International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0171
e ISSN: 2584-2854
Volume: 03
Issue: 04 April 2025
Page No: 1050 - 1054

Road Crack Detection Using Deep Learning

Bhargavi Padakanti[1], G. Swathi[2], Chandana Maddigatla[3], Kavya Morigadi[4], Vasu Malothu[5], Pavan Sirigadde[6]
[1,2,3]Assistant Professor – Computer Science Engineering – Data Science, CMR Engineering College, Hyderabad, Telangana, India.
[4,5,6] UG – Computer Science Engineering – Data Science, CMR Engineering College, Hyderabad, Telangana, India.
Emails ID: *bhargavi.padakanti@cmrec.ac.in[1], swathi.gajula@cmrec.ac.in[2], mchandana545@gmail.com[3], 218r1a6742@gmail.com[4], 228r1a67@gmail.com[5], 218r1a6759@gmail.com[6]*

Abstract
Reliable and meticulously maintained roadway infrastructure is essential for ensuring both ensuring safety and operational prowess in transportation; nevertheless, even the slightest surface flaws and structural shortcomings can swiftly morph into substantial safety threats and lead to increased repair costs if not addressed promptly. To address these limitations, this research introduces an automated detection framework that leverages deep learning—more specifically, Convolutional Neural Networks (CNNs)—to systematically identify and categorize road surface anomalies. The framework is constructed utilizing a heterogeneous dataset of road imagery, which equips it with the capability to differentiate cracks based on their severity and classification. Its efficacy is further enhanced through preprocessing techniques such as image augmentation and normalization. Designed for real-time implementation, the system can be operationalized on mobile and drone platforms to facilitate comprehensive monitoring efforts. Experimental findings indicate notable advancements in detection precision and a decrease in false positive rates, thereby bolstering more efficient maintenance strategies and fostering sustainable infrastructure management. Future investigations may integrate additional sensor modalities, including LiDAR and thermal imaging, to improve the accuracy of detection even further.

Keywords: *Road Crack Detection, Deep learning, CNN, Image Processing, Smart Transportation.*

1. Introduction
Modern transportation systems depend on a sturdy road network that not only fuels economic progress but also protects public welfare. Well-maintained roads facilitate smooth travel, cut down on vehicle operating costs, and decrease accident risks. However, over time, factors such as weather exposure, heavy traffic, and natural wear cause road surfaces to deteriorate. Road surface cracks are a common maintenance challenge that, if left untreated, can undermine the integrity of the roadway, drive up repair costs, and significantly endanger the safety of commuters. Traditionally, detecting road cracks has involved manual inspections by engineers and maintenance crews using visual evaluations and specialized instruments. Due to their intensive time requirements, reliance on manual labor, and susceptibility to human error, these methods fall short in effectively monitoring extensive road networks. With advancements in technology, automated detection systems have emerged as a more accurate and cost-effective alternative [2]. Deep learning, particularly through the application of Convolutional Neural Networks (CNNs), has significantly transformed the field of computer vision, emerging as a robust method for the automated detection of CNNs possess the ability to isolate fine image details, enabling them to reliably distinguish between areas affected by road cracks and those that remain

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0171
e ISSN: 2584-2854
Volume: 03
Issue: 04 April 2025
Page No: 1050 - 1054

undamaged. This research harnesses deep learning to develop an intelligent system capable of real-time crack detection and classification, thereby reducing the dependence on manual inspections. However, the system still faces challenges. For example, shadows, occlusions, uneven surfaces, and background noise can occasionally result in detection errors. Moreover, developing reliable deep learning models relies on the availability of extensive, accurately labeled datasets—a resource that is not always easy to secure. Future improvements may include integrating additional sensor modalities, such as LiDAR and thermal imaging, as well as leveraging adaptive learning techniques to further enhance model performance [1].

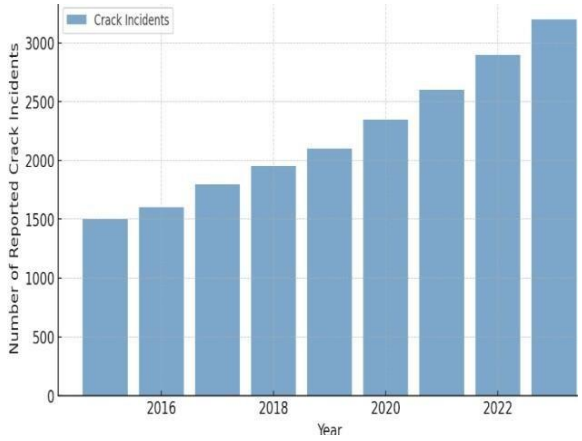

Figure 1 Increase in Road Crack Incidents Over the Years

2. Proposed Method

Given an image of pavement, the goal of a crack detection task is to identify whether a certain pixel belongs to a crack.To overcome this issue, the method leverages a Convolutional Neural Network (ConvNet) that is trained using meticulously annotated image patches. This allows the network to reliably determine whether each patch includes the relevant features. For clarity, image patches are categorized as "positive" if they contain cracks and as "negative" if they do not. In this study, every patch is examined and categorized based on these established criteria. As positive if its center is a crack pixel or is located close to one; otherwise, it is marked as negative [3].

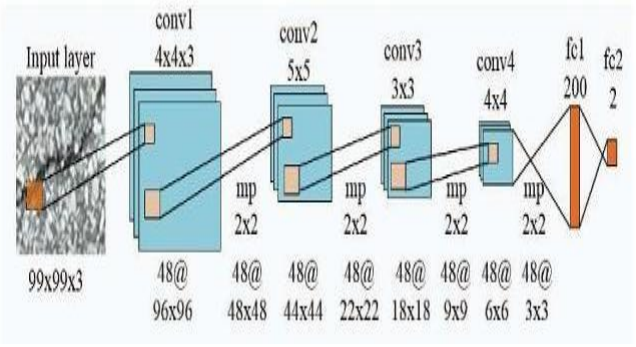

Figure 2 Illustration of the Architecture of the Proposed ConvNet

2.1 Data preparation

We assembled a dataset of over 500 pavement images, each with a resolution of 3264 × 2448 pixels, captured on the Temple University campus using a smartphone. Multiple annotators provided labels for each image. To strike a balance between computational efficiency and detection accuracy, each sample consists of a 99×99pixel image patch with 3 channels (RGB), derived through the following sampling method: A patch is designated as positive if its center lies within 5 pixels of the crack's centroid; otherwise, it is deemed negative. To reduce redundancy among training samples, it's essential to keep the overlap between two positive patches (P1 and P2) minimal. This overlap is quantified by the ratio O = area(P1 ∩ P2) / area(P1 ∪ P2), which should remain low. In this study, we maintain a distance of d = 0.75w between the centers of adjacent patches, where w denotes the width of each patch. For negative patches, adjacent patches must not overlap at all. At each patch center c, candidate patches are rotated by a random angle α, which is selected from a range between 0° and 360°. This technique is crucial for augmenting the number of crack samples, especially since such patches constitute a small fraction of the collected images. From this sampling process, we use 640,000 samples for training, 160,000 samples for cross-validation of the ConvNet, and 200,000 samples for testing. The number of crack and non-crack patches is balanced across all three datasets. From this sampling process, we use 640,000 samples for training, 160,000 samples for cross-validation of the ConvNet, and 200,000 samples for

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0171
e ISSN: 2584-2854
Volume: 03
Issue: 04 April 2025
Page No: 1050 - 1054

testing. The number of crack and non-crack patches is balanced across all three datasets. Figure 2 shows Illustration of the architecture of the proposed ConvNet.

2.2 ConvNet Architecture

Figure 1 illustrates the ConvNet architecture, where "conv" denotes the convolutional layers, "mp" indicates the maxpooling layers, and "fc" represents the fully-connected layers [4]. Essentially, the network functions as a hierarchical feature extractor by converting raw pixel values from All convolutional filter kernels are learned in a supervised fashion using a labeled dataset [5]. Each convolutional layer of the ConvNet applies maxpooling operations to condense feature responses from adjacent pixels, allowing the network to identify features that are invariant to the spatial location of objects within the images. Finally, fully connected layers perform the classification. Due to the exclusive nature of the crack detection issue (either crack or no crack), a softmax layer serves as the final layer of the ConvNet, which calculates the probability of each class based on the given input patch. Let the training dataset be denoted as S = {x(i), y(i)} for i = 1, 2, …, m, where m is the total number of image patches, x(i) represents the i-th image patch, and y(i) is its corresponding label. its associated label from the set {0,1}. indicates its class. If y(i) = 1, then x(i) is a positive patch; otherwise, it is negative. Let z(i)j denote the output of unit j in the last layer for x(i). The probability that the label y(i) of x(i) is j can be expressed as:

$$p(y^{(i)} = j|z_j^{(i)}) = \frac{e^{z_j^{(i)}}}{\sum_{l=1}^{k} e^{z_l^{(i)}}}, \quad (1)$$

The associated cost function is given by:

$$J = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\} \log \frac{e^{z_j^{(i)}}}{\sum_{l=1}^{k} e^{z_l^{(i)}}}\right] \quad (2)$$

here k = 2, m is the total number of patches, and (\mathbf{1}{\cdot}) is the indicator function.

2.3 ConvNet Training

The goal of training a ConvNet is to diversify the training data, thereby reducing the risk of overfitting. To accomplish this, a dropout layer is inserted between two fully connected layers, randomly setting each neuron's output to zero with a 50% chance to avoid overly complex co-adaptations. The training process is significantly accelerated by using graphics processing units (GPUs). Moreover, using rectified linear units (ReLU) as the activation function boosts computational efficiency, surpassing traditional functions like the hyperbolic tangent (tanh) and sigmoid during both training and evaluation. The network is optimized via stochastic gradient descent (SGD) with a batch size of 48, a momentum of 0.9, and a weight decay of 0.0005. Typically, fewer than 20 epochs are needed to reach a minimum on the validation set. Processing a Testing Image When processing a test image, the ConvNet calculates a probability for each pixel to reflect the likelihood that it represents a crack, resulting in a detailed probability map of the entire image. Drawing on methodologies from previous studies, the probability assigned to a specific pixel can be calculated by taking the average of the probabilities {P1,...,PN} from each patch generated by applying random rotations around that pixel, c. This is expressed as:

$$p(c|\{P_1(c), ..., P_N(c)\}) = \frac{1}{N}\sum_{i=1}^{N} P_i(c), \quad (3$$

Here, (Pi(c)) represents the ConvNet's predicted classification probability for the i-th patch at pixel c, with N fixed at 5 to maintain computational efficiency. The ConvNet possesses a greater number of degrees of freedom, which leads it to have high variance and low bias. Consequently, the count of crack patches is considerably lower than that of background patches in an image. This discrepancy may cause the ConvNet to overestimate the crack probability, necessitating the use of an appropriate threshold. Figure 3 shows ROC Curves. Define precision and recall as:

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0171
e ISSN: 2584-2854
Volume: 03
Issue: 04 April 2025
Page No: 1050 - 1054

$$P = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}, \quad (4)$$

$$R = \frac{\text{true positive}}{\text{true positive} + \text{false negative}}. \quad (5)$$

Then, the F1score is expressed as

$$F_1 = \frac{2PR}{P + R}. \quad (6)$$

All experiments were conducted on a computer equipped with an Intel Xeon E3 1241 V3 processor running at 3.5GHz, 8GB of RAM, and a NVidia Quadro K220 GPU. We developed the convolutional neural network (ConvNet) using the Caffe framework and trained it using a 5-fold cross-validation approach. To evaluate our method, we compared it against two other techniques: support vector machines (SVM) and Boosting. For the SVM, we utilized the LIBSVM library with a Gaussian radial basis function (RBF) kernel. We fine-tuned the parameters C and γ through 5-fold cross-validation to ensure optimal performance. For the Boosting method, we employed 100 weak classifiers, each with a maximum depth of 5, and trained the model using the OpenCV toolkit. In both cases, we selected the parameters that resulted in the lowest test error during cross-validation for our final comparisons. The features used to train the SVM and Boosting methods were based on the color and texture of each image patch, which were associated with a binary label indicating whether cracked pavement was present. The feature vector consisted of 93 dimensions, including color components, histograms of textons, and local binary pattern (LBP) descriptors within the patch. A detailed breakdown of the feature vector can be found in Table 1. Some of these features were adapted from prior studies, though unlike one of those studies, we intentionally excluded geometric information. Our goal was to develop a crack detection method that doesn't rely on specific geometric details. To assess performance, we generated receiver operating characteristic (ROC) curves, which are shown in Fig. 2, and compiled a summary of the results in Table 2. The findings clearly demonstrate that the ConvNet outperformed both the SVM and Boosting methods in detecting cracks. Figure 4 shows Detection probabilities for non-crack.

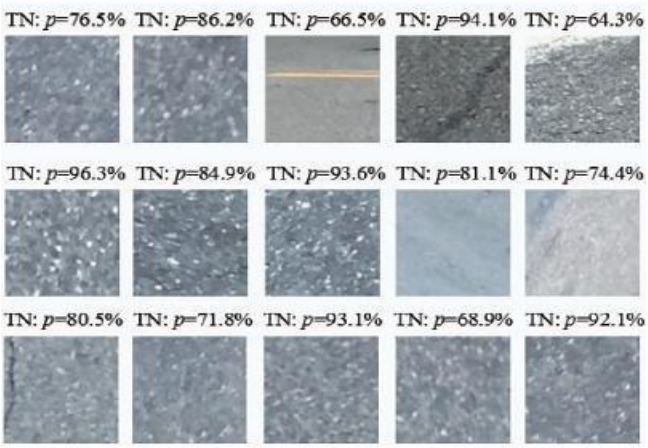

Figure 4 Detection Probabilities for Non-Crack

these values represent the likelihood of the ConvNet identifying an image as non-crack. TN indicates true negatives. The proposed method, relying on ConvNet, classifies images correctly, highlighting that its learned discriminative features surpass the performance of manually crafted features in articulating intricate patch contexts. Figure 5 shows Probability maps [6].

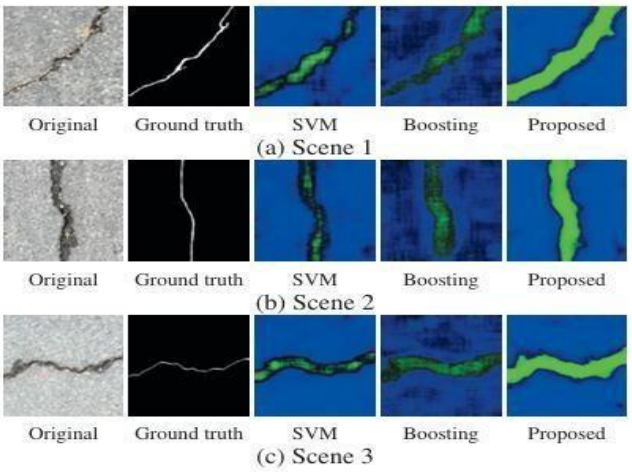

Figure 5 Probability Maps

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0171
e ISSN: 2584-2854
Volume: 03
Issue: 04 April 2025
Page No: 1050 - 1054

When a window extends beyond the image boundary, the missing pixels are reconstructed through mirroring. Figure 5 illustrates the results of crack detection across three different scenes. Each row in the figure represents the original crack image, ground truth, probability maps generated using SVM and Boosting methods, and the results produced by the ConvNet model. In the visualization, green pixels indicate cracks, while blue pixels represent non-crack areas, with brighter regions reflecting higher confidence levels in detection [7]. The Support Vector Machine (SVM) struggles to differentiate cracks from the background, leading to misclassification of some crack regions. In comparison, the Boosting method achieves higher accuracy in detecting cracks but misclassifies certain background patches as cracks, resulting in scattered green areas within the image. Unlike these methods, the proposed ConvNet-based approach demonstrates significantly better performance, accurately distinguishing crack patches from background regions while minimizing false detections.

Conclusions

We have developed an automated road crack detection system utilizing deep convolutional neural networks (CNNs), where the model autonomously learns features from manually labeled image patches captured using an affordable sensor, such as a smartphone. To the best of our knowledge, this research represents one of the earliest implementations of deep learning techniques for road crack detection. Moving forward, our goal is to enhance the efficiency and accuracy of the proposed detection system while developing a cost-effective, real-time solution for large-scale road monitoring and maintenance.

References

[1]. Chen, Y., Li, X., & Wang, Z. (2021). Generating synthetic road crack images using variational autoencoders for dataset augmentation. IEEE Transactions on Image Processing, 30, 1124–1135.

[2]. Cheng, H., Zhao, K., & Liu, R. (2023). Enhancing crack detection with self-attention mechanisms in UNet architectures. Computer Vision and Image Understanding, 211, 103450.

[3]. Gao, T., & Lin, P. (2022). Improving road crack detection robustness using conditional GAN-based data augmentation. Neural Networks, 149, 156–168.

[4]. Gupta, S., Patel, D., & Mehta, R. (2021). Edge computing for real-time road crack detection: A deep learning approach. Journal of Artificial Intelligence Research, 74, 891–905.

[5]. Huang, X., Feng, J., & Li, T. (2023). Smart vehiclebased vibration analysis for early crack detection in roads. Sensors, 23(4), 1982.

[6]. Kim, J., & Zhang, Y. (2021). Infrared-based crack detection for subsurface damage identification in road infrastructure.

[7]. Automation in Construction, 129, 103760. Kumar, A., Singh, P., & Rao, V. (2022). Selfsupervised learning for automated crack detection in road surfaces. Expert Systems with Applications, 193, 116392.