Signal and Image Processing Letters
Vol. 7., No. 1, 2025, pp. 93-104
ISSN 2714-6677
https://simple.ascee.org/

Classification of Road Damage in Sidoarjo Using CNN Based on Inception Resnet-V2 Architecture

Fathima Zahrah [a,1,*], I Gede Susrama Mas Diyasa [a,2], Wahyu Syaifullah Jauharis Saputra [a,3]

[a] Departement of Data Science, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia
[1] fazahrabawedan@gmail.com; [2] igsusrama.if@upnjatim.ac.id; [3] wahyu.s.j.saputra.if@upnjatim.ac.id
* corresponding author

ARTICLE INFO

Keywords
Road Damage Classification
Convolutional Neural Network
Inception ResNet-V2
Crack
Pothole
Sidoarjo

ABSTRACT

Road damage is a serious issue in Sidoarjo Regency, posing risks to road users' safety. This study aims to classify road surface conditions using a Convolutional Neural Network (CNN) model based on the Inception ResNet-V2 architecture. The research develops an image-based classification model by combining secondary data from Kaggle and primary data obtained through Google Street View API scraping, along with training strategies such as data augmentation, class balancing, early stopping, and model checkpointing. A total of 885 images were used, categorized into three classes: potholes, cracks, and undamaged roads. The model was trained over 20 epochs with early stopping triggered at epoch 15, when validation accuracy reached 95.95%. Evaluation on the test set showed a test accuracy of 83%. The undamaged road class achieved the highest performance with an F1-score of 0.89, while the pothole class recorded an F1-score of 0.79. The lowest performance was observed in the cracked road class, with an F1-score of 0.65, indicating the model's limited ability to detect fine crack features. This limitation is likely due to class imbalance and visual similarity between classes. Although the model demonstrated good generalization for the two majority classes, the performance gap between validation and test accuracy highlights the need to improve detection for minority classes. Future work is recommended to explore advanced augmentation techniques, increase the representation of minority class data, and consider alternative architectures or ensemble methods to enhance the model’s sensitivity to subtle road damage features.

1. Introduction

Based on the results of the IMD World Competitiveness Ranking (WCR) research in 2023, Indonesia ranked 51st out of 67 countries in terms of infrastructure, including road infrastructure [1]. However, roads play an important role in supporting community mobility, goods distribution, and economic and social growth [2]. This is evidenced by data from the Ministry of Public Works and Public Housing (PUPR) in 2023, showing that Indonesia only has approximately 43.98% of roads in good condition. East Java, as the province with the largest road infrastructure, recorded that 37% of regency/city roads are in poor condition and ranks third nationally in terms of the highest number of damaged roads [3]. Road damage that is not immediately addressed can endanger road user safety and potentially cause fatal accidents [4], even to the point of claiming road users' lives [5]. Road damage is recorded as one of the factors causing traffic accidents, where the Indonesian National Police recorded 40 accident cases in 2021 that were directly caused by damaged road conditions [6]. Additionally, the impact of road damage has a fairly strong relationship with accidents, although not statistically significant [7].

Sidoarjo is one of the areas in East Java Province that has relatively poor road conditions. The Regent of Sidoarjo, Ahmad Muhdlor Ali, mentioned that as of March 2024, there were 39 points of damaged and potholed roads spread across 15 sub-districts out of a total of 18 existing sub-districts [8]. Amid the high urgency for road infrastructure repair, the process of identifying and maintaining damage is still carried out manually. This method generally requires considerable time and human resources, especially in areas that are extensive or difficult to access [6]. The Head of the Technical Division of the Public Works and Highways Service of Sidoarjo Regency stated that the collection of road damage data in the area still heavily depends on community reports via telephone and field inspections. Common types of damage such as cracks and potholes are frequently found and have quite high accident risk levels. Sidoarjo residents have also submitted many complaints regarding the existence of road potholes in various locations [9], and several accident cases have been reported due to riders falling into these potholes [5]. Furthermore, observations at several locations show that the most frequently found types of damage are cracks and potholes [10], making both relevant representations for the damage classification used in this research.

The utilization of digital technology is needed to improve efficiency in handling road damage. One potential approach is online damage reporting through image uploads, which are then analyzed using Convolutional Neural Network (CNN)-based models. This research focuses on developing a road damage image classification model, which in the future can be integrated into online reporting systems. In this study, CNN with Inception ResNet-V2 architecture is used to classify road damage images. CNN is known to be capable of effectively extracting visual features from images [11]. Additionally, the Inception ResNet-V2 architecture is known to have superior performance in handling complex image data and producing accurate classifications. This architecture combines Inception modules, which allow the model to recognize features from various scales in images, as well as residual connections that function to maintain information flow between layers so that training is more stable, efficient, and capable of producing high accuracy even on complex image datasets [12].

The superiority of this architecture has also been proven in previous research, such as by Jinkang Wang et al. in a study titled "A Real-Time Bridge Crack Detection Method Based on an Improved Inception-ResNet-V2 Structure." The results of that research showed that the Inception ResNet-V2 architecture achieved 99% accuracy, indicating its ability to effectively recognize complex patterns. In that research, Inception ResNet-V2 was used to classify two categories: cracked and not cracked. The data used in model training was a combination of three datasets that formed a larger bridge crack detection dataset, namely the Bridge Surface Crack Dataset (BSDC). The model was trained using a learning rate of 0.001 and 50 epochs [13].

Research related to road damage classification has also been conducted by Arif Riyandi et al. in a study titled "Classification of Damaged Road Images Using Convolutional Neural Network Method." That research classified road images obtained from the Kaggle platform into three categories: undamaged roads, cracked roads, and potholed roads. In the preprocessing stage, four stages were used: conversion to grayscale, balance filter, mean filter, and median filter. The results of that research showed training accuracy of 88% and validation accuracy of 99% with a preprocessing model using the grayscale method [14]. Nevertheless, the results of that research showed signs of overfitting in the developed model.

Different from that research, this study implements the Inception-ResNet-V2 architecture which has proven superior in handling complex image data. Instead of using filters explicitly as in previous research, this study utilizes residual connections and Inception modules to perform automatic and deep feature extraction. The Inception structure uses various convolutional kernel sizes to capture features at various scales, while simultaneously reducing the number of parameters to keep model complexity low. Meanwhile, residual connections allow direct signal flow between layers, both forward and backward, thus helping maintain gradient stability, accelerating the training process, and improving model performance while reducing overfitting risk [13]. Additionally, this research also uses road image data from the Sidoarjo area obtained through integration with the Google Street View API, which is expected to improve model accuracy in classifying road conditions in that area.

The purpose of this research is to build a road damage image classification model using InceptionResNet-V2 architecture, focusing on the Sidoarjo Regency area. This model is expected to be integrated into an online road damage reporting system based on image uploads by the community, which is currently unavailable. The presence of this model is expected to be an initial step toward a digital reporting system that helps the Public Works and Highways Service in conducting road identification and maintenance more quickly, efficiently, and data-driven.

2. Method

This research was conducted through several systematic stages, starting from data collection to model evaluation. These stages were designed to ensure optimal training processes and comprehensive evaluation. The overall workflow of this research is illustrated in Fig. 1. As shown in the Figure 1, the research begins with secondary and primary data collection, followed by data preprocessing stages such as data splitting, normalization, and image augmentation to enrich dataset variation. Subsequently, a Convolutional Neural Network (CNN) model with Inception-ResNetV2 architecture is used in the training stage. The model is then evaluated using performance metrics such as accuracy and confusion matrix. The entire research process was conducted using the Google Colab platform with T4 GPU support to accelerate model training.

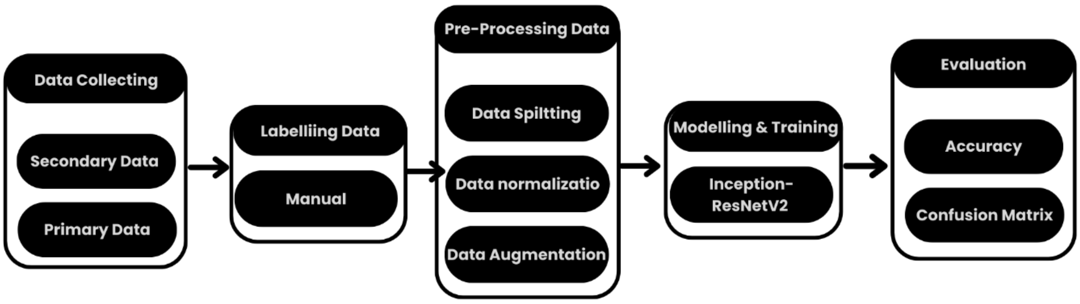

Fig. 1. Research Stage

2.1. Data Collecting

In the data collection process, this research uses two types of data: secondary data and primary data. Secondary data was obtained from the Kaggle platform titled "Road Damage". This dataset consists of 734 road images classified into three road condition categories: 335 pothole images, 200 crack images, and 199 images of roads without damage. Meanwhile, primary data was obtained from road images in the Sidoarjo Regency area, specifically on roads with regency/city road status. Primary data collection was conducted using scraping methods with the assistance of the Street View Static API accessed through Google Cloud Console. A total of 100 images were successfully collected as primary data. The following are examples of the data Fig. 2.

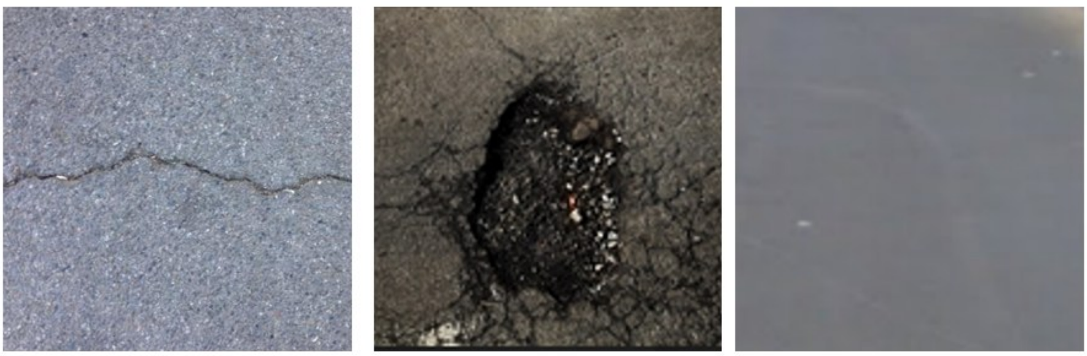

Fig. 2. Sample Image

Fig. 2 displays sample road images for each category: cracks, potholes, and undamaged. Image acquisition was conducted in the Sidoarjo Regency area, on roads that have regency/city road status. In the primary data collection process, latitude and longitude coordinates were first collected. These coordinates were then validated using the Google Roads API to ensure that the selected points were actually located on road segments. After successful validation, road images were captured based on these coordinates using the Street View Static API from Google Cloud.

2.2. Labelling Primary Data

After the image acquisition process, primary data underwent a manual labeling stage conducted by the researcher together with road construction experts. Each image was carefully examined and categorized into one of three predetermined classes: potholes, cracks, or roads without damage. This labeling process was carried out based on damage features visible in the images, with the objective of ensuring that the classification conforms to the criteria used in the secondary dataset and guaranteeing uniformity of data types between both datasets. This manual labeling is important for maintaining data consistency and ensuring accuracy and reliability in the training and evaluation processes in the subsequent modeling stage [15].

2.3. Pre-Processing Data

This stage includes various techniques, so the preprocessing methods used need to be adjusted to the characteristics of the data being used [16]. In this research, data preprocessing begins with dividing the data into three parts: training data, validation data, and testing data. All secondary data was used as training and validation data, with an allocation of 80% for training and 20% for validation. The division of primary data was based on guidelines from the Ministry of Public Works and Public Housing (PUPR), which states that the training dataset must include at least 10% of data originating from the studied area for each class or category [17]. Therefore, part of the primary data was included as additional training data. In this case, the researcher added 28 images as additional training data, equivalent to 10% of the total secondary training data.

However, for the cracked road and undamaged road categories, the researcher added more than 10% primary data because the amount of training data from secondary data for both categories was less compared to the pothole category. This was done to prevent overfitting due to imbalanced data quantities between categories [18]. Although the additional amount has not fully matched the amount of data in the most numerous category, this addition is expected to help reduce the risk of overfitting. The primary data not used as additional training data was allocated as testing data. This dataset division is a crucial stage to ensure that the model can effectively generalize to data that has never been seen before [19]. The detailed data distribution is presented in Table 1.

Table 1. Data Distribution

Category	Training Data	Data Validasi	Additional Training Data	Testing Data
Potholes	284	51	28	6
Cracks	160	40	65	17
Undamaged Roads	159	40	70	62
Total	603	131	163	85

Table 1 shows differences in the amount of data in each category. This is due to the limited amount of data in the secondary dataset obtained from Kaggle, as well as the primary data collected. Subsequently, the image data underwent a normalization process by dividing each pixel value by the maximum value of 255. This step converts pixel values to a range of 0 to 1, allowing the model to learn faster and more stably during the training process. After normalization, the image data underwent an augmentation stage to increase variation in the dataset. Augmentation was performed on training data by applying various transformations, such as random rotation up to 20 degrees, vertical and horizontal shifts of 20%, zoom in and zoom out up to 20%, horizontal flipping, and filling empty areas using the nearest pixel values.

In the final preprocessing stage, data was divided into batches of size 64, which are groups of data processed simultaneously in one training iteration. The use of this batch size helps accelerate the training process and produces more stable model weight updates. In CNN, the preprocessing stage is relatively fewer compared to other algorithms, because CNN has the ability to automatically optimize filters (kernels) during the training process [20].

2.4. Modelling & Training Model

In the process of developing a road damage level classification model, a Convolutional Neural Network (CNN) architecture based on Inception-ResNetV2 with a transfer learning approach was used. This approach aims to improve training efficiency by utilizing weights that have been previously trained on the ImageNet dataset [21]. Inception-ResNetV2 is a hybrid architecture combining two popular CNN models: Inception and ResNet. This architecture combines the advantages of Inception modules—which process input data at various scales through several parallel convolution operations—with residual connections from ResNet that enable efficient training of deeper networks. Residual connections or skip connections allow input from one layer to be passed directly to deeper layers, helping to overcome the vanishing gradient problem in complex networks [22].

Based on Fig. 3, the Inception-ResNetV2 architecture consists of several main components that work sequentially. The process begins with the stem section that receives input in the form of 299x299 pixel images with three color channels (RGB), then extracts basic features using 3×3 convolution and pooling operations. After that, the model passes through three main types of modules: five blocks of Inception-ResNet-A that use small kernels (1×1 and 3×3) to extract local features, ten blocks of Inception-ResNet-B that use combinations of large kernels (1×7 and 7×1) to capture broader spatial patterns, and five blocks of Inception-ResNet-C that return to using small kernels to detect detailed features. Between these three modules are reduction modules (Reduction-A and Reduction-B) that function to reduce the spatial dimensions of feature maps while maintaining important information. After passing through all blocks, the final feature maps are processed with average pooling, followed by dropout with a keep probability of 0.8, and concluded with a softmax layer for 1000-class classification, as in the original architecture [23].

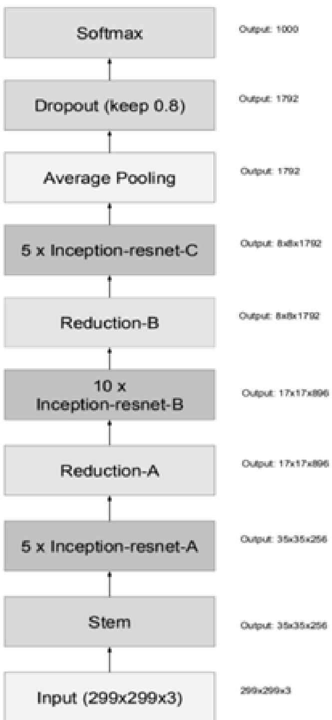

Fig. 3. Architecture Inception Resnet-V2

In this research, the Inception-ResNetV2 model was used as a feature extractor by freezing the weights from ImageNet training. The final layer of the model was then modified to suit the needs of road damage level classification. This modification was done by replacing the fully connected layer in the original architecture with several new layers: Global Average Pooling, followed by a Dense layer with ReLU activation function, Batch Normalization, Dropout with a rate of 0.5, and concluded with a Dense layer with softmax activation function that adjusts to the number of classes in the road damage data.

The model was trained using training and validation data that had been prepared previously. The training process was conducted for 20 epochs using the Adam optimizer, categorical crossentropy loss function, learning rate of 0.000001, and accuracy metric as a performance benchmark. The Adam optimizer was chosen because its results showed its superiority in most experiments, particularly in computer vision classification tasks, where it achieved the highest accuracy and fastest convergence [24]. To optimize the training process, two callback functions were used: EarlyStopping with a patience parameter of 5 to automatically stop training if validation accuracy does not improve for five consecutive epochs, and ModelCheckpoint which functions to save the model with the best performance during the training process.

2.5. Evaluation

Model evaluation was conducted in two main stages: during the training process and after training using test data from the Sidoarjo area. During the training stage, model performance was monitored through training and validation data accuracy metrics. This monitoring aims to evaluate the model's ability to learn data patterns and detect symptoms of overfitting or underfitting. Accuracy values were recorded at each epoch and visualized to analyze the overall learning trends of the model.

After training was completed, the model was further evaluated using test data to measure the performance of road damage level classification. Evaluation was conducted using accuracy metrics, confusion matrix, as well as additional metrics such as precision, recall, and F1-score. Accuracy provides a general overview of the proportion of correct predictions to the overall test data. The confusion matrix, as shown in Figure 4, illustrates the distribution of correct and incorrect predictions for each class, helping to identify error patterns that occur. Precision measures the proportion of correct positive predictions, recall assesses the model's ability to detect all positive cases, and F1-score combines both into one balanced metric, which is very useful for datasets with imbalanced class distributions [25].

As the basis for calculating these metrics, the following are the formulas for accuracy and the confusion matrix used in the model evaluation process.

$$Precision = TP / TP + FP \tag{1}$$

$$Recall = TP / TP + FN \tag{2}$$

$$F1 - Score = 2 \bullet (Recall \bullet Precision) / Recall + Precision \tag{3}$$

$$Accuracy = TP + TN / TP + FP + FN + TN \tag{4}$$

In these formulas, TP (True Positive) is the number of correct positive predictions, FP (False Positive) is incorrect positive predictions, FN (False Negative) is positive cases incorrectly predicted as negative, and TN (True Negative) is the number of correct negative predictions. These four metrics provide a comprehensive evaluation of classification model performance, both generally and per class [16].

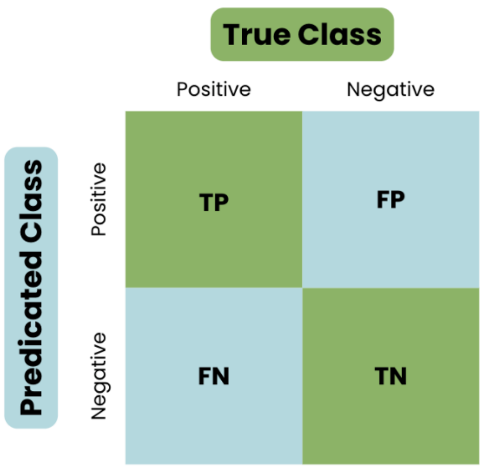

Fig. 4. Confusion Matrix

3. Results and Discussion

The collected data was then divided into training and validation data, and stored in structured folders to facilitate the process in subsequent modeling stages. During the training stage, this data was used to train the classification model using the previously prepared InceptionResNetV2 architecture. Table 2 shows the development of model accuracy during the training process. Based on Table 2, the training of the Inception ResNet V2 model showed significant performance improvements in the early epochs. Training accuracy increased from 73.90% in the first epoch to 91.66% in the 3rd epoch, while validation accuracy also rose from 34.10% to 64.16%. This indicates that the initial learning process was running effectively. From epochs 4 to 10, training accuracy remained stable above 95%, and validation accuracy continued to increase until reaching 94.22%, indicating good generalization capability of the model. After epoch 10, model performance began to stagnate. The highest validation accuracy was achieved at epoch 15 at 95.95%, then tended not to experience significant improvement and slightly decreased in subsequent epochs. This indicates that the model had reached its optimal point, and the potential for overfitting began to emerge.

The early stopping function used was applied appropriately, detecting that no improvement in validation accuracy occurred for five epochs after epoch 15. With a patience parameter set at 5, training was automatically stopped to prevent training from continuing without performance improvement. This strategy was implemented to avoid the risk of overfitting and maintain training process efficiency [26]. Additionally, the ModelCheckpoint function was also applied to save the best model weights during the training process. This mechanism automatically saves the model every time validation accuracy improves. Thus, although training was stopped at epoch 20, the model used for final testing was the model from epoch 15, when the highest validation accuracy (95.95%) was achieved. This strategy ensures that the selected model is the most optimal version in terms of generalization capability to test data. During the training period, the best model was recorded at epoch 15 with training accuracy of 97.42% and validation accuracy of 95.95%.

Table 2. Accuracy Training

Epoch	Training Accuracy	Validation Accuracy
1	0.7390	0.3410
2	0.8799	0.3873
3	0.9166	0.6416
4	0.9530	0.7110
5	0.9608	0.7688
6	0.9387	0.8382
7	0.9560	0.8960
8	0.9559	0.8844
9	0.9470	0.8671
10	0.9655	0.9422
11	0.9677	0.9422
12	0.9507	0.9075
13	0.9492	0.9480
14	0.9652	0.9538
15	0.9742	0.9595
16	0.9550	0.9306
17	0.9756	0.9538
18	0.9721	0.9595
19	0.9671	0.9595
20	0.9447	0.9538

Fig. 5 shows the training and validation accuracy graph that demonstrates stable and consistent learning patterns throughout the training process. Both curves experienced gradual improvement until reaching a convergence point, reflecting that the model was not only able to learn from training data but also had good generalization capability to validation data. There were no striking differences between the two curves, indicating that the training process proceeded in a balanced manner without significant overfitting indications. The stability of the curves in the final phase of training reinforces confidence that the model had achieved optimal performance, and the training process had run effectively until the automatic stopping point. After the model was saved at epoch 15, it was used to make predictions on previously separated test data. The purpose of this testing was to evaluate the model's generalization capability to new data that had never been encountered during the training or validation process. Evaluation using test data was conducted by calculating several classification metrics such as accuracy, precision, recall, and F1-score, as well as displaying a confusion matrix to see the distribution of model predictions for each class.

The following graph illustrates the changes in training process accuracy. Based on the graph, it is evident that at the beginning of training, particularly at the 3rd epoch, there was a decrease in accuracy and an increase in loss. However, subsequently, the model demonstrated significant performance improvement until reaching its optimal performance at the 11th epoch. This indicates that the model successfully learned the patterns from the data optimally before eventually experiencing stagnation.

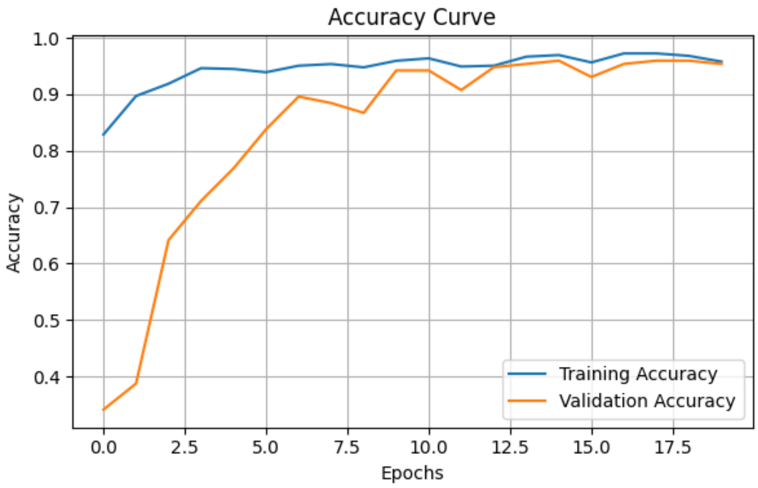

Fig. 5. Accuracy Training Curve

The evaluation results are displayed in Table 3. Model evaluation on test data shows that the model has fairly good overall classification performance, with accuracy reaching 83%. Among the three tested classes, the undamaged road class showed the best performance, with precision of 0.92, recall of 0.87, and f1-score of 0.89. This indicates that the model is highly reliable in recognizing road conditions that have not experienced damage. Meanwhile, the pothole road class also showed fairly good performance, especially in the recall aspect which reached 0.87, meaning that most pothole road cases were successfully recognized by the model. However, its precision was slightly lower (0.72), indicating that there were still some false positive predictions for this class.

In the cracked road class, model performance was relatively lower compared to the other two classes, with precision and recall both valued at 0.65. The f1-score value which was also 0.65 shows that the model was not yet optimal in distinguishing cracked road conditions from other classes. This could be caused by visual similarities between cracks and other road surfaces, or by imbalanced training data distribution. Overall, the model demonstrates satisfactory classification capability, especially in detecting undamaged and potholed road conditions, but still has room for improvement in crack detection. The training strategies that have been implemented have proven capable of producing a model with fairly good generalization to real test data.

Table 3. Evaluation Model Testing

Label	Precision	Recall	F1-score	Support
Potholes	0.72	0.87	0.79	15
Cracks	0.65	0.65	0.65	17
Undamaged Roads	0.92	0.87	0.89	62
Accuracy			0.83	94

Additionally, Fig. 6 presents the confusion matrix that displays the number of correct and incorrect predictions for each class. Based on the confusion matrix, the model demonstrates reasonably good performance in classifying road images into three classes: pothole, crack, and undamaged road.

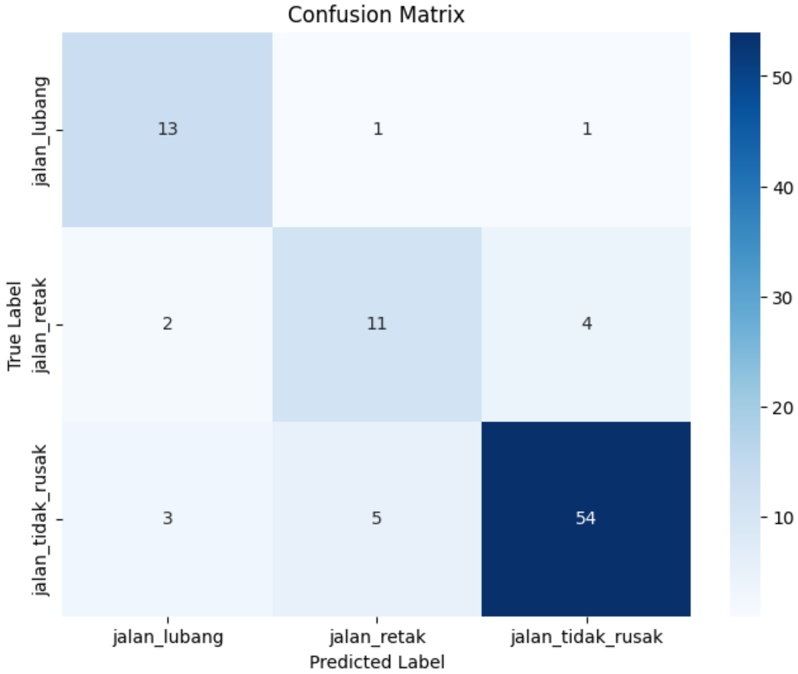

Fig. 6. Confusion Matrix Evaluation

In the pothole class, 13 out of 15 data points were correctly classified, while one data point each was misclassified as pothole and undamaged road. These errors are likely attributed to the small size of the potholes, resulting in visual features that are not sufficiently prominent for the model to recognize as damage, and tend to resemble crack textures or intact road surfaces. For instance, as shown in Fig. 7, there is a small pothole in the upper left corner. Additionally, the image contains tire stains that resemble cracks, causing the model to tend toward predicting cracked roads.

Fig. 7. Sampel Data Testing of Pothole

For the pothole class, the model's performance was not as robust as the other two classes, where only 11 out of 17 data points were correctly classified, while two data points were misclassified as pothole and four others as undamaged road. The misclassification into the undamaged road class is presumed to occur due to very thin or faint cracks, causing the model to perceive the road surface as still being in good condition. For example, Fig. 8 exhibits thin cracks that closely resemble the undamaged road class, leading to incorrect predictions by the model.

Fig. 8. Sampel Data Testing of Crack

Meanwhile, the model demonstrated excellent performance on the undamaged road class, with 54 out of 62 data points correctly classified. Only three data points were classified as pothole and five as pothole, indicating a relatively low error rate. Overall, the model achieved satisfactory accuracy on the test data with tolerable misclassification errors. Nevertheless, accuracy improvement is still required, particularly for the pothole class, with emphasis on enhancing the model's capability to distinguish the visual characteristics of fine cracks from normal road surfaces.

Overall, the model exhibits reasonably good classification performance for road conditions in urban areas such as Sidoarjo. The test results demonstrate the model's reliable generalization capability, particularly for detecting undamaged and potholed road conditions. However, this study has limitations regarding the quantity and diversity of data utilized. The secondary dataset from Kaggle is relatively limited in road condition variations, while the primary dataset collected independently only encompasses regency/city-level roads. This limitation prevents comprehensive testing of the model against road conditions beyond these categories, such as provincial, national, or rural roads. Therefore, dataset expansion and balancing constitute crucial steps in further development to enable better generalization across various road environment types.

4. Conclusion

This study successfully implemented the Inception ResNet-V2 architecture to classify road damage levels in Sidoarjo Regency into three categories: undamaged roads, cracked roads, and potholes. The model was trained using a combination of secondary data from Kaggle and primary data collected via Google Street View API. Training results showed a peak validation accuracy of 95.95% at the 15th epoch, and a test accuracy of 83%. The best performance was observed in the undamaged road class with an F1-score of 0.89, while the lowest performance was found in the cracked road class (F1-score 0.65). Additionally, the model struggled to detect small potholes due to the dominance of large pothole images in the training set, resulting in an F1-score of 0.79 for that class. Although the Inception ResNet-V2 architecture proved effective in extracting complex image features and showed stable training behavior, this study still faced significant limitations. One major issue was class imbalance, which negatively affected detection performance—particularly for the cracked road class, which functioned as a minority class and was frequently misclassified.

Future research should consider additional strategies to address this issue, such as cost-sensitive learning, synthetic oversampling (e.g., SMOTE), or adjusting the loss function weights. Moreover, exploring enhanced model architectures such as a hypothetical Inception-ResNet-V3—which integrates deeper multi-scale convolution modules, improved residual connections, and attention mechanisms—could lead to better sensitivity in detecting subtle or visually ambiguous damage features. The proposed model shows strong potential for integration into image-based road damage reporting systems for public use, which could significantly improve the efficiency of infrastructure monitoring and maintenance. Beyond its practical implications, this study also contributes theoretically to the development of image classification systems for public infrastructure, and demonstrates the applicability of modern deep learning architectures within the context of limited and complex local datasets in Indonesia.

References

[1] I. N. Sannikova, E. A. Prikhodko, and A. A. Muhitdinov, “Assessment of the universities impact on global competitiveness based on rankings,” In *E3S web of conferences*, vol. 296, p. 08009, 2021.

[2] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, “Road damage detection using deep neural networks with images captured through a smartphone,” *arXiv preprint arXiv:1801.09454*, 2018.

[3] Y. He, B. Liu, C. Xu, and D. Wu, “Literature Review on Public Transport and Land Use: Based on CiteSpace Statistical Analysis,” *Land*, vol. 14, no. 5, p. 1096, 2025.

[4] A. Alfarrarjeh, D. Trivedi, S. H. Kim and C. Shahabi, "A Deep Learning Approach for Road Damage Detection from Smartphone Images," *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5201-5204, 2018.

[5] H. A. Geziary, H. El Sayed, S. J. Hussain, and H. I. Sakr, “Road safety: the potholes of neglect,” *East Mediterr Health J*, vol. 1, p. 252-55, 2004.

[6] A. A. Mohammed, K. Ambak, A. M. Mosa, and D. Syamsunur, “A review of traffic accidents and related practices worldwide,” *The Open Transportation Journal*, vol. 13, no. 1, 2019.

[7] H. Etehad, S. H. Yousefzadeh-Chabok, A. Davoudi-Kiakalaye, D. A. Moghadam, H. Hemati, and Z. Mohtasham-Amiri, “Impact of road traffic accidents on the elderly,” *Archives of gerontology and geriatrics*, vol. 61, no. 3, pp. 489-493, 2015.

[8] S. S. Liu, M. F. A. Arifin, W. T. Chen, and Y. H. Huang, “Emergency repair scheduling model for road network integrating rescheduling feature,” *Applied Sciences*, vol. 11, no. 4, p. 1447, 2021.

[9] S. Agarwal, N. Mittal, and A. Sureka, “Potholes and bad road conditions: mining twitter to extract information on killer roads,” In *Proceedings of the ACM India joint international conference on data science and management of data*, pp. 67-77, 2018.

[10] H. U. Sajid, R. Kiran, X. Qi, D. S. Bajwa, and D. Battocchi, “Employing corn derived products to reduce the corrosivity of pavement deicing materials,” *Construction and Building Materials*, vol. 263, p. 120662, 2020.

[11] I. G. S. M. Diyasa, A. D. Alhajir, A. M. Hakim and M. F. Rohman, "Reverse Image Search Analysis Based on Pre-Trained Convolutional Neural Network Model," *2020 6th Information Technology International Seminar (ITIS)*, pp. 1-6, 2020.

[12] Y. Zheng, H. Wang, and Y. Hao, “Mobile application for monitoring body temperature from facial images using convolutional neural network and support vector machine,” In *Mobile Multimedia/Image Processing, Security, and Applications 2020*, vol. 11399, pp. 53-63, 2020.

[13] J. Wang, X. He, S. Faming, G. Lu, H. Cong and Q. Jiang, "A Real-Time Bridge Crack Detection Method Based on an Improved Inception-Resnet-v2 Structure," in *IEEE Access*, vol. 9, pp. 93209-93223, 2021.

[14] A. Riyandi, T. Widodo, and S. Uyun, “Classification of Damaged Road Images Using the Convolutional Neural Network Method,” *Telematika: Jurnal Informatika dan Teknologi Informasi*, vol. 19, no. 2, pp. 147-158, 2022.

[15] M. Maniat, C. V. Camp, and A. R. Kashani, “Deep learning-based visual crack detection using Google Street View images,” *Neural computing and applications*, vol. 33, no. 21, pp. 14565-14582, 2021.

[16] I. G. S. M. Diyasa, A. Fauzi, A. Setiawan, M. Idhom, R. R. Wahid and A. D. Alhajir, "Pre-trained Deep Convolutional Neural Network for Detecting Malaria on the Human Blood Smear Images," *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 235-240, 2021.

[17] E. Ranyal, A. Sadhu, and K. Jain, “Road condition monitoring using smart sensing and artificial intelligence: A review,” *Sensors*, vol. 22, no. 8, p. 3044, 2022.

[18] M. A. Lones, “How to avoid machine learning pitfalls: a guide for academic researchers,” *arXiv preprint arXiv:2108.02497*, 2021.

[19] I. G. S. M. Diyasa, D. A. Prasetya, H. A. C. Kuswardhani, and C. Halim, “Detection of Abnormal Human Sperm Morphology Using Support Vector Machine (SVM) Classification,” *Information Technology International Journal*, vol. 2, no. 2, pp. 57-63, 2024.

[20] W. S. J. Saputra, E. Y. Puspaningrum, W. F. Syahputra, A. P. Sari, Y. V. Via and M. Idhom, "Car Classification Based on Image Using Transfer Learning Convolutional Neural Network," *2022 IEEE 8th Information Technology International Seminar (ITIS)*, pp. 324-327, 2022.

[21] S. Sangsawad and S. Jintawatsakoon, "Quality Assessment of Cocoa Bean Using Regression Convolutional Neural Network," *2023 27th International Computer Science and Engineering Conference (ICSEC)*, pp. 1-6, 2023.

[22] C. Peng, Y. Liu, X. Yuan, and Q. Chen, “Research of image recognition method based on enhanced inception-ResNet-V2,” *Multimedia Tools and Applications*, vol. 81, no. 24, pp. 34345-34365, 2022.

[23] S. Riyadi, A. S. Salsabila and A. R. Puspita Dewi, "Effectiveness of Inception V3and Mo bileN et V2Models in Classifying Bird Species Based on Physical Characteristics," *2024 IEEE 14th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 265-269, 2024.

[24] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, “The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study,” *Multimedia Tools and Applications*, vol. 82, no. 11, pp. 16591-16633, 2023.

[25] S. Farhadpour, T. A. Warner, and A. E. Maxwell, “Selecting and interpreting multiclass loss and accuracy assessment metrics for classifications with class imbalance: Guidance and best practices,” *Remote Sensing*, vol. 16, no. 3, p. 533, 2024.

[26] B. M. Hussein and S. M. Shareef, “An empirical study on the correlation between early stopping patience and epochs in deep learning,” In *ITM Web of Conferences*, vol. 64, p. 01003, 2024.