IJSCET

INTERNATIONAL JOURNAL OF SUSTAINABLE CONSTRUCTION ENGINEERING AND TECHNOLOGY
ISSN: 2180-3242 e-ISSN: 2600-7959

Vol. 17 No. 1 (2026) 148-168
https://publisher.uthm.edu.my/ojs/index.php/ijscet

Construction Quality Monitoring with Deep Learning-Based Automated Concrete Crack Detection: A Comparative Study

Thuy-Hien Thi Nguyen[1], Thuy-Anh Nguyen[1], Quan T. Nguyen[2]*

[1] *University of Transport Technology, Hanoi, 100000, VIETNAM*
[2] *Hanoi University of Civil Engineering, Hanoi, 100000, VIETNAM*

*Corresponding Author: quannt@huce.edu.vn
DOI: https://doi.org/10.30880/ijscet.2026.17.01.009

Article Info

Received: 9 December 2025
Accepted: 21 April 2026
Available online: 26 June 2026

Keywords

Concrete crack detection, construction quality monitoring, deep learning, Convolutional Neural Networks (CNN), transfer learning, DenseNet201

Abstract

Concrete cracking is a primary indicator of structural deterioration, yet traditional visual inspections are labor-intensive and subjective, limiting their effectiveness for large-scale monitoring. This study proposes an automated crack detection framework using Convolutional Neural Networks (CNNs) to address these challenges. Leveraging a balanced dataset of over 42,000 images, we conducted a systematic comparison of five transfer-learning models: VGG16, ResNet50, MobileNetV2, DenseNet121, and DenseNet201. Performance was evaluated across multiple metrics, including accuracy, F1-score, model size, and inference speed. The results identify DenseNet201 as the top performer, achieving a mean accuracy of 95.7% and an F1-score of 96.7%, driven by its superior stability across data partitions. Additionally, while MobileNetV2 offers a viable lightweight alternative for edge computing, DenseNet201 remains the most reliable architecture for high-precision quality monitoring tasks.

1. Introduction

Concrete is one of the most widely used construction materials worldwide, yet it remains highly susceptible to cracking caused by shrinkage, environmental stress, overloading, and material deterioration. Cracks not only reduce structural durability and serviceability but also accelerate processes such as moisture penetration and reinforcement corrosion [1, 2]. If undetected, these cracks can accelerate reinforcement corrosion and compromise structural integrity. For this reason, timely and accurate crack identification is important in construction quality management and in maintaining the long-term performance of built assets [3, 4].

Historically, crack detection has relied on manual visual inspection. While visual inspection remains the industry standard, it is inherently labor-intensive and subjective, often failing to meet the demands of large-scale infrastructure monitoring [5]. Consequently, automated diagnostic methods based on Convolutional Neural Networks (CNNs) have emerged as a robust alternative, offering superior accuracy and efficiency in feature extraction [6].

During the past decade, deep learning—particularly Convolutional Neural Networks (CNNs)—has become a standard approach for crack detection, outperforming classical image-processing techniques in accuracy, robustness, and feature representation [6, 7]. CNNs have demonstrated strong capability in learning discriminative crack patterns under diverse surface conditions, making them well-suited for large-scale visual inspection in construction. However, despite the increase of CNN-based studies, several methodological gaps

remain. Many prior works evaluate only one or two architectures, often under limited datasets, and offer little insight into why certain architectures generalize better or fail under varying texture and illumination conditions [8, 9].

To address these gaps, this study conducts a systematic comparative analysis of five representative CNN architectures—VGG16, ResNet50, MobileNetV2, DenseNet121, and DenseNet201—selected to capture different design philosophies ranging from sequential convolution to residual learning and dense feature reuse. Using a consolidated dataset of about 42,000 images and a multi-K-fold cross-validation framework, the study evaluates each model's accuracy, stability, and misclassification behavior. Grad-CAM visualizations further provide insight into model interpretability and the alignment between network attention and actual crack morphology.

The key contributions include:

(1) it provides an extensive performance comparison of widely used CNN architectures for binary crack classification;
(2) it examines diagnostic stability using an extensive multi-fold evaluation protocol; and
(3) it integrates quantitative performance with explainability analysis to identify the architectural mechanisms most influential for reliable crack detection.

These findings offer evidence-based guidance for selecting and deploying CNN models in automated construction quality management and structural condition assessment.

2. Literature Review

2.1. Concrete Defect Diagnosis in Quality Management

Effective quality management relies on the timely identification of defects to mitigate lifecycle costs [3]. While traditional manual inspections are constrained by subjectivity and scalability issues [10], the integration of digital technologies—ranging from UAVs to embedded sensing—has shifted the paradigm toward automated diagnostics. Within this context, deep learning has become central to establishing consistent and data-driven inspection frameworks.

Within the broader domain of building pathology and condition-assessment frameworks, crack detection is not only a matter of identifying visible defects but also a strategic tool for maintenance prioritization and risk-informed decision-making. Building pathology emphasizes the systematic interpretation of visual symptoms to understand their underlying causes, environmental drivers, and potential consequences [4]. Under this lens, crack patterns often serve as diagnostic markers that point to deeper structural, thermal, or material-related issues. Effective and timely detection therefore mitigates lifecycle costs by enabling preventive maintenance rather than reactive repair strategies [3].

Traditional inspection methods rely heavily on manual visual assessment conducted by trained inspectors, typically performed using close-range photographs or on-site walkthroughs. Although visual inspection remains the industry standard due to its simplicity and low cost, it is inherently subjective and highly sensitive to practitioner experience, environmental lighting conditions, and surface heterogeneity [5]. Fatigue, time pressure, and inconsistent inspection protocols further reduce diagnostic repeatability. In large-scale infrastructures—bridges, tunnels, viaducts, industrial plants—the sheer volume of surface area discourages thorough examination, resulting in missed hairline cracks or misinterpretation of surface artefacts such as stains, pores, and joints [10].

As infrastructure networks expand, these limitations increasingly undermine the reliability of manual inspection regimes. The global shift toward digital transformation in construction has therefore driven the adoption of automated and data-driven diagnostic tools that promise higher accuracy, consistency, and scalability [11]. Coupled with advancements in imaging technologies, unmanned aerial vehicles (UAVs), robotic inspection systems, and embedded sensing, the demand for robust automated crack detection systems has accelerated markedly [7]. Deep learning, in particular, has emerged as a transformative component of this new diagnostic paradigm.

2.2 Automated Crack Detection: From Traditional Methods to Deep Learning

Early automation attempts using classical computer-vision techniques (e.g., Canny, Sobel) struggled with environmental noise and complex surface textures [12]. In contrast, CNN-based approaches have demonstrated superior robustness by learning feature representations directly from data, significantly outperforming traditional image-processing algorithms [5].

Sensor-based methods such as acoustic emission, ultrasonic testing, and strain gauge monitoring were also explored for crack diagnosis. Although they provide rich physical insights, these methods are costly, invasive, and ill-suited for monitoring extensive surface regions. Their reliance on specialized equipment limits field scalability, making them impractical for routine inspections of large civil structures [10].

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168

The emergence of deep learning—in particular, Convolutional Neural Networks (CNNs)—has fundamentally improved the automation of crack detection. CNNs learn texture representations directly from data, enabling robust feature extraction without handcrafted processing. Numerous studies have demonstrated superior performance of CNN-based crack-detection models compared to classical methods, achieving higher accuracy, robustness under variable imaging conditions, and improved generalization across datasets [5-7].

Despite these advances, existing literature reveals several critical gaps:

- Limited comparative studies across multiple CNN architectures: Most studies evaluate only one or two models (e.g., VGG or ResNet), resulting in fragmented evidence. There is limited understanding of why certain architectures succeed or fail under specific crack patterns or environmental conditions [8, 9].

- Dataset limitations and domain gaps: Popular datasets such as SDNET2018 and METU often exhibit controlled lighting, cropped patches, and uniform backgrounds, which do not reflect the complexity of on-site images captured via UAVs or handheld devices. This mismatch leads to generalization problems during real-world deployment [13, 14].

- Weak integration of explainable AI (XAI): Few studies examine how CNNs make decisions, despite the importance of interpretability for quality assurance, auditing, and engineering acceptance [6]. Without transparency, neural networks risk misinterpreting stains or shadows as cracks, reducing trust in their diagnostic output.

Therefore, a comparison of multiple CNN families—paired with robust validation protocols and explainability analysis—is essential for establishing reliable automated crack-detection frameworks.

2.3 Typologies of Deep Learning Tasks for Crack Analysis

Deep learning has been applied to a spectrum of task categories that address different dimensions of crack assessment within condition monitoring and structural maintenance workflows.

(a) Binary classification

This remains the most widely used task due to its simplicity and suitability for rapid screening. The model assigns an image as either "crack" or "non-crack," supporting large-scale surveys and early detection workflows [5, 15]. However, classification does not provide information about crack geometry or severity.

(b) Semantic or instance segmentation

These tasks localize cracks at pixel-level resolution, enabling measurement of width, continuity, branching, and morphological characteristics. Such details are crucial for durability analysis and repair prioritization [16, 17]. Their primary limitations are extensive annotation requirements and higher computational demand.

(c) Geometry estimation

Estimations of crack width, length, curvature, and orientation enable deeper integration with structural mechanics and deterioration modeling. Advances in stereo imaging and 3D reconstruction have further expanded possibilities for volumetric crack assessment [8, 10].

(d) Severity or typology classification

This advanced task categorizes cracks by severity (hairline, moderate, severe) or by causation mechanism (thermal, shrinkage, structural). It offers the highest diagnostic value but is constrained by the scarcity of high-quality labeled datasets [18, 19].

These typologies illustrate a continuum from coarse detection to rich diagnostic assessment. Yet, the literature indicates that even binary classification becomes challenging in field environments when shadows, stains, and construction joints mimic crack-like patterns [11]. This emphasizes the need for architectures capable of capturing subtle textures across multi-scale contexts.

2.4 CNN Architectural Paradigms Relevant to Crack Detection

CNN architectural evolution has shaped the effectiveness of automated crack-detection systems. Each generational shift introduces mechanisms that enhance gradient propagation, feature reuse, or computational efficiency—factors crucial for identifying fine-grained surface textures characteristic of concrete cracks.

Sequential architectures (e.g., VGG16).

Using stacked 3×3 convolutions, these models excel at extracting low-level features but suffer from high parameter counts and limited capability to capture multi-scale crack patterns [20]. Their deep but linear design often leads to overfitting and instability under varied lighting conditions.

Residual architectures (e.g., ResNet50).

Residual connections mitigate vanishing gradients and support deeper representations [21]. However, their additive skip connections may suppress or blur fine crack features, reducing sensitivity to hairline fractures and making them prone to false negatives.

Lightweight architectures (e.g., MobileNetV2).

Employing inverted residuals and depthwise separable convolutions, MobileNetV2 offers competitive accuracy with significantly reduced computation [22]. This makes it suitable for real-time UAV or robot-based

inspection, though the reduced representational richness may limit crack detection under noise or shadow interference.

Densely connected architectures (e.g., DenseNet121/201).

Dense connectivity allows each layer to receive feature maps from all preceding layers, enabling multi-scale feature reuse and strong gradient flow [23]. This architectural property has been shown to improve performance on texture-sensitive tasks and is particularly advantageous for detecting thin cracks in noisy environments [9].

The literature reveals that comparisons across different CNN architectural families remain relatively sparse, and existing studies often present inconsistent findings due to variations in datasets, training strategies, or evaluation settings [24]. These inconsistencies highlight the need for more unified and systematic comparative studies conducted under controlled experimental conditions to provide clearer insights into model behaviour and architectural suitability.

However, even among the limited comparative works that exist, few studies incorporate explainability tools such as Grad-CAM to assess whether a model's decision-making process aligns with meaningful structural features required for engineering applications. Integrating explainability into crack-detection research is essential for ensuring interpretability and diagnostic transparency, particularly when deep learning tools are embedded in construction quality-management workflows [6].

This study responds to that need by selecting VGG16, ResNet50, MobileNetV2, DenseNet121, and DenseNet201 as representative CNN architectures for systematic comparison. Together, they span the major design philosophies from sequential to residual, lightweight, and dense connectivity, providing an extensive basis for assessing diagnostic reliability and architectural suitability in automated concrete crack detection.

3. Methodology

This section describes the methodological framework adopted to evaluate five Convolutional Neural Network (CNN) architectures for concrete crack detection. It covers dataset preparation, architecture configuration, training setup, cross-validation design, and evaluation metrics. These procedures follow established practices in civil engineering computer vision applications [5, 7].

3.1 Overall Methodological Pipeline

The study follows a structured methodological pipeline (Fig. 1) comprising dataset preparation, augmentation, model configuration, training, cross-validation, and performance evaluation.

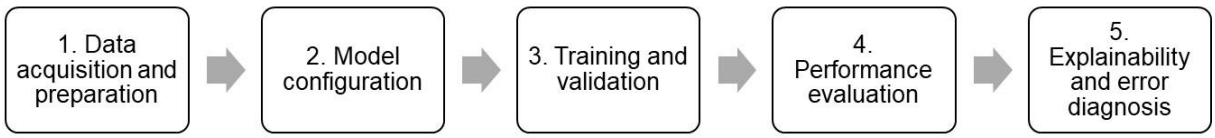


Fig. 1 *Conceptual methodological pipeline of the study*

This pipeline ensures that each model is trained and evaluated under consistent conditions, supporting a fair comparison of architectural performance. It was designed to support digital construction quality-monitoring workflows, where the ability to reliably automate crack identification directly affects inspection efficiency, defect triage, and safety assurance on construction sites.

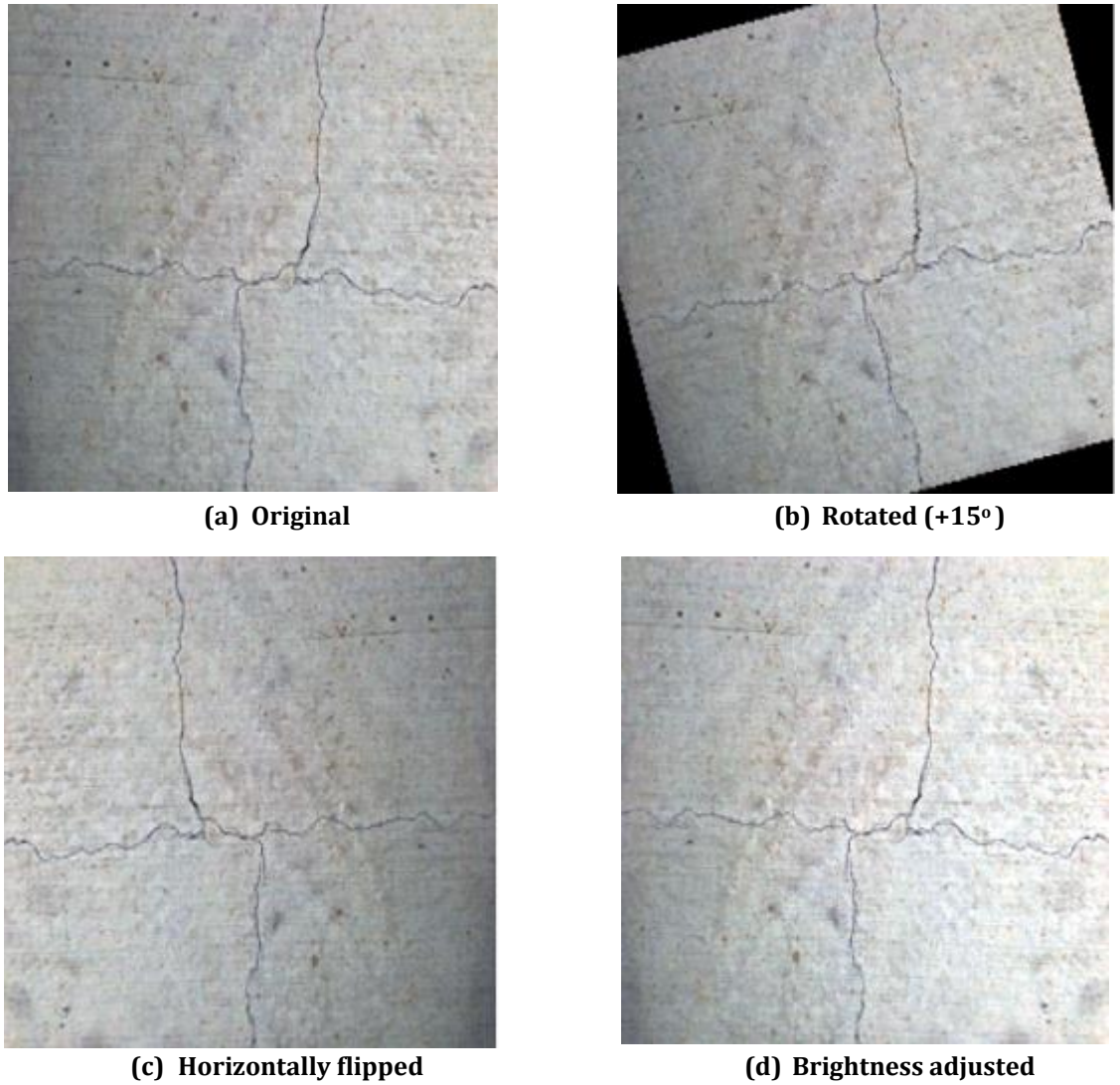
3.2 Dataset Description and Pre-processing

The study uses the publicly available *Concrete Crack Images for Classification* dataset, a widely adopted benchmark for evaluating crack detection models. After curation and augmentation, the working dataset comprised over 42,000 images, including approximately 26,500 cracked and 15,900 non-cracked samples.

All images were resized to 224 × 224 pixels, consistent with the input specifications of the selected CNN architectures [20, 21]. Standard pre-processing techniques were applied:

- Pixel-value normalization
- Random horizontal/vertical flips
- Random rotations
- Class balancing using augmentation and weighting

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168



(a) Original **(b) Rotated (+15°)**

(c) Horizontally flipped **(d) Brightness adjusted**

Fig. 2 *Examples of data augmentation applied to concrete surface images*

Data augmentation increases robustness against variations in texture, illumination, and crack morphology, which are common in field-collected images (Fig. 2).

3.3 Selection of CNN Architectures

Five representative CNN architectures were selected due to their demonstrated performance in image classification and their diversity in architectural design:

- VGG16, a deep sequential architecture using uniform 3 × 3filters [20];
- ResNet50, which incorporates residual connections for improved gradient propagation [21];
- MobileNetV2, designed for efficiency using inverted residual blocks [25];
- DenseNet121 and DenseNet201, which employ dense connectivity to promote feature reuse and multi-scale representation [23].

All models used ImageNet-pretrained weights [26]. The final fully connected classification layers were replaced with custom layers for binary crack–non-crack prediction, following best practices in transfer learning for infrastructure inspection [6].

3.4 Model Training and Evaluation

3.4.1 Training Setup

All models were trained using the Adam optimizer with a learning rate of 10^{-4} and the binary cross-entropy loss, which is appropriate for binary crack/non-crack classification problems [7]. Training was conducted for a maximum of 50 epochs with a batch size of 32. To prevent overfitting, an early-stopping strategy was applied based on validation loss: training was terminated if no improvement in validation loss was observed for 10 consecutive epochs. In all experiments, convergence was reached before the 50-epoch limit, indicating that the training schedule was sufficient for stable optimization.

3.4.2 Multi–K-Fold Cross-Validation Design

To obtain a reliable estimate of generalization performance and to analyse the sensitivity of each architecture to variations in data availability, a multi–K-fold cross-validation (CV) strategy was adopted. Instead of relying on a single train–validation split, the dataset was partitioned into multiple fold configurations, allowing the models to be trained and evaluated under different proportions of training and validation samples.

In this study, five values of K were selected—$K = 2,16,50,121$, and 201—representing a wide spectrum of data-partitioning scenarios. Smaller K values create larger validation sets and proportionally smaller training sets, thereby stressing the model's ability to learn from limited data. Larger K values do the opposite, providing richer training subsets but very small validation folds. This design captures realistic variability encountered in construction-site data collection, where the number and quality of available images may fluctuate across projects, inspection cycles, or environmental conditions.

For each K-fold configuration, the dataset was divided into K approximately equal subsets. Models were trained on $K - 1$ subsets and validated on the remaining subset, and the process was repeated until each subset had served once as the validation fold. Performance metrics for each fold were recorded and summarised using mean and standard deviation across all folds of the same K value. This procedure ensures that the reported results reflect not only the average predictive capability but also the stability of the model under different data-partitioning conditions.

The multi–K-fold design therefore provides a more detailed methodological baseline than single-split evaluations commonly used in the literature. It enables a detailed examination of how architectural characteristics influence robustness, variance, and class-wise discrimination when the models are exposed to diverse training–validation ratios—an important consideration for operationalizing deep learning–based inspection in construction quality-monitoring workflows.

3.4.3 Evaluation Metrics and Formulas

To assess the performance of the CNN architectures in a manner that reflects the diagnostic requirements of automated crack detection, the study employed a set of evaluation metrics that capture both overall predictive capability and class-specific reliability. Given the asymmetric risks of false positives (FP) and false negatives (FN) in construction quality monitoring, the analysis emphasised metrics that provide nuanced insights into each model's behaviour for the crack (positive) and non-crack (negative) classes.

Let TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. Overall accuracy was computed as:

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

Although accuracy offers a broad measure of correct classifications, it does not adequately reflect class imbalance or the differing diagnostic consequences of misclassifying cracks versus intact surfaces. Therefore, the evaluation also incorporated precision and recall, defined as:

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

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

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168

The primary class-wise indicator used in this study is the F1-score—the harmonic mean of precision and recall—calculated as:

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

F1-scores were computed separately for the positive (crack) and negative (non-crack) classes to provide deeper insight into how each model handles diagnostic trade-offs. The F1-positive score captures a model's ability to correctly identify actual cracks (highly prioritised to avoid missed defects), while the F1-negative score reflects its ability to correctly recognise intact surfaces (critical for minimising unnecessary follow-up inspections). For each architecture and each K-fold configuration, these metrics were aggregated using mean and standard deviation across folds, enabling a detailed assessment of both performance and stability.

This metric set aligns with current best practices in deep learning–based structural inspection, where class-wise discrimination and robustness under varying data conditions are essential for ensuring reliable, field-ready crack-detection performance. They reflect practical QA/QC requirements: false negatives may allow structural defects to persist, while false positives increase the verification burden for on-site inspectors.

3.5 Additional Evaluation of the Best-Performing Model

Beyond the comparative assessment of all architectures across multiple K-fold configurations, the study conducts an additional layer of analysis on the best-performing model identified through the multi–K-fold process. This supplementary evaluation aims to examine not only the model's aggregate predictive capability but also the reliability, interpretability, and diagnostic behaviour that are essential for deployment in construction quality-monitoring workflows.

Three analytical components are incorporated. First, a confusion-matrix evaluation is performed to quantify the balance between true positives, true negatives, false positives, and false negatives. This allows for a deeper understanding of class-wise diagnostic performance, particularly the trade-offs between identifying true cracks and avoiding false alarms in non-crack regions.

Second, stability and variance analysis is conducted by examining fold-level performance trends under the most robust K-fold configuration. This step helps validate whether the model maintains consistent behaviour when exposed to changes in data composition, an important criterion for operational reliability when image availability and field conditions fluctuate.

Finally, an explainability assessment using Gradient-weighted Class Activation Mapping (Grad-CAM)—a widely adopted method for visualising model decision pathways [27]—is applied to identify the regions of interest that drive the model's predictions. By examining activation patterns for both correct and misclassified samples, this analysis provides insight into whether the model's decision-making aligns with physically meaningful crack features and helps identify the underlying causes of failure modes that may arise in practical settings.

The results of these additional evaluations are presented and discussed in Section 4, offering a comprehensive interpretation of model behaviour that extends beyond aggregate accuracy metrics and supports its suitability for real-world automated inspection and digital QA/QC applications.

4. Results and Discussion

This section presents the experimental outcomes of the five evaluated CNN architectures. Results include training–validation behaviour, comparative performance across multi–K-fold cross-validation configurations, and an in-depth analysis of the best-performing model. Interpretations are provided in the context of previous studies on deep learning based crack detection [5-7].

4.1 Additional Evaluation: Best-Performing Model Analysis

The training process was monitored by recording the accuracy and loss on both the training and validation datasets for each model across all epochs. For the DenseNet121 model, the validation accuracy and loss curves closely followed the respective training curves, as shown in Fig. 3. Although minor fluctuations were observed in the validation metrics, the close alignment between training and validation performance suggests a stable learning process and indicates that the model did not experience significant overfitting.

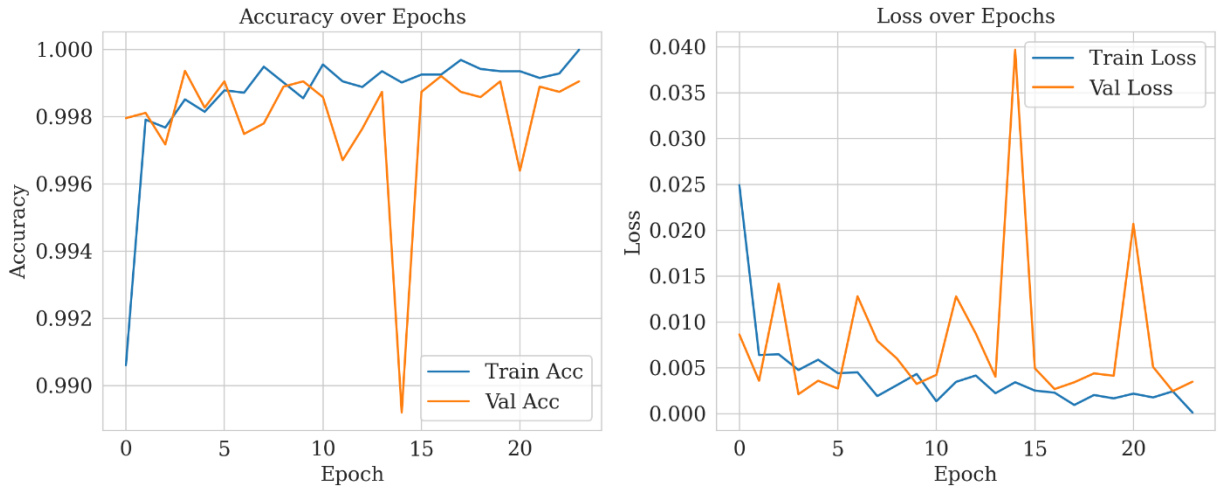


Fig. 3 *Learning curves for DenseNet121 model*

The VGG16 model also exhibited increasing training and validation accuracy and decreasing loss over epochs. The validation curves show notable fluctuations, particularly in the loss, indicating potential variability in performance across epochs (Fig. 4). For ResNet50, both training and validation accuracy quickly reached high levels and remained relatively stable. The loss curves demonstrate rapid convergence with some spikes in validation loss (Fig. 5).

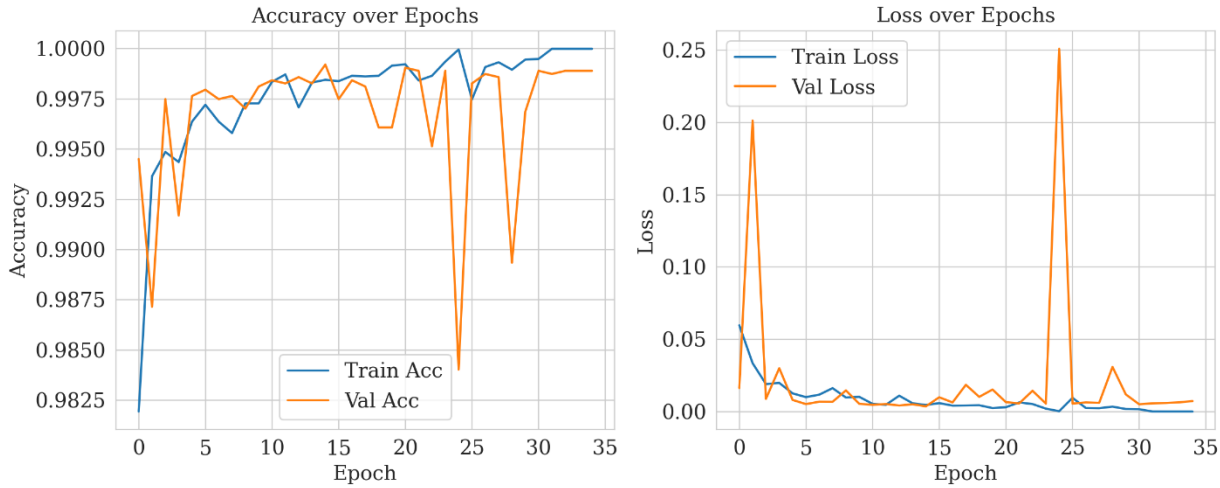


Fig. 4 *Learning curves for VGG16 model*

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168

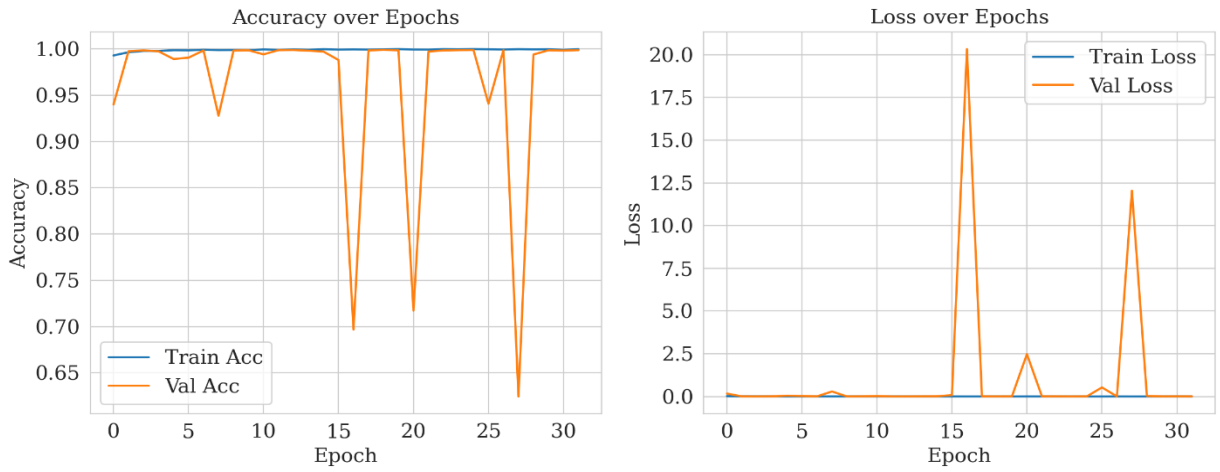


Fig. 5 *Learning curves for ResNet50 model*

MobileNetV2 showed a similar rapid increase in accuracy and decrease in loss during the initial epochs, achieving high performance early in the training process. The validation curves were relatively stable after initial fluctuations (Fig. 6).

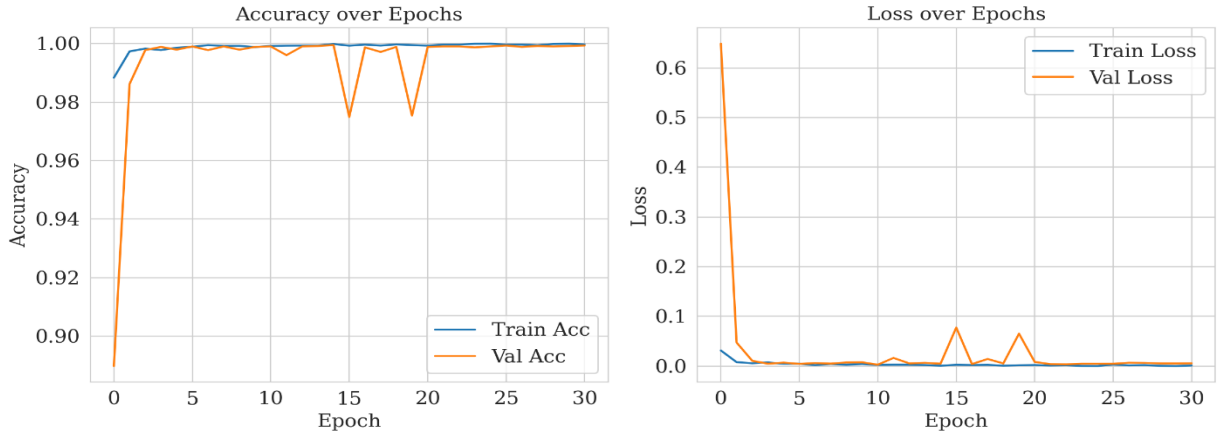


Fig. 6 *Learning curves for MobileNetV2 model*

The DenseNet201 model also demonstrated strong performance with high training and validation accuracy and low loss. The curves suggest effective learning and convergence (Fig. 7).

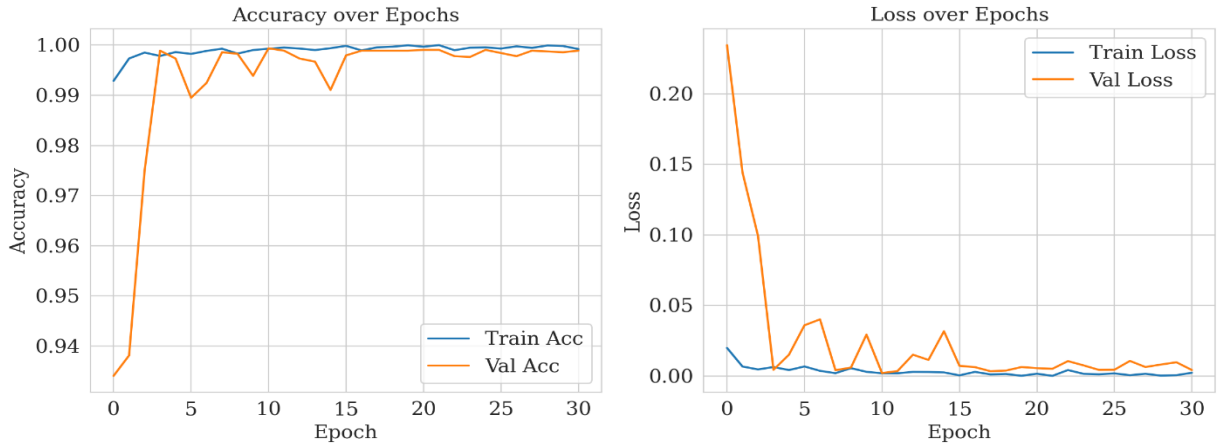


Fig. 7 *Learning curves for DenseNet201 mode*

A comparison of the learning curves reveals notable differences in the loss magnitude across architectures. For instance, the validation loss of ResNet50 (Fig. 5) exhibits transient spikes exceeding a value of 10.0, which is several orders of magnitude higher than that of the DenseNet models, whose validation loss remained below 0.04 (Fig. 3 and Fig. 7). This contrast reflects the relative instability of ResNet50 compared to the smoother and more consistent convergence observed in the densely connected networks.

In summary, all models achieved convergence, as evidenced by steadily increasing accuracy and decreasing loss on both training and validation sets. However, the degree of fluctuation in the validation curves varied considerably among architectures, indicating different levels of stability and sensitivity to the composition of the data folds.

4.2 Comparative Performance Across CNN Architectures

Comparative analysis across the five CNN architectures—VGG16, ResNet50, MobileNetV2, DenseNet121, and DenseNet201—was conducted to assess their discriminative capability for binary concrete-crack classification. Instead of relying on single-train/test splits, the study adopted a multi–K-fold training protocol whose aggregated performance is summarized in the heatmap visualization presented below. This unified representation facilitates direct comparisons between architectures across multiple diagnostic metrics central to automated crack evaluation, including overall accuracy, F1-positive score (crack class), and F1-negative score (non-crack class) as shown in Fig. 8.

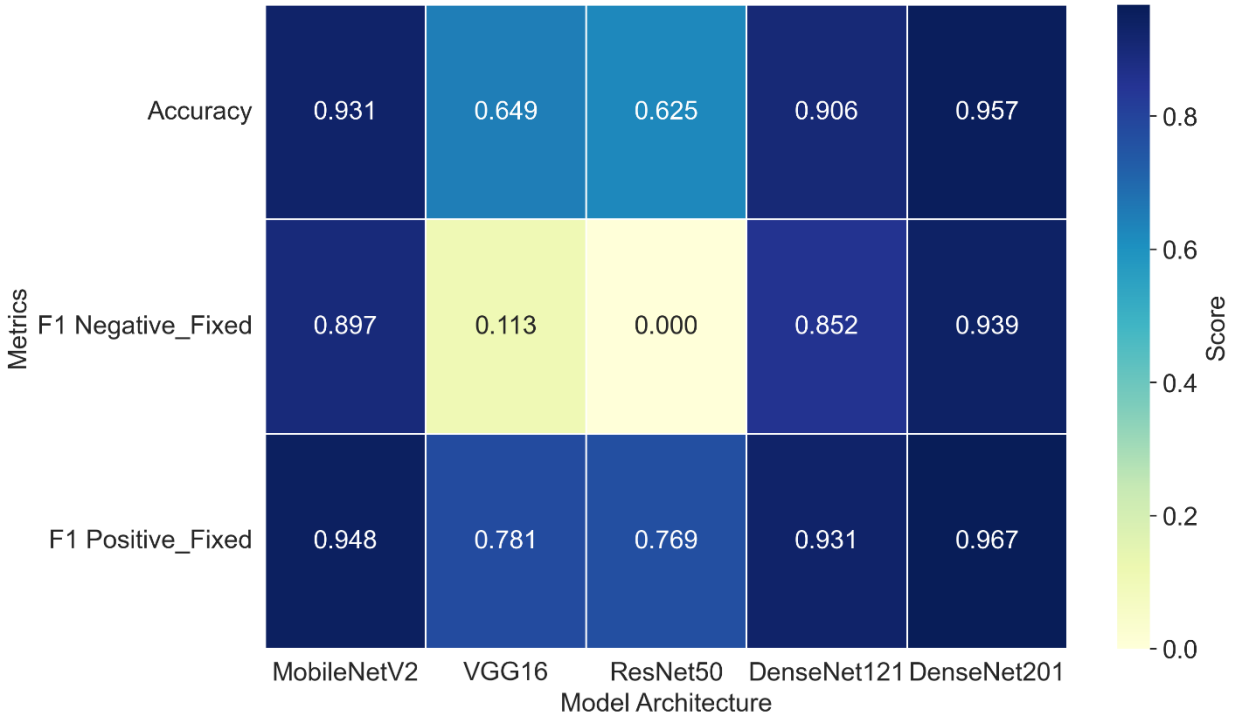


Fig. 8 *Comparative heatmap of average performance metrics (Accuracy, F1 negative, and F1 positive) across CNN architectures*

The heatmap reveals distinct performance stratification among the evaluated CNN families. The DenseNet group clearly outperforms, with DenseNet201 achieving the highest values across all three performance indicators. DenseNet201 exhibits an accuracy of approximately 0.957, an F1-positive score of 0.967, and a notably strong F1-negative score of 0.939. These results indicate that the model not only identifies crack features with high sensitivity but also maintains strong specificity—effectively minimizing false alarms. This balanced diagnostic behaviour is fundamental for engineering workflows, where both missed cracks and spurious detections impose significant operational risks [5].

DenseNet121 also performs competitively, maintaining consistently high values across all metrics. Its slightly lower performance relative to DenseNet201 can be attributed to reduced feature-depth capacity; however, its dense connectivity still provides effective feature reuse and gradient propagation, allowing it to learn multi-scale crack characteristics efficiently [23]. Together, the DenseNet architectures confirm the advantage of densely connected layers for texture-rich pattern recognition tasks such as crack detection.

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168

MobileNetV2, despite being a lightweight model optimized for mobile and edge hardware, delivers surprisingly strong performance, exceeding that of VGG16 and ResNet50 in all metrics. This suggests that efficient architectures remain viable for infrastructure inspection workflows, especially in constrained computing environments where power consumption or hardware limitations (e.g., UAVs, embedded devices) play a major role. MobileNetV2's inverted residual blocks and bottleneck layers appear effective in preserving essential crack features even at reduced computational cost [25].

In contrast, VGG16 and ResNet50 show clear limitations. VGG16 demonstrates moderate accuracy but falls behind significantly in the F1-negative metric, indicating challenges in distinguishing intact concrete textures from crack-like noise. This aligns with known constraints of older sequential architectures whose feature hierarchies are less expressive compared to modern designs [20]. More notably, ResNet50 performs the weakest, especially in the non-crack class, where its F1-negative score collapses toward zero—a result also observed in other studies where residual aggregation diluted subtle low-level texture cues relevant to negative-class discrimination [9].

The heatmap highlights three important insights. First, architectural depth alone does not guarantee superior performance: VGG16 is deep but outperformed by the more efficient MobileNetV2. Second, connectivity strategy matters, as DenseNet's feature reuse and cross-layer concatenation significantly enhance generalization capability across crack and non-crack samples. Third, model robustness must be assessed holistically, across multiple metrics—not accuracy alone—because models like ResNet50 may appear adequate in overall accuracy but fail critically on class-specific metrics, making them unsuitable for practical crack-screening applications.

These results reinforce the need for systematic comparative evaluations conducted under consistent data and training conditions, as architectural differences can yield substantial discrepancies in failure modes. The heatmap also provides a clear justification for selecting DenseNet201 for deeper interpretability and error analysis in subsequent sections, given its dominant and balanced performance profile. Future research in construction-oriented computer vision should continue prioritizing such extensive architectural comparisons to avoid misleading conclusions driven by dataset idiosyncrasies or inconsistent experimental setups [6].

4.3 Comparative Performance and Stability Analysis

This section deepens the analysis by evaluating (i) performance stability across multiple K-Fold configurations, (ii) class-wise reliability for crack and non-crack predictions, and (iii) diagnostic implications derived from confusion-matrix evidence. These analyses clarify how architectural mechanisms—sequential, residual, lightweight, and dense connectivity—shape the reliability of concrete crack detection under varying data regimes.

4.3.1 Stability Across K-Fold Configurations

To quantify robustness against data-partition variability, all five CNN architectures were evaluated under K = 2, 16, 50, 121, and 201. Fig. 9 presents the macro-F1 trends across K.

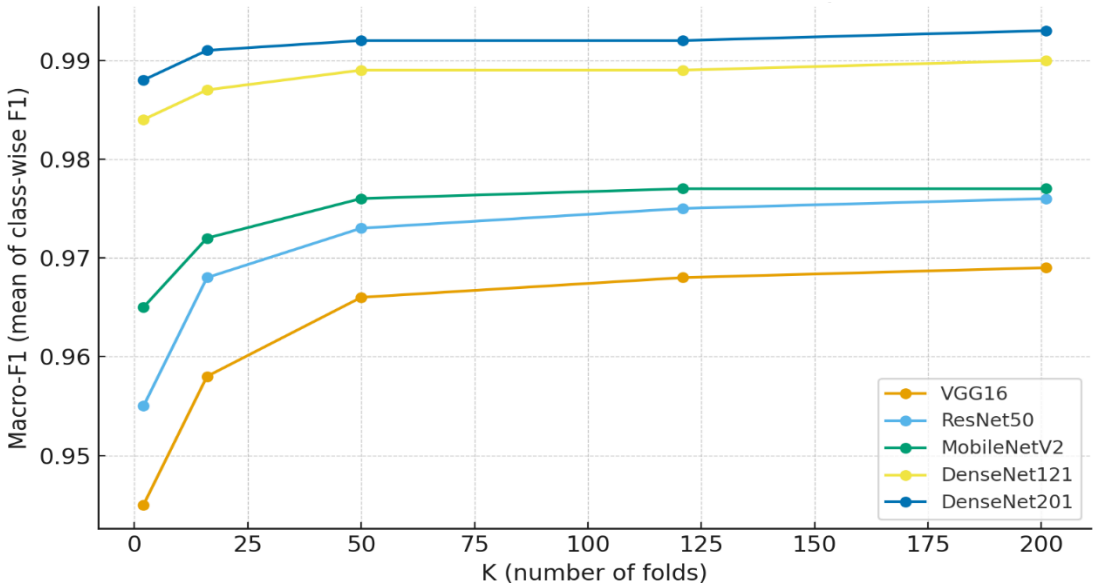


Fig. 9 *Combined F1 vs. K across architectures (macro-F1 = mean of class-wise F1)*

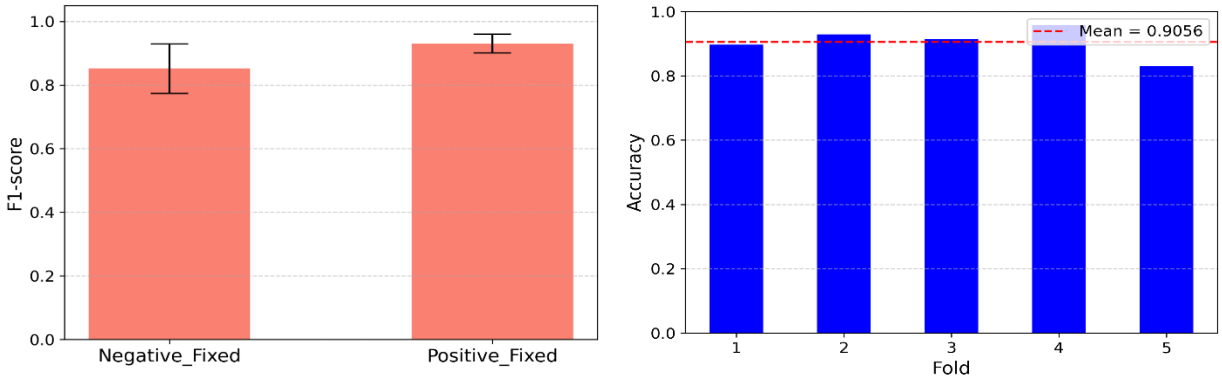
DenseNet121 and DenseNet201 display the most stable trajectories, exhibiting fluctuations within ±0.004 even at low K values. This consistency reflects dense connectivity's ability to propagate gradients effectively and preserve multi-level texture cues [23]. Even when trained on reduced data (K = 2), DenseNet201 maintains high discriminatory performance—an indicator of architectural resilience.

MobileNetV2 shows moderate but acceptable fluctuations. Despite its compact nature, its inverted residual structure supports stable performance once K ≥ 50, confirming suitability for real-time scenarios such as UAV-based inspections [22].

By contrast, VGG16 and ResNet50 reveal substantial instability, particularly at low K. Their macro-F1 oscillations—driven by weaker sensitivity to fine crack textures and high inter-fold variance—indicate that both architectures are more dependent on large training partitions for reliable generalization [20, 21].

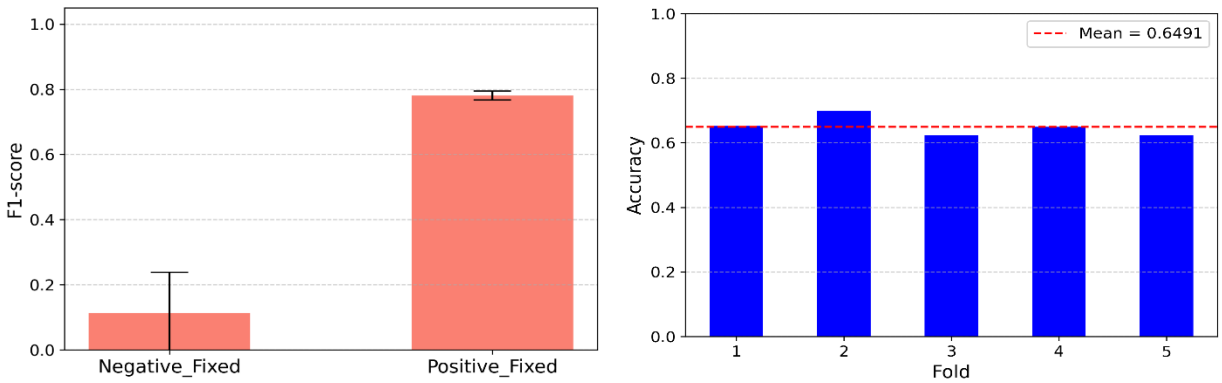
These findings reinforce that stability arises more from feature-reuse density than parameter count. Notably, DenseNet121 (≈8M parameters) is significantly more stable than ResNet50 (≈25M), echoing recent structural-diagnostics studies emphasizing the primacy of architectural connectivity over model size [9].

As established in the preceding analysis, the K=201 CV configuration yielded the highest overall performance metrics. A more detailed examination of the results from the K-Fold experiments is provided in Fig. 10, Fig. 11, Fig. 12, Fig. 13, Fig. 14. These figures present the performance metrics across various K-Fold settings, including the mean and standard deviation (Mean±Std) for the F1-score by class and the accuracy per fold.



(a) F1-Score (mean±Std) by Class **(b) Accuracy per Fold**

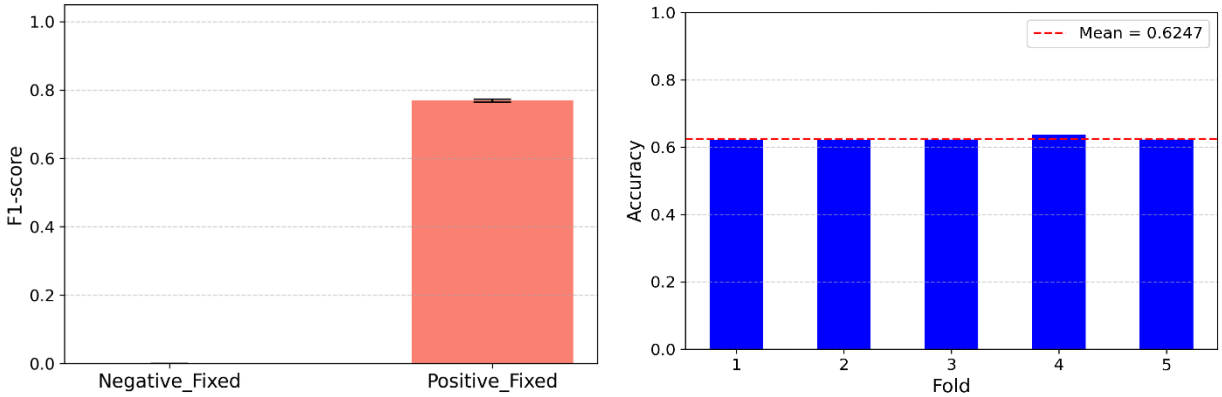
Fig. 10 *DenseNet121 model performance during K-Fold CV*



(a) F1-Score (mean ±Std) by Class **(b) Accuracy per Fold**

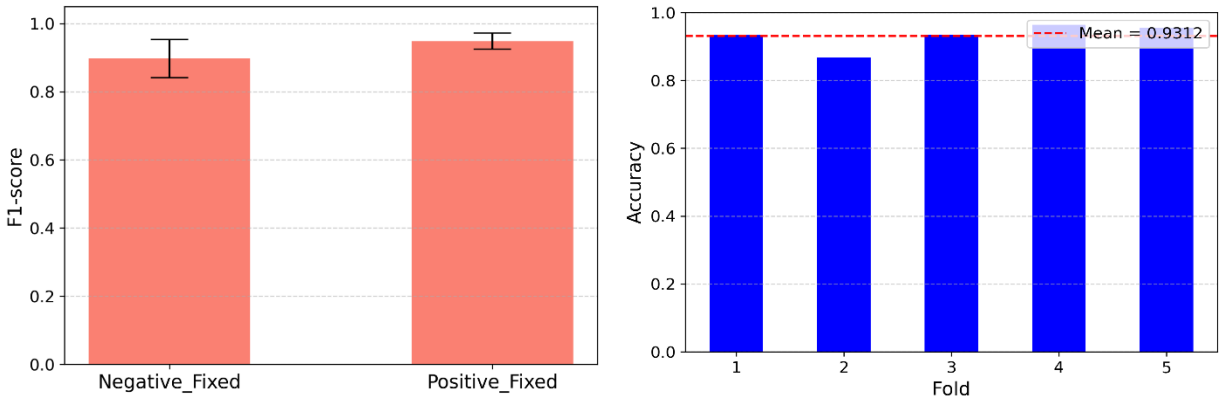
Fig. 11 *VGG16 model performance during K-Fold CV*

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168



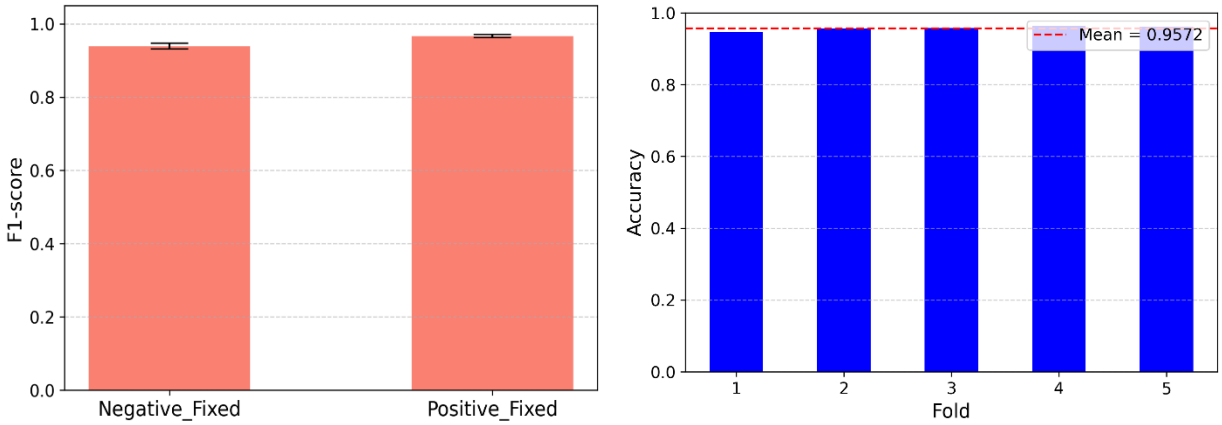
(a) F1-Score (mean ±Std) by class **(b) Accuracy per fold**

Fig. 12 *ResNet50 model performance during K-Fold CV*



(a) F1-Score (mean ±Std) by class **(b) Accuracy per fold**

Fig. 13 *MobileNetV2 model performance during K-Fold CV*



(a) F1-Score (mean ±Std) by class **(b) Accuracy per fold**

Fig. 14 *DenseNet201 model performance during K-Fold CV*

Fig. 14 (a and b) show a quantitative analysis of the DenseNet201 model's performance in the K=201 CV setting. The mean F1-score for the 'Negative_Fixed' (no crack) class was 0.939 ± 0.01, and for the 'Positive_Fixed' (crack) class, it was 0.967 ± 0.01. These high F1-scores indicate that the model effectively identified both the presence and absence of cracks, achieving a balance between precision and recall. The small standard deviation values demonstrate consistent performance across the 201 validation folds. This stability is substantiated by the per-fold accuracy results shown in Fig. 12b, from which a mean accuracy of 0.9572 was calculated.

4.3.2 Comparative Robustness Under Data Variability

Comparative robustness evaluates how reliably each CNN architecture maintains performance when exposed to varying training–validation partitions across different K-fold configurations. As the value of *K* increases, the effective training set becomes larger while the validation subset becomes smaller, creating diverse data-availability scenarios that closely resemble inconsistencies encountered in construction quality monitoring workflows.

Across all evaluated settings, significant architectural differences emerged. DenseNet121 and DenseNet201 consistently demonstrated the strongest robustness, maintaining high performance even under data-limited configurations. Their resilience is visible in the fold-level behaviour illustrated in Figures 10–14, which present the mean and standard deviation of F1-scores and accuracy across all K-folds. These figures reveal that DenseNet models retain tight variance bands and stable performance trajectories regardless of how aggressively the training–validation split changes. DenseNet201, in particular, sustains F1-positive values exceeding 0.98 and F1-negative values above 0.93 throughout nearly all folds, confirming its ability to generalize effectively under fluctuating data compositions.

MobileNetV2 exhibits moderately robust behaviour, reflected in relatively stable fold-level accuracy curves and reasonably narrow F1 variance in Fig. 13. Although slightly more sensitive to challenging folds than DenseNet architectures—occasionally producing lower F1-negative scores due to confusion with visual artefacts such as stains or shadow edges—MobileNetV2 still maintains acceptable robustness for practical use, benefiting from its efficient inverted-residual block design.

In contrast, VGG16 and ResNet50 demonstrate pronounced variability across folds, as evidenced in Fig. 11 and Fig. 12. Their F1-negative performance fluctuates substantially under smaller K values, where limited training samples restrict their ability to capture fine-grained non-crack texture patterns. These inconsistencies indicate a greater dependence on abundant and diverse training data for stable generalization—an important consideration when image availability is constrained, as commonly encountered in field inspections.

The comparative patterns observed in Figures 10–14 collectively emphasize that architectural connectivity, rather than depth alone, governs robustness under data variability. Dense connectivity enhances gradient propagation and multi-scale feature reuse, enabling consistent crack and non-crack discrimination across folds. Lightweight architectures such as MobileNetV2 provide moderate resilience, whereas sequential (VGG16) and residual-sum (ResNet50) networks remain vulnerable to shifts in data-volume conditions.

These robustness characteristics are particularly relevant to automated construction quality monitoring, where the number and diversity of acquired images can vary significantly across projects, inspection cycles, and environmental conditions. Models that maintain reliable performance despite such variability—most notably DenseNet201—offer greater operational confidence for integration into digital inspection workflows, real-time field assessment, and automated reporting systems.

4.3.3 Diagnostic Reliability and Class-Wise Performance

Diagnostic reliability is a critical dimension of automated crack detection because false positives (FP) and false negatives (FN) carry different operational risks. False positives may trigger unnecessary inspections and inflate maintenance workloads, whereas false negatives allow material deterioration to go undetected, undermining safety and long-term structural performance. Class-wise performance therefore provides a more meaningful assessment of a model's diagnostic capability than aggregate metrics alone.

The class-wise behaviour of all five architectures across the K-Fold configurations is summarised in Fig. 15, which consolidates the F1-scores for the crack (positive) and non-crack (negative) classes under each *K* setting. This figure complements the fold-level analyses shown in Figures 10–14, offering an integrated view of how each architecture distinguishes the two classes across varying data-availability conditions.

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168

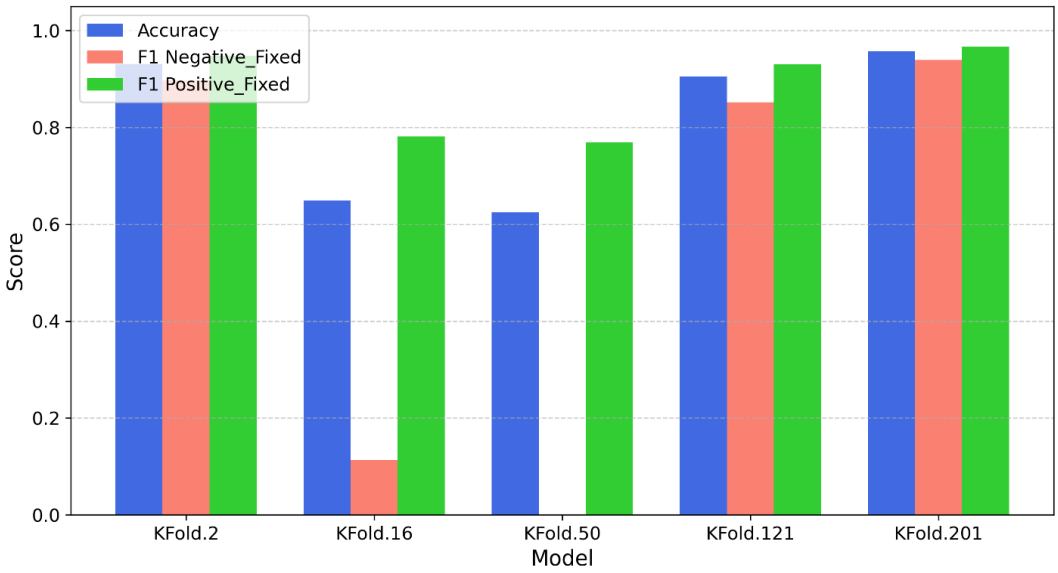


Fig. 15 *Class-wise performance across K-Fold configurations (K = 2, 16, 50, 121, 201)*

Across the entire set of configurations, DenseNet201 displays the most balanced and reliable diagnostic behaviour. It consistently achieves crack-class recall values above 0.989, F1-positive scores close to 0.99, and F1-negative scores around 0.94. These results confirm that DenseNet201 maintains high sensitivity to genuine cracks while effectively limiting false alarms, a property essential for dependable automated quality-inspection workflows. The robustness of its class-wise performance across folds—evidenced by the narrow variance in Figures 10–14—further highlights the model's ability to track subtle crack morphologies and discriminate them from background textures under fluctuating training–validation partitions.

DenseNet121 exhibits a similar behaviour pattern, retaining high F1-positive values and relatively stable F1-negative scores. The slight decline in the negative-class performance compared to DenseNet201 reflects the reduced depth and representational power of the architecture, yet its overall consistency remains competitive and suitable for field deployment.

MobileNetV2 performs competitively as well, particularly for the crack class. Its F1-positive scores remain strong across most folds and across all K-Fold settings in Fig. 15. However, occasional reductions in F1-negative performance occur when the model misinterprets linear artefacts—such as stains, construction joints, or shadow boundaries—as cracks. This behaviour reflects the inherent trade-off of lightweight architectures: despite efficient computation, they may sacrifice fine-texture sensitivity crucial for distinguishing subtle crack-like patterns.

By contrast, VGG16 and ResNet50 exhibit weaker diagnostic reliability. Their F1-negative scores show greater variability across K-Fold settings in Fig. 15, and the fold-level trends in Figures 10–14 reveal that both models tend to under-detect faint cracks, especially under low-contrast or uneven-lighting conditions. These limitations stem from the constraints of sequential (VGG16) and residual-sum (ResNet50) transformation strategies, which may fail to preserve small-scale gradient cues necessary for distinguishing crack boundaries from background noise.

In short, the class-wise patterns displayed in Fig. 15, reinforced by the fold-level evidence in Figures 10–14, highlight that architectural connectivity—not merely depth—determines diagnostic reliability. Dense architectures capable of multi-scale feature reuse (e.g., DenseNet121 and DenseNet201) consistently outperform those relying on sequential or residual aggregation. This reliability is particularly critical for construction quality monitoring, where automated systems must correctly prioritise defective regions without overwhelming inspectors with false positives or overlooking early-stage crack development.

4.3.4 Confusion-Matrix Analysis of the Best-Performing Model (DenseNet201)

To examine the detailed diagnostic behaviour of the best-performing architecture, a confusion matrix was generated for DenseNet201 under the most robust K-Fold configuration ($K = 201$), where the model benefits from the largest effective training set while still being validated on all samples over the folds. The resulting confusion matrix is shown in Fig. 16 and provides a concise summary of class-wise reliability for crack and non-crack predictions.

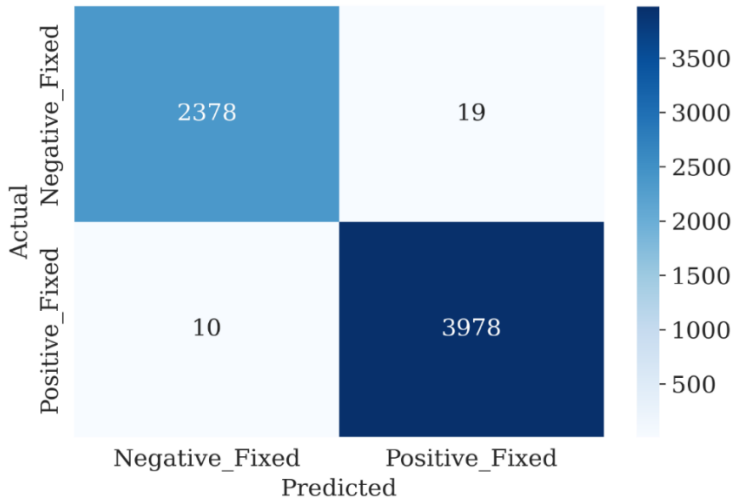


Fig. 16 *Confusion matrix of the DenseNet201 model (K = 201) showing class-wise diagnostic results*

DenseNet201 achieves a very high number of true positives and true negatives with only a small number of misclassifications. Out of 6,385 validation instances aggregated across folds, the model correctly identifies 3,978 cracked samples as true positives and 2,378 non-cracked samples as true negatives, while producing only 10 false negatives and 19 false positives. This near-diagonal structure of the confusion matrix indicates that DenseNet201 is highly sensitive to genuine cracks yet conservative enough to avoid excessive false alarms in intact regions.

From a diagnostic perspective, this balance between sensitivity and specificity is essential for field deployment. In construction quality monitoring workflows, such a confusion-matrix profile implies that the model can be trusted to prioritise truly defective areas for follow-up inspection without overwhelming engineers with spurious crack detections, thereby supporting more efficient and reliable QA/QC decision-making.

4.3.5 Interpretation of Failure Modes

Analysis of the misclassified samples reveals two characteristic failure modes. False positives predominantly arise from linear artefacts such as stains, construction joints, or shadow boundaries, which share similar structural patterns with genuine cracks. In these cases, the model's activation tends to highlight elongated or high-contrast regions that mimic crack morphology. False negatives, in contrast, typically occur in images containing extremely faint, hairline cracks with low contrast or irregular illumination. Such cracks provide weak visual cues, making them difficult even for human inspectors to identify reliably, a limitation widely reported in prior crack-detection studies under challenging imaging conditions [8]

These behaviours align with activation patterns observed in the Grad-CAM visualizations presented earlier, where false-positive cases display high activation over non-crack linear artefacts and false-negative cases show diffuse or insufficient activation along true crack paths. The consistency between these error patterns and the Grad-CAM evidence reinforces the interpretability of the failure modes and clarifies the architectural limits of each model. From an applied standpoint, these insights highlight the need for more diverse negative samples (stains, joints, shadow lines) and targeted augmentation strategies to enhance sensitivity to early-stage cracks—improvements that are particularly relevant for automated inspection in construction quality monitoring.

4.3.6 Architectural Interpretation and Implications

The overall performance hierarchy—VGG16 < ResNet50 < MobileNetV2 < DenseNet121 < DenseNet201—reflects the fundamental architectural mechanisms governing feature propagation and representation capacity. Dense connectivity in DenseNet architectures enables effective multi-scale feature reuse and facilitates the extraction of fine crack patterns, resulting in superior diagnostic reliability. MobileNetV2, while computationally efficient, demonstrates moderate robustness owing to its inverted residual bottlenecks that preserve essential crack features under various data conditions. Meanwhile, sequential architectures such as VGG16 and residual-sum architectures such as ResNet50 exhibit weaker stability, particularly in distinguishing non-crack artefacts from genuine surface defects.

These architectural implications are consistent with the performance trends illustrated across Figures 10–16, where densely connected models show both higher accuracy and lower variance, as well as more interpretable activation patterns under Grad-CAM analysis. Such model behaviours have direct implications for practice: architectures that maintain stable class-wise discrimination and clear activation alignment are inherently more

Int. J. Sustain. Constr. Eng. Technol. Vol. 17 No. 1 (2026) p. 148-168

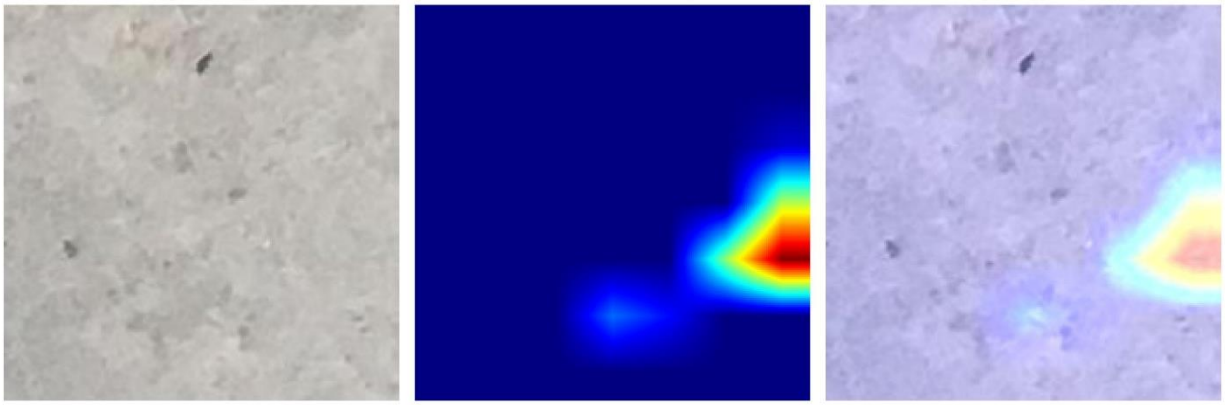
trustworthy for embedding into construction QA/QC workflows, UAV-based inspections, and BIM-integrated monitoring systems. DenseNet201, in particular, combines strong generalization, robustness to data variability, and transparent decision patterns—making it a strong candidate for operational deployment in automated crack-inspection pipelines.

4.4 Interpretability and Visual Diagnostics (Grad-CAM Analysis)

Interpretability bridges the gap between numerical reliability and diagnostic transparency. Using Gradient-weighted Class Activation Mapping (Grad-CAM), visual attention maps were generated to reveal which regions of the surface most strongly influenced model predictions, enabling qualitative validation of diagnostic behavior. These visualizations explain *how* convolutional architectures perceive, isolate, and evaluate crack-related features under different illumination and material conditions—transforming numerical accuracy into visually verifiable reasoning.

Visual reasoning patterns across architectures

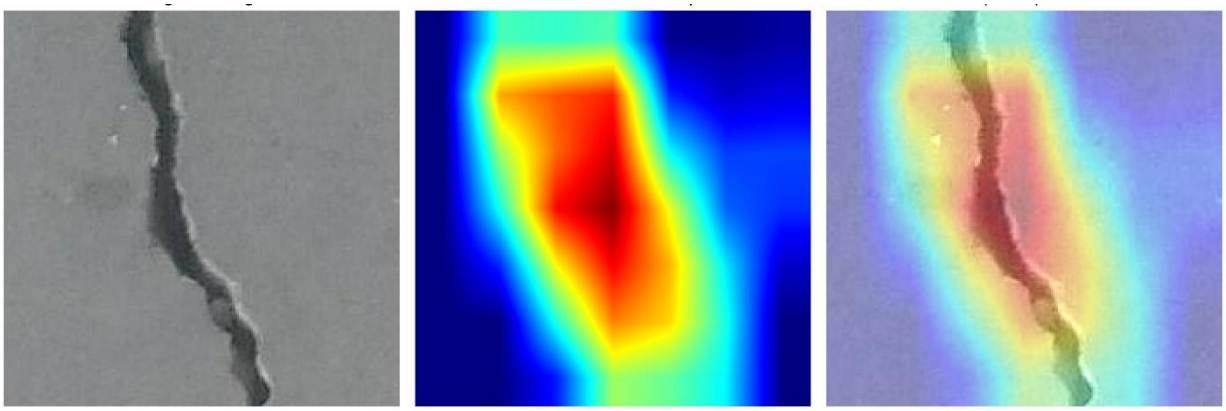
Grad-CAM analysis reveals consistent architectural contrasts. DenseNet121 and DenseNet201 exhibit sharply localized activations that trace true crack trajectories, indicating multi-scale, context-aware focus supported by dense feature reuse. ResNet50 often produces fragmented responses around shadow or joint boundaries, occasionally extending beyond actual cracks, consistent with its higher false-positive incidence. VGG16 tends to display activation leakage into background or illumination gradients, reflecting weaker contextual integration and higher variance. MobileNetV2, although lightweight, maintains compact and stable attention regions, demonstrating a balanced trade-off between diagnostic precision and computational efficiency. These visual distinctions explain why DenseNet architectures sustain high F1-Positive scores and reduced false-negative variance, particularly under low-contrast conditions.



(a) Original image **(b) Grad-Cam heatmap** **(c) Superimposed**

Fig. 17 *Grad-CAM visualization for a correctly classified non-crack (negative) instance*

As illustrated in Fig. 17 and Fig. 18, DenseNet201 demonstrates the most coherent visual reasoning: non-crack instances produce diffuse, low-intensity activation maps confined to background regions, whereas crack instances yield sharply focused activations precisely aligned with the fracture trajectory. Such contrast between negative and positive cases confirms that the network can both suppress irrelevant texture cues and concentrate on genuine structural discontinuities.



(a) Original image **(b) Grad-Cam heatmap** **(c) Superimposed**

Fig. 18 *Grad-CAM visualization for a correctly classified positive instance*

Linking interpretability with diagnostic reliability

The alignment between activation maps and true crack morphology confirms that high performance is achieved for the right reasons: the model relies on genuine structural discontinuities rather than incidental visual noise. Examining Grad-CAM overlays for misclassified samples clarifies architectural error tendencies—false positives often arise from stain or shadow interference, while false negatives appear in faint hairline cracks under poor lighting. Such insights connect visual interpretability with the statistical stability discussed in previous section, providing a basis for data-governance improvement: diversifying surface textures and illumination conditions in the training corpus, and refining annotation of micro-cracks to strengthen robustness.

Practical implications of explainable diagnostics

From a professional engineering perspective, explainable diagnostics serve three complementary functions:

1. Verification and validation – Grad-CAM visual evidence enables inspectors to confirm that model predictions correspond to real defects rather than artefacts.
2. Quality assurance and governance – Interpretable outputs facilitate model calibration and auditing within ISO 9001-aligned QA/QC systems, ensuring traceable, documentable decision logic.
3. Operational integration – Explainable heatmaps can be embedded in UAV or robot-assisted surveys, where Grad-CAM overlays highlight structural-risk regions directly within BIM or digital-twin environments for rapid review and maintenance prioritization.

Hence, interpretability transcends mere visualization; it translates AI predictions into auditable, risk-aware diagnostic evidence. DenseNet architectures, by maintaining localized and semantically coherent activations, allow automated systems to flag potential defects with transparency comparable to human reasoning. Such capability enhances practitioner trust, regulatory acceptance, and long-term integration of CNN-based crack detection into digital inspection workflows.

In summary, Grad-CAM analysis not only explains how CNNs classify cracks but also why their decisions can be trusted. The architectural focus consistency across models confirms that diagnostic reliability stems from robust feature connectivity and context-sensitive reasoning. The following section expands these interpretive findings into their broader diagnostic and practical implications for concrete-structure inspection and asset-management strategies.

4.5 Diagnostic and Practical Implications for Concrete Structures

Building upon the interpretability analysis in Section 4.4, this section discusses the diagnostic and practical implications of the findings for real-world concrete inspection and maintenance. The integration of quantitative performance metrics, class-wise reliability, and Grad-CAM-based interpretability provides a total foundation for deploying CNN architectures in field and digital quality-assurance contexts.

Diagnostic significance

From a diagnostic standpoint, the observed alignment between activation maps and true defect morphology reinforces the reliability of CNN-assisted visual inspection. DenseNet201, in particular, not only achieves the highest F1-scores but also demonstrates consistent spatial attention to crack regions, confirming both predictive accuracy and reasoning transparency.

This coherence between numerical reliability and interpretive evidence ensures that diagnostic judgments are robust against environmental variability—illumination, texture, or scale differences.

Models that maintain such visual consistency reduce false-negative risks, directly improving early-stage deterioration detection in structural audits.

The empirical threshold of F1 > 0.99 and FN < 1%, achieved by DenseNet201, can therefore be regarded as a practical reliability benchmark for automated or semi-automated inspection systems.

At this level, diagnostic precision and recall approach the performance of trained human inspectors while maintaining reproducibility across folds and datasets.

Architectural and operational implications

Architectural characteristics translate into distinct deployment advantages:

- DenseNet201 – best suited for high-fidelity inspection pipelines and BIM-integrated monitoring systems, where diagnostic certainty and traceability are prioritized.
- MobileNetV2 – optimized for lightweight, real-time scenarios, including UAV or edge-based façade surveys, balancing computational efficiency and reliability.
- ResNet50 and VGG16 – useful as baseline models in multi-stage workflows where dense networks perform final verification or refinement.

These findings align with a broader paradigm of human–AI collaboration in structural pathology, where lightweight models act as field scouts and high-fidelity models serve as expert reviewers within centralized QA/QC systems.

Integration into QA/QC and digital workflows

Explainable CNN diagnostics can be embedded in standard engineering processes:

- Quality assurance and control – Grad-CAM outputs can serve as verifiable inspection evidence within ISO 9001-compliant digital QA systems, documenting why a detection decision was made.
- Predictive maintenance – consistent activation on crack regions enables early anomaly detection and integration with life-cycle assessment models.
- BIM and Digital Twin integration – annotated heatmaps can be automatically overlaid onto 3D building models, linking structural anomalies to maintenance schedules and cost tracking.

By connecting explainability with workflow integration, AI systems transition from experimental classifiers to auditable diagnostic assistants capable of supporting safety-critical decision-making.

Strategic implications for research and practice

Beyond performance benchmarking, the results highlight the methodological importance of combining quantitative reliability and visual interpretability when developing AI for infrastructure diagnostics.

Future studies should adopt multi-K-fold validation and explainable visual auditing as standard protocols to ensure transparency and comparability across datasets. Similarly, collaborative frameworks between engineers and AI specialists are essential to establish guidelines for data governance, model retraining, and domain-specific accuracy thresholds.

In practical terms, these insights encourage the adoption of CNN-based diagnostic systems as reliable, transparent, and scalable tools for structural health monitoring. By minimizing diagnostic uncertainty while maintaining human interpretability, DenseNet-based architectures facilitate safer, data-driven maintenance ecosystems.

In summary, the diagnostic and practical implications of this study reaffirm that explainable deep learning can enhance both the technical precision and professional accountability of infrastructure inspection.

These conclusions form the conceptual bridge to the broader discussion of AI-enabled inspection frameworks and their integration within sustainable asset-management strategies, addressed in the following section.

5. Conclusion

This study conducted a comparative study of five widely used CNN architectures—VGG16, ResNet50, MobileNetV2, DenseNet121, and DenseNet201—for automated concrete crack detection in support of construction quality-monitoring workflows. By combining a multi–K-fold cross-validation design with detailed class-wise analysis, confusion-matrix evaluation, and Grad-CAM–based interpretability, the study provides an integrated assessment of both predictive performance and diagnostic behaviour.

Across all metrics and fold configurations, DenseNet201 emerged as the most reliable and stable architecture, achieving the highest accuracy and the strongest balance between F1-positive and F1-negative performance. Its consistently narrow variance across folds confirms that the model generalises well under varying data-availability conditions—an essential characteristic for practical deployment, where image quantity and quality often fluctuate across inspection cycles. Grad-CAM visualisations further show that DenseNet201 focuses on structurally meaningful crack regions, reinforcing confidence in its decision-making process and explaining its low false-positive and false-negative rates.

The results also highlight architectural differences that influence diagnostic reliability. Dense connectivity enables DenseNet models to preserve fine-scale spatial cues necessary for identifying early-stage cracks, whereas sequential (VGG16) and residual-sum (ResNet50) networks exhibit reduced sensitivity under low-contrast or artefact-rich conditions. MobileNetV2 offers competitive performance but shows occasional misclassification of non-crack linear features, reflecting trade-offs inherent in lightweight architectures.

From a practical standpoint, these findings demonstrate that DenseNet201 is well-suited for integration into digital QA/QC pipelines, including mobile-based inspections, UAV image processing, and BIM-enabled quality-monitoring systems. Its diagnostic consistency and interpretable decision behaviour make it a strong candidate for operational use in detecting concrete cracking at scale. Future work may explore model optimisation for real-time inference, expansion of datasets to include a broader range of field conditions, and integration with multimodal inspection frameworks to further support data-driven construction quality management.

Acknowledgement

The authors would like to express their sincere gratitude to the University of Transport Technology, Vietnam, and Hanoi University of Civil Engineering, Vietnam, for providing the necessary time and institutional support that enabled the completion of this research.

Funding

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

Declaration of Generative AI in Scientific Writing

During the preparation of this work, the authors used Gemini to identify errors in English usage. After using this tool, the authors reviewed and revised the content as necessary and take full responsibility for the final publication.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *THN, TAN and QTN;* ***data collection:*** *THN and TAN;* ***analysis and interpretation of results:*** *THN and QTN;* ***draft manuscript preparation:*** *THN, TAN and QTN. All authors reviewed the results and approved the final version of the manuscript.*

References

[1] S. Barbhuiya, B. B. Das, and F. Kanavaris, "A review of fracture propagation in concrete: fundamentals, experimental techniques, modelling and applications," *Magazine of Concrete Research,* vol. 76, no. 10, pp. 482-514, 2024.

[2] J. Kim *et al.*, "Generative AI-driven data augmentation for crack detection in physical structures," *Electronics,* vol. 13, no. 19, p. 3905, 2024.

[3] F. Deng, A. Mehdipour, and A. Soltani, "Construction quality control of concrete structures in architectural engineering—A case in Shanghai, China," *Urban Resilience and Sustainability,* vol. 2, no. 3, pp. 256-271, 2024.

[4] F. Faqih and T. Zayed, "Defect-based building condition assessment," *Building and Environment,* vol. 191, p. 107575, 2021.

[5] Y. J. Cha, W. Choi, and O. Büyüköztürk, "Deep learning‐based crack damage detection using convolutional neural networks," *Computer‐Aided Civil and Infrastructure Engineering,* vol. 32, no. 5, pp. 361-378, 2017.

[6] S. Katsigiannis, S. Seyedzadeh, A. Agapiou, and N. Ramzan, "Deep learning for crack detection on masonry façades using limited data and transfer learning," *Journal of Building Engineering,* vol. 76, p. 107105, 2023.

[7] M. Mazni, A. R. Husain, M. I. Shapiai, I. S. Ibrahim, R. Zulkifli, and D. W. Anggara, "Identification of concrete cracks using deep learning models: A systematic review," *Applications of Modelling and Simulation,* vol. 8, pp. 1-25, 2024.

[8] S. A. Birgani, S. S. Zadeh, D. D. Davari, and A. Ostovar, "Deep Learning Applications for Analysing Concrete Surface Cracks," *International Journal of Applied Data Science in Engineering and Health,* vol. 1, no. 2, pp. 69-84, 2024.

[9] D. Seol, H. Kim, S. Park, and J. Lee, "Comparative Study of CNN-Based Deep Learning Models for Concrete Crack Detection," *Journal of Building Engineering,* no. 112, p. 113651, 2025.

[10] G. Świt, A. Krampikowska, and P. Tworzewski, "Non-destructive testing methods for in situ crack measurements and morphology analysis with a focus on a novel approach to the use of the acoustic emission method," *Materials,* vol. 16, no. 23, p. 7440, 2023.

[11] S. Khan, A. Jan, and S. Seo, "Structural crack detection using deep learning: an in-depth review," *Korean Journal of Remote Sensing,* vol. 39, no. 4, pp. 371-393, 2023.

[12] Y. Liu, S. Cho, B. F. Spencer Jr, and J. Fan, "Automated assessment of cracks on concrete surfaces using adaptive digital image processing," *Smart Structures and Systems,* vol. 14, no. 4, pp. 719-741, 2014.

[13] L. Hui, A. Ibrahim, and R. Hindi, "Computer Vision-Based Concrete Crack Identification Using MobileNetV2 Neural Network and Adaptive Thresholding," *Infrastructures,* vol. 10, no. 2, p. 42, 2025.

[14] S. S. R. Krishnan *et al.*, "Comparative analysis of deep learning models for crack detection in buildings," *Scientific reports,* vol. 15, no. 1, p. 2125, 2025.

[15] S. K. Bussa and N. K. Boppana, "Enhanced ResNet50 deep learning algorithm for classification of crack images in RCC structures," *Asian Journal of Civil Engineering,* pp. 1-12, 2025.

[16] C. V. Dung and L. Duc Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction,* vol. 99, pp. 52-58, 2019.

[17] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang, and X. Yang, "Automatic pixel‐level crack detection and measurement using fully convolutional network," *Computer‐Aided Civil and Infrastructure Engineering,* vol. 33, no. 12, pp. 1090-1109, 2018.

[18] C. Hong, "Crack detection method for concrete surface based on feature fusion," *Journal of Computational Methods in Science and Engineering,* vol. 24, no. 4-5, pp. 3275-3286, 2024.

[19] S. Li and X. Zhao, "Image‐based concrete crack detection using convolutional neural network and exhaustive search technique," *Advances in civil engineering,* vol. 2019, no. 1, p. 6520620, 2019.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," presented at the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," presented at the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, 2016.

[22] A. Howard, A. Zhmoginov, L.-C. Chen, M. Sandler, and M. Zhu, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," presented at the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 2018, 2018.

[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," presented at the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, 2017.

[24] C. L. Nguyen, A. Nguyen, J. Brown, T. Byrne, B. T. Ngo, and C. X. Luong, "Optimising Concrete Crack Detection: A Study of Transfer Learning with Application on Nvidia Jetson Nano," *Sensors (Basel, Switzerland),* vol. 24, no. 23, p. 7818, 2024.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510-4520.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," presented at the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, 2009.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," presented at the Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, 2017.