Intelligent Methods in Engineering Sciences 4(3): 74-84, 2025

IMIENS

INTELLIGENT METHODS
IN ENGINEERING SCIENCES

https://www.imiens.org

International Open Access

December, 2025

e-ISSN 2979-9236

Research Article

https://doi.org/10.58190/imiens.2025.152

CrackNet: Pavement Crack Detection and Classification Based on Deep Learning Models

Zubair Saeed [a,b,*], ***Ali Raza*** [c]

[a] *Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA*
[b] *Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Doha, Qatar.*
[c] *Department of Computer Science and Engineering, Hamad Bin Khalifa University (HBKU), Doha, Qatar.*

ARTICLE INFO

Article history:
Received 08 September 2025
Accepted 16 October 2025

Keywords:
Crack detection,
Deep learning,
Fracture detection,
InceptionV4,
MobileNet,
SqueezeNet

ABSTRACT

The identification of pavement cracks is essential for reducing traffic accidents and minimizing road maintenance costs. Existing crack detection methods frequently encounter challenges related to inefficiencies and accuracy, resulting in billions of dollars spent globally on road repairs annually. This study introduces an enhanced deep learning-based network aims at improving the accuracy of pavement crack detection. The proposed CrackNet utilizes advanced geometric augmentation techniques to enhance the model performance in identifying cracks across a variety of road conditions. We introduced CrackNet, a custom-designed deep-learning classification framework for pavement images. CrackNet combines pre-trained backbone feature extractors (MobileNetV2, InceptionV4, SqueezeNet) with a lightweight classifier head and a comprehensive preprocessing + augmentation pipeline to improve generalizability and address class imbalance. We evaluated three CrackNet variants (Sqz-CrackNet, Mob-CrackNet, Incep-CrackNet), each distinguished by its backbone, and found Incep-CrackNet achieved the highest accuracy of 96.05%, surpassing the Mob-crackNet and Sqz-crackNet, which attained accuracies of 95.44% and 90.06%, respectively. These findings underscore the effectiveness of the proposed deep learning framework in accurately detecting pavement cracks, representing a significant advancement over traditional detection methods.

1. INTRODUCTION

Generally, the modes of transportation are road, rail, water, and air. The transport system of Pakistan is primarily dependent on road transport, which makes up 90% of nationalpassenger traffic and around 96 % of freight movement. Overthe past several years, road traffic has grown day by day which resulting in severe road accidents and impacting the efficiency ofthe road. Pakistan is gifted with a naturally geo-strategic location. As it is at the crossroads of Asia with China, this makes it an attractive and shortest route for transportation to the Central Asian Republic. Road transportation is the backbone of the economy of Pakistan because imports and exports are done by road. And other countries also use the roads of Pakistan for their transport of goods. Friction between the road and vehicle, extreme weather conditions, and material aging impact the efficiency of roads. This is manifested in theform of cracks on the road. A crack is a condition of the roadthat occurs when concrete is completely or incompletelyseparated into two parts. The major causes of cracks on roadsare water accumulation, oil spills, drying shrinkage, stripping, and fatigue cracks [1].

The key component influencing road effectiveness is pavement deterioration. The identification of pavement deterioration in a timely and efficient manner is a critical element in pavement care. Cracks are the first sign of several forms of pavement issues. Pavement cracks not only degradepavement look and driving convenience, but they may also readily spread and create extensive destruction to the pavement, reducing ultimate operational productivity anddurability [2,3]. As a result, early crack identification and early management of cracked pavement can lower the economic cost of pavementrestoration while also ensuring the safety of cars and drivers using the roadway. That's why it is crucial to measure pavement crack type detection and develop repair and maintenance strategies to improve the usability of roads and public safety. Detecting road cracks in Pakistan roads helps vehicles move smoothly without getting stuck in these cracks. Assessing the quality of road surfaces by detecting road cracks is beneficial to reduce maintenance costs. It is vital to keep roads in good condition for driving safety assurance.And it is mandatory to identify cracks timely

[*] **Corresponding Author:** zubairsaeed602@gmail.com

manner for their repair. In the past, a lot of work is done by researchers in pavement crack detection by using a deep learning approach,but still, there is a need for improvement. In the current study,we have implemented a model on raw images taken from {4]. We normalized these raw images andtrained 80% dataset on three deep learning models, MobileNet V2, Inception V4, and SqueezeNet, and achieved higheraccuracy than found in the literature.

2. LITERATURE REVIEW

DL-based solutions have been successfully applied in multiple domains, including disease detection in humans [5–11], plant disease identification [12], [13], and various multidisciplinary applications such as UAV-based object detection, wearable sensing, and intelligent systems [14–18]. Similarly, DL techniques are also being used for the early crack detection and classification. In [3], Otsu's thresholding technique is used for programmed break identification and arrangement framework for substantial surface pictures, and their fogginessis taken out by utilizing a Wiener channel. Wavelet change and Singular Value Decomposition (SVD) are utilized for theremoval of non-uniform light in pictures. An arbitrary backwood calculation is chosen to characterize the pictures into explicit sorts i.e., minor, moderate, serious, and complex breaks. In [2], the authors proposed a mechanized road crack recognition framework dependent onYOLO v2. The dataset comprised 9,053 pictures with avehicle-mounted cell phone. These pictures were partitionedinto classes named wheel mark part, development joint imprint, equivalent stretch, development joint imprint, Partial asphalt, in general asphalt, Rutting, knock, pothole, detachment, crosswalk obscure, and white line obscure. In 19], investigators addressed a profound learning-based strategy for road break recognition by using ConvNet. A Dataset of 500 pictures is gathered at the TempleUniversity grounds by utilizing a cell phone. The model achieved 0.8696 accuracy and 0.8965 F1 scores. In [20], the authors present an incorporated AI model for programmed bridge crack recognition and groupingin metropolitan regions is done by using division, noise reduction, highlight extraction, and break arrangement. The dataset was gathered from various breaks in Tehran metropolitan region. The proposed crossbreed model is a blend of decision trees and SVM, which achieved 93.86% accuracy. In [21], the authors classified different street asphalt surface breaks. This input image was of 1536×2048 pixels, obtained by an optical gadget. In [22], researchers addressed a strategy for recognizingpixel-level street breaks. The dataset was gathered by discovery cameras. The proposed convolutional encoder-decoder network utilized ResNet-152, and the deconvolution strategies of SegNet, FCN, and ZDNet. The outcomes show that the proposed technique is ideal for detecting breaks in discovery pictures at the pixel level. In [1], a computerized and smart system is used for break location. The dataset is first preprocessed and then features are extracted using convolutional neural networks on MATLAB with a precision rate of 90%. In [23], the authors presented a novel mechanized break recognition system using the STRUM (spatially tuned strong multifeatured) classifier. By utilizing AI (Artificial Intelligence), the need to tune boundaries mechanically is eliminated. Element calculation incorporates the scale-spaceof the neighborhood and highlights the size of the break. STRUM classifier achieves 69% precision. In [24], researchers presented a break recognition strategy witha limited quantity of datasets by using contrast and the conventional edge location techniques. In [25], experts presented a learning model based on the YOLOv4 tofind connecting breaks by an automated flying vehicle, UAV (unmanned aerial vehicle). [26] provided an extensive overview of advancements in asphalt pavement technologies, emphasizing sustainability, smart diagnostics, and data-driven practices. Their insights help frame the importance of integrating intelligent systems such as CrackNet into modern pavement assessment workflows. [27] introduced a YOLO-based deep learning architecture for detecting and classifying pavement distress types. Their success in applying real-time object detection supports the viability of lightweight yet accurate networks in field-ready applications. [28] developed an image processing pipeline based on crack pixel density that effectively quantifies and classifies pavement cracks with high reliability. Their work demonstrates how structural metrics can enhance interpretability and accuracy in classification tasks. This strategy has low identification productivity and high work cost. The detailed comparison of literature is shown in Table 1.

2.1. Paper contribution

Pavement Management Systems (PMS) are gaining importance in pavement surveying, monitoring, and restoration. To enhance pavement management, an increasing number of transportation agencies are creating andenhancing pavement management systems. Pavement crack analysis, which comprises crack identification and detection of crack progression, is an important and necessary function in PMS. Many transportation authorities are now accumulating pavement image data on a regular and frequentbasis. As a result, this research proposed a method for analyzing existing cracks that uses previous data as a benchmark. The contributions are listed below,

Comprehensive literature review of existing methodologies. Found research gaps and where improvements can be made.

The authors presented a method for utilizing previous crack records as a standard for pavement crack evaluation, which may considerably improvepavement crack analysis effectively.

The network makes full use of the semanticinformation of hierarchical convolution features andis very effective for crack detection under a complex scene.

Detailed comparison of results achieved in balancedvs imbalanced datasets.

Table 1. Comparison of Previous Studies.

Research	Problem	Technique Used	Results
[3]	Review and analysis of crack detection and classification based on crack types	Image processing and Machine learning	Break recognition and characterization methodfor common foundationare effective and widely used
[2]	Automated road crack detection	DCNN	Developed a dataset using google road
[19]	Road crack detection	DCNN	Accuracy: 86.96%
[20]	Automated road crack detection andclassification	Integrated machinelearning	Accuracy: 83.86%
[21]	Crack detection and characterization	Image processing	-
[22]	Road crack detection	Encoder-decoder network	Results shows the proposed technique is ideal for detecting breaks in pictures at the pixel level
[1]	Bridge crack detection	CNN	Accuracy: 90%
[23]	Automated road crack detection	Machine learning and Computer vision	-
[24]	Bridge crack detection	Image processing	Accurately identify cracksin collected images
[25]	Bridge crack detection	YOLO-V4-FPM	-
[28]	Pavement crack detection and classification	Crack pixel density-based image processing	Effectively segments and classifies cracks using automated pipeline
[26]	Advancement in asphalt pavement	Review of pavement design and maintenance tools	Highlights recent innovations, emphasizing sustainability and smart diagnostics

3. MATERIALS AND METHODS

Roads are considered important infrastructure for every country; therefore, special care is needed to detect these cracks before a major loss. Deep learning has proven as the most powerful tool for visual recognition and produces the highestaccuracies as compared to the traditional machine learning approach. Figure 1 shows the block diagram of our research methodology, where the dataset of the cracks in pavement is first normalized, then fed into three deep learning models i.e., SqueezeNet, MobileNetV2, and InceptionV4, and finally classified by their softmax layer, which is a fully connected layer, as cracked and non-cracked images.

3.1. Dataset Distribution

Structural defect (SDNET2018) is an annotated dataset used for AI-based crack detection training and validation. The collection includes 56,000 pictures of cracked and non-cracked bridges, pavement and wall samples as shownin Figure 3, Figure 4, and Figure 5, respectively. This dataset comprises fractures ranging in size from0.06mm to 25mm, allowing us to identify the tiniest and deepest apparent fissures. SDNET2018 images were shot with a 16-megapixel Nikon camera at a working distance of 500 mm without zoom. The ISO was set to 125, and the image resolution was 4068 x 3456 px. The surface lighting varied between 1500 and 3000 lx. Figure 3-5 shows the kind of imagesthis dataset contains.

Each image in the dataset is divided into 256x256 pixels and covers a corporeal region of around 1000 x 850 mm. Cracked and non-cracked categories. The pavement images have been acquired from the roads and the sidewalks, in which 2608 images contain cracks, while 21726 images belong to the non-cracked category. Images of walls and pavements were shot on the Utah State University campus. The dataset has been acquired from Kaggle, which is publicly available for researchers [4].

3.2. Data Augmentation

We have augmented cracked images (only) eight times, as shown in Table 2 and the augmented samples are shown in Figure 6.

Table 2. Augmented Dataset.

Sr. No	Class	Images After Augmentation
1.	Crack	2,608 x 8=20,864
2.	Non-Crack	21,726
	Total	42,590

3.3. Preprocessing

Raw images are preprocessed for each model. The following steps are adopted to obtain the required pre-processed images.

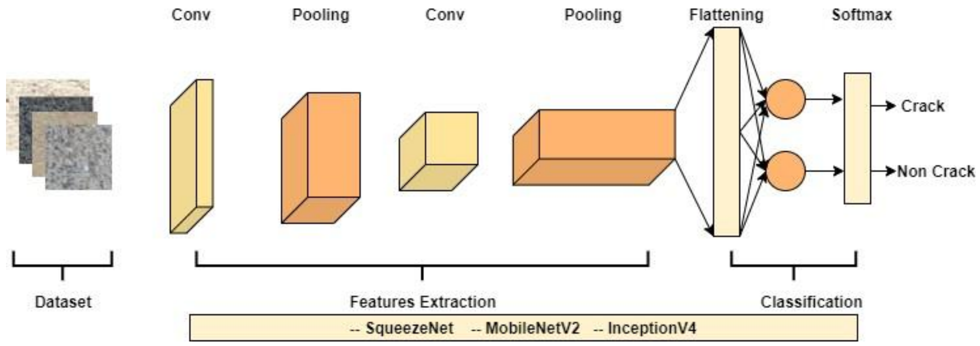


Figure 1. Block Diagram of Autonomous Crack Detection.

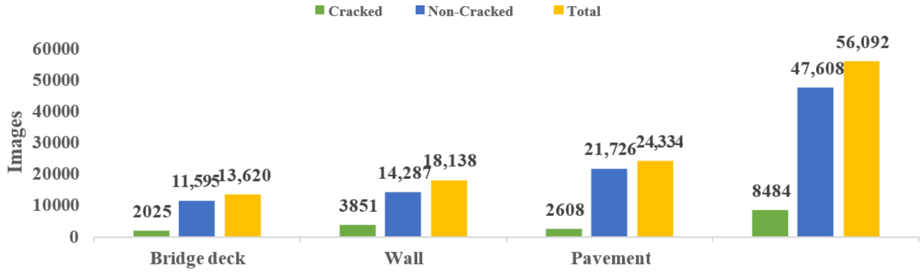


Figure 2. SDNET2018 Dataset Distribution.

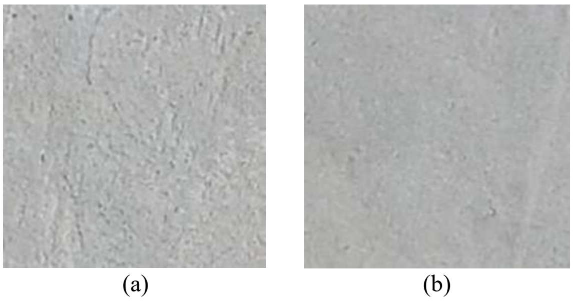

(a)
(b)

Figure 3. Crackled and non-cracked bridge deck images

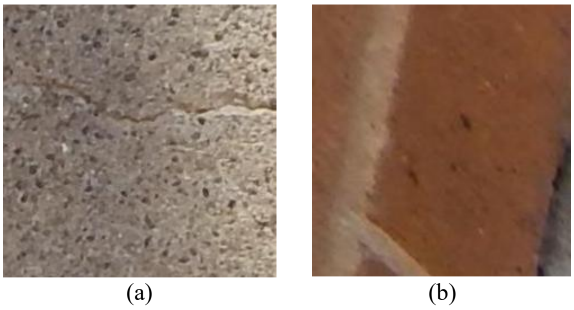

(a)
(b)

Figure 4. Crackled and non-cracked pavement images.

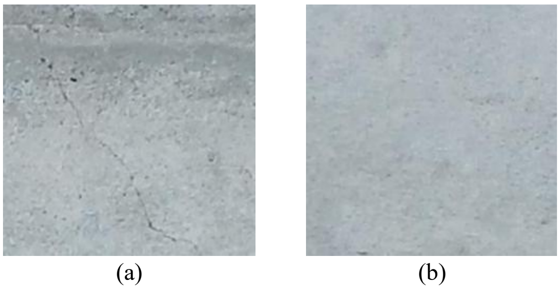

(a)
(b)

Figure 5. Crackled and non-cracked wall images.

3.3.1. Resize

The input images were of size 256 x 256, which were resized to 299 x 299 for InceptionV4 (shown in Figure 7) while 224x224 for MobileNetV2 and SqueezeNet as shown in Figure 8. Figure 9 shows the individual cracked and non-cracked raw images split into respective red, green, and blue channels.

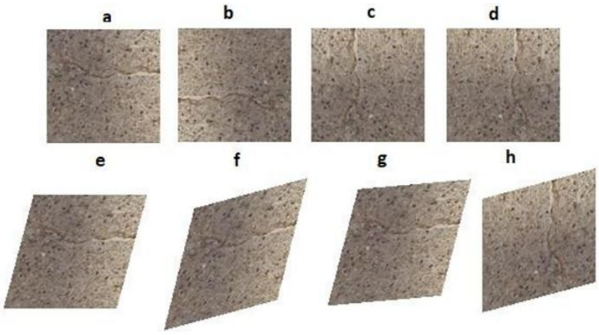


Figure 6. Image Augmentation a) Original image b) 180-degree rotated c) 90-degree left rotated d) 90-degree right rotated e) 15-degree right rotated f) 30- degree right rotated g) 25-degree right rotated h) 15-degree left rotated.

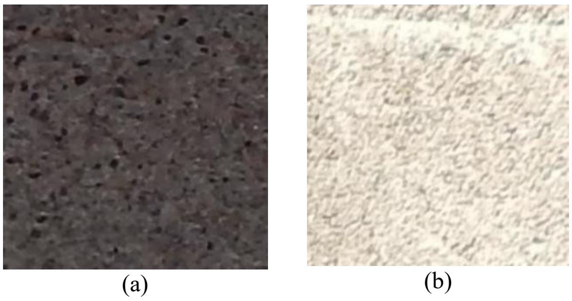


Figure 7. Resized 299x299 image for InceptionV4 (a) Cracked (b) Non-cracked.

Figure 8. Resized 224x224 image for MobileNetV2 and SqueezeNet (a) Cracked (b) Non-cracked.

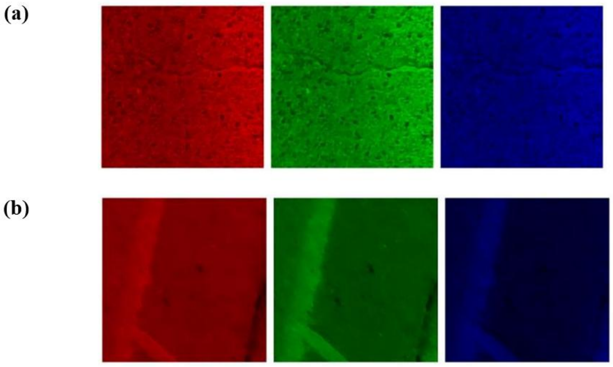


Figure 9. RGB channels of (a) Cracked and (b) Non-cracked images.

3.3.2. Normalization

Normalization is done because, during the training phase,the model multiplies weights and adds to biases. These initialinputs form activations, which we will then backpropagate using gradients to train the model. Here, each feature must have a comparable range so that gradients don't spiral out of control. Inputs with big decimal numbers can interrupt or impede the learning process in deep learning models, which process inputs with modest weight values. As a result, normalizing the pixel values so that each pixel value has a value between 0 and 1 is recommended in practice. The aim of normalization also comes from calibrating the diverse pixel intensities into a normal distribution, which improves the image's appearance for the visualizer [29].

Resized images are normalized into a fixed mean of [0.485, 0.456, 0.406] and standard deviation of [0.229, 0.224,0.225] for RGB images having red, green, and blue channels. This means an STD was calculated by using ImageNet [30], which contains 14 million images from 1000 classes, as done. Figure 9 and Figure 10 show cracked and non-cracked images before and after Normalization. The pixel intensities for cracked images before normalization ranged from 45 to 220, and after normalization, 0.2-0.8. Non-cracked image in Figure 11 shows 25-to-170-pixel values, and after normalization it shows 0 to 0.6.

3.4. Dataset Distribution

The 80% dataset has been randomly separated for training purposes, while 20% is reserved for validation, as shown in Tables 2 and 3. The original data + trained model is called Scheme 1, and the augmented data + trained model is called Scheme 2.

Table 3. Dataset Distribution of Scheme 1.

Class	Images	Training	Validation
Crack	2,608	2086	522
Non- Crack	21,726	17,020	4706
Total	24,334	19,106	5,228

Table 4. Dataset Distribution of Scheme 2.

Class	Images	Training	Validation
Crack	20,864	16,691	4,173
Non- Crack	21,726	17,020	4,706
Total	42,590	33,711	8,879

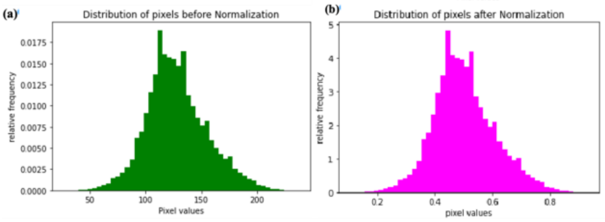


Figure 10. Cracked image before and after normalization.

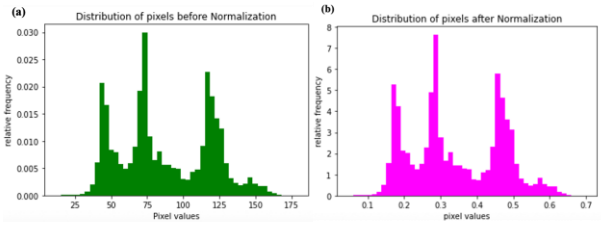


Figure 11. Non-cracked image before and after normalization.

3.5. Transfer Learning

Transfer learning is a machine learning approach in which a model that has been built for a certain task is utilized againas the reference basis for a second task model [31] as shown in Figure 12.

In this article, we utilized trained base networks on a dataset named as ImageNet [30] such as Mobilenet v2 [32], Inception v4 [33] and SqueezeNet [34] and a task (classification of 1000 classes), and then reassigned or transferred the learning characteristics to a second target network to be trained in a desired dataset- SDNET2018 [35],[4] and task (crack detection and classification). This approach's utilization succeeded in classifying pavement cracks because the models were trained on generic attributes of 1000 classes, which means that they are suited for both the base and the target tasks classification.

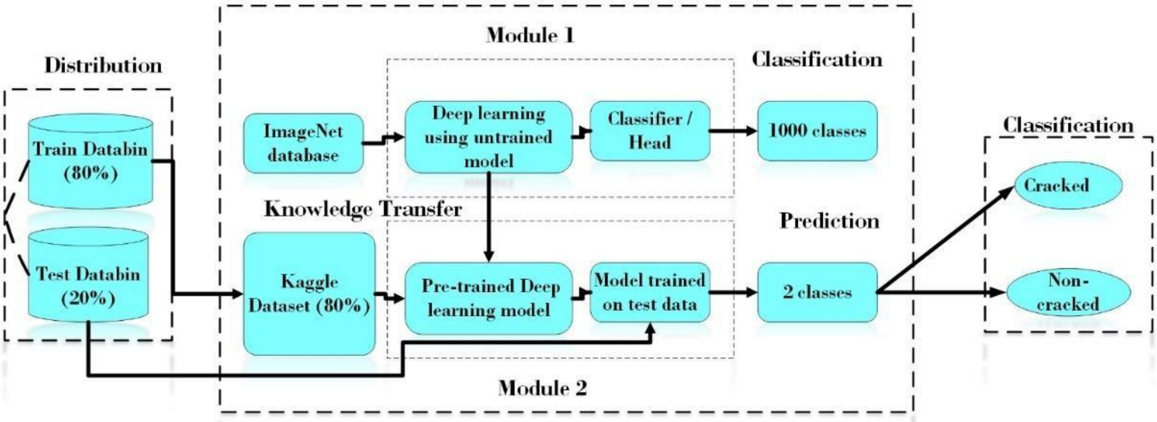


Figure 12. Block Diagram of Transfer learning Approach [31].

3.6. CrackNet Architecture

CrackNet is the task-specific classifier head used on top of the frozen/pre-trained backbones. The head comprises a global average pooling layer followed by a fully connected layer and a two-unit softmax output for binary classification (cracked vs non-cracked). During transfer learning we freeze the backbone (pre-trained on ImageNet) and unfreeze the final layers of the backbone and the CrackNet head for target-task fine-tuning. The same CrackNet head is used across the three variants; Sqz-CrackNet, Mob-CrackNet and Incep-CrackNet refer only to the backbone choice.

MobileNetV2: The MobileNetV2 deep learning model is used as a backbone network in this study. The architectural details of MobileNetV2 can be found in [32].

InceptionV4: The complete architecture of InceptionV4, and its details can be referred from [33].

SqueezeNet: SqueezeNet, where each building block is known as the fire module, which is one squeezed layer, and one expanded layer combined to form a fire block. This model has four fire blocks followed by a convolution and pooling layer. Detailedarchitecture layers and blocks can be seen for further understanding in [34].

3.7. Hyperparameters (HP) Selection and Tuning

Here, each input images are resized according to the model requirement and then normalized into mean and standard deviation calculated by using ImageNet [30]. The tuning procedure evaluated a compact set of candidate values for each hyperparameter and selected values that (i) produced stable, monotonic decreases in validation loss, and (ii) achieved high validation accuracy without evident overfitting. The candidate values and final choices are summarized in Table 5. Specifically:

3.7.1. Batch size (BS)

The number of training instances forwarded to the networks is referred to as the batch size. A training dataset can be broken down into one or more batches. The learning technique is known as batch gradient descent when all training samples are supplied in a single batch. The learning technique is known as stochastic gradient descent when the BS is one. The learning algorithm is known as mini-batch gradient descent when the BS is greater than one but less thanthe training size. The BS’s used in mini-batch gradient descent are 32, 64, and 128. The larger the batch size, the more accurate the network. With a power of 2, the BS rangesfrom 1 to 1024 [36]. Increasing the batch sizereduces model performance by causing it to gravitate to sharpminimizers of the training function [37]. The batch size is determined by the application; after extensive research, here we decided to use BS of 64.

3.7.2. Learning Rate

One of the most essential factors used to optimize the models is the learning rate. It reduces error by adjusting network weights. Choosing a learning rate that is too low andan LR that is too high might harm model performance. A low LR will result in little updates to the network weights, slowing down the training process, whereas a large LR will result in divergent behavior in the error. Training should begin with a high LR since random weights are far from idealat the outset, and learning rates will fine-grained the network weights by reducing their value overtraining. The approach provides could begin with a large value, such as 0.5, and thenprogress to lower values such as 0.05, 0.005, and so on. [36] recommended using tiny LR for bigger issues [37,38], which is why we selected 0.0005 LR here.

3.7.3. Epochs

The number of epochs indicates how many times the trainingdataset has been traversed. Every epoch indicates that the training sample can modify the model's internal parameters. There may be one or more batches. The number of epochs might be somewhere between 0 and infinity. Hundreds of thousands of epochs are often chosen. This enables the network to effectively reduce errors. A designer must study the error and accuracy learning curves to identify the optimalepoch value. These curves can help in assessing the model's learning states, such as over-learned, under-learned, or training-suitable [36]. We used a total of 5 epochs after repeated testing, which was then divided into 200 epochs for a total of 1000 epochs.

3.7.4. Optimization

The Adam optimizer (AO) is a combination of two gradient descent algorithms: the gradient descent algorithm (GDA) and Root Mean Square Propagation (RMSP). AO builds on the attributes or positive characteristics of the priortwo systems to produce a more optimum gradient descent. Inthis scenario, the gradient descent rate is tuned so that there is little variation when it reaches the global minimum while making big steps to avoid the local minimum obstacles alongthe way. As a consequence, using the advantages of the priorprocedures to efficiently achieve the global minimum [39]. The graph in Figure12 clearly shows how AO surpasses the other optimizers in terms to training cost (minimal) and performance by a significant margin (high).

Table 5. Hyperparameter Search Summary And Final Choices.

Hyperparameter	Candidate Values	Selected Value (Used in Study)
Batch size	{32, 64, 128}	64
Learning rate	{1e-3, 5e-4, 1e-4, 5e-5}	5e-4
Optimizer	{SGD, Adam}	Adam
Max epochs (max)	up to 1000	up to 1000 (best checkpoint used; best epoch ≈ 5)

Table 6. Deep Learning Models Comparison

Deep learning Model	Layers Depth	Convolution	Parameters
MobileNetV2	52	1×1 convolution	3.504M
InceptionV4	22	1×1 convolution 3×3 convolution	43 M
SqueezeNet	18	1×1 convolution	5MB

3.8. Feature Extraction

Convolution and pooling layers in all three models (MobileNetV2, InceptionV4, SqueezeNet) conduct feature extraction, and a fully connected layer, named as a softmax layer transforms these extracted features into the final output, such as classification.

3.8.1. Convolution layer

A convolution layer is made up of a series of numerical computations, such as convolution, which is a subset of lineartransformations. Pixel measurements are stored in a - two-dimensional (2D) grid (vector of integers) and a matrix of variables called kernel, an optimizable feature extractor, which is applied to each digital image [40].

3.8.2. Pooling Layer

A pooling layer performs a standard down-sampling function, reducing the in-plane dimension of the feature map to incorporate translational coherence to minor distortions and to reduce the number of future trainable features. It has two kinds, Max pooling and Global average pooling [41].

3.9. Classification

Final classification of all three models is done with the help of a fully connected softmax layer. The softmax layer of SqueezeNet, MobileNetV2, and InceptionV4 has two outputs (cracked and non-cracked), and average pooling has been used for the assignment of the output class during training and validation. total of 1000 epochs. The transfer learning approach is applied. Pre-trained weights of models trained on ImageNet are frozen, and the last layers are unfrozen to make the data learn on the Pavement dataset. Adam optimizer is used for optimization purposes.

4. RESULTS AND DISCUSSION

A frequent drawback identified in the existing research for vision-based analysis on road/pavement structures is that the established techniques only considered a single structure, thus their capacity to generalize when tested on more varied data has to be verified that's why we used a Kaggle dataset that had a lot of varied images. Training and validation of all the deep learning models i.e., SqueezeNet, MobileNetV2, and InceptionV4 done on the graphics processing unit (GPU) TitanXp, having Intel Xeon® CPU E5-2687W v4 @ 3.0GHz. All experiments are performed by creating a separate virtual environment using PyCharm. The batch size for training and validation is kept 64 images. All models were trained on 80% dataset with a learning rate of 0.0005 and a total of 5 epochs, which is further divided into 200 epochs, so combining all, we have a total of 1000 epochs. The transfer learning approach is applied. Pre-trained weights of models trained on ImageNet are frozen, and the last layers are unfrozen to make the data learn on the Pavement dataset. Adam optimizer is used for optimization purposes.

4.1. Scheme 1

Figure13 shows the overall accuracy achieved against each of the deep learning models in scheme 1. SqueezeNet has achieved an overall accuracy of 84.69%, whereas MobileNetV2 and InceptionV4 achieved 92.45% and 95.10%, respectively. InceptionV4 has the highest accuracy as compared to both models. Figure14 shows the training accuracy VS training loss along with the validation accuracy VS validation loss of all models against each epoch. Initial training and validation accuracies are improved during each epoch, and finally, the highest training and validation accuracies are achieved at the fifth epoch. Similarly, the training and validation losses of each model were low andimproved in the fourth and fifth epochs.

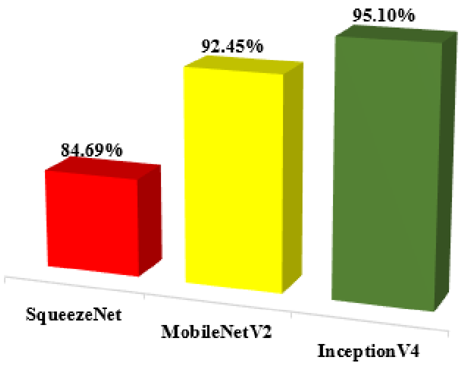


Figure 13. Overall Accuracy of Deep Learning Models Of Scheme 1

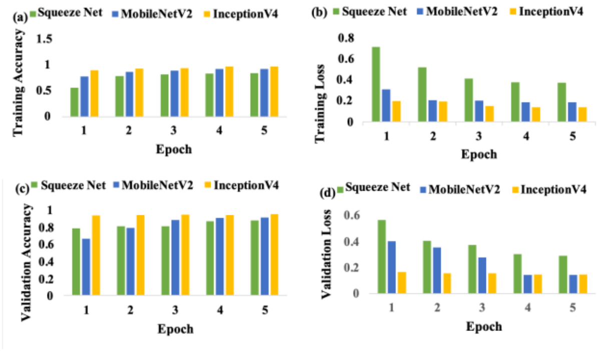
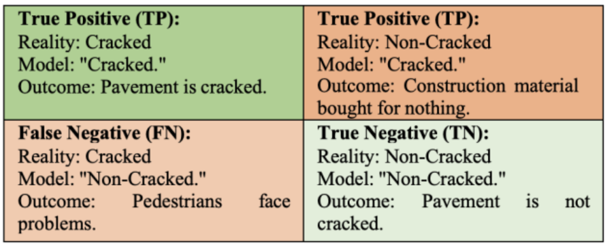


Figure 14. Scheme 1 (a)Training Loss (b) Training Accuracy (c)Validationaccuracy (d) Validation loss



True Positive (TP): Reality: Cracked Model: "Cracked." Outcome: Pavement is cracked.	**True Positive (TP):** Reality: Non-Cracked Model: "Cracked." Outcome: Construction material bought for nothing.
False Negative (FN): Reality: Cracked Model: "Non-Cracked." Outcome: Pedestrians face problems.	**True Negative (TN):** Reality: Non-Cracked Model: "Non-Cracked." Outcome: Pavement is not cracked.

Figure 15. Intelligent Pavement Crack Confusion Matrix.

Table 7 gives us detailed insight into how the model classifies each class. The Inception model has the highest TP (20069) and TN (20860). A model must have high TN and TP, then only can good classification be done. FP and FN need to be low, as in the inception case it is lowest among the other two.

Table 7. Confusion Matrix of Scheme 2 .

Model	**True Positive (TP)**	**False positive (FP)**	**False Negative (FN)**	**True Negative (TN)**
InceptionV4	20069	886	795	20860
MobileNetV2	19,868	943	996	20803
SqueezeNet	18797	2165	2067	19581

When we train deep learning models on an augmented dataset, then we see significant accuracy improvements in all three models. Parameters mentioned in Table 8 are calculated using Table 7 elements. Their respective mathematical equations are given below,

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FN} \quad (2)$$

$$Specificity = \frac{TN}{TN+FP} \quad (3)$$

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

If we see in detail, then it is apparent that the accuracy of inception model increased from 95.10% to 96.05% (0.95%), MobileNet v2 increased 92.54% from 95.44% (2.9%), and that of squeeze net increased from 84.69% to 90.06% (5.37%) as shown in Table 8. The sensitivity and specificity of the three models are in the following order,

$$SN_{inception} > SN_{mobile} > SN_{squeeze}$$
$$SP_{inception} > SP_{mobile} > SP_{squeeze}$$
$$P_{inception} > P_{mobile} > P_{squeeze}$$

Table 8. Dataset Distribution of Scheme 2. Scheme 2

Model	Precision(P)	Sensitivity (SN)	Specificity (SP)	Accuracy(A)
InceptionV4	96.189%	95.771%	96.328%	96.054%
MobileNetV2	95.226%	94.904%	95.430%	95.449%
SqueezeNet	90.092%	89.671%	90.451%	90.068%

Scheme 2 contained the dataset trained on the augmented dataset.Here, we calculated the efficiency of models based on the confusion matrix and classification merits. A confusion matrix tells us about the detailed working of a trained model, as shown in Figure15.

4.2. Model prediction

Figure 16 shows model predictions when randomly selected test images from each class were given to the trained models. Some images show the false prediction against each deep learning model. Figure 17 shows the results of each class wherethe trained models had false predictions. The first row shows the results where the model has false positive predictions, and thesecond row shows the examples of false negatives predicted by the model. A false positive is an outcome in which the model forecasts the positive class erroneously. A false negative is an outcome in which the model forecasts the negative class inaccurately.

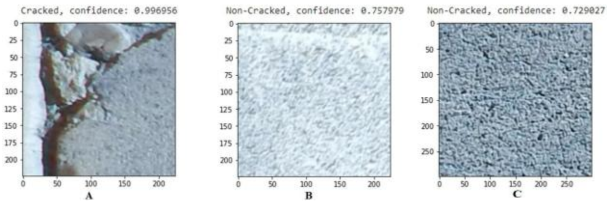

Cracked, confidence: 0.996956
Non-Cracked, confidence: 0.757979
Non-Cracked, confidence: 0.729027
A
B
C

Figure 16. Random Prediction of Class with Confidence using A) InceptionV4, B) MobileNetV2, and C) SqueezeNet Scheme 2.

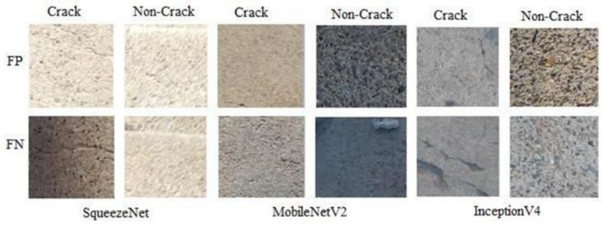


Figure 17. Samples Having False Positive (FP) and False Negative (FN) Scheme 1.

4.3. Why inception v4 performs better?

Inceptionv4 is a deep neural network that trains deeper networks, increases the accuracy. To make a good model, wefirst have to make sure that its performance on the training data is good. While designing a CNN, the filter size is a verydeciding factor. Inception uses 1 X 1,3 X 3, 5 X 5 filters, a convolution layer, and a pooling layer, and stacks all the outputs together. To reduce computations, 1 X 1 convolution is applied before 3 X 3 and 5 X 5 convolutions. The inceptionmodel is made up of these inception blocks repeated atdifferent locations, as well as a fully linked layer and a SoftMax layer. This thing increases its accuracy and performance [33]. That’s why inceptionv4 accuracy is higher other two neural SqueezeNet and Mobilenetv2. Figure 18 shows the operational accuracy of various deep learning models trained on ImageNet. Inception v4 tops all of them significantly.

4.4. Advantages

Pavement cracks can be caused by an inadequately structured joint, deformation of the asphalt layer, microcracks showing up from a foundational coating, or separation caused by defective activity. As a result, early crack identification and early restoration of cracked pavement can lower the economic cost of pavement repair while also ensuring the welfare of cars and drivers using the roadway.

4.5. Comparative Analysis

Li et al achieved 94% accuracy [42] and A. Ahmadi et al 93.86% [20], but our proposed methodology pre-processing + InceptionV4 achieved 95.10%. Du et al. achieved 73.64% with Yolo [27], and our pre-processing SqueezeNet achieved 84.69% accuracy. Safaei et al had 80% accuracy [28] while Lei Zhang et al obtained 86.96% [19] and our pre-processing + MobileNetV2 achieved 92.45% accuracy. Our proposed work surpasses them all in accuracy measurement. Detailed analysis is shown in Table 9.

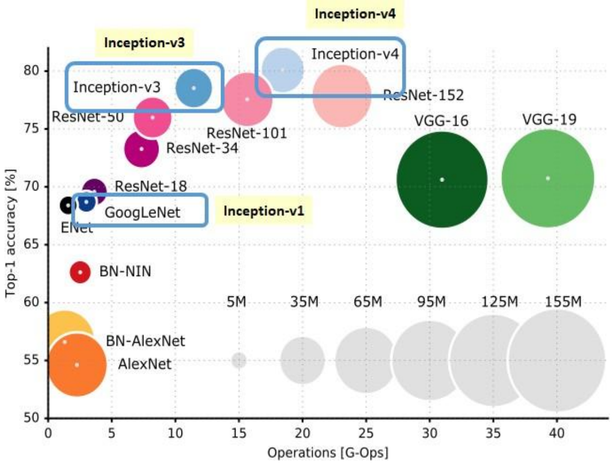


Figure 18. Operationals performed vs accuracy for multiple deep learning models [33]

5. CONCLUSIONS & RECOMMENDATIONS

The growing demand for efficient and autonomous roadway crack detection underscores the importance of developing robust and accurate classification models. In this study, we introduced CrackNet**,** a custom-designed DL framework capable of classifying pavement images as either cracked or non-cracked. To assess its effectiveness, we implemented and evaluated three variants i.e., Sqz-CrackNet, Mob-CrackNet, and Incep-CrackNet each optimized for different levels of efficiency and feature extraction capability. All three models demonstrated superior classification accuracy, significantly outperforming their respective baseline architectures. These performance gains can be attributed to the careful integration of model-specific architectural enhancements, as well as the implementation of a comprehensive data augmentation pipeline and preprocessing techniques to improve generalizability and address class imbalance. Among the proposed variants, Incep-CrackNet achieved the highest accuracy, showing promise for real-world deployment in edge environments.

Table 9. Comparison of State-of-The-Art Versus Our Approach.

Contribution	Dataset	Methodology	Accuracy
[42]	3D pavement images	CNN	94%
[28]	Midwest Transportation Center (MTC) Dataset	Image Processing techniques	80%
[27]	PD images Dataset	Yolo	73.64%
[19]	Self-500 pavement images	Deep Convolutional NeuralNetwork (DCNN)	86.96%
[20]	400 samples captured inTehran urban area	Integrated machine learning model	93.86%
Proposed Work: Original Dataset	SDNET2018	SqueezeNet	84.69%
		MobileNetV2	92.45%
		InceptionV4	95.10%
Proposed Work: Augmented Dataset		SqueezeNet	90.068%
		MobileNetV2	95.449%

This research can be extended to enable object-level crack detection and localization using advanced versions of the YOLO (You Only Look Once) framework. This will not only enhance the system's capability to detect the presence of cracks but also identify their specific locations and dimensions. The most effective model will be deployed on the OAK-D (Open-Source AI Kit–Depth) platform to develop a portable, real-time crack detection tool suitable for field use by road maintenance personnel. Additionally, fine-tuning of the YOLO models will be explored to further improve detection accuracy under diverse environmental and surface conditions.

Declaration of Ethical Standards

This research did not involve human participants, animals, or the use of identifiable personal data and therefore did not require ethics approval or informed consent.

Credit Authorship Contribution Statement

Conceptualization: Zubair Saeed; Methodology: Zubair Saeed; Software: Zubair Saeed and Ali Raza; Validation: Zubair Saeed; Formal analysis: Ali Raza; Investigation: Zubair Saeed and Ali Raza; Writing – original draft: Zubair Saeed; Writing – review & editing: Zubair Saeed and Ali Raza; Visualization: Zubair Saeed; Supervision: Zubair Saeed.

Declaration of Competing Interest

The authors declare that they have no known financial or non-financial competing interests or relationships that could have appeared to influence the work reported in this paper.

Funding / Acknowledgements

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The dataset used in the study is publicly available and can be accessed through: digitalcommons.usu.edu/all_datasets/48/.

References

[1] L. Zhang, G. Zhou, Y. Han, H. Lin, and Y. Wu, "Application of Internet of Things Technology and Convolutional Neural Network Model in Bridge Crack Detection," *IEEE Access*, vol. 6, pp. 39442–39451, 2018, doi: 10.1109/ACCESS.2018.2855144.

[2] V. Mandal, L. Uong, and Y. Adu-Gyamfi, "Automated Road Crack Detection Using Deep Convolutional Neural Networks," in *Proc. 2018 IEEE Int. Conf. on Big Data (Big Data)*, 2018, pp. 5212–5215, doi: 10.1109/BigData.2018.8622327.

[3] S. S. N. Sheerin, S. Kavitha, and G. Raghuraman, "Review and Analysis of Crack Detection and Classification Techniques Based on Crack Types," *International Journal of Applied Engineering Research*, vol. 13, no. 8, pp. 6056–6062, 2018, doi: 10.37622/IJAER/13.8.2018.6056-6062.

[4] M. Maguire, S. Dorafshan, and R. J. Thomas, "Structural Defects Network (SDNET) 2018," Kaggle dataset, 2018. [Online]. Available: https://www.kaggle.com/datasets/aniruddhsharma/structural-defects-network-concrete-crack-images

[5] A. Raza, M. U. Khan, Z. Saeed, S. Samer, A. Mobeen, and A. Samer, "Classification of eye diseases and detection of cataract using digital fundus imaging and Inception-V4," in *Proc. Int. Conf. on Frontiers of Information Technology (FIT)*, 2021, pp. 137–142.

[6] Z. Saeed, M. U. Khan, A. Raza, H. Khan, J. Javed, and A. Arshad, "Classification of pulmonary viruses X-ray and detection of COVID-19 based on invariant of Inception-V3 deep learning model," in *Proc. 2021 Int. Conf. on Computing, Electronic and Electrical Engineering (ICE Cube)*, Quetta, Pakistan, 2021, pp. 1–6, doi: 10.1109/ICECube53880.2021.9628338.

[7] M. U. Khan, M. A. Abbasi, Z. Saeed, M. Asif, A. Raza, and U. Urooj, "Deep learning-based intelligent emotion recognition and classification system," in *Proc. 2021 Int. Conf. on Frontiers of Information Technology (FIT)*, Islamabad, Pakistan, 2021, pp. 25–30, doi: 10.1109/FIT53504.2021.00015.

[8] Z. Saeed, O. Bouhali, J. X. Ji, R. Hammoud, N. Al-Hammadi, S. Aouadi, and T. Torfeh, "Cancerous and non-cancerous MRI classification using dual DCNN approach," *Bioengineering*, vol. 11, no. 5, p. 410, 2024.

[9] M. U. Khan, Z. Saeed, A. Raza, Z. Abbasi, S. Z. E. Z. Ali, and H. Khan, "Deep learning-based decision support system for classification of COVID-19 and pneumonia patients," *Journal on Advanced Research in Electrical Engineering (JAREE)*, vol. 6, no. 1, 2022.

[10] S. Z. H. Naqvi, M. U. Khan, A. Raza, Z. Saeed, Z. Abbasi, and S. Z. E. Z. Ali, "Deep learning-based intelligent classification of COVID-19 and pneumonia using cough auscultations," in *Proc. 2021 6th Int. Multi-Topic ICT Conf. (IMTIC)*, Jamshoro and Karachi, Pakistan, 2021, pp. 1–6, doi: 10.1109/IMTIC53841.2021.9719740.

[11] U. Nawaz, M. Anees-ur-Rahaman, and Z. Saeed, "A review of neuro-symbolic AI integrating reasoning and learning for advanced cognitive systems," *Intelligent Systems with Applications*, vol. 26, p. 200541, 2025, doi: 10.1016/j.iswa.2025.200541.

[12] Z. Saeed, M. U. Khan, A. Raza, N. Sajjad, S. Naz, and A. Salal, “Identification of leaf diseases in potato crop using deep convolutional neural networks,” in *Proc. 2021 16th Int. Conf. on Emerging Technologies (ICET)*, Islamabad, Pakistan, 2021, pp. 1–6, doi: 10.1109/ICET54505.2021.9689807.

[13] Z. Saeed, A. Raza, A. H. Qureshi, and M. H. Yousaf, “A multi-crop disease detection and classification approach using CNN,” in *Proc. 2021 Int. Conf. on Robotics and Automation in Industry (ICRAI)*, Rawalpindi, Pakistan, 2021, pp. 1–6, doi: 10.1109/ICRAI54018.2021.9651409.

[14] Z. Saeed, M. H. Yousaf, R. Ahmed, S. A. Velastin, and S. Viriri, “On-board small-scale object detection for unmanned aerial vehicles,” *Drones*, vol. 7, no. 5, p. 310, 2023, doi: 10.3390/drones7050310.

[15] A. Ishtiaq, Z. Saeed, M. U. Khan, A. Samer, M. Shabbir, and W. Ahmad, “Fall detection, wearable sensors and artificial intelligence: A short review,” *Journal on Advanced Research in Electrical Engineering (JAREE)*, vol. 6, no. 2, 2022.

[16] A. Raza, Z. Saeed, A. Aslam, S. M. Nizami, K. Habib, and A. N. Malik, “Advances, applications, and challenges of lithography techniques,” in *Proc. 2024 5th Int. Conf. on Advancements in Computational Sciences (ICACS)*, Lahore, Pakistan, 2024, pp. 1–6, doi: 10.1109/ICACS60934.2024.10473245.

[17] Z. Saeed, M. N. M. Awan, and M. H. Yousaf, “A robust approach for small-scale object detection from aerial-view,” in *Proc. Int. Conf. on Digital Image Computing: Techniques and Applications (DICTA)*, 2022, pp. 1–7.

[18] U. Nawaz, Z. Saeed, and K. Atif, “A novel transformer-based approach for adult facial emotion recognition,” *IEEE Access*, vol. 13, pp. 56485–56508, 2025, doi: 10.1109/ACCESS.2025.3555510.

[19] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, “Road crack detection using deep convolutional neural network,” in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2016, pp. 3708–3712, doi: 10.1109/ICIP.2016.7533052.

[20] A. Ahmadi, S. Khalesi, and A. Golroo, “An integrated machine learning model for automatic road crack detection and classification in urban areas,” *International Journal of Pavement Engineering*, 2021, doi: 10.1080/10298436.2021.1905808.

[21] H. Oliveira and P. L. Correia, “CrackIT—An image processing toolbox for crack detection and characterization,” in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2014, pp. 798–802, doi: 10.1109/ICIP.2014.7025160.

[22] S. Bang, S. Park, H. Kim, and H. Kim, “Encoder–decoder network for pixel-level road crack detection in black-box images,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 34, no. 8, pp. 713–727, 2019, doi: 10.1111/mice.12469.

[23] P. Prasanna, K. J. Dana, N. Gucunski, B. Basily, S. La, R. S. Lim, and H. Parvardeh, “Automated crack detection on concrete bridges,” *IEEE Trans. Automation Science and Engineering*, vol. 13, no. 2, pp. 591–599, 2016, doi: 10.1109/TASE.2014.2354314.

[24] L. Lei, J. Fan, and Y. Wang, “Crack detection of concrete structures using image processing techniques under limited data conditions,” *Journal of Computing in Civil Engineering*, vol. 32, no. 4, 2018.

[25] Y. Yu, Z. Zhang, J. Wang, and X. Chen, “UAV-based crack detection using YOLOv4 deep learning model,” *Automation in Construction*, vol. 125, p. 103606, 2021, doi: 10.1016/j.autcon.2021.103606.

[26] L. You, X. Chen, and H. Liu, “A review of vision-based pavement crack detection methods,” *Measurement*, vol. 104, pp. 79–93, 2017, doi: 10.1016/j.measurement.2017.03.010.

[27] Y. Du, H. Wang, J. Sun, and X. Zhang, “Real-time pavement distress detection and classification using YOLO-based deep learning framework,” *Automation in Construction*, vol. 125, p. 103609, 2021, doi: 10.1016/j.autcon.2021.103609.

[28] S. Safaei, M. S. M. Nasir, and A. Golroo, “Pavement crack detection and quantification using crack pixel density-based image processing,” *International Journal of Pavement Engineering*, vol. 23, no. 6, pp. 1935–1947, 2022, doi: 10.1080/10298436.2021.1905821.

[29] P. Sane and R. Agrawal, “Pixel normalization from numeric data as input to neural networks for machine learning and image processing,” in *Proc. 2017 Int. Conf. on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India, 2017, pp. 2221–2225, doi: 10.1109/WiSPNET.2017.8300154.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[31] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, “A comprehensive survey on transfer learning,” *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021, doi: 10.1109/JPROC.2020.3004555.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[33] S.-H. Tsang, “Review: Inception-v4—Evolved from GoogLeNet, merged with ResNet,” *Towards Data Science*, 2018.

[34] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, “SqueezeNext: Hardware-aware neural network design,” in *Proc. IEEE CVPR Workshops*, 2018, pp. 1719–1728.

[35] S. Dorafshan, R. J. Thomas, and M. Maguire, “SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks,” *Data in Brief*, vol. 21, pp. 1664–1668, 2018, doi: 10.1016/j.dib.2018.11.015.

[36] H. Jindal, N. Sardana, and R. Mehta, “Analyzing performance of deep learning techniques for web navigation prediction,” *Procedia Computer Science*, vol. 167, pp. 1739–1748, 2020, doi: 10.1016/j.procs.2020.03.384.

[37] K. Nakamura, B. Derbel, K. J. Won, and B. W. Hong, “Learning-rate annealing methods for deep neural networks,” *Electronics*, vol. 10, no. 16, 2021, doi: 10.3390/electronics10162029.

[38] D. R. Wilson and T. R. Martinez, “The need for small learning rates on large problems,” in *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, vol. 1, 2001, pp. 11–19, doi: 10.1109/IJCNN.2001.939002.

[39] S. Bock and M. Weis, “A proof of local convergence for the Adam optimizer,” in *Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8852239.

[40] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, “Convolutional neural network (CNN) for image detection and recognition,” in *Proc. 2018 First Int. Conf. on Secure Cyber Computing and Communication (ICSCCC)*,Jalandhar, India, 2018, pp. 278–282, doi: 10.1109/ICSCCC.2018.8703316.

[41] M. Ahmadi, S. Vakili, J. M. P. Langlois, and W. Gross, “Power reduction in CNN pooling layers with a preliminary partial computation strategy,” *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)*, Montreal, QC, Canada, 2018, pp. 125–129, doi: 10.1109/NEWCAS.2018.8585433.

[42] B. Li, K. C. P. Wang, A. Zhang, E. Yang, and G. Wang, “Automatic classification of pavement crack using deep convolutional neural network,” *International Journal of Pavement Engineering*, vol. 21, no. 4, pp. 457–463, 2020, doi: 10.1080/10298436.2018.1485917.