Road Surface Crack Detection Using Deep Learning in Smart Cities

Fatmanur Polat[1], Muhammed Emre Çolak[2], İrfan Kılıç [3]

[1,3]Department of Software Engineering, Faculty of Engineering, Fırat University, Elazığ, Türkiye.
[2]Department of Artificial Intelligence and Data Science Engineering, Faculty of Engineering, Fırat University, Elazığ, Türkiye.
[1]fatmannurpolat@gmail.com, [2]memrecolak@firat.edu.tr, [3]irfankilic@firat.edu.tr
[1](ORCID: 0009-0005-5374-9933), [2](ORCID: 0000-0003-2317-2385),[3](ORCID: 0000-0001-5079-2825)

Abstract
This study aims to develop an automatic and high-accuracy system for detecting cracks on asphalt road surfaces as part of smart city applications. The proposed method employs deep learning techniques, specifically a MobileNetV2-based convolutional neural network, trained on a large, balanced dataset of 40,000 images. Preprocessing steps such as resizing, normalization, and data augmentation were applied to improve model generalization. The system achieved 99.78% accuracy on the training set and 99.73% on the test set, demonstrating its capability for real-time operation on low-cost devices. The results indicate the proposed approach is a feasible and practical solution for urban infrastructure maintenance planning, with potential integration into mobile and embedded systems for real-world deployment.

Keywords: asphalt crack, deep learning, image processing, smart city, road maintenance

1. Introduction

Rapid urbanization has increased the load on transportation infrastructure, causing various types of road surface degradation. Cracks on asphalt surfaces pose safety hazards and lead to higher maintenance costs. Traditional detection methods are labor-intensive, time-consuming, and prone to human error. Smart city technologies aim to address these challenges by integrating automated monitoring systems and data-driven decision-making into urban management. Early, accurate crack detection enables more efficient maintenance planning, offering both economic and environmental benefits. Recent studies highlight the success of deep learning approaches, particularly convolutional neural networks (CNN), YOLO, EfficientNet, and MobileNet models, achieving high accuracy (99%) while being lightweight enough for real-time and mobile applications [1], [2], [3], [4], [5].

This paper presents a MobileNetV2-based deep learning system trained on the public “Surface Crack Detection” dataset from Kaggle. The model is optimized for low-cost device integration and evaluated with metrics such as accuracy, F1-score, and Dice coefficient to assess its suitability for smart city applications.

1.1. Related Works

The timely and accurate detection of cracks on asphalt surfaces is of great importance for ensuring the sustainability of urban transportation infrastructure, increasing road safety, and planning maintenance and repair processes more effectively [6], [7]. Traditional crack detection methods generally rely on field observations and manual measurements, resulting in high labor requirements, time loss, and vulnerability to human error [8]. Especially in regions where urbanization is rapidly increasing, the applicability of these methods has decreased, and the need for automatic, fast, and reliable solutions has become evident.In this context, in recent years, artificial intelligence (AI)-based methods have offered groundbreaking advances in the detection and classification of asphalt cracks, particularly through image processing, machine learning (ML), and deep learning (DL) techniques [9], [10]. Deep learning models are frequently preferred in this field due to their ability to perform automatic feature extraction, generalization learning from large datasets, and high accuracy rates [11]. Convolutional neural network (CNN)-based architectures, such as EfficientNet, YOLOv8s, Faster R-CNN, and ResNet; It attracts attention with both its real-time detection capacity and mobile compatibility [12], [13]. For example, Chen et al. [14] achieved over 94% accuracy by analyzing preprocessed crack images with an EfficientNet-based model; the study reported that data augmentation strategies significantly increased model performance. Similarly, the Faster R-CNN-based system proposed by Zou et al. [15] successfully distinguished various crack types (e.g., block, linear, alligator cracks) and provided 30% faster and more accurate results than traditional methods. An improved lightweight semantic segmentation model based on

[1]Corresponding author

BiSeNetv2 detected road surface cracks with high accuracy by effectively combining low- and high-level features, and achieved an improvement of up to 10.14% in F1 score compared to conventional models [16]. Another study proposed CP-YOLOX and SViT based deep learning models for the detection of structural deteriorations (cracks, voids, weak bonds, etc.) on the road surface using 3D GPR horizontal plane images [17]. In particular, methods that use manually extracted features, such as HOG and LBP, in conjunction with classifiers, such as SVM and KNN; It has achieved high success with low parameter requirements. To overcome the limitations of traditional laser scanning-based methods, Jin et al. combined the oriented gradient histogram with the improved watershed algorithm to effectively detect road cracks based on direction and edge features on range images [18]. Li et al. aimed to increase both accuracy (95.2%) and efficiency (95.8% recall) in visual safety inspections of dams by proposing a method that enables automatic, rapid, and high-accuracy detection of cracks on the dam surface using the YOLOv8 algorithm on high-resolution images collected by an UAV [19]. Hu et al. proposed MobiLiteNet, a lightweight DL-based road nuisance detection method, to improve road infrastructure management and intelligent transportation systems. This method combines the techniques of channel attention, KD, and model compression to achieve high accuracy and low computational cost. It can be integrated into mobile devices to overcome traditional inspection challenges [20].

This study aims to develop a crack detection system that simultaneously meets the key criteria of high accuracy, low hardware requirements, and real-time performance, within the framework of the literature summarized above. To this end, an optimized model is proposed by integrating the deep learning-based MobileNetV2 architecture, data augmentation strategies, and traditional classifiers. In terms of accuracy and field applicability, the proposed model is competitive with the current approaches in the literature.

2. Material and Methods

2.1. Dataset

We used the Surface Crack Detection dataset shared on Kaggle [21], containing 40,000 color images (227×227 pixel) evenly split between positive (crack) and negative (no crack) classes. The data was divided into training (80%), validation (10%), and test (10%) sets.

- Positive: Images containing cracks (20,000 ima
- Negative: Crack-free images (20.000 images)

The balanced structure of the dataset ensures equal distribution across classes. The images were obtained from various surface types (asphalt, concrete, etc.) and represent a single surface segment. The dataset was split into training, validation, and test segments at a ratio of 80% / 10% / 10%. Accordingly, the distribution is as shown in Table 1.

Table 1. Class distribution of road crack detection dataset

Dataset	Number of images
Training Set	~32.000
Validation Set	~4.000
Test Set	~4.000

The images were loaded by opening the .zip file in Google Colab. The directory structure was reorganized to suit the ImageDataGenerator function.

2.2. Preprocessing and Data Augmentation

Various preprocessing and data augmentation techniques were applied to the images to enable the model to generalize better:

- Resizing to 224×224 pixels
- Normalization to [0–1]
- Augmentation techniques:
  - Horizontal flipping
  - Random rotation (±20°)
  - Zoom (20%)
  - Brightness adjustments

These were applied via ImageDataGenerator in Keras, with only resizing and normalization used for validation and test sets.

2.3. The Proposed Deep Learning Model

We adopted a transfer learning approach using MobileNetV2 [22] pre-trained on ImageNet. The base network was frozen, and only the final layers were retrained for binary classification (crack/no crack) using a sigmoid activation. The architecture uses "inverted residual blocks" with expansion, depthwise convolution, and projection layers to reduce computational cost while maintaining accuracy. Dice Loss was used as the primary loss function to handle class imbalance effectively.

A transfer learning approach was adopted in this study, and the MobileNetV2 architecture was chosen as the base model. This architecture, which was previously trained on ImageNet, was chosen for its low parameter count and high accuracy. MobileNetV2 is a lightweight and fast CNN architecture suitable for integration into mobile devices. A key feature of the model is its bottleneck blocks, which minimize information loss by preserving features from input to output.

During model training, the body of the pre-trained MobileNetV2 network was left frozen, and only the final layers were retrained. This allows the model to learn the required outputs for the new task and progress faster using the previously trained weights' information (Figure 1).

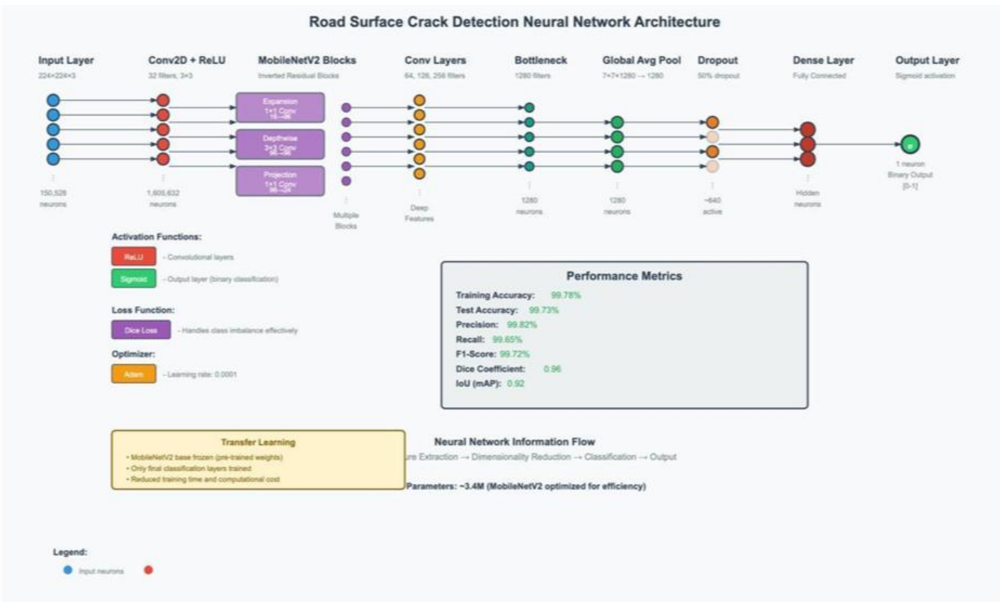


Figure 1. The proposed deep learning model architect

The model's output performs binary classification using a sigmoid activation function that generates values between 0 and 1. This structure allows the model to probabilistically distinguish between the presence and absence of cracks. The model structure is schematically illustrated in Figure 1. This structure reduces both training time and memory usage, allowing the model to be used in real-time applications. An image classification-based approach was used to detect asphalt cracks in this study. This study aims to design an optimized and lightweight DL architecture that determines whether the input surface images contain cracks.

Training Parameters;

- Optimizer: Adam
- Learning rate: 0.0001
- Batch size: 32
- Epochs: 50 (with EarlyStopping)
- Callbacks: ReduceLROnPlateau, EarlyStopping
- Dropout: 50%
- Environment: Google Colab with NVIDIA Tesla P100 GPU

The following steps were taken to combat overfitting:

- ReduceLROnPlateau: The learning rate was reduced when the validation loss became stationary.
- EarlyStopping: Training was stopped when validation performance did not improve.
- Dropout: 50% was applied.
- Data augmentation: The model was allowed to see various variations.

3. Results and Discussion

After our model was trained according to the applications and libraries given in Table 2, the performance results given in Table 3 were obtained.

Table 2. Hardware and software components used in training environment

Component	**Version**
Python	3.8
TensorFlow	2.x
Keras	Integrated (TF 2.x)

The Dice score and the intersection over union (IoU) or mean average precision (mAP) value were also calculated. The Dice score and mAP value yielded state-of-the-art results.

- Training accuracy: **99.78%**
- Test accuracy: **99.73%**
- Dice coefficient: **0.96**
- IoU (mAP): **0.92**

Precision, recall, and F1-score were also above 99%, demonstrating strong performance.

Table 3. Model training performance metrics

	Precision	Recall	F1-Score	Accuracy
Train	%99.87	%99.70	%99.78	%99.78
Test	%99.82	%99.65	%99.72	%99.73

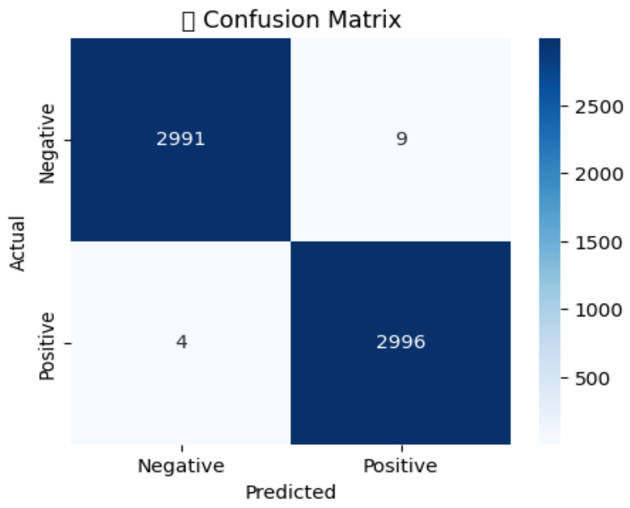


Figure 2. Confusion Matrix

The confusion matrix showed balanced and high-quality predictions across classes (Figure 2). However, false negatives were observed primarily in images with very small or low-contrast cracks.

Accuracy, precision, recall and F1-score were calculated according to equation (3-6).

$$ccuracy = \frac{TP+TN}{TP+FN+TN+FP} \quad (3)$$

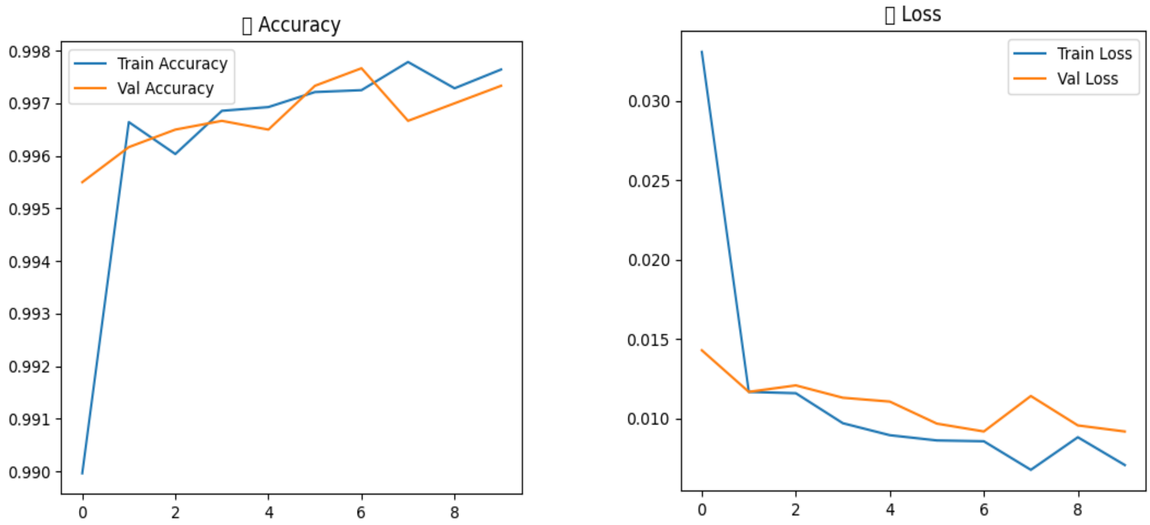
$$recall = \frac{TP}{TP+FP} \quad (4)$$

$$precision = \frac{TP}{TP+FN} \quad (5)$$

$$F1 - score = \frac{recall\ x\ precision}{recall + precision} \quad (6)$$

TP: True-Positive, FN= False-Negative, TN: True-Negative, FP: False-Positive

Figure 3 shows the accuracy and loss graphs for training and testing.



Samples x 1000
a) Accuracy

Sample x 1000
b) Loss

Figure 3. Accuracy and loss plots a) Accuracy b) Loss

The accuracy graph shows the model's correct classification rate on the training and validation data at the end of each epoch. The loss graph visualizes the change in the loss values during the training process, representing the model's error. Figure 4 presents the predictions of the proposed model using the corresponding sample images. Figure 5 shows the sample images that the model incorrectly predicted.

Based on these values, the following evaluation metrics were calculated:

- Accuracy: The proportion of all samples that the model correctly predicted.
- Precision: The fraction of samples predicted as belonging to the positive class that were actually positive.
- Recall: The fraction of true positives that were correctly predicted.
- F1-Score: The harmonic mean of Precision and Recall is important for imbalanced classes.
- Dice Coefficient: A similarity measure that shows the extent to which the true and predicted classes overlap.
- IoU: The ratio of the intersection of the predicted and true crack regions to the union.

According to the results, the model generally demonstrated high performance on the test set. However, the FN rate increased on some surfaces, particularly those with small cracks. This suggests that some fine cracks may be overlooked due to their blending with the surface's natural patterns. Figure 4 shows the images that our model predicted correctly.

Figure 4. Examples correctly classified by the proposed model

Figure 5 shows the images in which the model incorrectly predicted that the objects were outside the road. The accuracy of the model was evaluated visually using examples of correct and incorrect predictions. Correct predictions reveal the model’s strengths, whereas incorrect predictions reveal areas for improvement. Figure 6 shows where the model focused on its predictions by extracting heat memories using the Grad-CAM method.

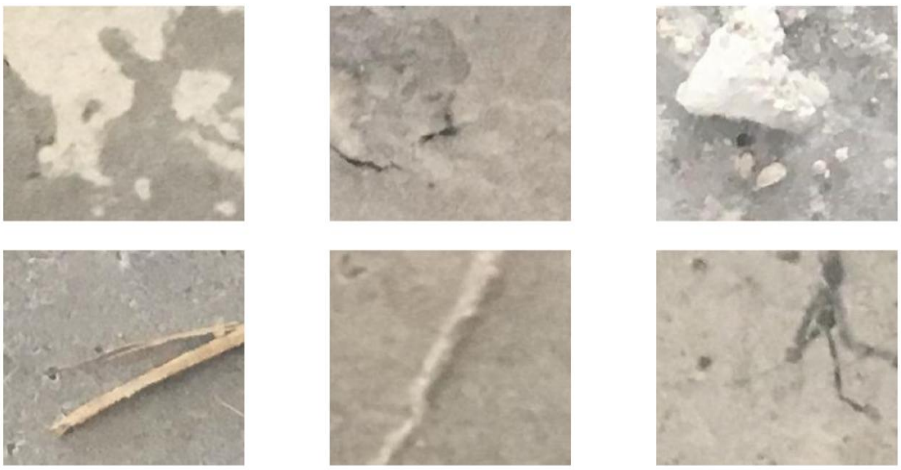


Figure 5. Examples incorrectly classified by the proposed model

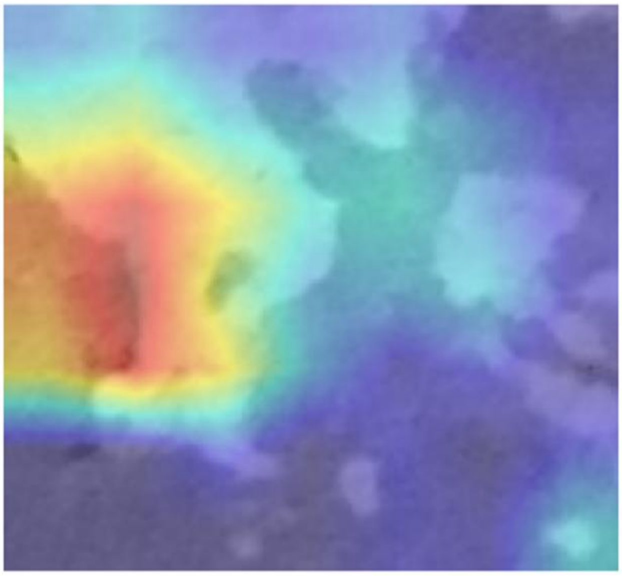


Figure 6. Heat map created with the Grad-CAM method

Grad-CAM displays the regions in the image to which the model pays the most attention during classification as a heat map. This visual aids in interpreting the decision-making process of the model and understanding the areas it focuses on. Figure 7 shows the number of broken/unbroken data points in the training dataset after model training.

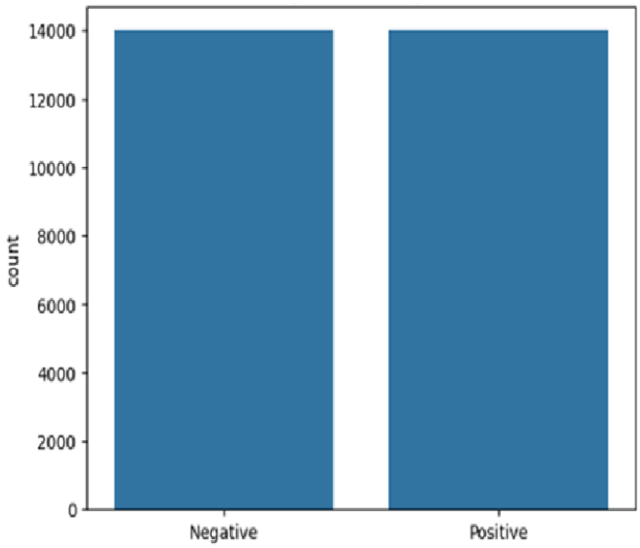


Figure 7. Class distribution plot in the training dataset

Grad-CAM displays the regions in the image to which the model pays the most attention during classification as a heat map. This visual aids in interpreting the decision-making process of the model and understanding the areas it focuses on. Figure 7 shows the number of broken/unbroken data points in the training dataset after model training.

Correctly classified examples confirmed robust learning. Misclassifications often included images with non-road textures or very subtle cracks. Grad-CAM heatmaps revealed the model focused on crack regions, validating its interpretability. Slight overfitting was detected, with training performance slightly surpassing validation. Strategies to mitigate this included aggressive data augmentation, increased dropout, EarlyStopping, testing with simpler architectures, and cross-validation.

4. Conclusions

This study demonstrates the feasibility of using MobileNetV2-based deep learning models for automated, high-accuracy asphalt crack detection in smart city contexts. The system achieved over 99% accuracy with low computational requirements, making it suitable for mobile and embedded deployment. Future work should focus on expanding datasets to include diverse environmental conditions, implementing multi-class classification to identify crack types, enabling mobile deployment using TensorFlow Lite or ONNX, incorporating continual learning and online updates, collaborating with municipalities for field testing, and openly sharing models and code for community advancement.

Through extensive experimental testing, the deep learning-based asphalt crack detection model developed in this study has demonstrated high accuracy and reliability. Built on the MobileNetV2 architecture, the model achieved an accuracy of 99.70% and above, achieved an F1 score, and demonstrated superior performance with a Dice coefficient (0.96) in both pixel-based segmentation and image-based classification tasks. These results support the practical application of the proposed method. The model successfully detected both fine and complex cracks as well as prominent and wide cracks. Effective data augmentation strategies, a transfer learning approach, and a carefully structured model architecture drive the success of this approach. Furthermore, the low hardware requirements and fast processing time of the model enhance its usability in real-time systems and enable its integration with mobile platforms.

However, some environmental challenges could negatively impact the model's performance. The detection accuracy was particularly low under environmental influences such as low-contrast asphalt surfaces, shadows, and dirt. Furthermore, the model's high adaptation to the training data introduced a tendency for overfitting, leading to performance degradation on data types it has not yet encountered. Although the dataset encompasses various surface and lighting conditions, it is insufficient to cover all real-world possibilities. This demonstrates the need for larger and more representative datasets to enhance the model's generalizability. Overall, this study presents a deep learning-based, lightweight, and mobile-friendly solution for asphalt crack detection, offering a compelling alternative to existing methods in terms of accuracy, efficiency, and practical application.

5. References

[1] R. Xie, M. Chen, M. Tao, K. Ding, and H. Chen, "Gradient and self-attention enabled convolutional neural network for crack detection in smart cities," in *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*, 2023, pp. 2110–2117. doi: 10.1109/ICPADS60453.2023.00286.

[2] Z. Topbaş, Ö. Erdaş Çiçek, and Ş. Gülcü, "DERİN ÖĞRENME İLE ASFALT ÇATLAKLARININ TESPİTİNDE VERİ ARTIRIMI VE EVRİŞİMSEL BLOK SEÇİMİNİN ETKİSİ," *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 11, no. 23, pp. 172–189, 2024, doi: 10.54365/adyumbd.1453460.

[3] L. S. Furtado, I. S. Bessa, N. de O. Gurjão, and J. B. Soares, "Integrating smart city technologies for sustainable pavement infrastructure," *Canadian Journal of Civil Engineering*, vol. 52, no. 5, pp. 569–582, 2025, doi: 10.1139/cjce-2024-0217.

[4] H. Chu *et al.*, "Deep Learning Method to Detect the Road Cracks and Potholes for Smart Cities," *Computers, Materials & Continua*, vol. 75, pp. 1863–1881, Jul. 2023, doi: 10.32604/cmc.2023.035287.

[5] M. Alkhedher, A. Alsit, M. Alhalabi, S. AlKheder, A. Gad, and M. Ghazal, "Novel pavement crack detection sensor using coordinated mobile robots," *Transp Res Part C Emerg Technol*, vol. 172, p. 105021, 2025, doi: https://doi.org/10.1016/j.trc.2025.105021.

[6] L. Zhang, F. Yang, Y. Daniel Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3708–3712. doi: 10.1109/ICIP.2016.7533052.

[7] 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*, Jul. 2018, doi: 10.48550/arXiv.1801.09454.

[8] H. Oliveira and P. Correia, "CrackIT - An Image Processing Toolbox for Crack Detection and Characterization," in *2014 IEEE International Conference on Image Processing, ICIP 2014*, Jul. 2014. doi: 10.1109/ICIP.2014.7025160.

[9] J. Xing, Y. Liu, and G. Zhang, “Improved YOLOV5-Based UAV Pavement Crack Detection,” *IEEE Sens J*, vol. PP, p. 1, Jul. 2023, doi: 10.1109/JSEN.2023.3281585.
[10] Y. Wu, S. Li, J. Li, Y. Yu, J. Li, and Y. Li, “Deep learning in crack detection: A comprehensive scientometric review,” *Journal of Infrastructure Intelligence and Resilience*, vol. 4, no. 3, p. 100144, 2025, doi: https://doi.org/10.1016/j.iintel.2025.100144.
[11] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” Jul. 2019. doi: 10.48550/arXiv.1905.11946.
[12] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” Jul. 2018, doi: 10.48550/arXiv.1804.02767.
[13] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans Pattern Anal Mach Intell*, 2017, doi: 10.1109/TPAMI.2016.2577031.
[14] X. Chen *et al.*, “Autonomous Crack Detection for Mountainous Roads Using UAV Inspection System,” *Sensors*, vol. 24, no. 14, 2024, doi: 10.3390/s24144751.
[15] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, “CrackTree: Automatic crack detection from pavement images,” *Pattern Recognit Lett*, vol. 33, no. 3, pp. 227–238, 2012, doi: https://doi.org/10.1016/j.patrec.2011.11.004.
[16] H. Yu, Y. Deng, and F. Guo, “Real-time pavement surface crack detection based on lightweight semantic segmentation model,” *Transportation Geotechnics*, vol. 48, p. 101335, 2024, doi: https://doi.org/10.1016/j.trgeo.2024.101335.
[17] J. Yang, K. Ruan, J. Gao, S. Yang, and L. Zhang, “Pavement Distress Detection Using Three-Dimension Ground Penetrating Radar and Deep Learning,” *Applied Sciences*, vol. 12, no. 11, 2022, doi: 10.3390/app12115738.
[18] H. Jin, F. Wan, and O. Ruan, “Pavement Crack Detection Fused HOG and Watershed Algorithm of Range Image,” Jul. 2018, pp. 475–488. doi: 10.1007/978-3-319-59463-7_47.
[19] X. Li *et al.*, “Dam Crack Detection Studies by UAV Based on YOLO Algorithm,” in *2023 2nd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC)*, 2023, pp. 104–108. doi: 10.1109/RAIIC59453.2023.10281120.
[20] Y. Hu, N. Chen, Y. Hou, X. Lin, B. Jing, and P. Liu, “Lightweight deep learning for real-time road distress detection on mobile devices,” *Nat Commun*, vol. 16, no. 1, p. 4212, 2025, doi: 10.1038/s41467-025-59516-5.
[21] “Surface Crack Detection.” Accessed: Jul. 12, 2025. [Online]. Available: https://www.kaggle.com/datasets/arunrk7/surface-crack-detection/data
[22] [A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *ArXiv*, vol. abs/1704.0, 2017, [Online]. Available: https://api.semanticscholar.org/CorpusID:12670695