IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1578~1592
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1578-1592

Image feature extraction for road surface damage classification

Octaviani Hutapea, Sarifuddin Madenda, Hustinawaty, Iffatul Mardhiyah
Doctoral Program of Information Technology, Gunadarma University, Depok, Indonesia

Article Info

Article history:

Received Sep 19, 2025
Revised Jan 22, 2026
Accepted Feb 6, 2026

Keywords:

Convolutional neural network
Cracks
Feature extraction
Patching
Potholes

ABSTRACT

Road surface deterioration poses a critical risk to driving safety and comfort, necessitating timely and accurate detection to support effective maintenance. Manual inspection methods are often inefficient, underscoring the need for automated approaches based on computer vision. This study investigates the integration of feature extraction techniques histogram of oriented gradients (HOG) and local binary pattern (LBP) with convolutional neural network (CNN) architectures ResNet50 and InceptionV3 for the classification of road damage. A dataset of 1,580 images was categorized into five damage types: alligator crack, longitudinal crack, other crack, patching, and potholes. Experimental results indicate that HOG–ResNet50 achieved 79% accuracy, while LBP–InceptionV3 yielded the best performance at 97%. The contributions of this study are threefold: i) an automated framework is proposed that combines texture-based features with deep learning for road damage detection, ii) the LBP–InceptionV3 combination is shown to provide superior accuracy compared to conventional pairings, and iii) the approach offers a scalable and reliable alternative to manual inspection methods, supporting more efficient road maintenance planning.

Corresponding Author:

Octaviani Hutapea
Doctoral Program of Information Technology, Gunadarma University
Depok, Indonesia
Email: octaviahutapea@gmail.com

1. INTRODUCTION

The advancement of information and communication technology in transportation includes navigation systems in aircraft, GPS-based route determination, intelligent transportation systems, and technologies for detecting road surface damage [1]. It is of paramount importance to consider road surface damage, as it has the potential to impact the comfort of road users. The maintenance and supervision of road surface damage are frequently conducted manually, even in the present era. This will undoubtedly be a time-consuming process, involving the input of numerous experts and potentially higher costs during the construction phase [2]. Consequently, the automatic detection of road surface damage may be potential solution to this problem.

Visual observation of road damage can be enhanced through the application of digital image processing technologies. Image processing facilitates the development of automated methods for visual inspection with high levels of accuracy in detecting, classifying, and quantifying road surface damage [3]. In addition, the application of data augmentation techniques can improve detection model performance by increasing image variation during the training phase. For example, random cropping and transformation of crack images have been shown to improve mean precision by up to 4.1% compared to conventional mask region-based convolutional neural network (R-CNN), supported by instance segmentation for more optimal detection and mapping of road damage [4]. Furthermore, augmentation based on geometric transformations, such as rotation and random cropping, has been employed to represent variations in

perspective and crack positioning within images, thereby enhancing model's ability to identify cracks with diverse shapes and orientations [5].

Feature extraction plays a crucial role in determining the effectiveness of road damage detection models. Conventional methods such as histogram of oriented gradients (HOG) and grayscale HOG effectively represent texture patterns and have been successfully integrated with convolutional neural networks (CNN) for pavement damage detection [6]. Wavelet-based transformations have also been applied to capture multiscale information and improve CNN classification performance, achieving F1-scores of up to 0.96 in identifying damage types such as alligator cracks, linear cracks, and crack-free surfaces [7]. The synergy between augmentation techniques, feature extraction methods, and modern CNN architectures not only enables automatic and accurate detection of road damage but also supports the identification of severity levels and the recording of damage locations. Furthermore, deep learning approaches combined with infrared imaging and EfficientNet-B3 models have demonstrated high accuracy in classifying crack severity and assessing road damage levels, reaching up to 96% accuracy [8], [9].

A novel pixel-level crack detection model, Crack U-Net, is proposed with a U-shaped convolutional architecture. Trained on 3,000 pavement crack images using the Adam optimizer, the model demonstrates high robustness to noise, background variation, and image quality. Experimental evaluation shows that Crack U-Net achieves superior performance with average accuracy of 96.1%, F1-score of 0.94, and intersection over union (IoU) of 0.92, outperforming conventional methods that fail under noisy or fragmented crack conditions [10]. This study employs a deep neural network based on an attention mechanism and residual structure to detect road damage using three datasets: DeepCrack, CrackForest, and Road Image dataset. The proposed method demonstrates higher precision compared to several previous techniques [11].

To enhance the resolution of road crack images, a super-resolution generative adversarial network (SRGAN) with a pre-trained VGG19 backbone is employed. The super-resolved images are subsequently classified using a CNN with a pre-trained ResNet50 model. Correlation analysis is conducted to evaluate prediction performance before and after super-resolution, demonstrating that the proposed approach is more stable, efficient, and effective compared with existing methods. Experimental results show that the model achieves an accuracy of 98.2% on the evaluated dataset [12]. A pre-trained Faster R-CNN model was used for crack detection on concrete road surfaces using 323 high-resolution images, generating 1,128 bounding boxes. The results show reliable detection under various weather conditions, but accuracy decreases significantly in low-light environments, with up to 50% fewer cracks detected around sunset and nighttime [13].

Automatic and real-time road damage detection system developed using deep learning models on resource-constrained edge devices. It evaluates RetinaNet152 and MobileNetV2 architectures, focusing on performance optimization through hyperparameter tuning [14]. In addition to implementation-focused studies, a comprehensive review of vision-based pothole detection methods discusses various computer vision and deep learning approaches, including CNN- and you only look once (YOLO)-based models, providing a broader methodological context for pothole detection research [15].

Deep learning-based image processing has been applied for road damage detection in Semarang using the YOLO algorithm. The detected damage locations were mapped with global navigation satellite system (GNSS) integration, achieving an average accuracy of 88% and a Kappa coefficient of 86%, with positional accuracy reaching an root mean square error (RMSE) of ±5.6 m [16]. In a related study, road damage detection on Tegar Beriman Street, Bogor Regency, was performed using the k-nearest neighbor (KNN) method. The pavement damage was classified into three categories: cracks, alligator cracks, and potholes. Images were acquired with a Canon EOS M100 camera (6000×4000 resolution; 35×52 cm frame), and the model achieved an average accuracy of 83% [17].

The detection of road damage on pavement is achieved through the application of diverse image processing techniques, including the steerable Gaussian filter, projection integral, and image texture analysis. These techniques are utilized to ascertain the condition of the asphalt pavement surface. In the construction of the model, the light gradient boosting machine, deep neural network, and CNN are employed for the purpose of recognizing a variety of damage patterns, including longitudinal, transverse, diagonal, minor fatigue, and severe cracks. The study employed a dataset comprising 12,000 data points, yielding an average model accuracy exceeding 0.93. The highest level of accuracy was achieved through the utilization of light gradient boosting, with a recorded accuracy rate of 0.96 [18].

A CNN model was developed to detect damage in concrete pavements by applying max pooling, optimization techniques, and hyperparameter tuning to improve performance. The proposed model achieved an accuracy of 97.8% and provides an efficient and economical solution for detecting cracks in low-pixel-density concrete surface images without requiring high-cost image capture devices [19]. In comparative studies, crack detection is often used to simplify and accelerate road damage detection systems. However, several studies focus only on crack detection without measuring crack depth. Since road surfaces vary in structure, a generalized approach is required to effectively detect cracks. Image-based crack detection has

proven to be an effective method for reducing the complexity and time required for manual inspection, allowing human resources to be utilized more efficiently [20].

Another study proposes a crack classification method for road pavements using a multi-stage image preprocessing framework, including gray value transformation, K-means clustering, tensor voting, non-maximum suppression, and the random sample consensus (RANSAC) algorithm. Experimental results show that the method achieves an average F1-score of 0.7879. The method also reaches a classification accuracy of 96%, outperforming conventional image processing and neural network-based approaches [21].

Although previous studies have achieved strong performance in crack detection using handcrafted descriptors, wavelet-based transformations, CNN classifiers, and pixel-level segmentation models, most research remains focused primarily on crack-type damage and offers limited coverage of more complex defects such as patches and potholes. Additionally, many existing approaches rely on single feature-learning strategy, making them less effective in capturing both fine-grained texture variations and larger structural patterns across diverse road conditions. These limitations highlight the need for a more adaptive and discriminative framework capable of handling multiple damage categories under varying environmental conditions. To address these gaps, this study proposes a hybrid feature-learning approach that explicitly integrates traditional texture descriptors with deep CNN architectures. Two configurations are introduced: i) HOG features combined with the initial layer of ResNet50 and ii) local binary pattern (LBP) features fused with the InceptionV3 architecture initialized with ImageNet weights. This dual pipeline design aims to enhance classification accuracy, improve generalization across multiple road damage types, and provide a more efficient and reliable solution for automated pavement-condition assessment.

2. METHOD

The objective of this research is to identify the stages of road damage using a CNN model. The initial stages of this research are the acquisition and analysis of data from images of road pavement damage. These images were taken following the procedures outlined in the survey guidelines. The labeling process is conducted using Roboflow, which delineates bounding box for each distinct type of road damage, including alligator cracks, longitudinal cracks, other cracks, patching, and potholes shown in Figure 1.

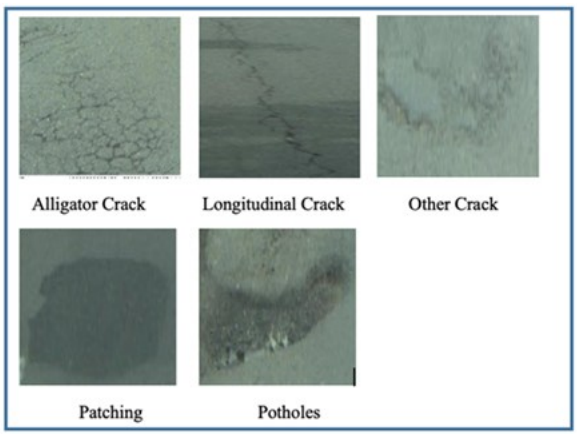

Figure 1. Image of type road crack

Data pre-processing is the transformation of data formats and structures to align with the requirements of an analytical process. The image cropping procedure is employed to obtain images of specific road damage types, which have been labelled using bounding boxes at the subsequent labeling stage. The size of each image of road damage is adjusted in accordance with the specifications of the model that is to be created. Following the size adjustment process, image data augmentation is conducted using a straightforward method, namely rotation and flip. Feature extraction is conducted on images of road damage to extract texture-based features, which serve as a representation of the image for the purpose of classifying damage objects. The data set is formed by dividing the data into categories according to the type of damage observed. The CNN model is employed for the formation of the damage type identification model, with training, validation, and testing data from the dataset. System testing and validation are conducted using a

confusion matrix to assess the accuracy of the model. The coordinates of damage distribution on the road are determined by identifying location of each damage point on the map. Figure 2 illustrates the research stages.

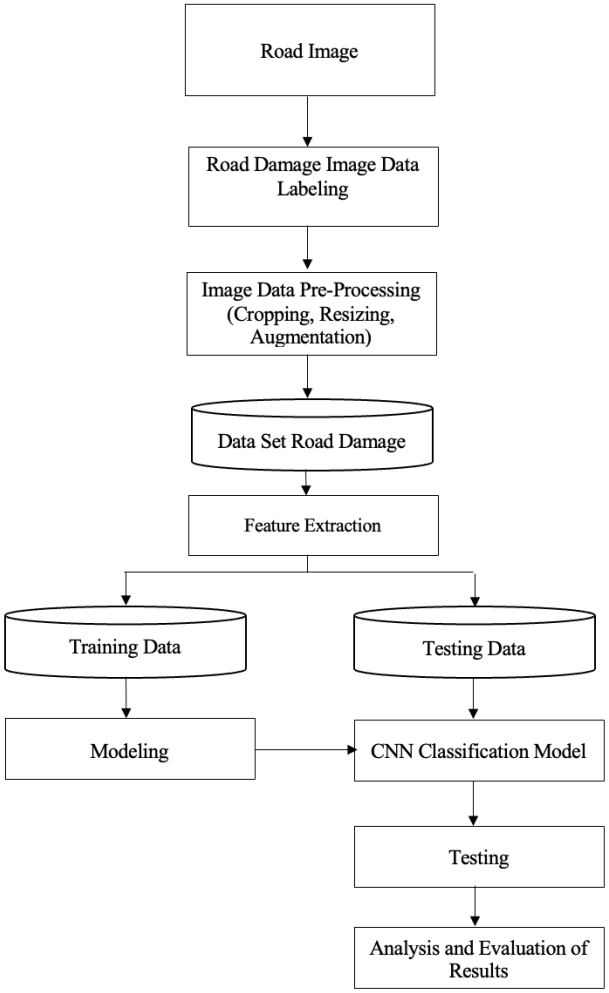

Figure 2. Research stages of road surface damage classification

2.1. Acquisition road image

The acquisition of data pertaining to the condition of the road surface was conducted using the Hawkeye 2000 vehicle. The road images were captured using a forward-facing camera with a minimum viewing angle of 120° from the vehicle's frontal axis. During the road survey conducted with the KSJJ Hawkeye 2000 system, the image acquisition interval was set to a maximum of 10 meters [22]. The following example illustrates a road image produced by the Hawkeye 2000 road network survey vehicle, as depicted in Figure 3. This figure depicts the exemplar form of the image that will undergo processing or identification. The image in question contains a multitude of objects, which presents a significant challenge in this study. The objective is to identify the type of road damage, which is constrained by the predetermined limitations of this study.

2.2. Road damage image data labeling

Image labeling is conducted via the delineation of each damage type on the road image with a bounding box, as illustrated in Figure 4. The labeling process for road images entails the identification of five distinct damage categories: alligator crack, longitudinal crack, other crack, patching, and potholes. The following sections present examples of images corresponding to each damage type. Figure 4 illustrates an example of an image that has been labeled with the specific type of damage, in this case, alligator cracks. In a single image of a road, it is possible to identify more than one type of damage.

Figure 3. Road image from HawkEye2000 vehicle

Figure 4. Example of labeling road damage image data

2.3. Image data pre-processing

The preprocessing of image data, as illustrated in Figure 5, commences with the initial step of cutting the image in accordance with the bounding box delineated for each image. The input image is then adjusted to align with the requisite dimensions of the model to be formed. The objective of augmentation of road damage image data is to produce image variations for each class, thereby increasing the diversity of the data set. Furthermore, the augmentation stage assists in ensuring a more balanced availability of images for each damage type.

The process of data augmentation, which is employed during the data processing stage, serves to enhance the diversity of datasets utilized in the training of models. Furthermore, data augmentation can enhance model generalization by incorporating minor alterations to the training data without compromising the intrinsic characteristics of the data set. Initially, the image of the specific type of road damage that has been extracted from the preceding stage is imported into the program. Subsequently, the image of the type of road damage is augmented using the horizontal flip technique, a 90° rotation, and an 180° rotation. Table 1 presents a detailed account of the distribution of each category of damage before and after augmentation. As indicated in Table 1, the initial data set comprises 316 images. Following the application of the data augmentation process, the total number of data points increased to 1,580 images. The types of road damage detected include cracks, patching, and potholes, which exhibit distinct characteristics. This presents a challenge in this study, specifically in developing a model that can accurately recognize all types of damage.

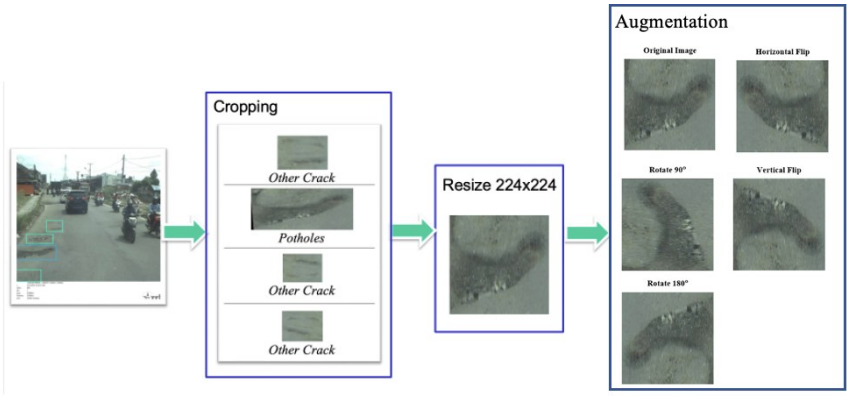

Figure 5. Visualization of pre-processing stages of road image data

Table 1. Data distribution based on damage type before and after augmentation

Type of road damage	Before augmentation	After augmentation
Alligator crack	99	495
Long crack	66	330
Other crack	46	230
Patching	44	220
Potholes	61	305
Total	316	1580

2.4. Histogram of oriented gradient

The HOG function, which is part of the scikit-image library, is employed for the purpose of calculating HOG features from an image. The process of extracting features commences with the conversion of an RGB image to grayscale, as HOG is designed to operate on the intensity of grayscale pixels to calculate the gradient. The HOG function employs a filter operator, such as the Sobel filter, to calculate the horizontal and vertical gradients of a grayscale image. The gradients are employed to ascertain the magnitude and direction of the intensity variation at each pixel. The image is then subdivided into smaller units, typically 8×8 pixels. In each cell, a histogram of the gradient orientations is calculated [23].

The objective of this histogram is to collate information regarding the gradient orientations present within the specified cell. Subsequently, multiple cells are aggregated to form a block, with the number of cells typically being either 2×2 or 3×3. The histograms of the cells in the block are normalized to mitigate the effects of changes in lighting and contrast. This normalization is achieved through the implementation of a method such as L2-Hys. The final feature vector is obtained by concatenating the block-normalize is defined as a vector that points in the direction of the most rapid increase in intensity. Mathematically, the process is expressed in (1) [24], where the histogram of the entire image is computed to yield a one-dimensional feature representation based on gradient information. $f(x,y)$ denotes the pixel intensity at coordinate (x,y). The term $f(x,y+1)-f(x,y-1)$ represents the vertical gradient component, capturing intensity changes along the y-axis, while $f(x+1,y)-f(x-1,y)$ represents the horizontal gradient component, capturing intensity changes along the x-axis.

$$\nabla f\ (x,y) = \begin{bmatrix} f(x,\ y+1)-f(x,\ y-1) \\ f(x+1,\ y)-f(x-1,\ y) \end{bmatrix} \quad (1)$$

2.5. Local binary pattern

LBP feature extraction is an image extraction operator that calculates the LBP pattern for each pixel in a grayscale image. This is achieved by comparing the neighboring pixel values with the center pixel value. The outcome of LBP feature extraction is a new image, wherein each pixel represents the LBP pattern of the surrounding local area. The initial step in the process of extracting features from an image using the LBP pattern is to transform the original 3-dimensional image, comprising the color values red, green, and blue, into a grayscale image. The dimensions of the road image, specifically its height and width, are employed to ascertain the boundaries of the loop during the processing of each pixel. The LBP image should be initialized by creating an empty array with the same dimensions as the gray image and the uint8 (unsigned 8-bit integer) data type, which will be used to store the LBP value of each pixel. The eight neighboring pixels are initialized for comparison with the center pixel, with consideration given to their relative position in

coordinates (dy, dx). The result of the comparison between the center pixel and the neighboring pixels that have been initialized will be stored in position (y, x). The histogram of the new image, generated using the LBP pattern, is then calculated. The histogram obtained from the new extracted image is converted into a one-dimensional array with a float32 data type. The LBP process is represented in the mathematical notation of (2) and (3). The variables used in the LBP formulation are defined as follows: *Gc* denotes the gray value of the central pixel, while *Gp* represents the gray value of its neighboring pixels. The parameter *P* indicates the number of neighbors considered, and *R* specifies the radius of the neighborhood. The histogram is normalized so that the total frequency is 1, thereby facilitating fair and consistent comparisons between images, as illustrated in Figure 6 [25].

$$LBP_{P,R} = \sum_{p=1}^{p} 2^{(p-1)} \times x\,(Gp - Gc) \tag{2}$$

$$X_{(x)} = \begin{cases} 1, & x \geq 0 \\ 0, & else \end{cases} \tag{3}$$

Figure 6 illustrates the process of feature extraction using the LBP operator. Each pixel intensity within a 3×3 neighborhood is compared with the central pixel value (175). Neighboring pixels with intensity values greater than or equal to the central pixel are assigned a value of 1, while those with lower values are assigned 0. This comparison generates a binary pattern, which is subsequently converted into its decimal equivalent (150). The resulting decimal values across the image form the LBP histogram, which serves as a compact descriptor of local texture information.

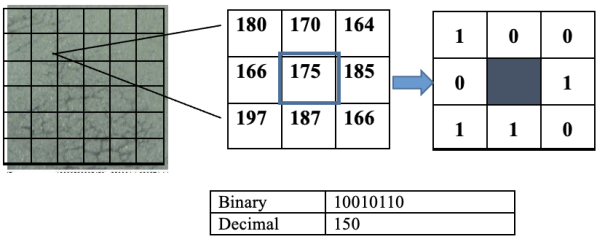

Figure 6. Process feature extraction using LBP

2.6. Model architecture

A CNN is typically composed of one or more fully connected (FC) layers, which are analogous to the input and output layers found in feedforward neural networks. The final convolutional or pooling layer provides the input to the FC layer, which generates the final output of the CNN. In the context of image classification, a CNN can be conceptualized as a composite of two principal components: a feature extraction module and a classification module. Both the convolutional and pooling layers are responsible for feature extraction [26]. The research methodology employed not only utilized feature extraction from pre-existing CNN models but also incorporated feature extraction using the HOG and LBP techniques. These techniques were applied and combined with the CNN model to address the diversity of features associated with each type of damage, with the objective of enhancing the accuracy of the resulting model.

2.6.1. ResNet50

The process of feature extraction on the ResNet50 architecture will be integrated with the calculation results obtained from feature extraction using the HOG technique. The ResNet50 architecture utilized in the study employs a pre-trained model with weights trained on the ImageNet dataset. It is necessary to initialize the ResNet50 model without an initial layer for feature extraction. This section comprises multiple convolutional layers and residual blocks, which are trained to extract features from images. Subsequently, the image is converted to a NumPy array using the Python imaging library (PIL). Modifications to the Keras model are implemented by incorporating a batch dimension into the image array. The images are then adjusted to the format expected by ResNet50, including pixel normalization (e.g., reducing pixel values to a range that matches the model training). Feature extraction is performed with the ResNet50 model to make predictions on the processed images. This involves producing a feature vector from the average pooling layer and then transforming the feature results into a one-dimensional vector. Model architecture using feature extraction HOG and ResNet50 shown in Figure 7.

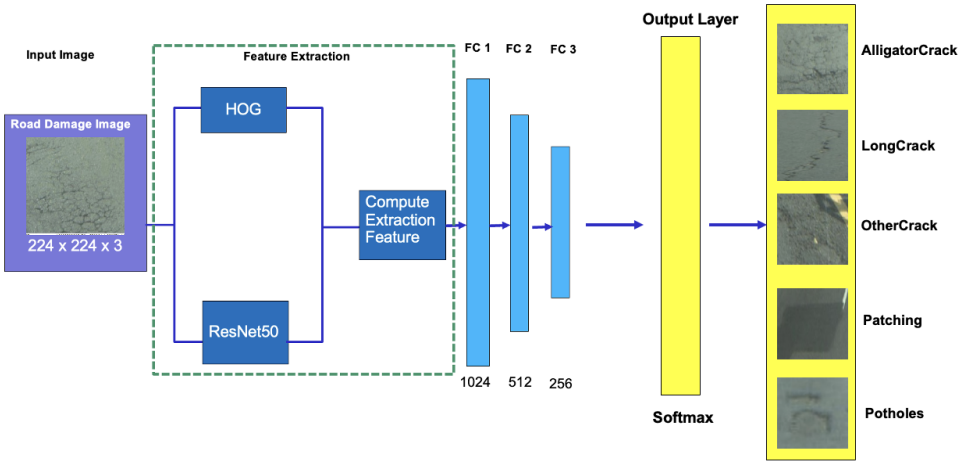

Figure 7. Model architecture using feature extraction HOG and ResNet50

2.6.2. InceptionV3

This study will employ a comparison of the HOG extraction with the ResNet50 architecture, utilizing the basic InceptionV3 model with ImageNet weights and feature extraction using LBP. The primary architectural structure of InceptionV3 comprises several pivotal components, including the input layer, which receives images with a fixed dimension. The utilized size is (224, 224, 3). Convolutional layers are employed for the initial extraction of features from the image through the application of convolutions. This layer employs a variety of filter sizes, enabling the capture of features at disparate scales. Batch normalization is subsequently applied following the convolutional layer, with the objective of stabilizing and accelerating the training process. Model architecture using feature extraction LBP and InceptionV3 shown in Figure 8.

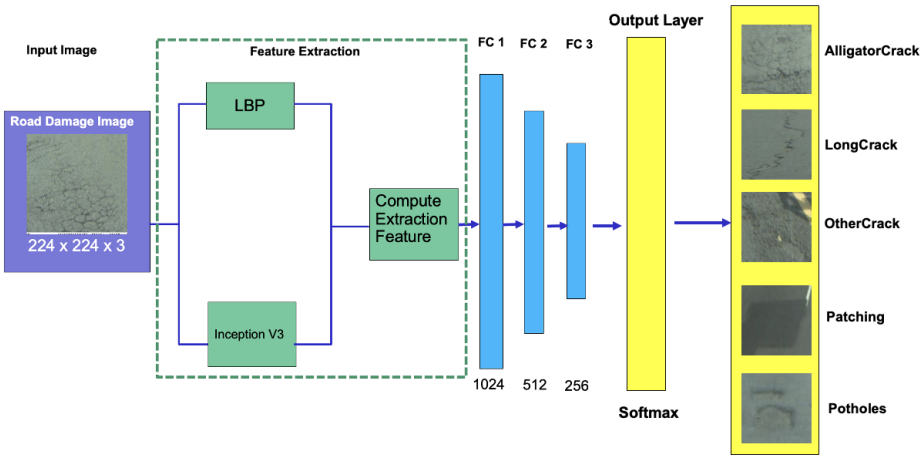

Figure 8. Model architecture using feature extraction LBP and InceptionV3

Inception modules represent the primary building blocks of InceptionV3. Each Inception module comprises multiple convolutional paths with diverse filter sizes, in addition to a pooling path. This module combines the results of the various paths using the technique of concatenation, thereby enabling the model to capture features at different scales and levels of detail. The filter kernels utilized have dimensions of 1×1, 3×3, and 5×5, in addition to 3×3 pooling, which enables the capture of information with varying receptive field sizes. Reduction modules are employed to diminish the spatial dimension and augment the number of feature channels, thereby reducing the computational load while maintaining the integrity of pertinent information. A global average pooling layer is introduced at the conclusion of the model, replacing the FC layer. This process yields a one-dimensional tensor for each channel, which reduces the spatial dimension while maintaining the integrity of the feature information.

3. RESULTS AND DISCUSSION

The data utilized for the classification of road damage types comprises 1,580 data points. In this study, five distinct types of road damage were identified on paved roads: alligator crack, longitudinal crack, other crack, patching, and potholes. At the training stage, each model is also configured. This includes the Adam optimizer, a learning rate of 0.0001, a batch size of 32, the use of categorical loss, and the metrics of accuracy. Additionally, three epochs are initialized, with values of 10, 50, and 100, respectively. In this study, the dataset was divided into three experimental scenarios to evaluate the robustness of the proposed models. The first scenario allocated 80% of the data for training, 10% for validation, and 10% for testing. The second scenario employed a 70%, 15%, and 15% split, while the third scenario utilized 60% for training, 20% for validation, and 20% for testing. These variations in data partitioning were designed to assess the consistency and generalization performance of the models under different proportions of training and evaluation datasets. Table 2 illustrates the results of feature extraction for each damage type using the HOG and ResNet50 algorithms, as well as the LBP and InceptionV3 methods.

Table 2 represent the numerical feature vectors generated by each extraction pipeline. These numerical outputs highlight the distinct encoding patterns produced by HOG+ResNet50 and LBP+InceptionV3, providing insight into the descriptive capacity of each method prior to classification. The table does not report loss- or accuracy-based metrics; rather, it presents raw feature descriptors that illustrate how each algorithm captures structural variations across the five categories of road surface damage. The extracted feature values vary depending on the damage type. Alligator cracks produced higher values compared to other damage types due to their complex crack patterns, which are more readily detected across all methods. Both alligator cracks and other cracks yielded relatively smaller and more balanced values across the methods, reflecting consistent characteristics regardless of the extraction technique. In the case of the LBP+InceptionV3 combination, a notably high value of 12.613 was observed for longitudinal cracks, indicating that the linear patterns were effectively captured. Similarly, for patching, the same method produced a high value of 10.388, attributable to the significant textural differences between patching materials and the original pavement surface, which allowed LBP to clearly distinguish the contrast.

Table 2. Feature extraction results

Feature extraction	Type of road damage				
	Alligator crack	Longitudinal crack	Other crack	Patching	Potholes
HOG+ResNet50	0.749168	0.042307	0.174297	0.012343	0
	0.082943	0.015201	0	0.004288	0.029468
	0.022727	0.032387	0.015664	0.053842	1.17567
	etc..	etc..	etc..	etc..	etc..
LBP+InceptionV3	0.526182	12.61310	0.719508	10.38806	0.680684
	0.027246	0.508347	0.015483	0.060447	0.354046
	0.05018	0.198934	0.027148	0.08447	0.019285
	etc..	0.113472	0.159563	0.969368	0.478809
		etc..	etc..	etc..	etc..

3.1. Convolutional neural network models

Baseline experimentation was first conducted using two widely adopted CNN architectures, ResNet50 and InceptionV3. These models serve as representative deep-learning benchmarks for evaluating the capability of pure CNN-based feature extraction in classifying road-surface damage across five categories. To provide a reliable comparison, both architectures were trained under identical hyperparameter settings and data preprocessing pipelines. As shown in Figure 9, the ResNet50 baseline demonstrates substantial performance limitations, achieving an overall accuracy of only 0.35 and producing near-zero precision, recall, and F1-scores across most classes. The corresponding learning curves further reveal poor convergence behavior, as the testing accuracy remains almost stagnant throughout training, indicating that ResNet50 is unable to learn sufficiently discriminative representations from the dataset. In contrast, the InceptionV3 baseline yields noticeably stronger results, reaching an overall accuracy of 0.83 with more balanced evaluation metrics across most categories. Its training dynamics also exhibit stable convergence, with training accuracy approaching 0.98 and testing accuracy consistently maintained within the 0.78–0.85 interval, highlighting a significantly enhanced feature-learning capability relative to ResNet50.

Despite the improved performance observed in InceptionV3, the gap between its training and testing accuracy indicates limited generalization when relying solely on deep feature extraction. This persistent discrepancy suggests that deep CNN architectures alone may be insufficient for capturing fine-grained variations in road-surface patterns. Thereby motivating the development of a more comprehensive hybrid framework that integrates handcrafted descriptors with CNN-derived representations to enhance robustness and classification reliability.

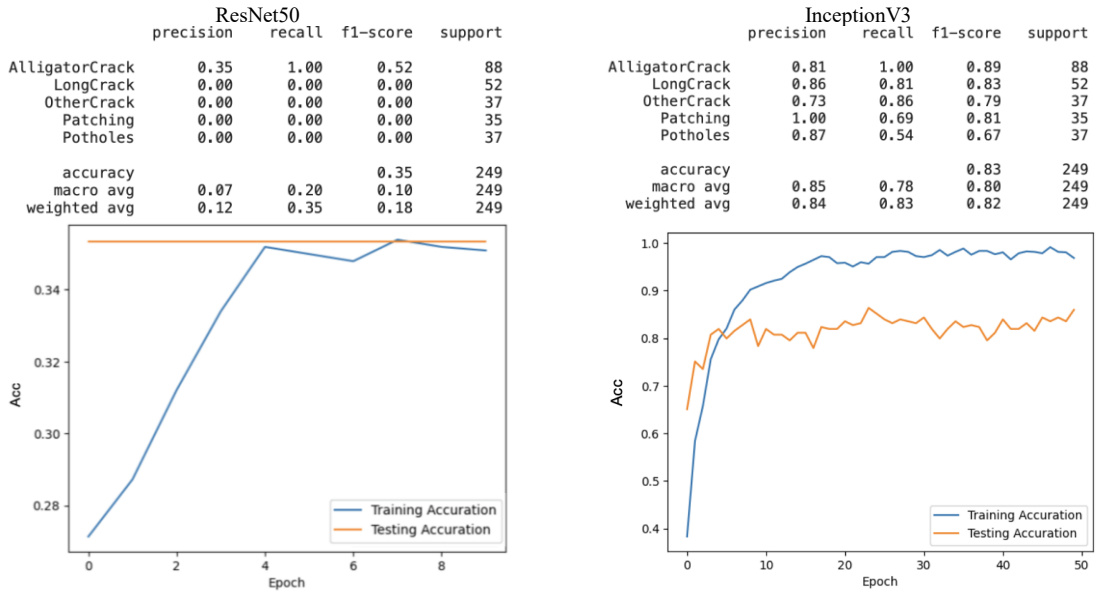

Figure 9. Baseline performance of ResNet50 and InceptionV3 on road-damage classification

3.2. HOG+ResNet50

Table 3 presents the accuracy, precision, recall, and F1-score values obtained from the testing phase with feature extraction HOG and ResNet50. Before presenting the detailed results, it is important to highlight that the evaluation was conducted across multiple data-splitting configurations and training durations to examine the model's robustness under varying learning conditions. This approach allows a more comprehensive assessment of how the balance between training, validation, and testing data influences the overall performance of the HOG–ResNet50 feature extraction framework.

Table 3 presents the performance of the HOG–ResNet50 model under different data-splitting scenarios and epochs. The model demonstrated stable accuracy between 0.73 and 0.79, with the best result achieved under the 60:20:20 split at epoch 50 (precision, recall, and accuracy of 0.79). This indicates that a larger validation portion provides stronger feedback during training, enabling better generalization despite reduced training data. In contrast, the 80:10:10 and 70:15:15 splits produced slightly lower but consistent results, suggesting that insufficient validation may limit performance improvements. Overall, the findings emphasize the role of data partitioning in shaping model robustness, particularly for diverse damage types such as cracks, patches, and potholes.

Table 3. Accuracy, precision, recall, and F1-score values HOG and ResNet50 feature extraction

Data split train:val:test	Epoch	Precision	Recall	F1-score	Accuracy
80:10:10	10	0.74	0.75	0.72	0.75
	50	0.77	0.78	0.77	0.78
	100	0.74	0.75	0.73	0.75
70:15:15	10	0.72	0.73	0.71	0.73
	50	0.74	0.75	0.73	0.75
	100	0.76	0.77	0.76	0.77
60:20:20	10	0.74	0.74	0.74	0.74
	50	0.79	0.79	0.78	0.79
	100	0.74	0.75	0.74	0.75

Figure 10 shows that the training loss decreases steadily while training accuracy increases consistently, whereas validation accuracy remains in the 0.70–0.75 range with fluctuating validation loss, indicating mild overfitting and convergence near epoch 50. The confusion matrix and classification report reflect balanced class performance with an overall accuracy of 0.78, and the receiver operating characteristic (ROC) curves confirm strong class separability with area under the curve (AUC) values approaching 0.90 across all categories. Moreover, epoch 50, as reported in Table 3, yields the best overall result with an accuracy of 0.79, reinforcing the model's optimal performance at mid-range training iterations.

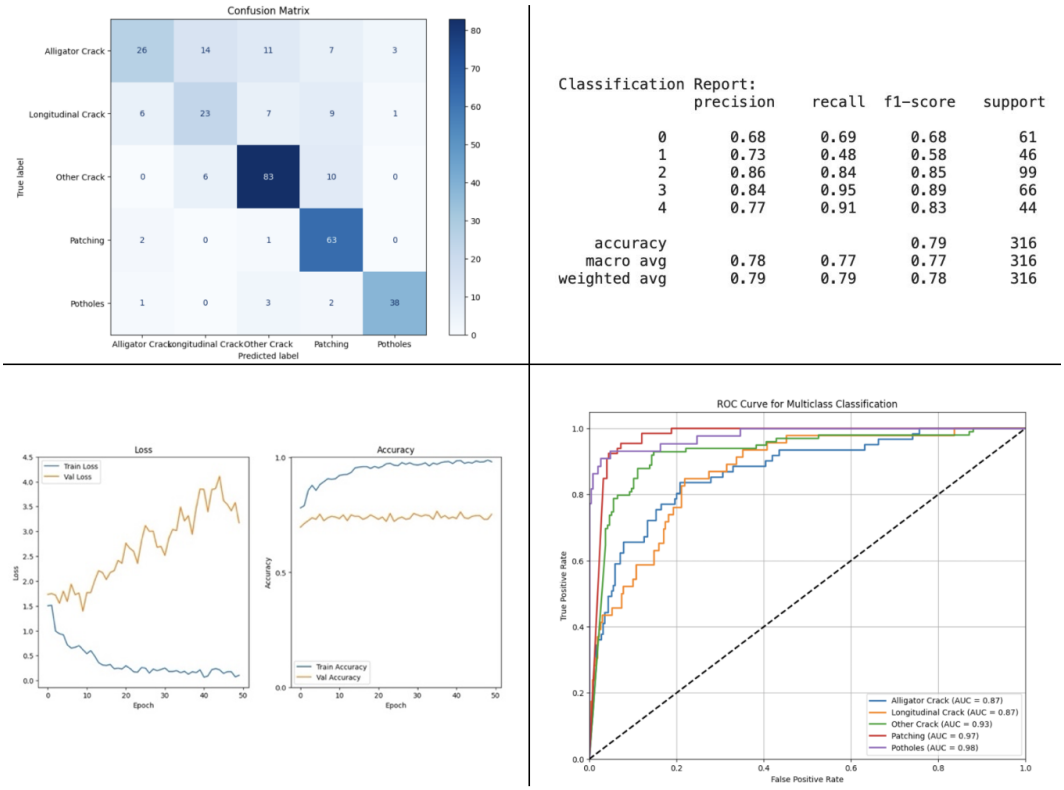

Figure 10. Best result training and validation curves with confusion matrix, classification report, and ROC for HOG+ResNet50

3.3. LBP+InceptionV3

Table 4 presents the accuracy, precision, recall, and F1-score values obtained from the testing phase with feature extraction LBP and InceptionV3. Table 4 shows that the LBP+InceptionV3 approach achieved consistently high performance across all scenarios, with accuracy, precision, recall, and F1-score above 0.89. The best results were obtained with the 70:15:15 split at epoch 50, reaching 0.97 across all metrics. These findings confirm the effectiveness of combining local texture descriptors with deep feature representations for robust road damage classification. This result is further illustrated in Figure 10, where the training–validation curves show stable convergence and peak accuracy, confirming the model's robustness in road damage classification.

Table 4. Accuracy, precision, recall, and F1-score values LBP and InceptionV3 feature extraction

Data split train:val:test	Epoch	Precision	Recall	F1-score	Accuracy
80:10:10	10	0.92	0.92	0.92	0.92
	50	0.94	0.94	0.94	0.94
	100	0.96	0.96	0.95	0.96
70:15:15	10	0.92	0.92	0.92	0.92
	50	0.97	0.97	0.97	0.97
	100	0.96	0.96	0.96	0.96
60:20:20	10	0.89	0.89	0.89	0.89
	50	0.92	0.91	0.91	0.91
	100	0.92	0.91	0.91	0.91

Figure 11 demonstrates a significantly improved performance compared to the previous model. The training curves show a steady decrease in training loss and a consistent increase in training accuracy, while the validation accuracy remains high at approximately 0.95–0.97 with relatively minor fluctuations in validation loss, indicating strong generalization and minimal overfitting. The confusion matrix shows that nearly all classes are predicted with high accuracy, with very few misclassifications across categories. This is further supported by the classification report, where precision, recall, and F1-scores range from 0.94 to 1.00,

resulting in an overall accuracy of 0.97, which is substantially higher than that of the earlier model. Moreover, the ROC curves indicate excellent class separability, with all AUC values approaching 1.000, reflecting near-perfect discriminative capability. Overall, these results demonstrate that LBP+InceptionV3 configuration achieves optimal performance, exhibiting better generalization, lower error rates, and more consistent evaluation metrics compared to the results presented in Figure 10.

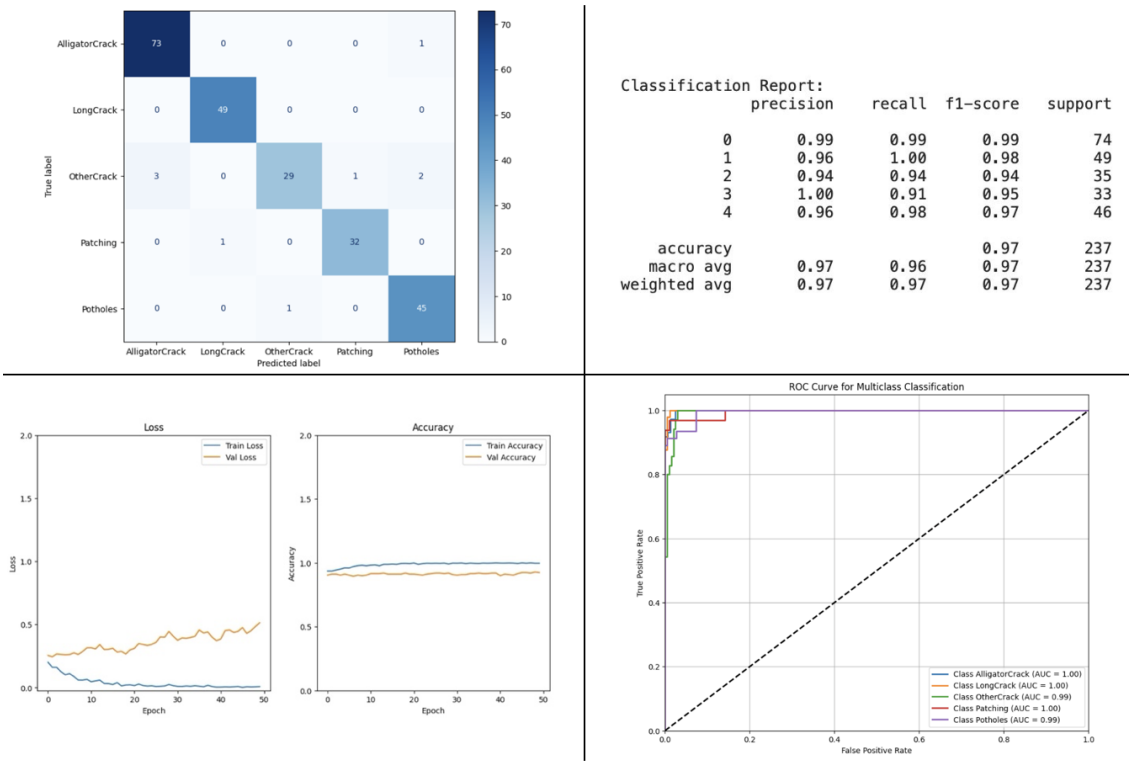

Figure 11. Best result training and validation curves with confusion matrix, classification report, and ROC for LBP+InceptionV3

3.4. Inference performance and statistical significance analysis

The performance assessment presented in this section corresponds to the best-performing model, specifically the configuration achieving the highest evaluation scores as shown in Table 4, obtained using the 70:15:15 data split at 50 training epochs. The inference analysis shows that the proposed model attains an average processing time of 0.0004768 s per image, equivalent to approximately 2,097 FPS. This performance markedly exceeds that of common lightweight architectures such as MobileNetV3 and SqueezeNet, which under optimized inference frameworks report throughput in the order of hundreds to a few thousand samples per second [27], [28]. Recent works demonstrate that lightweight CNN can indeed deliver low-latency inference suitable for real-time edge deployment. For instance, RDPNet a single-path CNN optimized for CPU-based devices achieves significantly faster inference than conventional multi-branch models, thanks to depthwise-separable convolution and structural re-parameterization [29].

System resource monitoring indicates that the performance evaluation was conducted on a high-performance computing (HPC) environment equipped with 1007.66 GB of RAM, of which 128.98 GB (13.4%) was utilized during execution, and a 256-core CPU. These specifications confirm that the model was tested on a platform optimized for large-scale parallel computation. Recent studies have shown that HPC architectures can substantially accelerate deep learning inference by exploiting micro-batching strategies and extensive task parallelism. A recent comprehensive benchmarking study on distributed deep learning frameworks reported that with up to 1,024 GPUs in a HPC cluster, deep CNN models such as ResNet50 achieved dramatic speed-ups compared to conventional workstation setups. This finding supports assertion that inference and training throughput can be significantly enhanced by leveraging HPC infrastructures.

A McNemar test was conducted to compare the prediction outcomes between model LBP+InceptionV3 (Model A) and model HOG+ResNet50 (Model B). The results showed: i) significant difference in performance, ii) (A correct, B wrong): 60, and iii) (A wrong, B correct): 7 where, model LBP+InceptionV3 correctly classified substantially more samples that model HOG+ResNet50 misclassified. The test statistic with continuity correction was χ^2 =40.36, with a corresponding p-value of 1.72×10^{-9} (≈0.00000000172), indicating a highly significant difference ($p < 0.05$). These findings confirm that model LBP+InceptionV3 performs significantly better than model HOG+ResNet50 on the same test set.

4. CONCLUSION

This study demonstrated that integrating handcrafted texture descriptors with deep CNNs significantly enhances road-damage classification performance. While pure CNN baselines showed varying capabilities ResNet50 performing poorly and InceptionV3 achieving moderate generalization the hybrid configurations yielded substantial improvements across all evaluation metrics. The HOG+ResNet50 model achieved stable mid-range accuracy (up to 0.79), whereas the LBP+InceptionV3 model delivered consistently superior results, reaching 0.97 accuracy, precision, recall, and F1-score under the 70:15:15 split at 50 epochs. The training–validation dynamics, confusion matrices, and ROC analyses collectively verified the robustness, strong class separability, and minimized overfitting of the proposed approach. Statistical validation using the McNemar test further confirmed that LBP+InceptionV3 performs significantly better than HOG+ResNet50 ($p \approx 1.72\times10^{-9}$). In addition to accuracy improvements, the proposed model demonstrated exceptional inference efficiency. With an average processing speed of 0.0004768 s per image (≈2,097 FPS), the model markedly outperformed common lightweight architectures and aligns with recent evidence highlighting the benefits of HPC infrastructures for deep learning workloads. These findings indicate that the LBP+InceptionV3 configuration not only delivers state-of-the-art accuracy but also satisfies real-time operational requirements, making it a highly effective and deployable solution for automated road-damage detection in intelligent transportation systems. Future work may extend this approach toward detecting structural damage in reinforced concrete and steel components, as well as adapting the framework for more complex tasks such as pixel-level segmentation and spatial localization, enabling broader applicability across diverse civil infrastructure systems.

ACKNOWLEDGMENTS

The authors acknowledge the support provided by Gunadarma University.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Octaviani Hutapea	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Sarifuddin Madenda		✓			✓		✓			✓		✓		
Hustinawaty	✓	✓		✓	✓	✓				✓		✓		
Iffatul Mardhiyah		✓		✓					✓	✓		✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

INFORMED CONSENT

This study does not involve human participants or personal data. Therefore, informed consent is not required.

ETHICAL APPROVAL

This research does not involve human subjects or animals, therefore ethical approval was not required.

DATA AVAILABILITY

The data that support the findings of this study were obtained from Bina Marga and are not publicly available due to data usage restrictions. The data are available from the corresponding author, [OH], upon reasonable request and with permission from Bina Marga.

REFERENCES

[1] S. Gössling, "ICT and transport behavior: a conceptual review," *International Journal of Sustainable Transportation*, vol. 12, no. 3, pp. 153–164, 2018, doi: 10.1080/15568318.2017.1338318.

[2] R. Hossen *et al.*, "RoadDamageBD," *Mendeley Data, V1*, 2025, doi: 10.17632/km53tmscxw.1.

[3] B. Benmhahe and J. A. Chentoufi, "Automated pavement distress detection, classification and measurement: a review," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, pp. 708–718, 2021, doi: 10.14569/IJACSA.2021.0120882.

[4] M. Zhao, X. Xu, X. Bao, X. Chen, and H. Yang, "An automated instance segmentation method for crack detection integrated with CrackMover data augmentation," *Sensors*, vol. 24, no. 2, 2024, doi: 10.3390/s24020446.

[5] A. Ashraf, A. Sophian, and A. A. Bawono, "Crack detection, classification, and segmentation on road pavement material using multi-scale feature aggregation and transformer-based attention mechanisms," *Construction Materials*, vol. 4, no. 4, pp. 655–675, 2024, doi: 10.3390/constrmater4040036.

[6] G.-H. Chen *et al.*, "Detection of highway pavement damage based on a CNN using grayscale and HOG features," *Sensors*, vol. 22, no. 7, 2022, doi: 10.3390/s22072455.

[7] R. Alfanz, R. Fahrizal, T. P. Utomo, T. Firmansyah, F. Muhammad, and I. M. Muztahidul, "Implementation of wavelet method and backpropagation neural network on road crack detection based on image processing," *SINERGI*, vol. 28, no. 3, 2024, doi: 10.22441/sinergi.2024.3.005.

[8] J. Estilong and T. Palaoag, "Literature review on road damage detection and severity recognition: leveraging computer vision," *Journal of Information Systems Engineering and Management*, vol. 10, no. 5s, pp. 498–512, 2025, doi: 10.52783/jisem.v10i5s.670.

[9] S. Matarneh, F. Elghaish, D. J. Edwards, F. P. Rahimian, E. Abdellatef, and O. Ejohwomu, "Automatic crack classification on asphalt pavement surfaces using convolutional neural networks and transfer learning," *Journal of Information Technology in Construction*, vol. 29, pp. 1239–1256, 2024, doi: 10.36680/j.itcon.2024.055.

[10] J. Huyan, W. Li, S. Tighe, Z. Xu, and J. Zhai, "CrackU-net: a novel deep convolutional neural network for pixelwise pavement crack detection," *Structural Control and Health Monitoring*, vol. 27, no. 8, 2020, doi: 10.1002/stc.2551.

[11] P. Jing, H. Yu, Z. Hua, S. Xie, and C. Song, "Road crack detection using deep neural network based on attention mechanism and residual structure," *IEEE Access*, vol. 11, pp. 919–929, 2023, doi: 10.1109/ACCESS.2022.3233072.

[12] K. Sathya, D. Sangavi, P. Sridharshini, M. Manobharathi, and G. Jayapriya, "Improved image based super resolution and concrete crack prediction using pre-trained deep learning models," *Journal of Soft Computing in Civil Engineering*, vol. 4, no. 3, pp. 40–51, 2020, doi: 10.22115/SCCE.2020.229355.1219.

[13] K. Hacıefendioğlu and H. B. Başağa, "Concrete road crack detection using deep learning-based Faster R-CNN method," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 46, no. 2, pp. 1621–1633, 2022, doi: 10.1007/s40996-021-00671-2.

[14] H. Mahmudah, A. S. Aisjah, S. Arifin, and C. A. Prastyanto, "Detecting road damage utilizing retinanet and mobilenet models on edge devices," *IAES International Journal of Artificial Intelligence*, vol. 14, no. 2, pp. 1430–1440, 2025, doi: 10.11591/ijai.v14.i2.pp1430-1440.

[15] Y. Safyari, M. Mahdianpari, and H. Shiri, "A review of vision-based pothole detection methods using computer vision and machine learning," *Sensors*, vol. 24, no. 17, 2024, doi: 10.3390/s24175652.

[16] B. Sasmito, B. H. Setiadji, and R. R. Isnanto, "Object detection: real-time road damage detection and geolocation using YOLOv8 and GNSS integration," *Ingénierie des systèmes d information*, vol. 30, no. 9, pp. 2321–2329, 2025, doi: 10.18280/isi.300909.

[17] J. Kusumaningrum, S. Madenda, Karmilasari, and Nahdalina, "Detection and classification of road damage based on image morphology and K-NN method (k nearest neighbour)," *International Journal of Engineering and Advanced Technology*, vol. 11, no. 5, pp. 86–90, 2022, doi: 10.35940/ijeat.E3543.0611522.

[18] N.-D. Hoang and Q.-L. Nguyen, "Computer vision-based recognition of pavement crack patterns using light gradient boosting machine, deep neural network, and convolutional neural network," *Journal of Soft Computing in Civil Engineering*, vol. 7, no. 3, pp. 21–51, 2023, doi: 10.22115/SCCE.2023.367276.1547.

[19] M. Padsumbiya, V. Brahmbhatt, and S. P. Thakkar, "Automatic crack detection using convolutional neural network," *Journal of Soft Computing in Civil Engineering*, vol. 6, no. 3, pp. 1–17, 2022, doi: 10.22115/SCCE.2022.325596.1397.

[20] M. R. S. Zawad, M. F. S. Zawad, M. A. Rahman, and S. N. Priyom, "A comparative review of image processing based crack detection techniques on civil engineering structures," *Journal of Soft Computing in Civil Engineering*, vol. 5, no. 3, pp. 58–74, 2021, doi: 10.22115/SCCE.2021.287729.1325.

[21] H. Wang, J. Tao, and H. Li, "Pavement crack classification and recognition algorithm combined with tensor voting and RANSAC," *IEEE Access*, vol. 12, pp. 72117–72130, 2024, doi: 10.1109/ACCESS.2024.3403893.

[22] ARRB Systems, "Hawkeye 2000 series," *ARRB Systems*. 2021. Accessed: Nov. 11, 2025. [Online]. Available: https://arrbsystems.com/app/uploads/2025/01/H2000.pdf?utm_source=chatgpt.com

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, vol. 1, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[24] S. K. G. Manikonda and D. N. Gaonkar, "Islanding detection method based on image classification technique using histogram of oriented gradient features," *IET Generation, Transmission & Distribution*, vol. 14, no. 14, pp. 2790–2799, 2020, doi: 10.1049/iet-gtd.2019.1824.
[25] K. Petranek, P. Janecka, and J. Vanek, "Using local binary patterns for object detection in images," *Global Journal of Computer Sciences: Theory and Research*, vol. 5, no. 1, pp. 07–12, 2015, doi: 10.18844/gjcs.v5i1.24.
[26] A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, and D. De, "Fundamental concepts of convolutional neural network," in *Recent Trends and Advances in Artificial Intelligence and Internet of Things*, Cham, Switzerland: Springer, 2020, pp. 519–567, doi: 10.1007/978-3-030-32644-9_36.
[27] A. Howard *et al.*, "Searching for MobileNetV3," in *2019 IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.
[28] I. J. Ratul, Y. Zhou, and K. Yang, "Accelerating deep learning inference: a comparative analysis of modern acceleration frameworks," *Electronics*, vol. 14, no. 15, 2025, doi: 10.3390/electronics14152977.
[29] J. Xu, Y. Zhao, and F. Xu, "RDPNet: a single-path lightweight CNN with re-parameterization for CPU-type edge devices," *Journal of Cloud Computing*, vol. 11, no. 1, 2022, doi: 10.1186/s13677-022-00330-5.

BIOGRAPHIES OF AUTHORS

Octaviani Hutapea holds a master's degree in Information Systems Management (PLSI) from Gunadarma University, completed in 2020. Prior to that, she earned a bachelor's degree in Informatics Engineering from the same university in 2017. Her academic background combines strong technical expertise with strategic information systems management. Her research interests include image processing, machine learning, deep learning, and computer networks. She can be contacted at email: octaviahutapea@gmail.com.

Prof. Sarifuddin Madenda holds a Doctor of Electronics and Image Processing, Universite de Bourgogne, France. Currently he is active as the head of Ph.D. programs of Information Technology and a lecturer at Ph.D. program at Gunadarma University. His research interest is in image and video processing, multimedia data compression, content-based image and video retrieval, steganography: encryption, decryption, coding and decoding of multimedia secret documents, real time system architecture (FPGA and ASIC design). He can be contacted at email: sarif@staff.gunadarma.ac.id.

Dr. Hustinawaty holds a Doctor of Information Technology degree from Gunadarma University, Indonesia, awarded in 2012. She also obtained her MMSI degree from Gunadarma University, Indonesia, in 1994. Currently, she serves as the head of the Master of Information Systems Management Study Program at Gunadarma University, Indonesia. Her research interests encompass information systems management and information systems technology. She can be contacted at email: hustina@staff.gunadarma.ac.id.

Dr. Iffatul Mardhiyah holds a study internship in the Doctoral Program at Université de Bourgogne, France, in 2013. Subsequently, she earned a doctoral degree in Information Technology from Universitas Gunadarma, Indonesia, in 2016. Prior to that, she received both a bachelor's degree (2009) and a master's degree (2012) in Mathematics from Universitas Indonesia. In addition, she obtained a second master's degree in Industrial Engineering and Management from Universitas Gunadarma in 2023. Her research interests lie in the fields of mathematics and computer science. She can be contacted at email: iffatul@staff.gunadarma.ac.id.