Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
2025, vol. 30, no. 1, pp. 187–194
https://doi.org/10.2478/acss-2025-0020
https://content.sciendo.com

1862
RIGA TECHNICAL UNIVERSITY

CrackNet-VGG: A Deep Learning Framework for Automated Detection of Surface Cracks in Concrete Structures

Saiqa Murtaza[1], Muhammad Imran[2*], Muhammad Rizwan Rashid Rana[3]
[1–3]*Department of Robotics & Artificial Intelligence, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan*

***Abstract* – The reliable and timely detection of cracks in concrete structures is essential for maintaining the safety, functionality, and longevity of civil infrastructure, including buildings, bridges, highways, and dams. Structural cracks can emerge due to multiple factors such as material fatigue, environmental stressors, seismic activity, and thermal expansion, necessitating accurate and efficient monitoring systems. Traditional inspection techniques, including manual visual inspection and non-destructive testing, are labour-intensive, prone to subjectivity, and often lack scalability. To address these limitations, the research presents CrackNet-VGG, a deep learning-based framework that leverages the VGG16 convolutional neural network architecture for automatic binary classification of surface cracks in concrete images. The proposed model leverages transfer learning by fine-tuning the VGG16 architecture on concrete surface datasets, utilising its convolutional layers for robust feature extraction and its fully connected layers for final binary classification. The model is trained and evaluated on publicly available benchmark datasets, categorised into two classes: cracked and non-cracked surfaces. Experimental results demonstrate that CrackNet-VGG achieves a high classification accuracy of 96.07 %, with 95.57 % precision and 95.31 % recall, surpassing several baseline deep learning models in terms of accuracy. These results validate the applicability of CrackNet-VGG as an effective solution for automated concrete crack detection in real-world scenarios.**

***Keywords* – Concrete crack detection, convolutional neural network, deep learning, structural health monitoring, VGG16 architecture.**

I. Introduction

The safety and longevity of civil infrastructure, such as bridges, buildings, roads, and dams, are critical to public welfare and economic stability [1]. However, these structures are prone to deterioration due to factors such as aging, insufficient maintenance, outdated design standards, and exposure to environmental stresses or natural disasters [2]. Events like earthquakes, extreme temperature fluctuations, and windstorms can exacerbate structural weaknesses, leading to cracks, corrosion, and other critical defects. Traditionally, visual inspection has been the most common approach for assessing the condition of concrete structures [3]. While effective to some extent, manual inspection is time-consuming, labour-intensive, and often subjective, especially when dealing with large-scale or difficult-to-access areas. Recent advancements in technology have introduced automated inspection methods, including the use of Unmanned Aerial Vehicles (UAVs) and deep learning techniques, which offer significant improvements in accuracy, efficiency, and reliability for detecting surface defects in concrete infrastructure [4].

Timely and accurate detection of cracks is among the most important components when monitoring concrete infrastructure, as these defects indicate the safety, reliability, and serviceability of a structure [5]. Additionally, natural disasters such as earthquakes and environmental stressors such as extreme weather can exacerbate structural degradation and, additionally, significantly disrupt transportation and communications systems, ultimately hindering emergency response capability and delaying community recovery. Hence, designing and maintaining infrastructure to withstand one or several stressors is vitally important to minimise damage and conserve lives. However, critical, crack detection is still mostly done by manual inspection, which is subject to many constraints. Manual inspection is not only labour-intensive but also time-consuming and generally needs a sizable team to survey large areas [6]. Consequently, this demand for automated solutions has spurred possibilities of creating solutions to rectify the inefficiencies and challenges conventional approaches demonstrate.

Investment in infrastructure can cause many problems, including overcrowding, traffic jams, and power outages. These issues negatively affect economic growth and overall quality of life for both individuals and communities [7]. Structural degradation is often apparent by visible indicators, such as cracking, scaling, spalling, and efflorescence. Generally, inspections are performed based on inspectors' knowledge, experience, and engineering abilities by subjectively observing these defects [8]. This type of inspection is subjective, labour-

* Corresponding author's e-mail: dr.imran@szabist-isb.edu.pk
Article received 2025-07-28; accepted 2025-11-27

intensive and time-consuming; additionally, it will require inspection of not easily accessible parts of complex structures. This presents real challenges [9]. To address these limitations, this study proposes a semi-automated inspection model that leverages image processing and deep learning techniques to detect surface flaws in concrete structures. Manual inspections not only cause disruptions to traffic flow leading to delays and congestion, but also pose serious safety risks to workers, particularly when conducted on active roadways or in hazardous conditions. Moreover, human factors such as fatigue, distraction, and personal bias can adversely affect the accuracy and consistency of the inspection outcomes [10].

To address the limitations of traditional manual pavement inspection methods, advanced technologies such as Machine Learning (ML) and Computer Vision (CV) offer promising solutions for automating the pavement crack detection process. These technologies can efficiently analyse high-resolution images and sensor data acquired from inspection vehicles, drones, or stationary systems to provide detailed and objective assessments of pavement conditions. Automated detection systems not only enable the monitoring of extensive road networks in significantly less time but also minimise dependence on manual labour, thereby reducing operational costs and associated human errors. Moreover, such systems enhance safety by limiting the exposure of workers to hazardous road environments and improving the accuracy and consistency of the collected data. To meet these needs, this study introduces CrackNet-VGG, a robust and lightweight deep learning model designed for real-time pavement crack detection. The main contributions of this study are as follows:

- A novel crack detection model named CrackNet-VGG is proposed, which integrates a VGG16-based encoder with a customised decoder using skip connections to enhance feature preservation and segmentation accuracy.
- Extensive comparative analysis and ablation studies validate the effectiveness of the proposed model, achieving superior accuracy, precision, and recall compared to existing baselines.

The rest of the paper is organised in the following way. Section 2 reviews the existing literature on crack detection and segmentation. Section 3 presents the suggested CrackNet-VGG architecture and a justification of the model's architecture. Section 4 describes the design of the experimental setup, results and their discussion, as well as comparative analysis and ablation studies. Lastly, Section 5 summarises the paper and discusses possible future work.

II. Literature Review

Automated crack detection has increasingly become a focus of research in the area of Structural Health Monitoring (SHM) due to the considerable need for accurate, scalable, and timely methods to evaluate and monitor infrastructure. Conventional inspection methods often require reliance on manual labour and are, therefore, not only subjective but also time-consuming and subject to human error. Recent literature has shifted to developing and applying computer vision and deep learning methods to detect and segment cracks in pavements and concrete structures.

Ni et al. proposed an automated solution using a deep learning method that integrates multiscale feature representation to perform pixel-level analysis at different depths [11]. They used a GoogleNet-based convolutional neural model designed for differentiating between patterns of cracks. To improve the localisation capabilities of the detection method, the authors established a hierarchical level of merging features between layers. Another study reported an overall detection accuracy and precision score of 80.13 %. Sharma et al. [12] recognised the importance of continuous monitoring of cracks for their development, depth, and severity, since this knowledge is essential for infrastructure maintenance. They also sought crack detection techniques that could achieve both speed and accuracy. Zou et al. [13] introduced DeepCrack, a deep learning-based system for crack detection in pavement images. This model employs convolutional neural networks to recognise high-level structural patterns, such as ridges that relate to cracks. The DeepCrack framework extends SegNet's encoder-decoder architecture, fusing features extracted from layers in both the encoder and decoder models to get the most accurate representation of cracks in pavement images. Hsieh et al. [14] performed a review of image segmentation for cracks in pavement images with machine learning and deep learning models, including publicly available datasets. The models identified with the highest accuracy in the segmentation of cracks included Fully Convolutional Networks (FCNs) and U-Net models.

Nguyen et al. [15] released a two-stage CNN architecture to detect and segment road cracks at the pixel level. In the first stage, the model reduces noise and other unwanted artefacts, decreasing the search area to only possible locations of cracks as output. In the second stage, the model takes the refined first stage output and learns the contextual features of cracks properly. The authors reported F1-scores overall greater than 0.91 across 3 datasets, showing their model is reliable and capable. Another study, using a conventional Convolutional Neural Network (CNN) [16], which also used road cracks as test objects, produced a precision of 0.89 and a recall of 0.93. While the results are promising, the authors also acknowledged challenges with noisy backgrounds, which created inconsistencies in detection. Another model, using a Deep Encoder-Decoder architecture [17], produced a recall of 71.98 % and a precision of 77.68 %. While their model provided an end-to-end framework for segmentation and was overall usable, their model had moderate recall and low Intersection over Union (IoU), primarily with more complicated crack patterns.

Existing approaches have demonstrated varying degrees of success in crack detection. However, they still face several limitations, including poor performance in noisy environments, low precision for fine or narrow cracks, and inefficiencies in real-time inference. Additionally, many models struggle with generalizability due to limited datasets or lack of robustness against complex backgrounds. These gaps highlight the pressing need for a more accurate, efficient, and scalable

solution. To address these shortcomings, this study introduces CrackNet-VGG, a VGG16-based deep convolutional neural network optimised for binary classification of concrete surface cracks, offering superior accuracy, precision, and real-time applicability compared to existing methods.

III. The Proposed Model

The proposed crack detection and segmentation framework – CrackNet-VGG – is a hybrid encoder–decoder architecture following the VGG16 model as presented in Fig. 1. The architecture begins with input images that have undergone pre-processing with grayscale conversion, binarization, intensity normalization, and noise smoothing by way of filters. These pre-processing tasks improve the visual quality and structural consistency of the input data to ensure robust extraction of features for the encoder. In the encoder, VGG16 pre-trained on ImageNet is used to extract hierarchical features from input images. Convolutional blocks of VGG16 were used to build the encoder layers and to extract low- to high-level visual features that are required to detect fine structural cracks. Skip connections were used between the encoder and decoder layers using the corresponding layers to maintain spatial characteristics. The decoder is built using successive up-sampling operations with convolutional layers, which also reconstructs the spatial resolution of feature maps while progressively localising the crack regions. As these features are up-sampled, the skip connections to corresponding encoder outputs provide contextual knowledge and sharpen boundary localisation. The final output is a crack segmentation mask that classifies each pixel as a crack colour or background, thereby enabling the localisation of structural damage on concrete surfaces accurately and understandably at the pixel level.

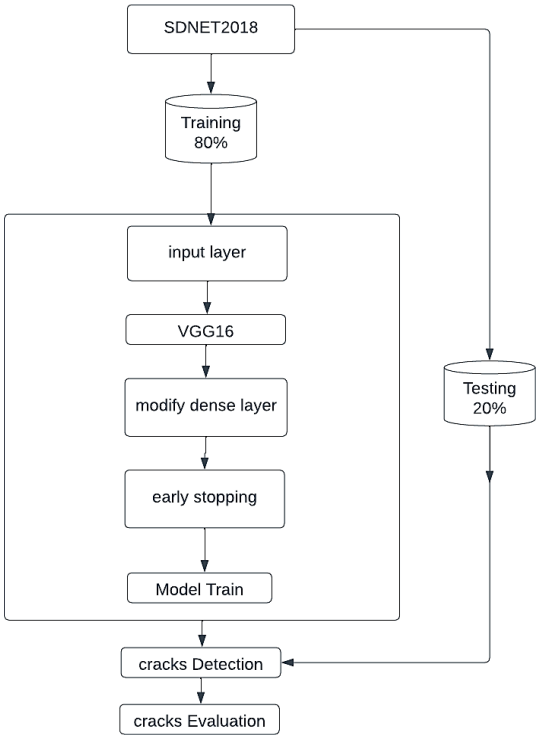

Fig. 1. Process flow of VGG16 for CrackNet-VGG.

A. Dataset Collection

The proposed model, CrackNet-VGG, for detecting and segmenting cracks within concrete surfaces was trained and evaluated utilising the two publicly available datasets: Concrete Damage Surface-I (CDS-I) and Concrete Damage Surface-II (CDS-II). CDS-I relates to the SDNET2018 dataset, which includes a total of 40 000 images divided equally across the two classes (20 000 images depicting cracked surfaces (positive class) and 20 000 images of non-cracked surfaces (negative class). The illuminated surface images in CDS-I were taken under wet and dry conditions, reflecting various lighting and surface compositions. CDS-I included no pre-applied data augmentation whatsoever. The 40 000 images in CDS-I contain preserved visible characteristics existing in the original data, maintaining realism.

The second dataset, CDS-II, deals with surface crack detection and is likewise a binary classification problem – cracked or not cracked images. This dataset contains additional variance in both surface texture and environmental variances to support training and increase the generalization of a trained model. Each dataset followed a standard train/test split of 80/20, using the majority of the data for training and only 20 % for testing. This standard split assures the model can be evaluated on generalization with new data. Data augmentation methods such as flipping horizontally and vertically, rotating, and zooming were used in the training to increase the diversity of the input image and avoid overfitting. Together these datasets acted as a good and varied dataset for performance evaluation of the proposed deep learning-based crack detection framework. The datasets are described in Table I.

TABLE I
Dataset Details

Dataset	Class	Total Images	URL
CDS-I	Cracked and Non- Cracked	40 000	https://tinyurl.com/3a7vyz75
CDS-II	Cracked and Non-Cracked	20 000	https://tinyurl.com/avjjcw65

B. Data Pre-processing

A systematic pre-processing pipeline is implemented to improve image quality, limit noise, and standardise the input format before images are fed into the deep learning model for crack detection. The raw crack images are obtained in the field and can sometimes contain several artefacts, such as surface scratches, variable lighting, and lens distortions [18]. Brightness differences from lens vignetting can affect the quality of images, and when diffusers are implemented, LED illumination can also cause this variance across the image plane [19]. Artefacts can blur fine details in crack images. If the model has not been designed with the artificial characteristics of the images, then artefacts can affect the model's performance.

In pre-processing, the first step was to create grayscale images from the raw Red-Green-Blue (RGB) images. The reasoning for converting to grayscale was twofold: from a processing standpoint, it also reduced the computational burden; and, from a feature selection standpoint, grayscale (intensity-based) information emphasises the structural aspects

of the fracture, which are needed to identify the cracks (the colour information is not as useful). After converting to grayscale, each grayscale image was thresholded to a binary format (two-tone) where the crack regions were represented in white (value =1) and the background was represented in black (value = 0). This binary format also helps enhance the contrast of the defect, highlight the difference between defective and non-defective regions, and make the results more visually effective. After applying thresholding, each image was normalized to a floating-point representation where the pixel values are described between 0 and 1 [20]. This was helpful in providing numerical stability to scale in the next layers of the neural network. After the normalization process, each image was smoothed using a filter (i.e., a Gaussian or a median filter) to help remove high-frequency noise and any small irrelevant artefacts that could be misidentified as crack features. Maintaining the boundaries of the cracks is paramount, but minimising information that could be misidentified is also important.

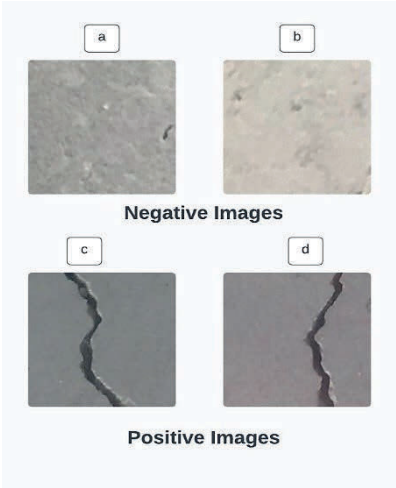

Fig. 2. Examples of the negative and positive images.

C. VGG16 Encoder (Pre-trained on ImageNet)

The VGG16 model is then incorporated as the foundational architecture, leveraging its pre-trained weights on the ImageNet dataset. The top layers of the model, responsible for generic image classification, are removed, and the remaining layers are frozen to retain the learned features. Subsequently, a custom dense layer tailored for binary classification is added to the model. This modification ensures that the model is equipped to distinguish between cracked and non-cracked surfaces. Figure 3 shows the model of VGG16.

The encoder component of the proposed hybrid model leverages the VGG16 architecture as a foundational backbone for hierarchical feature extraction. Originally introduced by Simonyan and Zisserman, VGG16 is a deep convolutional neural network composed of 13 convolutional layers and five max-pooling layers, arranged in a sequential manner with uniform 3×3 kernel filters and ReLU activation functions. Its structural simplicity and proven representational capacity make it a strong candidate for extracting low- to high-level visual features in crack detection tasks. The encoder component of the proposed hybrid model leverages the VGG16 architecture as a foundational backbone for hierarchical feature extraction.

Formally, given an input RGB image $I \in R^{224\times224\times3}$, the encoder processes it through a series of convolutional operation:

$$F_l = \mathrm{ReLU}\left(W_l * F_{l-1} + b_l\right), \tag{1}$$

where F_l denotes the output feature map at layer l, W_l, and b_l represent the learnable weights and biases of the l-th convolutional layer, and * denotes the convolutional operator. Each convolutional block is followed by a max-pooling operation that reduced the spatial resolution by factor of 2, progressively capturing higher level abstraction.

The encoder structure is partitioned into five convolutional blocks, each comprising two or three convolutional layers followed by a max-pooling layer. Table II summarises the output resolution at each block level and the associated skip connections utilised in the decoding phase.

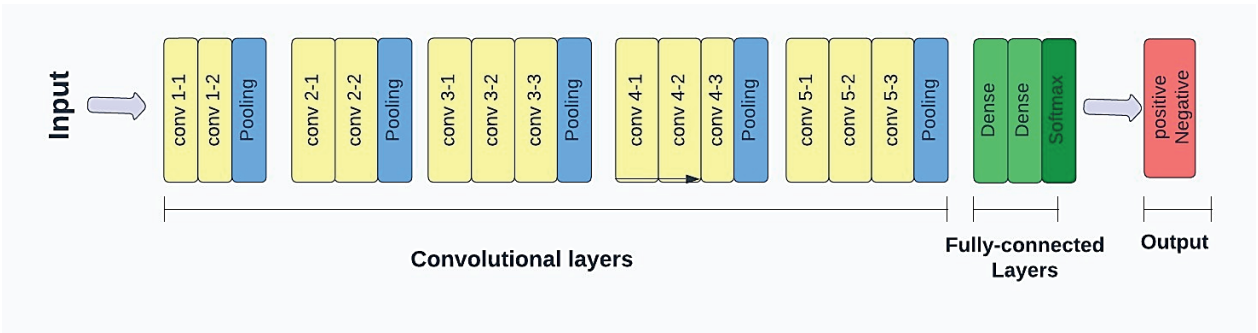

Fig. 3. The proposed VGG16 model structure.

TABLE II
VGG16-BASED ENCODER FEATURE MAP DIMENSIONS AND SKIP CONNECTION MAPPING

Block	Layer Depth	Output Size	Skip Connection
Block 1	2 Conv + Pool	112 × 112 × 64	✓
Block 2	2 Conv + Pool	56 × 56 × 128	✓
Block 3	3 Conv + Pool	28 × 28 × 256	✓
Block 4	3 Conv + Pool	14 × 14 × 512	✓
Block 5	3 Conv + Pool	7 × 7 × 512	✗

To preserve important spatial features lost during the down-sampling process, intermediate outputs from Blocks 1 to 4 are retained as skip connections and later fused with the corresponding up-sampling layers in the decoder. This approach enables the recovery of fine-grained boundary and texture information, which is critical for crack localisation. The encoder is initialised with weights pre-trained on the ImageNet dataset, enabling the model to benefit from generalized visual representations.

These pre-trained parameters $Ø_{VGG16}$ can either be fixed during training or fine-tuned to adapt to the specific distribution of crack images:

$$F_{\text{enc}} = \mathrm{E}\left(I;\ \varnothing_{\text{VGG16}}\right), F_{\text{enc}} \in R^{7\times7\times512}, \tag{2}$$

where E(·) denotes the feature extraction function implemented by the VGG16 encoder.

By leveraging transfer learning, the VGG16-based encoder facilitates faster convergence and improved generalization, particularly beneficial given the limited availability of annotated crack datasets.

D. Decoder Architecture with Skip Connections

Following the VGG16-based encoding stage, the decoder is designed to reconstruct high-resolution crack segmentation maps by progressively sampling the abstracted feature representations. Inspired by the U-Net architecture, the decoder employs a series of up-sampling and convolutional operations to recover spatial resolution while simultaneously integrating multi-scale contextual information through skip connections [21]. Each decoding stage performs the following operations: (i) up-sampling of the feature maps using nearest-neighbour interpolation or learnable transposed convolution, (ii) concatenation with the corresponding feature maps from the encoder (skip connections), and (iii) refinement through a sequence of convolutional layers activated by ReLU functions.

Formally, let $F_i^{(d)}$ denote the feature map at decoder stage i, and let s_i represent the skip connection from the encoder block i. The decoding operation can be described as follows:

$$F_i^{(d)} = \sigma\left(\mathrm{Conv}\left(\mathrm{Concat}\left(\mathrm{Up}\left(F_{i+1}^{(d)}\right), s_i\right)\right)\right), \tag{3}$$

where:

- Up(·) denotes an up-sampling operation (typically by a factor of 2);
- Concat(·) is the channel-wise concatenation of decoder and skip connection features;
- Conv(·) represents a 3×3 convolutional layer;
- σ(·) is the ReLU activation function.

The decoder mirrors the encoder in both structure and spatial resolution, progressively reconstructing the segmentation map through a series of up-sampling and convolutional operations. The deepest encoded representation, $F_{\text{enc}} \in R^{7\times7\times512}$, is initially up-sampled to a resolution of 14×14 and concatenated with the corresponding feature maps from Block 4 (s_4) via skip connections. This process continues through successive decoder stages, where feature maps are up-sampled and concatenated with skip connections from Blocks 3, 2, and 1, yielding progressively larger feature representations of dimensions 28×28, 56×56, and 112×112, respectively. A final up-sampling operation is applied to recover the full spatial resolution of 224×224, matching that of the input image.

At the final decoding stage, a 1×1 convolutional layer followed by a sigmoid activation function is used to project the multi-channel feature map into a single-channel probability map representing the likelihood of crack presence at each pixel. Mathematically, this is expressed as $Y' = \sigma\left(\mathrm{Conv}_{1\times1}\left(F_1^{(d)}\right)\right)$, where $Y' \in R^{224\times224\times1}$. Here, Y' denotes the predicted probability map, with each pixel value indicating the model's confidence in classifying that pixel as part of a crack. By integrating high-level semantic information from deeper layers with low-level spatial cues preserved through skip connections, the decoder effectively enhances the localisation of crack boundaries. This architectural strategy significantly improves the model's capacity to detect fine-grained and fragmented structural details that are often subtle and easily missed by conventional approaches.

E. Crack Segmentation Mask (Output)

The final output of the proposed hybrid model is a pixel-wise binary segmentation map that delineates crack regions from the background. This mask is derived from the final decoder feature map through a 1×1 convolutional layer followed by a sigmoid activation function, which maps the multi-channel feature representation into a single-channel probability distribution over the spatial domain of the input image. Let $F_1^{(d)} \in R^{224\times224\times C}$ denote the last decoded feature map, where C is the number of feature channels. The final segmentation mask Y' is computed as follows:

$$Y' = \sigma\left(W_{\text{out}} * F_1^{(d)} + b_{\text{out}}\right), \tag{4}$$

where:

- $W_{\text{out}} \in R^{1\times1\times C\times1}$ and $b_{\text{out}} \in R$ are the weights and bias of the 1×1 convolution layer;
- * denotes the convolution operation;
- σ(·) is the element-wise sigmoid activation function.

The resulting segmentation mask $Y' \in R^{224\times224\times1}$ contains values in the range [0, 1], representing the model's confidence in predicting each pixel as part of a crack. During inference, a threshold t is applied to convert the probabilistic output into a binary mask:

$$Y_{\text{bin}}(i,j) = \begin{cases} 1 & \text{if } Y'(i,j) \geq t, \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where (i, j) denotes the pixel coordinates in the image and t ∈ (0, 1) is a predefined threshold, commonly set to 0.5. The segmentation mask output enables precise localisation of cracks, even in cases where cracks are thin, discontinuous, or low in contrast. The binary output can be further processed using morphological operations or component analysis to quantify crack length, area, or orientation features critical for structural assessment applications.

IV. Results

The proposed model was implemented using Python and TensorFlow on a workstation equipped with an Intel Core i7 processor, 32 GB of RAM, and an NVIDIA RTX 3080 GPU with 10 GB of VRAM, enabling efficient parallel training. During training, the input images were resized to 224×224 pixels to match the VGG16 input requirements. A batch size of 16 was chosen to balance memory usage and convergence speed. The model was trained using the Adam optimizer with an initial learning rate of 0.0001, which was reduced dynamically based on validation loss. The binary cross-entropy loss function was used for segmentation, and early stopping with a patience of 10 epochs was applied to prevent overfitting. The training was conducted over 40 epochs with data augmentation techniques such as rotation, flipping, and zooming to enhance the model's robustness to image variability. These hyperparameters were empirically selected based on multiple experiments to ensure optimal segmentation performance.

The crack detection model achieved good results in overall performance across both benchmark datasets as reported in Table III. The CDS-I dataset accuracy was 95.68 % with a precision of 95.12 % and a recall of 94.87 %, which confirmed well-balanced and identifiable cracked surfaces and non-cracked surfaces. The CDS-II dataset demonstrated an even higher accuracy of 96.45 %, a precision of 96.02 % and a recall of 95.74 %. These results evidently indicate the model's solid ability to generalize across various surface conditions and image specifications while minimising the number of false positives and false negatives in crack detection.

TABLE III
Results in Terms of Accuracy, Precision and Recall

Dataset	Accuracy (%)	Precision (%)	Recall (%)
CDS-I	95.68	95.12	94.87
CDS-II	96.45	96.02	95.74

The performance of the proposed crack detection model was further evaluated using confusion matrices for two datasets: CDS-I and CDS-II, as shown in Fig. 4. In the case of CDS-I, the model effectively distinguished between cracked and non-cracked surfaces, demonstrating a strong ability to correctly classify both categories. The matrix shows a high number of true positives and true negatives, with relatively low instances of false positives and false negatives, indicating minimal misclassification. Similarly, the CDS-II confusion matrix reflects a consistent pattern of reliable detection, with the model accurately identifying most of the cracked and non-cracked surfaces. The balanced distribution of correct predictions across both classes highlights the model's robustness and generalization capability across different pavement datasets. These results reinforce the potential of the proposed system in supporting efficient and automated pavement condition monitoring.

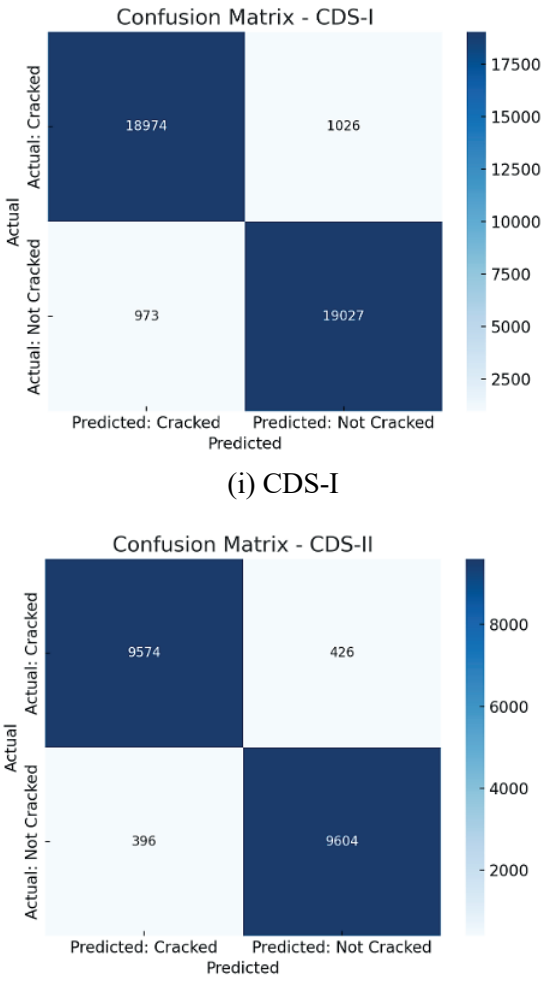

Fig. 4. Confusion matrix of CrackNet-VGG on CDS-I and CDS-II.

The training and validation loss curves of the proposed CrackNet-VGG model were analysed over 40 epochs to assess its learning behaviour and convergence, as illustrated in Fig. 5. The gradual decline and stabilization of both curves indicate effective optimisation, confirming the model's stability and strong generalization capability across the CDS-I and CDS-II datasets.

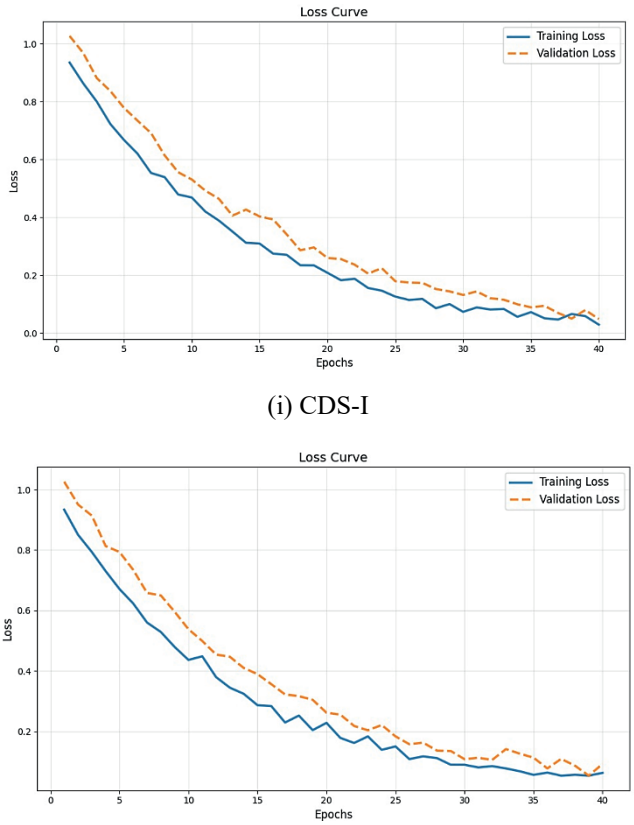

Fig. 5. Training and validation loss of CDS-I and CDS-II.

The comparative analysis evaluates the effectiveness of the proposed model against three established baseline methods to highlight its advancements in crack detection accuracy and reliability. While baseline approaches demonstrate promising results using fuzzy clustering (Baseline 1 [22]), transformer-based architecture (Baseline 2 [23]), and enhanced background classification (Baseline 3 [24]), they each exhibit limitations in handling low-contrast features, scalability to high-resolution data, or false detection rates. In contrast, the proposed VGG16-based encoder-decoder model with skip connections outperforms these methods by achieving superior pixel-level segmentation accuracy, especially under complex and noisy surface conditions, as evidenced by its performance on both CDS-I and CDS-II datasets. The results are shown in Fig. 6.

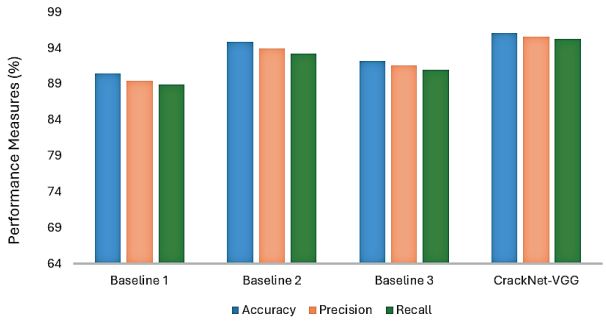

Fig. 6. Comparative analysis of CrackNet-VGG with baseline approaches.

The comparative analysis demonstrates the superior performance of the proposed CrackNet-VGG model over existing baseline methods. Baseline 1, based on fuzzy C-Means clustering, achieved an accuracy of 90.45 %, precision of 89.45 %, and recall of 88.95 %, indicating limitations in detecting fine cracks, especially in low-contrast images. Baseline 2, which employs a two-stage crack transformer-based framework, performed better with 94.91 % accuracy, 94.01 % precision, and 93.25 % recall, but its complexity and high dependency on pre-selection limit its generalizability. Baseline 3, which introduces background classification into CNNs, recorded an accuracy of 92.24 %, precision of 91.58 %, and recall of 91.02 %. In contrast, the proposed CrackNet-VGG model outperformed all baselines by achieving an accuracy of 96.07 %, precision of 95.57 %, and recall of 95.31 %. These results validate the effectiveness of the encoder-decoder architecture with skip connections in capturing both high-level and fine-grained features, enabling more accurate and robust crack segmentation across various surface conditions.

An ablation study is presented in Table IV to evaluate the learning performance of the respective elements in CrackNet-VGG with consistent removal/modification of elements in the architecture. Each of the architecture modifications resulted in lower performance, as highlighted in Table IV. While the complete version of the CrackNet-VGG implementation, which has skip connections, a pre-trained VGG encoder, and final decoder refinement elements with normalization and activation functions, achieved the highest performance illustrating an accuracy of 96.07 % (precision: 95.57 %, recall: 95.31 %), removing the skip connections resulted in a drop in accuracy (93.22 %) and recall (91.88 %), indicating the importance of multi-scale contextual information required for optimal crack segmentation.

TABLE IV
ABLATION STUDY

Variant	Accuracy	Precision	Recall
CrackNet-VGG (Full Modal)	96.07	95.57	95.31
Without Skip Connections	93.22	92.41	91.88
Without Pre-trained VGG (Random Init)	92.35	91.12	90.78
Basic Decoder (No Norm / Activations)	91.88	90.76	90.02

Removing pre-trained VGG weights (initialising the encoder at random) reduced our accuracy to 92.35 %, suggesting that large-scale image datasets play an important role in transfer learning to improve feature extraction mechanisms for the crack detection task, and also replacing the decoder with a basic decoder that has no normalization or non-linear activations reduced performance (accuracy: 91.88 %), thus indicating that the decoder is necessary for refinement steps that should happen in order to preserve the fine structural details of cracks. Overall, this demonstrates the validity of each element of the CrackNet-VGG architecture and how the enhancements of segmentations with architectural refinements support robust and accurate crack detection.

V. Conclusion

This study introduced CrackNet-VGG, a deep convolutional neural network model based on the VGG16 architecture, specifically optimised for the binary classification of cracks in concrete surfaces. The model addresses several limitations of traditional inspection methods and prior deep learning approaches, including subjectivity, time consumption, low precision in noisy backgrounds, and high inference time. The results of the proposed model confirm its effectiveness in real-time crack detection scenarios and highlight its potential for integration into practical structural health monitoring systems. Future work will focus on extending CrackNet-VGG to support multi-class classification for detecting various types of surface defects such as spalling, efflorescence, and scaling. Additionally, integration with drone-based image acquisition systems, edge computing devices, and real-time video processing pipelines will be explored to enable large-scale deployment in field environments. Incorporating transfer learning with domain-specific data and enhancing model robustness under varying lighting and environmental conditions will also be key areas for further research.

References

[1] G. Zhu *et al*., "A lightweight encoder–decoder network for automatic pavement crack detection," *Comput.-Aided Civ. Infrastruct. Eng*., vol. 39, no. 12, pp. 1743–1765, Jun. 2024. https://doi.org/10.1111/mice.13103

[2] Z. Qu, C. Y. Wang, S. Y. Wang, and F. R. Ju, "A method of hierarchical feature fusion and connected attention architecture for pavement crack detection," *IEEE Trans. Intell. Transp. Syst*., vol. 23, no. 9, pp. 16038–16047, Feb. 2022. https://doi.org/10.1109/TITS.2022.3147669

[3] H. Feng *et al*., "GCN-based pavement crack detection using mobile LiDAR point clouds," *IEEE Trans. Intell. Transp. Syst*., vol. 23, no. 8, pp. 11052–11061, Aug. 2021. https://doi.org/10.1109/TITS.2021.3099023

[4] M. A. M. Khan, S. H. Kee, A. S. K. Pathan, and A. A. Nahid, "Image processing techniques for concrete crack detection: A scientometrics literature review," *Remote Sens*., vol. 15, no. 9, May 2023, Art. no. 2400. https://doi.org/10.3390/rs15092400

[5] X. Xiang, Z. Wang, and Y. Qiao, "An improved YOLOv5 crack detection method combined with transformer," *IEEE Sens. J*., vol. 22, no. 14, pp. 14328–14335, Jun. 2022. https://doi.org/10.1109/JSEN.2022.3181003

[6] S. Xiao *et al*., "Pavement crack detection with hybrid-window attentive vision transformers," *Int. J. Appl. Earth Obs. Geoinf*., vol. 116, Feb. 2023, Art. no. 103172. https://doi.org/10.1016/j.jag.2022.103172

[7] B. Xu and C. Liu, "Pavement crack detection algorithm based on generative adversarial network and convolutional neural network under small samples," *Measurement*, vol. 196, Jun. 2022, Art. no. 111219. https://doi.org/10.1016/j.measurement.2022.111219

[8] V. P. Golding, Z. Gharineiat, H. S. Munawar, and F. Ullah, "Crack detection in concrete structures using deep learning," *Sustainability*, vol. 14, no. 13, Jul. 2022, Art. no. 8117. https://doi.org/10.3390/su14138117

[9] H. Li, J. Zong, J. Nie, Z. Wu, and H. Han, "Pavement crack detection algorithm based on densely connected and deeply supervised network," *IEEE Access*, vol. 9, pp. 11835–11842, Jan. 2021. https://doi.org/10.1109/ACCESS.2021.3050401

[10] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," *Alex. Eng. J*., vol. 57, no. 2, pp. 787–798, Jun. 2018. https://doi.org/10.1016/j.aej.2017.01.020

[11] R. Sharma, D. A. Potnis, and V. Chourasia, "Review of image-based concrete crack detection," in *Proc. 2021 Int. Conf. Adv. Technol., Manage. Educ. (ICATME)*, Bhopal, India, Jan. 2021.

[12] 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, Feb. 2012. https://doi.org/10.1016/j.patrec.2011.11.004

[13] Y.-A. Hsieh and Y. J. Tsai, "Machine learning for crack detection: Review and model performance comparison," *J. Comput. Civ. Eng*., vol. 34, no. 5, Jul. 2020, Art. no. 04020038. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000918

[14] N. H. T. Nguyen, S. Perry, D. Bone, H. T. Le, and T. T. Nguyen, "Two-stage convolutional neural network for road crack detection and segmentation," *Expert Syst. Appl*., vol. 186, Dec. 2021, Art. no. 115718. https://doi.org/10.1016/j.eswa.2021.115718

[15] J. Li, C. Yuan, X. Wang, G. Chen, and G. Ma, "Semi-supervised crack detection using segment anything model and deep transfer learning," *Autom. Constr*., vol. 170, Feb. 2025, Art. no. 105899. https://doi.org/10.1016/j.autcon.2024.105899

[16] T. Han, D. Jiang, Q. Zhao, L. Wang, and K. Yin, "Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery," *Trans. Inst. Meas. Control*., vol. 40, pp. 2681–2693, 2018. https://doi.org/10.1177/0142331217708242

[17] Q. Yuan, Y. Shi, and M. Li, "A review of computer vision-based crack detection methods in civil infrastructure: Progress and challenges," *Remote Sens*., vol. 16, no. 16, Aug. 2024, Art. no. 2910. https://doi.org/10.3390/rs16162910

[18] H. Kaveh and R. Alhajj, "Recent advances in crack detection technologies for structures: A survey of 2022–2023 literature," *Front. Built Environ*., vol. 10, Jul. 2024, Art. no. 1321634. https://doi.org/10.3389/fbuil.2024.1321634

[19] S. Duan *et al*., "Tunnel lining crack detection model based on improved YOLOv5," *Tunn. Undergr. Space Technol*., vol. 147, May 2024, Art. no. 105713. https://doi.org/10.1016/j.tust.2024.105713

[20] A. Hussain, K. N. Qureshi, R. W. Anwar, and A. Aslam, "A novel SCD11 CNN model performance evaluation with inception V3, VGG16 and ResNet50 using surface crack dataset," in *Proc. 2024 2nd Int. Conf. Unmanned Vehicle Syst. (UVS)*, Oman, Feb. 2024, pp. 1–7. https://doi.org/10.1109/UVS59630.2024.10467149

[21] A. Hussain and A. Aslam, "Ensemble-based approach using inception V2, VGG-16, and Xception convolutional neural networks for surface cracks detection," *J. Appl. Res. Technol*., vol. 22, no. 4, pp. 586–598, Aug. 2024. https://doi.org/10.22201/icat.24486736e.2024.22.4.2431

[22] M. Bhardwaj, N. U. Khan, and V. Baghel, "Fuzzy C-Means clustering based selective edge enhancement scheme for improved road crack detection," *Eng. Appl. Artif. Intell*., vol. 136, Oct. 2024, Art. no. 108955. https://doi.org/10.1016/j.engappai.2024.108955

[23] N. Jayanthi, T. Ghosh, R. K. Meena, and M. Verma, "Length and width of low-light, concrete hairline crack detection and measurement using image processing method," *Asian J. Civ. Eng*., vol. 25, no. 3, pp. 2705–2714, Dec. 2024. https://doi.org/10.1007/s42107-023-00939-0

[24] S. Matarneh, F. Elghaish, F. P. Rahimian, E. Abdellatef, and S. Abrishami, "Evaluation and optimisation of pre-trained CNN models for asphalt pavement crack detection and classification," *Autom. Constr*., vol. 160, Apr. 2024, Art. no. 105297. https://doi.org/10.1016/j.autcon.2024.105297

Saiqa Murtaza is a graduate of the Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad. Her research interests include machine learning and data mining.
E-mail: saiqamurtaza95@gmail.com
ORCID iD: https://orcid.org/0009-0001-9839-0048

Muhammad Imran is an Associate Professor at the Department of Robotics & Artificial Intelligence, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad. His research interests include computer vision, deep learning, etc.
E-mail: dr.imran@szabist-isb.edu.pk
ORCID iD: https://orcid.org/0000-0003-1860-0736

Muhammad Rizwan Rashid Rana is an Assistant Professor at the Department of Robotics & Artificial Intelligence, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad. His research interests include natural language processing, deep learning etc.
E-mail: rizwanrana315@gmail.com
ORCID iD: https://orcid.org/0000-0002-2382-1800