Structural Durability & Health Monitoring

Tech Science Press

https://doi.org/10.32604/sdhm.2026.073282

ARTICLE

Concrete Bridge Defect Monitoring and Quantitative Identification via U-Net and Mathematical Morphology

Caiping Huang*, Yulong Mei, Wangyuan Tian and Zihang Yu

School of Civil Engineering, Architecture and Environment, Hubei University of Technology, Wuhan, China

*Corresponding Author: Caiping Huang. Email: cphuang@hbut.edu.cn

Received: 15 September 2025; Accepted: 26 November 2025; Published: 24 August 2026

ABSTRACT: Bridge damage detection is critical to bridge maintenance practices. However, traditional inspection methods are plagued by high labour intensity and low operational efficiency. To enhance the intelligence, objectivity, and efficiency of bridge damage detection, this study proposes an automated approach for the identification and quantitative measurement of concrete defects. Specifically, this method adopts the Visual Geometry Group (VGG) network as the backbone of the U-Net architecture to perform semantic segmentation on images containing typical concrete defects, including spalling, cracks, and exposed reinforcement bars. Subsequently, mathematical morphology algorithms are employed to optimise the segmented images, thereby mitigating errors induced by semantic segmentation. Utilising MATLAB software, the area (or length) of concrete defects is quantified by referencing objects with known dimensions. To validate the proposed method, semantic segmentation experiments were conducted on defect images collected from 17 in-service reinforced concrete bridges in Xuchang City, Henan Province. Experimental results indicate that the VGG16-U-Net model not only accurately locates and classifies three common concrete defects under complex background conditions but also enables the quantification of their area or length. The obtained results fully meet the requirements of practical engineering inspections. For the identification of concrete spalling, cracks, and exposed reinforcement, the model achieved a Mean Pixel Accuracy (MPA) of 90.53% and a Mean Intersection over Union (MIoU) of 80.54%. Furthermore, the application of mathematical morphology to optimize segmentation errors further improved the accuracy of defect quantitative calculation, with the optimized absolute error ranging from 0.08% to 0.21%.

KEYWORDS: Concrete defects; deep learning; U-Net; mathematical morphology; quantitative calculation

1 Introduction

With the increase in bridge age and load, as well as the imperfect management system and maintenance work, the bridge structure has appeared with different degrees of defects [1]. Traditional damage inspection methods include manual inspection, regular inspection, and special inspection, etc. [2–4]. These inspections rely on manual operation, which not only requires a lot of manpower and material resources, but also has problems such as inspection blind spots and influence on bridge operation. And the accuracy of the results is subject to the subjective influence of technicians [5–7]. The use of UAVs to photograph bridge disease images to replace part of the manual operation allows the inspector to be remotely controlled in the background, which is safe and fast. However, the collected images still need to be screened one by one by the detection personnel, and the detection results are still subject to the subjective influence of the detection personnel [8–10].

With the rapid development of computer technology, the rise of computer vision technology makes image processing more intelligent and convenient, as an emerging field that partially replaces artificial vision inspection, vision-based methods have been used for defect detection, and they provide many advantages for structural health monitoring (SHM) applications [11]. Machine learning techniques, particularly deep neural networks, have led to breakthroughs in civil engineering applications for several research groups. Pang et al. [12] combined deep learning techniques with lightweight network design to detect and track concrete surface defects in real time. Arbaoui et al. [13] proposed a method that integrates wavelet multi-resolution analysis with deep learning for the efficient monitoring of cracks in concrete structures. This method enables the early detection of cracks before they become visible on the surface, thereby providing robust support for the maintenance of civil engineering structures. Pozzer et al. [14] employed a Siamese Neural Network with multimodal thermal and visible images to reduce false positives in subsurface delamination segmentation for concrete structures.

Cha et al. [15] proposed a structural vision detection method based on Faster Region-based Convolutional Neural Network (Faster R-CNN), which realized quasi-real-time simultaneous detection of five types of damages—concrete crack, steel corrosion with two levels (medium and high), bolt corrosion, and steel delamination. Huang et al. [16] established a depth residual network model to classify four common defects of concrete. Kruachottikul et al. [17] developed a visual defect detection system for reinforced concrete bridge infrastructure based on deep learning to classify different types of defects such as cracks, erosion, honeycomb, scaling, and spallation. Jin et al. [18] proposed a method based on deep learning and unmanned aerial vehicles (UAVs) to promote crack detection in structural tall buildings. Chen et al. [19] proposed a new road crack detection and classification method combining local binary model (LBM), principal component analysis (PCA), and support vector machine (SVM). The study involved the analysis of road surface images obtained through driving tests to accurately detect and classify cracks. Huang et al. [20] proposed a crack instance segmentation method based on deep learning (DL), and used Mask region-based convolutional neural network (MASK R-CNN) to segment cracks in shield tunnel lining images.

These studies have successfully achieved the automated classification and localization of concrete defects within images, but few address the quantitative dimension estimation of defects. During bridge health assessments, if the concrete defects in an image could be automatically identified by type and location, as well as have their geometric features (such as area and length) calculated, it would significantly enhance the efficiency and objectivity of bridge inspections.

Therefore, this study developed a comprehensive dataset of concrete defects (including spalling, cracks, and exposed steel rebar), utilized VGG16-U-Net for semantic segmentation of these defect images, and applied image processing techniques to the segmented images. The first step involves converting images containing multiple defects into single defect images, followed by grayscale and binarization to convert them into black and white images. Subsequently, mathematical morphology algorithms are utilized to optimize the contours of the defects in the aforementioned black and white images. Furthermore, the quantity of pixels within the defect region in the optimized segmented images is calculated using MATLAB software; the area of a single pixel is obtained by using a reference object with a known area, and then used to determine the area (or length) of concrete defects. Finally, this approach is utilized to automatically identify and quantitatively calculate the area or length of defects in three images, each containing spalling, crack, and exposed steel rebar, respectively, with the research route shown in Fig. 1.

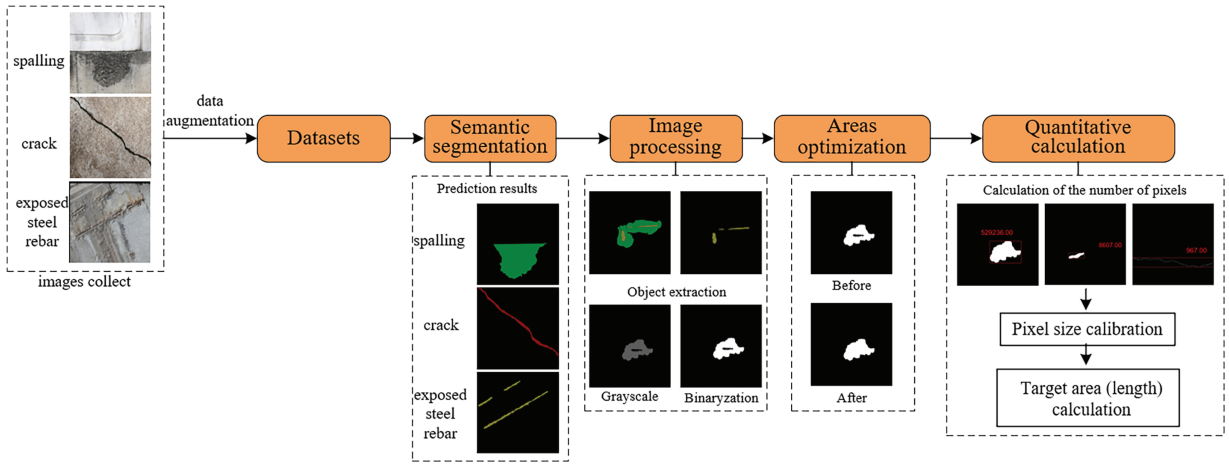


Figure 1: Flow chart for quantitative identification of concrete defects

2 Methodology

2.1 Semantic Segmentation

Semantic segmentation involves classifying each pixel in an image and distinguishing different categories with different colors. U-Networks (U-Net) is currently widely utilized for semantic segmentation.

The U-Net [21], a variant of the Fully Convolution Neural Network (FCN), was introduced in 2015. It is characterized by a symmetric U-shaped architecture consisting of an encoder-decoder and skip connections, providing a straightforward yet efficient design. The encoder is responsible for feature extraction, and other neural networks, such as the Visual Geometry Group Network (VGG) or Residual Network (ResNet101), can also be utilized in this capacity.

Compared with the residual connections in ResNet34, which tend to induce the "dilution" of fine-grained features (e.g., crack edges and micro-honeycomb structures), the complex operators integrated in EfficientNet pose notable deployment barriers in practical engineering scenarios—particularly for edge computing devices deployed at construction sites. In contrast, the depthwise separable convolutions adopted by MobileNet compromise the model's discriminative capability between defects and background interference (e.g., intrinsic concrete textures). By comparison, VGG16 adopts an architectural design characterized by "stacked 3 × 3 small convolutional kernels combined with gradual receptive field expansion." This design enables the model to more precisely capture critical defect details, such as millimeter-scale crack contours and the boundary contours of honeycomb pores. Furthermore, VGG16's standard convolutional architecture eliminates the need for specialized operators, thereby facilitating its direct adaptation to low-computational-power defect detection devices in on-site construction environments.

In this paper, the VGG16 is utilized as the encoder in the U-Net network. It consists of 16 layers, among which the first 13 layers are convolution layers and the last three layers are fully connected layers. In comparison to the original U-Net, the third, fourth, and fifth convolution layers have been increased from 2 layers to 3 layers, allowing for the extraction of more features. The size of convolution nuclei of each convolution layer is 3 × 3, the number of convolution nuclei from the first to the fifth layers is 64, 128, 256, 512, 512, respectively, the activation function is Rectified Linear Unit (ReLU), which is designed to incorporate nonlinear factors in order to enhance the expressiveness and performance of the model.

The architecture of VGG16-U-Net and the parameters of each layer constructed in this paper are illustrated in Fig. 2, encompassing four main components: encoder, decoder, feature fusion, and output layer.

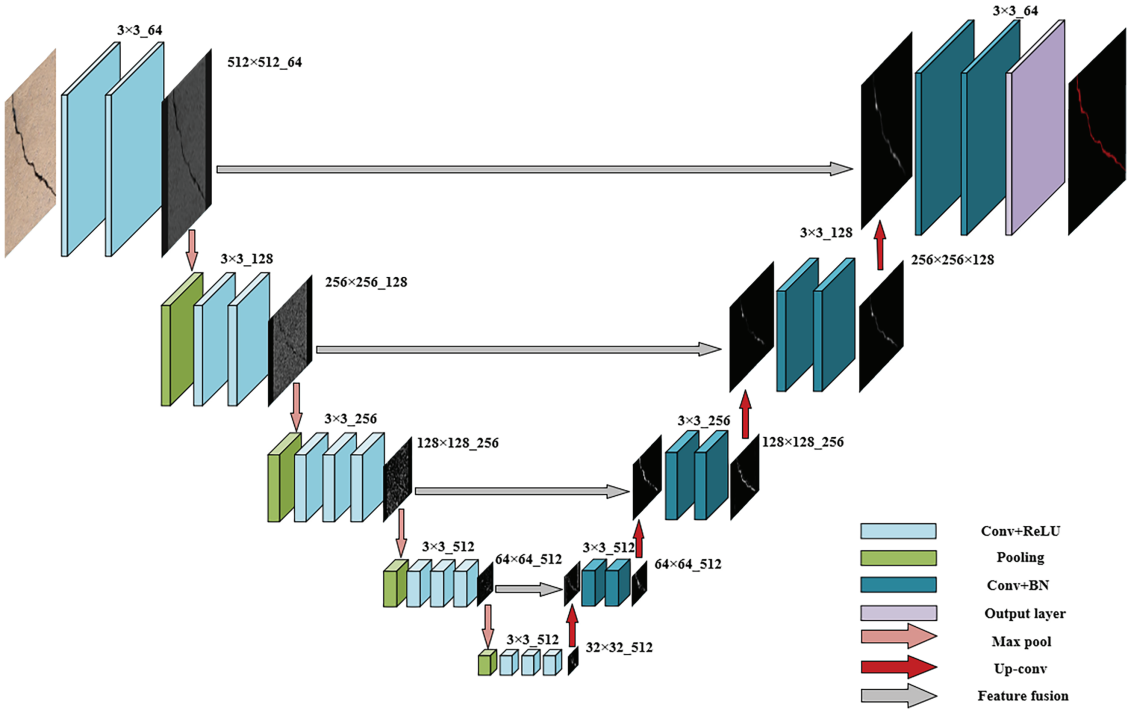


Figure 2: VGG16-U-Net Network structure

(1) Encoder: The feature extraction process is conducted by the convolution layer of the encoder, followed by dimension reduction of the feature map through the pooling layer. As depicted in Fig. 2, the image size is halved through Max pooling, leading to memory optimization and prevention of overfitting. The input image undergoes four rounds of maximization, resulting in the acquisition of four feature maps with varying sizes.

(2) Decoder: The decoder utilizes bilinear interpolation to up-sample the feature map generated by the encoder. Following each up-sampling operation, the size of the feature map is doubled. The incorporation of Batch Normalization accelerates the convergence of model training, mitigates the issue of untrainability caused by gradient vanishing or exploding, and enhances the overall generalization performance of the model.

(3) Feature fusion: In the VGG16-U-Net, feature fusion occurs four times. The up-sampled feature map is combined with the same-sized feature map from the encoder to create a new fused feature map, followed by a convolution operation. This fusion and convolution operation not only prevents the loss of edge features caused by feature extraction, but also extracts rich semantic information from deeper features.

(4) Output layer: After undergoing 4 rounds of feature fusion and 4 rounds of up-sampling, the feature map is restored to its original image size, and a segmentation map representing the probability of each class at every pixel position is generated using the Softmax function.

2.2 Semantic Segmentation Loss Function

The loss function is used to assess the degree of error between the model's predictions and the actual results. According to the error value calculated by the loss function, the model continuously optimizes the

training parameters so that the predicted value of the model is constantly close to the real value. In the semantic segmentation task of this paper, the number of pixels in the positive sample (target) in some images (cracks and exposure of reinforcement) is much less than that in the negative sample (background), and there is an imbalance between positive and negative samples. Dice loss function has a good performance in the task of segmentation with unbalanced positive and negative samples. Therefore, this paper adopts the Dice loss function as the main loss function to alleviate the problem of unbalanced positive and negative samples. The formula is as follows:

$$L_{Dice} = 1 - \frac{2\sum_{i=0}^{N} p_i t_i + \varepsilon}{\sum_{i=0}^{N} p_i + \sum_{i=0}^{N} t_i + \varepsilon} \tag{1}$$

In the formula, p_i is the predicted value, t_i is the true value, and ε is the smoothing coefficient, which is set to 10^{-5} to prevent the denominator from being zero or a minimum.

Dice loss function is used to calculate the loss by globally examining the similarity of all similar pixels between the prediction graph and the label graph. By ignoring a large number of background pixels, the problem of quantity imbalance of positive and negative samples is alleviated. Therefore, on some extremely unbalanced data, it may be impossible to learn the correct gradient descent direction, resulting in difficulties in training, causing instability in the training process of the model, and affecting the training results.

In this study, the Balanced Cross-Entropy loss function is used as an auxiliary loss function to adjust the above problems, guide the model to learn from different perspectives, and improve the generalization ability of the model. Balanced cross-entropy loss adds a weight coefficient to the Cross-Entropy loss function to balance the weights of positive and negative samples. For multi-class problems, corresponding weight adjustment can be made for each class to alleviate the influence of positive and negative sample imbalance [22]. The formula is as follows:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} x_i \alpha log\left(p_i\right) \tag{2}$$

In the formula, N is the total number of pixels; x_i is the label value of the I-th pixel point; p_i is the predicted value of the I-th pixel. α: balance coefficient, typically set to 0.25.

2.3 Semantic Segmentation Evaluation Index

Commonly used in semantic segmentation are Class Pixel Accuracy (CPA), Mean Pixel Accuracy (MPA), Intersection over Union IoU (IoU), and mean Intersection over Union (MIoU) as the evaluation index of the model. The definition of these indicators is related to the confusion matrix, as shown in Table 1, where the columns of the confusion matrix represent the number of pixels predicted for class i, the rows represent the number of pixels actually for class i, and the diagonal line Ci, where i represents the number of pixels correctly predicted for class I.

The meaning of CPA is: Among the predicted values of category *i*, the accuracy rate of pixels really belonging to category *i*; in other words, the model has many predicted categories of category *i*, among which there are right and wrong; CPA_i is the ratio between the number of pixels correctly predicted to belong to category *i* and the number of pixels predicted to belong to category *i* (summation of category *i* columns in Table 1). The calculation formula is as follows:

$$CPA_i = \frac{C_{i,i}}{\sum_{j=1}^{K} C_{j,i}} \tag{3}$$

Table 1: Confusion matrix

Confusion matrix		Predicted						Sum in rows
		Class 1	**Class 2**	**...**	**Class *i***	**...**	**Class *k***	
Ture	Class 1	$C_{1,1}$	$C_{1,2}$	...	$C_{1,i}$	...	$C_{1,k}$	$\sum_{j=1}^{K} C_{1,j}$
	Class 2	$C_{2,1}$	$C_{2,2}$	...	$C_{2,i}$	...	$C_{2,k}$	$\sum_{j=1}^{K} C_{2,j}$
	...	...	...	...	...	...	...	
	Class *i*	$C_{i,1}$	$C_{i,2}$	...	$C_{i,i}$	...	$C_{i,k}$	$\sum_{j=1}^{K} C_{i,j}$
	...	...	...	...	...	...	...	
	Class *k*	$C_{k,1}$	$C_{k,2}$	...	$C_{k,i}$	...	$C_{k,k}$	$\sum_{j=1}^{K} C_{k,j}$
Sum in columns		$\sum_{j=1}^{K} C_{j,1}$	$\sum_{j=1}^{K} C_{j,2}$		$\sum_{j=1}^{K} C_{j,i}$		$\sum_{j=1}^{K} C_{j,k}$	

MPA averages the sum of the CPA values of K categories. The calculation formula is as follows:

$$MPA = \frac{1}{K} \sum_{i=1}^{K} CPA_i \tag{4}$$

The meaning of IoU is: The ratio of the intersection and union of the prediction and reality of a certain category. For category *i*, the intersection of prediction and reality is the number of pixels correctly belonging to category *i*. The predicted and true sum is the sum of the number of pixels predicted to be class *i* (summation of class *i* columns in Table 1) and the number of pixels actually belonging to class *i* (summation of class *i* rows in Table 1) minus the number of pixels correctly predicted to belong to class *i*. The calculation formula is as follows:

$$IoU_i = \frac{C_{i,i}}{\sum_{j=1}^{K} C_{j,i} + \sum_{j=1}^{K} C_{i,j} - C_{i,i}} \tag{5}$$

MIoU averages the sum of the IoU values of K categories. The calculation formula is as follows:

$$MIoU = \frac{1}{K} \sum_{i=1}^{K} IoU_i \tag{6}$$

2.4 Image Processing

2.4.1 Object Extraction

During the process of image acquisition, it is common to encounter multiple types of defects within the same area. The presence of diverse objects in an image can significantly impact the quantitative calculation

result. Therefore, it is essential to transform semantic segmentation images containing multiple objects into images containing only one type of object.

The RGB mode, which consists of red (R), green (G), and blue (B) color components, is a widely utilized display mode for color images. In the Label link of this article, the color green represented spalling defects with an RGB channel value of [0, 128, 1], yellow represented exposed steel rebar defects with an RGB channel value of [128, 128, 0], and red represented crack defects with an RGB channel value of [128, 0, 0].

The region extraction method involves: Reading the R, G, and B channel values of each pixel in the MATLAB program; If the RGB channel value of a pixel matches the predefined green, yellow, and red values, the color of the pixel will be preserved. Otherwise, it will be set to black (i.e., background). For example, when extracting a green object, if the RGB channel value of a pixel is [0, 128, 1], its color will be preserved, and the color of the other pixels is set to black. The comparison between pre-extraction and post-extraction is illustrated in Fig. 3.

(**a**) Semantic segmentation image (**b**) Area of spalling (**c**) Area of exposed steel rebar

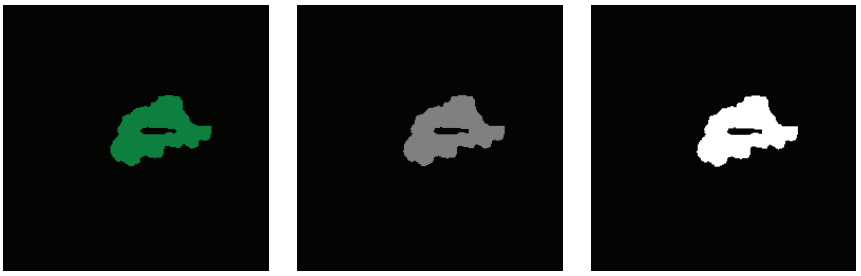
Figure 3: Comparison between pre-extraction and post-extraction

2.4.2 Grayscale and Binarization

In order to simplify the image information, the color images containing only a single object obtained from object extraction are processed by a grayscale and binarization technique. The process of graying an image involves converting a color image into a grayscale image, while the process of binarizing an image involves converting a grayscale image into a black and white image.

The grayscale and binarization technique applied to the image in this paper is outlined as follows:

1. Using the built-in function rgb2gray in the MATLAB program to calculate the gray value of each pixel in a color image. This is achieved by multiplying the channel values of the red, green, and blue by their respective weight coefficients and then summing them up to obtain the final gray value for each pixel.
2. Utilizing the maximum inter-class variance method to determine the boundary threshold value, pixels with a gray value less than or equal to the threshold are assigned 0 (black), while those with a greater gray value are assigned 255 (white), resulting in the conversion of grayscale images into binary images. The visual representation of this process is depicted in Fig. 4.



(**a**) Original image (**b**) Grayscale image (**c**) Binarization image

Figure 4: Grayscale and binarization images

2.5 Optimization of Target Areas

The target region predicted by semantic segmentation may contain errors such as segmentation discontinuity, small area voids, and noise points. As shown in Fig. 3 above, after the exposed steel rebar is segmented by semantic segmentation, a phenomenon of small holes within the spalling area may be observed. Directly calculating the area of the spalling region containing these small holes could lead to inaccuracies. Therefore, the binary morphology of the mathematical morphology method is employed to optimize the predicted target region of semantic segmentation.

The discipline of mathematical morphology, based on set theory, aims to optimize the geometry of objects in an image by utilizing structural elements to connect incoherent regions and remove incoherent ones. It involves four fundamental operations: dilation, erosion, open operation, and close operation.

The specific procedure of the expansion operation is as follows: Given an image A and a structuring element B of a certain shape (commonly rectangular, L-shaped, or cross-shaped), the center of B is moved across A from left to right, traversing all pixels of A from top to bottom. During this process, the maximum pixel value within the region covered by B is extracted and assigned to the corresponding pixel in the output image. In contrast, the erosion operation extracts the minimum pixel value within the region covered by B and assigns it to the corresponding pixel in the output image. The expansion operation is memoryless, meaning each operation is based solely on the original input image, and subsequent operations are not influenced by prior results, similar to the erosion operation. The expansion operation is illustrated in Fig. 5.

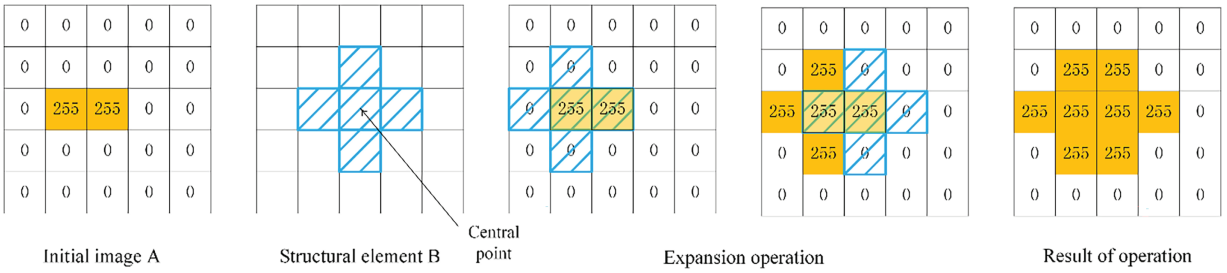


Figure 5: Dilation operation diagram

The specific operation of erosion is as follows: when the center of B traverses all the pixels of A, the minimum pixel value within the area covered by the structural element B is assigned to all pixels within that area during the erosion operation. The assignment is exclusively carried out on the original image. The erosion operation is illustrated in Fig. 6.

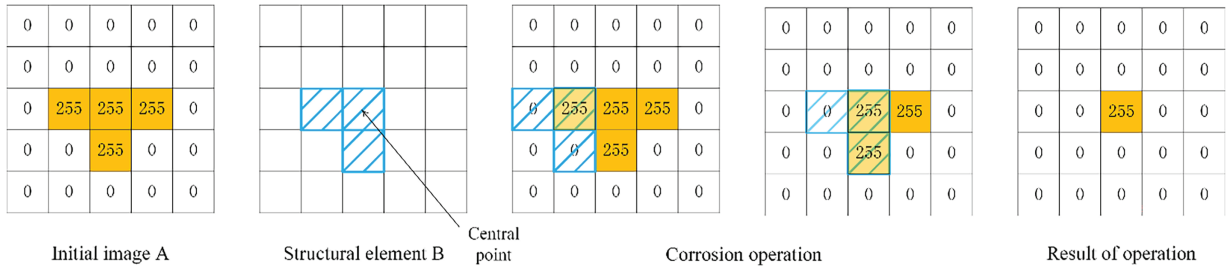


Figure 6: Erosion operation diagram

When optimizing target regions using the binary morphology, the shape and size of structural elements directly impact optimization results.

Regarding shape, Rectangular elements, being isotropic, uniformly affect expansion, erosion, opening, and closing operations to preserve original geometric contours. Cross-shaped elements primarily affect horizontal and vertical directions, making them suitable for linear targets. L-shaped elements are effective only in specific perpendicular directions.

Regarding size, small structural elements operate with limited amplitude, preserving fine details such as minute protrusions and irregular edges. They are suitable for handling noise and holes with fewer pixels. Large structural elements exert a substantial influence, enabling more efficient processing of noise and holes, such as concrete spalling areas corresponding to exposed reinforcing bars.

The opening operation consists of erosion operations followed by dilation operations, which are utilized to eliminate small, insignificant areas in the image. The closing operation consists of dilation operations followed by erosion operations, which are utilized to fill small holes and connect discontinuities in the target area.

2.6 Quantitative Calculation Method

After optimizing the target area, this paper adopts the following method to calculate the area or length of the target area: The area of the target area is the product of the area of a single pixel and the number of pixels in the target area; The length of the target area is the product of the minimum peripheral circle diameter of a single pixel point and the number of pixels in the target area. The number of pixels is obtained by the regionprops function, which measures the attributes of an image region in MATLAB.

The number of pixels in the diseased area can be calculated directly according to the above method. For cracks, because they have a certain width, it is not accurate to directly sum the length of all pixels. In this paper, the Hilditch algorithm is used in a MATLAB program to extract the skeleton of cracks and calculate the length of cracks. Then, the sum of the number of pixels from the crack skeleton to the contours at both ends of the crack is calculated, and the maximum calculated value is taken as the final crack width [23–25].

The extraction effect is illustrated in Fig. 7, while the calculation results of pixel points are depicted in Fig. 8.

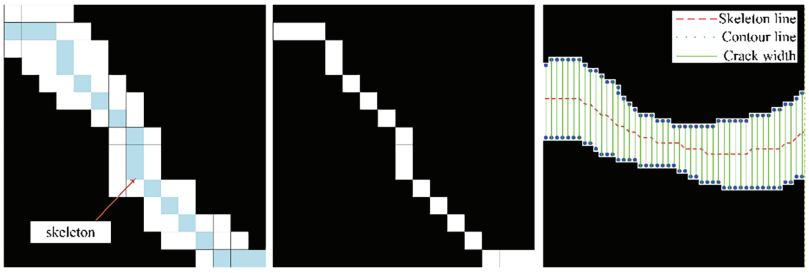


Figure 7: Skeleton extraction

The area of a single pixel was obtained using a square reference with a known actual area of 25 cm^2. The specific method is as follows:

(1) The same shooting equipment takes 200 images of reference A against different backgrounds, with the camera positioned 1 m away from A and parallel to its plane.

(2) The LabelMe polygon labeling software is utilized to outline reference A in the aforementioned 200 images and generate the corresponding label images, which are subsequently processed through grayscale and binarization. According to the aforementioned method, the pixel count of the reference object A region in the above 200 images is calculated using MATLAB. The error in calculating the pixel count of region A from 200 images is 0.017%, indicating that the outlines of reference A are relatively accurate and the error is insignificant.

(3) The actual area of reference object A is divided by the number of pixels in the calculated area of reference object A using the aforementioned method. The area of a single pixel is 0.0021 cm^2, and the diameter of the peripheral circle of a single pixel is 0.0648 cm, as determined by the shooting method employed in this paper. The process is shown in Fig. 9.

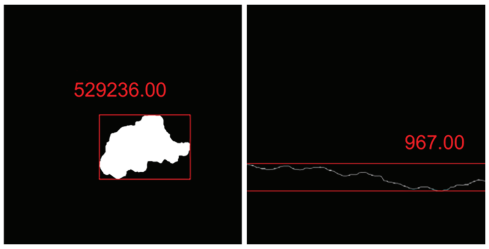


Figure 8: Pixel point calculation

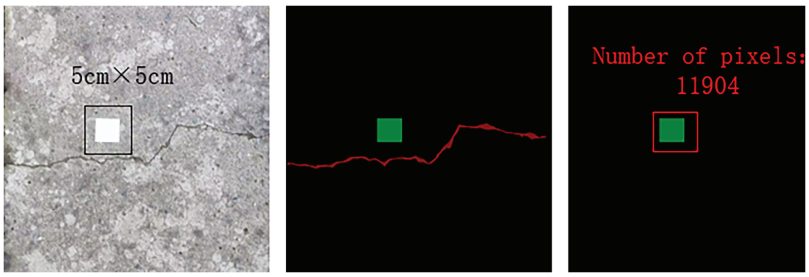


Figure 9: Single pixel size calculation

3 Semantic Segmentation Experiments

3.1 Datasets

Using Andor's iXon Ultra 897 EMCCD camera, concrete defect images were captured from 17 in-service reinforced concrete bridges in Xuchang City, Henan Province. A total of 3535 samples were obtained at a resolution of 512 × 512. The occurrence rates of three defects—cracks, spalling, and exposed reinforcement—were 28%, 32%, and 40%, respectively. Four data augmentation methods—translation, mirroring, random noise overlay, and brightness adjustment—were applied to fully utilize the limited sample size. Subsequently, the dataset was randomly divided into training, validation, and test sets at an 8:1:1 ratio.

LabelMe polygon labeling software is used to outline the concrete defects in the images, generating the corresponding label images, with each defect category being assigned a distinct color. The label process and label images are shown in Fig. 10.

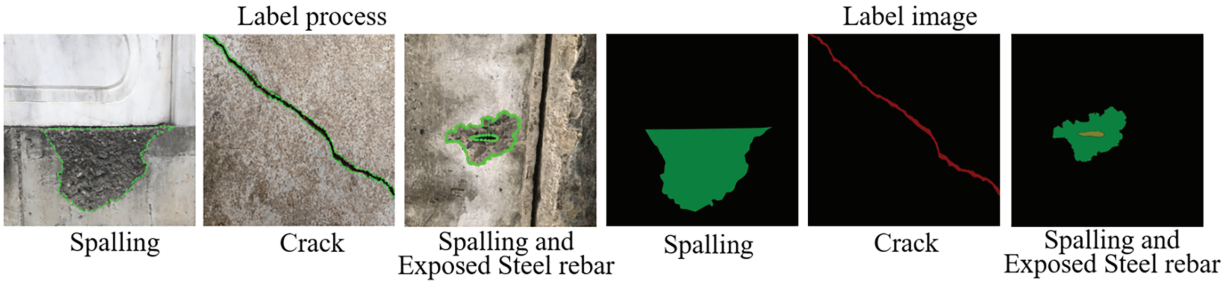


Figure 10: Label process and label images

3.2 Implementation Details

The model of semantic segmentation is developed using a Python program (version 3.7), leveraging the deep learning framework Pytorch 1.5.1, with Windows 10 operating system and an NVIDIA GeForce RTX 3060Ti graphics card. The model undergoes a total of 100 iterations of training, with a batch size of 1. During the initial 50 iterations, the backbone network remains frozen, and the transfer learning is conducted using the pre-trained weights obtained from the extensive dataset. During the last 50 iterations, all convolution layers participate in model training. The initial learning rate is set to 0.001 and dynamically adjusts during the training process. The momentum of 0.9 is employed, with the adaptive matrix estimation algorithm (Adam) serving as the optimizer.

3.3 Comparison of Results from U-Net, PSPNet, HRNet, and DeeplabV3+

Using the same dataset, the loss function (Dice loss) is applied to each of the five models: VGG16-U-Net, Resnet50-U-Net, PSPNet, HRNet, and DeeplabV3+. The semantic segmentation results for concrete defect images from the five models are presented in Table 2 under consistent experimental conditions and configurations. The evaluation metrics in the tables represent experimental results obtained after incorporating pre-weights.

As shown in Table 2, the VGG16-U-Net model exhibited the optimal performance: it reached a Mean Intersection over Union (MIOU) of 78.73% (surpassing PSPNet, DeepLabv3+, HRNet, and ResNet50-U-Net by 7.38%, 7.31%, 3.7%, and 2.1%, respectively) and a Mean Pixel Accuracy (MPA) of 88.95% (surpassing the same models by 2.97%, 3.04%, 0.6%, and 1.26%, respectively).

Table 2: Comparison table of experimental results of different semantic segmentation models

Models	Loss	IoU/%			CPA/%			MioU (%)	MPA (%)
		Exposed Steel Rebar	Spalling	Crack	Exposed Steel Rebar	Spalling	Crack		
PSPNet	Dice	70.59	87.33	56.15	85.57	93.59	78.81	71.35	85.98
DeeplabV3+	Dice	66.31	84.12	63.82	82.42	90.93	84.37	71.42	85.91
Resnet50-U-Net	Dice	71.39	84.47	74.33	86.01	92.13	85.94	76.63	87.69
HRNet	Dice	73.05	86.02	66.01	86.36	93.56	85.14	75.03	88.35
VGG16-U-Net	Dice	74.64	85.69	75.87	86.96	92.46	87.44	78.73	88.95

By comparing the semantic segmentation images output from each model, the accuracy of concrete defect segmentation can be visually assessed. Fig. 11 presents a comparison of segmentation results obtained from VGG16-U-Net, Resnet50-U-Net, PSPNet, and DeeplabV3+. It is evident that all four models exhibit higher accuracy in detecting spalling defects with minimal error compared to the labeled image. In terms of tiny cracks, VGG16-U-Net produces the closest result to the labeled image, while PSPNet shows more areas of missed detection, indicating its inferior feature extraction capability for cracks. As for exposed concrete reinforcement, both PSPNet and DeeplabV3+ demonstrate some missed detections, whereas VGG16-U-Net achieves the most accurate segmentation.

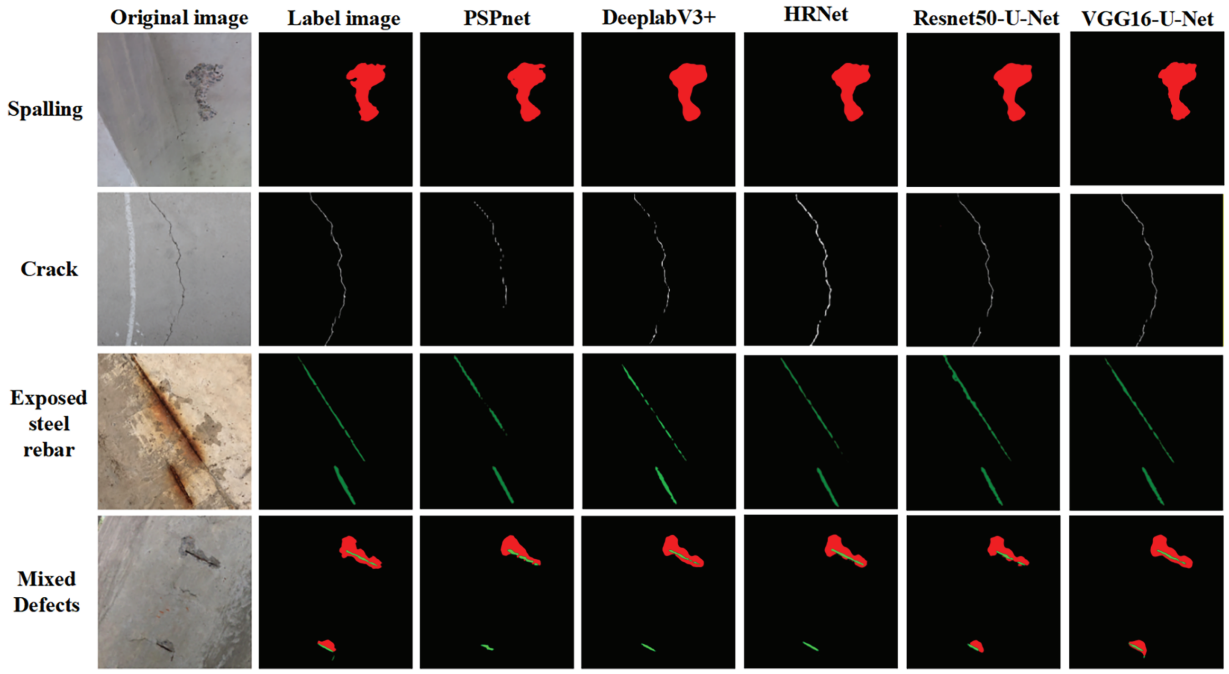


Figure 11: Comparison of the semantic segmentation results of five models

In the detection of mixed defects, DeepLabv3+, HRNet, and PSPNet exhibited partial omissions in identifying concrete spalling, while ResNet50-U-Net failed to accurately segment some exposed steel rebars. In contrast, VGG16-U-Net showed no obvious missing or false detections and could effectively identify small-sized steel rebars—indicating its superior suitability for the semantic segmentation of multi-class concrete defects.

3.4 Experimental Results

The VGG16-U-Net model was trained under the same environment with two different loss function configurations, respectively, namely Dice loss and a combination of Dice loss and Balanced Cross-Entropy loss. The predicted results, which had been incorporated with pre-weights on the same test set, are presented in Table 3, measured in percentage (%).

Table 3: Comparison of evaluation indexes for identifying concrete defects

Loss Function	CPA			MPA	IoU			MIoU
	Spalling	Crack	Exposed Steel Rebar		Spalling	Crack	Exposed Steel Rebar	
Dice	92.46	87.44	86.96	88.95	85.69	75.87	74.64	78.73
Dice+BCE	93.17	89.23	89.28	90.53	86.56	76.86	78.19	80.54

The identifying results demonstrate that the utilization of Dice loss combined with Balanced Cross-Entropy loss yields CPA of 93.17%, 89.23% and 89.28%, for VGG16-U-Net in detecting spalling, crack, and exposed steel rebar, respectively. The MPA is recorded at 90.53%, while the MIoU is recorded at 80.54%. Compared with the Dice loss function alone, utilizing a combination of Dice loss and Balanced Cross-Entropy loss resulted in an increase of 2.32%, 0.71%, and 1.79% in CPA, as well as an increase of 3.55%, 0.87%, and 0.99% in IoU. The results demonstrate that the combined loss function effectively directs VGG16-U-Net to prioritize the target region and accurately identify various types of concrete defects amidst complex backgrounds, especially in scenarios with unbalanced image samples.

The optimal image segmentation achieved by VGG16-U-Net is illustrated in Fig. 12. The white boxes indicate discontinuities and incorrectly predicted parts of the segmented image effect in the VGG16-U-Net model. The VGG16-U-Net model exhibits high segmentation accuracy and minimal prediction error when applied to targets with relatively large areas, such as spalling defects. However, for targets with relatively small areas like exposed steel rebar and cracks, there are some tiny discontinuities and prediction errors due to the complex conditions of sample imbalance, background clutter, and inconsistent light intensity. In the case of a subtle crack, the model segmentation shows a limited number of prediction errors. For exposed steel rebar, the segmentation results exhibit minor discontinuities and a few prediction errors. Nevertheless, the overall contours of the defects are predicted accurately. These findings indicate that the VGG16-U-Net achieves effective segmentation performance for images containing multi-object and multi-class concrete defects.

Classic semantic segmentation errors are illustrated in Fig. 13. In the task of identifying concrete spalling defects, the inherent complexity of concrete textures combined with surface contamination on structural components often makes it challenging for semantic segmentation models to accurately distinguish defect areas, leading to discontinuities in some samples. To address this issue, the segmentation results require optimization using the mathematical morphology method described in Section 2.5 of this paper.

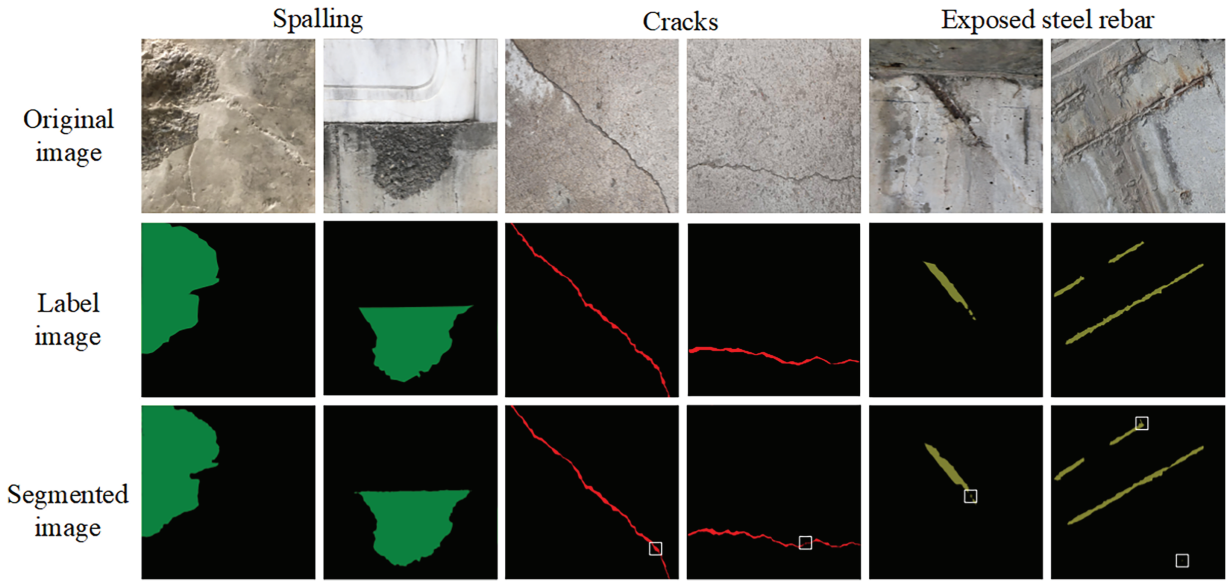


Figure 12: Comparison of label image and predicted image

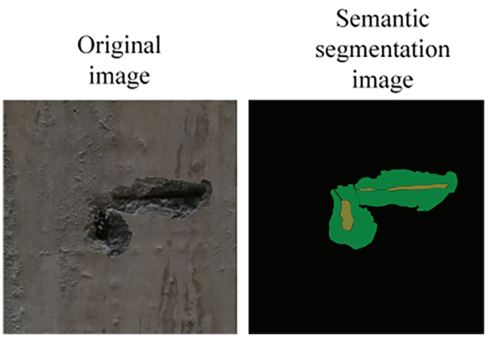


Figure 13: Examples of semantic segmentation errors

4 Optimization Results

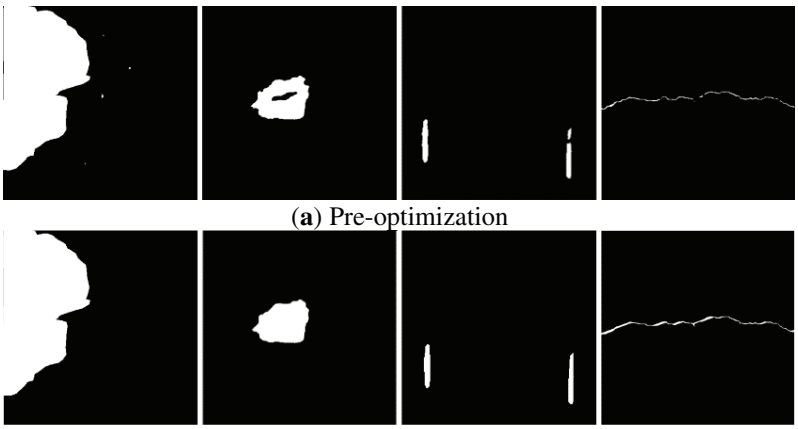
In order to enhance the accuracy of quantitative recognition, after target extraction, grayscale and binarization of the semantically segmented images are performed, region optimization of the defects in the binary images is further conducted according to the method described in Section 2.5.

1. Spalling and exposed steel rebar defects:

If there are small irrelevant areas surrounding the defects in the segmented image, the open operation is chosen to eliminate these insignificant regions. In cases where there are voids within the defects or discontinuities in the segmentation, the close operation is employed to fill these voids or connect the discontinuities. If both scenarios occur, the open operation takes precedence, followed by the close operation.

2. Crack defects:

Compared with spalling and exposed steel rebar, the shape of the crack is more complex, leading to discontinuous segmentation. Therefore, the closed operation is utilized to connect the discontinuities. Additionally, due to the slender nature of the crack compared to the exposed steel rebar, rectangular structural elements are selected for calculation. The comparison of pre- and post-regional optimization is illustrated in Fig. 14.



(**a**) Pre-optimization

(**b**) Post-optimization

Figure 14: Comparison of pre-optimization and post-optimization

5 Quantitative Calculation

5.1 Quantitative Calculation Results

In this paper, three types of concrete defect images were randomly selected from the dataset, and the area or length of the defects was calculated using the quantitative method described in Section 2.6. The calculation process is illustrated in Fig. 15.

Table 4 presents the area or length of the concrete defects. Due to the irregular shape of the actual defects, it is not possible to measure them directly. However, by enlarging the image to its maximum size and using a polygon annotation tool to mark along the contour of the defects, a label image with a contour very close to that of the actual defects can be generated. Therefore, the area or length of the defects calculated from this label image using the quantitative methods described in Section 2.4 can be considered as the actual value.

The calculated values of the three different types of concrete defects, as shown in Table 4, the calculation accuracy of exposed rebar area, length, spalling area, and crack length has been significantly improved after processing with the optimization algorithm proposed in this paper. After optimization, the absolute value of the error ranges from 0.08% to 0.23%, among which the optimization effect of spalling disease is the most obvious, and the absolute value of the calculation error is reduced from 5.1% to 0.21%, because the spalling area in the calculation example contains exposed rebar. After the exposed rebar area is extracted from the spalling area, the original area will produce voids, which causes a large error in the calculation of the spalling area. Mathematical morphology is used to bridge the void region and improve the calculation accuracy. The semantic model used in this paper is more accurate in the segmentation of crack defects, so the error before and after optimization does not change much. The above shows that the method proposed in this paper can accurately and intelligently identify and quantitatively calculate three types of common concrete defects, and the comparison of calculation results before and after optimization also shows that the mathematical morphology algorithm can effectively optimize the prediction errors generated by the semantic segmentation model.

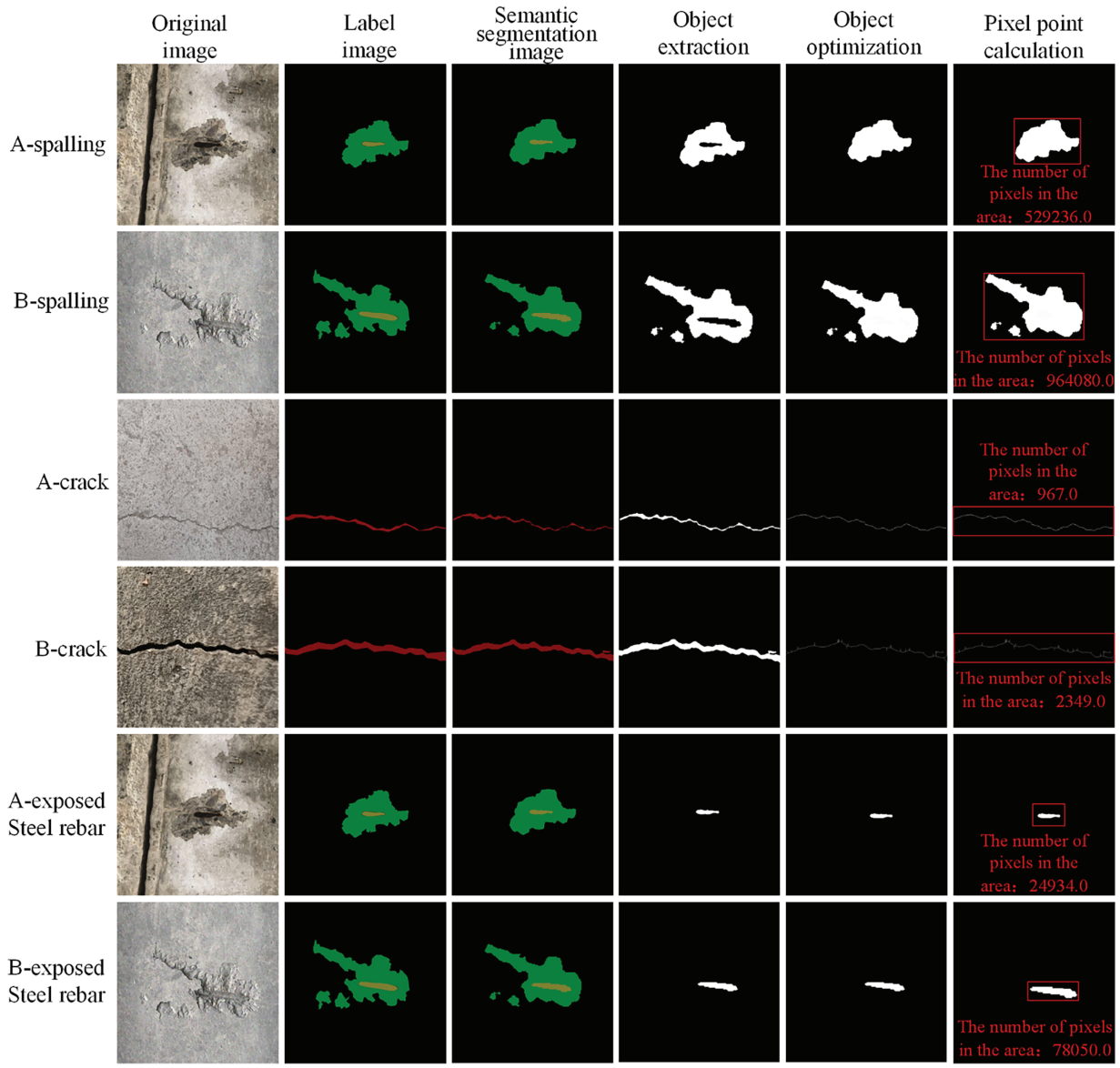


Figure 15: Quantitative calculation process of defect identification

5.2 Comparison with Transformer-Based and Attention Mechanisms Approaches

To further validate the effectiveness of the proposed method in terms of crack identification accuracy and quantification error, it was compared with two representative crack detection methods from recent years: the Feature Pyramid Transformer (FPT) network based on the Transformer architecture (Ding et al., 2024) [26] and the crack segmentation model based on an optimized DeepLabv3+ (Zhang et al., 2023) [27]. The comparison results are shown in Table 5.

Table 4: Quantitative calculation results of concrete defects

Types	Before optimization				After optimization			
	Actual value	Calculated values	Error	Absolute error proportion	Actual value	Calculated values	Error	Absolute error proportion
A-spalling area (cm^2)	1113.75	1057.35	−56.4	5.10	1113.75	1111.40	−2.35	0.21
B-spalling area (cm^2)	1547.21	1498.51	−48.70	3.15	1547.21	1543.7	−3.50	0.23
A-crack length (cm)	62.60	62.27	−0.33	0.50	62.60	62.66	0.06	0.10
B-crack length (cm)	76.23	75.97	−0.26	0.34	76.23	76.28	0.05	0.07
A-crack width (cm)	0.90	0.91	−0.01	1.1	0.90	0.91	−0.01	1.1
B-crack width (cm)	1.52	1.50	−0.02	1.32	1.52	1.50	−0.02	1.32
A-exposed steel rebar area (cm^2)	52.47	52.09	−0.38	0.72	52.47	52.36	−0.11	0.21
B-exposed steel rebar area (cm^2)	63.24	62.78	−0.46	0.73	63.24	63.12	−0.12	0.19
A-exposed steel rebar length (cm)	45.94	46.03	−0.09	0.20	45.94	45.98	−0.04	0.08
B-exposed steel rebar length (cm)	54.95	55.07	−0.12	0.22	54.95	54.91	−0.04	0.07

Table 5: Comparison of crack segmentation accuracy and quantification error among different methods

Models	Dataset	IoU/%	Absolute error proportion
DeeplabV3+(N-S)	4568	63.65	8.7
FPT	2300	62.15	4.78
VGG16-U-Net	990	76.86	1.32

In terms of Intersection over Union (IoU), the proposed method achieves 76.86% on the self-built dataset, surpassing the 62.15% of the FPT network and the 63.65% of DeepLabv3+(N-S), indicating superior accuracy in pixel-level segmentation of crack regions. Regarding dataset scale, this study completed training using only 990 images, whereas FPT and DeepLabv3+ relied on over 2300 and 4568 images, respectively, indicating that the proposed VGG16-U-Net, combined with mathematical morphology, achieves higher data utilization.

Regarding the absolute error ratio, our method achieves an absolute error of 1.32% in crack width calculation—significantly lower than FPT's 4.78% and DeepLabv3+'s 8.7%—validating the effectiveness of mathematical morphology post-processing in enhancing quantification accuracy. In summary, our approach maintains high segmentation precision while offering superior engineering practicality and data efficiency.

6 Conclusion

This paper proposes a concrete defect identification and quantitative calculation method based on VGG16-U-Net and mathematical morphology. The main conclusions are as follows:

(1) By utilizing VGG16 as the backbone network, the VGG16-U-Net model is capable of accurately classifying various types of concrete defects for every pixel, even in complex backgrounds. The utilization of Dice loss combined with Balanced Cross-Entropy loss yields CPA of 93.17%, 89.23% and 89.28%, for VGG16-U-Net in detecting spalling, crack, and exposed steel rebar, respectively. The MPA is recorded at 90.53%, while the MIoU is recorded at 80.54%. The combination loss function effectively directs VGG16-U-Net to prioritize the target area and accurately identify various types of concrete defects in complex backgrounds, especially in scenarios with unbalanced image samples. Runtime evaluation on an RTX 3060 Ti shows that the present model processes 512 × 512 images at 38 FPS with 2.1 GB GPU memory. If real-time defect detection is required in field applications, lightweight networks can be incorporated to reduce computational complexity and achieve efficient real-time defect detection.

(2) By employing mathematical morphology to optimize the segmentation errors in semantic segmentation images, the accuracy of the quantitative calculation of concrete defects is further enhanced.

(3) The aforementioned calculation results are derived from the calibration method and calculation method proposed in this paper. If the shooting method is changed, it is necessary to recalibrate the size of individual pixels. Future research can further explore the integration of unmanned aerial vehicles (UAVs) for image acquisition, thereby enhancing shooting stability and accuracy, and consequently improving the precision and reliability of defect identification.

(4) Although the proposed VGG16-U-Net combined with mathematical morphology methods can achieve reliable defect identification and quantification, future versions could incorporate attention mechanisms or Transformer modules to enhance the ability to capture subtle defect features such as micro-cracks and irregular spalling boundaries. These upgrades are expected to improve the segmentation accuracy in complex backgrounds further and enhance the framework's adaptability to diverse concrete defect scenarios, which is highly consistent with the development direction of AI-driven structural health monitoring technology.

Acknowledgement: The author sincerely thanks the College of Civil Engineering, Architecture, and Environment of Hubei University of Technology for its support and help. The author sincerely thanks the National Natural Science Foundation of China for its funding.

Funding Statement: The work described in this paper was supported by the National Natural Science Foundation of China (51708188).

Author Contributions: Conceptualization and design, writing—review & editing, supervision: Caiping Huang; experimental execution, data analysis, writing—original draft: Yulong Mei; resources, data collection, writing—review: Wangyuan Tian; algorithm programming, model training—review: Zihang Yu. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang C, Lai SX, Wang HP. Structural modal parameter recognition and related damage identification methods under environmental excitations: a review. Struct Durab Health Monit. 2025;19(1):25–54. doi:10.32604/sdhm.2024.053662.
2. Cha YJ, Chen JG, Büyüköztürk O. Output-only computer vision based damage detection using phase-based optical flow and unscented Kalman filters. Eng Struct. 2017;132:300–13. doi:10.1016/j.engstruct.2016.11.038.
3. Kim H, Ahn E, Shin M, Sim SH. Crack and noncrack classification from concrete surface images using machine learning. Struct Health Monit. 2019;18(3):725–38. doi:10.1177/1475921718768747.
4. Wu Y, Li S, Li J, Yu Y, Li J, Li Y. Deep learning in crack detection: a comprehensive scientometric review. J Infrastruct Intell Resil. 2025;4(3):100144. doi:10.1016/j.iintel.2025.100144.
5. Quirk L, Matos J, Murphy J, Pakrashi V. Visual inspection and bridge management. Struct Infrastruct Eng. 2018;14(3):320–32. doi:10.1080/15732479.2017.1352000.
6. Sun H, Lu D, Li X, Tan J, Zhao J, Hou D. Research on multi-apparent defects detection of concrete bridges based on YOLOR. Structures. 2024;65:106735. doi:10.1016/j.istruc.2024.106735.
7. Sun Z, Li J, Brilakis I, Besklubova S, Liang B, Liu Z. Visual-semantic alignment for automatic structural defect detection and diagnosis of prestressed concrete bridges. Autom Constr. 2025;180:106522. doi:10.1016/j.autcon.2025.106522.
8. Lei B, Wang N, Xu P, Song G. New crack detection method for bridge inspection using UAV incorporating image processing. J Aerosp Eng. 2018;31(5):04018058. doi:10.1061/(asce)as.1943-5525.0000879.
9. Zollini S, Alicandro M, Dominici D, Quaresima R, Giallonardo M. UAV photogrammetry for concrete bridge inspection using object-based image analysis (OBIA). Remote Sens. 2020;12(19):3180. doi:10.3390/rs12193180.
10. Gwon GH, Lee JH, Kim IH, Baek SC, Jung HJ. Image-to-image translation-based structural damage data augmentation for infrastructure inspection using unmanned aerial vehicle. Drones. 2023;7(11):666. doi:10.3390/drones7110666.
11. Feng D, Feng MQ. Computer vision for SHM of civil infrastructure: from dynamic response measurement to damage detection—a review. Eng Struct. 2018;156:105–17. doi:10.1016/j.engstruct.2017.11.018.
12. Pang D, Zhang H, Cao Y, Jiang Y. Lightweight and high-performance object detector and tracking model for concrete surface defect detection. Eng Appl Artif Intell. 2025;160:111968. doi:10.1016/j.engappai.2025.111968.
13. Arbaoui A, Ouahabi A, Jacques S, Hamiane M. Wavelet-based multiresolution analysis coupled with deep learning to efficiently monitor cracks in concrete. Frat Ed Integrità Strutturale. 2021;15(58):33–47. doi:10.3221/igf-esis.58.03.
14. Pozzer S, Ramos G, Rezazadeh Azar E, Osman A, El Refai A, López F, et al. Enhancing concrete defect segmentation using multimodal data and siamese neural networks. Autom Constr. 2024;166:105594. doi:10.1016/j.autcon.2024.105594.
15. Cha YJ, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Comput Aided Civ Infrastruct Eng. 2017;32(5):361–78. doi:10.1111/mice.12263.
16. Huang C, Zhai KK, Xie X, Tan J. Deep residual network training for reinforced concrete defects intelligent classifier. Eur J Environ Civ Eng. 2022;26(15):7540–52. doi:10.1080/19648189.2021.2003250.
17. Kruachottikul P, Cooharojananone N, Phanomchoeng G, Chavarnakul T, Kovitanggoon K, Trakulwaranont D. Deep learning-based visual defect-inspection system for reinforced concrete bridge substructure: a case of Thailand's department of highways. J Civ Struct Health Monit. 2021;11(4):949–65. doi:10.1007/s13349-021-00490-z.

18. Jin T, Zhang W, Chen C, Chen B, Zhuang Y, Zhang H. Deep-learning-and unmanned aerial vehicle-based structural crack detection in concrete. Buildings. 2023;13(12):3114. doi:10.3390/buildings13123114.
19. Chen C, Seo H, Jun CH, Zhao Y. Pavement crack detection and classification based on fusion feature of LBP and PCA with SVM. Int J Pavement Eng. 2022;23(9):3274–83. doi:10.1080/10298436.2021.1888092.
20. Huang H, Zhao S, Zhang D, Chen J. Deep learning-based instance segmentation of cracks from shield tunnel lining images. Struct Infrastruct Eng. 2022;18(2):183–96. doi:10.1080/15732479.2020.1838559.
21. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015; 2015 Oct 5–9; Munich, Germany. p. 234–41. doi:10.1007/978-3-319-24574-4_28.
22. Andreieva V, Shvai N. Generalization of cross-entropy loss function for image classification. Mohyla Math J. 2021;3:3–10. doi:10.18523/2617-7080320203-10.
23. Kim BC, Son BJ. Crack detection of concrete images using dilatation and crack detection algorithms. Appl Sci. 2023;13(16):9238. doi:10.3390/app13169238.
24. Ren Y, Huang J, Hong Z, Lu W, Yin J, Zou L, et al. Image-based concrete crack detection in tunnels using deep fully convolutional networks. Constr Build Mater. 2020;234:117367. doi:10.1016/j.conbuildmat.2019.117367.
25. Zhang J, Qian S, Tan C. Automated bridge surface crack detection and segmentation using computer vision-based deep learning model. Eng Appl Artif Intell. 2022;115:105225. doi:10.1016/j.engappai.2022.105225.
26. Ding W, Xia Z, Shu JP, Ye JL, Xiang YQ. Crack identification and detection of concrete bridge towers based on negative pressure adsorption wall-climbing robots and a transformer. China J Highw Transp. 2024;37(2):53–64. (In Chinese). doi:10.19721/j.cnki.1001-7372.2024.02.005.
27. Zhang XJ, Yuan JH, Yue XJ, Zhang WF. Crack segmentation and feature quantification of concrete beams based on optimized DeepLabv3+. Sci Technol Eng. 2023;23(9):3794–803. (In Chinese). doi:10.12404/j.issn.1671-1815.2023.23.09.03794.