International Journal of GEOMATE, Dec., 2024 Vol.27, Issue 124, pp.1-15
ISSN: 2186-2982 (P), 2186-2990 (O), Japan, DOI: https://doi.org/10.21660/2024.124.4696
Geotechnique, Construction Materials and Environment

CRACK DETECTION AND MEASUREMENT IN CONCRETE USING CONVOLUTION NEURAL NETWORK AND DBSCAN SEGMENTATION

Apisak Jutasiriwong[1], and *Wanchai Yodsudjai[2]

[1,2]Department of Civil Engineering, Kasetsart University, Thailand

*Corresponding Author, Received: 17 Aug. 2024, Revised: 27 Oct. 2024, Accepted: 29 Oct. 2024

ABSTRACT: Crack detection and measurement are essential for assessing the structural integrity of reinforced concrete (RC) structures, but challenges such as surface variability and class imbalance complicate accurate detection. This study introduces an approach integrating Convolutional Neural Networks (ConvNets), adaptive sliding windows, and DBSCAN-based semantic segmentation to address these challenges and enhance crack detection and quantification. The method was evaluated on various surface types, including painted masonry and concrete pavement, with a particular focus on overcoming class imbalance. To tackle this issue, the resampling (RS) technique was applied, achieving the best balance between precision and recall, with an F1 score of 0.836 during validation. The adaptive sliding window algorithm, optimized for lower magnification factors, further enhanced crack localization, improving IoU, recall, and precision. In semantic segmentation, the proposed method performed competitively on the DeepCrack dataset, achieving an IoU of 0.671, comparable to state-of-the-art models. Additionally, the measurement algorithm, which captures crack features such as length, width, and orientation, was tested on multiple surfaces. For painted masonry, it achieved a precision of 0.99, recall of 0.845, and IoU of 0.838, while for concrete pavement, it achieved a precision of 0.983, recall of 0.835, and IoU of 0.823. When applied to the DeepCrack dataset ground truth, it yielded a recall of 0.884, precision of 0.971, and IoU of 0.860. The results demonstrate the robustness and adaptability of this framework, offering an effective solution for automated crack detection and measurement across diverse surfaces.

Keywords: Crack detection, Convolutional Neural Networks, DBSCAN, Structural Health Monitoring, Crack feature quantification.

1. INTRODUCTION

Reinforced concrete (RC) is a widely employed construction material that combines concrete with steel rebars to enhance tensile strength. However, the integration of steel rebars introduces the risk of corrosion, accelerating the degradation of RC structures. The maintenance and rehabilitation of aging infrastructure, including RC structures, represent a substantial financial burden. In 1998, the United States allocated approximately 0.2% of its GDP, equivalent to 15.6 billion USD, for infrastructure maintenance and rehabilitation, excluding pipelines [1]. The estimated cost of maintaining aging infrastructure in 2021 exceeded 2.3 trillion USD. Monitoring and addressing surface cracks in RC structures early can prevent deterioration and reduce rehabilitation expenses.

Structural degradation is an inevitable process that begins early in a structure's lifecycle. Structural health monitoring (SHM) is vital for mitigating potential failures, as even minor structural damage can affect the overall system [2-4]. Traditionally, engineers inspect cracks visually using crack width gauges, capturing photographs for documentation [5]. By comparing crack widths over time, engineers can predict deterioration [6]. However, this manual process is time-consuming, labor-intensive, and prone to human error. Recent research has shifted towards object detection algorithms that automate crack classification and localization [7]. Automated robotic systems and image-based techniques are increasingly used to monitor inaccessible structures such as steel bridges [8]. UAV-mounted cameras allow for the non-contact inspection of hard-to-reach areas, expanding the applicability of SHM [9].

Artificial intelligence (AI), specifically Convolutional Neural Networks (ConvNets), has shown great promise in automating crack detection and quantification, significantly improving accuracy and reducing costs [10-13]. ConvNets extract image features through convolutional and pooling layers, followed by classification via fully connected layers. By automating crack detection, including width and orientation measurements, ConvNets reduce human labor and error. Recent advancements in non-contact optical techniques, such as Digital Image Correlation (DIC), have demonstrated enhanced accuracy in monitoring crack propagation and displacement fields in RC beams, making them a valuable complement to AI-driven methods like ConvNets [14]. While training ConvNets requires large datasets and computational resources, their application is invaluable in large-scale infrastructure projects, enabling faster and more precise structural assessments. Automated systems are essential for

maintaining vital infrastructure such as bridges and highways. Artificial Neural Networks have also been successfully used to predict road surface deterioration, demonstrating the applicability of machine learning algorithms in infrastructure monitoring. By integrating ConvNet models, this approach allows for accurate prediction of pavement conditions and provides a framework for further advancements in crack detection and structural health monitoring [15].

The adoption of ConvNets in crack detection has gained momentum in recent years, particularly due to the availability of pre-trained models that simplify deployment [16]. Sliding-window frameworks integrated with ConvNets are commonly used for localizing cracks within images, but they can be computationally intensive due to overlapping regions being processed multiple times [17]. The sliding window method involves scanning an image systematically, but its resource demands highlight the need for optimization. Despite this challenge, AI-driven crack detection has wide applications in monitoring transportation infrastructure, industrial plants, and hazardous environments such as dams and nuclear power facilities, where human inspection poses significant risks. Automating these processes enhances safety, ensures comprehensive monitoring, and facilitates preventive maintenance.

Other researchers have explored various methods for detecting and analyzing cracks in infrastructure using AI techniques. AI-based techniques for detecting infrastructure cracks have been extensively researched. Cha, et al. [18] adapted Faster R-CNN, a highly efficient object detection method developed by Girshick, et al. [19], Girshick [20], Redmon, et al. [21], and Ren, et al. [22] for classifying concrete and steel damages, achieving 90.9% average precision in concrete crack detection. Alternative methods, such as YOLOv3, have been utilized for real-time pavement crack detection [23]. Cheng, et al. [24] investigated a U-Net-based semantic segmentation approach, outperforming other ConvNet architectures in accuracy and computational efficiency for concrete crack detection and quantification.

In the realm of semantic segmentation, various approaches have been utilized for crack detection. Li, et al. [25] employed Crack Seed Clustering and Filtering with the DBSCAN algorithm to extract crack pixels from background pixels and noise, enhancing the efficiency of finding the Weighted Mean Intensity Path (WMIP) between crack seeds. Alipour, et al. [26] developed CrackPix, a pixel-level crack detection approach based on Fully Convolutional Network (FCN), which includes five architectures (FCN32s, FCN16s, FCN8s, FCN4s, and FCN2s) that use skip connections for multiscale feature fusion. CrackPix addresses class imbalance in crack detection using a weighted loss function and specifically targets fine-grained cracks by integrating additional FCN4s and FCN2s architectures for further refinement. Choi and Cha [27] introduced autoencoder ConvNets, another method commonly used for semantic segmentation. Tabernik, et al. [28] proposed SegDecNet++ model demonstrated strong performance across various datasets with different characteristics, achieving high IoU scores. On the CFD dataset [29], characterized by light urban road surfaces, it achieved 64.14% IoU, and on the DeepCrack dataset [30], which includes a mix of concrete walls and asphalt surfaces, it reached 69.78% IoU. The model faced challenges on the CrackTree200 dataset [31], which features very thin cracks on light pavements, with an IoU of 20.12%, and on the GAPs384 dataset [32], known for low-contrast cracks on dark asphalt, where it achieved 39.05% IoU. Despite these challenges, SegDecNet++ excelled on the Rissbilder dataset [33], which includes complex structures and varying lighting conditions, with a 67.95% IoU, and accurately identified non-defective images on the Non-crack dataset [34], achieving a near-perfect 99.53% IoU. This highlights the model's robustness and adaptability to diverse environments and surface types.

Image processing techniques offer improved precision compared to manual methods. Thresholding, edge detection, and wavelet transforms have been used to process images and identify crack features [35]. Object detection algorithms are increasingly central to vision-based monitoring, allowing for automated classification and localization of cracks. These algorithms address the challenges of processing large amounts of data, which is essential for practical, large-scale infrastructure monitoring.

Crack quantification is essential for assessing the condition of concrete structures and determining reinforcement needs. Yang, et al. [36] proposed a process for extracting crack morphological features, involving labeling operations, skeletonizing operations, and calculations of crack length, average width, and coverage ratio. These steps allow engineers to extract valuable information such as crack width and length. Miao and Srimahachota [37] introduced a modified Distance Transform Method (DTM) for crack quantification. This method involves binary image conversion, labeling, thinning, pruning, modified DTM, and calculations of thickness and length. This process aids engineers in quantifying the length and width of cracks in concrete structures. Kang, et al. [38] evaluated the performance of crack quantification by comparing predicted crack sizes with actual crack sizes, calculating the error rate, and comparing predicted and actual sizes in a graph. The error rates of different approaches, such as the Mask R-CNN + A* algorithm, were compared to determine which method performed better in defect quantization.

Recent advances in image processing and

machine learning provide an opportunity to develop novel approaches for effective crack analysis in RC structures. In this context, our research, described in the next chapter, aims to explore the application of perspective transform, adaptive sliding windows, DBSCAN, skeletonization, and principal component analysis to accurately measure crack features in images. By integrating these techniques, we aim to create a more efficient and precise system for crack detection and assessment, overcoming the limitations of existing methodologies. To address these challenges, our research integrates perspective transform, adaptive sliding windows, DBSCAN, skeletonization, and principal component analysis into a unified framework for automated crack detection and measurement. This comprehensive approach not only enhances detection accuracy and efficiency but also lays the groundwork for future advancements in real-time structural health monitoring and maintenance across diverse infrastructure types. As research and development in this field continue, such innovative approaches are likely to play an increasingly important role in the future of infrastructure monitoring and maintenance.

2. RESEARCH SIGNIFICANCE

This research presents a novel approach to crack detection and quantification in reinforced concrete structures using ConvNets, adaptive sliding windows, and DBSCAN-based semantic segmentation. The proposed method not only accurately detects cracks but also provides precise quantification of crack features such as length, width, and orientation. The crack measurement process is rigorously evaluated by replicating the detected crack features and comparing them against ground truth data. This validation ensures the reliability of the measurements. The approach offers practical benefits for infrastructure monitoring by enabling scalable, precise, and cost-effective analysis, essential for maintaining the integrity of aging structures.

3. METHODOLOGY

3.1 Dataset

In this research, we collected images of various surfaces, including painted masonry, terrazzo, and concrete pavement, from indoor and outdoor environments at Kasetsart University's Bangken Campus. The images were taken with smartphone cameras at a distance of 0.8 to 1.5 meters from the surface of interest, resulting in 925 images with a resolution of 1772 x 3362 pixels, referred to as original images (OMs). These OMs were then divided into 91 small images (SMs) of 256 x 256 pixels each using a sliding window method with 128-pixel overlap as depicted in Fig. 1.

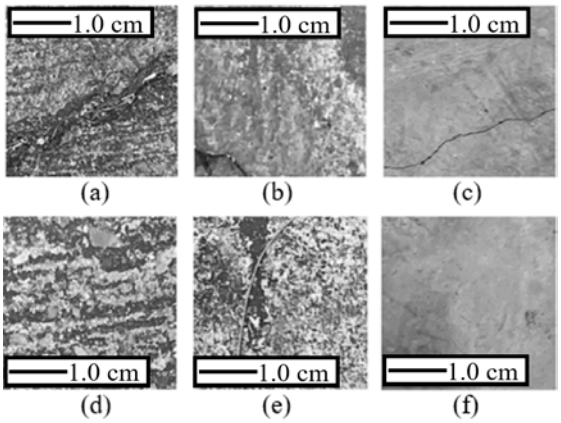

Fig. 1 Sample images: (a)-(c) Positive crack images (PMs); (d)-(f) Negative images (NMs).

This resulted in a total of 66,865 SMs. These SMs were manually labeled as either positive images (PMs) containing crack features or negative images (NMs) without crack features. The total number of NMs and PMs were 57,514 and 9,251, respectively, with the PMs making up only 13.83% of the total SMs, leading to a highly imbalanced dataset. The NMs and PMs were then divided into training, validation, and testing datasets in the ratio of 0.7:0.15:0.15, with the testing dataset being kept separate and not accessible to the researchers until the evaluation phase. The training dataset consists of 40,260 unique NMs and 6,475 unique PMs, while the validation dataset contains 8,627 unique NMs and 1,387 unique PMs.

3.2 Model

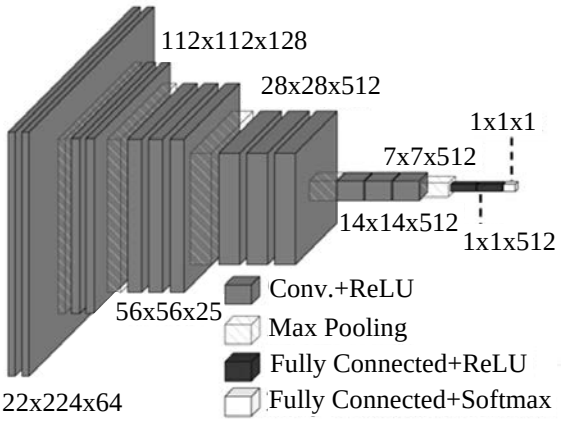

Fig. 2 Architecture of the proposed ConvNet model with layers for feature extraction and classification.

In this research, a Convolutional Neural Network (ConvNet) was utilized for feature extraction, specifically employing the VGG16 pretrained model as shown in Fig. 2. The VGG16 model was trained and evaluated on its ability to classify cracks in images using the validation set. The pretrained VGG16 model was adapted for the task with only the

fully connected layers set as trainable, while the bottom layers were not trained during the process. This approach allowed for efficient and effective binary classification of cracks in images, leveraging the robustness of the VGG16 architecture while minimizing the need for additional customizations.

3.3 Imbalance Data

The dataset used for training the ConvNet in this research is imbalanced, with a small number of positive images (PMs) compared to negative images (NMs). This can lead to poor performance of the ConvNet during the training process. To address this issue, instead of training with only ordinary datasets, several methods were applied in this research to mitigate the effect of imbalanced data. These methods include resampling, undersampling, and bias adjustment with class weighting.

Resampling involves replicating the PMs in the training dataset until the number of PMs is equal to the number of NMs. Undersampling involves randomly removing NMs from the training dataset until the number of NMs equals the number of PMs. Bias adjustment with class weighting involves setting the bias and weight of the output layer based on the proportion of PMs and NMs in the dataset.

3.4 Data Balancing Techniques

In addressing the notable imbalance in our dataset, which features a smaller number of positive images (PMs) compared to negative images (NMs), we implemented several methods. These techniques were aimed at mitigating the skewed distribution's impact on the ConvNet training process. The applied methods include resampling, undersampling, and bias adjustment with class weighting, each tailored to enhance model training under conditions of data imbalance.

Resampling (RS): This method involves duplicating the PMs in the training dataset until their count equals that of the NMs. By creating additional copies of the under-represented class, resampling balances the dataset, ensuring that the model is equally exposed to both classes during training. This exposure is crucial for preventing the model from becoming biased towards the more prevalent class.

Undersampling (US): In contrast, undersampling reduces the number of NMs to match the PMs count. This is achieved by randomly removing samples from the over-represented class. While this method leads to a balanced dataset, it does so by reducing the overall data size, which might limit the model's exposure to varied examples in the larger class.

Bias Adjustment with Class Weighting (BACW): This approach involves modifying the bias and weights in the model's output layer to reflect the proportion of PMs to NMs. By adjusting these parameters, the model is encouraged to pay more attention to the minority class, thereby compensating for the imbalance during the learning process.

Each of these models was trained using the same ConvNet architecture but with different dataset compositions or adjustments, providing a diverse perspective on how data balancing techniques can influence model performance.

3.5 Training and Evaluation

In this research, we employed a detailed and systematic approach for training and evaluating Convolutional Neural Networks (ConvNets) for the task of crack detection in reinforced concrete (RC) structures.

3.5.1 Technical Aspects of Training

Throughout the training process, the binary cross-entropy loss function was utilized to evaluate the model's performance. To optimize the ConvNets' weights at the end of each training batch, we used the Adaptive Moment Optimization (Adam) algorithm with cross entropy as the loss function. The model's predictions were categorized into four types: true positive (tp), false positive (fp), true negative (tn), and false negative (fn). Based on these categorizations, we calculated precision and recall using Eq. (1) and Eq. (2):

$$\text{Precision} = \frac{tp}{tp+fp} \quad (1)$$

$$\text{Recall} = \frac{tp}{tp+fp} \quad (2)$$

3.5.2 Monitoring Model Performance

We monitored two key metrics: the Area Under the Precision-Recall Curve (AUC-PR) and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), calculated at the end of each batch. The training was strategically terminated based on the AUC-PR values; if no improvement was observed within the last twenty epochs for the validation dataset, the training was ceased. This criterion ensured the timely halting of training, avoiding overfitting while capturing the model's optimal performance.

3.5.3 Validation Dataset as the Benchmark

Given the inherent differences in the training datasets due to various data balancing techniques, the validation dataset served as a consistent benchmark for comparing the performance of models. This homogeneity in the validation dataset was crucial for a fair and unbiased assessment of each model's ability.

3.5.4 Model Selection and Threshold Adjustment

The model that achieved the highest AUC-PR in the validation phase was selected for further evaluation. Subsequently, after assessing the best

model for each dataset, the model with the highest F1 score across all datasets was chosen for the final evaluation. The F1 score, which represents a balance between precision and recall, was calculated using the Eq. (3):

$$F1 = \frac{2\times Precision\times Recall}{Precision+Recall} \quad (3)$$

The optimal threshold for each model was adjusted across values from 0 to 1 to maximize the F1 score.

3.5.5 Final Model Application and Comparative Analysis

The model with the highest AUC-PR during validation was selected for evaluation. The F1 score, balancing precision and recall, was used to choose the optimal model. Threshold adjustments were made to maximize this score. The selected model was then applied to the testing dataset, comparing results to the validation dataset to assess generalization capability and practical performance.

3.6 Dataset for Measuring Crack Features

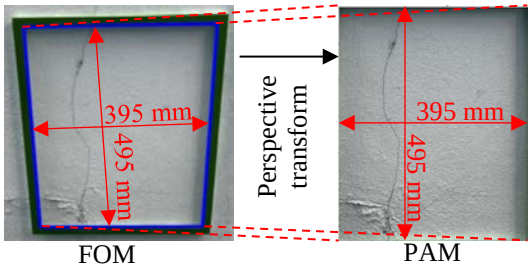

Fig. 3 Example of perspective transforming process from a FOM to a PAM.

In order to measure the length, average width, and orientation of crack features in an image, the author collected a separate dataset of images, referred to as framed original images (FOMs). These images were taken with a green frame of inner width and inner height of 395.0 mm and 495.0 mm placed on the surface of the wall as shown in Fig. 3. The perspective adjusted images (PAMs) of these FOMs were created using the perspective transform feature in the OpenCV library. The reference width and height are the dimensions of the green frame. These PAMs were then used to measure the desired crack features.

3.7 Detecting Crack Region with ConvNet and Adaptive Sliding Windows

To search for crack regions in a PAM, an adaptive sliding window method can be employed with the ConvNet. The proposed adaptive sliding window methodology for detecting cracks in concrete structures using the ConvNet as shown in Fig. 4 can be summarized in the following steps: (1) define the sliding window size as 256x256 pixels and establish an overlap of 128 pixels for horizontal and vertical traversals; (2) perform an initial scan of the PAM employing the sliding window and the ConvNet; (3) filter the windows based on a predefined threshold value (e.g., 0.8) to designate active scanning regions; (4) expand the borders of the identified active scanning regions by a predetermined number of pixels (e.g., 256 pixels); (5) enlarge the original image by a specific magnification factor (e.g., 2 times); (6) traverse the newly defined active scanning regions with the sliding window on the enlarged image, collecting output values from the ConvNet; (7) repeatedly execute the filtering, border expansion, and sliding window application process, while comparing the quantity of active windows in the current iteration to those in the previous iteration.

The iteration process terminates if the number of active windows in the current iteration, prior to border extension, does not meet or exceed the number of active windows in the preceding iteration. In such cases, the results from the previous iteration are employed for crack detection. In conjunction with the stopping criterion, this adaptive sliding window strategy promises enhanced efficiency and accuracy in identifying cracks in RC structures, facilitating more effective maintenance and rehabilitation efforts within the industry. The quantitative values for the border extension, magnification factor, and threshold value can be fine-tuned to optimize the algorithm's performance based on the specific characteristics of the dataset and the targeted application.

The resulting image containing only the crack regions at the original resolution is called a region perspective adjusted image (RPAM). This RPAM can then be further processed and analyzed to measure the cracks' length, average width, and orientation. To evaluate the performance of detecting crack regions with ConvNet and adaptive sliding windows, we use precision, recall, and Intersection over Union (IoU). IoU is calculated as the area of overlap between the detected crack regions and the ground truth regions divided by the area of their union. Each step will be evaluated separately, using the input from the previous step as the ground truth, considering errors or missing detections from the previous process.

3.8 Semantic Segmentation

The proposed methodology for crack detection in reinforced concrete structures using a combination of adaptive thresholding and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) for semantic segmentation, as shown in Fig. 5, can be described in further detail as follows:

1) Adaptive thresholding is applied to each window Region Perspective Adjusted Image (wRPAM) individually. The wRPAMs are first

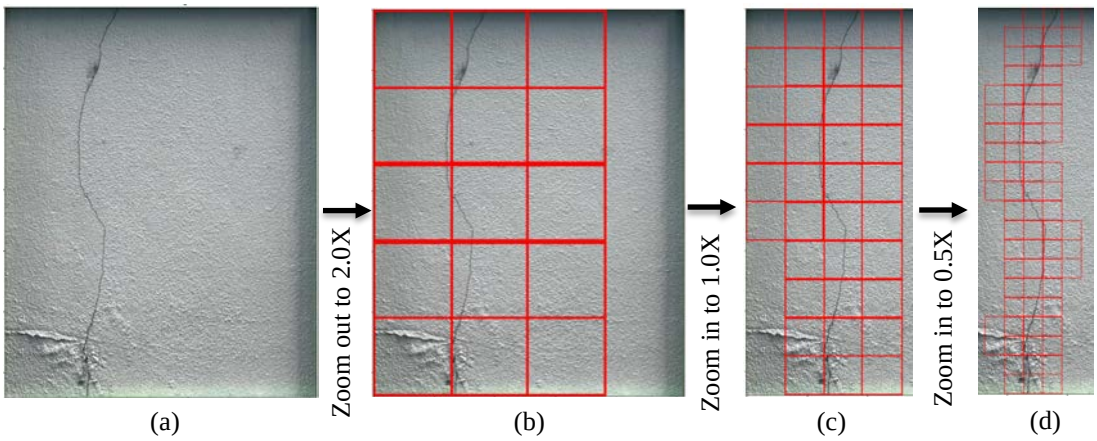

Fig. 4 Crack detection using adaptive sliding windows at varying image resolutions. The window size remains constant at 256x256, while the image resolution changes iteratively to refine positive regions. (a) Perspective adjusted image (PAM). (b) Image zoomed out to 2x for initial detection. (c) Image zoomed in to 1.0x, refining detection on previous positive areas. (d) Image further zoomed in to 0.5x, continuing detection on refined regions.

converted to grayscale images to reduce complexity and focus on the intensity variations. By processing each wRPAM separately, the adaptive thresholding accounts for local changes in lighting conditions, contrast, and noise that can occur across different sections of the image. This localized approach results in more accurate segmentation, reduces the impact of noise, and enhances computational efficiency by processing smaller portions of the image. Additionally, this stepwise processing allows for parallelization opportunities, potentially speeding up the overall process.

2) After adaptive thresholding, the binary images obtained from each wRPAM are combined into a single, larger image. This step is essential for preserving the spatial relationships between the segmented regions and maintaining the overall context of the original image. Combining the wRPAMs ensures that the final binary image represents the complete structure, including any cracks and background areas.

3) Shapes and sizes, representing crack features, while also being robust to noise. Using DBSCAN, the proposed method can accurately distinguish between crack features (assigned a value of 1) and background (assigned a value of 0) in the binary image. The output of this process, a semantically segmented image representing crack features and background, will be referred to as a semantic segmented image (SSM) for use in subsequent analyses.

This combination of adaptive thresholding and DBSCAN offers a comprehensive and efficient approach for detecting and segmenting cracks in reinforced concrete structures. By accounting for local variations in image characteristics and reducing the impact of noise, this method improves the accuracy and reliability of crack detection in such structures. The performance of the semantic segmentation will be measured using precision, recall, IoU, with the detected crack regions from the ConvNet and adaptive sliding windows serving as the ground truth. This approach will account for any errors or missing detections from the previous step, ensuring a comprehensive evaluation.

3.9 Crack Measurement

Skeletonization is a morphological operation applied to the SSM to generate one-pixel wide skeletons, preserving the structure and connectivity of the original crack features while simplifying the representation. Intersection points, where multiple crack branches converge, are detected using a convolution operation with a 3x3 kernel that identifies pixels with more than two neighboring pixels as shown in Fig. 6. After dividing skeleton lines into branches by removing intersections, points within each branch are sorted from one end to the other. The length, width, and orientation of each branch are analyzed using the proposed framework as illustrated in Fig. 7.

Length computation is summarized in the following steps: (1) Iterate through the divided skeleton branches, starting from one end of a branch and progressing to the other end; (2) Analyze adjacent skeleton pixels and increment the length count based on their relative orientation: 1.00 pixel for vertically or horizontally oriented points, and 1.41 pixels for diagonally oriented points (45 degrees); (3) Sum the length counts to determine the total length of the branch; (4) Repeat the process for all skeleton branches in the image.

Orientation computation is summarized in the following steps: (1) select a skeleton branch and start

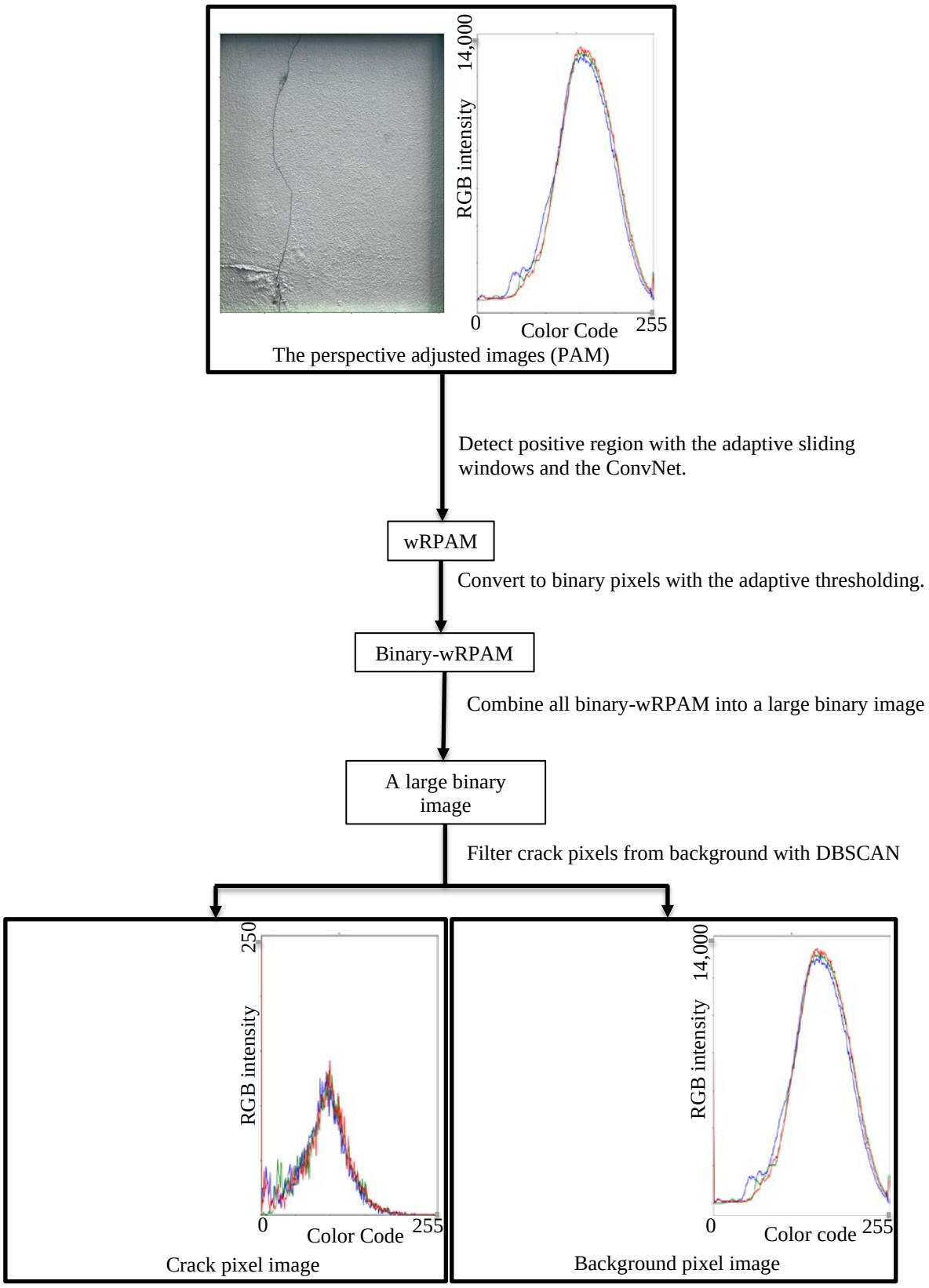

Fig. 5 Semantic segmentation process combining adaptive thresholding and DBSCAN: The Perspective Adjusted Image (PAM) is analyzed with ConvNet and adaptive sliding windows. Crack regions are converted to binary images through thresholding, then refined using DBSCAN to segment crack pixels from the background. The final segmented cracks are validated by comparing RGB intensity distributions.

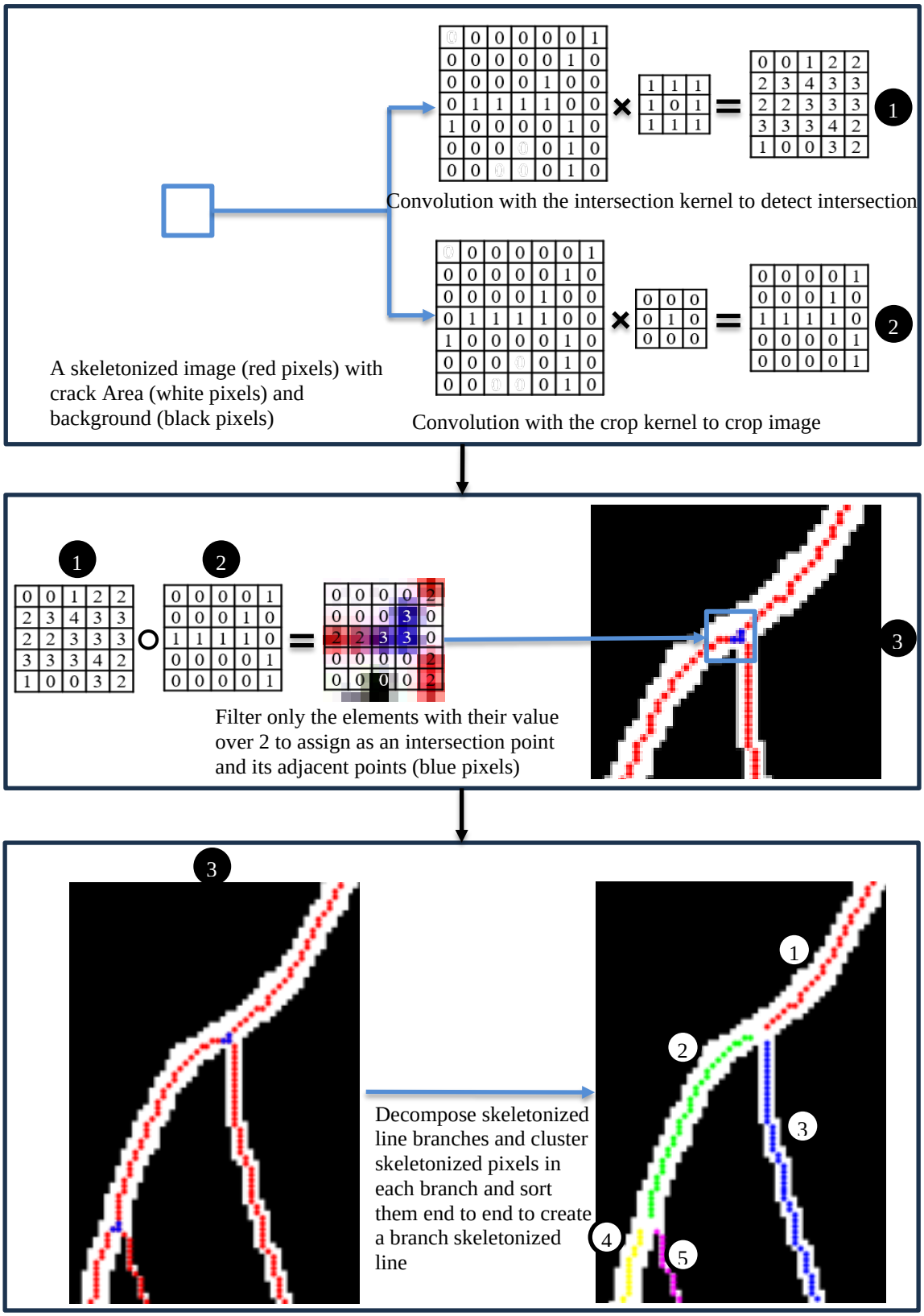

Fig. 6 Skeletonization and decomposition of crack features: (1) Convolution with intersection and crop kernels to detect and segment intersections; (2) Filtering intersection points with a threshold to isolate intersection and adjacent pixels; (3) Decomposing skeletonized crack branches into clusters for sorting and reconstruction. The process results in distinct branches marked by different colors for accurate length, orientation, and width analysis.

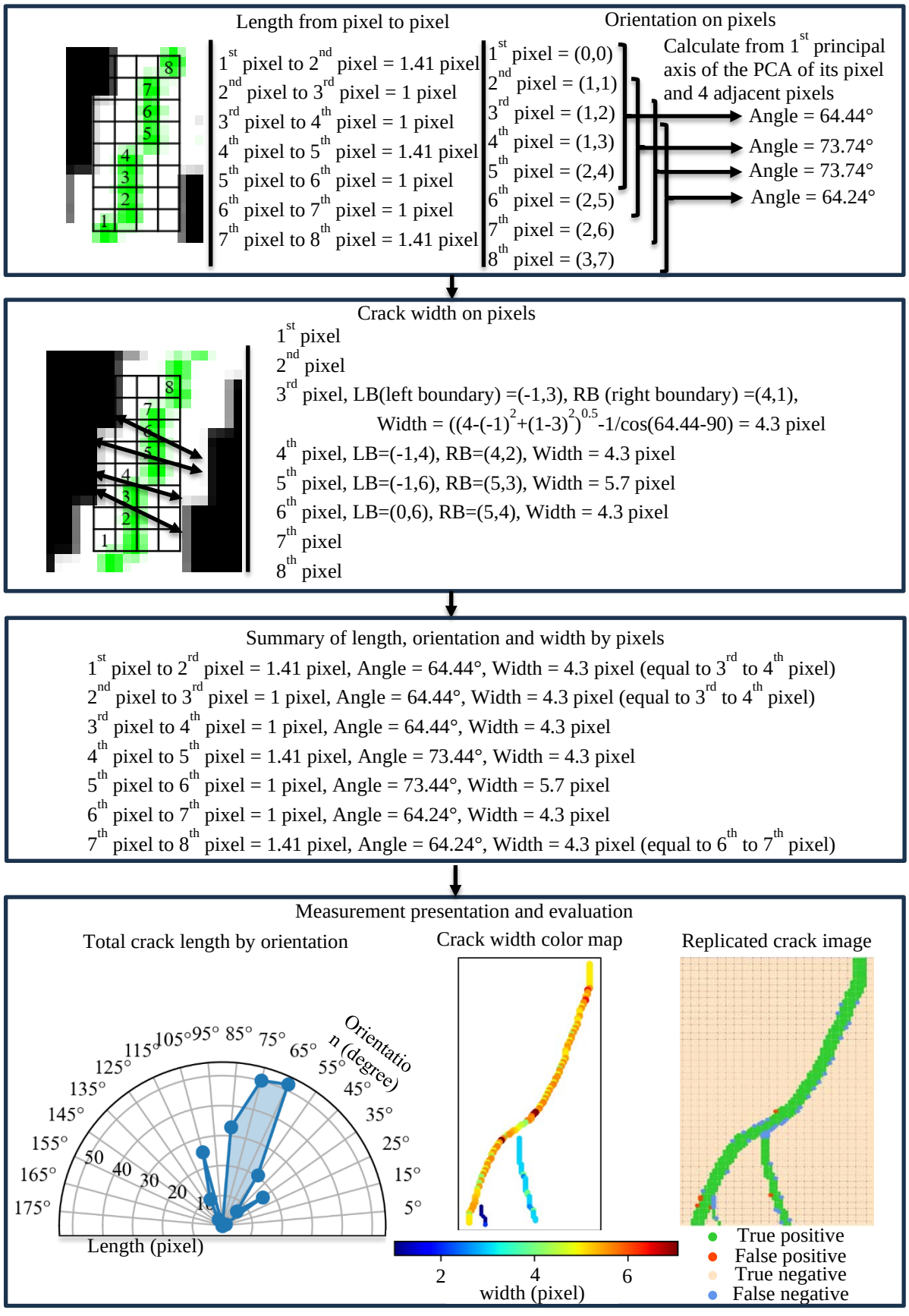

Fig. 7 shows the algorithm to measure length, orientation, and width of cracks, the crack image replicated from the measurement, and the performance of the proposed algorithm.

from one end; (2) define a segment consisting of the starting point and the next five consecutive points along the branch; (3) perform Principal Component Analysis (PCA) on the segment's coordinates to identify the principal axes of the data; (4) calculate the orientation of the segment as the angle formed by the first principal axis of the PCA result; (5) shift the starting point to the next position along the branch and repeat steps 2-4 until the other end of the branch is reached; (6) apply the orientation computation process to all skeleton branches in the image.

Width computation is summarized in the following steps: (1) choose a skeleton branch and start from one end; (2) determine the local orientation of the skeleton line at the starting point using the computed orientation; (3) project a line perpendicular to the local orientation from the starting point, extending to both sides of the crack feature until reaching the boundary (transition from 1-color to 0-color in the SSM); (4) calculate the width of the crack at the starting point as the length of the projected line; (5) shift the starting point to the next position along the branch and repeat steps 2-4 until the other end of the branch is reached; (6) compute the average width for the entire skeleton branch by averaging the width measurements; (7) perform the width computation process for all skeleton branches in the image. By following this systematic process, the framework accurately computes the length, orientation, and width of crack features in the SSM.

To assess the performance of the proposed measurement framework, we suggest synthesizing crack features based on the computed measurements and comparing them to the original SSM. This comparison will effectively evaluate the accuracy and reliability of the framework. The evaluation process consists of the following steps: (1) use the computed measurements (coordinates and width) for each point in each branch to synthesize crack features in a blank image with the same dimensions as the original SSM; (2) compare the synthesized crack image to the original SSM on a pixel-by-pixel basis; (3) calculate recall (sensitivity) and precision (positive predictive value) to determine the framework's performance;(4) analyze the results to identify potential areas for improvement or optimization.

The performance of the crack measurement step will be evaluated using precision, recall, and IoU, with the semantically segmented image (SSM) serving as the ground truth. This ensures that the evaluation considers any inaccuracies or omissions from the semantic segmentation process.

4. RESULT AND DISCUSSION

4.1 Convnet Performance on Imbalanced Data

The ConvNet was trained and evaluated for its ability to detect and quantify crack features in images. The model was trained using the ordinary datasets, and the datasets that were balanced using three different resampling methods: class weight adjustment, oversampling and undersampling. The resulting models were evaluated on a validation dataset consisting of 8,627 negative examples and 1,387 positive examples, as well as a testing dataset with a similar composition.

For the validation datasets, the AUC-PR performance varied more significantly, with the highest value of 0.905 for the model trained with original dataset (OD) at epoch 7 and the lowest value of 0.88 for US at epoch 3 as shown in Fig. 8. The BACW and RS demonstrated AUC-PR values of 0.9 and 0.896, reached at epoch 9 and epoch 3, respectively. The varying AUC-PR performance suggests that the choice of dataset balancing technique impacts the precision-recall tradeoff in the crack detection task, with RS and BACW techniques potentially yielding better results than US techniques.The model with the highest F1 score in the validation dataset was selected for further analysis. By adjusting the threshold range from 0 to 1. the RS as shown in Fig. 9 demonstrated the highest F1 score of 0.836 at a threshold of 0.8. The OD and BACW yielded comparable F1 scores of 0.832 and 0.828, respectively, at thresholds of 0.287 and 0.648. The US had the lowest F1 score of 0.809 at a threshold of 0.774.

These findings suggest that the RS, which employs resampling to balance the training dataset, performs best in terms of balancing precision and recall, achieving the highest F1 score. The OD and BACW exhibit similar performance, with slightly lower F1 scores. The US has the lowest F1 score, indicating that it may not be the most effective method for addressing dataset imbalance in this particular case.

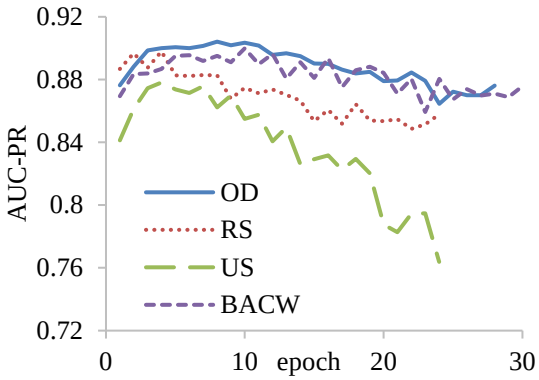

Fig. 8 Evolution of AUC-PR across training epochs for ConvNet models with different data balancing techniques: OD (Original Dataset), RS (Resampling), US (Undersampling), and BACW (Bias Adjustment with Class Weighting).

Furthermore, the epoch numbers at which the highest AUC-PR values were reached indicate that

the RS and US datasets reached optimal performance more quickly (at epoch 3) compared to the OD and BACW (at epochs 7 and 6, respectively). This suggests that resampling and undersampling techniques may lead to faster convergence during training, potentially reducing the overall training time.

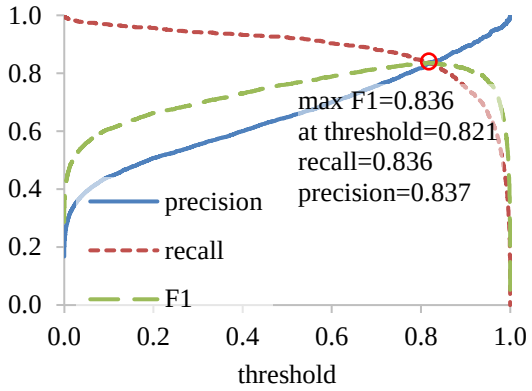

Fig. 9 Performance analysis of the RS model on the validation dataset.

Overall, these results demonstrate that the resampling and the bias adjustment with class weighting techniques can be effective in addressing imbalanced datasets for crack detection tasks, with resampling yielding the best performance in terms of F1 score. The analysis of epoch numbers at which the highest AUC-PR values were reached, along with the outlier prediction analysis, provide insights into potential improvements and optimizations for future work.

The selected RS-Model at threshold of 0.821 yielded comparable performance on the testing dataset. With the F1 score of 0.822, recall of 0.837 and precision of 0.807, the RS-Model is robust to use in real world applications.

4.2. Performance of Crack Localization

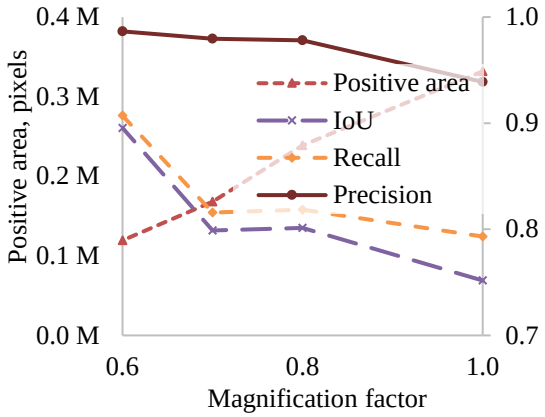

Fig. 10 Positive area, recall, precision, and IoU at various magnification factors during crack localization.

The adaptive sliding window approach was evaluated for crack detection across images with varying surface types and scales. As shown in Fig. 10, the analysis demonstrates that the choice of magnification factor significantly influences the method's performance. At a magnification factor of 1.0, the Intersection over Union (IoU) was 0.752, recall was 0.793, and precision was 0.939. As the magnification factor decreased to 0.8, 0.7, and 0.6, the positive area reduced while IoU, recall, and precision improved, reaching peak performance (IoU = 0.896, recall = 0.907, precision = 0.987) at a magnification factor of 0.6.

These findings suggest that smaller windows are more effective at accurately localizing cracks, as they better capture the fine details of crack features. However, reducing the positive area also decreases the workload for subsequent semantic segmentation, potentially enhancing efficiency. The adaptive sliding window technique exhibited robust performance across different surfaces, demonstrating flexibility in handling varying textures and crack characteristics.

When categorized by surface type, performance metrics indicated high precision and recall across both painted masonry and concrete pavement. For painted masonry, precision, recall, and IoU were 0.982, 0.937, and 0.921, respectively. For concrete pavement, the values were 0.923, 0.909, and 0.845, respectively. The results highlight the method's adaptability across different structural materials, making it suitable for a wide range of real-world applications.

4.3 Semantic Segmentation

The semantic segmentation step was evaluated using precision, recall, and IoU, with results analyzed separately for painted masonry and concrete pavement surfaces. The outputs from crack localization served as the ground truth.

For painted masonry, the segmentation achieved a precision of 0.933, recall of 0.940, and IoU of 0.881, indicating high accuracy with minimal false positives and substantial overlap between predicted and actual crack regions. For concrete pavement, the performance was slightly lower, with a precision of 0.933, recall of 0.882, and IoU of 0.829. The lower recall and IoU suggest some crack features may be missed in more complex surfaces like concrete pavement, though the method remains highly reliable overall.

Compared to previous studies, such as those employing DeepCrack and SCCDNet-D32 models, which achieved IoU scores of 0.636 and 0.655, respectively, our method's performance of 0.671 demonstrates significant improvements in crack detection accuracy, particularly in diverse surface types. Additionally, unlike models such as SegDecNet++ which require extensive dataset-

specific training, our approach achieves competitive results with a more generalized framework. This generalization reduces the need for extensive preprocessing and retraining, making it more suitable for real-world applications. The research fills a gap in existing methodologies by offering a flexible solution that balances high accuracy with practical applicability across different crack features and surface types. Future applications of this study include real-time structural health monitoring systems for critical infrastructure, where the adaptability of the method to diverse conditions will be crucial.

To further assess the method's robustness, it was applied to the DeepCrack dataset for comparison with other state-of-the-art models. Our approach achieved an IoU of 0.671, ranking third behind SegDecNet++ (0.698) and CrackFormer (0.679). The IoU scores for other methods on the same dataset include SCCDNet-D32 (0.655) and DeepCrack (0.636) [28]. Notably, our method did not require extensive dataset-specific training or labeling, which is a key advantage over more specialized models. Despite its competitive performance, the proposed semantic segmentation approach offers greater practicality for real-world applications due to its reduced dependency on customized training data.

4.4 Crack Feature Measurement

The proposed crack measurement framework was evaluated by comparing the synthesized crack features against the ground truth in Fig. 11 for two surface types: painted masonry and concrete pavement. In both cases, the replicated crack images closely resemble the original cracks in terms of length, width, and orientation. For painted masonry, the measurements show high precision (0.990), indicating that nearly all detected features are true positives, with a recall of 0.845, demonstrating good sensitivity in capturing actual cracks. The Intersection over Union (IoU) for painted masonry is 0.838, reflecting substantial overlap between the replicated and original cracks.

For concrete pavement, the framework also performed well, with a precision of 0.983, recall of 0.835, and IoU of 0.823. Despite the complex texture and irregularities inherent in concrete surfaces, the method effectively captures the essential crack features with minimal false positives and solid overall alignment between the synthesized and ground truth crack regions. The results indicate that the proposed measurement algorithm is reliable and performs consistently across different surface types.

In addition to the main experiments, the measurement algorithm was tested on the ground truth of the DeepCrack dataset to evaluate its pure performance, independent of errors introduced in previous processes. By directly applying the measurement algorithm to the ground truth images and comparing the replicated crack features against the original dataset, the algorithm achieved a recall of 0.884, precision of 0.971, and IoU of 0.860. These results highlight the effectiveness of the measurement approach when no prior process errors are present, demonstrating the algorithm's strong capability in accurately quantifying crack features in an ideal scenario.

Overall, the high precision and IoU across all evaluations confirm that the measurement algorithm is robust, with reliable performance that generalizes well across different datasets and surface types. The framework not only captures crack dimensions and orientations with high accuracy but also maintains strong consistency even when applied to diverse conditions and textures. This consistency underscores the framework's potential for practical applications in structural health monitoring and crack analysis.

5. CONCLUSION AND RECOMMENDATIONS

This study presents an integrated methodology for crack detection, segmentation, and measurement in reinforced concrete structures using Convolutional Neural Networks (ConvNets), adaptive sliding windows, and semantic segmentation with DBSCAN. The proposed approach effectively addresses challenges related to crack detection and analysis across diverse surface types, including painted masonry and concrete pavement. The results demonstrate that the combination of adaptive sliding windows and semantic segmentation yields high precision, recall, and Intersection over Union (IoU) values, reflecting the method's accuracy and robustness.

In the evaluation of different data balancing techniques, the resampling (RS) approach outperformed other methods in handling dataset imbalance, achieving the highest F1 score and rapid convergence during training. The adaptive sliding window technique was shown to be flexible and effective in crack localization, particularly at lower magnification factors, which improved performance metrics such as IoU and recall. The semantic segmentation process, validated using both custom datasets and the DeepCrack benchmark, demonstrated competitive performance against state-of-the-art models while requiring less extensive dataset-specific training.

The crack measurement framework successfully quantified critical crack features such as length, width, and orientation, with high precision and recall across different surface types. The evaluation of the measurement algorithm on the DeepCrack dataset without prior process errors showed the algorithm's pure performance, yielding a recall of 0.884, precision of 0.971, and IoU of 0.860, confirming its reliability in an ideal scenario.

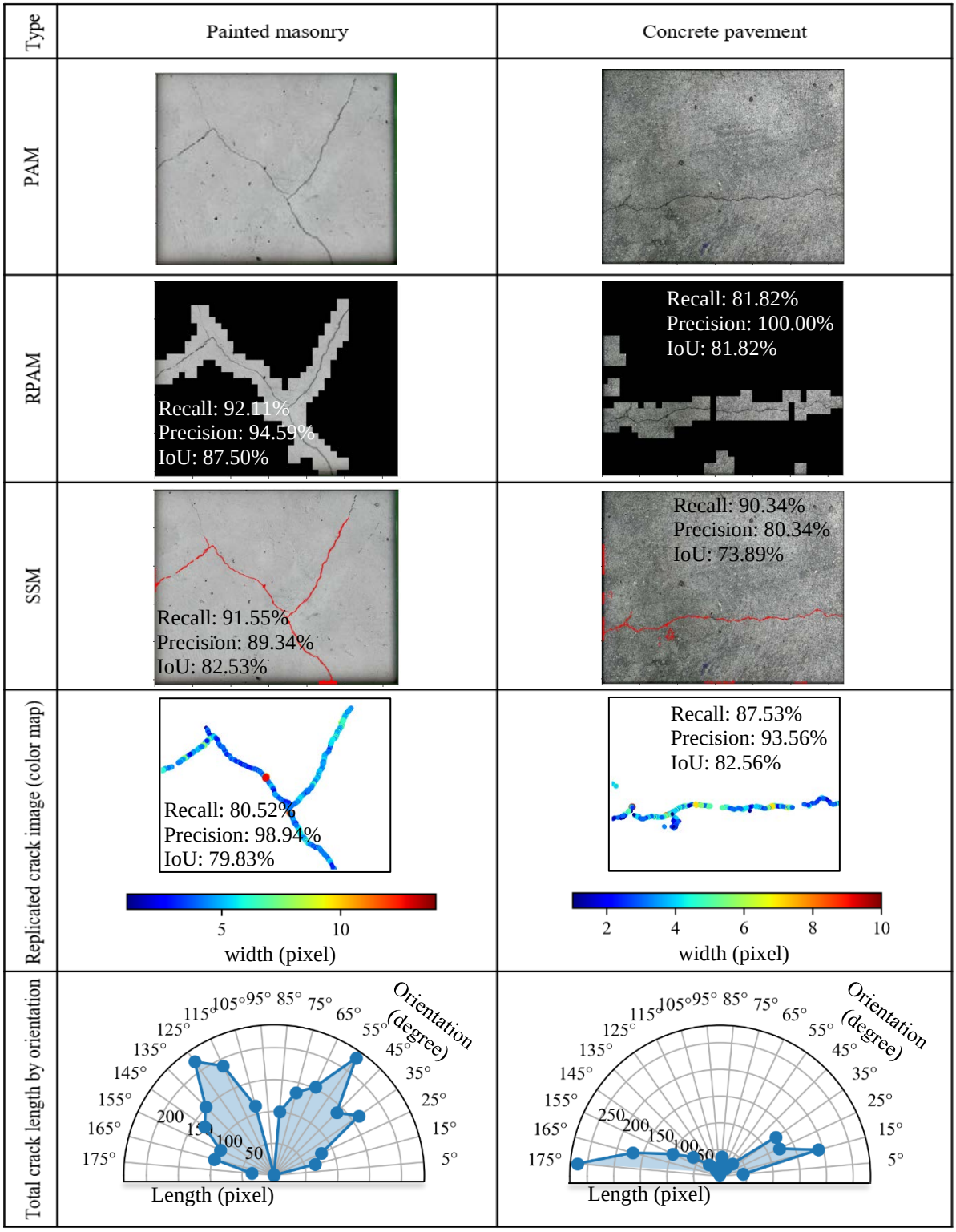

Fig. 11 shows examples of measurement of cracks on (1) painted masonry, and (2) concrete pavement.

5.1 Suggestions for Future Research

While the proposed methodology shows significant promise, further improvements can be pursued. Future work could focus on optimizing the semantic segmentation algorithm for better generalization across more complex and diverse surfaces. Additionally, integrating more advanced deep learning architectures, such as transformer-based models, could further enhance the precision and computational efficiency of crack detection and

analysis.

Expanding the dataset with a broader range of crack types, environmental conditions, and structural materials could also improve the model's robustness. Moreover, exploring real-time applications of the proposed approach through deployment on mobile platforms or UAVs could extend its utility in structural health monitoring, especially in inaccessible areas.

In conclusion, the proposed framework offers a practical, efficient, and adaptable solution for crack detection and quantification, providing valuable insights for infrastructure monitoring and maintenance. The findings lay a solid foundation for future advancements in automated crack analysis using deep learning techniques.

6. ACKNOWLEDGMENTS

We would like to express our gratitude to Kasetsart University for providing the necessary facilities for this research. Special thanks to Crack Detection Group for their invaluable assistance, insightful comments, and suggestions. We also appreciate the efforts of our colleagues who contributed to the peer review process, enhancing the quality and impact of our work.

7. REFERENCES

[1] Angst U. M., Challenges and Opportunities in Corrosion of Steel in Concrete, Materials and Structures, Vol. 51, No. 1, 2018, p. 4.

[2] Qian C. and Du W., Analysis Method of Apparent Quality of Fair-Faced Concrete Based on Convolutional Neural Network Machine Learning, Journal of Building Engineering, Vol. 80, 2023, p. 108154.

[3] Yodsudjai W., Vanrak P., Suwanvitaya P., and Jutasiriwong A., Corrosion Behavior of Reinforcement in Concrete with Different Compositions, Journal of Sustainable Cement-Based Materials, Vol. 10, No. 3, 2020, pp. 1-20.

[4] Yodsudjai W. and Pattarakittam T., Factors Influencing Half-Cell Potential Measurement and Its Relationship with Corrosion Level, Measurement, Vol. 104, 2017, pp. 159-168.

[5] Niemeier W., Riedel B., Fraser C., Neuss H., Stratmann R., and Ziem E., New Digital Crack Monitoring System for Measuring and Documentation of Width of Cracks in Concrete Structures, in Proc. of 13th FIG Symp. on Deformation Measurement and Analysis and 14th IAG Symp. on Geodesy for Geotechnical and Structural Engineering, Lisbon, Vol. 12, 2008, pp. 1-9.

[6] Jirawattanasomkul T., Kongwang N., Likitlersuang S., Yodsudjai W., Charuvisit S., and Sato Y., Failure Analysis of Dapped-End Cracking in Posttensioned Bridge Girder, Journal of Bridge Engineering, Vol. 26, No. 11, 2021, p. 04021082.

[7] Fu R., Cao M., Novák D., Qian X., and Alkayem N. F., Extended Efficient Convolutional Neural Network for Concrete Crack Detection with Illustrated Merits, Automation in Construction, Vol. 156, 2023, p. 105098.

[8] Mirbod M. and Shoar M., Intelligent Concrete Surface Cracks Detection Using Computer Vision, Pattern Recognition, and Artificial Neural Networks, Procedia Computer Science, Vol. 217, 2023, pp. 52-61.

[9] Shibano K., Morozova N., Shimamoto Y., Alver N., and Suzuki T., Improvement of Crack Detectivity for Noisy Concrete Surface by Machine Learning Methods and Infrared Images, Case Studies in Construction Materials, Vol. 20, 2024, p. e02984.

[10] Kaseko M. S., Lo Z. P., and Ritchie S. G., Comparison of Traditional and Neural Classifiers for Pavement-Crack Detection, Journal of Transportation Engineering, Vol. 120, No. 4, 1994, pp. 552-569.

[11] Yan W. Y. and Yuan X. X., A Low-Cost Video-Based Pavement Distress Screening System for Low-Volume Roads, Journal of Intelligent Transportation Systems, Vol. 22, No. 5, 2018, pp. 376-389.

[12] Ni T., Zhou R., Gu C., and Yang Y., Measurement of Concrete Crack Feature with Android Smartphone APP Based on Digital Image Processing Techniques, Measurement, Vol. 150, 2020, p. 107093.

[13] Rubanga D. P., May-Cuevas S. A., Arvelyna Y., and Shimada S., Fracture-Fault Detection Using Deep Learning with Stepwise Elimination from Satellite Images in Djibouti, International Journal of GEOMATE, Vol. 25, No. 108, 2023, pp. 241-248.

[14] Aryanto A., Revolis M., Oribe Y., and Yo H. J. G. J., Application of Digital Image Correlation Method in Rc and Frc Beams Under Bending Test, International Journal of GEOMATE, Vol. 24, No. 101, 2023, pp. 118-125.

[15] Uaisova M., Zharlykassov B., Aldasheva D., Artykbayeva A., and Radchenko P. J. G. J., The Use of ANN and Machine Learning Algorithms to Predict Road Surface Deterioration, International Journal of GEOMATE, Vol. 27, No. 121, 2024, pp. 136-143.

[16] Opara J. N., Thein A. B. B., Izumi S., Yasuhara H., and Chun P.-J., Defect Detection on Asphalt Pavement by Deep Learning, International Journal of GEOMATE, Vol. 21, No. 83, 2021, pp. 87-94.

[17] Cha Y.-J., Choi W., and Büyüköztürk O., Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks, Computer-

Aided Civil and Infrastructure Engineering, Vol. 32, No. 5, 2017, pp. 361-378.
[18] Cha Y.-J., Choi W., Suh G., Mahmoudkhani S., and Büyüköztürk O., Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types, Computer-Aided Civil and Infrastructure Engineering, Vol. 33, No. 9, 2018, pp. 731-747.
[19] Girshick R., Donahue J., Darrell T., and Malik J., Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, in 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587.
[20] Girshick R., Fast R-CNN, in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448.
[21] Redmon J., Divvala S., Girshick R., and Farhadi A., You Only Look Once: Unified, Real-Time Object Detection, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.
[22] Ren S., He K., Girshick R., and Sun J., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Trans Pattern Anal Mach Intell, Vol. 39, No. 6, 2017, pp. 1137-1149.
[23] Nie M. and Wang C., Pavement Crack Detection Based on Yolo v3, in 2019 2nd International Conference on Safety Produce Informatization (IICSPI), 2019, pp. 327-330.
[24] Cheng J., Xiong W., Chen W., Gu Y., and Li Y., Pixel-Level Crack Detection using U-Net, in TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea, 2018, pp. 0462-0466.
[25] Li H., Song D., Liu Y., and Li B., Automatic Pavement Crack Detection by Multi-Scale Image Fusion, IEEE Transactions on Intelligent Transportation Systems, Vol. 20, No. 6, 2019, pp. 2025-2036.
[26] Alipour M., Harris D. K., and Miller G. R., Robust Pixel-Level Crack Detection Using Deep Fully Convolutional Neural Networks, Journal of Computing in Civil Engineering, Vol. 33, No. 6, 2019, p. 04019040.
[27] Choi W. and Cha Y.-J., SDDNet: Real-Time Crack Segmentation, IEEE Transactions on Industrial Electronics, Vol. 67, No. 9, 2020, pp. 8016-8025.
[28] Tabernik D., Šuc M., and Skočaj D., Automated Detection and Segmentation of Cracks in Concrete Surfaces Using Joined Segmentation and Classification Deep Neural Network, Construction and Building Materials, Vol. 408, 2023, p. 133582.
[29] Liu H., Miao X., Mertz C., Xu C., and Kong H., Crackformer: Transformer Network for Fine-Grained Crack Detection, in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3783-3792.
[30] Liu Y., Yao J., Lu X., Xie R., and Li L., DeepCrack: A Deep Hierarchical Feature Learning Architecture for Crack Segmentation, Neurocomputing, Vol. 338, 2019, pp. 139-153.
[31] Zou Q., Cao Y., Li Q., Mao Q., and Wang S. J. P. R. L., CrackTree: Automatic Crack Detection from Pavement Images, Pattern Recognition Letters, Vol. 33, No. 3, 2012, pp. 227-238.
[32] Eisenbach M. *et al.*, How to Get Pavement Distress Detection Ready for Deep Learning? A Systematic Approach, in 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2039-2047.
[33] Pak M. and Kim S., Crack Detection Using Fully Convolutional Network in Wall-Climbing Robot, in Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 2019, 2021, pp. 267-272.
[34] Dorafshan S., Thomas R. J., and Maguire M. J. D. i. b., SDNET2018: An Annotated Image Dataset for Non-contact Concrete Crack Detection Using Deep Convolutional Neural Networks, Data in Brief, Vol. 21, 2018, pp. 1664-1668.
[35] Kim B., Natarajan Y., Preethaa K. R. S., Song S., An J., and Mohan S., Real-Time Assessment of Surface Cracks in Concrete Structures Using Integrated Deep Neural Networks with Autonomous Unmanned Aerial Vehicle, Engineering Applications of Artificial Intelligence, Vol. 129, 2024, p. 107537.
[36] Yang X., Li H., Yu Y., Luo X., Huang T., and Yang X., Automatic Pixel-Level Crack Detection and Measurement Using Fully Convolutional Network, Computer-Aided Civil and Infrastructure Engineering, Vol. 33, No. 12, 2018, pp. 1090-1109.
[37] Miao P. and Srimahachota T., Cost-Effective System for Detection and Quantification of Concrete Surface Cracks by Combination of Convolutional Neural Network and Image Processing Techniques, Construction and Building Materials, Vol. 293, 2021, p. 123549.
[38] Kang D., Benipal S. S., Gopal D. L., and Cha Y.-J., Hybrid Pixel-Level Concrete Crack Segmentation and Quantification Across Complex Backgrounds Using Deep Learning, Automation in Construction, Vol. 118, 2020, p. 103291.