INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 20, Issue: 2, Month: April, Year: 2025
Article Number: 6872, https://doi.org/10.15837/ijccc.2025.2.6872

CCC Publications
AGORA UNIVERSITY PRESS

Optimized VGG Network with Dilated Residual Convolution and Path Enhancement for Crack Image Segmentation

WANG Xiaofang, WU JiaLing, CHEN Xin, HOU JunNiang,CHEN Peichun

WANG Xiaofang*
Geely University of China, China
123 Chengjian Avenue Section 2, Eastern New Area, Chengdu, Sichuan Province, China
939549393@qq.com *Corresponding author:939549393@qq.com

WU JiaLing
Chengdu College of Electronic Science and Technology of China
No. 1, Baiye Road, High-tech West Zone, Chengdu, Sichuan Province, China

CHEN Xin
Chengdu College of Electronic Science and Technology of China
No. 1, Baiye Road, High-tech West Zone, Chengdu, Sichuan Province, China

HOU JunNiang
Geely University of China, China
123 Chengjian Avenue Section 2, Eastern New Area, Chengdu, Sichuan Province, China

CHEN Peichun
Research Servicesr, Coventry University, Coventry, UK

Abstract

Image crack segmentation is a critical task in infrastructure maintenance, as accurate crack detection is essential for structural health monitoring and preventing potential risks. Although traditional Convolutional Neural Networks (CNN) have achieved certain successes in image crack detection, they still exhibit limitations in handling noisy images and detecting fine cracks. This study proposes a VGG network based on dilated residual convolution and path-enhanced optimization to improve the accuracy and efficiency of image crack segmentation. The study utilizes fuzzy morphological filtering for preprocessing, introduces dilated convolution to expand the receptive field, employs residual structures to enhance feature transmission, and incorporates path enhancement modules to boost network performance. The experimental evaluation was conducted using the SDNET2018, METU Dataset, and CFD datasets. The results show that the proposed method achieves a mean intersection over union (MIoU) of 94.60% and a mean accuracy of 96.18%. The improved algorithm demonstrates significant advantages in noise interference handling and detail processing.

Keywords: Crack Segmentation; Dilated Residual Convolution; Path Enhancement Module; Neural Network

1 Introduction

The aging infrastructure and increasing traffic loads have made crack detection a critical aspect of ensuring the safety of buildings and roads. Accurate image crack segmentation aids in the timely identification and repair of structural defects, preventing potential safety hazards [1]. In recent years, Convolutional Neural Networks (CNN) have been widely adopted in the field of image crack detection due to their powerful feature extraction and representation capabilities [2] [3]. Crack segmentation techniques based on CNN [4] and VGG [5] architectures have become a research hotspot.

For instance, Wu et al. [6] combined the Retinex image enhancement algorithm with the VGG_19 model to study efficient crack image segmentation, focusing on crack length and width information. However, improvements in detection accuracy are still needed. Hao et al. [7] proposed a multi-task enhancement method based on Faster-RCNN, aimed at detecting fine cracks, but the model's detection performance requires further enhancement. Vishal et al. [8] introduced an automated pavement distress analysis system based on the YOLOv2 deep learning architecture, which offers high operational speed but performs poorly in terms of crack detection quantity and clarity. Sukhad et al. [9] developed an algorithm based on SqueezeNet and texture encoding layers, which shows good robustness and effectiveness but lacks real-time detection performance. Madiha et al. [10] presented a transfer learning method using the VGG19 network, which improves model generalization but consumes substantial computational resources.

In 2015, a fully convolutional network architecture (UNet) was proposed for image segmentation [11]. This architecture combines high-resolution features with low-resolution contextual information, using a symmetrical structure and skip connections, making it highly effective in crack segmentation tasks. Building on this UNet architecture, Vladimir et al. [12] proposed the DAUNet model, which offers advantages in terms of lightweight design and training time but has poor noise handling performance. Fan et al. [13] proposed a UNet-based convolutional neural network for automated crack segmentation, which efficiently handles irregular and noisy images, achieving automated segmentation. However, its performance in detail segmentation is suboptimal. Zhou et al. [14] introduced the UNet++ architecture, which enhances crack segmentation performance through dense skip connections and a deep supervision mechanism.

Furthermore, researchers have applied attention mechanisms to image crack segmentation. Luo et al. [15] proposed the STrans-YOLOX model, which combines CNN and Swin Transformer, incorporating a global attention guidance module. Although it enhances multi-scale feature extraction capabilities, further improvements in crack detection accuracy and denoising are needed. Zhang et al. [16] introduced the APLCNet algorithm, which effectively improves crack detection accuracy through channel attention and spatial attention mechanisms, but its performance in handling crack contours is inadequate. Saberironaghi et al. [17] developed a crack segmentation model combining a multi-scale attention mechanism with an efficient UNet, which excels in handling complex backgrounds and fine cracks, though it demands significant computational resources and has a complex training process. Katsamenis et al. [18] proposed the R2AU-Net model, which integrates attention mechanisms and few-shot learning, allowing dynamic fine-tuning to optimize crack segmentation performance based on user feedback. However, the model's high computational resource requirements and training complexity remain challenges.

In addition, Fan et al.'s [19] FPHBN network can effectively extract image features but has poor detection efficiency. Xie et al. [20] proposed a fully convolutional neural network and deep supervision network (HED) for edge detection from image to image, optimizing computational speed but performing poorly in detail handling.

To address the shortcomings of existing methods in noise handling and fine crack detection, this study proposes a VGG network based on dilated residual convolutions and path enhancement optimization, aiming to improve the accuracy and efficiency of image crack segmentation. The study employs fuzzy morphological filtering for preprocessing to reduce noise impact on crack detection. In the network structure, dilated convolution modules are used to expand the receptive field, allowing the network to capture richer contextual information. Simultaneously, residual structures enhance the transmission and fusion of features at different levels. The path enhancement module further optimizes feature paths, enhancing the model's segmentation performance. Finally, deconvolution and

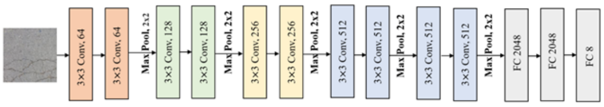


Figure 1: Schematic Diagram of the Network Structure

upsampling modules restore the image size, achieving crack segmentation.

2 Realted

2.1 VGG Network

The VGG model is one of the most commonly used convolutional neural network models, proposed by Simonyan and Zisserman. This model processes images by sequentially applying multiple 3×3 convolutions, simulating the receptive field of the image. In this study, we use the VGG_13 network model as the base network for crack image segmentation. The network structure is shown in figure 1.

The VGG_13 network consists of 5 convolution blocks and 3 fully connected layers. The number of convolution kernels in the 5 convolution blocks are 64, 128, 256, 512, and 512 respectively. Each convolution block consists of 2 convolution layers, each with a convolution kernel size of 3×3. The convolution kernels slide over the input feature map, extracting features through weighted summation over local areas, as shown in equation 1.

$$Y_{i,j,k} = \mathrm{ReLU}(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{c=0}^{C-1} X_{i+m,j+n,c} \cdot W_{m,n,c,k}) \tag{1}$$

where $Y_{i,j,k}$ represents the value of the k-th channel at position (i,j) in the output feature map, $X_{i+m,j+n,c}$ represents the input image, W is the convolution kernel, and b_k represents the bias of the k-th channel. Each convolution block is followed by a max pooling layer, which is used for downsampling with a pooling kernel of 2×2, as shown in equation 2.

$$Y_{i,j,k} = \max_{m,n} X_{i+m,j+n,k} \tag{2}$$

where (i, j) is the top-left position of the pooling window. The last three layers of the network are fully connected layers, computed as shown in equation 3.

$$y = \sigma(Wx + b) \tag{3}$$

where W is the weight matrix, x is the input feature, and σ is the activation function, with ReLU used in the first two layers and Sigmoid in the last layer. To reduce the number of network parameters and speed up model training, the original number of nodes in the fully connected layers was reduced from 4096, 4096, 1000 to 2048, 2048, 8, respectively, ultimately achieving crack segmentation.

2.2 Dilated Convolution

Dilated convolution [21] , also known as atrous convolution, significantly expands the receptive field without increasing the number of parameters. In image segmentation, dilated convolution effectively captures multi-scale information, improving segmentation accuracy. It extends the receptive field by inserting holes (zeros) between convolution kernel elements without increasing the number of parameters, as shown in equation 4.

$$Y_{i,j,k} = \mathrm{ReLU}(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\sum_{c=0}^{C-1} X_{i+d\cdot m,j+d\cdot n,c} \cdot W_{m,n,c,k} + b_k) \tag{4}$$

where d is the dilation rate, indicating the spacing between kernel elements.

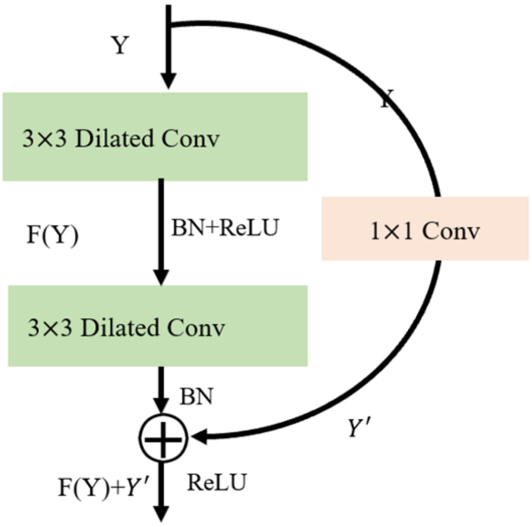


Figure 2: Dilated Residual Convolution

This study integrates residual modules [22] with dilated convolution, proposing dilated residual convolution to obtain deeper feature information. The structure of the dilated residual convolution is shown in Figure 2.

As shown in 2 , the dilated residual convolution consists of two 3×3 dilated convolution layers and one 1×1 convolution layer residual. Convolution layers are followed by batch normalization (BN) and ReLU activation function. $F(Y)$ represents the residual path, where the identity connection path Y' after a 1×1 convolution is added to the data path across the convolution, achieving the final result without introducing additional computational and parameter overhead.

3 Research method

This study proposes an improved VGG_13 model, combining residual dilated convolution and path enhancement to enhance the accuracy and detail processing capability of image segmentation. The input images are denoised using fuzzy morphological filtering before being used to train the improved neural network model. The model is based on an encoder-decoder architecture. The encoder, consisting of convolutional blocks, residual dilated convolution blocks, and path enhancement modules, extracts features from the crack images. The decoder, composed of deconvolution and upsampling modules, performs the crack segmentation.

3.1 Image Preprocessing

To effectively reduce crack noise and better enhance crack edge and detail information, this study proposes using fuzzy morphological filtering [23] for crack preprocessing. Fuzzy morphological filtering combines the advantages of fuzzy logic and mathematical morphology. Mathematical morphology enhances crack edge information and fills and connects cracks. Since grayscale images inherently possess fuzziness, using fuzzy logic effectively classifies and filters noise, thus avoiding noise amplification during dilation or erosion processes. The input image is converted to a grayscale image to facilitate fuzzy morphological operations. The image is then normalized and resized to 224x224 to standardize the dataset for the neural network. Finally, fuzzy morphological filtering is applied, consisting of fuzzy dilation, fuzzy erosion, and fuzzy opening and closing operations. Fuzzy dilation enhances image brightness by combining each pixel with the maximum value in its neighborhood, as calculated in Formula 5.

$$U_{A\oplus B}(s) = \sup_{p\in B} \min(U_A(\mu-\gamma), U_B(\gamma)) \tag{5}$$

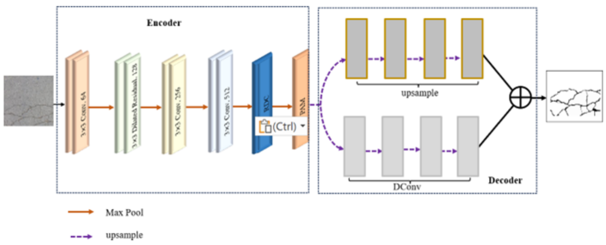


Figure 3: Schematic Diagram of the Improved VGG Network Model

where U_A represents the fuzzy set of the original crack image, U_B represents the fuzzy set of the structuring element, sup represents the supremum, and $minrepresents$ the minimum value. Fuzzy erosion weakens image brightness by combining each pixel with the minimum value in its neighborhood, as shown in equation 6.

$$U_{A\ominus B}(s) = \inf_{p\in B} \max(U_A(\mu+\gamma), 1 - U_B(\gamma)) \tag{6}$$

where inf represents the infimum, and max represents the maximum value. In the erosion operation, for each pixel in the image, the maximum value of its neighborhood corresponding to the structuring element is taken, and the minimum value of these maxima is used as the eroded pixel value. This way, erosion retains smaller pixel values and reduces larger ones in each local area. Fuzzy opening first performs fuzzy erosion followed by fuzzy dilation to remove small objects and noise, as calculated in equation 7.

$$U_{A\cdot B}(s) = (U_A \ominus B) \oplus B \tag{7}$$

Fuzzy closing first performs fuzzy dilation followed by fuzzy erosion to fill small holes and fine cracks in the image, as shown in equation 8.

$$U_{A\cdot B}(s) = (U_A \oplus B) \ominus B \tag{8}$$

Through fuzzy morphological filtering, crack image noise is significantly reduced while preserving and enhancing crack detail information.

3.2 Improved VGG_13 Model

To improve the accuracy and detail processing capability of crack segmentation, this study proposes an improved VGG crack segmentation network. Based on the VGG_13 network, this model integrates residual dilated convolution and path enhancement modules to enhance image segmentation performance. The improved model adopts an encoder-decoder architecture, as shown in Figure 3. The encoder retains the first four convolution layers of VGG_13, where the first, third, and fourth layers extract basic features of crack images through standard convolution operations. The second layer uses the dilated residual convolution module shown in Figure 2, utilizing convolution layers with different dilation rates to capture multi-scale features, thus expanding the receptive field and enhancing the ability to capture complex edges and details without significantly increasing computational cost. The fifth convolution block uses the improved dilated convolution module (DCM) to further enhance the network's ability to capture features at different scales. The RDC introduces a higher-level dilated convolution structure and multi-scale feature fusion mechanism, effectively enhancing the model's feature extraction capability and robustness. Following the RDC, a path enhancement module is introduced to improve feature expression capabilities at different levels. The PAM module enhances deep feature representation by utilizing shallow feature information through top-down feature transmission. The decoder part restores the image size through deconvolution and upsampling operations, achieving crack segmentation.

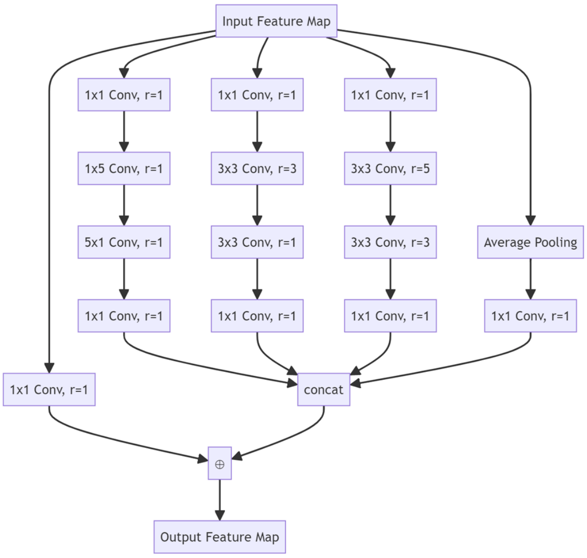


Figure 4: Residual Dilated Convolution Module

3.2.1 Residual Dilated Convolution Module

To enhance the network's feature extraction capability, an improved residual dilated convolution module (RDC) is proposed. This module captures feature information over a larger range by introducing convolution layers with different dilation rates and multi-scale feature fusion mechanisms, improving the ability to capture complex scenes and detail features. The RDC structure is shown in Figure 4.

As shown above, the input feature map Y is transformed through a series of convolution layers with different dilation rates. This configuration ensures effective capture and combination of multi-scale features. The improved residual dilated convolution module consists of four branches, with convolution dilation rates of 1, 3, and 5, respectively. These branches aim to capture features at different scales, and the convolution operation is shown in equation 9.

$$F_i(Y) = \sigma(W_i * Y + b_i) \tag{9}$$

where W_i and b_i are the weights and biases of the $*$ convolution layer, and $*$ represents the convolution operation. In the entire residual dilated convolution module, 1×1 convolution is used to capture local features and reduce dimensions. The 3×3 convolution (dilation rate=1) extracts standard convolution details. The 3×3 convolution (dilation rate=3) captures mid-range context by expanding the receptive field. The 3×3 convolution (dilation rate=5) extracts broader contextual information.

After processing through different branches of dilated convolution, all branch outputs are concatenated and fused through another 1×1 convolution, as shown in equation 10.

$$F(Y) = \sigma(W_f * \text{concat}(F_1(Y), F_2(Y), F_3(Y), F_4(Y) + b_f) \tag{10}$$

where W_f and b_f are the weights and biases of the 1×1 convolution layer. The final output is obtained through residual connection by adding the input feature map processed by the 1×1 convolution layer to the output of the improved residual dilated convolution module, as shown in equation 11.

$$I = Y^{''} + F(Y) \tag{11}$$

where $Y^{''}$ represents the result processed through the 1×1 convolution layer. The feature map processed by the dilated residual module retains the original features while enhancing the multi-scale context. To reduce the issue of local detail retention deficiency and increased computational complexity caused by high dilation rates, this study adjusts the convolution kernel size and fusion method. The dilation rates of 2 and 5 are adjusted to receptive fields of 5 and 10, respectively.

Spatially separable convolutions replace a 5×5 module with two 5×1 and 1×5 convolution modules, maintaining the receptive field while reducing network parameters and increasing nonlinearity and model expressiveness. A 1×1 convolution is added after each branch to adjust linear excitation and dimensions, reducing parameters and computation, enhancing efficiency. Global average pooling makes feature layers more easily transformed into classification probabilities, avoiding overfitting. Global average pooling computes the feature vector by averaging all pixel values in the feature map, as shown in equation 12.

$$\text{GAP} = \frac{1}{H \times M} \sum_{h=1}^{H} \sum_{m=1}^{M} Y_{ab} \tag{12}$$

where H is the height of the feature map, M is the width of the feature map, $Y_a b$ represents the pixel value at row a and column b in the feature map. $\sum_{h=1}^{H} \sum_{m=1}^{M} Y_{ab}$ represents the sum of all pixel values in the feature map, and $\frac{1}{H \times M}$ is the normalization factor.

3.2.2 Path Enhancement Module

To compensate for information loss caused by continuous pooling and convolution operations and improve the accuracy and efficiency of crack segmentation, this study proposes using a Path Augmentation Module (PAM) to optimize the model. The PAM module enhances deep feature representation by utilizing shallow feature information through top-down feature transmission. The PAM module extracts feature maps from shallow (low-level) and deep (high-level) layers of the network. Shallow feature maps contain detail and location information, while deep feature maps contain semantic information. The shallow feature map F_low is upsampled and element-wise added to the deep feature map, resulting in fused features, then processed by a convolution layer, as shown in equation 13.

$$F_{\text{fused}}(i,j) = F_{\text{low}}^{\text{up}}(i,j) + F_{\text{high}}(i,j) \quad \text{for } i \in [1, G_{\text{high}}], j \in [1, G_{\text{high}}] \tag{13}$$

The fused result is further processed by a convolution layer, as shown in equation 14.

$$F_{\text{enhanced}}(i,j) = \sum_{m=1}^{C} \sum_{n=1}^{C} \omega_{nm} \cdot F_{\text{fused}}(i+m-1, j+n-1) + b \tag{14}$$

where C is the convolution kernel size, ω_{nm} is the convolution kernel weight. By effectively utilizing shallow feature information, the PAM module enhances deep feature representation, improving the accuracy and detail capture of crack segmentation.

3.3 Loss Function

To ensure the model focuses on important detail information and enhances the detection and segmentation of fine cracks, this study uses the Dice coefficient loss function to optimize network parameters. The Dice coefficient is a statistical measure of similarity between two sample sets. During forward propagation, the predicted values and ground truths are computed, and the Dice loss function is calculated as shown in equation 15.

$$\text{Loss}_{\text{Dice}} = 1 - \frac{2\sum_{\psi=1}^{O} Q_\psi D_\psi}{\sum_{\psi=1}^{O} Q_\psi + \sum_{\psi=1}^{O} D_\psi} \tag{15}$$

where Q_ψ represents the predicted value of the ψ-th pixel,D_ψ represents the true label of the ψ-th pixel, and O represents the total number of pixels. During backpropagation, the gradient of the loss function concerning the predicted value Q_ψ is calculated, as shown in equation 16.

$$\frac{\partial \text{Loss}_{\text{Dice}}}{\partial Q_\varphi} = \frac{\partial}{\partial Q_\varphi}\left(1 - \frac{2\sum_{\varphi=1}^{O} Q_\varphi D_\varphi}{\sum_{\varphi=1}^{O} Q_\varphi + \sum_{\varphi=1}^{O} D_\varphi}\right) \tag{16}$$

Let Z=$\sum_{\varphi=1}^{O} Q_\varphi + \sum_{\varphi=1}^{O} D_\varphi$ and E=$\sum_{\varphi=1}^{O} Q_\varphi$, then the Dice loss function is simplified as shown in equation 17.

$$\text{Loss}_{\text{Dice}} = 1 - \frac{2\sum_{\varphi=1}^{O} Q_\varphi D_\varphi}{Z+E} \tag{17}$$

Next, the gradient is calculated by applying the chain rule, computing the numerator and denominator, resulting in the final gradient formula for the Dice loss function concerning the predicted value Q_ψ, as shown in equation 18.

$$\frac{\partial \text{Loss}_{\text{Dice}}}{\partial Q_\varphi} = \frac{\partial}{\partial Q_\varphi}(1 - \frac{2\sum_{\varphi=1}^{O} Q_\varphi D_\varphi}{Z+E}) \tag{18}$$

Through substitution and simplification, the final gradient formula is derived. Using gradient descent, the gradient is calculated, and network weights are updated to minimize the loss function, enhancing the modelś segmentation performance.

4 Experiment analysis

4.1 Experimental Environment

The network is implemented using the deep learning framework TensorFlow, and the experiments are conducted on an HP Z8 G4 graphics workstation. The device is equipped with two Intel Xeon 5218R CPUs, each with 20 cores and a clock speed of 2.1 GHz, totaling 40 cores. The graphical processing capability is provided by an A4000 series 16GB GPU. The system memory is 128GB, and the storage solution includes a 512GB SSD and a 2TB HDD. The operating system is Linux, and tests are conducted on self-built datasets, SDNET2018, Crack500, and CFD datasets. Dataset information is shown in table 1.

Table 1: Dataset Information

No	Dataset Name	Data Description
1	Self-built Dataset	12000 manually labeled crack images
2	SDNET2018	56000 images of mixed concrete
3	Crack500	500 crack images
4	CFD	118 crack images

In total, the dataset contains 68,618 images. To expand data resources, this study uses rotation, translation, and flipping methods for data augmentation to increase data diversity. After augmentation, the dataset contains 117,236 images. To test the model's effectiveness, the study uses the SDNET2018 and a random 70% portion of the self-built dataset for model training, with validation on the self-built dataset, Crack500, and CFD datasets.

4.2 Evaluation Metrics

To quantitatively analyze the differences between algorithms, common image segmentation evaluation metrics are used to compare the performance of this algorithm with other algorithms, including Accuracy, Recall, Mean Pixel Accuracy (MPA), and Mean Intersection over Union (MIoU). Accuracy evaluates the overall accuracy of segmentation results, calculated as shown in equation **??**.

$$\text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN} \tag{19}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. Accuracy reflects the proportion of correctly classified samples, ranging from [0, 1], with higher values indicating better segmentation performance. Recall evaluates the model's detection capability, calculated as shown in equation 20.

$$\text{Recall} = \frac{TP}{FN+TP} \tag{20}$$

Recall reflects the proportion of actual positives correctly identified, ranging from [0, 1], with higher values indicating better segmentation performance. Mean Pixel Accuracy evaluates the pixel classification accuracy of each category, calculated as shown in equation 21.

$$\text{MPA} = \frac{1}{T}\sum_{q=1}^{L}\frac{TP_q}{FN_q + TP_q} \tag{21}$$

where L is the number of categories, and TP_q and FN_q are the true positive and false negative of the $q-th$ category, respectively. MPA reflects the average accuracy of each category, ranging from [0, 1], with higher values indicating better segmentation performance. Mean Intersection over Union evaluates the overlap between segmentation results and actual results, calculated as shown in equation 22.

$$\text{MIoU} = \frac{1}{T}\sum_{q=1}^{L}\frac{TP_q}{FN_q + TP_q + FP_q} \tag{22}$$

where L is the number of categories, TP_q FN_q and FP_q are the true positive, false positive, and false negative of the q-th category, respectively. $MIoU$ reflects the average intersection over union of each category, ranging from [0, 1], with higher values indicating better segmentation performance.

4.3 Analysis of Image Denoising with Different Algorithms

To validate the effectiveness of image preprocessing, the study uses fuzzy mathematical morphology filtering, bilateral filtering, Gaussian filtering, and wavelet filtering for denoising, as shown in Figure **??**.

From Figure 5, it can be seen that fuzzy mathematical morphology filtering effectively filters out white noise in images without affecting the detail features of cracks in the original image, while wavelet filtering shows poor noise reduction performance and significant detail loss during denoising. To objectively validate the denoising performance of algorithms, the study randomly selects 200 crack images for denoising using fuzzy mathematical morphology filtering, bilateral filtering, Gaussian filtering, and wavelet filtering, and evaluates them using the Structural Similarity Index (SSIM), Mean Squared Error (MSE), and Normalized Correlation (NC), with results retained to two decimal places, as shown in Table 2.

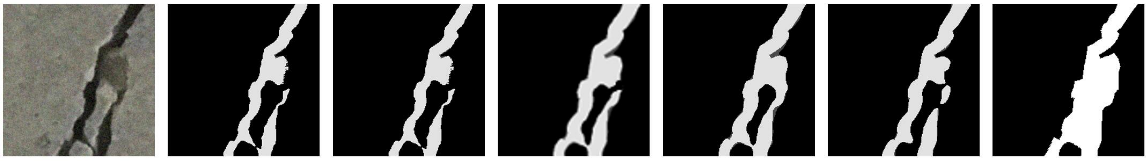
Table 2: Performance Evaluation of Different Filters

Filter	SSIM	MSE	NC
Fuzzy Mathematical Morphology	0.98	0.27	0.99
Bilateral Filtering	0.42	0.03	0.93
Gaussian Filtering	0.22	0.20	0.92
Wavelet Filtering	0.18	0.12	0.91

As shown in Table 2, the best SSIM, MSE, and NC results are all from fuzzy mathematical morphology filtering, indicating its excellent performance in handling complex structures and retaining details. This is due to fuzzy mathematical morphology filtering based on fuzzy set theory, which allows for effective handling of uncertainty and fuzziness. Crack edges are often fuzzy and irregular, and fuzzy set theory can better describe and retain these fuzzy edges through pixel gray level fuzzification. Additionally, fuzzy mathematical morphology filtering uses multi-scale structuring elements to process images at different scales, effectively capturing structural features of different scales. Thus, fuzzy mathematical morphology filtering can remove noise while preserving image structure and details, enhancing segmentation accuracy and reliability.

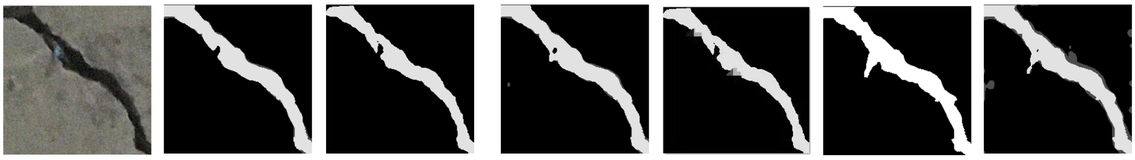
4.4 Analysis of Segmentation Effects with Different algorithms

To validate the effectiveness of our algorithm, the study trains and validates the model on self-built datasets and SDNET2018, and verifies it on self-built, Crack500, and CFD crack datasets using CNN [4], HED [20], VGG_19 [10], UNet++ [14] ,and our algorithm, as shown in Figures 5 to 7.



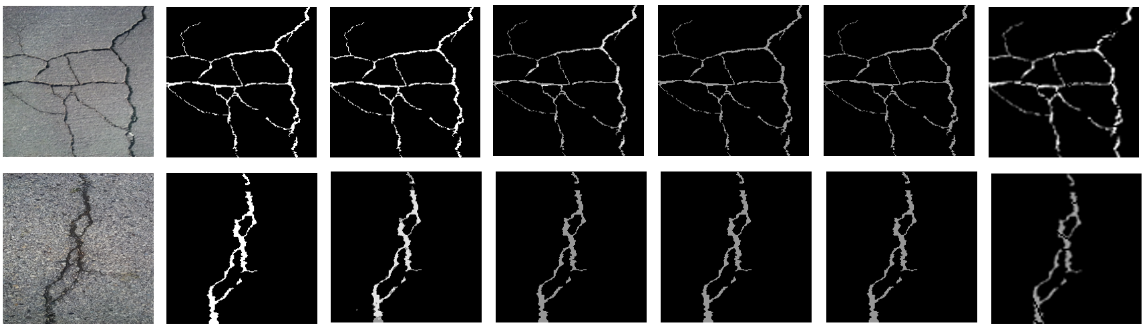
(a) Original Image (b) Label (c) Our Algorithm (d) UNet++ (e) VGG_19 (f) HED (g) CNN

Figure 5: Segmentation Effects on Self-Built Dataset with Different Algorithms



(a) Original Image (b) Label (c) Our Algorithm (d) UNet++ (e) VGG_19 (f) HED (g) CNN

Figure 6: Segmentation Effects on CFD Dataset with Different Algorithms



(a) Original Image (b) Label (c) Our Algorithm (d) UNet++ (e) VGG_19 (f) HED (g) CNN

Figure 7: Segmentation Effects on Crack500 Dataset with Different Algorithms

From Figures 6 to 8, it can be observed that HED and CNN perform the worst on the self-built dataset, CDF dataset, and Crack500 dataset. On the self-built dataset, these methods incorrectly identify imprints as cracks, adversely affecting image segmentation results. In the CFD dataset, HED and CNN are heavily influenced by road particle noise, identifying noise particles as cracks, which impacts segmentation accuracy. On the Crack500 dataset, both algorithms exhibit poor segmentation clarity. VGG_19 performs better, as it can somewhat filter out noise in the CFD dataset and correctly handle imprints in the self-built dataset. However, it still has room for improvement, especially in segmenting clear areas on the Crack500 dataset. UNet++, with its U-shaped architecture and skip connections, performs better than the previous three methods, but it still lacks in handling fine crack details accurately and clearly. Our algorithm effectively filters out road particle noise and imprint noise, accurately identifies fine crack boundaries, and achieves better crack detail segmentation. This performance is attributed to our algorithm's use of fuzzy mathematical morphological filtering, which reduces noise while preserving detail features. The multi-scale dilated convolution module and path enhancement module help maintain fine features while capturing contextual information. To objectively evaluate the accuracy of our algorithm in image crack segmentation, this study uses the self-built dataset, Crack500, and CFD datasets for validation. The experiments use accuracy (ACC), mean pixel accuracy (MPA), mean intersection over union (MIoU), and recall as evaluation metrics. The average values of these metrics are taken to eliminate machine errors, with results rounded to two decimal places as shown in Tables 3 to 5.

Table 3: Segmentation Performance Analysis on Self-Built Dataset with Different Algorithms

Algorithm	Accuracy	MPA	MIoU	Recall
CNN	92.31	94.11	92.04	90.02
HED	94.33	94.74	92.51	92.53
VGG19	95.26	95.03	93.58	95.95
UNet++	96.34	96.11	93.78	96.16
Our Algorithm	96.97	96.86	94.98	96.87

Table 4: Segmentation Performance Analysis on Crack500 Dataset with Different Algorithms

Algorithm	Accuracy	MPA	MIoU	Recall
CNN	92.31	93.11	90.54	91.16
HED	94.33	94.74	91.08	92.53
VGG19	95.26	95.63	91.70	93.86
UNet++	95.84	96.21	92.78	95.21
Our Algorithm	96.37	97.36	93.18	95.92

Table 5: Segmentation Performance Analysis on CFD Dataset with Different Algorithms

Algorithm	Accuracy	MPA	MIoU	Recall
CNN	88.13	89.02	89.25	89.12
HED	91.03	91.78	91.89	90.80
VGG19	94.56	94.81	94.33	94.14
UNet++	94.85	95.01	94.89	94.78
Our Algorithm	95.21	95.87	95.63	96.31

From Tables 3 to 5, it is evident that CNN performs relatively poorly across all datasets, particularly in complex scenarios where it struggles to capture fine crack features. Although the HED algorithm outperforms CNN, it still falls short compared to VGG_19 and UNet++, indicating room for improvement in feature extraction and detail recovery. Despite decent accuracy, HED's performance in mean intersection over union (MIoU) and recall is somewhat lacking, possibly due to its heavy reliance on deep networks for feature extraction, which can hinder detail recovery. UNet++, as an improved version of Unet, demonstrates better segmentation performance than CNN, HED, and VGG_19, but it still does not surpass our proposed algorithm, highlighting the architectural optimizations in our method. Our algorithm exhibits outstanding segmentation performance on the self-built dataset, CRACK500 dataset, and CFD dataset. It achieves the highest accuracy (96.97%) and mean pixel accuracy (96.86%) across all datasets, indicating excellent fitting capability to the training data and precise extraction and segmentation of crack features. On the CRACK500 dataset, our algorithm also leads in all metrics, particularly in accuracy (96.37%) and mean pixel accuracy (97.36%). This demonstrates not only its superior performance on the self-built dataset but also its strong generalization ability, maintaining high performance across different datasets. The CFD dataset, often containing more complex crack scenarios, shows our algorithm with significantly higher mean intersection over union (91.89%) and recall (94.19%) compared to other algorithms, underscoring its advantage in handling complex and noisy environments. The exceptional performance of our algorithm in crack segmentation tasks is attributed to its innovative encoder-decoder design, combining dilated residual concepts and multi-scale feature extraction, enabling better capture of crack details and features. The integration of deconvolution and upsampling effectively restores image details during the decoding process. Furthermore, its excellent performance across different datasets showcases its strong generalization ability and robustness.

4.5 Comparison of Different Iteration Numbers

To explore the patterns of network training, we conducted experiments on the self-built dataset to study how different iteration numbers correspond to different experiment results, as shown in Table 6. As shown in Table 6, with the increase in iteration numbers, all evaluation metrics (Accuracy, MPA, MIoU, Recall) significantly improve, particularly in the first four iterations. The metrics reach their peak when the iteration number is 4. Beyond this point, further increases in iteration numbers result in a gradual decline in detection performance. The experiments indicate that the proposed algorithm achieves the best detection results at four iterations. Moderately increasing the iteration number can enhance the model's detection performance, but excessive iterations may lead to a plateau or even a slight decrease in performance. Selecting an appropriate number of iterations is crucial for optimizing the model.

Table 6: Analysis of Different Iteration Numbers

iteration number	Accuracy	MPA	MIoU	Recall
Number =1	88.84	91.65	89.69	93.12
Number =3	95.73	95.65	93.79	96.12
Number =4	96.95	96.86	94.98	96.87
Number =5	96.82	96.80	93.11	96.73
Number=6	96.46	96.73	92.91	96.48

4.6 Runtime and Efficiency Comparison

To analyze the real-time performance of various algorithms in image crack segmentation, we conducted segmentation time statistics for CNN [4], HED [20], VGG_19 [10], UNet++ [14], and our algorithm on the self-built dataset, Crack500, and CFD datasets. The average value of the three results was taken for comparative experiments. Additionally, we observed the frames processed per second (FPS) and the peak speed per second (FLOPS). The experimental results are shown in Table 7. From Table 6, it is evident that our algorithm leads with an average segmentation time of 0.09 seconds. In terms of FPS, our algorithm achieves 9.39, indicating good real-time performance. The high efficiency in segmentation time is due to the encoder part quickly extracting multi-scale features through multiple convolutional layers and dilated residual convolutions, expanding the receptive field without increasing computational load. The high FPS is attributed to the efficient decoder design, which rapidly restores image details through upsampling and deconvolution operations, reducing processing latency. UNet++ follows closely, also demonstrating good real-time performance. The other three algorithms are comparatively slower in processing speed. In terms of FLOPS, our

Table 7: Comparison of Runtime and Efficiency of Different Algorithms

Algorithm	Average Segmentation Time (s)	FPS	FLOPS
CNN	0.27	7.35	91.30
HED	0.21	8.43	112.32
VGG19	0.43	7.86	189.15
Unet++	0.11	9.21	199.01
Our Algorithm	0.09	9.39	234.38

Table 8: Ablation Study Results

Algorithm	ACC	MPA	MIoU	Recall	FPS	FLOPS
VGG13	94.56	94.81	90.33	93.12	7.66	136.52
VGG+denoising	95.05	94.91	92.13	93.89	7.90	138.88
VGG+residual dilation+denoising	95.57	95.19	92.54	95.65	8.65	192.81
VGG+enhanced path+denoising	95.08	95.07	92.88	95.38	8.45	195.66
Our Algorithm	96.09	96.36	93.05	96.09	9.39	234.38

algorithm significantly outperforms the other four, achieving 234.38. This advantage stems from the multi-scale residual dilated module and path enhancement module, which capture features at different scales through multi-scale convolution operations, thereby significantly increasing the FLOPS value while maintaining high-quality segmentation images. The experiments demonstrate that our algorithm finds a good balance between processing speed, computational efficiency, and segmentation accuracy, making it suitable for various practical application scenarios. To further validate the superiority of our proposed algorithm in image segmentation, ablation experiments were conducted on the self-built dataset and the CFD dataset. We compared our algorithm with VGG13, VGG+denoising, VGG+residual dilation+denoising, and VGG+enhanced path+denoising modules. The performance was evaluated using ACC, accuracy, MPA, MIoU, and average segmentation time. The results are shown in Table 8.

From Table 7, it is evident that our algorithm demonstrates superior effectiveness. By introducing dilated residual modules and path enhancement modules into the VGG network for feature extraction, our algorithm can capture more image details and complex crack features, thereby improving segmentation accuracy. The use of deconvolution and upsampling in the decoder allows the restoration of the original resolution of the image while retaining high-level feature information, ensuring the preservation and accurate segmentation of crack details. The filtering method helps smooth the image while preserving and enhancing important details, which accounts for the superior performance in ACC, MPA, MIoU, and Recall. In terms of average segmentation time and FPS, our algorithm demonstrates a clear advantage in operational efficiency. Thanks to the efficient computational framework of the algorithm, optimized process flow, parallel upsampling and deconvolution, and GPU acceleration, these technologies enhance processing speed and frame rate per second. The high FLOPS data indicates the algorithm's efficiency in floating-point operations per second, resulting from the careful balance between computational complexity and execution efficiency in the algorithm's design. Through the reasonable design of the network structure and optimization of the computation process, our algorithm achieves efficient image segmentation without sacrificing accuracy.

5 Conclusion

This study proposes an improved VGG-based neural network model for crack detection, featuring dilated residual convolution and path enhancement integration. The experimental results indicate significant improvements in crack segmentation accuracy and detail segmentation precision. Specifically, fuzzy morphological filtering effectively reduces noise interference in input images, enhancing crack detection accuracy. The introduction of dilated convolution and residual structures significantly expands the network's receptive field and strengthens feature extraction capabilities. The Path Augmentation Module (PAM) enhances the network's ability to express features at different scales and capture details through top-down feature transmission and multi-scale feature fusion. Validation on the CDF, self-built, and Crack500 datasets demonstrates the algorithm's notable advantages in detail handling, segmentation accuracy, and efficiency. However, it is worth noting that the introduction of dilated convolution and path enhancement modules increases overall computational load, particularly when processing high-resolution images. Additionally, noise interference in complex backgrounds may still affect detection accuracy, especially when the contrast between cracks and the background is low, leading to false detections or missed detections. Future research will focus on optimizing lightweight network structures, exploring the use of depthwise separable convolutions and pointwise convolutions to reduce model parameters and computational complexity, thereby improving efficiency in high-resolution image processing. Further exploration and validation of the algorithm in various domains and scenarios, such as crack detection in bridges, tunnels, and roads, will expand its application range and practical value. In conclusion, the proposed algorithm demonstrates significant advancements in the field of image crack segmentation. With continuous refinement and optimization, this algorithm is expected to exhibit strong practical applicability across various image segmentation tasks, promoting the advancement and application of crack detection technology.

References

[1] Black & Veatch. The risks of aging infrastructure and the value of asset management, 2021.

[2] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv preprint arXiv:1411.4038*, 2015.

[3] H. Li, W. Wang, M. Wang, L. Li, and V. Vimlund. A review of deep learning methods for pixel-level crack detection. *Journal of Traffic and Transportation Engineering (English Edition)*, 9(6):945–968, 2022.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[5] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arxiv preprint arxiv:1409.1556. *arXiv preprint arXiv:1409.1556*, 2014.

[6] J. Wu and X. Zhang. Tunnel crack detection method and crack image processing algorithm based on improved retinex and deep learning. *Sensors*, 23(22):9140, 2023.

[7] S. Hao, L. Shao, and S. Wang. A faster rcnn airport pavement crack detection method based on attention mechanism. *Academic Journal of Science and Technology*, 2023.

[8] V. Mandal, L. Uong, and Y. Adu-Gyamfi. Automated road crack detection using deep convolutional neural networks. *Journal of the American Society for Information Science and Technology*, 69(10):2145–2157, 2018.

[9] S. Anand, S. Gupta, V. Darbari, and S. Kohli. Crack-pot: Autonomous road crack and pothole detection. In *Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–6, Canberra, ACT, Australia, 2018.

[10] M. Zeeshan, M. Adnan, W. Ahmad, and F. Z. Khan. Structural crack detection and classification using deep convolutional neural network. *Pakistan Journal of Engineering and Technology*, 2021.

[11] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer, 2015.

[12] V. Polovnikov, D. Alekseev, I. Vinogradov, and V. G. Lashkia. Daunet: Deep augmented neural network for pavement crack segmentation. *IEEE Access*, 2021.

[13] Runze Fan, Yuhong Liu, Rongfen Zhang, and Jingyu Li. A road scene semantic segmentation model based on multi-scale attention mechanism. *Computer Engineering*, 49(2):288–295, 2023.

[14] Zhiqiang Zhou, Muhammad Rafiqul Siddiquee, Nasir Tajbakhsh, and Jun Liang. Unet++: A nested u-net architecture for medical image segmentation. *IEEE Transactions on Medical Imaging*, 2018.

[15] Hao Luo, Jun Li, Li Cai, and Ming Wu. Strans-yolox: Fusing swin transformer and yolox for automatic pavement crack detection. *MDPI*, 2023.

[16] Zhihua Zhang, Yanan Wen, Haowei Mu, and Xiaoping Du. Road crack detection combined with dual attention mechanism. *Chinese Journal of Image and Graphics*, 07:2240–2250, 2022.

[17] Arash Saberironaghi and Jun Ren. Depthcracknet: A deep learning model for automatic pavement crack detection. *Journal of imaging*, 10(5):100, 2024.

[18] Ioannis Katsamenis, Evangelos Protopapadakis, Nikos Bakalos, Anastasios Varvarigos, and Athanasios Doulamis. A few-shot attention recurrent residual u-net for crack segmentation. *arXiv*, 2023.

[19] Yi Fan, Tian Wu, Guangyu Cao, Yue Zhao, Yifeng Yang, and Yukun Li. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[20] Sa Xie and Zhuowen Tu. Holistically-nested edge detection. *arXiv preprint arXiv:1504.06375*, 2015.

[21] Fisher Yu and Vlad Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2016.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778)*. IEEE, 2016.

[23] Yuan Chen, Guo-Qing Zhou, Zhi-Hua Wang, and Hong Zhang. Fuzzy mathematical morphology and its application to image processing. *Pattern Recognition Letters*, 62:92–102, 2015.

Cite this paper as:

WANG, X.; WU, J.; CHEN, X.; HOU, J.; CHEN, P. (2025). Optimized VGG Network with Dilated Residual Convolution and Path Enhancement for Crack Image Segmentation, *International Journal of Computers Communications* & *Control*, 20(2), 6872, 2025.
https://doi.org/10.15837/ijccc.2025.2.6872