



PERFORMANCE AND EFFICIENCY TRADE-OFFS IN IMAGE-BASED CONCRETE CRACK DETECTION: A COMPARATIVE STUDY OF TRADITIONAL MACHINE LEARNING AND DEEP LEARNING MODELS

Rouabah Slim^{*1}, Fekir Mohamed², Mahdab Salim³, Bentchikou Ibrahim⁴, NaasYoucef⁵ and AbdelkaderRabah⁶

^{1,3,5,6}Department of Electronic and telecommunication, University of Djilali Bounaama, Khemis Miliana, Ain Defla, Algeria.

^{2,4}Department of Electrical Engineering, University of Djilali Bounaama, Khemis Miliana, Ain Defla, Algeria.

¹<http://orcid.org/0009-0008-2295-9470>, ²<http://orcid.org/0000-0001-9743-9543>, ³<http://orcid.org/0009-0007-8691-977X>,

⁴<http://orcid.org/0009-0008-0303-7024>, ⁵<http://orcid.org/0009-0003-0653-728X>, ⁶<http://orcid.org/0000-0002-3738-3485>

E-mail: *s.rouabah@univ-dbkm.dz, m.fekir@univ-dbkm.dz, s.mahdab@univ-dbkm.dz, b.bentchikou@univ-dbkm.dz, y.naas@univ-dbkm.dz, r.abdelkader@univ-dbkm.dz

ARTICLE INFO

Article History

Received: February 13, 2026

Reviewed: March 13, 2026

Accepted: April 01, 2026

Published: May 29, 2026

Keywords:

Concrete Crack Detection,
Structural Health Monitoring,
Machine Learning,
Deep Learning,
Image Processing.

ABSTRACT

Structural crack detection is a key task in structural health monitoring, as cracks often indicate early-stage damage in civil infrastructure. Traditional inspection methods rely on manual visual assessment, which is time-consuming, subjective, and potentially unsafe. To overcome these limitations, automated crack detection approaches based on machine learning and deep learning have gained increasing attention. This study presents a comparative analysis of traditional machine learning and deep learning models for image-based concrete crack detection. Handcrafted features extracted using the Histogram of Oriented Gradients (HOG) descriptor were classified using Support Vector Machine, Random Forest, and K-Nearest Neighbors, while deep learning approaches included a custom Convolutional Neural Network and a MobileNetV2-based transfer learning model. Experiments conducted on a balanced dataset of 40,000 images demonstrate that deep learning models achieve superior accuracy and robustness.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

Structural safety and durability are fundamental concerns in civil engineering, particularly for infrastructure such as bridges, buildings, pavements, and tunnels. Cracks are among the most common forms of structural deterioration and often represent early signs of material degradation. If left undetected, cracks may propagate and lead to severe structural failures. Consequently, timely and reliable crack detection plays a vital role in structural health monitoring and maintenance planning [1],[2]. Traditional crack inspection methods are predominantly based on manual visual assessment performed by trained inspectors. Although effective in certain scenarios, these methods are labor-intensive, time-consuming, and prone to human subjectivity. Moreover, manual inspections can be unsafe when applied to large-scale or inaccessible structures, motivating the need for automated crack detection systems [3].

Early research in automated crack detection primarily relied on classical image processing techniques, such as edge detection, thresholding, and morphological operations. While these methods demonstrated initial success, they were often sensitive to noise, lighting conditions, and surface texture variations, limiting their robustness in practical applications [4],[5]. To overcome these limitations, researchers began incorporating handcrafted feature extraction techniques combined with machine learning classifiers. Handcrafted descriptors such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Gray-Level Co-occurrence Matrices (GLCM) have been widely used to represent crack patterns in images [6], [7]. These features are typically classified using traditional machine learning models, including Support Vector Machines (SVM), Random Forests (RF), and K-Nearest Neighbors (KNN).

Such approaches have shown improved performance compared to purely image processing-based methods; however, their effectiveness strongly depends on the quality of feature engineering and parameter selection [8], [9]. In recent years, deep learning has emerged as a powerful alternative for crack detection. Convolutional Neural Networks (CNNs) are capable of automatically learning hierarchical feature representations directly from raw image data, reducing the reliance on handcrafted features. Several studies have reported significant performance improvements using CNN-based models for crack classification and segmentation tasks [10],[11]. Furthermore, transfer learning approaches using pretrained networks such as VGG, ResNet, and MobileNet have demonstrated strong generalization ability and reduced training time, particularly when large labeled datasets are available [12], [13]. Despite the superior accuracy achieved by deep learning models, their higher computational requirements and reliance on specialized hardware remain important considerations for real-world deployment. In contrast, traditional machine learning models often offer lower computational complexity and simpler implementation, making them attractive for resource-constrained environments. Therefore, a systematic comparison between machine learning and deep learning approaches, considering both classification performance and computational efficiency, is essential for informed model selection [14], [15].

Structural crack detection has been extensively studied using image processing, machine learning, and, more recently, deep learning techniques. Early research in this domain focused primarily on traditional image processing methods, such as edge detection and thresholding, to identify crack patterns in structural images. While these approaches demonstrated initial success, they were often sensitive to noise, illumination variations, and background texture, limiting their robustness in real-world applications [5], [16]. To overcome these limitations, researchers introduced handcrafted feature extraction methods combined with classical machine learning classifiers. Texture-based features, including Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Gray-Level Co-occurrence Matrices (GLCM), were widely adopted for crack representation [4],[10]. These features were typically coupled with classifiers such as Support Vector Machines (SVM), Random Forests (RF), and K-Nearest Neighbors (KNN), achieving improved accuracy compared to purely image processing-based methods [9],[17]. Although these approaches reduced sensitivity to environmental variations, their performance heavily depended on feature selection and parameter tuning. Support Vector Machines have been particularly popular in crack detection tasks due to their strong generalization ability in high-dimensional feature spaces. Several studies reported satisfactory crack classification accuracy using SVM combined with HOG or texture features [8],[18].

Random Forest classifiers were also explored for their robustness to noise and ability to handle complex decision boundaries, especially in large datasets [7]. However, classical machine learning approaches often struggle to capture the wide variability in crack shapes, widths, and orientations encountered in real-world scenarios. The emergence of deep learning, particularly Convolutional Neural Networks (CNNs), marked a significant shift in crack detection research. CNN-based models automatically learn hierarchical features directly from raw images, eliminating the need for handcrafted feature extraction. Early CNN architectures demonstrated superior performance over traditional methods in crack classification and segmentation tasks [2],[11]. These models showed strong robustness against background clutter and illumination changes. To further enhance performance and reduce training time, transfer learning techniques have been widely adopted. Pretrained deep networks such as VGG, ResNet, and MobileNet have been fine-tuned for crack detection tasks, often achieving high accuracy even with limited training data [19],[20]. Lightweight models such as MobileNet have gained attention due to their reduced computational complexity, making them suitable for deployment in resource-constrained environments [12]. Despite the effectiveness of deep learning approaches, their higher computational cost and training requirements remain a concern for practical implementation.

Consequently, recent studies emphasize the importance of comparative evaluations between traditional machine learning and deep learning models, considering not only accuracy but also computational efficiency [3],[13]. Such comparisons provide valuable insights for selecting appropriate models based on application requirements and available resources. Although numerous studies have addressed crack detection using either classical machine learning or deep learning techniques, comprehensive comparative analyses that evaluate both paradigms under consistent experimental conditions are still limited. This work aims to address this gap by systematically comparing handcrafted feature-based machine learning models and deep learning approaches for structural crack detection, with a particular focus on classification performance and computational efficiency. Motivated by these observations, this paper presents a comprehensive comparative study of traditional machine learning and deep learning models for structural crack detection. Classical classifiers based on handcrafted HOG features are evaluated alongside end-to-end deep learning models, including a custom CNN and a MobileNetV2-based transfer learning approach. The models are assessed on a balanced concrete crack image dataset using standard performance metrics and computational cost analysis. The main contributions of this work are as follows:

- A systematic comparison of handcrafted feature-based machine learning and deep learning models for crack detection.
- An evaluation of classification performance and computational efficiency under consistent experimental conditions.
- Practical insights into the trade-offs between accuracy and resource requirements for structural crack detection systems.

The remainder of this paper is organized as follows. Section 2 describes the dataset and methodology. Section 3 presents experimental results and discussion. Finally, Section 4 concludes the paper and outlines future research directions.

II. DATASET DESCRIPTION AND METHODOLOGY

II.1 DATASET DESCRIPTION

In this study, a publicly available concrete crack image dataset was employed to evaluate the performance of machine learning and deep learning models for structural crack detection. The dataset consists of 40,000 grayscale images, evenly divided into two classes: crack and no-crack, with 20,000 images per class. The balanced nature of the dataset ensures that classification results are not biased toward a particular class and allows for a fair comparison among different models. The images, in Figure 1, represent concrete surface patches containing various crack patterns, including differences in crack width, orientation, texture, and background complexity.

Such variations make the dataset suitable for assessing the robustness of crack detection algorithms under realistic conditions. All images were resized to a uniform resolution prior to feature extraction and model training to maintain consistency across experiments. For experimental evaluation, the dataset was split into 80% for training and 20% for testing, resulting in 32,000 training images and 8,000 testing images. Stratified sampling was applied to preserve the original class distribution in both subsets. This dataset has been widely used in recent studies on crack detection and structural health monitoring, making it a reliable benchmark for comparative analysis [20].

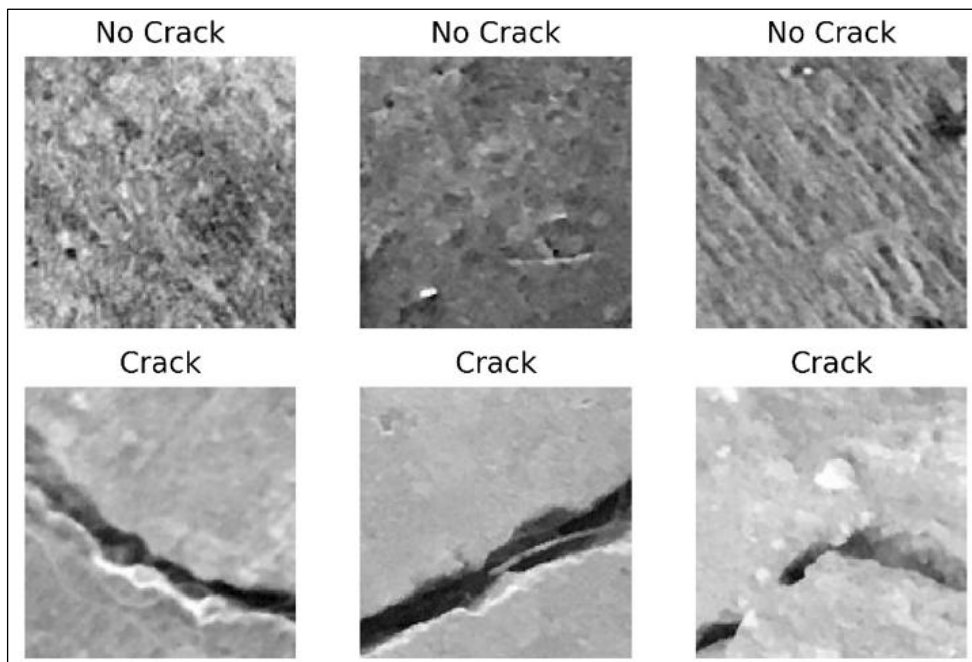


Figure 1: Sample images from the concrete crack dataset showing cracked and non-cracked concrete surfaces.

Source: Authors, (2026).

II.2 OVERAL METHODOLOGY

The overall methodology adopted in this study aims to provide a fair and systematic comparison between traditional machine learning and deep learning approaches for structural crack detection. The complete workflow of the proposed framework consists of four main stages: image preprocessing, feature extraction, model training, and performance evaluation. In the first stage, raw concrete surface images are preprocessed to ensure uniformity and consistency across all experiments. This includes resizing the images to a fixed resolution and normalizing pixel values when required. Preprocessing reduces variations caused by image scale and facilitates effective feature extraction and model learning [21]. In the second stage, two different learning paradigms are employed. For traditional machine learning models, handcrafted features are extracted using the Histogram of Oriented Gradients (HOG) descriptor. HOG is particularly effective in capturing edge and gradient information, which is well-suited for representing crack patterns in concrete surfaces [6]. The extracted feature vectors are then used as input to classical classifiers, namely Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN). In parallel, deep learning models are designed to perform end-to-end learning directly from image data. A custom Convolutional Neural Network (CNN) is developed to automatically learn hierarchical feature representations.

In addition, a transfer learning approach based on MobileNetV2 is adopted, leveraging pretrained weights to improve learning efficiency and generalization. Transfer learning has been shown to be particularly effective in structural inspection tasks, especially when dealing with complex visual patterns such as cracks [2],[13]. In the third stage, model training is conducted under consistent experimental conditions. Traditional machine learning models are trained using CPU-based computation, reflecting their standard execution environment, while deep learning models utilize GPU acceleration to handle the increased computational demand. To improve training stability and prevent overfitting in deep learning models, early stopping and model checkpointing strategies are applied based on validation performance [22]. Finally, in the evaluation stage, all models are assessed using the same training and testing splits to ensure a fair comparison. Performance is measured using standard classification metrics, including accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). In addition to predictive performance, computational efficiency is evaluated by recording training time and inference time for each model. This dual evaluation strategy allows for a comprehensive analysis of the trade-offs between classification accuracy and computational cost, which is essential for practical deployment in real-world structural health monitoring systems [3].

II.3 FEATURE EXTRACTION USING HISTOGRAM OF ORIENTED GRADIENTS (HOG)

Handcrafted feature extraction plays an important role in traditional machine learning-based crack detection, as the quality of extracted features directly influences classification performance. In this study, the Histogram of Oriented Gradients (HOG) descriptor was employed to extract discriminative features from concrete surface images prior to classification using classical machine learning models.

HOG is a widely used feature descriptor that captures local shape and texture information by analyzing the distribution of gradient orientations within an image. The fundamental idea behind HOG is that the appearance and shape of objects can be effectively characterized by the spatial distribution of intensity gradients or edge directions [6]. This property makes HOG particularly suitable for crack detection, as cracks typically manifest as elongated, edge-like structures with strong directional gradients. In the feature extraction process, each image is first divided into small spatial regions known as cells. For each cell, gradient magnitudes and orientations are computed using first-order derivatives. These gradients are then accumulated into orientation histograms, which represent the local edge distribution within each cell. To improve robustness against illumination changes and local contrast variations, the histograms are normalized over larger spatial regions referred to as blocks. The normalized histograms from all blocks are concatenated to form a high-dimensional feature vector representing the entire image.

In this work, HOG features were extracted using consistent parameter settings across all experiments to ensure fair comparison between classifiers. The resulting feature vectors capture essential structural characteristics of cracks, including orientation, continuity, and spatial distribution. These descriptors were subsequently used as input to the Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN) classifiers. Although HOG features provide a compact and interpretable representation of crack patterns, they typically result in high-dimensional feature vectors. While this enhances descriptive power, it also increases computational complexity during model training, particularly for tree-based ensemble methods. This limitation is addressed and discussed in the experimental results section, where computational efficiency is analyzed alongside classification performance. Overall, HOG offers a strong baseline feature extraction technique for structural crack detection and remains a popular choice in comparative studies involving traditional machine learning and deep learning approaches [8],[18].

II.4 DEEP LEARNING MODELS

Deep learning models were employed in this study to perform end-to-end structural crack detection directly from image data, eliminating the need for handcrafted feature extraction. Two deep learning approaches were considered: a custom Convolutional Neural Network (CNN) and a transfer learning model based on MobileNetV2. These models were selected to represent both a conventional deep learning architecture and a lightweight pretrained network suitable for practical deployment.

II.4.1 Custom Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are designed to automatically learn hierarchical feature representations from raw image inputs. A CNN typically consists of convolutional layers, pooling layers, and fully connected layers. The convolutional layers extract local spatial features, while deeper layers capture higher-level semantic information relevant to crack patterns. Given an input image I , the output feature map F_k of the k -th convolutional filter can be expressed in Equation 1.

$$F_k = \sigma(I * W_k + b_k) \quad (1)$$

where W_k denotes the convolutional kernel, b_k is the bias term, $*$ represents the convolution operation, and $\sigma(\cdot)$ is a nonlinear activation function, typically the Rectified Linear Unit (ReLU). Pooling layers are applied after convolution to reduce spatial dimensionality and improve translational invariance. Max pooling was used in this study, defined in Equation 2.

$$P(x) = \max_{i \in \Omega} F(i) \quad (2)$$

where Ω denotes the pooling region.

Following a sequence of convolutional and pooling layers, the extracted feature maps are flattened and passed to fully connected layers for classification. The final output layer uses a sigmoid activation function to perform binary classification between crack and no-crack classes as given by Equation 3:

$$\hat{y} = \sigma(w^T z + b) \quad (3)$$

where z represents the flattened feature vector.

The network was trained using the binary cross-entropy loss function given by Equation 4.

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (4)$$

To prevent overfitting and reduce unnecessary training epochs, early stopping was applied based on validation loss, and model checkpointing was used to retain the best-performing model during training.

II.4.2 Mobilenet2 With Transfer Learning

To evaluate a computationally efficient deep learning approach, MobileNetV2 was employed using a transfer learning strategy. MobileNetV2 is a lightweight CNN architecture specifically designed for resource-constrained environments, offering a favorable trade-off between accuracy and computational cost. The core building block of MobileNetV2 is the depthwise separable convolution, which decomposes standard convolution into two operations: depthwise convolution and pointwise convolution. This significantly reduces the number of parameters and computational complexity. The depthwise convolution operation can be expressed by Equation 5.

$$F_d = I * W_d \quad (5)$$

where W_d is a depthwise convolution kernel applied independently to each input channel. The pointwise convolution then combines the output channels as given by Equation 6.

$$F_p = F_d * W_p \quad (6)$$

where W_p is a 1×1 convolution kernel.

In this study, a MobileNetV2 model pretrained on the ImageNet dataset was used as a feature extractor. The original classification layers were replaced with task-specific fully connected layers to adapt the network to the crack detection problem. Only the newly added layers were trained, while the pretrained layers were kept frozen to preserve learned representations and reduce training time. Similar to the custom CNN, early stopping, learning rate reduction, and model checkpointing were employed during training to enhance generalization and stability.

II.4.3 Training Strategy and Computational Considerations

All deep learning models were trained using GPU acceleration proposed by Kaggle platform to accommodate the high computational demands of convolutional operations. Input images were normalized prior to training to improve convergence. Hyperparameters such as batch size and learning rate were selected empirically and kept consistent across experiments. While deep learning models demonstrated strong performance in crack detection, they required higher training time and computational resources compared to traditional machine learning approaches. These aspects are quantitatively analyzed and discussed in the experimental results section.

II.4.4 Summary of Deep Learning Models

The custom CNN provides a baseline deep learning architecture capable of learning crack-related features directly from images, while MobileNetV2 offers a lightweight alternative with reduced computational cost. Together, these models allow for a comprehensive evaluation of deep learning-based crack detection under different performance and efficiency constraints.

III. EXPERIMENTAL RESULTS AND DISCUSSION

III.1 PERFORMANCE EVALUATION

The experimental results obtained from the comparative evaluation of machine learning and deep learning models for structural crack detection are summarized in Table 1. All models were evaluated using the same training and testing splits to ensure a fair comparison. Performance was assessed using accuracy, precision, recall, F1-score, specificity, sensitivity, ROC-AUC, and additional reliability metrics including Matthews Correlation Coefficient (MCC) and Cohen's Kappa. Among the traditional machine learning approaches, the Random Forest classifier achieved the highest accuracy of 96.01%, outperforming the linear SVM (94.86%) and KNN (71.64%). The superior performance of Random Forest can be attributed to its ensemble nature, which allows it to model non-linear decision boundaries and reduce overfitting through aggregation. The linear SVM also demonstrated strong performance, indicating that HOG features provide good linear separability for crack detection.

In contrast, the KNN classifier exhibited significantly lower accuracy and F1-score despite achieving extremely high precision (99.94%). This behavior indicates that KNN was highly conservative in predicting the crack class, resulting in poor recall (43.30%) and sensitivity. Such imbalance suggests that KNN is less suitable for large-scale crack detection tasks where comprehensive detection is required. Deep learning models substantially outperformed traditional machine learning approaches. The custom CNN achieved an accuracy of 99.45% with an F1-score of 99.45%, while MobileNetV2 delivered the highest overall performance, achieving 99.85% accuracy and an F1-score of 99.85%. Both deep learning models exhibited near-perfect ROC-AUC values, indicating excellent discriminative capability. To further analyze deep learning behavior during training, training and validation accuracy and loss curves are presented in Figures 2.

Table 1: Performance comparison of machine learning and deep learning models for structural crack detection.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Specificity (%)	Sensitivity (%)	ROC-AUC (%)	LogLoss	MCC (%)	Kappa (%)	Training Time	Inference Time
SVM	94.86	96.05	93.58	94.8	96.15	93.58	98.43	15.85	89.75	89.72	283.19	50.86
Random Forest	96.01	97.28	94.67	95.96	97.35	94.67	99.34	17.64	92.06	92.03	4711.11	1.06
KNN	71.64	99.94	43.3	60.42	99.98	43.3	88.61	4.37	52.53	43.28	20.57	158.92
CNN	99.45	99.55	99.35	99.45	99.55	99.35	99.97	1.68	98.9	98.9	101.73	2.52
MobileNetV2	99.85	99.75	99.95	99.85	99.75	99.95	100	0.54	99.7	99.7	714.38	18.71

Source: Authors, (2026).

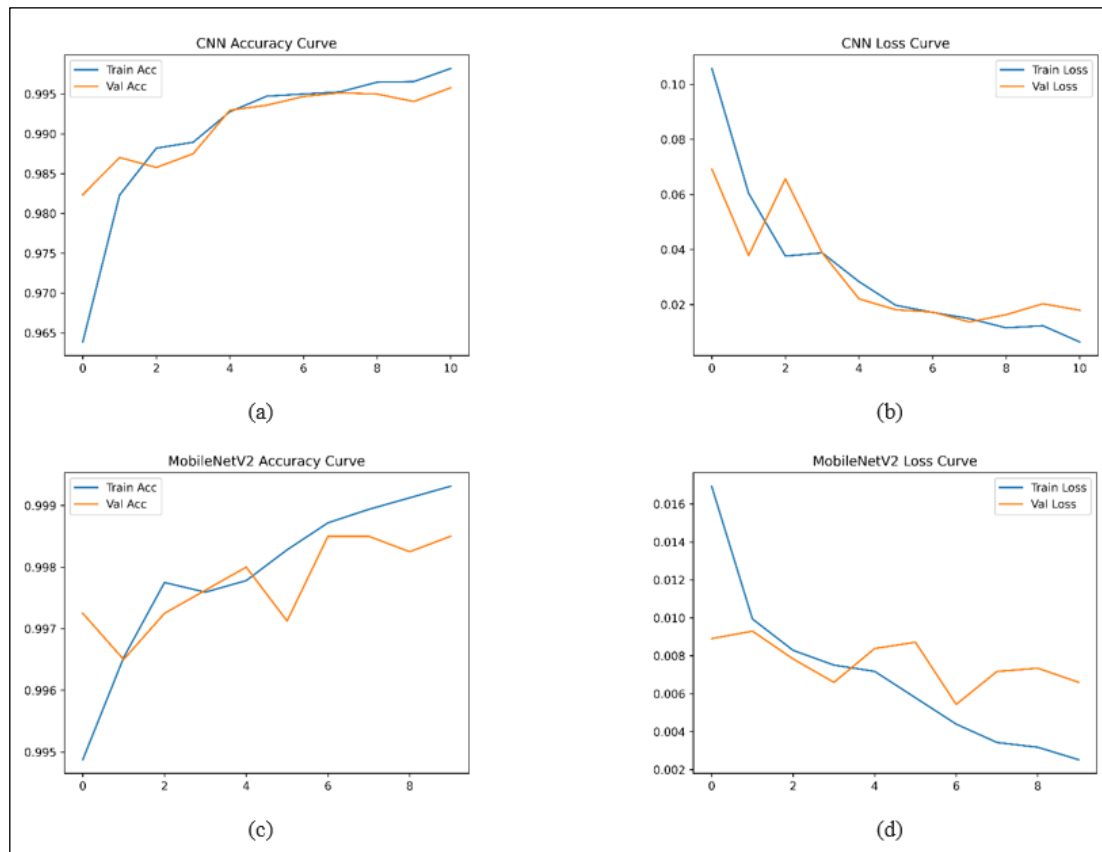


Figure 2: Training and validation accuracy and loss curves of CNN and MobileNetV2.

Source: Authors, (2026).

The accuracy curves show rapid convergence for both models, with close alignment between training and validation performance. This alignment indicates strong generalization and limited overfitting. The loss curves decrease smoothly across epochs and stabilize at low values, confirming effective optimization. MobileNetV2 reaches stable accuracy and loss earlier than the custom CNN and maintains lower validation loss, which supports the superior performance metrics reported in Table 1.

III.2 ANALYSIS OF CLASSIFICATION METRICS

Precision and recall provide important insight into model behavior beyond accuracy alone. The stable training behavior observed in Figure 2 aligns with the high classification metrics reported in Table 1. The deep learning models achieved a balanced trade-off between precision and recall, with both CNN and MobileNetV2 exceeding 99% in all major metrics. This indicates that these models are capable of reliably detecting cracks while minimizing false positives and false negatives. The Random Forest classifier showed strong precision (97.28%) and recall (94.68%), confirming its robustness among classical approaches.

The SVM classifier also performed consistently, though with slightly reduced recall compared to Random Forest. In contrast, KNN's high precision but low recall highlights its limitation in capturing the full variability of crack patterns in the dataset. The MCC and Cohen's Kappa values further support these observations. Deep learning models achieved near-perfect agreement scores, while Random Forest and SVM demonstrated strong but comparatively lower correlation values. KNN recorded the lowest agreement metrics, reinforcing its weaker classification reliability.

III.3 COMPUTATIONAL EFFICIENCY ANALYSIS

In addition to classification performance, computational efficiency was evaluated by measuring training and inference time for each model. The results reveal a clear trade-off between accuracy and computational cost. Among the classical machine learning models, KNN required negligible training time but suffered from extremely high inference time (approximately 159 seconds) due to its instance-based nature. This makes KNN impractical for real-time crack detection applications. The linear SVM demonstrated moderate training time (approximately 283 seconds) and very fast inference, making it suitable for deployment scenarios where rapid predictions are required.

The Random Forest model, trained on the full dataset, required the longest training time (approximately 78 minutes). This confirms the high computational cost associated with tree-based ensemble methods when applied to high-dimensional handcrafted features. However, its inference time remained low, indicating suitability for offline training and online prediction scenarios. Deep learning models exhibited a different computational profile. The custom CNN achieved fast training (approximately 102 seconds) due to early stopping and efficient architecture design. MobileNetV2, while requiring longer training time (approximately 714 seconds), provided the best balance between performance and efficiency, benefiting from transfer learning and a lightweight architecture. Although deep learning models require GPU acceleration, their inference times remained reasonable, supporting their potential for real-time deployment.

III.4 COMPARATIVE DISCUSSION

The experimental results clearly demonstrate that deep learning approaches outperform traditional machine learning models in terms of classification accuracy and robustness for structural crack detection. The ability of CNN-based models to automatically learn hierarchical feature representations allows them to capture complex crack patterns that handcrafted features may fail to represent adequately. However, traditional machine learning models, particularly SVM and Random Forest, still offer competitive performance with lower hardware requirements and simpler implementation. These models may be preferable in environments where GPU resources are unavailable or limited. Overall, MobileNetV2 emerged as the most effective model, achieving the highest accuracy and balanced performance while maintaining reasonable computational efficiency. This makes it a strong candidate for practical structural health monitoring systems, especially in scenarios requiring high accuracy and efficient deployment.

III.5 KEYS OBSERVATIONS

The following observations can be drawn from the experimental study:

- Deep learning models significantly outperform traditional machine learning models in crack detection accuracy.
- Random Forest provides the best performance among classical machine learning approaches but at high computational cost.
- KNN is not suitable for large-scale crack detection due to poor recall and slow inference.
- MobileNetV2 offers the best trade-off between accuracy and efficiency, making it suitable for real-world applications.

IV. CONCLUSION AND FUTURE WORKS

IV.1 CONCLUSION

This study presented a comprehensive comparative analysis of traditional machine learning and deep learning models for structural crack detection using image processing techniques. Handcrafted feature-based classifiers, including Support Vector Machine (SVM), Random Forest (RF), and K-Nearest Neighbors (KNN), were evaluated alongside deep learning approaches based on a custom Convolutional Neural Network (CNN) and a transfer learning model using MobileNetV2. Experimental results demonstrated that deep learning models significantly outperform traditional machine learning approaches in terms of classification accuracy and robustness. Among all evaluated models, MobileNetV2 achieved the highest performance, attaining near-perfect accuracy and balanced precision–recall characteristics, while maintaining reasonable computational efficiency through its lightweight architecture.

The custom CNN also delivered strong results, confirming the effectiveness of end-to-end learning for crack detection tasks. Among the classical machine learning models, Random Forest achieved the best performance but required substantially higher training time due to the high dimensionality of handcrafted HOG features. The linear SVM provided competitive accuracy with lower computational cost, while KNN showed limitations in scalability and recall performance, making it less suitable for large-scale crack detection applications. Overall, the findings highlight the trade-off between accuracy and computational efficiency across different modeling approaches. While deep learning models are preferable when high accuracy is required and GPU resources are available, traditional machine learning methods remain viable alternatives in resource-constrained environments.

IV.2 FUTURE WORKS

Although the proposed study provides valuable insights into crack detection performance across multiple modeling paradigms, several directions for future research can be explored. First, extending the analysis to include crack segmentation rather than binary classification would provide more detailed structural damage information. Second, evaluating model performance under varying environmental conditions, such as illumination changes and surface noise, would further assess robustness in real-world scenarios. Future work may also investigate hybrid approaches that combine handcrafted features with deep learning representations to balance interpretability and performance. Additionally, deploying lightweight deep learning models on edge devices or embedded systems could facilitate real-time structural health monitoring in practical applications. Finally, testing the proposed framework on additional datasets and real inspection data would enhance generalization and support broader adoption.

V. AUTHOR'S CONTRIBUTION

Conceptualization: Rouabah Slim, Fekir Mohamed, Mahdab Salim, Bentchikou Ibrahim, Naas Youcef and Abdelkader Rabah.

Methodology: Rouabah Slim, Fekir Mohamed and Abdelkader Rabah.

Investigation: Rouabah Slim, Fekir Mohamed, Mahdab Salim and Naas Youcef.

Discussion of results: Rouabah Slim, Fekir Mohamed, Mahdab Salim, Bentchikou Ibrahim, Naas Youcef and Abdelkader Rabah.

Writing – Original Draft: Rouabah Slim.

Writing – Review and Editing: Rouabah Slim, Fekir Mohamed.

Resources: Rouabah Slim.

Supervision: Fekir Mohamed, Mahdab Salim.

Approval of the final text: Rouabah Slim, Fekir Mohamed, Mahdab Salim, Bentchikou Ibrahim, Naas Youcef and Abdelkader Rabah.

VI. ACKNOWLEDGMENTS

We would like to extend our sincere gratitude to the University of Djilali Bounaama, Khemis Miliana for providing the necessary materials, laboratory facilities, and technical support that were essential for the successful completion of this project. Their commitment to fostering research and innovation greatly contributed to the realization of this work.

VII. REFERENCES

- [1] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A Review on Computer Vision-Based Defect Detection and Condition Assessment of Concrete and Asphalt Civil Infrastructure", *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, Apr. 2015. <https://doi.org/10.1016/j.aei.2015.01.008>
- [2] Y. 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, May 2017. <https://doi.org/10.1111/mice.12263>
- [3] C. Yeum and S. Dyke, "Vision-Based Automated Crack Detection for Bridge Inspection", *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 10, pp. 759–770, Oct. 2015. <https://doi.org/10.1111/mice.12141>
- [4] H. Oliveira and P. L. Correia, "Automatic Road Crack Detection and Characterization", *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 155–168, Mar. 2013. <https://doi.org/10.1109/TITS.2012.2208630>
- [5] Y. Fujita and Y. Hamamoto, "A Robust Automatic Crack Detection Method from Noisy Concrete Surfaces", *Machine Vision and Applications*, vol. 22, no. 2, pp. 245–254, Feb. 2011. <https://doi.org/10.1007/s00138-009-0244-5>
- [6] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005, pp. 886–893. <https://doi.org/10.1109/CVPR.2005.177>
- [7] S. Dorafshan, R. Thomas, and M. Maguire, "Comparison of Deep Convolutional Neural Networks and Edge Detectors for Image-Based Crack Detection in Concrete", *Construction and Building Materials*, vol. 186, pp. 1031–1045, Oct. 2018. <https://doi.org/10.1016/j.conbuildmat.2018.08.022>
- [8] J. Mukkamala, P. N. Srinivasu, A. Srinivas, A. Nithya, and S. Kandulapati, "Crack Detection on Concrete Images Using Classification Techniques in Machine Learning", *Journal of Critical Reviews*, vol. 7, no. 19, pp. 1236–1241, 2020.
- [9] M. Salman, S. Mathavan, K. Kamal, and M. Rahman, "Pavement Crack Detection Using the Gabor Filter", in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, The Hague, Netherlands, 2013, pp. 2039–2044. <https://doi.org/10.1109/ITSC.2013.6728529>
- [10] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang, "DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection", *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1498–1512, Mar. 2019. <https://doi.org/10.1109/TIP.2018.2878966>
- [11] N. Nguyen, H. Le, S. Perry, and T. Nguyen, "Pavement Crack Detection Using Convolutional Neural Network", in *Proceedings of the International Symposium on Information and Communication Technology*, Danang, Vietnam, 2018, pp. 251–256. <https://doi.org/10.1145/3287921.3287949>
- [12] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications", **arXiv preprint**, arXiv:1704.04861, Apr. 2017.
- [13] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep Convolutional Neural Networks with Transfer Learning for Computer Vision-Based Data-Driven Pavement Distress Detection", *Construction and Building Materials*, vol. 157, pp. 322–330, Dec. 2017. <https://doi.org/10.1016/j.conbuildmat.2017.09.110>
- [14] S. Verma, B. Mishra, A. Bajaj, and A. Kumar, "Comparative Analysis of Deep Learning Models for Efficient Crack Detection in Concrete Surfaces", *International Journal of Hybrid Intelligent Systems*, vol. Preprint, 2025. <https://doi.org/10.1177/14485869251362458>
- [15] N. Sholevar, A. Golroo, and S. Roghani Esfahani, "Machine Learning Techniques for Pavement Condition Evaluation", *Automation in Construction*, vol. 136, p. 104190, Apr. 2022. <https://doi.org/10.1016/j.autcon.2022.104190>
- [16] P. Prasanna et al., "Automated Crack Detection on Concrete Bridges", *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 591–599, Apr. 2016. <https://doi.org/10.1109/TASE.2014.2354314>
- [17] Z. Fan, Y. Wu, J. Lu, and W. Li, "Automatic Pavement Crack Detection Based on Structured Prediction with the Convolutional Neural Network", *arXiv preprint*, arXiv:1802.02208, Feb. 2018.
- [18] H. Li, D. Song, Y. Liu, and B. Li, "Automatic Pavement Crack Detection by Multi-Scale Image Fusion", *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2025–2036, Jun. 2019. <https://doi.org/10.1109/TITS.2018.2856928>
- [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, pp. 1–12, 2019. <https://doi.org/10.1155/2019/6520620>
- [20] Ç. F. Özgenel, *Concrete Crack Images for Classification (Version 2)*, Mendeley Data, 2019. <https://doi.org/10.17632/5y9wdsg2zt.2>
- [21] R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed., Cham, Switzerland: Springer, 2022. <https://doi.org/10.1007/978-3-030-34372-9>
- [22] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures", in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., Berlin, Germany: Springer, 2012, pp. 437–478. https://doi.org/10.1007/978-3-642-35289-8_26