Bridge Crack Detection and Classification Using CrackDet-ViT: A Vision Transformer and CNN-Based Segmentation Framework

Jibin Jacob Mani*

*Kristu Jayanti (Deemed to be University), Bengaluru, India; jibin.jm@kristujayanti.com

*Corresponding Author: jibin.jm@kristujayanti.com

DOI: https://doi.org/10.30212/JITI.202604.004

Submitted: Nov. 21, 2025 Accepted: Jan. 07, 2026

ABSTRACT

Bridge crack detection is a critical task in structural health monitoring. Traditional manual inspection methods suffer from inefficiency and issues such as false positives and missed detections. However, existing automated models still face limitations in handling complex backgrounds and multi-scale cracks. Therefore, there is a need for a high-accuracy crack detection method. In this paper, we propose CrackDet-ViT, a bridge crack detection and segmentation model that integrates RegNet, ViT, and Mask R-CNN. The model uses RegNet to extract local features, ViT to capture global information, and Mask R-CNN for crack object detection and pixel-level segmentation, thereby improving detection accuracy and segmentation performance. Experimental results show that CrackDet-ViT achieves a mean Average Precision (mAP) of 87.5% on the SDNET2018 dataset and 84.7% on the Kaggle - Crack Detection Challenge dataset, outperforming existing models. Overall, CrackDet-ViT demonstrates excellent performance and robustness, making it suitable for bridge crack detection in complex environments.

Keywords: Structural health, CNN, ViT, Instance segmentation, Crack detection, Deep learning

1. Introduction

Bridges are subject to various factors over time, such as load-bearing stresses and environmental corrosion, which can lead to the formation of cracks that can jeopardize structural safety. Therefore, early detection and classification of cracks are crucial for bridge maintenance. Traditional detection methods mainly rely on manual inspections and sensor-based monitoring [1]. Manual Inspection is influenced by human subjectivity, resulting in low efficiency and inconsistent accuracy, while sensor-based approaches, although capable of providing high-precision data, are costly, complex to deploy, and difficult to scale [2][3].

Deep learning has significantly advanced the automation of crack detection, with Convolutional Neural Networks (CNNs) excelling in image feature extraction and improving detection accuracy [4]. Despite these advancements, challenges remain [5]. CNNs primarily rely on local convolution operations, which limits their ability to capture the global structure of cracks, often leading to false positives or missed detections in complex backgrounds [6]. Object detection models can identify crack regions, but typically output only bounding boxes, lacking precise boundary information.

Semantic segmentation models offer pixel-level detection but often struggle with small cracks and noise robustness. Consequently, an effective solution must combine object detection and segmentation capabilities while enhancing global feature perception [7][8].

To address these challenges, we propose CrackDet-ViT, a bridge crack detection framework that integrates CNNs and Transformers, and employs Mask R-CNN for crack object detection and segmentation, thereby improving both accuracy and robustness [9]10]. The framework utilizes RegNet as the CNN backbone to extract local features and introduces ViT to model global features, enhancing crack shape perception. Additionally, Mask R-CNN combines Region Proposal Network (RPN) to generate candidate regions and a fully Convolutional Network (FCN) to predict crack masks, achieving pixel-level crack segmentation [11]. This approach not only identifies crack location and categories but also generates complete crack contours, significantly improving detection performance. The main contributions of this paper are as follows:

- Proposed CrackDet-ViT: A framework that integrated local features from RegNet and global features from ViT, enhancing the accuracy and robustness of crack detection.
- Adopted Mask R-CNN for crack object detection and instance segmentation, enabling precise localization and shape analysis.
- Optimized feature fusion strategies: Introduced multi-scale feature fusion and attention mechanisms to improve the model's adaptability in complex environments.

2. Related Work

2.1 Traditional Methods of Economic Cycle Forecasting

Traditional bridge crack detection methods mainly include manual inspection, image processing techniques, sensor monitoring, machine learning approaches, and early deep learning models [12][13][14]. Manual inspection relies on inspectors visually examining bridges or using tools to measure crack dimensions. However, this approach is highly subjective, inefficient, and prone to false positives and missed detections [15]. Image processing techniques can detect cracks under ideal conditions, but are highly sensitive to complex backgrounds and varying lighting conditions, limiting their generalization capability [16]. Sensor monitoring methods, such as strain gauges, fiber Bragg grating sensors, and ultrasonic sensors, provide high accuracy but are expensive to deploy, difficult to install, and not easily scalable [17]. Machine learning methods use manually extracted features for crack classification. While these methods improve detection automation to some extent, they heavily rely on feature engineering, which limits their generalization ability [18]. The rise of deep learning has advanced automatic crack detection, with early CNN models able to learn crack features automatically, avoiding manual feature engineering. However, these models have high computational complexity, tend to overfit small datasets, and limited capability in modeling crack geometrics [7].

The CrackDet-ViT model combines RegNet for local feature extraction, ViT for global feature capture, and both crack detection and pixel-level segmentation. It accurately identifies crack shapes and enhances robustness in complex backgrounds. This approach offers a comprehensive solution for bridge crack detection.

2.2 The Application of Deep Learning in Crack Detection

In recent years, numerous studies have explored deep learning techniques to improve crack

detection accuracy and robustness [19][20]. Deep CNN-based approaches automatically learn crack features from images to detect and classify cracks [21][22]. These methods perform well under simple conditions; however, their performance often degrades in complex backgrounds or noisy environments. Other methods use U-Net networks for semantic segmentation of cracks, predicting crack regions on a pixel-by-pixel basis. This allows for more accurate delineation of crack boundaries [23]. Especially in noisy scenes, where false positives or missed detections are more likely. Additionally, some research has applied object detection techniques to locate the positions of cracks for recognition [24]. These methods handle large cracks well, but their performance on accurately segmenting small cracks is subpar, and they only output bounding boxes, lacking fine-grained detail restoration. ResNet based architectures have also been applied to crack detection due to their deep hierarchical feature extraction capability [25]. Although ResNet improves representation learning, it still faces challenges in extracting fine crack details and maintaining robustness. Other approaches combine traditional image processing with deep learning by first extracting edge information of cracks and then applying deep learning for classification [26][27]. In addition, due to its deep network structure, ResNet can learn more hierarchical features. Some studies have applied it to crack detection, which has improved the ability of feature extraction, but still has shortcomings in crack detail extraction and robustness [28]. Another method combines traditional image processing techniques with deep learning, by first extracting edge information of cracks and then performing deep learning classification [29]. This type of method has improved the detection ability of small cracks, but its ability to suppress noise is weak, and the image processing part often reduces the end-to-end efficiency of the system [30].

The CrackDet-ViT leverages RegNet for local feature extraction and ViT for global information, achieving improved accuracy and robustness, particularly in complex environments, while delivering precise crack boundaries.

2.3 The Application of Transformer in Computer Vision

In recent years, Transformers have gained significant attention in computer vision tasks [31]. Initially, CNNs became the mainstream architecture in image tasks due to their ability to extract local features [32]. However, CNNs have limited capability in modeling long-range dependencies, which restricts their performance in complex visual tasks. To address this limitation, Transformers were introduced to the field of computer vision, especially the ViT model [33][34]. Other variants, such as the Swing Transformer and Hybrid Transformer, combine local convolutions with self-attention mechanisms, further enhancing computational efficiency and accuracy, and demonstrating strong generalization ability, particularly on large-scale datasets [35][36]. Additionally, DETR applies Transformers to object detection tasks, and through an end-to-end training strategy, it successfully reduces the complexity of candidate box generation in traditional object detection methods [37][38][39].

The CrackDet-ViT combines RegNet, ViT, and Mask R-CNN to achieve local-to-global feature fusion, improving crack detection performance in complex environments. ViT helps capture directional and morphological features of cracks. Compared to traditional CNN models, CrackDet-ViT performs better in detecting small cracks and handling complex backgrounds.

3. Method

3.1. Overview of Our Model

The CrackDet-ViT model combines RegNet, ViT, and Mask R-CNN, aiming to improve the accuracy of bridge crack detection, classification, and pixel-level segmentation. The overall architecture of the model is shown in Figure 1. Through the organic integration of these three modules, CrackDet-ViT achieves a comprehensive processing flow from crack feature extraction to precise segmentation.

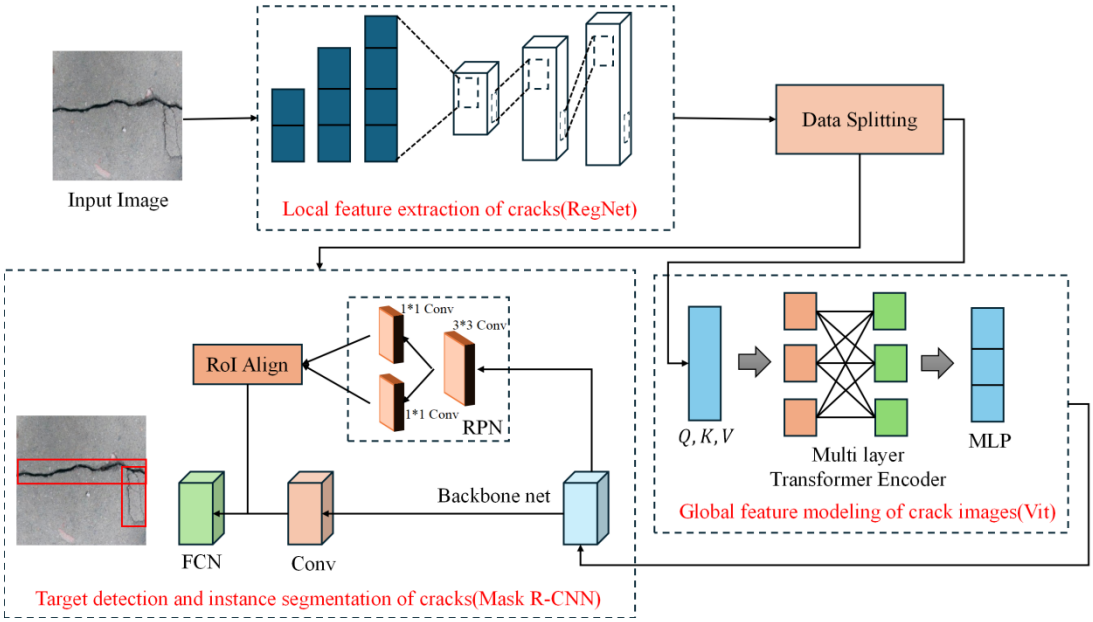

Figure 1. The overall architecture of the CrackDet-ViT model

RegNet, as the CNN backbone, is primarily responsible for extracting local features from the input image [40]. It learns the fundamental texture, edge, and shape information of cracks through efficient convolution operations and utilizes the Feature Pyramid Network (FPN) technique to extract features at multiple scales. This approach enhances the model's ability to detect cracks of varying sizes. Compared to traditional CNNs, RegNet's architecture is optimized for efficient local feature extraction and exhibits greater robustness in complex environments. The extracted local features are subsequently passed to the ViT module, which captures global information through a self-attention mechanism. This allows ViT to model crack direction and shape more effectively. Unlike traditional local convolution operations, ViT divides the image into patches and computes the relationships between them, improving the model's ability to capture the global shape of cracks, especially in complex backgrounds and long-range dependencies [41]. ViT helps improve the accuracy of crack detection, particularly in recognizing various crack shapes and locations. Mask R-CNN is employed for crack object detection and instance segmentation. It generates candidate regions for cracks using the RPN, performs RoI Align to ensure precise localization, and applies a FCN for pixel-level segmentation to produce accurate crack masks. This process accurately extracts crack shape and boundary information, enabling detection that is not limited to region localization but also includes

fine morphological details. CrackDet-ViT integrates the local features from RegNet, global features from ViT, and Mask R-CNN to achieve high-precision crack detection and segmentation. The collaborative function of multiple modules enhances the model's robustness in complex environments and ensures accurate crack shape extraction, providing a precise solution for automated bridge crack detection.

3.2. Extracting Local Features of Cracks

As one of the key components of the CrackDet-ViT model, RegNet is primarily responsible for extracting local crack features from the input image. This module learns fundamental textures, edges, and crack shapes through efficient convolution operations, and utilizes a Feature Pyramid Network (FPN) to extract multi-scale features. As illustrated in Figure 2, the RegNet architecture enhances computational efficiency through multiple convolution layers and depthwise separable convolutions, and strengthens its adaptability to cracks of different sizes by merging features at multiple levels.

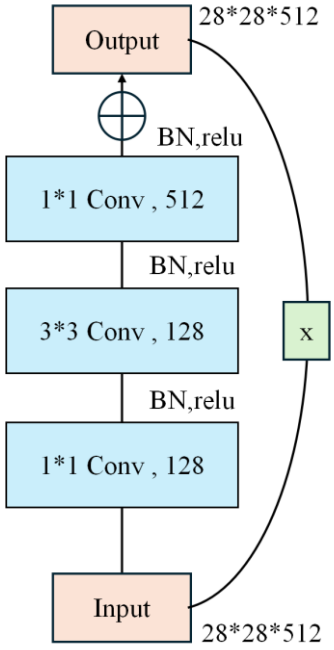

Figure 2. The structure of RegNet is used for local feature extraction.

After the input image undergoes a series of convolution operations. These features are processed through deep convolutions and pointwise convolutions, which reduce computational complexity while retaining key local information. Through this approach, RegNet can extract local features of cracks in the image while maintaining high computational efficiency. Assuming the input image size is I, the convolution kernel is K, and the bias term is b, the output feature map O after convolution is defined as in (1).

$$O = K * I + b \quad \text{[Formular 1]}$$

Furthermore, RegNet introduces the FPN technique during feature extraction to effectively fuse features at different scales, enhancing the model's ability to detect cracks at various sizes. Through

multi-level feature fusion, RegNet can better adapt to changes in crack sizes, particularly when the image contains cracks at multiple scales, allowing for more accurate localization of crack regions. Assuming the feature map at the l-th layer is F_l, with L representing the number of layers and α_l representing the weight coefficient of each feature map, the multi-scale feature map F_{multi} after FPN fusion as in (2).

$$F_{multi} = \sum_{l=1}^{L} \quad \alpha_l F_l \qquad \text{[Formular 2]}$$

To further improve computational efficiency, RegNet employs Depthwise Separable Convolution. Assuming the feature map of the input image is X, and the convolution kernels are K_d and K_p, the calculation process of depthwise separable convolution is defined as in (3).

$$X_d = K_d * X, \quad X_p = K_p * X_d \qquad \text{[Formular 3]}$$

This method significantly reduces the computational load and improves feature extraction efficiency, making it particularly suitable for large-scale datasets and high-resolution images.

RegNet's feature extraction module, while ensuring high computational efficiency, is able to capture detailed features of cracks, such as edges, textures, and shapes, providing high-quality local features for subsequent global feature modeling and crack segmentation. Finally, through residual connections, the stability of the gradients during the feature learning process is ensured, further enhancing the network performance. Assuming the input is X and the output is Y, the calculation formula for the residual connection is defined as in (4).

$$Y = F(X) + X \qquad \text{[Formular 4]}$$

Through this design, the RegNet module fully leverages the local feature extraction capabilities of convolutional neural networks, while combining depthwise separable convolution and multi-scale fusion techniques to enhance the overall performance of crack detection tasks.

3.3. Global Feature Modeling of Crack Images

In the CrackDet-ViT model, the ViT module is responsible for global feature modeling of crack images, aiming to enhance the model's ability to perceive the overall shape, direction, and other long-range dependencies of cracks. Figure 3 illustrates the architecture of the ViT module, which includes key steps such as Patch Embedding, Self-Attention, and Feature Fusion.

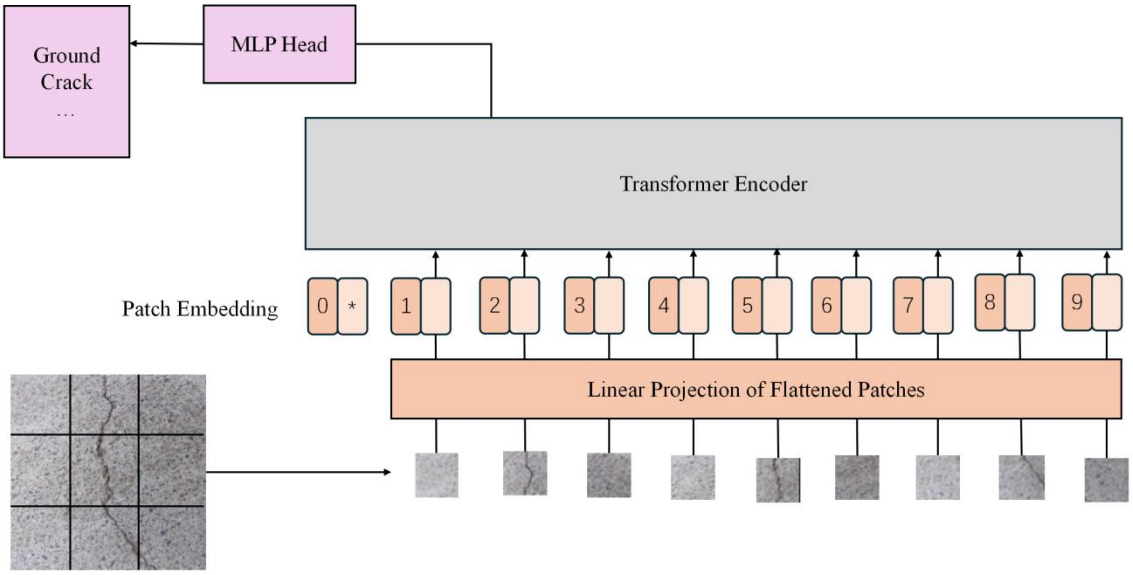

Figure 3. The structure of ViT is used for global feature modeling

The ViT model divides the input image into multiple fixed-size patches. Assuming the input image size is $H \times W \times C$, it is divided into P patches of size $P_h \times P_w$. Each patch is converted into a vector using Patch Embedding, where I_i represents the i-th patch in the image, W is the learned weight matrix, b is the bias term, and x_i is the embedded feature vector for the patch. This process maps the input image into a high-dimensional vector space, which facilitates further processing, as defined in (5).

$$x_i = Flatten(I_i) \cdot W + b \quad \text{[Formular 5]}$$

In the Self-Attention mechanism, ViT calculates the Query, Key, and Value for each input patch and uses this information to compute the relationship between every pair of patches. Q, K, and V denote the query, key, value matrices, respectively, and d_k is the dimension of the key matrix. The Self-Attention computation formula is defined as in (6).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad \text{[Formular 6]}$$

After several Self-Attention operations, ViT extracts global features from the image through multiple Transformer encoder layers, obtaining a global context representation for each patch. These representations not only contain local information but also fuse global dependencies. At this stage, the output of ViT is a vector of length $N \times d$, where N is the number of patches, and d is the embedding dimension. These global features are then passed into an MLP for further processing. In the MLP, features undergo a series of nonlinear transformations through fully connected layers, where W_1 and W_2 are weight matrices, b_1 and b_2 are bias terms, with ReLU as the activation function. The processing formula as in (7).

$$MLP(x) = ReLU(xW_1 + b_1)W_2 + b_2 \quad \text{[Formular 7]}$$

Finally, the output of ViT is passed to the Mask R-CNN module for object detection and pixel-level segmentation. This approach allows ViT to not only improve the accuracy of crack detection but also enhance the robustness of the model through global feature modeling, especially in complex

backgrounds or multi-scale crack scenarios, where it can better recognize the shape, direction, and location of cracks.

3.4. Target Detection and Instance Segmentation of Cracks

In the CrackDet-ViT model, the Mask R-CNN module handles both the crack object detection and instance segmentation tasks. Unlike traditional object detection methods, Mask R-CNN not only detects the location of cracks but also performs fine pixel-level segmentation, generating precise crack masks. Figure 4 presents the architecture of the Mask R-CNN module, clearly depicting the entire process from generating candidate regions, region alignment, to pixel-level segmentation.

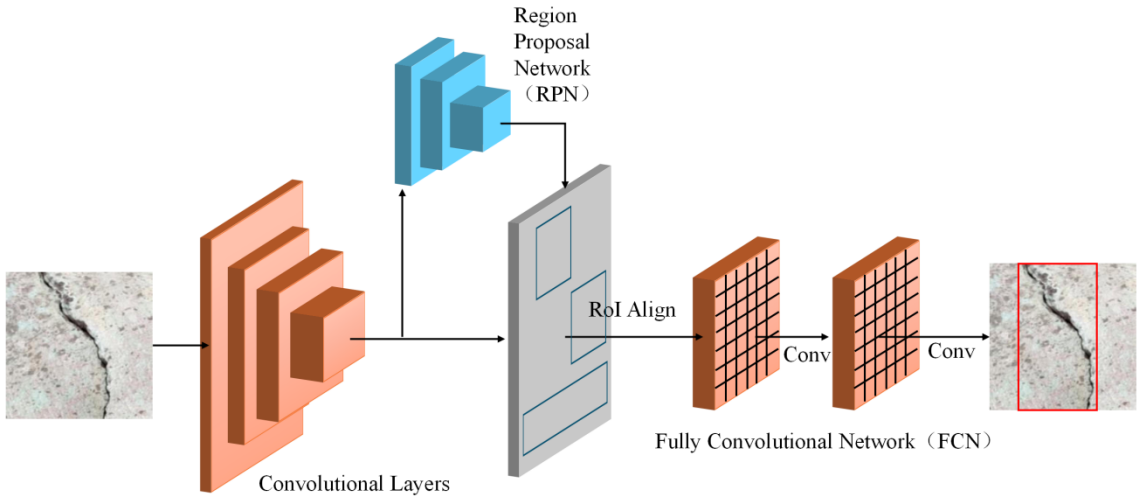

Figure 4. Structure of mask R-CNN for object detection and instance segmentation.

The RPN generates candidate regions for cracks through a sliding window. The core goal of the RPN is to quickly generate potential crack regions from the input image and provide a bounding box for each candidate region. Assuming the input feature map is F, the RPN uses convolution operations to generate a set of anchor boxes and calculates the IoU between each anchor box and the ground truth crack region using the following formula as shown in (8).

$$IoU = \frac{Area-of-Intersection}{Area-of-Union} \quad \text{[Formular 8]}$$

Based on this calculation, the RPN assigns a probability to each anchor box, indicating whether the anchor box contains the crack region, and generates crack candidate boxes. This process not only speeds up object localization but also effectively reduces computational load.

Next, the Region of Interest (RoI) Align is used for candidate region alignment. Traditional RoI Pooling methods result in boundary information loss due to the quantization process. However, RoI Align avoids quantization errors by accurately computing the pixel values in each candidate region, allowing for more precise feature extraction. Let the candidate region be R; the output feature map after RoI Align is calculated by the following formula as shown in (9).

$$F_{aligned} = RolAlign(F, R) \quad \text{[Formular 9]}$$

After candidate region alignment, the FCN is responsible for performing pixel-level segmentation on each candidate region and generating crack masks. The FCN processes the input feature map through a fully convolutional network, outputting a binary mask for each pixel's class prediction. For each candidate region R, the FCN output is a binary mask MR with shape $H \times W$, indicating whether each pixel within the region belongs to the crack area. The output process of the FCN can be expressed by the following formula as shown in (10).

$$M_R = \sigma\left(W * F_{aligned} + b\right) \quad \text{[Formular 10]}$$

Where W represents the convolution kernel, σ represents the activation function, and b represents the bias term. Finally, Mask R-CNN generates the complete morphological information for each crack region based on the output mask.

Mask R-CNN plays a crucial role in CrackDet-ViT by effectively performing crack object detection and instance segmentation, helping the model detect crack locations more accurately in complex backgrounds while providing fine pixel-level segmentation.

4. Experiment

4.1 Datasets

SDNET2018 and Kaggle Crack Detection Challenge were used to test and evaluate the performance of the CrackDet-ViT model. These datasets were chosen because they provide a diverse range of crack images, covering various crack types, lighting conditions, and background complexities, making them suitable for crack detection, classification, and pixel-level segmentation tasks. The evaluation using these datasets enables a comprehensive verification of the CrackDet-ViT model's robustness and accuracy in various complex environments.

Table 1. Overview of Datasets Used in the Experiment.

Dataset Name	Number of Images	Image Type	Annotation Type	Dataset Characteristics
SDNET2018	12,000 images	Bridge surface crack images	Crack location, category, segmentation mask	Contains cracks under different lighting and complex backgrounds
Kaggle - Crack Detection Challenge	2,000 images	Road and bridge crack images	Crack location, classification labels, pixel-level masks	Contains cracks under complex backgrounds and various lighting conditions

The SDNET2018 dataset is provided by NASA and contains crack images from different bridge surfaces [42]. The dataset offers crack images under varying lighting and complex background

conditions, making it an excellent resource for testing CrackDet-ViT on diverse crack shapes and complex environments.

The Kaggle - Crack Detection Challenge is a publicly available dataset that includes crack images from roads and bridges under different backgrounds and lighting conditions [43]. The dataset provides bounding boxes and pixel-level labels for cracks, making it suitable for object detection and pixel-level segmentation tasks.

4.2 Experimental Setup and Configuration

Table 2. Hardware and Software Configuration for Experiments

Component	Specification
GPU	NVIDIA A100 (40GB VRAM)
CPU	AMD EPYC 7452 (32 cores)
Memory	256GB DDR4
Storage	4TB SSD
Operating System	Ubuntu 20.04 LTS
Deep Learning Frameworks	PyTorch 1.10, Detectron2 (Mask R-CNN)
CUDA	11.2
cuDNN	8.1
Python Version	3.8

The loss functions used during training included localization loss, classification loss, and mask loss, ensuring effective crack detection and segmentation. For dataset partitioning, 70% of the SDNET2018 dataset was used for training and 30% for testing, while 80% of the Kaggle Crack Detection Challenge dataset was used for training and 20% for testing. The hardware environment is summarized in Table 2.

4.3 Evaluation Metric

In this study, several evaluation metrics were used to comprehensively assess the performance of the CrackDet-ViT model in crack detection, classification, and pixel-level segmentation tasks, enabling accurate measurement of the model's detection accuracy, segmentation ability, and robustness [44].

The mean Average Precision (mAP) calculates the average value of AP (Average Precision) at different thresholds. AP reflects the balance between precision and recall at different IoU thresholds, where AP_i represents the average precision for the i-th category, and N is the number of categories. mAP provides a comprehensive performance evaluation of the model in multi-class crack detection tasks is defined as in (11).

$$mAP = \frac{1}{N}\sum_{i=1}^{N} \quad AP_i \qquad \text{[Formular 11]}$$

IoU is a key metric for evaluating object detection and image segmentation performance. It represents the ratio of the intersection to the union of the predicted and ground truth regions. A higher IoU indicates closer alignment between predictions and actual annotations. Let A and B denote the predicted and ground truth regions, respectively. The IoU is defined as in (12).

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} = \frac{|A \cap B|}{|A \cup B|} \quad \text{[Formular 12]}$$

Precision measures the proportion of actual crack pixels among the pixels predicted as crack regions by the model. The higher the Precision, the fewer false positives the model generates in its predictions. TP represents the number of pixels correctly predicted as cracks, and FP represents the number of pixels incorrectly predicted as cracks. Higher Precision means the model has higher accuracy in locating crack regions as in (13).

$$Precision = \frac{TP}{TP+FP} \quad \text{[Formular 13]}$$

Recall measures the proportion of actual crack regions correctly identified by the model. A higher Recall means the model can identify more true crack regions, but it may lead to more false positives. FN represents the number of crack regions missed by the model. Higher Recall means the model can capture as many crack regions as possible as in (14).

$$Recall = \frac{TP}{TP+FN} \quad \text{[Formular 14]}$$

Dice Coefficient is a metric for evaluating image segmentation accuracy, particularly suitable for pixel-level segmentation tasks. A and B represent the predicted and ground truth masks, respectively. The Dice Coefficient effectively measures the overlap between the model's predicted and actual crack regions, making it an important metric for assessing crack segmentation performance as in (15).

$$DiceCoefficient = \frac{2 \times |A \cap B|}{|A| + |B|} \quad \text{[Formular 15]}$$

4.4 Comparative Experimental Results and Analysis

In this section, we compare the performance of the CrackDet-ViT model with other mainstream models on two datasets, focusing on the comparison of five evaluation metrics: mAP, IoU, Precision, Recall, and Dice Coefficient. The comparison models include: YOLOv8, Faster R-CNN, DeepLabV3+, Mask R-CNN, and Swin Transformer. These models represent the cutting-edge technologies in object detection and segmentation, and most were proposed within the past three years, making them suitable benchmarks for validating the advantages of CrackDet-ViT. Table 3 presents the experimental results.

Table 3. Comparison of model performance on two datasets.

Model	Dataset	mAP (%)	IoU (%)	Precision (%)	Recall (%)	Dice Coefficient (%)
CrackDet-ViT	SDNET2018	87.5	79.3	89.2	85.1	90.3
	Kaggle Challenge	84.7	76.2	87.9	82.5	88.4
Swin Transformer [45]	SDNET2018	86.2	78.1	88.5	83.2	89.8
	Kaggle Challenge	83.5	75.0	86.2	80.7	86.8
YOLOv8[46]	SDNET2018	81.5	73.4	84.9	78.2	85.4
	Kaggle	79.8	70.5	83.5	76.9	82.7

	Challenge					
Faster R-CNN [27]	SDNET2018	83.4	74.2	86.3	80.6	87.5
	Kaggle Challenge	80.9	72.4	84.8	78.7	85.6
DeepLabV3+ [47]	SDNET2018	82.8	75.3	85.5	79.0	86.2
	Kaggle Challenge	80.1	70.8	83.9	76.5	84.3
Mask R-CNN [48]	SDNET2018	84.9	76.7	87.3	81.4	88.6
	Kaggle Challenge	81.5	73.1	85.0	77.9	85.2

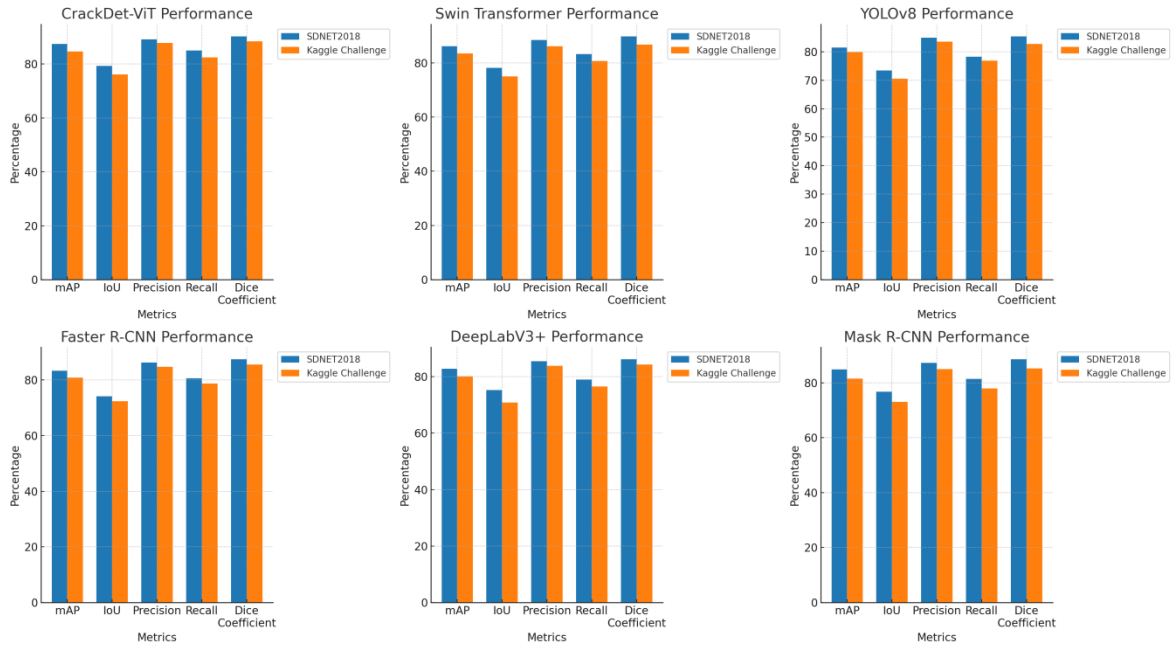

Figure 5. The overall architecture of the CrackDet-ViT model

As shown in Figure 5, the performance of each model on the two datasets is presented. On the SDNET2018 dataset, CrackDet-ViT achieves an mAP of 87.5%, which is an improvement of approximately 6% over YOLOv8 and about 4% over Faster R-CNN. This significant improvement indicates that CrackDet-ViT, while considering both detection accuracy and recall, offers higher average precision, making crack detection more accurate in complex backgrounds. Additionally, compared to DeepLabV3+ and Mask R-CNN, CrackDet-ViT also demonstrates a higher mAP, indicating better performance balance and robustness in crack detection tasks. On the Kaggle - Crack Detection Challenge dataset, CrackDet-ViT achieves an mAP of 84.7%, which is an improvement of about 5% over YOLOv8 and 1.2% over Swin Transformer. This result indicates that, even with a greater variety of crack types and more complex backgrounds, CrackDet-ViT can maintain high detection accuracy, fully demonstrating its advantages over existing state-of-the-art object detection models. For the IoU (Intersection over Union) metric, CrackDet-ViT achieves an IoU of 79.3% on

the SDNET2018 dataset, improving by approximately 5% over YOLOv8 and 4% over DeepLabV3+. This improvement highlights CrackDet-ViT's advantage in crack segmentation tasks, as it can more accurately localize crack regions, thereby enhancing overall segmentation performance. Similarly, on the Kaggle Challenge dataset, CrackDet-ViT shows a 5% improvement in IoU compared to YOLOv8 and Mask R-CNN, demonstrating its accuracy and robustness in crack segmentation tasks. In terms of Precision and Recall, CrackDet-ViT also shows significant improvements. Compared to YOLOv8, CrackDet-ViT's Precision on the SDNET2018 dataset improves by about 4.3%, while Recall improves by 8.7%, indicating that CrackDet-ViT is more accurate in detecting and recognizing cracks while capturing a great number of true crack regions. Compared to Faster R-CNN and Mask R-CNN, CrackDet-ViT shows improvements in both precision and recall, further demonstrating its advantages in model balance and robustness. Regarding the Dice Coefficient, CrackDet-ViT demonstrates the most substantial improvement, particularly in crack segmentation accuracy. Compared to YOLOv8 and Faster R-CNN, CrackDet-ViT improves by over 5% on the SDNET2018 dataset, indicating that it significantly enhances pixel-level segmentation accuracy and captures more precise crack contours. This provides CrackDet-ViT a clear advantage in complex backgrounds and multi-scale crack segmentation tasks.

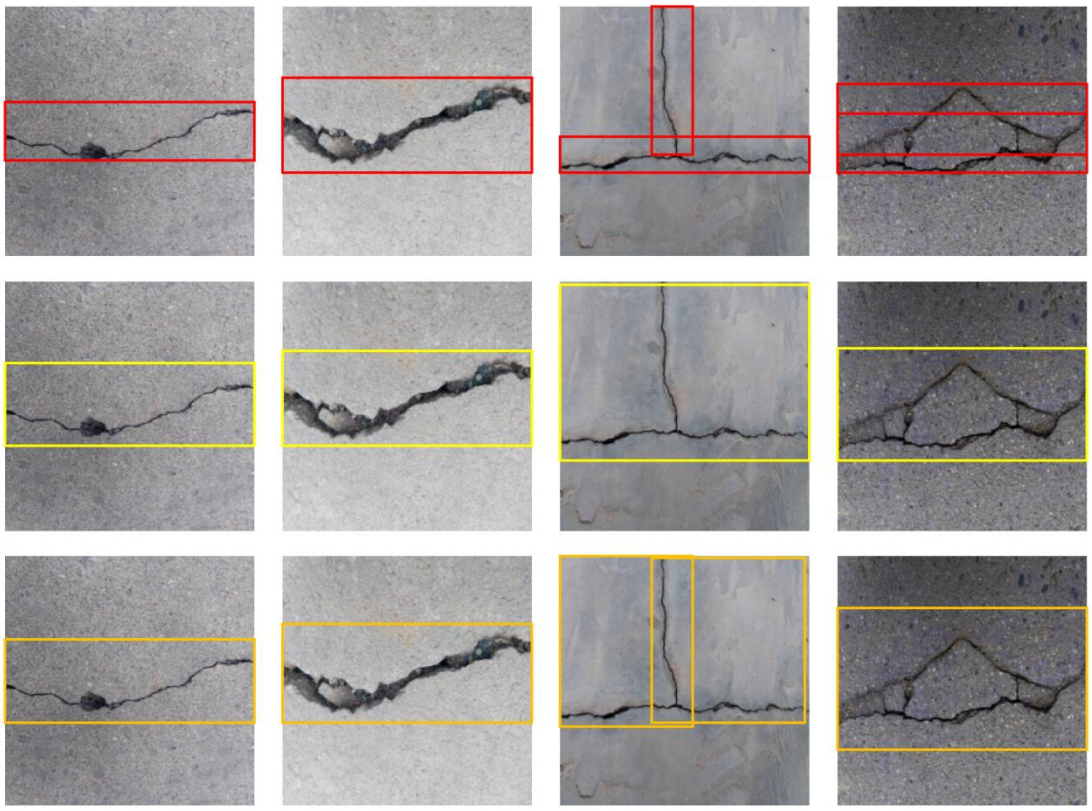

Figure 6. Comparison of accuracy trends with baseline for multiple models

As shown in Figure 6, the MS-RL DeepSeek model surpasses all other comparison models across multiple evaluation metrics, particularly excelling in capturing long-term trends in economic cycles and predicting short-term fluctuations in financial markets. This indicates that MS-RL

DeepSeek can provide more accurate predictions and risk assessments in complex, dynamically changing economic environments. These results confirm the effectiveness of multi modal data fusion and decision optimization based on reinforcement learning and demonstrate its potential in applications related to economic and financial risk assessment.

4.5 Ablation Experimental Results and Analysis

In this section, we present the ablation study of CrackDet-ViT, where different modules of the model are systematically removed to assess the importance and effectiveness of each component [49]. These experiments were conducted on the SDNET2018 and Kaggle Crack Detection Challenge datasets, with a focus on the impact of removing individual modules: RegNet, ViT, and Mask R-CNN. The results shown in table 4 highlight the importance of each module and demonstrate the contribution of the complete model to overall performance.

Table 4. Ablation study results on SDNET2018 and kaggle - crack detection challenge dataset

Model Variant	Dataset	mAP (%)	IoU (%)	Precision (%)	Recall (%)	Dice Coefficient (%)
CrackDet-ViT	SDNET2018	87.5	79.3	89.2	85.1	90.3
	Kaggle	84.7	76.2	87.9	82.5	88.4
w/o RegNet	SDNET2018	84.1	75.2	86.5	80.8	86.4
	Kaggle	81.3	72.0	85.2	77.6	84.5
w/o ViT	SDNET2018	83.6	73.8	84.2	79.5	85.2
	Kaggle	80.8	70.4	83.8	75.9	83.2
w/o Mask R-CNN	SDNET2018	82.4	72.4	83.0	78.1	84.6
	Kaggle	79.5	69.1	82.7	74.8	82.6

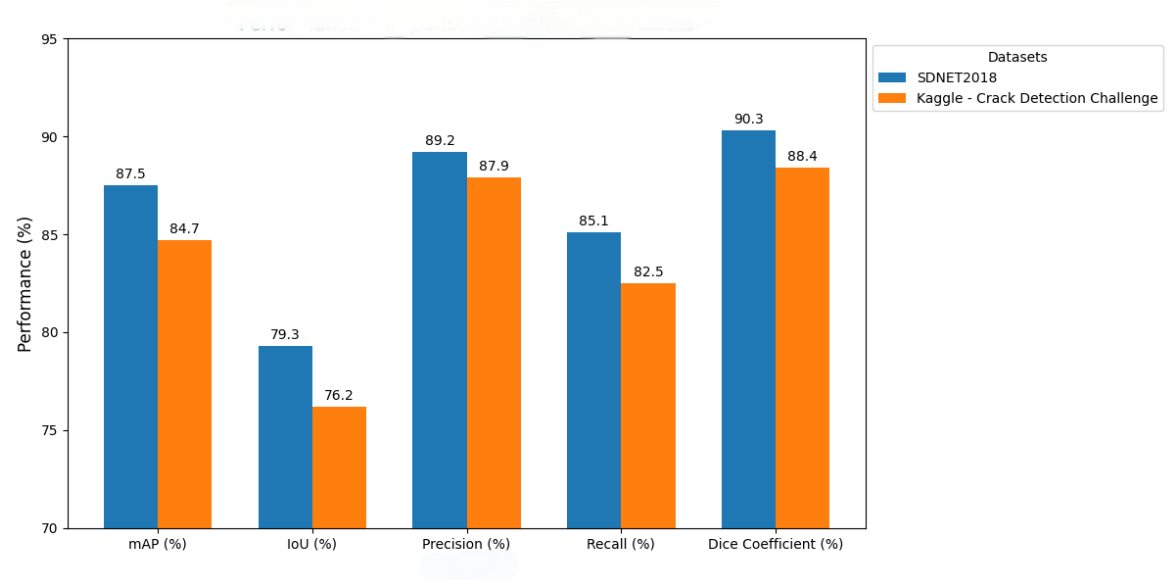

Figure 7. The overall architecture of the CrackDet-ViT model

As shown in Figure 7, each module in the CrackDet-ViT model significantly contributes to its overall performance. Removing RegNet leads to a noticeable decline in all performance metrics. On the SDNET2018 and Kaggle Challenge datasets, the mAP drops by approximately 3.4% on both datasets, indicating that local feature extraction is crucial for detecting detailed crack characteristics. Both IoU and Dice Coefficient also decrease, highlighting the importance of accurate local feature representation in segmentation tasks. Similarly, when ViT is removed, performance declines. The mAP decreased by 3.9% on both the SDNET2018 and Kaggle Challenge datasets. This underscores the importance of global feature modeling in capturing the overall context of cracks, which helps the model detect and segment cracks in more complex scenarios. Removing Mask R-CNN, which is responsible for object detection and instance segmentation, also results in a performance drop, particularly in segmentation accuracy. The mAP decreases by approximately 5.1% on the SDNET2018 dataset and 5.2% on the Kaggle Challenge dataset, and both Dice Coefficient and IoU decrease significantly. This indicates that the ability to generate precise crack masks through instance segmentation is crucial for achieving high-quality segmentation results. When both RegNet and ViT are removed, the model's performance further declines, with mAP dropping by more than 6% on both datasets. This confirms that local feature extraction and global feature modeling play critical roles in the model's effectiveness. Finally, when both RegNet and Mask R-CNN are removed, performance declines significantly, with mAP dropping by more than 8% on SDNET2018 dataset and 8.4% on the Kaggle Challenge dataset. Without these two key modules, the model's ability to accurately detect cracks and perform fine-grained segmentation is severely compromised.

By removing any two modules from RegNet, ViT, and Mask R-CNN, Tables 5 illustrates the extent of the contribution of these modules to the overall model performance. These results provide a deeper understanding of the impact of each module on the model's performance [50].

Table 5. Ablation study results on SDNET2018 and kaggle - crack detection challenge dataset (Removing Two or More Modules).

Model Variant	Dataset	mAP (%)	IoU (%)	Precision (%)	Recall (%)	Dice Coefficient (%)
CrackDet-ViT	SDNET2018	87.5	79.3	89.2	85.1	90.3
	Kaggle	84.7	76.2	87.9	82.5	88.4
w/o RegNet & ViT	SDNET2018	80.9	70.1	80.4	75.3	82.9
	Kaggle	77.9	67.6	80.9	72.3	80.3
w/o RegNet & Mask R-CNN	SDNET2018	79.3	68.6	78.7	73.8	81.6
	Kaggle	76.3	65.9	78.3	70.4	78.9
w/o ViT & Mask R-CNN	SDNET2018	78.4	67.2	77.1	72.1	80.1
	Kaggle	75.6	64.2	77.2	69.2	77.4

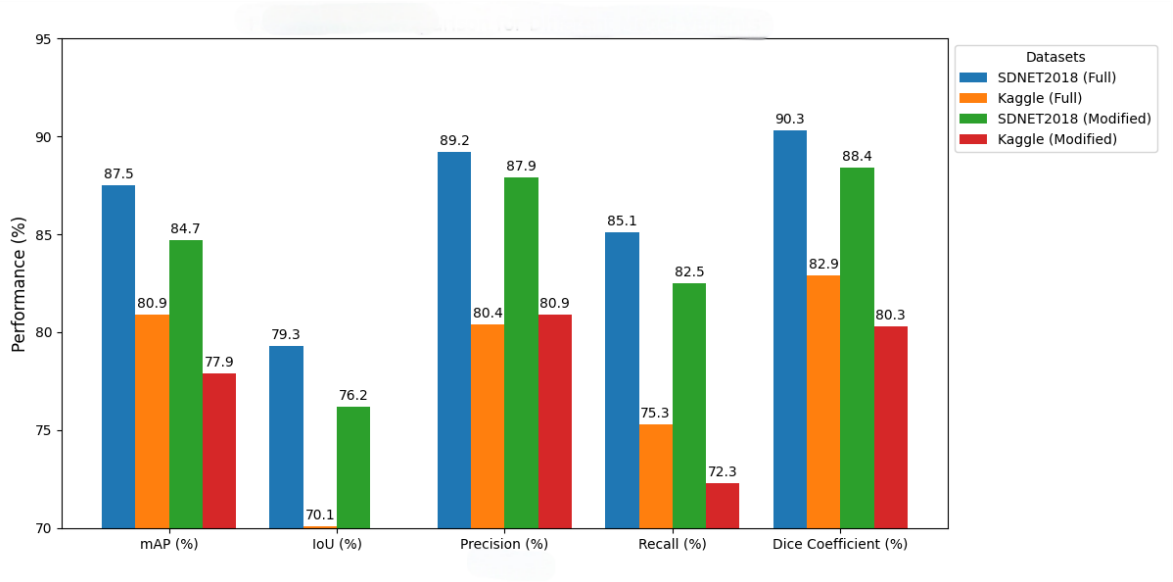

Figure 8. The overall architecture of the CrackDet-ViT model.

As shown in Figure 8 after removing two modules, the overall performance of CrackDet-ViT significantly declines. On the SDNET2018 dataset, after removing the RegNet and ViT modules, the mAP drops from 87.5% to 80.9%, and both IoU and Dice Coefficient show significant reductions. This indicates that RegNet and ViT are indispensable modules in the model, with the former responsible. Without these two modules, the model's accuracy in detecting cracks in complex backgrounds and across multiple scales is greatly reduced. Similarly, after removing the RegNet and Mask R-CNN modules, the mAP drops to 79.3%, and performance further weakens, validating the importance of Mask R-CNN in instance segmentation and fine crack detection. On the Kaggle - Crack Detection Challenge dataset, after removing RegNet and ViT, the mAP decreases to 77.9%, and the Dice Coefficient drops to 80.3%. This performance decline further demonstrates the crucial role of RegNet and ViT in the model's performance, especially in extracting crack details and modeling global information. Additionally, after removing ViT and Mask R-CNN, the mAP decreases to 75.6%, which further highlights the importance of Mask R-CNN in crack instance segmentation and detection. Without this module, the model cannot accurately segment crack regions.

Overall, removing any two modules significantly reduces the performance of CrackDet-ViT, emphasizing the importance of each module in the overall model. The combination of RegNet and ViT enables the model to accurately extract local crack features while capturing the global shape of cracks, and Mask R-CNN ensures precise pixel-level segmentation. Therefore, the synergistic integration of all three modules provides CrackDet-ViT with excellent crack detection and segmentation performance.

5. Conclusion and Discussion

This paper introduces CrackDet-ViT, an innovative model for bridge crack detection and classification that combines RegNet, ViT, and Mask R-CNN to achieve high-precision detection and pixel-level segmentation. RegNet extracts local features, ViT models global information, and Mask

R-CNN performs instance segmentation, addressing key challenges in crack detection.

CrackDet-ViT demonstrates excellent robustness across various crack types, complex backgrounds, and diverse lighting conditions. The model excels in detecting and segmenting cracks of different sizes, particularly in noisy and challenging environments. ViT improves generalization on unseen data, making the model more adaptable. By integrating advanced deep learning techniques, CrackDet-ViT delivers high accuracy and robustness, providing an effective solution for automated crack detection in complex environments. This makes CrackDet-ViT a powerful tool for infrastructure health monitoring with significant application potential.

Looking ahead, further improvements in the CrackDet-ViT are possible. Future work may involve integrating multimodal data, such as combining RGB and infrared images, to enhance detection accuracy under varying environmental conditions. Additionally, incorporating self-supervised learning could reduce reliance on labeled data. Optimizing the model for real-time crack detection, particularly for deployment edge devices, would further expand its applicability to large-scale infrastructure monitoring.

Acknowledgements

This article received no financial or funding support.

Conflicts of Interest

The author confirms that there are no conflicts of interest.

References

[1] Golding, V.P., Gharineiat, Z. and Munawar, H.S. Crack detection in concrete structures using deep learning. Sustainability, 2022, 14(13), 8117.

[2] Ali, R., Chuah, J.H. and Talip, M.S.A. Structural crack detection using deep convolutional neural networks. Automation in Construction, 2022, 133, 103989.

[3] Guo, F., Liu, J. and Lv, C. A novel transformer-based network with attention mechanism for automatic pavement crack detection. Construction and Building Materials, 2023, 391, 131852.

[4] Hamishebahar, Y., Guan, H. and So, S. A comprehensive review of deep learning-based crack detection approaches. Applied Sciences, 2022, 12(3), 1374.

[5] Umer, M.J. and Sharif, M.I. A comprehensive survey on quantum machine learning and possible applications. International Journal of E-Health and Medical Communications, 2022, 13(5), 1–17.

[6] Qiu, Q. and Lau, D. Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images. Automation in Construction, 2023, 147, 104745.

[7] Nguyen, S.D., Tran, T.S. and Tran, V.P. Deep learning-based crack detection: A survey. International Journal of Pavement Research and Technology, 2023, 16(4), 943–967.

[8] Xing, X., Wang, B. and Ning, X. Short-term OD flow prediction for urban rail transit control: A multi-graph spatiotemporal fusion approach. Information Fusion, 2025, 102950.

[9] Li, M., Yuan, J. and Ren, Q. CNN-transformer hybrid network for concrete dam crack patrol inspection. Automation in Construction, 2024, 163, 105440.

[10] Jin, J. and Zhang, Y. Innovation in financial enterprise risk prediction model: A hybrid deep learning technique based on CNN-transformer-WT. Journal of Organizational and End User Computing, 2024, 36(1), 1–26.

[11] Zhang, H., Zhang, R. and Sun, D. Analyzing the pore structure of pervious concrete based on the deep learning

framework of Mask R-CNN. Construction and Building Materials, 2022, 318, 125987.

[12] Shiau, W.L., Liu, C. and Cheng, X. Employees' behavioral intention to adopt facial recognition payment to service customers: From SQB and value-based adoption perspectives. Journal of Organizational and End User Computing, 2024, 36(1), 1–32.

[13] Ren, B. and Wang, Z. Strategic focus, tasks, and pathways for promoting China's modernization through new productive forces. Journal of Xi'an University of Finance and Economics, 2024, 1, 3–11.

[14] Jing, C. and Qing, W. The logic and pathways of new productive forces driving high-quality development. Journal of Xi'an University of Finance and Economics, 2024, 37(1), 12–20.

[15] Zhang, Y. and Yuen, K.V. Review of artificial intelligence-based bridge damage detection. Advances in Mechanical Engineering, 2022, 14(9), 16878132221122770.

[16] Inam, H., Islam, N.U. and Akram, M.U. Smart and automated infrastructure management: A deep learning approach for crack detection in bridge images. Sustainability, 2023, 15(3), 1866.

[17] Pozo, F., Tibaduiza, D.A. and Vidal, Y. Sensors for structural health monitoring and condition monitoring. 2021.

[18] Zhang, Q., Barri, K. and Babanajad, S.K. Real-time detection of cracks on concrete bridge decks using deep learning in the frequency domain. Engineering, 2021, 7(12), 1786–1796.

[19] Xia, M., Phillips, F. and Zhang, W. From carbon capture to cash: Strategic environmental leadership, AI, and the performance of US firms. Journal of Organizational and End User Computing, 2024, 36(1), 1–24.

[20] Wu, H.T., Li, J.X. and Chen, M.Y. Building a sustainable development education system for large organizations based on artificial intelligence of things. Journal of Organizational and End User Computing, 2024, 36(1), 1–19.

[21] Yu, Y., Rashidi, M. and Samali, B. Crack detection of concrete structures using deep convolutional neural networks optimized by enhanced chicken swarm algorithm. Structural Health Monitoring, 2022, 21(5), 2244–2263.

[22] Sannidhan, M., Martis, J.E. and Nayak, R.S. Detection of antibiotic constituent in *Aspergillus flavus* using quantum convolutional neural network. International Journal of E-Health and Medical Communications, 2023, 14(1), 1–26.

[23] Tran, T.S., Nguyen, S.D. and Lee, H.J. Advanced crack detection and segmentation on bridge decks using deep learning. Construction and Building Materials, 2023, 400, 132839.

[24] Kodipalli, A., Fernandes, S.L. and Dasar, S.K. Computational framework of inverted fuzzy C-means and quantum convolutional neural network towards accurate detection of ovarian tumors. International Journal of E-Health and Medical Communications, 2023, 14(1), 1–16.

[25] He, F., Li, H. and Ning, X. BeautyDiffusion: Generative latent decomposition for makeup transfer via diffusion models. Information Fusion, 2025, 103241.

[26] Silva, L.A., Leithardt, V.R.Q. and Batista, V.F.L. Automated road damage detection using UAV images and deep learning techniques. IEEE Access, 2023, 11, 62918–62931.

[27] Hacıefendioğlu, K. and Başaga, H.B. Concrete road crack detection using deep learning-based faster R-CNN method. Iranian Journal of Science and Technology, Transactions of Civil Engineering, 2022, 46(2), 1621–1633.

[28] Meng, X. Concrete crack detection algorithm based on deep residual neural networks. Scientific Programming, 2021, 2021(1), 3137083.

[29] Fang, F., Li, L. and Gu, Y. A novel hybrid approach for crack detection. Pattern Recognition, 2020, 107, 107474.

[30] Zhang, H., Yu, L. and Wang, G. Cross-modal knowledge transfer for 3D point clouds via graph offset prediction. Pattern Recognition, 2025, 162, 111351.

[31] Li, Q., Chen, H. and Huang, X. Oral multi-pathology segmentation with lead-assisting backbone attention network and synthetic data generation. Information Fusion, 2025, 118, 102892.

[32] He, M., Chen, D. and Liao, J. Deep exemplar-based colorization. ACM Transactions on Graphics, 2018, 37(4), 1–16.

[33] Khan, S., Naseer, M. and Hayat, M. Transformers in vision: A survey. ACM Computing Surveys, 2022, 54(10s), 1–41.

[34] Zhang, X., Li, Y. and Fortes, D.J. Evaluate multi-objective optimization model for product supply chain inventory control based on grey wolf algorithm. Journal of Organizational and End User Computing, 2024, 36(1), 1–24.

[35] Liu, Z., Lin, Y. and Cao, Y. Swin transformer: Hierarchical vision transformer using shifted windows. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 10012–10022.

[36] Thara, D., Premasudha, B. and Murthy, T. EEG forecasting with univariate and multivariate time series using windowing and baseline method. International Journal of E-Health and Medical Communications, 2022, 13(5), 1–13.

[37] Carion, N., Massa, F. and Synnaeve, G. End-to-end object detection with transformers. European Conference on Computer Vision, 2020, 213–229.

[38] Gao, Y., Zhou, M. and Metaxas, D.N. UTNet: A hybrid transformer architecture for medical image segmentation. Medical Image Computing and Computer Assisted Intervention, 2021, 61–71.

[39] Zuo, S., Xiao, Y. and Chang, X. Vision transformers for dense prediction: A survey. Knowledge-Based Systems, 2022, 253, 109552.

[40] Tan, M. and Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. International Conference on Machine Learning, 2019, 6105–6114.

[41] Han, K., Wang, Y. and Chen, H. A survey on vision transformer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(1), 87–110.

[42] Mohammed Abdelkader, E. On the hybridization of pre-trained deep learning and differential evolution algorithms for semantic crack detection and recognition in ensemble of infrastructures. Smart and Sustainable Built Environment, 2022, 11(3), 740–764.

[43] Islam, M.M., Hossain, M.B. and Akhtar, M.N. CNN based on transfer learning models using data augmentation and transformation for detection of concrete crack. Algorithms, 2022, 15(8), 287.

[44] Ali, L., Alnajjar, F. and Jassmi, H.A. Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. Sensors, 2021, 21(5), 1688.

[45] Hou, Q., Lu, C.Z. and Cheng, M.M. Conv2former: A simple transformer-style convnet for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[46] Safaldin, M., Zaghden, N. and Mejdoub, M. An improved YOLOv8 to detect moving objects. IEEE Access, 2024.

[47] Peng, H., Xue, C. and Shao, Y. Semantic segmentation of litchi branches using DeepLabV3+ model. IEEE Access, 2020, 8, 164546–164555.

[48] Yang, F., Huo, J. and Cheng, Z. An improved Mask R-CNN micro-crack detection model for the surface of metal structural parts. Sensors, 2023, 24(1), 62.

[49] Abate, A.F., Cimmino, L. and Lorenzo-Navarro, J. An ablation study on part-based face analysis using a multi-input convolutional neural network and semantic segmentation. Pattern Recognition Letters, 2023, 173, 45–49.

[50] Liu, J.J., Hou, Q. and Liu, Z.A. PoolNet+: Exploring the potential of pooling for salient object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 45(1), 887–904.