An Innovative UAV-Bridge Crack Detection System Using Deep Learning

Anthony Stone[1] and Alaa Sheta[1]

[1]*Computer Science Department, Southern Connecticut State University, CT, USA*
shetaa1@southernct.edu,stonea13@southernct.edu

Abstract

Concrete bridge degradation and cracking are phenomena prevalent among a significant portion of bridges in the United States. The risks and high costs associated with bridge repair and restoration necessitate effective structural health monitoring paradigms that rely on accurate, timely, and efficient inspection methodologies. This study proposes an efficient inspection system based on unmanned aerial vehicles and computer vision. We use a specialized dataset of bridge crack visual data to train a convolutional neural network to classify images as containing a crack or not. We adopt a lightweight transfer-learning approach, leveraging the EfficientNetB0 model for rapid model development and deployment. We deploy this model for application in a case study on bridge health inspection, using a modeled concrete bridge in a simulated environment. The RYZE DJI Tello drone was used to navigate the space and capture optical data for real-time autonomous crack detection. The proposed methodology demonstrates the efficacy of such an approach in structural health monitoring.

Keywords: Bridge Crack Inspection; Computer Vision; Convolutional Neural Network; Unmanned Aerial Vehicle; Structural Health Monitoring;

1. Introduction

The United States is home to more than 620,000 bridges, many of which are legacy infrastructure assets that have stood for several decades. Aging and continuous use subject these spans to deterioration, thereby posing an increasing threat to travelers' safety. The American Road and Transportation Builders Association (ARTBA) is a federation that compiles comprehensive reports and analyses on the condition of bridges in the U.S. ARTBA reported that 220,295 bridges need repair, including 74,472 bridges that need to be replaced entirely. The condition of spans in the U.S. as of 2025 is realized in Figure 1. A rating of "poor" indicates that at least one key component of the bridge is in poor condition. ARTBA estimates that the cost associated with conducting all of the identified repairs is around $467 billion [1].
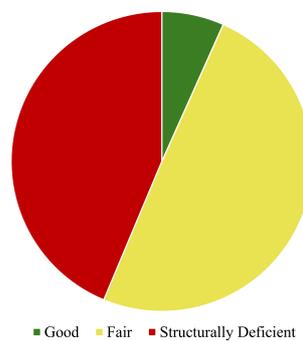

Figure 1: U.S. Bridges by Rating

A strategy for assessing the health and reliability of bridge infrastructure to ensure safe usage is necessary. This is known as structural health monitoring (SHM), which is an ongoing process involving the inspection, data processing, and analysis of infrastructure conditions [2]. To establish an effective SHM paradigm, the quality of these bridges must be assessed through regular, accurate, and robust examinations. In doing so, signs of degradation may be detected early, reducing the time required to take restorative measures and, in turn, lessening the overall financial burden associated with bridge maintenance.

Traditional methodology requires that an engineer or construction worker physically access the inspection site and conduct a manual/visual inspection of the bridge. It is often the case that a targeted area cannot be reached by simply approaching, so alternative techniques must be considered. Rope access is a method in which trained professionals use climbing equipment to conduct inspections. Trucks and aquatic vehicles equipped with lifts are another means of aiding challenging bridge health evaluations.

These traditional inspection strategies often compromise the safety of inspection professionals, who are exposed to working at heights, confined spaces, and other hazardous conditions. This methodology is also time-consuming, slowing the overall SHM workflow and risking that serious structural concerns go unnoticed for extended periods. Manual/visual inspection also suffers from subjectivity, as the surface health assessment is performed by a human and is thus prone to human error [3]. Wages, operational equipment, prolonged traffic interruptions, and assessment inaccuracies significantly increase the already steep cost of inspection and repair [4].

The challenges of safety, time efficiency, accuracy, and financial burden underscore the urgent need for alternative inspection techniques for SHM of bridges. Implementation of a method that provides accurate and timely detection of bridge damage, mitigates the risk of prolonged structural degradation, and thus better preserves the service life of the bridge. A more sophisticated monitoring technique will mitigate the hazards of manual inspection while reducing the overall cost of the SHM system.

A solution to the hazardous, inefficient, and costly nature of bridge inspection is to use unmanned aerial vehicles (UAVs) or drones and computer vision (CV) to compose an inspection system. A drone that can be piloted to an assessment area eliminates the need for a professional to physically approach the site. This, in turn, prevents the inspector from being placed in unsafe positions. The need for additional climbing equipment or large lift machinery has been eliminated, thereby significantly reducing inspection costs. This has the added benefit of allowing inspection without stopping or detouring traffic, thereby promoting efficiency. By leveraging CV, surface health assessments can be conducted rapidly while mitigating human subjectivity. Using UAVs and CV enables a safe, efficient, and accurate SHM paradigm that incurs a fraction of the typical cost.

This paper demonstrates the efficacy of adopting a UAV-based inspection methodology for assessing concrete bridge cracking. We show that such an inspection can be implemented using a relatively inexpensive UAV with simple hardware and a lightweight convolutional neural network (CNN), thereby significantly increasing computational and financial efficiency. This paper is organized as follows: Section 2 provides a review and summary of previous related works. Section 3 presents the proposed methodology and system workflow. The experimental setup, results, and scope are discussed in Section 4. Lastly, we conclude in Section 5.

2. Related work

To address the risks, costs, and efficiency problems associated with traditional inspection methodologies, researchers have investigated solutions grounded in UAVs. Generally, the drone is equipped with a camera and, via remote control, flown to different bridge elements [5]. The 'eyes in the sky' method involves displaying the UAV's streamed video feed on a screen at a ground station [4]. It was found in [6] that such UAV-integrated systems could achieve cost savings of up to 66%. While effective, this does not address the issue of subjectivity and the potentiality of human error in assessing bridge health.

Further enhancements to UAV-integrated SHM systems have been achieved through the use of computer vision (CV) techniques for processing and evaluating visual data collected by UAVs. SHM systems can leverage CV techniques to enable reliable inspection. Deploying a CV model for this task has the added benefit of allowing real-time detection and creating pipelines that are more efficient and scalable [7]. Advancements in artificial intelligence (AI) and deep learning (DL) have led to powerful CV techniques for detecting structural decline. CNNs are widely used because they provide an efficient approach to crack classification [8]. A variety of model architectures, including AlexNet, VGG, and ResNet, achieve high performance on crack detection tasks [9]. In application, these models demonstrate a profound ability to enhance SHM systems [4, 10, 11]. However, the size and complexity of these models can often make model training computationally expensive and time-consuming.

The use of transfer learning (TL) has been shown to be an effective approach, enabling the rapid development of high-performing DL models [12]. TL is the act of leveraging an already trained model for an adjacent task, reducing the overall training necessary to achieve desirable performance. Pretrained models are often trained on large-scale datasets such as ImageNet, resulting in highly expressive feature representations. In addition, pretrained models are readily available on open-source platforms, making the adoption and deployment of a TL model a viable alternative to building models from scratch [13].

Further exploration of UAV-based inspection examines the adoption of external equipment and the augmentation of UAVs with additional hardware components. Infrared thermography (IRT) cameras and light detection and ranging (lidar) sensors have been integrated into the workflow for generating three-dimensional bridge models [14]. These techniques can also be leveraged alongside onboard GPS to enable autonomous and semi-autonomous flights. However, implementing such systems in UAVs without GPS or navigation in GPS-denied environments remains challenging and remains an active area of research [15, 16]. When making UAV augmentations, battery life and flight performance must be considered, as adding sensors and cameras may increase a UAV's weight, thereby sacrificing battery life and flight performance [4]. UAVs with more powerful flight specifications, or UAVs equipped with sensors, are then selected for this purpose. External and internal components used to enhance autonomous control are often expensive, both computationally and financially [16].

3. UAV Bridge Crack Detect proposed system

The UAV Bridge Crack Detect (UAV-BCD) system is a UAV-based inspection system designed for efficient, rapid, and cost-effective detection of concrete bridge cracking. This system is composed of the following primary modules:

     Module 1: UAV-enabled data acquisition.
     Module 2: Cloud-based image processing and streaming.
     Module 3: Bridge crack detection model.

These modules operate together to enable real-time structural health assessments. A diagram of the proposed system workflow is shown in Figure 2, highlighting the key operations in each module.
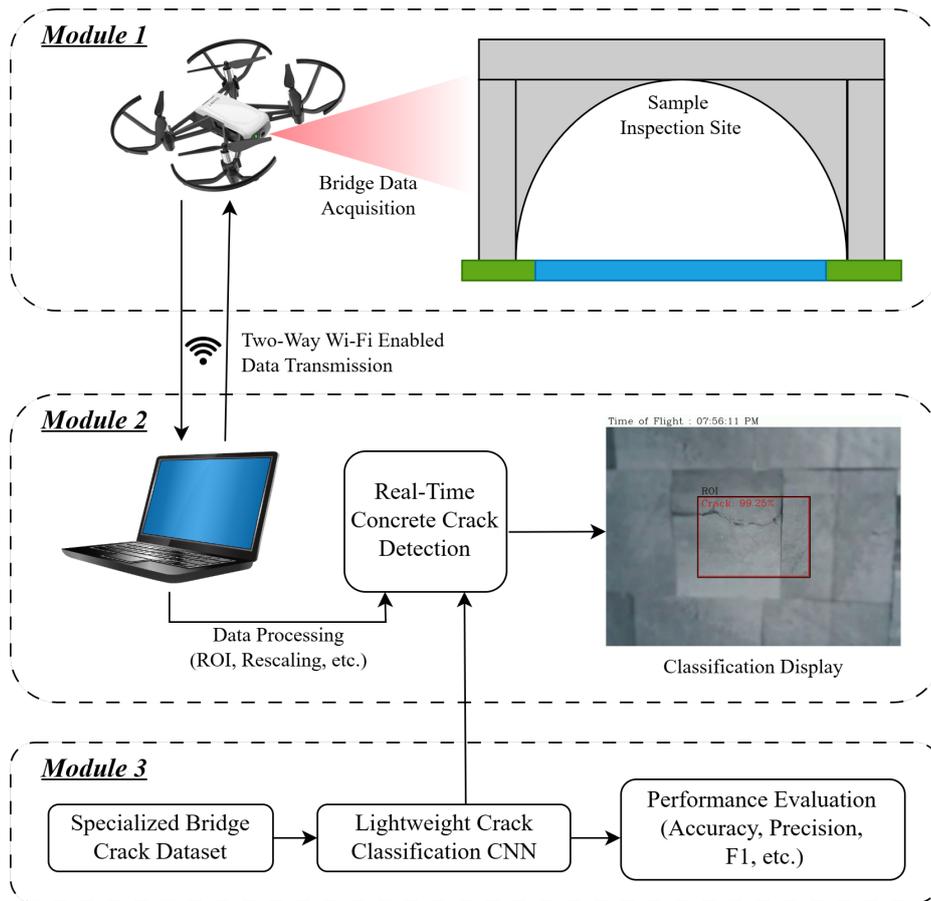
Figure 2: Diagram of the Proposed UAV-BCD Inspection System Workflow

3.1. UAV-enabled data acquisition

The UAV-enabled data-acquisition system is the first proposed module. Its role is to capture high-resolution visual data of the bridge surface and transmit it to the server for processing. In this module, UAV stability and control are considered and developed to ensure proper operation of subsequent phases of system operations.

3.1.1. UAV platform

We adopted a lightweight DJI Tello RYZE Drone as the prime aerial imaging platform. The Tello is a quadcopter weighing about 80 grams in flight and capable of up to 13 minutes of flight time. The drone is equipped with a 2.4 GHz Wi-Fi for data transmission. Figure 3 depicts the DJI Tello adopted for the realization of the proposed system.

Figure 3: DJI Tello Drone[17]

The Tello is equipped with two cameras. A downward-facing camera employs a vision system and a barometer to maintain positional stability during flight and execute corrections when disturbed. The secondary camera is oriented forward, providing an 82.6° field of view and recording HD 720p video at up to 30 frames per second in MP4 format. Additionally, the front camera features electronic image stabilization (EIS), enabling high-quality video during flight [17]. The Tello Software Development Kit (SDK) provides programming capabilities used to develop the proposed system.

3.1.2. Video acquisition

The UAV collects a high-resolution video of the bridge during flight. The video stream is sampled at 30 fps and then fed to the recognition system for crack detection. The DJI Tello's camera resolution was sufficient for the mission. The EIS helps ensure that visual capture is clean and unaffected by UAV flight turbulence.

3.1.3. Data transmission

The UAV's built-in Wi-Fi channel sends video data to the server in real time. This reduces the time-to-system-response after data reception, enabling real-time processing and crack detection.

3.1.4. Stabilization features

The DJI Tello lacks a mechanical gimbal but includes an Inertial Measurement Unit (IMU). The IMU communicates with the internal computer to estimate the UAV's location and motion. This helps the drone fly smoothly and respond to disturbances. This is helpful when working in confined spaces or in light winds.

3.1.5. Flight operation and safety considerations

A joystick-style controller was used to manually control the drone's movement. When operated manually, the drone offers greater flexibility, allowing the user to position it accurately near critical bridge structures. This operational strategy is particularly advantageous in locations where GPS signals may be attenuated or obstructed by a bridge's configuration.
Safety is key in UAV-assisted bridge inspection. The drone operator maintains a direct line of sight throughout the mission to avoid obstacles and comply with regulations. In addition, a graphical user interface (GUI) is implemented, providing the operator with a first-person perspective. Vehicles, pedestrians, and structural elements that may collide are kept away during inspections. Preflight checklists include battery, propeller, and communication stability checks before each mission.

3.2. Cloud-based image streaming and processing server

The second portion of the proposed system is the cloud-based image streaming and processing server, a critical component of the inspection pipeline. This module connects the UAV to the network to find cracks. It enables the CNN model to capture, process, and deliver visual information in real time. The server allows for fast, precise fracture location during flight operations by optimizing streaming, buffering, and preprocessing.

3.2.1. Communication protocol

The DJI Tello uses the User Datagram Protocol (UDP) for Wi-Fi communication. Packet transmission is performed directly from the server device to the Tello's UDP port and maintains synchronization throughout the flight.

3.2.2. Streamed data preprocessing

To optimize the images fed into the CNN, we apply a simple preprocessing step on the server to focus on a specific region of interest (ROI) in each frame. We use the predefined ROI as a 360×260 px window centred in the UAV video capture. The spliced region is then resized to fit the CNN model's input size of 224×224. This ROI-based method minimizes unnecessary rescaling of the entire image, thereby preserving essential crack-related properties that could be compromised during resizing. Preserving local features enhances the quality of the CNN model's inputs and improves detection performance.

3.2.3. Buffering and concurrency management

Once the raw visual data is cropped, it is fed to the CNN for real-time classification. CNN classification calls entail computational cost and become relevant when executed concurrently with tasks such as UAV navigation and video writing. To prevent a bottleneck, the CNN is set to perform one classification every $1/30$ second. This matches the Tello's stream frame rate, ensuring that each frame is classified while avoiding a computationally expensive excess of CNN passes.

3.2.4. Python-based processing environment

The backbone of the proposed system is a Python-based processing script that integrates UAV piloting, video handling, and model inference into a unified workflow. Programming elements related to the Tello are developed using the open-source library DjitelloPy, a Python interface to the Tello SDK [18]. The OpenCV-Python library provides the necessary utilities for processing the visual data. We also use this toolbox to visualize and log the classified data.

3.3. CNN bridge crack detection and assessment model

A lightweight transfer-learning strategy was employed to build a convolutional neural network for Module 3 using the TensorFlow framework. The foundation of the proposed model is EfficientNetB0, a variant within the EfficientNet family in the TensorFlow library. This model was selected for its simplicity and satisfactory transfer performance [19]. Subsequent layers were incorporated to categorize an image of concrete as 'positive' if a crack is detected and 'negative' if none is detected. We assess the efficacy of the training and validation cases utilizing the

following statistical metrics: accuracy, precision, recall, and F1-score. The formulas for these measures are shown below, where TP, TN, FP, and FN denote the total counts of true positive, true negative, false positive, and false negative classifications.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

The EfficientNetB0 model is transferred and pretrained on the ImageNet dataset to serve as the model feature extractor. Fine-tuning of the pretrained weights is enabled to enhance problem specificity. A global average pooling layer is added, reducing the $7 \times 7 \times 1280$ EfficientNetB0 output to a 1280-dimensional vector. A fully connected (dense) layer with 256 units receives this vector and outputs a 256-dimensional vector with ReLU activation, thus introducing nonlinearity. To mitigate overfitting, a regularization technique known as dropout is employed. Input units are randomly deactivated at a rate of 0.5. Finally, another dense layer with softmax activation outputs class probabilities for the input visual data. The CNN architecture proposed is summarized in Table 1.

Table 1: Architecture Summary of the Proposed CNN Model

Layer (Type)	Output Shape	Parameters
EfficientNetB0	(None, 7, 7, 1280)	4,049,571
Global Avg. Pooling 2D	(None, 1280)	0
Dense (256, ReLU)	(None, 256)	327,936
Dropout (0.5)	(None, 256)	0
Dense (2, Softmax)	(None, 2)	514

The Concrete Bridge Crack dataset developed by Zoubir et al. was selected for training a CNN. This dataset is curated explicitly for training models to detect cracks in concrete and is publicly available for academic use [20]. Included are 6,974 200×200 px .jpg images in RGB format. Of these images, 1,304 depict regions of bridge concrete with cracks and are labeled 'Positive'. The remaining 5,634 images are labeled 'Negative' and do not contain cracks. This dataset offers variation in photo angle, stains, and other concrete features to enhance model robustness. Samples from both classes of the dataset are displayed and labeled in Fig. 4.


Figure 4: Sample of Concrete Bridge Crack Dataset

To increase model robustness, images in this dataset were randomly horizontally flipped, rotated by small angles, and randomly zoomed. The data were then resized to 224×224 pixels to match the original EfficientNetB0 input size. The dataset was split into 80% for training and 20% for validation. The model was trained for 10 epochs with a batch size of 32. To increase computation speed, we use Google Colab's T4 GPU runtime environment. The performance of the TL EfficientNetB0 is given in Table 2, and we include the confusion matrix in Fig. 5. To facilitate accurate assessment, the class distribution was carefully quantified before training, and the dataset was partitioned at the image level to prevent overlap between the training and testing sets. We provide precision, recall, F1-score, and confusion matrix analysis rather than relying solely on overall accuracy, which can be misleading when data are unevenly distributed. These evaluations provide a more accurate representation of each class's performance and mitigate bias towards the prevailing class.

Table 2: CNN Performance

Metric	Training	Validation
Accuracy	0.9957	0.9982
Precision	0.9957	0.9984
Recall	0.9957	0.9982
F1-score	0.9957	0.9983
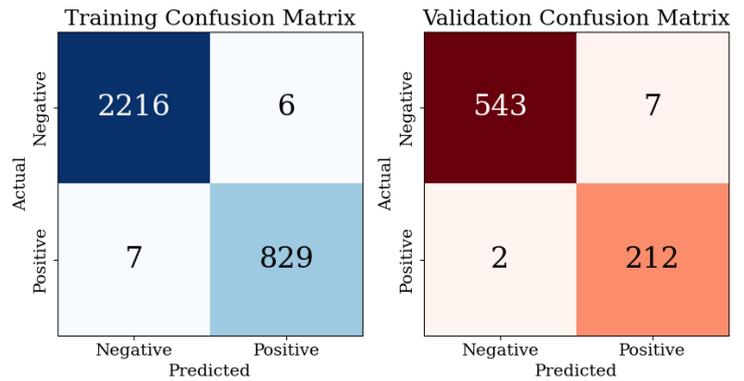
Figure 5: CNN Confusion Matrix

The particularly high recall for the crack class, along with balanced precision metrics, indicates that the model did not demonstrate bias in predicting the common negative class. It kept high sensitivity while minimizing false positives. This implies that the learned model effectively captures the unique characteristics of cracks, even with the uneven class distribution.

Model performance on the training and validation sets yields nearly perfect scores, indicating robust and reliable classification of cracks in concrete bridge imagery. This is further evident in the confusion matrix, which shows a high proportion of correct classifications and few errors.

3.4. End-to-end UAV-BCD system workflow

The suggested system operates as a single pipeline that connects UAV image capture with cloud processing and deep learning-driven crack detection. This process ensures that visual data collected during flight is transmitted, processed, and analyzed accurately and in real time. The following describes each step in the proposed UAV-BCD system.

- Step 1: The DJI Tello drone gathers visual data as it is piloted by hand to cover the structural area of interest.
- Step 2: The server prepossesses each frame as it arrives, extracting an ROI for crack detection.
- Step 3: The ROI segment is sent through a forward pass of the CNN model. A classification is performed at a rate matching the number of video frames arriving, enabling autonomous detection of bridge cracking.
- Step 4: The final results of the discovery are displayed with a GUI and written to a video file. This includes visual overlays such as bounding boxes, classification labels, and status markers indicating time of flight and any detected cracks.

4. UAV environmental setup and experimental results

4.1. Concrete Bridge Model for Experimentation

To test the proposed methodology in application, we simulate a bridge inspection scenario. The simulated setup establishes a nondestructive testing environment to investigate the efficacy of the proposed UAV-BCD in SHM paradigms.

The bridge is modeled by printing out images of concrete and covering a region to represent a bridge segment. To establish a base, an image is pulled from the 'Negative' class of the Concrete Bridge Crack Dataset. This image is printed on several A4 pages and then used to tile a 4.5×2.5 ft cardboard region. To simulate bridge damage, printouts of an image from the 'Positive' class are made similarly. The images of cracked concrete are sparsely distributed across the simulated span to model surface cracking. Figure 6 shows the assembled bridge model we utilize in this case study.

The experimental space allocated for this case study is an $8 \times 5 \times 8$ ft boundary. One of the vertical surfaces of this region hosts the simulated bridge surface, which is mounted 6 ft above ground level. The starting position for the Tello is defined by a grey marker on the ground, 7 ft from the target bridge. A visualization of the experiment space is pictured in Figure 7.

4.2. Inspection Simulation

The Tello is placed on the ground and powered on, allowing the laptop to connect to the Tello's Wi-Fi channel. When the Tello's front LED stops blinking and stays on, connectivity is fully established, and the main script may be executed. Once this is complete, the Tello is initialized, ensuring that its transmission features are enabled so it can receive commands and transmit a video feed.

The drone is piloted upwards and forward towards the modeled Inspection site and subsequently positioned to look down at the bridge surface. We maintain a distance of approximately 2 ft from the bridge model's surface and 3-5 ft from ground level. In Figure 8, we show an example of the Tello flying at an appropriate detection distance, followed by examples of autonomous crack detection output by the UAV-BCD system in Figure 9. The proposed UAV-BCD system is shown to be effective in this simulated inspection scenario. In a single flight, CNN correctly detects each instance of surface cracking with strong inference. Examples of real-time autonomous crack detection are shown in Figure 9.
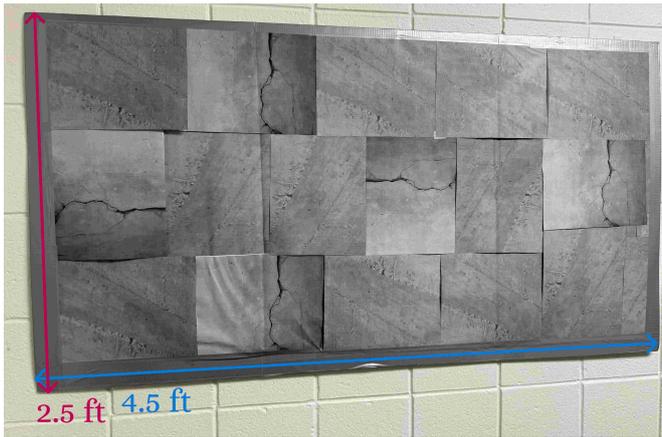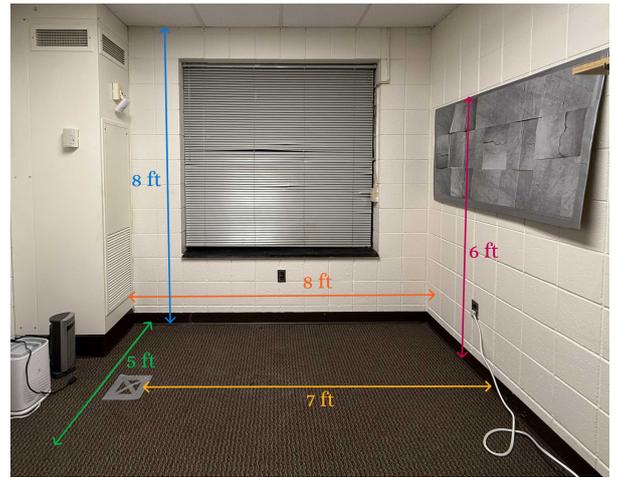
Figure 6: Developed Deteriorated Bridge Model

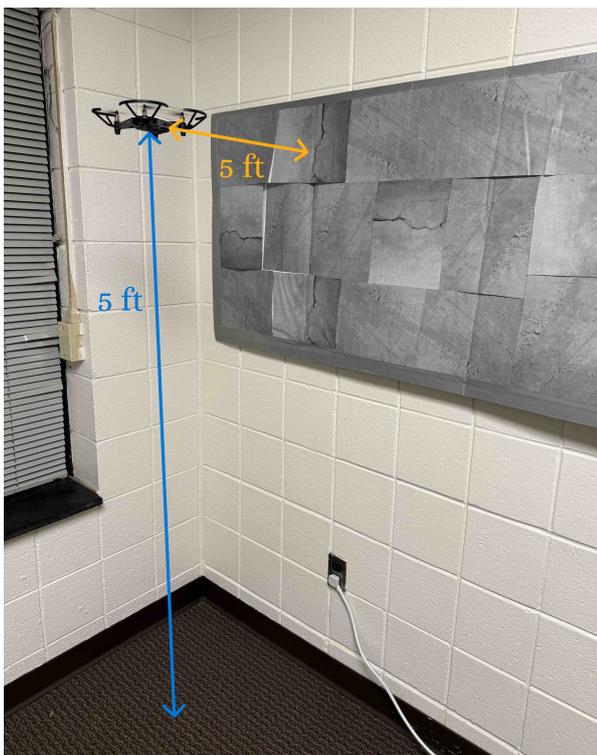

Figure 7: Environmental Setup

Figure 8: Bridge model inspection at an appropriate detection position

Figure 9: Examples of autonomous crack detection carried out by the proposed system.

4.3. Limitations and Real-World Generalization

Tests were conducted under controlled settings to demonstrate proof of concept and system integration. By printing high-resolution crack images onto the bridge surface, it was possible to achieve controlled lighting, imaging geometry, and ground-truth annotations. With this design, it is possible to conduct thorough examinations of the proposed architecture and to reproduce the results.

Although the dataset was small, it was sufficient to determine whether the architecture was effective. The high metrics may be explained by limited domain diversity, despite the use of data leakage prevention and class-specific performance evaluation using accuracy, recall, and

F1-score. Rather than attempting to infer field-level resilience from the data, one should focus on technical feasibility. Although the cloud component was designed to validate edge–cloud collaboration, this preliminary study did not evaluate network latency, bandwidth utilization, or scalability under large-scale deployment.

Validating real bridge structures across a variety of environmental conditions, utilizing datasets from different domains, and evaluating system performance are all areas of focus for the future. The fundamental framework of this study will be expanded across these phases to evaluate its generalizability and operational readiness. Although robust performance was observed on the chosen dataset, actual bridge settings exhibit greater unpredictability that is not fully captured in this context. Subsequent research will encompass cross-domain validation using heterogeneous, publicly available datasets, alongside evaluations across diverse real-world scenarios to assess generalization performance.

5. Conclusions

In this study, a lightweight CNN-based TL model for concrete crack classification was developed on an EfficientNetB0 backbone. This approach yielded a highly adept model, achieving near-perfect scores across all performance evaluation metrics. This model was integrated into a UAV-based inspection system to enable autonomous crack detection in real-time video streams. An experimental environment was designed to evaluate the proposed methodology, providing a proof of concept for its effectiveness as a component of an SHM system. The efficacy demonstrated in this case study shows a computationally and financially inexpensive inspection system that provides timely classifications in real-time deployment. Future work will experiment with the proposed method in a field setting on a real bridge and will implement the detection of other types of surface degradation.

References

[1] American Road & Transportation Builders Association, "2025 Bridge Report", (2025), available online: https://artbabridgereport.org/reports/2025-ARTBA-Bridge-Report.pdf.

[2] B. Zhang, Y. Ren, S. He, Z. Gao, B. Li, and J. Song, "A review of methods and applications in structural health monitoring (SHM) for bridges", *Measurement*, Vol. 245, (2025), pp. 116575.

[3] C. Kralovec and M. Schagerl, "Review of Structural Health Monitoring Methods Regarding a Multi-Sensor Approach for Damage Assessment of Metal and Composite Structures", *Sensors (Basel)*, Vol. 20, No. 3, (2020), pp. 826.

[4] B. J. Perry, Y. Guo, R. Atadero, and J. W. van de Lindt, "Streamlined bridge inspection system utilizing unmanned aerial vehicles (UAVs) and machine learning", *Measurement*, Vol. 164, (2020), pp. 108048.

[5] D. Gillins, C. Parrish, M. Gillins, and C. Simpson, "EYES IN THE SKY: BRIDGE INSPECTIONS WITH UNMANNED AERIAL", Oregon Department of Transportation, (2018).

[6] J. Wells and B. Lovelace, "Unmanned Aircraft System Bridge Inspection Demonstration Project Phase II Final Report", Minnesota Department of Transportation, (2017).

[7] F. Gouveia, V. Silva, J. Lopes, et al., "Automated Identification of Building Features with Deep Learning for Risk Analysis", *Discover Applied Sciences*, Vol. 6, (2024), pp. 466.

[8] D. Ai, G. Jiang, S.-K. Lam, P. He, and C. Li, "Computer vision framework for crack detection of civil infrastructure—A review", *Engineering Applications of Artificial Intelligence*, Vol. 117, (2023), pp. 105478.

[9] R. Fu, M. Cao, D. Novák, X. Qian, and N. F. Alkayem, "Extended efficient convolutional neural network for concrete crack detection with illustrated merits", *Automation in Construction*, Vol. 156, (2023), pp. 105098.

[10] D. Choi, W. Bell, D. Kim, and J. Kim, "UAV-Driven Structural Crack Detection and Location Determination Using Convolutional Neural Networks", *Sensors*, Vol. 21, No. 8, (2021), pp. 2650.

[11] A. Karmokar, N. Jani, A. Kalla, H. Harlalka, and P. Sonar, "Inspection of Concrete Structures by a Computer Vision Technique and an Unmanned Aerial Vehicle", *2020 International Conference on Computational Performance Evaluation (ComPE)*, (2020), pp. 338–343.

[12] J. Plested and T. Gedeon, "Deep Transfer Learning for Image Classification: A Survey", *arXiv preprint arXiv:2205.09904*, (2022).

[13] C. F. Ozgenel and A. Gonenc Sorguc, "Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings", *Proceedings of ISARC 2018*.

[14] A. Mirzazade, C. Popescu, J. Gonzalez-Libreros, T. Blanksvärd, B. Täljsten, and G. Sas, "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*, Vol. 13, No. 8, (2023), pp. 1633–1652.

[15] X. Wang, Y. R. Tan, W. Leong, S. Huang, R. Teo, and C. Xiang, "GPS Denied IBVS-Based Navigation and Collision Avoidance of UAV Using a Low-Cost RGB Camera", *arXiv preprint arXiv:2509.17435v1*, (2025).

[16] Z. Ameli, Y. Aremanda, W. A. Friess, and E. N. Landis, "Impact of UAV Hardware Options on Bridge Inspection Mission Capabilities", *Drones*, Vol. 6, No. 3, (2022), pp. 64.

[17] RYZE, "Tello", (2025), available online: https://www.ryzerobotics.com/tello.

[18] DJI Tello, "Tello - DJITelloPy API Reference", available online: https://djitellopy.readthedocs.io/en/latest/tello/.

[19] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", *arXiv preprint arXiv:1905.11946*, (2020), available online: https://arxiv.org/abs/1905.11946.

[20] H. Zoubir, M. Rguig, and M. Elaroussi, "Crack recognition automation in concrete bridges using Deep Convolutional Neural Networks", *MATEC Web of Conferences*, Vol. 349, (2021), pp. 03014.