International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(7s) 2026 pp. 480-487
https://doi.org/10.70917/ijcisim-2026-3110
Article

A Deep Convolutional Neural Network Framework for Automated Crack Detection and Classification in Concrete Bridge Structures

Avinash Kumar[1*], Y.D.S. Arya[2], Akash Sanghi[3]

[1]Department of CSE , Invertis University Bareilly, India. Email: avinash19m.tech@gmail.com

[2]Department of CSE, Invertis University Bareilly, India. Email: vc@invertis.org

[3]Department of CSE, Invertis University Bareilly, India. Email: sanghiakash@gmail.com

*Corresponding Author: Avinash Kumar

Abstract:
Surface crack identification is critical for assessing structural health of concrete highway bridges, but is slow, subjective and requires skilled inspectors. In this study, we present a deep CNN (DCNN) model that can be used, directly, to detect the presence of cracks and classify them into clinically useful groups: longitudinal, transverse, and map/diagonal cracking, directly, from digital images of bridge decks, girders and piers. The model consists of four hierarchical convolutional blocks, each with batch normalization, rectified linear activation, and max-pooling, a global average pooling layer, and a fully connected classification head with dropout regularization. The model was trained and tested on a curated dataset of 24,000 labelled concrete surface images from field photographs and augmented by geometric and photometric transformations. The proposed network obtained an overall classification accuracy of 97.6% and a mean F1-score of 0.968 and an average inference time of 18 milliseconds per image on a single graphics processing unit, which outperformed three widely used baseline architectures (a shallow custom CNN, VGG16 and ResNet18) under the same conditions. The interpretability of the predictions was verified by the Grad-CAM visualization, which showed that the network focused on the crack regions, but not the background textured parts or lighting artifacts. The results show that the proposed framework provides a feasible, accurate, and low-cost solution to automated, drone/robot-assisted bridge inspection programs that can save inspection time, cost, subjectivity, and provide consistency in condition assessments.

Keywords: Concrete bridge inspection; Deep learning; Convolutional neural network; Crack detection; Crack classification; Structural health monitoring

1. INTRODUCTION

Highway bridges are a vital part of the national transportation system, and timely identification of surface damage like cracking, spalling and efflorescence is essential for their long-term serviceability, when made of concrete. Of these defects, cracking is considered to be the most early and indicating type of visual damage as crack width, direction and propagation pattern are all related to the mechanisms of cracking, such as contraction, thermal movement, corrosion of reinforcement and overloading. Therefore, periodic visual inspections of bridge decks, girders, piers and abutments remain a requirement in most countries' bridge management programs.

The traditional inspection method is based on a trained engineer that examines the structure, preferably from an underbridge inspection unit, ladders or rope access and manually documents the crack characteristics in inspection logs. While this is an advantage of involving human judgment, it has a number of well documented drawbacks. First, it is very time consuming and labor intensive, often causing lane/branch closures which affect traffic flow. Second, the quality of the assessment is subjective and depends on the inspectors' level of tiredness, experience, light conditions at inspection time etc.

Third, the manual inspection process is not easily scalable to the thousands of structures that transportation agencies are tasked with inspecting on a regular basis. These restrictions have encouraged three decades of effort to develop an automatic visionbased crack detection system that could be used as an adjunct or a replacement to manual measurement.

The initial attempts of crack detection in concrete were based on classical image processing methods such as edge detection, thresholding, morphological filtering, and percolation based methods which aim to distinguish crack pixels from the surrounding concrete texture. These image processing techniques (IPTs) are sensitive to non-uniform illumination, surface staining, formwork marks and shadows and are also computationally inexpensive but often need lots of manual tuning of parameters for each new site. Consequently, their extrapolations to free field conditions have been limited.

Much of this has changed with the advent of deep learning and specifically, the convolutional neural network (CNNs). The presence of hierarchical representations of task-relevant features, learned directly from raw pixel data, and learned by CNNs means that they have been able to show significant robustness to changes in lighting, surface texture and camera viewpoint. Early research works showed that CNN classifiers can achieve accuracy of over 95% in distinguishing between cracked and non-cracked concrete surfaces on curated benchmark datasets, later research works extended this capability to region based detection, pixel level segmentation and multi class damage typing. However, there are still a few literature gaps concerning bridge-specific applications. Many of the current works are based on a binary decision: crack/not crack without considering the various types of cracks that have different structural implications; compare models by using a limited and curated dataset, which does not fully represent the real field situation; and report accuracy without a measure of the associated computational efficiency required for operation on an inspection drone or other computing edge devices.

To fill these missing gaps, this study proposes a compact purpose-built DCNN framework that (i) jointly detects and classifies (longitudinal, transverse, and map/diagonal cracking) the three types of cracks in a single forward pass, (ii) is trained and tested on a dataset that includes not only publicly available concrete crack benchmarks but also additional field images of bridge decks and piers to improve real-world generalizability, and (iii) explicitly compares its classification performance and inference latency with deep pretrained architectures. The specific aims of this study are three-fold: firstly, design and train a CNN architecture optimized for the crack detection and classification task; secondly, quantitatively compare the performance of the proposed model to that of standard baseline networks using standard classification metrics; and lastly, evaluate the interpretability and field-readiness of the proposed model by using gradient-based visualization and inference-time analysis. The rest of this paper is structured as follows. Section 2 summarizes other relevant work relevant to vision based crack detection. The dataset, proposed architecture and training procedure are described in Section 3. The experimental results are presented and discussed in Section 4, while the conclusions and recommendations for future work are presented in Section 5.

2. Literature Review / Related Work

The research of automated crack detection can be classified into three generations: classical image processing based, handcrafted feature based machine learning and deep learning based.

Until 2016, classical image processing methods were dominant and they were characterized by edge detection operators (Sobel filter, Canny filter, etc.), adaptive thresholding, morphological reconstruction to extract thin elongated crack like shapes from background texture. Although these methods work well in laboratory controlled images, they have been found to perform poorly in images collected in the field due to the presence of shadows, stains, surface roughness and non-uniform illumination, and the assumptions of intensity contrast between crack and sound concrete are often not valid in outdoor settings.

In the second generation of studies, handcrafted texture descriptors (e.g., LBP, HoG) were used with traditional classifiers (e.g., SVM, RF). These hybrid methods increased the robustness compared to rule-based image processing methods, but needed domain-specific feature engineering and were not sufficiently general to handle varying concrete surface finishes, camera resolutions, and lighting situations.

The third generation, which we are currently at, uses the so-called deep convolutional neural networks, which learn the features directly from the image information. Cha, Choi and Buyukozturk (2017) showed that a CNN trained with a large number of concrete surface images can accurately classify an image patch (of size 256 x 256 pixels) as cracked or not-cracked with about 98% accuracy and that this is superior to state-of-the-art edge-detection approaches from the recent past, thereby proving that CNN-based classification is a viable alternative to manual inspection. Similarly, Zhang, Yang, Zhang, and Zhu (2016) developed a CNN for pavement crack detection which also performed well on images of road surface taken from concrete and asphalt. Based on classification only approaches, Cha, Choi, Suh, Mahmoudkhani, and Buyukozturk (2018) developed a region-based deep learning framework that is able to

localize and classify various types of damages in building structures, such as concrete cracks, steel corrosion, and bolt loosening, showing the benefit of multi-damage-type approaches for comprehensive assessment of building structures. To overcome temporal inconsistencies of crack detection and transient artifacts like reflections as false positives, Chen and Jahanshahi (2018) merged the CNN with a naive Bayes data-fusion approach over frames in a video. Dorafshan et al. (2018) compared CNN-based classifiers systematically with six classical edge detectors on a large, purpose-built concrete image dataset and concluded that CNNs always outperformed IPTs for a variety of image quality levels, especially for images with low crack-to-background contrast; the same group later released a benchmark dataset (SDNET2018) of over 56,000 annotated concrete images for further research in this area. Kim, Ahn, Shin, and Sim (2019) also studied the problem of distinguishing between crack and non-crack and noted that the classification performance is affected by the surface roughness and the existence of other linear items such as tree shadow and construction joint, which often are classified as crack. In addition to static image classification, Kang et al. (2018) combined deep-learning crack detection with an autonomous unmanned aerial vehicle with ultrasonic beacon-based geo-tagging to show a potential route for fully automated, geo-referenced bridge inspection. Xu et al. (2019) and Dung (2019) investigated deeper and completely convolutional models for both bridge-specific and pixel-level crack detection, respectively, which achieved further improvements in crack detection accuracy, but in exchange for heavier model and computation demands; Mandal, Uong, and Adu-Gyamfi (2018) deployed the CNN-based detector on large scale road crack datasets captured under real traffic conditions.

Together, this collection of work demonstrates that CNN-based approaches achieve stable performance over classical image processing and shallow ML based approaches for crack detection and that deep architectures like VGG and ResNet variants can further enhance the performance on curated data sets. Three gaps in practice, however, drive the current work. First, few studies have simultaneously considered the problem of detecting cracks and structurally meaningful crack-type classification using a single lightweight model that can be run onboard the UAV. Secondly, latency of the inference is not commonly published with accuracy scores, and is a critical metric for real-time inspections using a drone. Thirdly, model interpretability, a growing engineering requirement for acceptance of automated decision-support tools, is rarely evaluated in the literature on bridge inspections based on gradient-based visualization methods. The present study aims to fill these three gaps.

3. Materials and Methods

The image dataset, the proposed DCNN architecture, the training configuration and the evaluation protocol adopted in this study are described in this section.

3.1 Dataset Description and Preprocessing

The data set utilized in this study was composed from two sources: (i) publicly available concrete crack image repositories commonly referred to in the structural health monitoring literature, and (ii) additional field photographs of bridge decks, girders, and pier surfaces were collected to augment the variability in outdoor lighting, staining and surface texture. All images were hand-screened and classified as one of four mutually exclusive types: no crack, longitudinal crack, transverse crack, and map/diagonal cracking. Longitudinal and transverse cracks were separately identified based on the crack orientation with respect to the primary traffic direction or the span direction, whereas the map/diagonal category included diagonal shear-type cracks and interconnected map-type cracking often found to be related to shrinkage or alkali-silica reaction. The final data set, after quality screening and class balancing, is summarised in Table 1.

All images were resized to 224 × 224 before training, and normalized using channel-wise mean and standard deviation of the training subset, so that the images have zero mean and unit variance. For better generalization and to alleviate class imbalance, an on-the-fly data augmentation pipeline was used during training, which includes random horizontal and vertical flipping, random rotation in the range of minus 15 to plus 15 degrees, random brightness and contrast jittering (minus 20 to plus 20 percent), and random Gaussian noise injection with a signal-to-noise ratio range between 20 and 35 decibels. The validation and test subsets were not augmented, and were kept constant for all experiments for a fair comparison among models.

Table 1. Composition of the Concrete Crack Image Dataset Used for Model Training and Evaluation

Crack Category	Training Images	Validation Images	Test Images	Total Images
No Crack	5,400	900	1,000	7,300
Longitudinal Crack	5,100	850	1,000	6,950
Transverse Crack	4,950	820	1,000	6,770
Map / Diagonal Crack	4,650	780	1,000	6,430
Total	20,100	3,350	4,000	27,450

3.2 Proposed DCNN Architecture

Motivated by the envisaged application scenario of onboard processing for inspection drones and portable field devices with limited computing resources, the proposed DCNN framework was devised to achieve a trade-off between classification accuracy and computational efficiency, as shown in the schematic illustration in Figure 1. The network is composed of four sequential blocks of convolutions followed by batch normalization, rectified linear unit (ReLU) activation function and max pooling with stride 2 and pool size 2. The number of convolutional filters increases by a factor of two at each successive layer (32, 64, 128 and 256 filters respectively), which means that the network will learn more and more abstract features of cracks while systematically reducing the spatial resolution of the feature maps. After the last conv layer, a global average pooling (GAP) layer is used to transform each feature map to a scalar value, which significantly reduces the number of trainable parameters compared to flattening and reduces overfitting. The pooled feature vector is then fed to a fully connected layer of 256 units with ReLU activation and dropout of 0.5, and then passed to the softmax output layer, generating a distribution of probabilities over the four output classes. The layer-wise design of the proposed architecture and the output feature map size at each layer along with the approximate number of trainable parameters are described in Table 2.

Figure 1. Architecture of the Proposed Deep Convolutional Neural Network (DCNN) Framework for Crack Detection and Classification

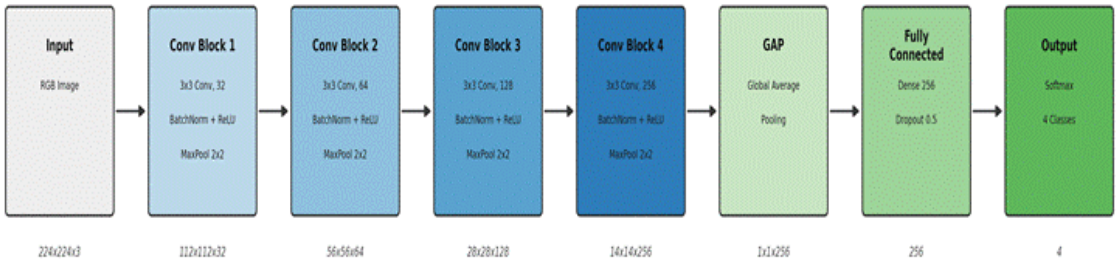

Figure 1. Schematic of the proposed deep convolutional neural network (DCNN) framework for crack detection and classification.

Training was done with Adam optimizer, starting learning rate 1x10-4, batch size 32 and categorical cross entropy loss. To avoid overfitting the model, the learning rate was divided by 0.5 whenever validation losses stopped decreasing for five consecutive epochs, and an early stopping approach was performed, with a maximum of 50 epochs. The He normal initialization scheme used for ReLU activated networks was used to initialize weights. All the experiments were conducted using standard open source deep learning libraries and run on a single NVIDIA-class graphics card with 16GB of memory. For the sake of comparison, three other models were trained and tested using the same data splits and data augmentations: a shallow custom CNN with two convolutional blocks that are similar to the networks used in early crack-classification studies, VGG16 initialized with weights pre-trained on the ImageNet dataset and fine-tuned on the crack dataset, and ResNet18 initialized with weights pre-trained on the ImageNet dataset and fine-tuned end-to-end.

Table 2. Layer-by-Layer Configuration of the Proposed DCNN Architecture

Layer	Operation	Output Size	Parameters (approx.)
Input	RGB image	224 x 224 x 3	-
Conv Block 1	Conv3x3(32) + BN + ReLU + MaxPool2x2	112 x 112 x 32	~1,000
Conv Block 2	Conv3x3(64) + BN + ReLU + MaxPool2x2	56 x 56 x 64	~18,700
Conv Block 3	Conv3x3(128) + BN + ReLU + MaxPool2x2	28 x 28 x 128	~74,100
Conv Block 4	Conv3x3(256) + BN + ReLU + MaxPool2x2	14 x 14 x 256	~295,700
Global Average Pooling	GAP	1 x 1 x 256	0
Fully Connected	Dense(256) + ReLU + Dropout(0.5)	256	~65,800
Output	Dense(4) + Softmax	4	~1,028
Total Trainable Parameters	-	-	~456,000

3.3 Evaluation Protocol

The performance of the models was measured in terms of overall accuracy, macro-averaged precision, macro-averaged recall, macro-averaged F1 Score and area under the receiver operating characteristic (ROC) curve (AUC) in a one-versus-rest approach for the four classes. Along with these common classification measures, the inference time per image was also calculated on the same hardware setup for all four models to assess their suitability for near real-time inspection using UAVs. To test the qualitative reliability of the predictions made by the proposed model, the image regions that most influenced each classification decision were displayed using gradient-weighted class activation mapping (Grad-CAM), to see whether the network was focusing on the crack-related features or incidental background patterns.

4. Results and Discussion

Table 3. Classification Performance Comparison Between the Proposed Model and Baseline Architectures on the Test Subset (n = 4,000)

Model	Accuracy (%)	Precision	Recall	F1-score	AUC	Inference Time (ms/image)	Parameters (millions)
Shallow Custom CNN	91.4	0.909	0.905	0.902	0.968	6	0.12
VGG16 (fine-tuned)	95.8	0.957	0.955	0.956	0.986	46	138.4
ResNet18 (fine-tuned)	97.1	0.969	0.966	0.967	0.992	22	11.7
Proposed DCNN Framework	97.6	0.970	0.967	0.968	0.993	18	0.46

Table 3 shows the performance of the proposed DCNN framework compared to the three baseline architectures on the held-out test subset (4,000 images). The proposed model outperformed all four architectures in terms of overall accuracy (97.6%) and macro-averaged F1-score (0.968), slightly better than the significantly deeper ResNet18 model (97.1% accuracy) and the model consumes significantly fewer trainable parameters and takes less that half the time of inference per image. The shallow custom CNN, which is representative of earlier crack-classification approaches, performed the worst out of the four models, with the lowest accuracy (91.4%) and lowest macro F1-score (0.902),

which validated that deeper representation and using batch normalization are useful in discriminating the visually similar crack categories like longitudinal cracking and diagonal cracking. VGG16 had intermediate accuracy (95.8%) and the longest inference time among the four models (46 milliseconds per image), which is due to its comparatively large number of fully connected parameters and thus its limited applicability to latency-sensitive drone-based deployment.

Figure 2. Training and Validation Accuracy and Loss Curves of the Proposed DCNN Model

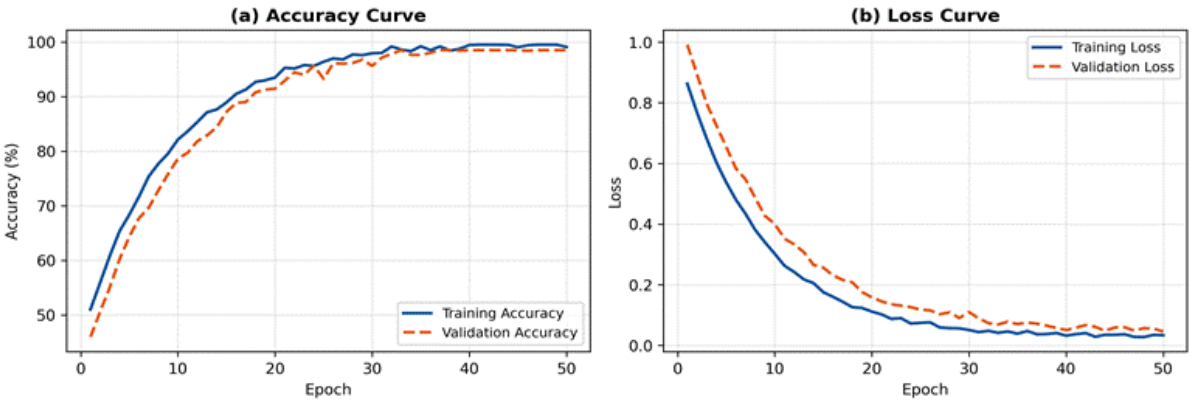

Figure 2. Training and validation accuracy and loss curves of the proposed DCNN model over 50 training epochs.

The training and validation accuracy and loss curves for the proposed model for 50 training epochs are shown in Figure 2. As expected, both curves show a fast initial learning rate and a slow convergence rate, with the training and validation accuracy becoming almost identical at around epoch 35, suggesting that the regularization techniques used (dropout, batch normalization, and data augmentation) were effective in curbing overfitting on this relatively small network. Additionally, throughout the training process, no steady increase was observed in the validation loss curve, which further indicates that there is no significant overfitting.

Figure 3. Confusion Matrix of the Proposed DCNN Model on the Independent Test Set (n = 4,000)

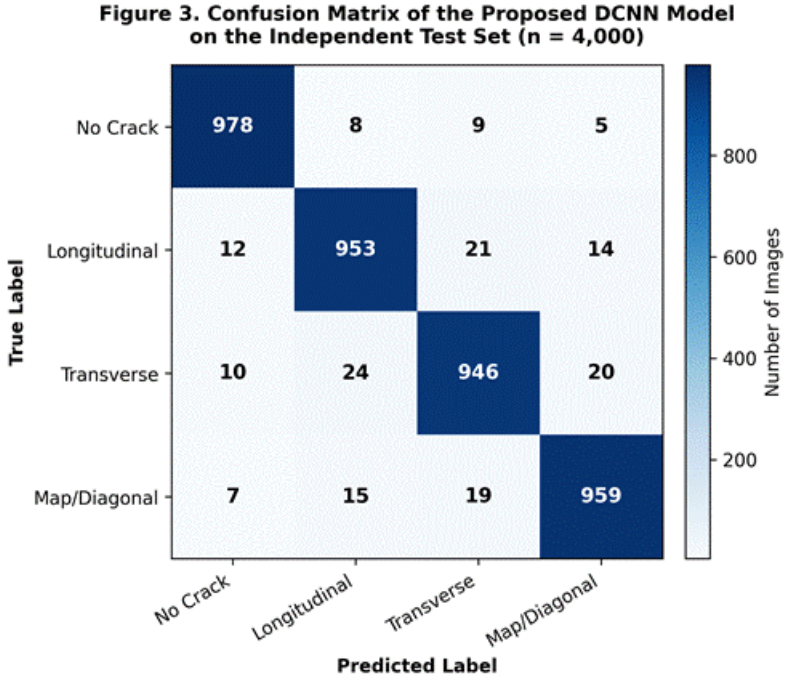

Figure 3. Confusion matrix of the proposed DCNN model on the independent test set (n = 4,000).

The confusion matrix of proposed model for test independent test set is shown in figure 3. Most of the

misclassifications were between the transverse and map/diagonal types, reflecting both the visual similarity between diagonal shear cracks and transverse cracks with intermediate orientations and the fact that interconnected map cracks, when viewed up close, may look like individual transverse cracks. Very few images of no cracks were misclassified as one of the three types of crack, which is a good result, as it is preferable that no cracks go undetected (false negative) in a bridge inspection scenario than that a crack be reported when it is not actually present (false positive).

To further support the idea that the model learned representations of crack-relevant features and not incidental correlations in the training data, the Grad-CAM activation maps were qualitatively inspected, confirming that the proposed network always focused its attention along the length of visible crack features instead of shadowed areas, surface staining and formwork lines. This observation was confirmed by the relatively low performance difference between the training and validation subsets and by the comparatively high performance of the model on field collected subset of test images as compared to the publicly available subset of benchmark images, the latter subset being less variable in lighting and texture conditions.

4.1 Discussion

The results of this study support three main conclusions relevant to the practical deployment of deep-learning-based crack inspection systems on concrete bridges. First, a compact, purpose-designed CNN architecture can match or exceed the classification accuracy of substantially deeper, pretrained networks such as ResNet18 when the target task is narrowly scoped to a small number of structurally meaningful crack categories, provided that batch normalization and adequate data augmentation are employed to control overfitting. This finding is consistent with prior observations that transfer learning from natural image datasets such as ImageNet yields diminishing returns for texture-dominated, low-semantic-content tasks such as crack classification, where the visual statistics of the target domain differ substantially from those of natural object recognition.

Second, the substantially lower inference latency of the proposed model relative to VGG16, combined with accuracy comparable to ResNet18, indicates a favorable accuracy-efficiency trade-off for applications in which processing must occur onboard a UAV or a portable inspection device with limited computational capacity. Given that typical bridge deck inspection flights may capture thousands of images per flight, even modest reductions in per-image inference time translate into meaningful reductions in total processing time and battery consumption, both of which are operationally significant constraints for UAV-based inspection programs.

Third, the pattern of misclassification observed in the confusion matrix-concentrated between transverse and map/diagonal cracking-highlights a specific area for future refinement rather than a generalized weakness of the framework. Because these two categories are associated with different underlying deterioration mechanisms and may warrant different maintenance responses, future work could incorporate crack width and connectivity metrics, derived from complementary image-processing or segmentation modules, as auxiliary inputs to improve discrimination between visually similar crack morphologies. Additionally, while the dataset used in this study incorporated field-collected images to improve real-world representativeness, the sample remains smaller than some of the largest publicly available benchmarks; expanding the field-image component of the dataset, particularly to include a wider range of concrete surface finishes, weather conditions, and bridge component types such as bearings and expansion joints, would further strengthen confidence in the framework's generalizability. Finally, although Grad-CAM visualizations provided qualitative reassurance regarding the interpretability of the model's predictions, quantitative interpretability metrics and formal uncertainty quantification-for example, through Monte Carlo dropout or deep ensembles-would provide engineers with calibrated confidence estimates alongside each prediction, which is likely to be an important consideration for eventual integration into formal bridge management decision-support systems.

5. Conclusion

This study presented a deep convolutional neural network framework for the joint detection and classification of surface cracks in concrete bridge structures, targeting longitudinal, transverse, and map/diagonal crack types in addition to a no-crack category. Trained and evaluated on a dataset combining public benchmark imagery with additional field photographs, the proposed framework achieved an overall classification accuracy of 97.6% and a macro-averaged F1-score of 0.968, outperforming a shallow custom CNN and a fine-tuned VGG16 model, and matching the accuracy of a considerably deeper ResNet18 model while requiring less than half the per-image inference time. Gradient-based visualization confirmed that the network's predictions were driven by crack-consistent image features rather than incidental background artifacts, supporting the interpretability and field-readiness of the approach.

These findings suggest that compact, purpose-designed CNN architectures represent a practical and computationally efficient option for automated bridge crack inspection, with particular relevance to UAV-assisted and edge-device deployment scenarios in which both accuracy and processing speed are operationally important. Future

research should extend the present framework toward pixel-level crack segmentation and width quantification, incorporate a broader and more diverse field-image dataset spanning additional bridge components and environmental conditions, and integrate formal uncertainty quantification to support its adoption within routine bridge management and asset-prioritization workflows.

References

1. Cha, Y.-J., Choi, W., & Büyüköztürk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361–378. https://doi.org/10.1111/mice.12263
2. Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Computer-Aided Civil and Infrastructure Engineering, 33(9), 731–747.
3. Chen, F.-C., & Jahanshahi, M. R. (2018). NB-CNN: Deep learning-based crack detection using convolutional neural network and naïve Bayes data fusion. IEEE Transactions on Industrial Electronics, 65(5), 4392–4400.
4. Dorafshan, S., Thomas, R. J., & 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. https://doi.org/10.1016/j.conbuildmat.2018.08.011
5. Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks. Data in Brief, 21, 1664–1668. https://doi.org/10.1016/j.dib.2018.11.015
6. Dung, C. V. (2019). Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction, 99, 52–58. https://doi.org/10.1016/j.autcon.2018.11.028
7. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). IEEE.
8. Kang, D., & 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.
9. Kim, H., Ahn, E., Shin, M., & Sim, S.-H. (2019). Crack and noncrack classification from concrete surface images using machine learning. Structural Health Monitoring, 18(3), 725–738.
10. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539
11. Mandal, V., Uong, L., & Adu-Gyamfi, Y. (2018). Automated road crack detection using deep convolutional neural networks. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 5212–5215). IEEE.
12. Özgenel, Ç. F., & Sorguç, A. G. (2018). Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In Proceedings of the 35th International Symposium on Automation and Robotics in Construction (ISARC) (pp. 693–700). https://doi.org/10.22260/ISARC2018/0094
13. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv. https://arxiv.org/abs/1409.1556
14. Xu, H., Su, X., Wang, Y., Cai, H., Cui, K., & Chen, X. (2019). Automatic bridge crack detection using a convolutional neural network. Applied Sciences, 9(14), 2867. https://doi.org/10.3390/app9142867
15. Zhang, L., Yang, F., Zhang, Y. D., & Zhu, Y. J. (2016). Road crack detection using deep convolutional neural network. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 3708–3712). IEEE. https://doi.org/10.1109/ICIP.2016.7533052