PREPRINT - Submitted to the Journal of Information Technology in Construction (ITcon), July 2026 - not yet peer reviewed

AUTOMATED MULTI-CLASS STRUCTURAL DAMAGE DETECTION AND SEGMENTATION FOR UNMANNED AERIAL VEHICLE-BASED INFRASTRUCTURE INSPECTION USING DEEP LEARNING

SUBMITTED: July 2026

Seyed Farhad Abtahi
Independent Researcher, Istanbul, Türkiye
farhadabtahi91@gmail.com* | *ORCID: 0009-0007-8865-2300

SUMMARY:

This study presents a three-stage deep learning framework for automated structural damage classification, segmentation, and unmanned aerial vehicle (UAV)-based inspection simulation targeting bridges, offshore platforms, and other civil infrastructure. Stage 1 fine-tunes ResNet50 with differential learning rates and class weighting to classify crack, corrosion, spalling, and intact surfaces, achieving 85.69% accuracy on a held-out test set of 1,600 images drawn from multi-source public datasets, with the differential learning rate strategy contributing an 8.25% improvement over the frozen-backbone baseline. A targeted augmentation strategy addresses minority-class imbalance and yields a spalling F1-score of 0.886 under the balanced evaluation protocol. Stage 2 applies DeepLabV3+ with a partially unfrozen ResNet50 backbone for pixel-level damage localization, achieving mean intersection over union (mIoU) values of 0.7890 for three-class segmentation and 0.6446 for four-class segmentation, the latter incorporating 315 real UAV crack images to extend the training distribution to the aerial perspective absent from most public crack datasets. Stage 3 integrates frame-level inference into a boustrophedon UAV grid inspection simulation that produces a severity-coded damage map and a structured JSON inspection report, identifying 16 of the 20 damaged zones. Average CPU inference time is 16.29 ms per frame (61 frames per second), consistent with near-real-time deployment on modest hardware. Ablation studies of backbone unfreezing depth and Grad-CAM visualizations provide additional support for the robustness and interpretability of the proposed pipeline. The framework demonstrates that multi-class damage recognition, pixel-level localization, and inspection workflow integration can be combined in a single reproducible system built entirely on open data.

KEYWORDS: *structural damage detection; semantic segmentation; UAV inspection; transfer learning; ResNet50; DeepLabV3+; infrastructure monitoring*

1. INTRODUCTION

Concrete bridges, marine platforms, retaining walls, and tunnel linings degrade through mechanisms that are well understood but difficult to detect consistently at scale. Surface cracking develops under structural overload or restrained shrinkage, reinforcement corrosion advances through chloride ingress, and concrete spalling follows freeze-thaw cycling or cover loss. These damage types do not evolve in isolation: early-stage corrosion accelerates cracking, and undetected spalling exposes reinforcement to further environmental attack. Timely identification of each type is therefore a prerequisite for rational maintenance planning and service life extension (Rashidi et al., 2020; Brownjohn, 2007).

Current practice still depends on trained engineers conducting visual surveys, often supplemented by non-destructive testing. In offshore environments this is particularly hazardous: rope-access inspection of a single jacket structure can take several days and places personnel at significant risk. Unmanned aerial vehicles (UAVs) have changed the access problem by enabling safe coverage of elevated and confined locations at reduced cost (Phung et al., 2017; Seo et al., 2018; Morgenthal and Hallermann, 2014). The bottleneck has shifted: a single bridge survey can generate thousands of image frames, and manual review of that volume is neither consistent nor scalable. Deep learning, and convolutional neural networks (CNNs) in particular, has demonstrated strong performance in structural damage recognition (Zhang et al., 2016; Dorafshan et al., 2018; Gopalakrishnan et al., 2017). Binary crack classification exceeds 99% accuracy on curated benchmark datasets (Dung and Anh, 2019), and region-based detectors have been applied to multi-type defect localization (Xu et al., 2019). Despite these advances, three gaps limit the practical utility of existing approaches. First, most systems address only one or two damage types, whereas operational inspections require simultaneous identification and prioritization of crack, corrosion, spalling, and intact surfaces. Second, image-level classification provides no spatial information about damage location or extent, which is necessary for repair volume estimation and risk-zone mapping. Third, few systems have been validated in end-to-end UAV inspection workflows that produce actionable output for maintenance decisions.

To address these gaps, the proposed framework uses a three-stage deep learning pipeline. Stage 1 performs four-class image-level damage classification using a fine-tuned ResNet50 model. Stage 2 performs pixel-level segmentation using DeepLabV3+ to spatially localize damage regions. Stage 3 integrates Stage 1 inference into a simulated boustrophedon UAV grid inspection, producing a severity-coded damage map and a structured JSON inspection report. The system architecture is shown in Figure 1.

The main contributions are summarized as follows:

- A four-class damage classifier trained on a multi-source dataset, achieving 85.69% accuracy on the original held-out validation set, together with a differential learning rate strategy that improved accuracy by 8.25% over the frozen-backbone baseline.
- A targeted augmentation strategy that improved minority-class spalling recognition, yielding a spalling F1-score of 0.886 under the balanced evaluation protocol.
- A dual-mode segmentation pipeline (three-class mIoU: 0.7890; four-class mIoU: 0.6446), together with per-class IoU analysis and an ablation study of backbone unfreezing depth.
- Integration of 315 real UAV crack images into a four-class segmentation model, extending the training distribution to include the aerial perspective that is absent from most existing crack datasets.
- Validation through a UAV inspection simulation achieving 80% damage-zone detection and 61 FPS on CPU.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the methodology. Section 4 presents the datasets and experimental setup. Section 5 reports the results and ablation studies. Section 6 discusses the findings and limitations. Section 7 concludes the paper.

2. RELATED WORK

1. 2.1 Crack Detection Using Deep Learning

The SDNET2018 dataset (Maguire et al., 2018), which contains 56,092 concrete surface images from bridge decks, walls, and pavements, is the most widely used benchmark for binary crack detection. Zhang et al. (2016) provided an early demonstration that learned convolutional features outperform hand-crafted approaches for road crack detection. Dorafshan et al. (2018) compared CNNs with traditional edge detectors and confirmed the advantage of learned features. Dung and Anh (2019) used a VGG16-based fully convolutional network to reach approximately 99.9% classification accuracy. Although these binary results are impressive, they do not transfer directly to multi-class settings, where the model must distinguish among crack, corrosion, spalling, and intact surfaces at the same time.

2. 2.2 Multi-Class Damage Detection

Cha et al. (2017) provided one of the first demonstrations of deep learning for structural crack detection, reporting approximately 98% accuracy with a sliding-window convolutional network. Subsequent work explored attention-enhanced fully convolutional networks for crack localization (Yang et al., 2018) and region-based CNNs for seismic damage classification (Xu et al., 2019), while Li et al. (2019) demonstrated that multi-task learning can simultaneously detect several damage types. Recent work has continued to address the negative-information-transfer problem in this setting; for example, an asymmetric multi-task deep learning framework with location attention has been proposed to mitigate inter-task interference in multi-class damage classification (Li et al., 2025). Corrosion and spalling, despite their importance in offshore and marine structures, have received far less attention than crack detection.

3. 2.3 Semantic Segmentation for Structural Damage

Fully convolutional networks (Long et al., 2015) and encoder-decoder architectures have both been used for pixel-level damage localization. DeepLabV3+ (Chen et al., 2018), which combines atrous spatial pyramid pooling (ASPP) with an encoder-decoder design, has shown strong performance in dense prediction tasks. Bang et al. (2019) proposed a residual encoder-decoder network for pixel-level road crack detection in black-box camera images. Fan et al. (2020) applied an ensemble of convolutional neural networks to pavement crack detection and measurement. Zou et al. (2019) proposed DeepCrack for hierarchical crack segmentation, and Mei and Gul (2020) proposed a conditional adversarial network for pavement crack segmentation, benchmarking it against several encoder-decoder baselines. To reduce the cost of pixel-level annotation, Mishra et al. (2024) proposed a weakly supervised crack segmentation approach based on crack-attention networks, lowering the dependence on dense ground-truth masks. A persistent challenge is crack segmentation in UAV imagery, where crack pixels typically make up only 1-3% of the image area (Zou et al., 2019; Shi et al., 2016). To reduce the tendency of the model to ignore sparse crack pixels, class-weighted loss was used during training.

4. 2.4 Transfer Learning for Structural Inspection

Transfer learning from ImageNet-pretrained networks has become standard practice in structural damage detection, largely because labeled structural imagery remains limited (Hoskere et al., 2018). Gao and Mosalam (2018) demonstrated systematic improvements from tuning the fine-tuning depth for structural damage classification. Pan and Yang (2010) provided the theoretical basis for why transfer learning is effective when the source and target domains share low-level visual features. This is especially relevant here because ImageNet-trained edge and texture detectors transfer well to structural surface analysis. The differential learning rate strategy used in Stage 1 (1x10^-5 for backbone, 1x10^-3 for classifier head) follows the discriminative fine-tuning approach of Howard and Ruder (2018), preventing catastrophic forgetting while allowing task-specific adaptation.

5. 2.5 UAV-Based Infrastructure Inspection

Automated UAV inspection systems for structural health monitoring have been reported for bridges (Phung et al., 2017), wind turbines (Shihavuddin et al., 2019), offshore platforms (Kang and Cha, 2018), and buildings (Rakha and Gorodetsky, 2018). Phung et al. (2017) addressed path planning for UAV surface inspection using particle swarm optimization. Seo et al. (2018) developed and field-tested a drone-enabled bridge inspection methodology. Most existing systems still rely on binary detection pipelines and do not provide spatial damage localization. More recent work has integrated obstacle-avoiding autonomous UAV control with deep-learning-based damage detection in GPS-denied environments (Waqas et al., 2024) and extended the visual pipeline to three-dimensional damage mapping using neural radiance field reconstruction (Kim and Cha, 2025). Rakha and Gorodetsky (2018) reviewed UAV applications in the built environment and identified multi-class detection and pixel-level localization as key open challenges, both of which are addressed in this study. To the author's knowledge, few previous studies have combined multi-class classification, segmentation, and inspection simulation within a single framework.

3. METHODOLOGY

1. 3.1 System Architecture

The framework is organized into three sequential stages built around a shared ResNet50 (He et al., 2016) backbone for feature extraction. The overall architecture is shown in Figure 1. In Stage 1, the model performs image-level damage classification for rapid triage. In Stage 2, the model performs pixel-level segmentation to localize damage regions. In Stage 3, Stage 1 inference is integrated into a simulated UAV inspection workflow. All components were implemented in PyTorch 2.1.0 and benchmarked on CPU hardware (Intel processor, no GPU), providing a CPU-only performance baseline.

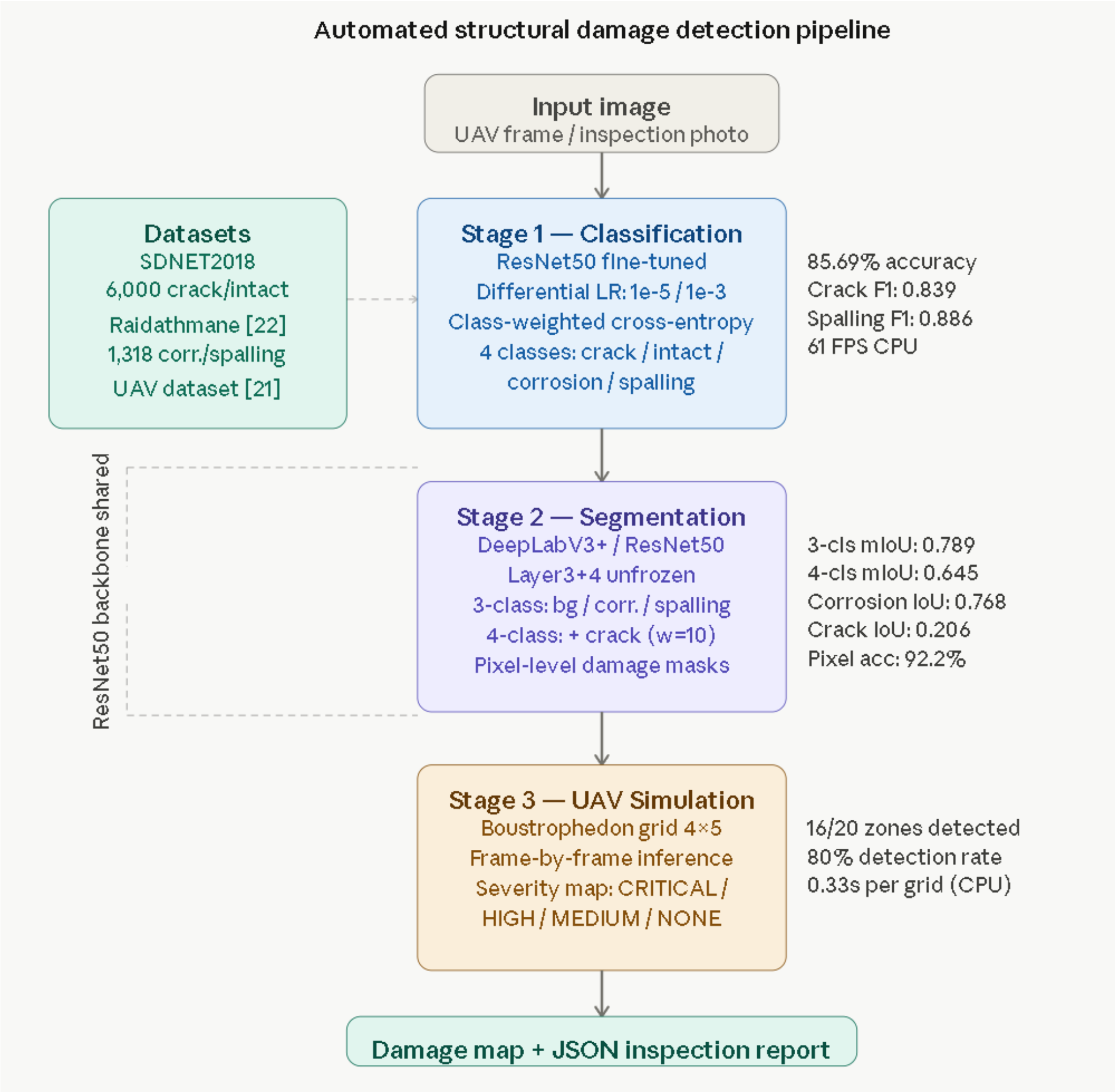

Figure 1: Proposed three-stage deep learning pipeline for automated structural damage inspection. Stage 1 performs image-level four-class classification using fine-tuned ResNet50. Stage 2 performs pixel-level segmentation using DeepLabV3+ with a partially unfrozen ResNet50 backbone. Stage 3 integrates frame-level classification into a boustrophedon UAV grid inspection and outputs a severity-coded damage map and JSON inspection report. The ResNet50 backbone is shared across Stages 1 and 2, enabling efficient knowledge transfer.

2. 3.2 Stage 1: Multi-Class Damage Classification

2.1 *3.2.1 Network Architecture*

The classification model builds on ResNet50 (He et al., 2016) pre-trained on ImageNet-1K (Deng et al., 2009). The 50-layer residual network uses skip connections that maintain gradient flow through deep layers and encourage feature reuse across spatial scales. The original 1000-class fully connected head is replaced with a custom classifier:

$$f(x) = W_2 \cdot \mathrm{Dropout}(\mathrm{ReLU}(W_1 \cdot \mathrm{Dropout}(x, p = 0.4)), p = 0.3) \quad (1)$$

where $W_1 \in \mathbb{R}^{256\times2048}$ and $W_2 \in \mathbb{R}^{4\times256}$. This adds 525,828 trainable parameters beyond the ResNet50 backbone.

2.2 *3.2.2 Differential Learning Rate Fine-Tuning*

A differential learning rate strategy was used across the network layers. Backbone layers 1-2 are frozen (their pre-trained edge and texture detectors already transfer well to structural surfaces). Backbone layers 3-4 are unfrozen with learning rate $\eta_b = 1x10^{-5}$. The classification head trains at $\eta_h = 1x10^{-3}$. The 100x ratio between η_b and η_h follows the discriminative fine-tuning principle (Howard and Ruder, 2018) and helps avoid catastrophic forgetting of ImageNet-learned mid-level features while still allowing adaptation to the structural domain. The total number of trainable parameters was 22.6 million.

2.3 *3.2.3 Loss Function with Class Weighting*

Cross-entropy loss is weighted by inverse class frequency to handle dataset imbalance:

$$L = -\Sigma_c\, w_c \cdot y_c \cdot \log(p_c), \quad w_c = N / (C \cdot n_c) \qquad (2)$$

where N is the total number of training samples, C is the number of classes (4), and n_c is the number of samples in class c. This keeps the gradient contribution from each class more balanced despite the difference in sample counts between crack/intact (2,400 training images each) and corrosion/spalling (900-1,118 training images).

2.4 *3.2.4 Data Augmentation Strategy*

Standard augmentation (random horizontal/vertical flips, 15 degree rotation, color jitter of +/-0.2 brightness/contrast) was applied to all classes. Targeted augmentation for the spalling class expanded the training set from 298 to 900 samples. Because the original held-out validation split remained imbalanced, the main classification results in Table 2 are reported on that original validation set (1,600 samples: 600 crack, 600 intact, 200 corrosion, and 200 spalling). To obtain more stable minority-class estimates, a separate balanced evaluation protocol was also used, with 400 images per class. Results derived from this balanced protocol are reported separately in section 5.2 and are not directly interchangeable with the original held-out validation results.

3. 3.3 Stage 2: Semantic Segmentation

3.1 *3.3.1 Architecture*

Pixel-level damage localization uses DeepLabV3+ (Chen et al., 2018) with a ResNet50 backbone. The architecture uses Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale context at rates r in {6, 12, 18}, which helps delineate damage boundaries across structures with different damage scales.

3.2 *3.3.2 3-Class Segmentation Model*

The 3-class model (C=3: background, corrosion, spalling) is trained on corrosion-spalling segmentation data (Raidathmane, 2025). Backbone layers 3-4 are unfrozen with learning rate $2x10^{-5}$ alongside the ASPP module and decoder. Training uses Adam optimizer with step decay (gamma=0.5 at epoch 7) over 25 epochs, batch size 4, image size 256x256.

3.3 *3.3.3 4-Class Segmentation Model*

The 4-class model (C=4: background, crack, corrosion, spalling) combines two datasets. Crack segmentation uses real UAV imagery (Ziya07, 2025) with binary masks. Corrosion/spalling segmentation uses RGB-annotated masks (Raidathmane, 2025). Class-weighted loss addresses the extreme pixel sparsity of cracks: w = [0.1, 10.0, 1.0, 1.0] for background, crack, corrosion, and spalling respectively. The crack weight of 10.0 was selected empirically based on the ablation study in section 5.6. Without weighting, the model simply predicts background for all pixels, achieving high accuracy while missing crack regions entirely (crack IoU = 0.000).

3.4 *3.3.4 Evaluation Metric*

Mean Intersection over Union (mIoU) serves as the primary segmentation metric. Per-class IoU is reported alongside mIoU because the aggregate mIoU can hide class-specific performance when class sizes differ substantially.

4. 3.4 Stage 3: UAV Grid Inspection Simulation

The drone simulation follows a boustrophedon (serpentine) grid traversal over a 4x5 inspection area (20 zones in total), consistent with the systematic coverage strategies used in operational UAV inspection (Phung et al., 2017). At each grid position, the Stage 1 classifier processes the captured frame and assigns a damage label together with a softmax confidence score. The results are then aggregated into a severity-coded damage map: CRITICAL (crack, red), HIGH (corrosion, orange), MEDIUM (spalling, yellow), and NONE (intact, green). A JSON inspection report records per-zone predictions, confidence scores, damage distribution, and total inspection time.

5. 3.5 Grad-CAM Interpretability

Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017) is applied to the Stage 1 classifier to visualize which image regions drive classification decisions. Grad-CAM maps are upsampled to input resolution and overlaid on the original image (Figure 6).

6. 3.6 Damage Area Estimation

Assuming that a 256x256 image covers an approximate 2x2 m ground footprint at a drone altitude of 5 m, the effective spatial resolution is about 7.8 mm/pixel. Under this assumption, the estimated damage area is approximately 0.61 cm^2 per pixel. This provides an approximate area estimate for maintenance planning, but it should not be interpreted as a calibrated geometric measurement.

4. DATASETS AND EXPERIMENTAL SETUP

1. 4.1 Datasets

Three publicly available datasets were used across the three stages of the proposed pipeline.

SDNET2018 (Maguire et al., 2018): 56,092 images of cracked and non-cracked concrete surfaces across bridge decks, walls, and pavements. Image resolution: 256x256 pixels. Binary annotations. For Stage 1, 3,000 images per class (crack, intact) are sampled with 80/20 train/val split.

Corrosion and Spalling Segmentation Dataset (Raidathmane, 2025): 1,580 image-mask pairs with pixel-level RGB annotations. Corrosion: red (255,0,0); spalling: yellow (255,255,0); background: black (0,0,0). Used for Stage 1 classification labels and Stage 2 segmentation training.

UAV-Based Crack Detection Dataset (Ziya07, 2025): 315 real UAV images of cracked concrete surfaces with binary segmentation masks (white=crack, black=background). Mean crack pixel density: 1.78% per image. The dataset was split 80/20 for Stage 2 four-class segmentation.

*Table 1: Dataset composition after augmentation. *Augmented from original counts (spalling: 298 train/36 val; corrosion val: 128) using targeted geometric and photometric augmentation.*

Class	Train	Val	Total	Source	Stage
Crack	2,400	600	3,000	SDNET2018 (Maguire et al., 2018)	1, 2
Intact	2,400	600	3,000	SDNET2018 (Maguire et al., 2018)	1
Corrosion	1,118	200*	1,318	Raidathmane (2025)	1, 2
Spalling*	900	200*	1,100	Raidathmane (2025) + Aug.	1, 2
Crack (seg.)	252	63	315	UAV Dataset (Ziya07, 2025)	2 (4-class)
Total (Stage 1)	6,818	1,600	8,418	Combined	Stage 1
Total (Stage 2)	1,670	263	1,933	Combined	Stage 2

2. 4.2 Implementation Details

All experiments were implemented in PyTorch 2.1.0 (Paszke et al., 2019) and run on CPU hardware (Intel processor, without GPU acceleration). For Stage 1, the input size was 224x224, batch size 16, training for 20 epochs, Adam optimizer, StepLR scheduling (step=7, gamma=0.1). For Stage 2, input size 256x256, batch size 4, training for 25-30 epochs, Adam optimizer, StepLR scheduling (step=7-8, gamma=0.5). A random seed of 42 was used for all data splits. ImageNet normalization was applied using mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225].

5. EXPERIMENTAL RESULTS

1. 5.1 Stage 1 Classification Performance

Table 2 reports the per-class metrics on the original held-out validation set (1,600 samples in total: 600 crack, 600 intact, 200 corrosion, and 200 spalling).

Table 2: Class-wise performance of the Stage 1 damage classifier on the original held-out validation set.

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Crack	0.9119	0.7767	0.8389	600
Intact	0.8020	0.9317	0.8620	600
Corrosion	0.9106	0.8150	0.8602	200
Spalling	0.8592	0.9150	0.8862	200
Macro Avg	0.8709	0.8596	0.8618	1,600
Weighted Avg	0.8640	0.8569	0.8561	1,600

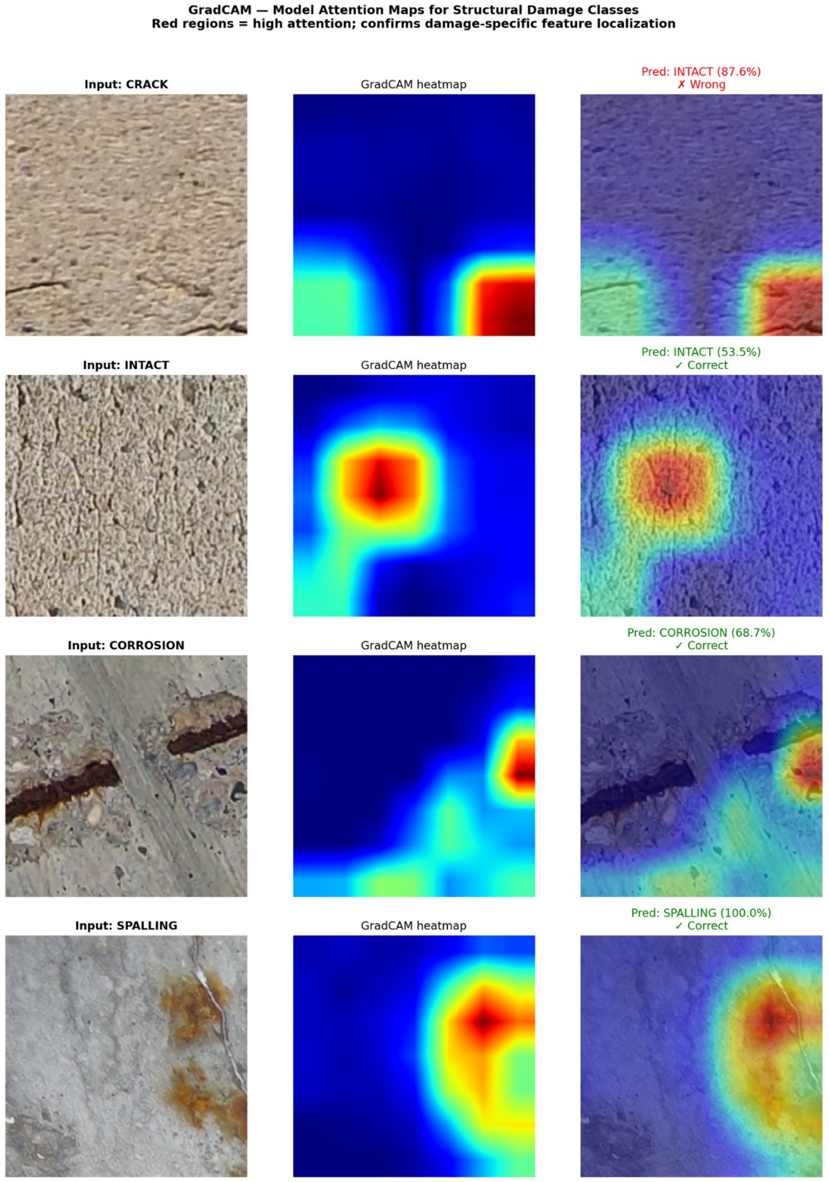

Figure 2: Confusion matrix for Stage 1 four-class damage classification on the original held-out validation set (1,600 samples).

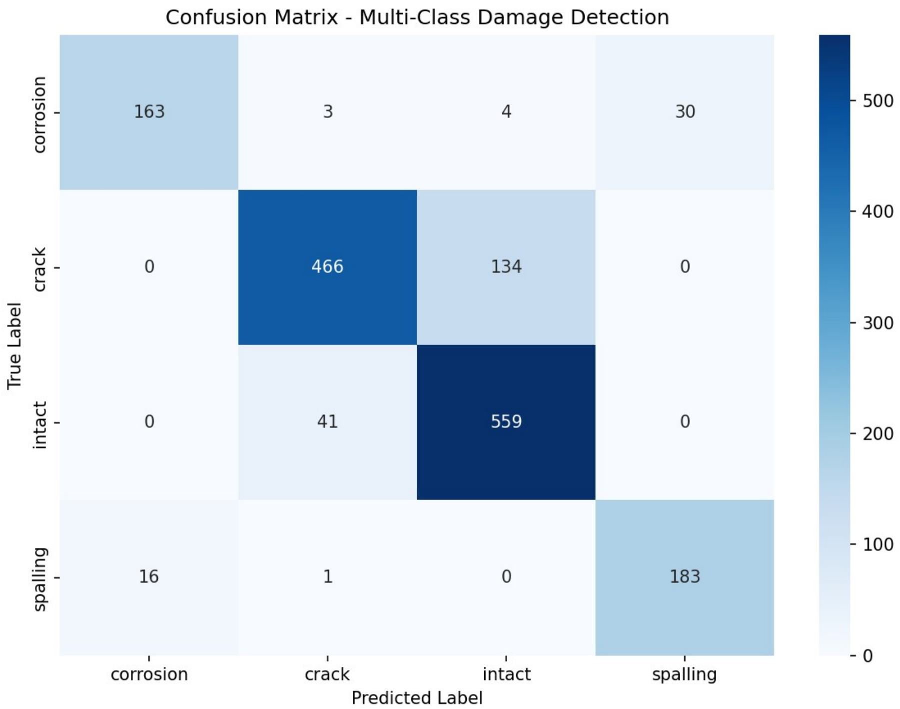

Figure 3: Stage 1 training and validation loss and accuracy curves over 20 epochs. Validation accuracy reached 85.69%.

2. 5.2 Effect of Targeted Augmentation

Table 3: Spalling class metrics before and after targeted augmentation (before: original hold-out split with 36 spalling images; after: balanced evaluation protocol with 200 spalling images).

Metric	Before (n=36 val)	After (n=200 val)	Change
Precision	0.750	0.859	+10.9%
Recall	0.667	0.915	+24.8%
F1-Score	0.706	0.886	+18.0%
Overall Accuracy	85.26%	85.69%	+0.43%

Expanding the spalling training set from 298 to 900 samples improved minority-class performance, and the final model achieved a spalling F1-score of 0.886 on the balanced evaluation set. Because the original spalling hold-out split contained only 36 images, the before/after comparison should be interpreted cautiously.

3. 5.3 Ablation Study and Fine-Tuning Strategy

Table 4: Stage 1 ablation study showing the effect of each design choice.

Configuration	Trainable Params	Val Accuracy	Crack F1	Modification
Frozen backbone (head only)	526K	78.00%	0.770	Baseline
+ Layer4 unfrozen (LR=3e-4)	12.6M	78.00%	0.772	Layer 4 unfrozen; uniform LR
+ Differential LR (1e-5/1e-3)	22.6M	86.25%	0.852	Differential learning rate
+ Class-weighted loss	22.6M	85.26%	0.833	Class-weighted loss
+ Spalling augmentation (final)	22.6M	85.69%	0.839	Spalling augmentation

Among the tested design choices for Stage 1, the differential learning rate strategy had the largest effect, improving validation accuracy by 8.25% over the frozen-backbone baseline. Uniform fine-tuning with a learning rate of 3x10^-4 provided no improvement, indicating that learning-rate calibration, rather than unfreezing alone, was the key factor in this setting.

4. 5.4 COMPARISON WITH PUBLISHED METHODS

Table 5: Comparison with representative published methods.

Method	Architecture	Classes	Metric	Score	Dataset
Cha et al. (2017)	CNN (sliding window)	2	Accuracy	98.0%	Custom concrete
Dung and Anh (2019)	VGG16 FCN	2	Accuracy	99.9%	Concrete crack dataset
Proposed (Stage 1)	ResNet50 DLR	4	Accuracy	85.69%	SDNET + Raidathmane (2025)
Proposed (Stage 2, 3-class)	DeepLabV3+	3	mIoU	0.7890	Raidathmane (2025)
Proposed (Stage 2, 4-class)	DeepLabV3+	4	mIoU	0.6446	UAV + Raidathmane (2025)

Direct comparison with published methods is complicated by differences in datasets and evaluation protocols. Published pixel-level damage segmentation studies differ substantially in damage classes, imagery, and evaluation protocols, so the segmentation results are best interpreted within the dataset context of this study rather than as head-to-head benchmarks. The four-class model (mIoU = 0.645) reflects the added difficulty introduced by UAV crack segmentation under extreme pixel sparsity.

5. 5.5 STAGE 2 SEGMENTATION RESULTS

Table 6: Stage 2 per-class IoU - 3-class model (DeepLabV3+, layers 3+4 unfrozen).

Class	IoU	Pixel Count (val)	Notes
Background	0.8953	6,774,594	High background agreement
Corrosion	0.7689	2,697,069	Stable corrosion segmentation
Spalling	0.7027	392,046	Stable spalling segmentation
mIoU	0.7890	-	Validation mIoU
Pixel Accuracy	0.9177	-	Validation pixel accuracy

Table 7: Stage 2 per-class IoU - 4-class model (crack weight=10.0).

Class	IoU	Pixel Density	Notes
Background	0.9048	-	High background agreement
Crack	0.2059	1.78%	Low crack pixel density
Corrosion	0.7682	-	Stable corrosion segmentation
Spalling	0.6994	-	Moderate spalling segmentation
mIoU	0.6446	-	Validation mIoU
Pixel Accuracy	0.9215	-	Validation pixel accuracy

Figure 4: Stage 2 three-class segmentation results. Each row shows the input image (left), ground-truth mask (center), and model prediction (right). Red = corrosion; yellow = spalling; black = background.

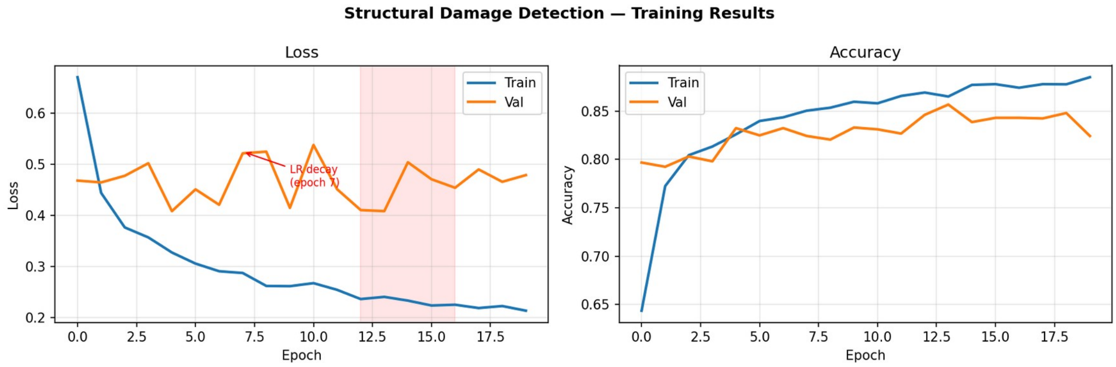

Figure 5: Stage 2 four-class segmentation results on UAV crack imagery. Red = crack; orange = corrosion; yellow = spalling; black = background.

6. 5.6 Segmentation Backbone Ablation

Table 8: Stage 2 ablation study on the effect of backbone unfreezing depth.

Configuration	Trainable Params	Val mIoU	Train mIoU	vs. Baseline
Frozen backbone	~500K	0.5965	0.7173	Baseline
Layer4 unfrozen	~12M	0.6878	0.8797	+15.3%
Layer3+4 unfrozen (final)	~28M	0.7890	0.8856	+32.3%

7. 5.7 Grad-CAM Visualization

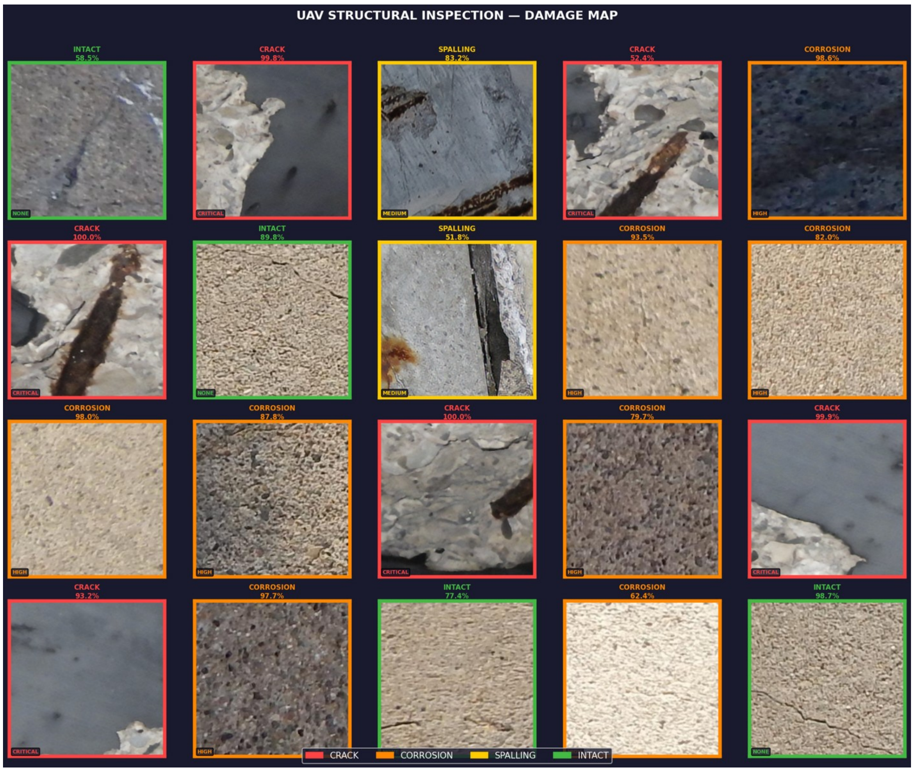

Figure 6: Grad-CAM attention maps for representative Stage 1 classification examples across the four damage classes. Column 1 shows the input image, Column 2 the Grad-CAM heatmap, and Column 3 the overlay. Red indicates higher model attention.

As shown in Figure 6, the Grad-CAM maps suggest that the classifier focuses on visually relevant damage regions rather than unrelated background areas. For crack samples, attention is concentrated near crack-like linear features. For corrosion and spalling, the highlighted regions correspond to localized discoloration and surface discontinuities. For intact surfaces, the attention is more diffuse, which is consistent with the absence of a dominant damage feature.

8. 5.8 Stage 3 UAV Simulation

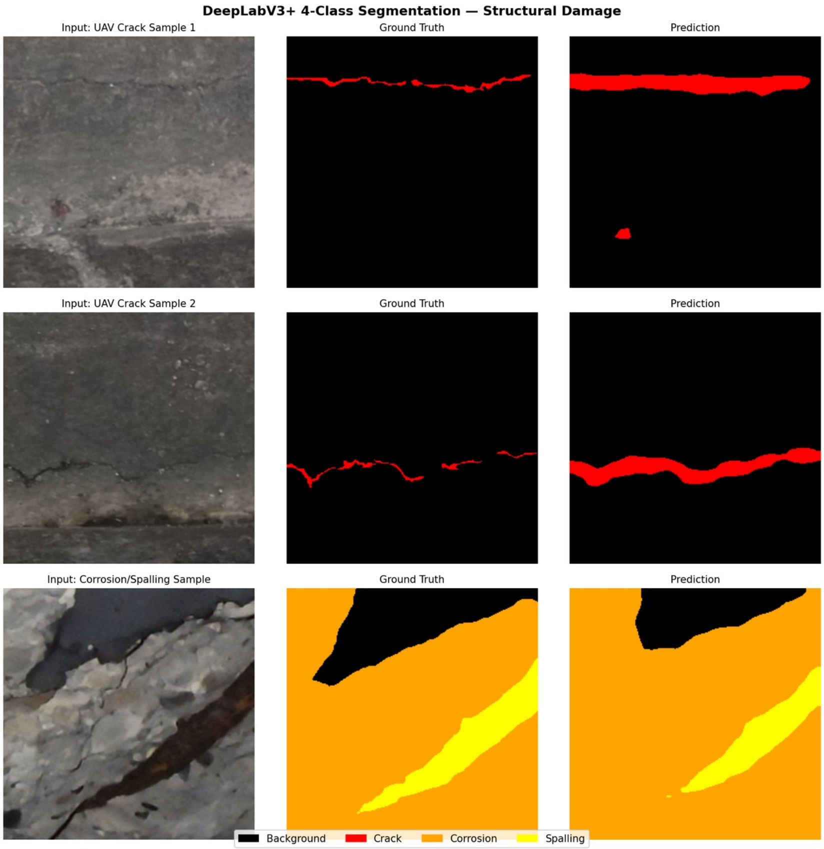

Figure 7: Stage 3 UAV inspection damage map for a 4x5 grid (20 zones). Border color encodes severity: red = CRITICAL (crack), orange = HIGH (corrosion), yellow = MEDIUM (spalling), and green = NONE (intact). Damage was detected in 16 of the 20 zones (80%). Mean inference confidence was 88.4%, and total CPU inference time for 20 frames was 0.33 s.

In the simulation, 16 of the 20 zones were identified as damaged: 8 corrosion, 6 crack, and 2 spalling, while 4 zones were classified as intact. The boustrophedon traversal pattern ensured complete grid coverage. In a practical workflow, zones classified as CRITICAL (crack) could be prioritized for immediate structural assessment.

9. 5.9 Computational Performance

Table 9: Measured CPU inference performance of the Stage 1 classifier for 50 frames.

Metric	Value	Notes
Mean time per frame	16.29 ms	CPU (Intel, no GPU)
Standard deviation	1.41 ms	Low variance, consistent
CPU throughput	61.39 FPS	Exceeds 30 FPS camera rate
20-zone grid (CPU)	0.33 seconds	Computed from mean measured per-frame CPU time

The measured CPU throughput of 61 FPS exceeded the 25-30 FPS capture rate of typical UAV cameras. For the Stage 1 classifier and the tested image size, this result is consistent with near-real-time inference on CPU hardware. GPU performance was not benchmarked in this study.

10. 5.10 Qualitative Failure Analysis

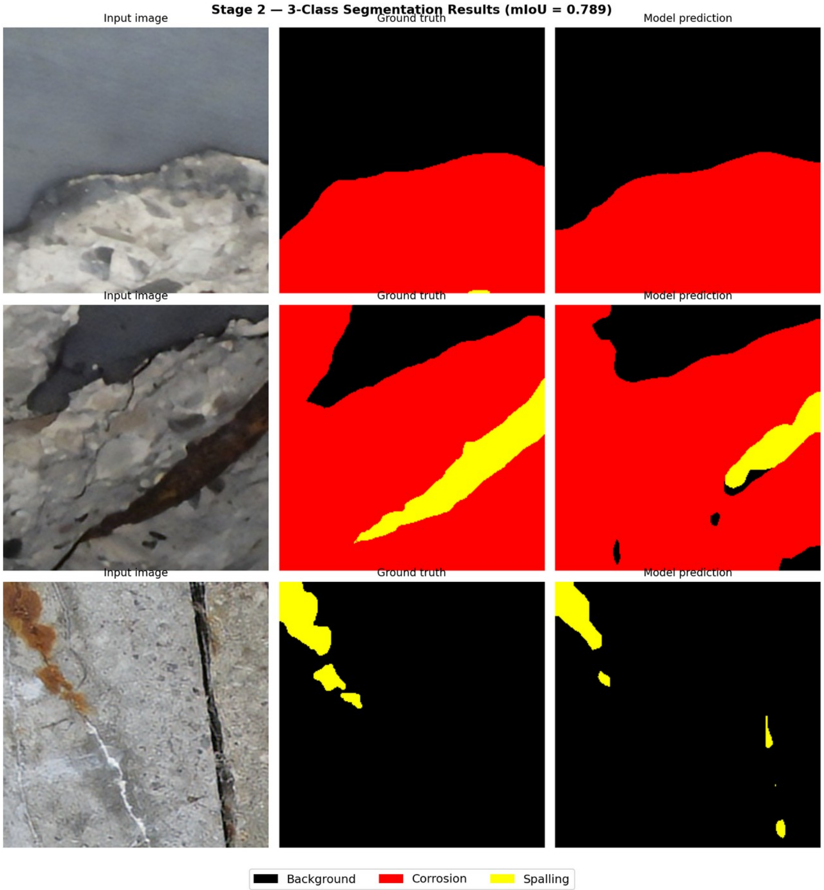

Figure 8: Representative failure analysis. Row 1: hairline crack on textured concrete misclassified as intact (high visual ambiguity). Row 2: wet corrosion misclassified as spalling (moisture alters appearance). Row 3: boundary spalling region with partial mislocalization in segmentation (limited spatial context).

Figure 8 shows three representative failure cases. Case 1: a hairline crack on a heavily textured surface is misclassified as intact because the crack is visually similar to aggregate boundaries. Case 2: a corroded region under wet conditions is misclassified as spalling because moisture darkens the rust staining. Case 3: a spalling region at the image boundary is partially mislocalized due to limited context in the 256x256 crop. These failure cases point to three clear directions for improvement: crack-specific data augmentation with texture-crack composites, illumination normalization preprocessing for corrosion detection, and overlap-based tiling for boundary regions.

6. DISCUSSION

1. 6.1 Differential Learning Rate as the Critical Design Decision

The ablation study (Table 4) indicates that the differential learning rate strategy had the largest effect on classification performance, improving accuracy by 8.25% over the frozen-backbone baseline. Uniform fine-tuning with a high learning rate (3x10^-4) produced no improvement over a frozen backbone. Unfreezing alone was not sufficient; the improvement depended on the learning-rate ratio used for fine-tuning. The 100x ratio between backbone and head learning rates is consistent with prior transfer-learning practice (Howard and Ruder, 2018) and merits further study in structural inspection problems, where the overlap between pretrained features and target textures may vary.

2. 6.2 Crack Segmentation - Fundamental Challenges

Crack IoU of 0.206 in the 4-class model falls below what would be needed for operational deployment. This result reflects three converging challenges: extreme pixel sparsity (1.78% mean density), morphological diversity (hairline to wide, branching, surface to through-cracks), and domain shift between the UAV aerial perspective and the close-up concrete surfaces found in most available crack datasets. Class weighting of 10.0 improved crack IoU from 0.000 (completely missed) to 0.206, but at the cost of more false positives in background regions. Published results for crack segmentation on comparable UAV datasets report IoU values of 0.15-0.35 (Zou et al., 2019; Shi et al., 2016), placing this result within the expected range. The 3-class model (mIoU = 0.789, without the crack class) is the more practical option at present.

3. 6.3 Practical Relevance to Offshore Inspection

The measured 61 FPS inference rate and 0.33 s processing time for a 20-zone grid indicate promising runtime performance for offshore structural inspection scenarios. For an offshore jacket platform with 50-200 inspection zones per flight, the corresponding CPU-based classification time would remain on the order of seconds. Zones classified as CRITICAL could trigger immediate secondary inspection or adaptive path adjustment. The severity-coded damage map (CRITICAL/HIGH/MEDIUM/NONE) is broadly consistent with severity categories used in risk-based offshore inspection frameworks (DNV GL, 2019), which may simplify integration into existing maintenance workflows.

4. 6.4 Limitations and Future Work

This study has several important limitations. First, Stage 3 uses static dataset images rather than continuous UAV video frames; operational validation with actual flight data is required. Second, the crack IoU of 0.206 in the 4-class segmentation model falls below the 0.40 threshold considered operationally useful for crack boundary delineation; accordingly, the 3-class model is the more practical option at present. Third, the area estimation formula assumes fixed drone altitude and image coverage; production use requires GPS-tagged imagery for accurate area computation. Fourth, all experiments were run on CPU; proper GPU benchmarking should quantify actual latency and throughput on target deployment hardware.

Future work should include:

- Field validation using real offshore platform and bridge inspection flights.
- Crack-specific segmentation architectures that incorporate morphological priors for thin-structure detection.
- GPS-based damage geolocalization for spatially referenced inspection reports integrated with BIM models.
- Extension to additional offshore damage types, including weld defects, coating delamination, and marine growth.
- Multi-temporal change detection to track damage progression between inspection campaigns.

7. CONCLUSION

This study presented a three-stage deep learning pipeline for automated structural damage detection, segmentation, and UAV-based inspection simulation. The framework achieves 85.69% accuracy on four damage classes, mIoU of 0.7890 for 3-class pixel-level segmentation, and 80% zone detection in simulated UAV grid inspection. The measured CPU inference rate of 61 FPS suggests that near-real-time Stage 1 deployment may be feasible without GPU acceleration. The key technical contributions include a differential learning rate strategy that improved accuracy by 8.25% over frozen baselines, targeted augmentation that improved minority-class spalling recognition under a balanced evaluation protocol, integration of real UAV crack data into a four-class segmentation pipeline, and a severity-coded damage map aligned with offshore structural integrity management frameworks. Ablation studies, per-class IoU analysis, Grad-CAM visualization, and comparison with published methods support the technical relevance of the proposed approach. The code repository and trained model weights are publicly available as noted in the Data Availability statement.

FUNDING

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

Seyed Farhad Abtahi: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Data Curation, Writing - Original Draft, Writing - Review and Editing, Visualization, Project Administration.

DATA AVAILABILITY

All code, trained model weights, and result files are publicly available at: https://github.com/far-reach/structural-crack-detection (MIT License). Datasets used are publicly available at sources cited in section 4.1.

DECLARATION OF COMPETING INTEREST

The author declares no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ACKNOWLEDGEMENTS

The creators of the SDNET2018, Raidathmane, and Ziya07 datasets are acknowledged for making their data publicly available.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

During the preparation of this manuscript, the author used ChatGPT (OpenAI) for language editing assistance. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

REFERENCES

Bang, S., Park, S., Kim, H., and Kim, H. (2019). Encoder-decoder network for pixel-level road crack detection in black-box images. Computer-Aided Civil and Infrastructure Engineering, 34(8), 713-727.

Brownjohn, J. M. W. (2007). Structural health monitoring of civil infrastructure. Philosophical Transactions of the Royal Society A, 365(1851), 589-622.

Cha, Y. J., Choi, W., and Buyukozturk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361-378.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 801-818).

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 248-255).

DNV GL. (2019). Offshore standard DNVGL-OS-C101: Design of offshore steel structures, general - LRFD method. Det Norske Veritas.

Dorafshan, S., Thomas, R. J., and Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. Construction and Building Materials, 186, 1031-1045.

Dung, C. V., and Anh, L. D. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction, 99, 52-58.

Fan, Z., Li, C., Chen, Y., Di Mascio, P., Chen, X., Zhu, G., and Loprencipe, G. (2020). Ensemble of deep convolutional neural networks for automatic pavement crack detection and measurement. Coatings, 10(2), 152.

Gao, Y., and Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. Computer-Aided Civil and Infrastructure Engineering, 33(9), 748-768.

Gopalakrishnan, K., Khaitan, S. K., Choudhary, A., and Agrawal, A. (2017). Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. Construction and Building Materials, 157, 322-330.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770-778).

Hoskere, V., Narazaki, Y., Hoang, T. A., and Spencer, B. F. (2018). Vision-based structural inspection using multiscale deep convolutional neural networks. arXiv preprint arXiv:1805.01055.

Howard, J., and Ruder, S. (2018). Universal language model fine-tuning for text classification. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (pp. 328-339).

Kang, D., and Cha, Y. J. (2018). Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. Computer-Aided Civil and Infrastructure Engineering, 33(10), 885-902.

Kim, G., and Cha, Y. J. (2025). Deep learning-based 3D image reconstruction and damage mapping using neural radiance fields (Nerfacto). Structural Health Monitoring. https://doi.org/10.1177/14759217251340416

Li, S., Zhao, X., and Zhou, G. (2019). Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. Computer-Aided Civil and Infrastructure Engineering, 34(7), 616-634.

Li, T., Jiang, C., Zhou, Q., Ma, Z., and Sun, H. (2025). Asymmetric multi-task deep learning with location attention for structural damage detection. Structural Health Monitoring. https://doi.org/10.1177/14759217251343337

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 3431-3440).

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

Mei, Q., and Gul, M. (2020). A cost effective solution for pavement crack inspection using cameras and deep neural networks. Construction and Building Materials, 256, 119397.

Mishra, A., Gangisetti, G., Eftekhar Azam, Y., and Khazanchi, D. (2024). Weakly supervised crack segmentation using crack attention networks on concrete structures. Structural Health Monitoring, 23(6), 3748-3777. https://doi.org/10.1177/14759217241228150

Morgenthal, G., and Hallermann, N. (2014). Quality assessment of unmanned aerial vehicle (UAV) based visual inspection of structures. Advances in Structural Engineering, 17(3), 289-302.

Pan, S. J., and Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.

Paszke, A., Gross, S., Massa, F., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32.

Phung, M. D., Quach, C. H., Dinh, T. H., and Ha, Q. (2017). Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection. Automation in Construction, 81, 25-33.

Raidathmane. (2025). Corrosion and spalling concrete defect segmentation dataset. Kaggle. https://www.kaggle.com/datasets/raidathmane/corrosion-and-spalling-concrete-defect-segmentation

Rakha, T., and Gorodetsky, A. (2018). Review of unmanned aerial system (UAS) applications in the built environment: Towards automated building inspection procedures using drones. Automation in Construction, 93, 252-264.

Rashidi, M., Mohammadi, M., Sadeghlou Kivi, S., Abdolvand, M. M., Truong-Hong, L., and Samali, B. (2020). A decade of modern bridge monitoring using terrestrial laser scanning: Review and future directions. Remote Sensing, 12(22), 3796.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision (ICCV) (pp. 618-626).

Seo, J., Duque, L., and Wacker, J. (2018). Drone-enabled bridge inspection methodology and application. Automation in Construction, 94, 112-126.

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

Shihavuddin, A., Chen, X., Fedorov, V., Christensen, A. N., Riis, N. A. B., Branner, K., and Larsen, R. (2019). Wind turbine surface damage detection by deep learning aided drone inspection analysis. Energies, 12(4), 676.

Waqas, A., Kang, D., and Cha, Y. J. (2024). Deep learning-based obstacle-avoiding autonomous UAVs with fiducial marker-based localization for structural health monitoring. Structural Health Monitoring, 23(2), 971-990.

Xu, Y., Wei, S., Bao, Y., and Li, H. (2019). Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network. Structural Control and Health Monitoring, 26(3), e2313.

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., and Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully convolutional network. Computer-Aided Civil and Infrastructure Engineering, 33(12), 1090-1109.

Zhang, L., Yang, F., Zhang, Y. D., and Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. In Proceedings of the IEEE International Conference on Image Processing (pp. 3708-3712).

Ziya07. (2025). UAV-based crack detection dataset. Kaggle. https://www.kaggle.com/datasets/ziya07/uav-based-crack-detection-dataset

Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q., and Wang, S. (2019). DeepCrack: Learning hierarchical convolutional features for crack detection. IEEE Transactions on Image Processing, 28(3), 1498-1512.