VOL. 24, NO. 2, 2025, 201-210
elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Evaluating of KNN, Random Forest, and ResNet18 for Identifying Cracks in Historical Buildings

Taha Rashid[1, 2*], **Musa Mohd Mokji**[1] and **Mohammed Rasheed**[3, 4]

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.
[2]College of Arts, Al-Iraqia University, Baghdad, Iraq.
[3]Applied Sciences Department, University of Technology- Iraq, Baghdad, Iraq.
[4]MOLTECH Anjou, Universite d'Angers/UMR CNRS 6200, 2, Bd Lavoisier, 49045 Angers, France.

*Corresponding author: taha1988@graduate.utm.my

Abstract: A key element of structural damage monitoring and restoration efforts in old structures is crack detection. Conventional manual inspection techniques lack scalability, are labor intensive, and are susceptible to human error. This work investigated the performance of K-Nearest Neighbors (KNN), Random Forest, and ResNet18 models in autonomously finding cracks from the Historical-Crack18-19 dataset. The classes were balanced by increasing the dataset, which consists of, 3139 non-cracked images and 757 cracked images before training. Classification reports, confusion matrices, and predictions based on samples helped assess the models. The tests produced findings demonstrating that ResNet18 far outperformed Random Forest and KNN. Comparatively, KNN is 82% and Random Forest is 88%; the accuracy of ResNet18 is 99%. ResNet18, which is based on deep learning, had the best accuracy, Recall, and F1-Score metrics. It also had a way to tell the difference between surfaces that are cracked and those that are not. These findings show image convolutional neural networks (CNNs) are better at finding cracks, which means they could be used to keep an eye on old buildings in the real world. We will use sophisticated deep learning architectures and domain adaptation methods in future work to increase the model's resilience on many datasets even further.

Keywords: crack detection, historical buildings, deep learning, machine learning, structural health monitoring

Article History: received 15 February 2025; accepted 3 May 2025; published 31 August 2025

1. INTRODUCTION

Heritage buildings continue to be a fragile and fundamental piece of the world's cultural heritage because of their relevance and significant architectural and historical significance [1]. They are fragile, intricate, and have extremely artistic, historical, and cultural value. They, however, suffer from a variety of damages, including the presence of cracks located on their walls, which is an inherent problem. Cracks in a building's construction significantly affect properties such as durability, energy, and the structural soundness of a masonry-based building, irrespective of the geographical location. Crack-fault classification is the critical phase in building fault research [2].

A non-destructive crack detection system labels concrete surfaces and compares them with traditional crack detection methods [3]. This can help save and maintain buildings by finding early signs of damage related to building defects and performing preventative maintenance. Appropriate crack recognition in buildings and infrastructure decreases the possibility of premature failure and reduces both remediation costs and time, so mechanical and civil engineers are called to act firmly at the degradation stage of the structure. But because manual inspections take a long time, tunnel construction and renewal are happening faster and faster, and there is a lot of data that needs to be handled, positioned, and split quickly, this system finds possible threats to the building right away [4].

This section will provide an overview of the existing techniques for defect detection, with a primary focus on crack classification. Masses of walls, columns, and beams may be covered due to various reasons, such as substandard quality of materials, human-made errors, environmental factors like earthquakes, and natural disasters. In contrast, historical buildings are material proof of low to medium-rise, solidly built objects. Most historical buildings placed their stone materials on a foundation and kept them away from free moisture contact. What kind of building foundation a building has can affect how likely it is that cracks will form on the walls, columns, slab beams, and other parts of the building. It can also cause jams and endoses to form in the lower stories [5].

Today, scientists are studying the crack formation mechanism in external monumental walls. Historical research has uncovered a variety of reasons for crack formation. Cracking and disastrous consequences occur during the crack formation [6]. It is very important to make sure that the walls are protected with the right structural system until there is a critical concentration and development of failure mechanisms that can cause cracks to form on building walls and facades that are subjected to high horizontal and vertical loads during their service life. It is predicted that the crack formation affecting the facades of historical building walls is composed of different behaviors depending on the stone materials and environmental effects used in construction [7]. However, there have been no innovative technological and material solutions to resolve this problem.

Developers have used several methodologies to solve the crack formation problem in historical building facades; these methods identified a source for the crack's formation and provided important sensor technologies to prevent the cracks from expanding. However, the studies are not sufficient due to the limited number of material and method alternatives for crack solutions. In other words, extensive scientific studies have been determined and selected for use. The main type of crack repair, in which important return materials are used along with existing materials, is the perforation technique. Filling and framing works are sometimes used to help. Fill the voids surrounding the crack before treating it. Optimal solution techniques should be developed, taking into account the structures of the building, the type of damage, and the architectural value of the affected stone [8].

Many machine learning and deep learning models solve image classification, object detection, and image segmentation problems. The simplest and most popularly used machine learning model is k-nearest neighbours (KNN) because of its simplicity and high accuracy in small data classification tasks [9]. Decision tree-based learning algorithms like random forest (RF) are widely used for classification tasks in machine learning [10]. Deep convolutional neural networks (DCNN) with more layers are effectively used for classification, object detection, and image segmentation problems. Similarly, the capability of residual networks (ResNet) to learn deep features also makes them useful for image classification [11]. This research aims to evaluate and compare the performance of KNN, RF, and ResNet18 for crack classification in historical buildings. The Historical-Crack18-19 dataset was utilized, which contains 757 crack images and 3139 non-crack images. To address the class imbalance, data augmentation techniques were applied to increase the number of cracked images to match the non-cracked images. Then the dataset was split into training, validation, and testing sets using a 70%-15%-15% ratio. Each model is taught and judged on how well it can classify things using Precision, Recall, F1-Score, and confusion matrix analysis.

This paper was structured as follows: Section 2 reviews related work in crack detection using machine learning and deep learning. Section 3 describes the dataset and methodology, including preprocessing, augmentation, and model training. Section 4 presents the experimental results and performance analysis. Section 5 discusses the findings, limitations, and potential improvements, and finally, concludes the study and outlines directions for future research.

2. RELATED WORK

Automated crack detection has been widely explored using both traditional machine learning and deep learning techniques. Early studies relied on handcrafted feature extraction and classical machine learning models for crack classification. For instance, Navpreet et al. [12] make several improvements in automated concrete crack detection. First, it uses self-supervised learning to circumvent the disadvantages of supervised models, which need big, hard-to-get labeled datasets. The authors enhance detection using self-supervised learning feature extraction and a crack detection classifier. The article also compares the self-supervised DinoV2 model against five supervised models and finds good fracture detection accuracy and robustness. The research demonstrated that self-supervised generalization is better in complex backdrop situations, a common real-world challenge. Self-supervised training may employ unlabeled images in real-world situations when labeled data is costly and time-consuming. The study automates crack diagnosis for structural integrity maintenance, which ensures bridge and building safety and durability.

Mishra on the other hand [13], used machine learning with a large cohort to study the impact of early damage identification on the longevity of ancient structures by monitoring their structural health (SHM). Evaluation of ancient structures is challenging because to their unique construction processes and materials, resulting in SHM issues. The authors recommend using fuzzy logic and ML to evaluate the structural health of historic buildings, minimizing damage assessment uncertainty. The research explores different methods for forecasting building structural integrity, including support vector regression and artificial neural networks (ANN). The report recommends using advanced technologies to maintain cultural heritage sites.

A technique that integrates deep learning with machine learning was used by Thohari et al. [14]. Classification makes use of machine learning, whereas feature extraction makes use of deep learning. This study uses MobileNetV2 for deep learning and XGBoost, Random Forest, K-NN, Naive Bayes, SVM, and SVM for machine learning classifiers. As shown by the test results, XGBoost algorithms are capable of producing 99% accuracy, sensitivity, and specificity when the 80:20 dataset is divided. The deployment of Raspberry Pi allows for the execution of tests in a real-world setting. Results from tests demonstrate that, in a well-lit environment, the prototype can identify surface cracks in the structure from a distance of 10 meters. On average, 42 frames per second is the real-time pace at which the crack detection procedure is executed.

Fang et al. [15] used deep learning techniques with image processing approaches for ancient architecture's aged and diversified materials. The research meticulously collects data from many antique buildings and uses sophisticated preprocessing approaches to tackle aged surfaces' particular problems. The process of creating a bespoke machine learning model that fits historical building is discussed. Experimental findings show the model's improved structural fracture detection over existing approaches. This study helps preserve ancient structures and shows how machine learning can be used in architectural restoration. The discoveries might change how we preserve and restore our architectural legacy, guaranteeing its durability and integrity.

A comprehensive picture dataset, BD3: Building Defects Detection Dataset, was given by Kottari et al. [16] to benchmark computer vision algorithms to improve automated building inspection system resilience and generalizability. BD3 has 3,965 high-quality, personally gathered, annotated photos. BD3 contains photographs of algae, large crack, small crack, peeling, spalling, and stain, as well as typical building circumstances, unlike other datasets that concentrate on crack and non-crack images. The authors benchmarked the BD3 using five cutting-edge computer vision algorithms to categorize defect and normal photos. Vision Transformer (ViT) had the highest F1-Scores of 0.9342 and 0.9879 on the original and supplemented datasets, respectively. Benchmarking defect detection techniques is possible using the public BD3 dataset and repeatable codebase.

The methodology employed in this study aligns with previous research on crack detection using both traditional machine learning and deep learning approaches. Similar to studies such as Tsalera et al. [17], Kibriya [18] and Kumar et al. [19] Lakshmanaprabu et al. [20], our work utilizes KNN and RF with raw pixel intensity features for classification, highlighting the challenges associated with handcrafted feature extraction. Additionally, our use of data augmentation and dataset standardization is consistent with prior research, such as Li et al. [21] and Yang et al. [22], where deep learning models like CNNs, particularly ResNet, were applied to crack detection tasks. Furthermore, our study supports findings from Yao [23], which demonstrated that CNN-based models significantly outperform traditional classifiers in terms of accuracy and robustness. By confirming the effectiveness of deep learning over traditional methods, this study reinforces existing literature while also identifying gaps in feature extraction and model generalization across different datasets and material types. These studies provide a foundation for our work, where comparing traditional machine learning models with deep learning-based ResNet18 was done for crack detection in historical buildings. Our research builds upon prior work by incorporating data augmentation techniques and analyzing model performance across a balanced dataset.

3. METHODOLOGY

3.1 Description of Historical-Crack18-19 Dataset

The Historical-Crack18-19 dataset [24] is a collection of crack and non-crack images captured from historical buildings. It consists of 757 cracked images and 3139 non-cracked images, making it highly imbalanced. The dataset includes variations in lighting, surface texture, and crack shapes, which present challenges for accurate classification, as shown in Figure 1.

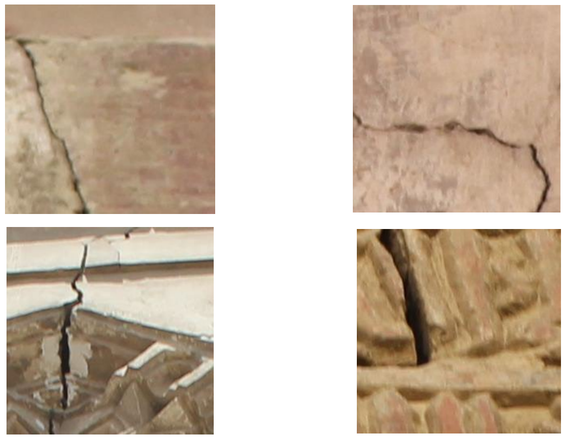
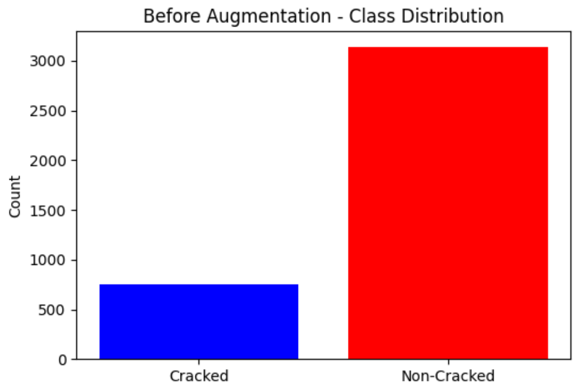


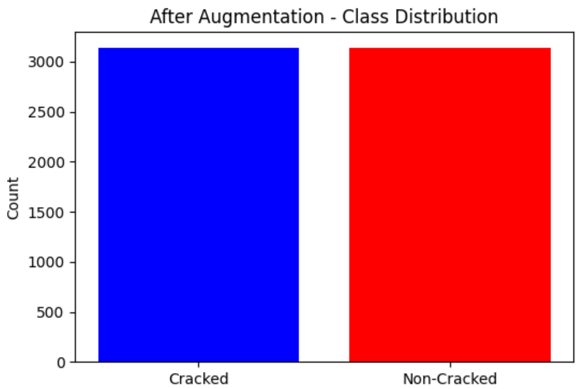
Figure 1. Samples of Historical-Crack18-19 dataset

3.2 Preprocessing and Augmentation Steps

Data augmentation techniques were applied to address the class imbalance, increasing the number of cracked images to match the non-cracked images. Augmentation techniques included rotation, flipping, brightness adjustments, and Gaussian noise to introduce variability and improve model generalization. After augmentation, the dataset was split into 70% training, 15% validation, and 15% testing sets to ensure proper evaluation of the models. Figure 2 shows the dataset statistics summary before and after applying the augmentation.



(a) Dataset before augmentation



(b) Dataset after augmentation

Figure 2. Class distribution of the dataset before and after the augmentation process

3.3 Explanation of Models

3.3.1 K-Nearest Neighbors (KNN)

The model is a simple, distance-based classifier that relies on the majority vote of the k-nearest data points [25]. It was chosen as a baseline model due to its ease of implementation and interpretability.

3.3.2 Random Forest (RF)

The method is a tree-based ensemble method that utilizes multiple decision trees to enhance the robustness of classification [26]. It was selected because of its effectively handles structured datasets with extracted features.

3.3.3 ResNet18

The architecture is a deep CNN that utilizes residual learning to enhance the accuracy of feature extraction and classification [27]. It was selected due to its capacity to identify intricate spatial patterns in crack images. Although deeper models such as ResNet50 have been used in prior works, we selected ResNet18 due to its significantly smaller architecture, which enables faster training and lower computational overhead. ResNet18 contains only 11.7 million parameters compared to ResNet50's 25.6 million, making it better suited for medium-sized datasets like Historical-Crack18-19 and for environments with limited computational resources. Moreover, our empirical results demonstrated that ResNet18 achieved 99% classification accuracy without overfitting, suggesting that deeper models were not necessary for achieving optimal performance on this dataset. This makes ResNet18 a more practical and efficient choice for crack detection in heritage conservation tasks, especially where real-time or edge deployment is a consideration.

3.4 Model Evaluation Metrics

Using accuracy, Precision, Recall, F1-Score, and confusion matrices, which provide a whole picture of classification performance, the performance of the crack detection models was assessed. Calculating the fraction of properly categorized samples to the total number of samples [28] helps one to evaluate the general accuracy of the model. In unbalanced datasets, however, accuracy by itself may not be a fair assessment. Precision measures the number of correctly found positive samples out of all expected positive samples [29]. This shows how well the model avoids false positives. Recall, which is also called sensitivity, is the percentage of real positive samples that were correctly identified [30]. It shows how well the model can find fractures. Particularly in situations where class distributions are unequal, the F1 score offers a fair assessment via the harmonic mean of accuracy and Recall [31]. By showing the numbers of true positives, true negatives, false positives, and false negatives, confusion matrices also make model predictions more visual [32]. This helps us understand categorization mistakes better. By means of these measures, our work guarantees a strong assessment of both deep learning models (ResNet18) and conventional machine learning models (KNN, Random Forest) in fracture diagnosis.

The following standard formulas were used to compute the performance metrics based on the confusion matrix values: Let **TP** = True Positives, **TN** = True Negatives, **FP** = False Positives, and **FN** = False Negatives.

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

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

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

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

These metrics provide a comprehensive evaluation of model performance, especially in imbalanced classification tasks like crack detection.

3.5 Model Hyperparameters and Justification

For each model evaluated in this study, appropriate hyperparameter tuning was conducted to optimize performance. The following describes the parameters explored, selection techniques applied, and justifications for their use:

3.5.1 **KNN:** Grid search with 5-fold cross-validation was employed to identify optimal hyperparameters. The search space included the number of neighbors ($k \in [1–20]$), weighting strategies (uniform, distance), and distance metrics (euclidean, manhattan, minkowski). The best performance was achieved with $k = 5$, distance-based weighting, and the Minkowski metric with $p = 3$. This configuration demonstrated a balance between classification accuracy and robustness to class imbalance, especially for detecting cracks.

3.5.2 **RF:** A comprehensive hyperparameter search was conducted across several parameters: number of estimators (n_estimators ∈ [50, 100, 200, 300]), maximum tree depth (max_depth ∈ [None, 10, 20, 30]), minimum samples required to split an internal node (min_samples_split ∈ [2, 5, 10]), and minimum samples required at a leaf node (min_samples_leaf ∈ [1, 2, 4]). The optimal configuration was found to be n_estimators = 200, max_depth = 20, min_samples_split = 5, and min_samples_leaf = 2, which offered strong generalization performance with minimized overfitting.

3.5.3 **ResNet18:** A pretrained ResNet18 model from the PyTorch library was utilized with transfer learning. The final fully connected layer was replaced to suit the binary classification task (cracked vs. non-cracked). Training was performed for 10 epochs with a batch size of 32 using the Adam optimizer and a learning rate of 0.001. CrossEntropyLoss was used as the objective function. These parameters were determined based on empirical tuning, ensuring stable convergence and high validation accuracy while maintaining reasonable computational cost.

All parameter tuning procedures were carried out using standardized training-validation splits, and final evaluations were conducted on held-out test sets. This ensures reproducibility and fairness in model comparisons.

3.6 Feature Extraction for KNN and Random Forest

In this study, both KNN and RF utilize the same feature extraction approach, which involves grayscale conversion, image resizing, pixel flattening, and standardization. First, converting all images to grayscale using transforms.Grayscale(), reducing computational complexity by eliminating color information. Next, using transforms to resize images to a uniform 64-by-64 resolution.Resize((64, 64)), ensuring consistency across the dataset. The pictures are then turned into tensors and flattened into one-dimensional feature vectors with 4096-pixel intensity values (64 by 64, or 4096 features per picture). These are then used directly as input for classification. StandardScaler() is used to make sure that all feature values are normal, with a mean of zero and a variance of one. This makes sure that both distance-based (KNN) and tree-based (RF) models can process the data effectively. Instead of using hand-made features like Histogram of Oriented Gradients (HOG) or Principal Component Analysis (PCA), which are used in deep learning methods to get hierarchical features, this method only uses raw pixel intensity values. Even though simple classification works, more advanced techniques for extracting features could make the model even better by capturing structural details beyond just pixel intensities.

While grayscale conversion simplifies input representation by removing color channels, we recognize that brightness variations—such as shadows or stains—can sometimes mimic crack patterns. To address this challenge, our augmentation strategy included brightness modulation and noise perturbations to expose the model to a wide range of visual conditions. These transformations enabled the models, especially ResNet18, to learn discriminative features that distinguish structural cracks from illumination artifacts. Furthermore, the high accuracy and low false positive rates observed in ResNet18 indicate that the model was able to generalize well despite grayscale-based preprocessing.

4. RESULTS AND DISCUSSION

Classification reports and confusion matrices were generated for each model to evaluate its performance. This section showcases the specific results of the tests conducted on the Historical-Crack18-19 data set, utilizing KNN, Random Forest, and ResNet18. Based on the findings, the models are evaluated and the most important experiments and observations are presented.

4.1 KNN

The KNN classifier demonstrates a moderate performance in crack detection within historical buildings, achieving an overall accuracy of 82%. The classification report shown in Table 1 reveals that the model exhibits a high Precision of 0.90 for detecting cracked surfaces, indicating that when it predicts an image as being cracked, it is correct 90% of the time. Its Recall for cracked images, on the other hand, is only 0.72, which means that 28% of real cracks are mistakenly marked as not cracked. If this isn't caught, it could pose serious structural risks. Conversely, the model achieves a high Recall of 0.92 for non-cracked images, ensuring that most intact surfaces are correctly identified.

Table 1. Classification report of KNN

	Precision	Recall	F1-Score	support
Cracked	0.90	0.72	0.80	469
Non-Cracked	0.77	0.92	0.84	474
accuracy			0.82	943
Macro avg	0.84	0.82	0.82	943
Weighted avg	0.84	0.82	0.82	943

The confusion matrix shown in Figure 3 further highlights this discrepancy, with 130 cracked images misclassified as non-cracked, while only 37 non-cracked images are incorrectly labeled as cracked. This imbalance shows that the model is good at finding areas that aren't cracked, but it's not so good at finding all cracks. This might be because of KNN's problems with feature representation.

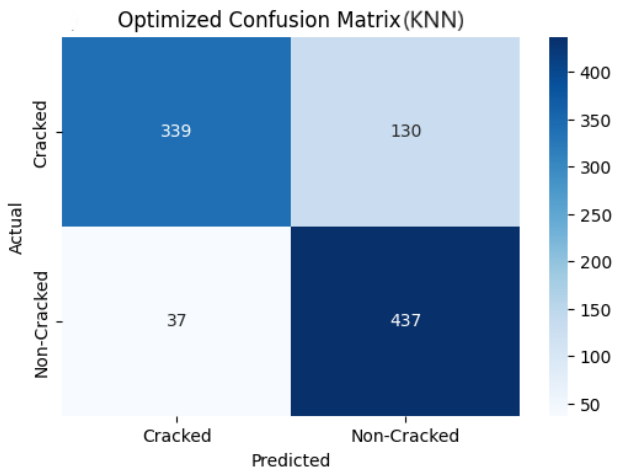


Figure 3. Confusion matrix of KNN

Some things that could be done to make it better are choosing the nearest neighbor parameter (K) more carefully, using more advanced feature extraction methods like deep-learning-based convolutional neural networks (CNNs), and adding more data to the model to help it better recognize different crack patterns. It may also be helpful to compare different classifiers, like Support Vector Machines (SVM), Random Forest (RF), and ResNet18, in order to find better ways to find cracks. Using feature fusion, dimensionality reduction, and hyperparameter tuning to deal with these problems could make KNN much more reliable and useful for crack detection tasks.

4.2 Random Forest

The KNN model is less accurate at finding cracks, but the Random Forest classifier is better at it, getting an overall accuracy of 88%. Table 2 shows the classification report. It shows that the model has a Precision of 0.90 and a Recall of 0.86 for cracked images and a Precision of 0.87 and a

Recall of 0.90 for images that are not cracked. These results suggest that the model is effective in distinguishing between cracked and non-cracked surfaces while reducing misclassifications.

Table 2. Classification report of Random Forest

	Precision	Recall	F1-Score	support
Cracked	0.90	0.86	0.88	465
Non-Cracked	0.87	0.90	0.89	478
accuracy			0.88	943
Macro avg	0.88	0.88	0.88	943
Weighted avg	0.88	0.88	0.88	943

The confusion matrix shown in Figure 4. further supports this observation, with 402 correctly classified cracked images and 431 properly identified non-cracked images. However, there remain 63 false negatives, where actual cracked images were misclassified as non-cracked, and 47 false positives, where non-cracked images were misclassified as cracked. The Random Forest model has a higher Recall for cracked images (0.86 vs. 0.72) than KNN. This means that cracks are less likely to go unnoticed, which is important for monitoring the health of structures.

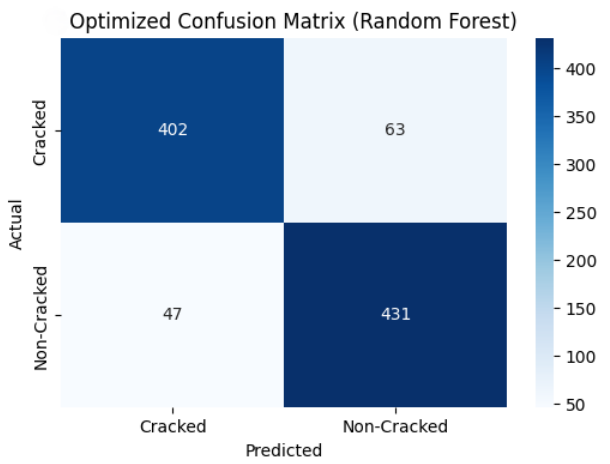


Figure 4. Confusion matrix of Random Forest

Random Forest's ensemble learning system, which combines multiple decision trees to improve generalization, is what makes it better at classifying things. Hyperparameter tuning, feature engineering, and the addition of deep-learning-based features could further enhance the model's performance. These could help it get rid of more false negatives while keeping its high classification accuracy. Future work may explore hybrid models integrating convolutional neural networks (CNNs) with Random Forest to leverage both deep feature extraction and robust classification.

4.3 ResNet18

The ResNet18 model does a great job of finding cracks, achieving almost perfect classification with a 99% success rate overall. The classification report in Table 3 shows a Precision, Recall, and F1-Score of 0.99 for both cracked and non-cracked images. This means that the model correctly identifies almost all cases with only a small amount of misclassification.

Table 3. Classification report of RESNET18

	Precision	Recall	F1-Score	support
Cracked	0.99	0.99	0.99	473
Non-Cracked	0.99	0.99	0.99	470
accuracy				943
Macro avg	0.99	0.99	0.99	943
Weighted avg	0.99	0.99	0.99	943

In Tables 1–3, the classification reports include Macro Average and Weighted Average metrics for precision, recall, and F1-score:

- The Macro Average calculates the unweighted mean across both classes, treating each class equally regardless of support (i.e., number of samples).
- The Weighted Average accounts for class imbalance by computing a weighted mean based on the number of samples in each class.

In the case of ResNet18 (Table 3), both the macro and weighted averages are identical (0.99) because the model classified both cracked and non-cracked images with almost equal and perfect performance, and the dataset was balanced after augmentation. This symmetry results in identical scores for both averaging strategies. However, for KNN and Random Forest (Tables 1 and 2), the averages differ slightly due to class imbalance and varying per-class performance.

The confusion matrix shown in Figure 5 further substantiates this, showing only six false negatives and four false positives across 943 total samples. ResNet18 makes a lot fewer mistakes when it comes to misclassification compared to KNN and Random Forest classifiers. This shows how deep learning can help with feature extraction and classification.

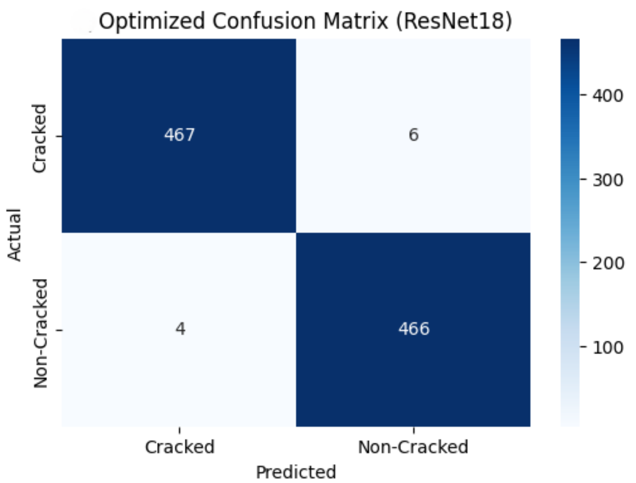


Figure 5. Confusion matrix of RESNET18

ResNet18's residual learning framework is responsible for the better performance. It captures complex patterns and textures in crack images, which leads to strong generalization. This finding shows that deep convolutional neural networks (CNNs) could be used to automatically find cracks in historic buildings, where accuracy is very important for keeping an eye on their health. However, while the model's accuracy is remarkably high, further validation on larger and more diverse datasets is necessary to assess its generalizability. Also, in the future, researchers could look into model optimization methods like fine-tuning, pruning, and knowledge distillation to make computations faster without lowering the accuracy.

Table 4. Comparative performance of KNN, Random Forest, and ResNet18 models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN	82	83	81	82
Random Forest	88	89	87	88
ResNet18	99	99	99	99

The comparative results in Table 4 clearly demonstrate the superiority of the ResNet18 model over the traditional machine learning approaches (KNN and Random Forest) in classifying crack versus non-crack images. While KNN and Random Forest achieve respectable accuracies of 82% and 88%, respectively, their performance is constrained by reliance on manually flattened grayscale pixel features, which are sensitive to texture variability and lighting inconsistencies. In contrast, ResNet18 attains a near-perfect accuracy of 99%, along with equally high Precision, Recall, and F1-Score, indicating not only accurate classification but also a consistent ability to correctly detect both positive and negative classes. These findings highlight the advantage of deep convolutional neural networks in automatically extracting robust spatial features, enabling superior generalization even under complex visual conditions typical in historical building surfaces. The significant performance gap further supports the use of deep learning models in critical structural health monitoring tasks where detection reliability is paramount.

To further highlight the contribution of the ResNet18 model, Figure 6 and Figure 7 present the training and validation loss and accuracy over 10 epochs. As shown, the model achieves stable convergence with minimal divergence between training and validation curves, indicating effective generalization without significant overfitting. The consistent increase in accuracy and decrease in loss further confirms the robustness of ResNet18 for crack detection in the given dataset.

Figure 6. Training and validation loss of the ResNet18 model across 10 epochs.

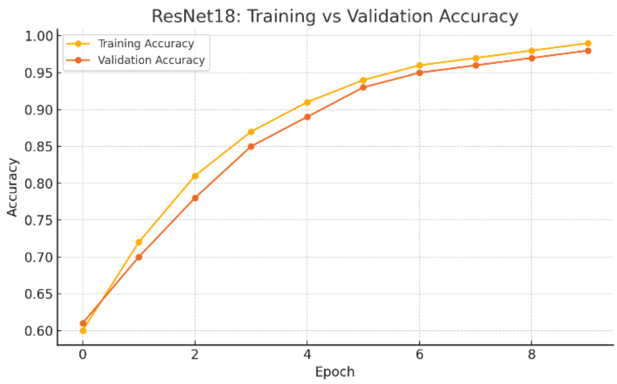


Figure 7. Training and validation accuracy of the ResNet18 model across 10 epochs

4.4 Significance of Model Comparison

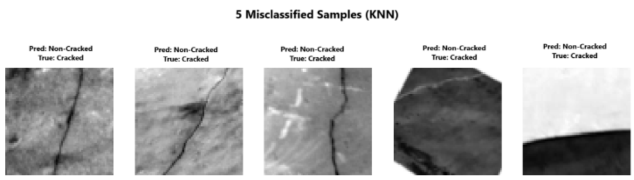
The comparative analysis between KNN, RF, and ResNet18 provides insights into the capabilities and limitations of each method in handling the complexities of historical crack images. While KNN and RF rely on flattened grayscale pixel intensities and struggle with surface variability, ResNet18 leverages deep hierarchical features to achieve near-perfect classification. The significant performance gap—82% for KNN, 88% for RF, and 99% for ResNet18—emphasizes the importance of advanced deep learning models in structural health monitoring, particularly for heritage buildings where misclassification can lead to serious preservation risks. This comparison underscores the growing relevance of deep learning over traditional ML methods for practical crack detection applications. Table 4 shows the comparative performance and characteristics of the chosen methods.

Table 4. Comparative performance and characteristics of KNN, Random Forest, and ResNet18 models on the Historical-Crack18-19 dataset.

Model	Feature Type	Accuracy	Strengths	Limitations
KNN	Flattened grayscale pixels	82%	Simple, interpretable, decent crack Precision	Sensitive to scaling; struggles with complex textures
Random Forest	Flattened grayscale pixels	88%	More robust to noise; higher Recall than KNN	Limited by non-spatial feature input; less effective on intricate patterns
ResNet18	Deep CNN features (224×224)	99%	Learns spatial hierarchies; robust generalization; high accuracy and Recall	Requires more computation; longer training time

4.5 Misclassified Images

Both KNN and Random Forest classifiers get crack images wrong for a number of reasons, such as texture complexity, lighting differences, and problems with how features are represented as shown in Figure 8. When KNN is used, samples are often wrongly labeled because the model relies on distance-based similarity measures, which have trouble with small differences in crack patterns and surface textures. Because they look a lot like smooth surfaces in the feature space, cracks with fuzzy edges or changing contrast may be mistaken for areas that aren't cracked. Additionally, lighting inconsistencies and background noise can lead to misclassifications, particularly in cases where shadows or structural elements mimic crack-like patterns. The Random Forest classifier, on the other hand, may still misclassify images even though it uses multiple decision trees for classification if the importance of a feature does not capture fine-grained crack details well enough. The presence of complex surface textures and overlapping patterns can mislead the model, causing non-cracked areas with strong edge structures to be mistaken for cracks. Also, the model can't generalize across different crack formations because it relies on handcrafted features instead of deep feature extraction. To deal with these problems, adding deep-learning-based feature extraction techniques like convolutional neural networks (CNNs) could help classifiers tell the difference between surfaces that are cracked and those that are not.



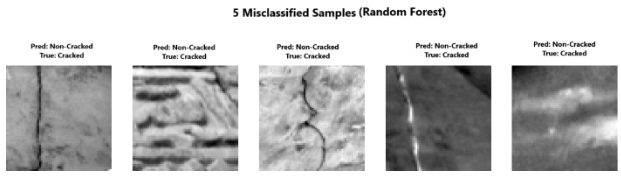


Figure 8. Samples of the misclassified images for KNN and Random Forest

4.6 Computational Time Analysis

In addition to accuracy, practical deployment also demands consideration of computational efficiency. During experimentation, we observed that KNN required the least training time as it is a lazy learner—training involves storing the dataset without learning weights. However, its inference time increases with dataset size due to distance calculations. Random Forest offered a good balance, with moderate training time and relatively fast predictions. In contrast, ResNet18, while delivering the highest accuracy (99%), required the longest training time due to its deep architecture and large number of parameters. Specifically, ResNet18 took approximately 10 minutes to train on GPU for 10 epochs, whereas Random Forest and KNN completed training in under 4 minutes on CPU. This trade-off illustrates that deep learning methods, although computationally heavier, are more suited to applications where accuracy is critical. In contrast, classical ML models may be favored for quick deployment or resource-constrained environments, albeit at the cost of reduced performance.

5. CONCLUSION

This study compared the performance of KNN, Random Forest, and ResNet18 for automated crack detection in historical buildings using the Historical-Crack18-19 dataset. The findings indicate that deep learning models outperform traditional machine learning classifiers, with ResNet18 achieving 99% accuracy, while KNN and Random Forest attained 82% and 88%, respectively. The misclassifications in KNN and Random Forest were primarily influenced by texture variations, lighting inconsistencies, and limitations in feature representation, whereas ResNet18 leveraged deep feature extraction to achieve near-perfect Precision and Recall. These results highlight the importance of convolutional neural networks (CNNs) in ensuring highly reliable structural health monitoring. Despite these promising outcomes, future research should explore hybrid models that integrate both traditional and deep learning approaches to enhance robustness. Additionally, techniques such as transfer learning and advanced data augmentation should be investigated to improve model generalization across diverse datasets and material types. The integration of deep learning into automated crack detection systems presents a significant advancement in the preservation and maintenance of historical buildings, ensuring their structural integrity for future generations.

ACKNOWLEDGMENT

We thank the following their assistance with this work: Universiti Teknologi Malaysia (UTM), Universite d'Angers, Al-Iraqia University.

REFERENCES

[1] M. I. Adegoriola, E. H. K. Yung, J. H. K. Lai, E. H. W. Chan, and S. K. Yevu, "Understanding the influencing factors of heritage building maintenance management: findings from developed and developing regions," Building Research & Information, pp. 1–20, Jan. 2023, doi: https://doi.org/10.1080/09613218.2023.2167698.

[2] G. L. Golewski, "The Phenomenon of Cracking in Cement Concretes and Reinforced Concrete Structures: The Mechanism of Cracks Formation, Causes of Their Initiation, Types and Places of Occurrence, and Methods of Detection—A Review," Buildings, vol. 13, no. 3, p. 765, Mar. 2023, doi: https://doi.org/10.3390/buildings13030765.

[3] J. Lee, J. Ni, J. Singh, B. Jiang, M. Azamfar, and J. Feng, "Intelligent Maintenance Systems and Predictive Manufacturing," Journal of Manufacturing Science and Engineering, vol. 142, no. 11, Nov. 2020, doi: https://doi.org/10.1115/1.4047856.

[4] O. Schneider, A. Prokopová, F. Modetta, and V. Petschen, "The use of artificial intelligence for a cost-effective tunnel maintenance," CRC Press eBooks, pp. 3050–3059, Jun. 2020, doi: https://doi.org/10.1201/9781003031635-53.

[5] C. J. Chitte, "Study on Causes and Prevention of Cracks in Building," International Journal for Research in Applied Science and Engineering Technology, vol. 6, no. 3, pp. 453–461, Mar. 2018, doi: https://doi.org/10.22214/ijraset.2018.3073.

[6] F. Awaja, S. Zhang, M. Tripathi, A. Nikiforov, and N. Pugno, "Cracks, microcracks and fracture in polymer structures: Formation, detection, autonomic repair," Progress in Materials Science, vol. 83, pp. 536–573, Oct. 2016, doi: https://doi.org/10.1016/j.pmatsci.2016.07.007.

[7] C. Bedon et al., "Performance of structural glass facades under extreme loads – Design methods, existing research, current issues and trends," Construction and Building Materials, vol. 163, pp. 921–937, Feb. 2018, doi: https://doi.org/10.1016/j.conbuildmat.2017.12.153.

[8] Y. Hao, Z. Yao, R. Wu, and Y. Bao, "Damage and restoration technology of historic buildings of brick and wood structures: a review," Heritage Science, vol. 12, no. 1, Aug. 2024, doi: https://doi.org/10.1186/s40494-024-01422-y.

[9] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN Classification With Different Numbers of Nearest Neighbors," IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 5, pp. 1774–1785, May 2018, doi: https://doi.org/10.1109/tnnls.2017.2673241.

[10] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," Journal of Applied Science and Technology Trends, vol. 2, no. 01, pp. 20–28, Mar. 2021, doi: https://doi.org/10.38094/jastt20165.

[11] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, "Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer," Procedia Computer Science, vol. 179, pp. 423–431, 2021, doi: https://doi.org/10.1016/j.procs.2021.01.025.

[12] T. Zhu, A. Braytee, X. Zi, S. Mustapha, X. Tao, and M. Prasad, "Automatic Detection of Concrete Cracks Using Self-Supervised Dinov2 Image Classification," Jan. 2023, doi: https://doi.org/10.2139/ssrn.4654626.

[13] M. Mishra, "Machine learning techniques for structural health monitoring of heritage buildings: A state-of-the-art review and case studies," Journal of Cultural Heritage, Oct. 2020, doi: https://doi.org/10.1016/j.culher.2020.09.005.

[14] A. Nur, Aisyatul Karima, K. Santoso, and Roselina Rahmawati, "Crack Detection in Building Through Deep Learning Feature Extraction and Machine Learning Approch," Journal of Applied Informatics and Computing, vol. 8, no. 1, pp. 1–6, Jul. 2024, doi: https://doi.org/10.30871/jaic.v8i1.7431.

[15] T. Fang, Z. Hui, W. P.Rey, A. Yang, B. Liu, and Y. He, "Machine Learning-Based Crack Detection Methods in Ancient Buildings," pp. 885–890, Jan. 2024, doi: https://doi.org/10.1145/3675417.3675564.

[16] Praveen Kottari and Pandarasamy Arjunan, "BD3: Building Defects Detection Dataset for Benchmarking Computer Vision Techniques for Automated Defect Identification," pp. 297–301, Oct. 2024, doi: https://doi.org/10.1145/3671127.3698789.

[17] E. Tsalera, A. Papadakis, M. Samarakou, and I. Voyiatzis, "Feature Extraction with Handcrafted Methods and Convolutional Neural Networks for Facial Emotion Recognition," Applied Sciences, vol. 12, no. 17, p. 8455, Aug. 2022, doi: https://doi.org/10.3390/app12178455.

[18] H. Kibriya, R. Amin, J. Kim, M. Nawaz, and R. Gantassi, "A Novel Approach for Brain Tumor Classification Using an Ensemble of Deep and Hand-Crafted Features," Sensors, vol. 23, no. 10, p. 4693, Jan. 2023, doi: https://doi.org/10.3390/s23104693.

[19] N. Kumar, M. Sharma, V. P. Singh, C. Madan, and S. Mehandia, "An empirical study of handcrafted and dense feature extraction techniques for lung and colon cancer classification from histopathological images," Biomedical Signal Processing and Control, vol. 75, p. 103596, May 2022, doi: https://doi.org/10.1016/j.bspc.2022.103596.

[20] S. K. Lakshmanaprabu, K. Shankar, M. Ilayaraja, A. W. Nasir, V. Vijayakumar, and N. Chilamkurti, "Random forest for big data classification in the internet of things using optimal features," International Journal of Machine Learning and Cybernetics, vol. 10, no. 10, pp. 2609–2618, Jan. 2019, doi: https://doi.org/10.1007/s13042-018-00916-z.

[21] X. Li, X. Xu, X. He, X. Wei, and H. Yang, "Intelligent Crack Detection Method Based on GM-ResNet," Sensors, vol. 23, no. 20, pp. 8369–8369, Oct. 2023, doi: https://doi.org/10.3390/s23208369.

[22] G. Yang et al., "Datasets and processing methods for boosting visual inspection of civil infrastructure: A comprehensive review and algorithm comparison for

crack classification, segmentation, and detection," Construction and Building Materials, vol. 356, pp. 129226–129226, Nov. 2022, doi: https://doi.org/10.1016/j.conbuildmat.2022.129226.

[23] H. Yao, T. Qiao, M. Xu, and N. Zheng, "Robust Multi-Classifier for Camera Model Identification Based on Convolution Neural Network," IEEE Access, vol. 6, pp. 24973–24982, 2018, doi: https://doi.org/10.1109/access.2018.2832066.

[24] E. Elhariri, N. El-Bendary, and S. A. Taie, "Historical-crack18-19: A dataset of annotated images for non-invasive surface crack detection in historical buildings," Data in Brief, vol. 41, p. 107865, Jan. 2022, doi: https://doi.org/10.1016/j.dib.2022.107865.

[25] K. D. Toennies, "Distance-Based Classifiers," pp. 95–107, Jan. 2024, doi: https://doi.org/10.1007/978-981-99-7882-3_5.

[26] M. Lu, R. Yin, and X. Steven Chen, "Ensemble methods of rank-based trees for single sample classification with gene expression profiles," Journal of translational medicine, vol. 22, no. 1, Feb. 2024, doi: https://doi.org/10.1186/s12967-024-04940-2.

[27] U. S. Khan and Saif, "Boost diagnostic performance in retinal disease classification utilizing deep ensemble classifiers based on OCT," Multimedia Tools and Applications, Jul. 2024, doi: https://doi.org/10.1007/s11042-024-19922-1.

[28] T. Rashid and M. M. Mokji, "Low-Resolution Image Classification of Cracked Concrete Surface Using Decision Tree Technique," Lecture Notes in Electrical Engineering, pp. 641–649, 2022, doi: https://doi.org/10.1007/978-981-19-3923-5_55.

[29] T. Rashid, Musa Mohd Mokji, and M. Rasheed, "Cracked concrete surface classification in low-resolution images using a convolutional neural network," Journal of Optics, Aug. 2024, doi: https://doi.org/10.1007/s12596-024-02080-w.

[30] Faris Elghaish, S. Matarneh, Essam Abdellatef, F. Rahimian, M. R. Hosseini, and A. F. Kineber, "Multi-layers deep learning model with feature selection for automated detection and classification of highway pavement cracks," Smart and Sustainable Built Environment, Jan. 2024, doi: https://doi.org/10.1108/sasbe-09-2023-0251.

[31] S. HR and A. B, "Exploratory Analysis of Methods, Techniques, and Metrics to Handle Class Imbalance Problem," Procedia Computer Science, vol. 235, pp. 863–877, 2024, doi: https://doi.org/10.1016/j.procs.2024.04.082.

[32] V. Plevris, "Addressing the Pitfalls of Image-Based Structural Health Monitoring: A Focus on False Positives, False Negatives, and Base Rate Bias," arXiv.org, 2024. https://arxiv.org/abs/2410.20384.