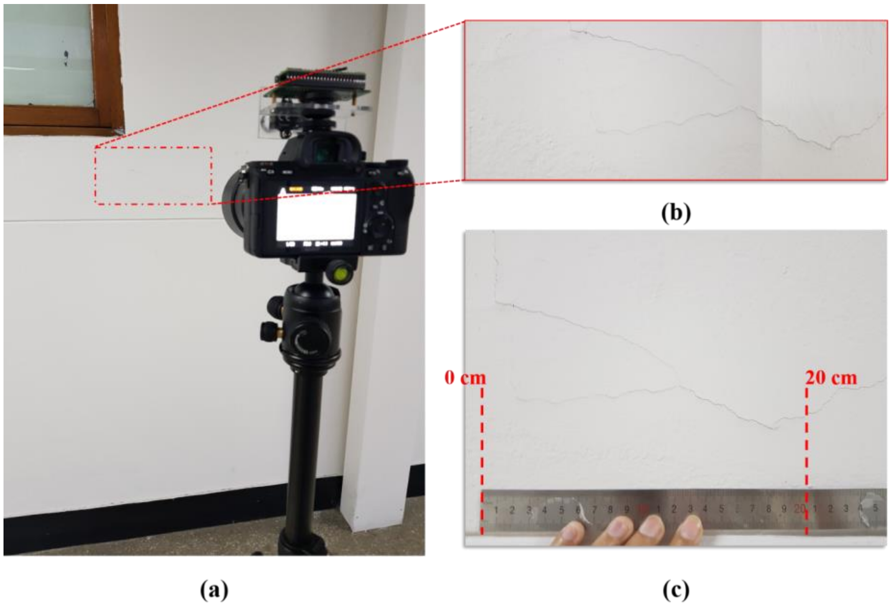
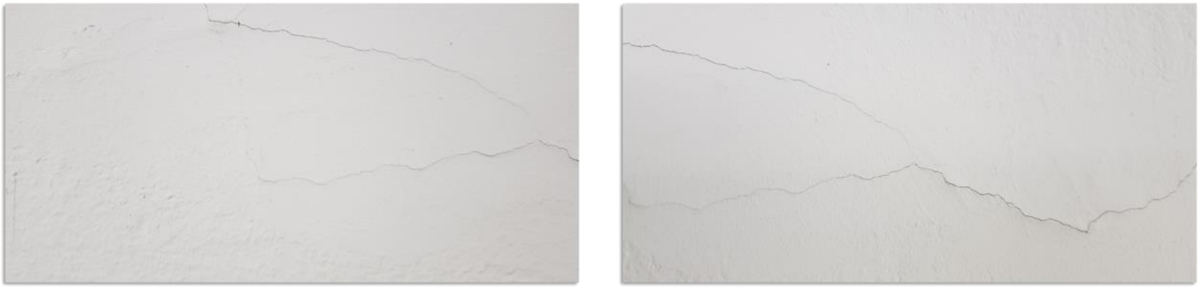


Figure 8. Lab scale experimental setup (a) Image capturing and geo-tagging system (b) Camera zoom view of extracted crack damage (c) Reference of crack scale dimension

For indoor testing, the ICGT system was mounted on a tripod stand, and three-dimensional positional data were acquired by manually measuring the actual translation of the camera without the GPS. The initial point was marked as the origin, and translation in any direction was recorded to track the relative positional coordinates. Initially, the system was placed at the origin, and then translated along the x-axis (toward the right) with a minor rotation. For better understanding, a close-up view of the camera viewpoint is shown in Figure 8(b), and a referenced measuring scale with the crack length in a real wall is given in Figure 8(c) for comparison with the transformation results. The two images of cracks captured by the translating and rotating camera are shown in Figure 9.


(a) Left side image

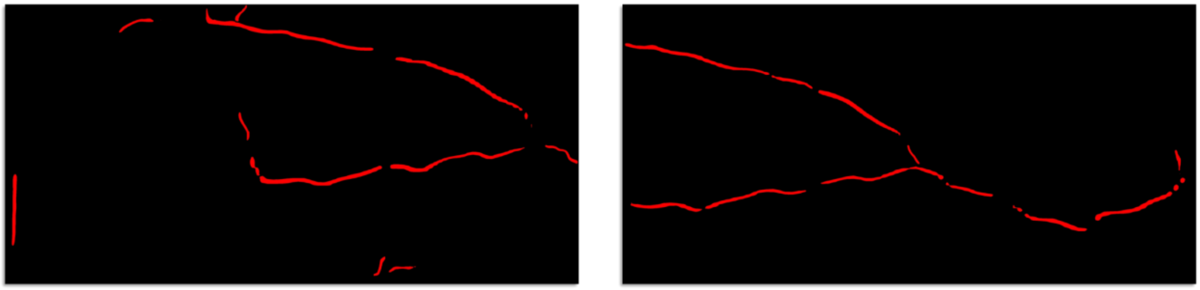
(b) Right side image

Figure 9. Images taken for lab-scale validation

Lab-scale Validation: Experimental Results

For every image, the corresponding georeferenced information is tagged and stored in the memory of the ICGT system. Before being fed to the CNN, the images were sliced into smaller resolutions as a post-inspection procedure for detecting possible minor cracks. The CNN inference is performed for offline damage detection, and the inference results are stored as vectors with instance segmentation of crack masks. The output of the Mask R-CNN-based crack detection was spliced together for every chopped image to reconstruct their original dimensions, and the results of the two images are shown in Figure 10 for indoor testing. Once the crack damages were detected and the masks were extracted, the image transformation matrix was calculated based on the georeferenced data gathered from the ICGT. The transformation matrix defines the homography of mask vectors, and the global coordinates are calculated using the pin-hole camera model. The procedure is repeated for every image, and the corresponding projection is performed in the world plane to map the results. In this test, translation was applied with 50% overlap, and due to close image acquisition, the rotation matrix was considered as unity for lab-scale testing.

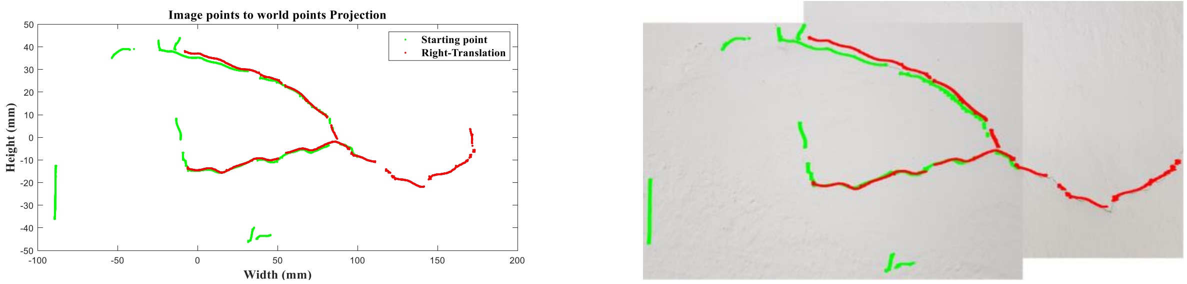
The results of CNN detection are mapped on to global coordinates using a coordinate transform and spliced together to create a mesh of cracks. The transformed image (Figure 11(a)) can be validated with a reference image (Figure 8(c)) showing the crack to be 20-cm long. In addition, Figure 11(b) demonstrate the results overlaid on top of the acquired image (see Figure 8(b)) for better realization of crack maps in the real world. Offline post-processing was performed on the MATLAB Computer Vision Toolbox and Python OpenCV Library in a system equipped with Intel i5-8500 CPU and 32 GB RAM. The average computational time for crack detection and crack surface map generation on 2 × 4K resolution images was 2.57 s, which can be improved with GPU-accelerated systems. The crack detection results using a GPU-equipped server (details presented earlier) took approximately 1.5 s for the same 2 × 4K images.



(a) Left side image

(b) Right side image

Figure 10. Mask RCNN prediction results



(a)

(b)

Figure 11. Image transformation results (a) Image to world plane (b) Crack map for real-world image

Field-scale Validation: Experimental Setup

The field-scale experiments were conducted on a bridge inspection site in Chuncheon, South Korea, using the ICGT system in a UAV. The hardware platform was a customized commercial UAV with an ICGT mounted on its gimbal at the front. The advantage of a custom drone is that its gimbal is designed to rotate 180° from top to bottom as well as left to right of the deck, enabling the camera to cover all possible angles in underside inspection. The hardware specifications of the UAV are listed in Table 5, and the ICGT-integrated UAV is shown in Figure 12. The onboard computer has an Intel i7-8559U processor with 16 GB RAM and an ROS that acts as an intermediate interface between the UAV and the ICGT. The UAV can perform underside bridge inspection and handle GPS-prohibited environment using a fusion of GPS and Velodyne three-dimensional LiDAR data [51].

Table 5. Hardware specifications of UAV integrated ICGT module.

Component Name	Component Type
Processor	Intel NUC Kit NUC817BEH
Flight controller	DJI A3 Pro
LiDAR module	Velodyne Puck LITE 100 m measurement range ±3 cm range accuracy
Transmission	DJI Lightbridge 2
Communication distance	3 km

Power supply module	2 × LiPo 6S 30000 mAh
Camera Gimbal Specifications	
Roll	±90°
Pitch	±90°
Motor	GBM6212H-150T
Payload range	1500–2800 g

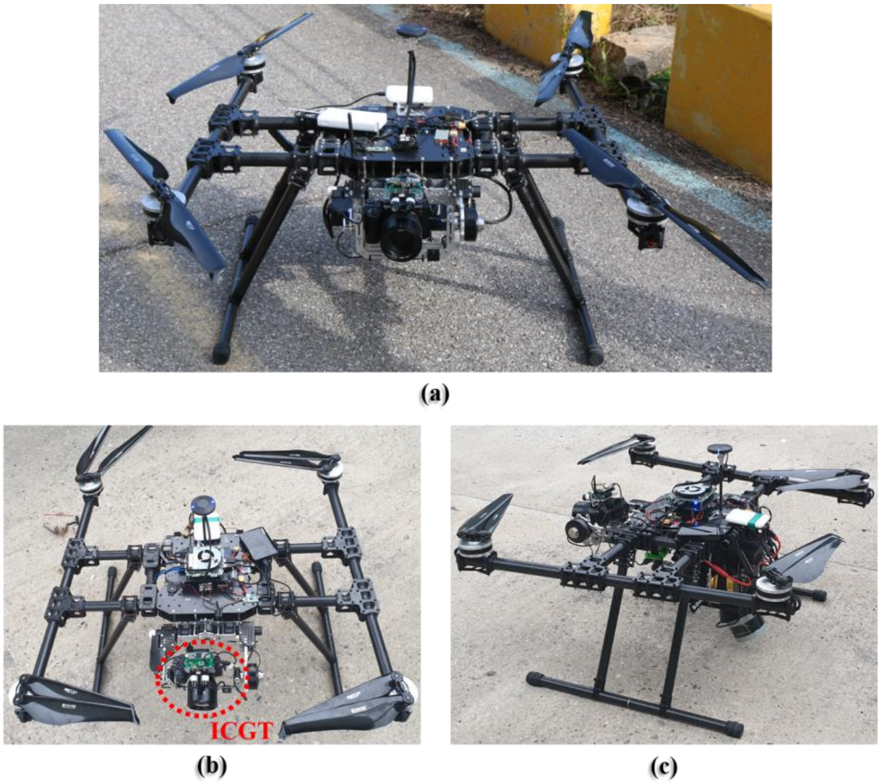


Figure 12. Hardware integration of UAV with ICGT system (a) Front view (b) Top view (c) Side view

The outdoor inspection was carried on the bridge deck. The side and bottom of the bridge as well as the piers were inspected for crack damage. The general view of the bridge under inspection is shown in Figure 13. The height and width of the bridge are 18 m and 7 m, respectively, with a span of 50 m between adjacent piers. The bridge deck as well as piers were considered for inspection, and the UAV was hovering at approximately 0.8 m/s to achieve 60% overlap between consecutive images.

Inspection procedure

The inspection is divided into three steps: flight take-off, inspection, flight landing and acquisition of inspection data (Figure 13).

Step 1: The UAV takes off under normal flight conditions and maneuvers to the target structure for inspection. During this state, the UAV starts hovering while the ICGT system is initialized, and the

ROS boots up. All communication between the UAV and the ground station is remotely controlled via rosservice through Team Viewer.

Step 2: The bridge is inspected for damages. The UAV acts as a master unit, while the ICGT is the slave unit controlled to start and stop data acquisition as well as image capture. Once it has reached the target structure, the “Start” command is transmitted to the drone, initiating the ICGT and data acquisition in parallel with image capture at 5 fps. The image capture as well as sensor data are stored simultaneously in memory with timestamps to geotag the corresponding image with its respective georeferenced data. The process flow of the bridge pier inspection is shown in Figure 13.

Step 3: After the inspection, the flight is prepared for landing. A “Stop” command is transmitted to cease and reset the ICGT data acquisition, and the flight lands on the ground. Once the inspection is completed and the drone has landed, the captured images and sensor data are collected from the ICGT for offline post-processing.

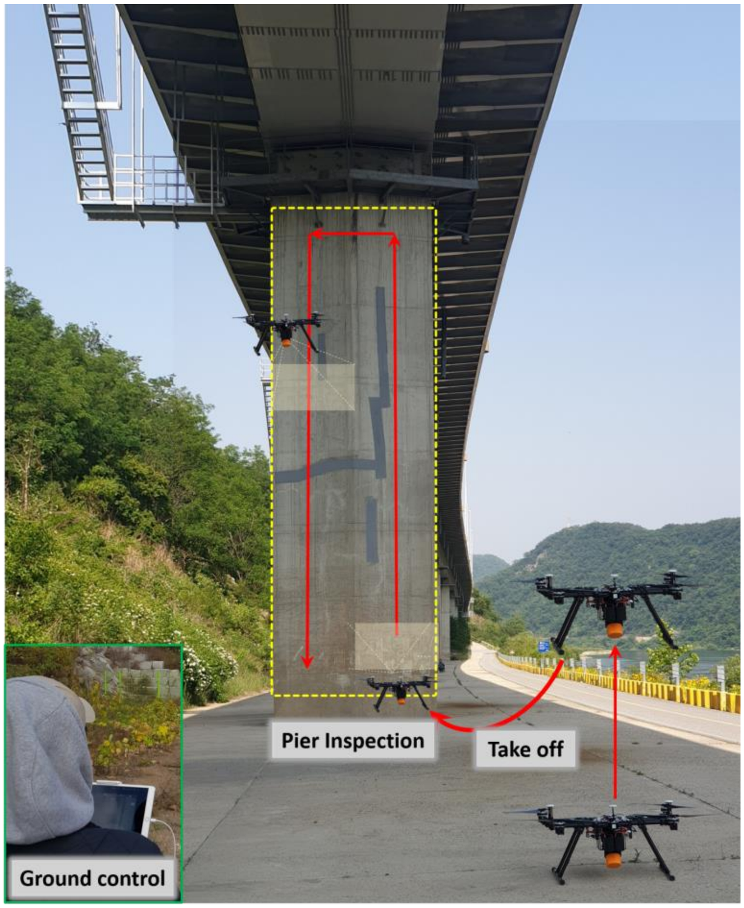


Figure 13. Experimental setup and inspection procedure of bridge pier

Experimental results

The inspection is performed vertically from the bottom-right to the top and continued from the top to the bottom-left along the bridge column (see Figure 13). During the test, the speed was kept constant, to maintain an overlap of 60%, for fast image mapping. The acquired images had a resolution of 3008 × 2000 pixels(Figure 14(a) – (c)), and 512 × 512 patches of each image were fed directly to the Mask R-CNN for crack damage detection. The results of the Mask R-CNN damage vectors are shown in Figure 14(d)–(f).

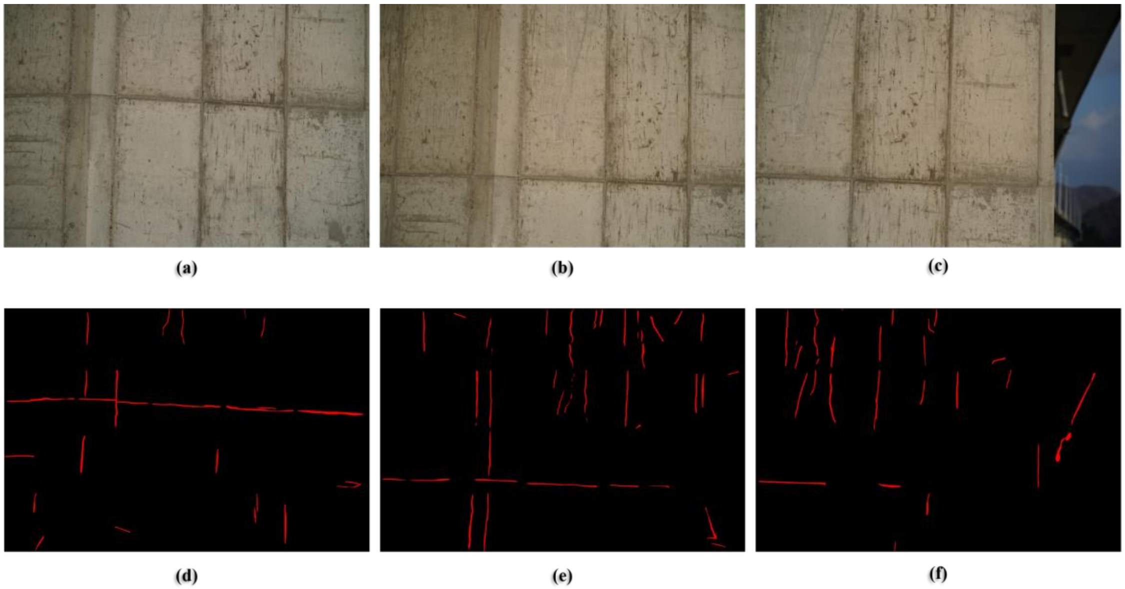


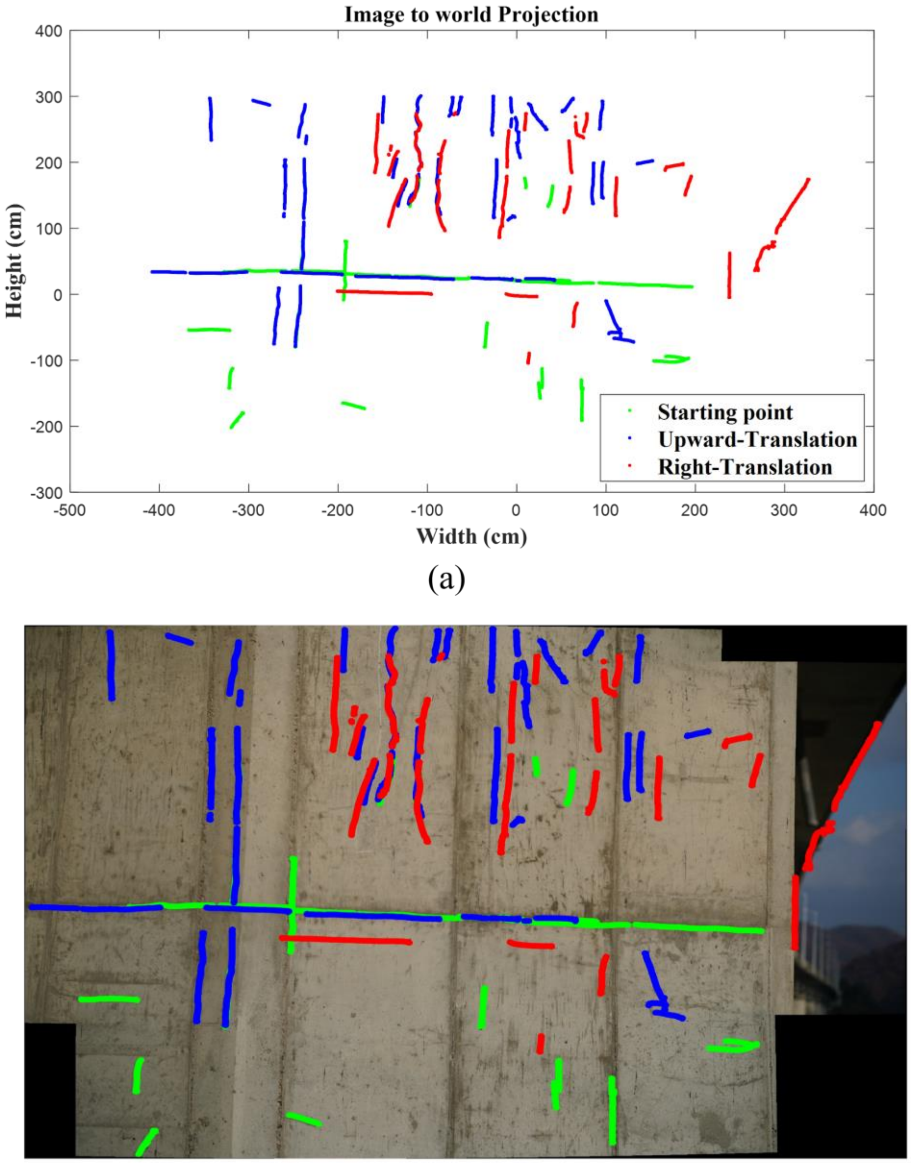
Figure 14. Images acquired from field inspection (a) Referenced origin image (b) Upward translated image from referenced image (c) Upward-right translated image

The experiments were performed on three images of the bridge piers and eight images of the bridge deck for crack localization. For data synchronization, an RTC was enabled to synchronize the IMU and LiDAR data with the camera images based on their timestamps. The results of the bridge pier are presented in Figure 15, and the results of the bridge deck are presented in Figure 16. Each image is paired with its corresponding IMU, LiDAR, and GPS data synchronously. Table 6 shows an example of the ICGT log file generated after synchronization with timestamps for three images taken as the field-scale validation. The 1D-LiDAR distance was kept constant at 2–3 m from the bridge pier and the IMU and 1D-LiDAR data are synchronized with the camera images using timestamps from the camera unit and the RTC on the ICGT.

Table 6. Georeferenced data from ICGT

No.	Filename	Time	Distance (m)	Longitude	Latitude	Altitude	Yaw	Roll	Pitch
1	DSC07462.JPG	16:31:38	2.49	127.6512204	37.82339	99.18125	359.94	0.12	-0.25
2	DSC07463.JPG	16:31:38	2.47	127.6512204	37.82339	99.18125	359.94	0.08	-0.25
3	DSC07464.JPG	16:31:39	2.42	127.6512207	37.82339	99.17252	359.81	0.08	-0.06

The relative image transformation of local coordinates to global coordinates with the images are showed in Figure 15(a). In the end, the Mask-RCNN-transformed results are projected onto the field images, according to the degree of orientation during the field test, for instant masking of cracks in the world coordinate system (Figure 15(b)). Based on crack prediction from the Mask R-CNN model, as shown in Figure 14(d)–(f), most cracks were well preserved and detected, but a few cracks were FPs due to the sensitivity of the model. To reduce FPs, the new dataset must be enhanced with improved training by including images with crack-like textures. In addition, Figure 16 shows the GBDM of the bottom deck of the bridge from the left side. The proposed method with the image slicing procedure could identify crack damages that went undetected with existing approaches. The input image with 3008 × 2000 resolution did not show any damage, while the 512 × 512 ones showed minor cracks. Image slicing is thus critical to identifying cracks and improve damage detection, albeit with the disadvantage of FPs.



(a)

(b)

Figure 15. Image transformation results (a) Image to world plane (b) GBDM (pier)

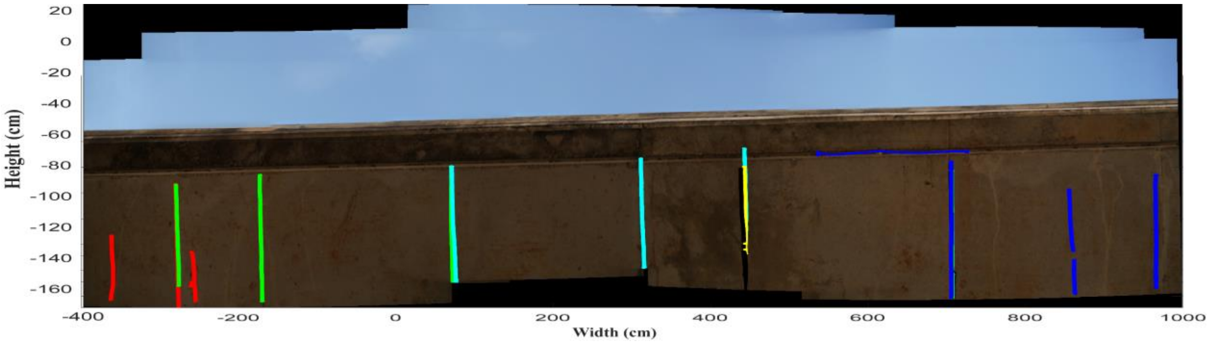


Figure 16. GBDM of deck (Bottom side)

Discussion and Comparison on Implementation Time

The computational and measurement speed of the proposed method supported fast image analysis and damage mapping at flight speeds of 0.8 m/s and a camera frame rate of 5 fps, while covering a bridge section of 1.2 m × 13 m. The computational time for the Mask RCNN-based detection for a single 2K image was approximately 1.722 s and 2.77 s for a 4K image with a CPU-powered system defined earlier in the text. Similarly, the computational time for image projection of 4K images was 5.9 s for a total of three images used for lab-scale validation. Crack detection for an average bridge with 100 high-resolution images can take approximately 14 min. In addition, crack detection can result in some false positives as shown in the GBDM of bridge pier (see Figure 15(b)). The FPs are the result of concrete formwork making crack like pattern and these can be minimized by adding another class for formwork crack during training to upskill the neural network. Table 7 showed timing comparison of the proposed method with state-of-the-art robotics platforms for bridge inspection.

Table 7. Timing comparison between state-of-the-art systems

Literature	Area (m^2)	Time (min)
Proposed method	1.2 × 13	14
ROCIM [14]	4 × 2.5	23
RABIT [7]	10.6 × 1	33

Conclusions

This paper proposed instant crack damage detection using an ICGT with deep CNN for automated inspection. The system was mounted on a drone for aerial inspection. First, the ICGT was developed to control a camera shutter, and acquire GPS, IMU, and 1D-LiDAR data for generating georeferenced inspection images. Second, a type of deep CNN, the Mask R-CNN method was tailored for rapid and accurate crack identification using the bridge inspection images. Lastly, the Mask R-CNN prediction results from each inspection image were directly projected on to the global coordinate system, where the projected damages from each image were combined to generate a GBDM.

The key findings of the study are as follows:

- ICGT system for acquiring data from GPS, IMU and LiDAR, while controlling a camera shutter to capture georeferenced images with synchronized timestamps.

- A ICGT interface designed flexibly to work with ROS, which allows communication between any master units running the ROS.
- A recall of 81.6% and a mAP of 77.5% for dataset-2 with proposed configuration.
- Through experimental validation, the proposed method successfully projected damage on to a scaled global damage map using the developed Mask R-CNN and georeferenced data from the ICGT. A lab-scale experiment projected two images containing continuous crack paths on to a global damage map, directly constructing a scaled crack propagation path using the ICGT and trained Mask R-CNN. A field experiment using a UAV also generated a scaled damage map of a bridge pier and a bridge deck with cracks through ICGT interfaced with the UAV and trained Mask R-CNN model.
- The computational time for crack damage prediction using a CPU-powered system from one 2K-resolution image is 1.722 s, while the same system takes 5.9 s to project three such images for field validation.

Manual damage inspection often consumes a lot of time to generate damage report. However, the proposed system can be used as an automatic inspection technique adopted to obtain a generalized overview of the bridge structure. The proposed method is a first-of-its-kind approach to realize the UAV–ICGT functionality with a deep CNN detection algorithm. The automated inspection approach was implemented at the lab-scale as well as validated in the field. The performance of the new system was better than that of existing approaches, as confirmed through the experimental evaluation. The system with all available resources will be provided as open-source material (reference link: *https://github.com/rakehsaleem*) after the publication of the paper, for future collaboration and improvements.

In the future, this system can be extended to multiclass damage detection such as corrosion, spalling, efflorescence, and formwork. An extension to this approach for damage detection would be to project the image on to a three-dimensional point cloud and using building information modeling (BIM) to advance damage detection for bridges.

Declaration of interests

The author(s) declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding sources

The author(s) disclosed the receipt of the following financial support for the research, authorship and/or publication of this article. This research was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) funded by the MSIT (2019R1F1A1059393) and by the Chung-Ang University Young Scientist Scholarship (CAYSS) grants in 2019.

References

[1] Y. Dong, R. Song, and H. Liu, “Bridges Structural Health Monitoring and Deterioration Detection Synthesis of Knowledge and Technology,” Alaska University Transportation Center, Alaska Department of Transportation and Public Facilities, Report, 2010. Accessed: Dec. 13, 2019. [Online]. Available: https://scholarworks.alaska.edu:443/handle/11122/7476.

[2] S. Dorafshan and M. Maguire, "Bridge inspection: human performance, unmanned aerial systems and automation," *J Civil Struct Health Monit*, vol. 8, no. 3, pp. 443–476, Jul. 2018, doi: 10.1007/s13349-018-0285-4.

[3] C. A. Chua and P. Cawley, "Crack growth monitoring using fundamental shear horizontal guided waves," *Structural Health Monitoring*, pp. 779–792, Oct. 2019, doi: 10.1177/1475921719882330.

[4] H. Kim, E. Ahn, M. Shin, and S.-H. Sim, "Crack and Noncrack Classification from Concrete Surface Images Using Machine Learning," *Structural Health Monitoring*, vol. 18, no. 3, pp. 725–738, May 2019, doi: 10.1177/1475921718768747.

[5] R. Ali and Y.-J. Cha, "Subsurface damage detection of a steel bridge using deep learning and uncooled micro-bolometer," *Construction and Building Materials*, vol. 226, pp. 376–387, Nov. 2019, doi: 10.1016/j.conbuildmat.2019.07.293.

[6] I.-H. Kim, H. Jeon, S.-C. Baek, W.-H. Hong, and H.-J. Jung, "Application of Crack Identification Techniques for an Aging Concrete Bridge Inspection Using an Unmanned Aerial Vehicle," *Sensors*, vol. 18, no. 6, p. 1881, Jun. 2018, doi: 10.3390/s18061881.

[7] P. Prasanna *et al.*, "Automated Crack Detection on Concrete Bridges," *IEEE Trans. Automat. Sci. Eng.*, vol. 13, no. 2, pp. 591–599, Apr. 2016, doi: 10.1109/TASE.2014.2354314.

[8] A. Ortiz, F. Bonnin-Pascual, E. Garcia-Fidalgo, and J. P. Company-Corcoles, "Vision-Based Corrosion Detection Assisted by a Micro-Aerial Vehicle in a Vessel Inspection Application," *Sensors*, vol. 16, no. 12, p. 2118, Dec. 2016, doi: 10.3390/s16122118.

[9] Z. Zhu, S. German, and I. Brilakis, "Visual retrieval of concrete crack properties for automated post-earthquake structural safety evaluation," *Automation in Construction*, vol. 20, no. 7, pp. 874–883, Nov. 2011, doi: 10.1016/j.autcon.2011.03.004.

[10] J.-W. Kim, S.-B. Kim, J.-C. Park, and J.-W. Nam, "Development of Crack Detection System with Unmanned Aerial Vehicles and Digital Image Processing," p. 11.

[11] Catbas F. Necati, Brown David L., and Aktan A. Emin, "Parameter Estimation for Multiple-Input Multiple-Output Modal Analysis of Large Structures," *Journal of Engineering Mechanics*, vol. 130, no. 8, pp. 921–930, Aug. 2004, doi: 10.1061/(ASCE)0733-9399(2004)130:8(921).

[12] P. J. Sanchez-Cuevas, P. Ramon-Soria, B. Arrue, A. Ollero, and G. Heredia, "Robotic System for Inspection by Contact of Bridge Beams Using UAVs," *Sensors*, vol. 19, no. 2, p. 305, Jan. 2019, doi: 10.3390/s19020305.

[13] B. J. Lee, D. H. Shin, J. W. Seo, J. D. Jung, and J. Y. Lee, "Intelligent Bridge Inspection Using Remote Controlled Robot and Image Processing Technique," *ISARC Proceedings*, pp. 1426–1431, Jun. 2011.

[14] R. S. Lim, H. M. La, and W. Sheng, "A Robotic Crack Inspection and Mapping System for Bridge Deck Maintenance," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 367–378, Apr. 2014, doi: 10.1109/TASE.2013.2294687.

[15] B. Li, K. Ushiroda, L. Yang, Q. Song, and J. Xiao, "Wall-climbing robot for non-destructive evaluation using impact-echo and metric learning SVM," *Int J Intell Robot Appl*, vol. 1, no. 3, pp.

255–270, Sep. 2017, doi: 10.1007/s41315-017-0028-4.

[16] Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning–based data anomaly detection method for structural health monitoring:," *Structural Health Monitoring*, Feb. 2018, doi: 10.1177/1475921718757405.

[17] J. Huang, D. Li, H. Li, G. Song, and Y. Liang, "Damage identification of a large cable-stayed bridge with novel cointegrated Kalman filter method under changing environments," *Structural Control and Health Monitoring*, vol. 25, no. 5, p. e2152, 2018, doi: 10.1002/stc.2152.

[18] M. Mohtasham Khani *et al.*, "Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines," *Structural Health Monitoring*, p. 1475921719883202, Nov. 2019, doi: 10.1177/1475921719883202.

[19] R. Davoudi, G. R. Miller, and J. N. Kutz, "Computer Vision Based Inspection Approach to Predict Damage State and Load Level for RC Members," *Structural Health Monitoring 2017*, vol. 0, no. shm, 2017, doi: 10.12783/shm2017/14225.

[20] A. Khaloo, D. Lattanzi, A. Jachimowicz, and C. Devaney, "Utilizing UAV and 3D Computer Vision for Visual Inspection of a Large Gravity Dam," *Front. Built Environ.*, vol. 4, 2018, doi: 10.3389/fbuil.2018.00031.

[21] K.-B. Lee and H.-S. Shin, "An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels," *2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML)*, pp. 7–11, Aug. 2019, doi: 10.1109/Deep-ML.2019.00010.

[22] R. Parlange, "Vision-based autonomous navigation for wind turbine inspection using an unmanned aerial vehicle," 2019.

[23] A. S. M. Shihavuddin *et al.*, "Wind Turbine Surface Damage Detection by Deep Learning Aided Drone Inspection Analysis," *Energies*, vol. 12, no. 4, p. 676, Jan. 2019, doi: 10.3390/en12040676.

[24] B. F. Spencer, V. Hoskere, and Y. Narazaki, "Advances in Computer Vision-Based Civil Infrastructure Inspection and Monitoring," *Engineering*, vol. 5, no. 2, pp. 199–222, Apr. 2019, doi: 10.1016/j.eng.2018.11.030.

[25] D. Kang and Y.-J. Cha, "Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 10, pp. 885–902, 2018, doi: 10.1111/mice.12375.

[26] K. Jang, N. Kim, and Y.-K. An, "Deep learning–based autonomous concrete crack evaluation through hybrid image scanning," *Structural Health Monitoring*, vol. 18, no. 5–6, pp. 1722–1737, Nov. 2019, doi: 10.1177/1475921718821719.

[27] M. A. Akbar, U. Qidwai, and M. R. Jahanshahi, "An evaluation of image-based structural health monitoring using integrated unmanned aerial vehicle platform," *Struct Control Health Monit*, vol. 26, no. 1, p. e2276, Jan. 2019, doi: 10.1002/stc.2276.

[28] B. Lei, Y. Ren, N. Wang, L. Huo, and G. Song, "Design of a new low-cost unmanned aerial vehicle

and vision-based concrete crack inspection method:," *Structural Health Monitoring*, Feb. 2020, doi: 10.1177/1475921719898862.

[29] S. Song, S. Jung, H. Kim, and H. Myung, "A method for mapping and localization of quadrotors for inspection under bridges using camera and 3 D-LiDAR," 2019.

[30] C. Brooks *et al.*, "Evaluating the Use of Unmanned Aerial Vehicles for Transportation Purposes," Apr. 2015, Accessed: Mar. 02, 2020. [Online]. Available: https://trid.trb.org/view/1352503.

[31] H. M. La *et al.*, "Mechatronic Systems Design for an Autonomous Robotic System for High-Efficiency Bridge Deck Inspection and Evaluation," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1655–1664, Dec. 2013, doi: 10.1109/TMECH.2013.2279751.

[32] S. Gibb, H. M. La, T. Le, L. Nguyen, R. Schmid, and H. Pham, "Nondestructive evaluation sensor fusion with autonomous robotic system for civil infrastructure inspection," *Journal of Field Robotics*, vol. 35, no. 6, pp. 988–1004, 2018, doi: 10.1002/rob.21791.

[33] Firoozi Yeganeh Sayna, Golroo Amir, and Jahanshahi Mohammad R., "Automated Rutting Measurement Using an Inexpensive RGB-D Sensor Fusion Approach," *Journal of Transportation Engineering, Part B: Pavements*, vol. 145, no. 1, p. 04018061, Mar. 2019, doi: 10.1061/JPEODX.0000095.

[34] S. Jiang and J. Zhang, "Real-time crack assessment using deep neural networks with wall-climbing unmanned aerial system," *Computer-Aided Civil and Infrastructure Engineering*, vol. n/a, no. n/a, doi: 10.1111/mice.12519.

[35] R. Wang and Y. Kawamura, "An Automated Sensing System for Steel Bridge Inspection Using GMR Sensor Array and Magnetic Wheels of Climbing Robot," *Journal of Sensors*, 2016. https://www.hindawi.com/journals/js/2016/8121678/ (accessed Mar. 02, 2020).

[36] H. M. La, "Automated Robotic Monitoring and Inspection of Steel Structures and Bridges," *arXiv:1705.04888 [cs]*, May 2017, Accessed: Mar. 02, 2020. [Online]. Available: http://arxiv.org/abs/1705.04888.

[37] A. Moussa and N. El-Sheimy, "A FAST APPROACH FOR STITCHING OF AERIAL IMAGES," *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. XLI-B3, pp. 769–774, Jun. 2016, doi: 10.5194/isprsarchives-XLI-B3-769-2016.

[38] Q. Liu, W. Liu, L. Zou, J. Wang, and Y. Liu, "A NEW APPROACH TO FAST MOSAIC UAV IMAGES," in *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Sep. 2012, vol. XXXVIII-1-C22, pp. 271–276, doi: https://doi.org/10.5194/isprsarchives-XXXVIII-1-C22-271-2011.

[39] B. Sensortec, "Intelligent 9-axis absolute orientation sensor," p. 106, 2016.

[40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time

Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[42] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *arXiv:1512.02325 [cs]*, vol. 9905, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[43] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv:1703.06870 [cs]*, Jan. 2018, Accessed: Dec. 08, 2019. [Online]. Available: http://arxiv.org/abs/1703.06870.

[44] S. Dorafshan, R. J. Thomas, and M. Maguire, "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks," *Data Brief*, vol. 21, pp. 1664–1668, Nov. 2018, doi: 10.1016/j.dib.2018.11.015.

[45] A. N. S, *ImageScraper: A simple image scraper to download all images from a given url*. .

[46] "VGG Image Annotator." http://www.robots.ox.ac.uk/~vgg/software/via/via_demo.html (accessed Dec. 15, 2019).

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385 [cs]*, Dec. 2015, Accessed: Dec. 09, 2019. [Online]. Available: http://arxiv.org/abs/1512.03385.

[48] B. Kim and S. Cho, "Image-based concrete crack assessment using mask and region-based convolutional neural network," *Structural Control and Health Monitoring*, vol. 26, no. 8, p. e2381, 2019, doi: 10.1002/stc.2381.

[49] "Camera Calibration Toolbox for Matlab." http://www.vision.caltech.edu/bouguetj/calib_doc/ (accessed Dec. 08, 2019).

[50] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000, doi: 10.1109/34.888718.

[51] F. Vanegas, K. J. Gaston, J. Roberts, and F. Gonzalez, "A Framework for UAV Navigation and Exploration in GPS-Denied Environments," in *2019 IEEE Aerospace Conference*, Mar. 2019, pp. 1–6, doi: 10.1109/AERO.2019.8741612.