

Advanced crack detection and segmentation on bridge decks using deep learning

Thai Son Tran ^{a,*}, Son Dong Nguyen ^b, Hyun Jong Lee ^{b,*}, Van Phuc Tran ^a

^a Research and Development Department, IRIS Technology, 301 Daeyang AI Center, 209 Neungdong-ro, Gunja-dong, Gwangjin-gu, 05006 Seoul, South Korea

^b Civil and Environmental Engineering, Sejong University, 209 Neungdong-ro, Gunja-dong, Gwangjin-gu, 05006 Seoul, South Korea



ARTICLE INFO

Keywords:

Bridge deck
Crack detection
Crack segmentation
High-resolution image
Deep learning

ABSTRACT

Detecting and measuring cracks on a bridge deck is crucial for preventing further damage and ensuring safety. However, manual methods are slow and subjective, highlighting the need for an efficient solution to detect and measure crack length and width. This study proposes a novel process-based deep learning approach for detecting and segmenting cracks on the bridge deck. Five state-of-the-art object detection networks were evaluated for their performance in detecting cracks: **Faster RCNN-ResNet50**, **Faster RCNN-ResNet101**, **RetinaNet-ResNet50**, **RetinaNet-ResNet101**, and **YOLOv7**. Additionally, two object segmentation networks, U-Net, and pix2pix, were optimized by experimenting with various network depths, activation functions, loss functions, and data augmentation to segment the detected cracks. The results showed that YOLOv7 outperformed both Faster RCNN and RetinaNet with both ResNet50 and ResNet101 backbones in terms of both speed and accuracy. Furthermore, the proposed U-Net is better than the mainstream U-Net and pix2pix networks. Based on these results, **YOLOv7** and the proposed U-Net are integrated for detecting and segmenting cracks on a bridge deck. The proposed method was then applied to two bridges in South Korea to test its performance, and the results showed that it could detect crack length with an **accuracy of 92.38 percent**. Moreover, the proposed method can determine crack width and classify it with an R^2 value of 0.87 and an average accuracy of 91 percent, respectively. In summary, this study provides an efficient and reliable method for detecting, measuring, and classifying cracks on a bridge deck surface.

1. Introduction

The use of deep learning techniques for crack detection in civil structures such as buildings, bridges, dams, roads, and others has seen a significant increase in recent years. With advanced deep learning techniques, researchers have developed and applied various algorithms to classify, detect, and segment cracks in different civil structures. In terms of classifying cracks, many researchers have proposed various approaches, mainly based on **convolutional neural networks** (CNNs) and deep neural networks (DNNs) [1–3]. These networks are trained using images of cracks, allowing them to extract features such as shape, size, and texture relevant to crack detection. Additionally, CNNs can be pre-trained on large datasets such as ImageNet [4] and fine-tuned on smaller images of cracks, allowing for more efficient training and better performance. For example [5], utilized deep learning and infrared thermography to classify asphalt pavement crack severity. They trained CNN models from scratch and with transfer learning on three image types:

visible, infrared, and fusion. The highest accuracy was obtained using **fusion images for deep learning and visible images for transfer learning**. These models effectively detected no cracks and low cracks across all image types. However, misclassifications were more prevalent for medium and high crack categories in all image types. In an extension of [5,6] focused on classifying the fatigue crack severity level of asphalt pavement. They applied Grad-CAM to interpret the **CNN models used for severity-level classification**. The authors reported that all CNN models achieved an accuracy higher than 95 percent.

The application of classification algorithms only can categorize crack and non-crack images or classify different crack types without giving information about the location of cracks in an image [7,8]. Therefore, it seems impossible to accurately quantify cracks from images. Due to this reason, various object detection algorithms such as **Faster RCNN** [9], **Single shot multi-box detector (SSD)** [10], **RetinaNet** [11], and **Yolo** [12] have been used and significantly improved to detect cracks in different civil structures. For example [13,14], applied Faster RCNN to detect cracks on the road pavement, [15] proposed a Multi-task Enhanced dam

* Corresponding authors.

E-mail addresses: ttson2992@gmail.com (T.S. Tran), hlee@sejong.ac.kr (H.J. Lee).

Nomenclature	
TP _{obj}	the number of positive objects that are correctly detected as positive
TN _{obj}	the number of negative objects that are correctly detected as negative
FP _{obj}	the number of negative objects that are falsely detected as positive
FN _{obj}	the number of positive objects that are falsely detected as negative
AO	area of overlap
AU	area of union
TP _{pixels}	the number of positive pixels that appear in the same position in both the labeled and predicted images
TN _{pixels}	the number of negative pixels that appear in the same position in both labeled and predicted images
FP _{pixels}	the number of predicted positive pixels as negative pixels in the labeled images
FN _{pixels}	the number of predicted negative pixels as positive pixels in the labeled images

crack image detection method based on Faster R-CNN [16], also used Faster RCNN to detect cracks on the bridge. [17–19] used RetinaNet and YOLOv3 to detect road damage and other surface objects on the road. [20] improved the YOLOv3 algorithm to detect cracks on the bridge surface.

The need for crack information extends beyond detection and includes the requirement for crack segmentation. Image processing and deep learning methods are the two main approaches to segmenting cracks from an image. In terms of crack segmentation-based image processing methods, researchers used threshold, Otsu, filtering techniques, etc., to segment cracks from an image [21,22,23]. Despite the successful application of these image-processing approaches in crack detection, concerns regarding their accuracy and efficiency still require attention [7]. In contrast, crack segmentation-based deep learning shows good accuracy and analysis speed performance. For example [24], implemented a U-Net network with a pre-trained ResNet-34 encoder for crack segmentation on pavement images, achieving an impressive F1-score of approximately 96 percent on the CFD dataset and about 73 percent on Crack500. In [25], SegCrack, a hierarchical Transformer-based model, showed remarkable results with an F1-score of 96.05 percent and a mIoU of 92.63 percent on concrete crack segmentation. Additionally [26], presented a robust CNN-based crack detection method using infrared thermography, where fusion images outperformed visible images for all models. Feature Pyramid Networks (FPN) is the top-performing choice in accuracy and simplicity.

Moreover, some researchers combined different techniques to solve complex tasks. For example [27,28], proposed a method to automatically detect and segment pavement cracks by the combination of different deep learning and image processing techniques. [29] successfully segmented cracks in close-range building facade inspection images using deep learning techniques. From this research, it can be seen that combination of various techniques shows a good performance in solving a difficult task. This reason motivates us to propose a novel method based on different deep learning algorithms to efficiently detect and segment cracks on bridge deck surface from a complex background and high-resolution image.

The proposed method begins with object detection algorithms to detect the presence of cracks. This study evaluated state-of-the-art object detection algorithms for their accuracy and processing time performance, including Faster R-CNN-ResNet50, Faster RCNN-ResNet101, RetinaNet-ResNet50, RetinaNet-ResNet101, and Yolov7. Following the detection of cracks, the final step is to optimize the network architecture

and other factors, such as loss function and activation function in segmentation algorithms U-Net [30] and pix2pix [31], to find the optimum network to segment the crack and determine its length and width. Through the utilization of various deep learning techniques, this study aims to propose an efficient method for detecting and segmenting cracks in the complex background and high-resolution images from bridge decks surface and provide recommendations for future research in this field. The primary contribution of this work can be summarized as follows:

- After evaluating five state-of-the-art object detection networks, the study recommends adopting the network that is the best in both speed and accuracy for detecting cracks on bridge decks, mainly when dealing with high-resolution images.
- Optimize segmentation algorithms for precise crack segmentation on complex bridge deck backgrounds.
- Develop an application for determining crack length and width on bridge decks.
- The proposed method is critical to ensure effective and efficient crack detection in real-world applications.

The paper is organized as follows. Section 2, "Related Work," provides a comprehensive overview of current studies that have been developed to classify, detect, or segment cracks on bridge decks. Section 3, "Methodology," presents the proposed method for detecting and segmenting cracks on bridge decks. Section 4, "Data Collection," discusses the dataset used in this study, including its acquisition and preparation. Section 5, "Model Evaluation," presents the metrics used to evaluate crack detection and segmentation and describes the experimental setup used in this study.

Sections 6 and 7 are dedicated to the two main aspects of the proposed method. Section 6, "Crack Detection based on Object Detection Algorithms," discusses the process of model training and presents the results of three object detection networks that were used to detect cracks on bridge decks. Section 7, "Crack Width Assessment based on Segmentation Algorithms," presents how to optimize the segmentation network to determine the length and width of the detected cracks accurately. Section 8, "Application," applies the proposed method to two bridges in South Korea to test its performance. Finally, Section 9, "Conclusion and Recommendation," draws significant research findings and provides recommendations for future research.

2. Related work

Detecting and quantifying cracks on a bridge deck is crucial to prevent further damage and ensure its safe use. As a result, many automated crack detection algorithms for bridge decks have been developed. Previous studies on automated crack detection algorithms for bridge decks have focused on identifying visually distinct cracks with high contrast. These types of cracks can be detected using standard image processing techniques such as edge following, image thresholding, and morphology operations [32,33]. In 2014, [34] overcame the limitations of previous studies by presenting an automated crack detection algorithm called STRUM (spatially tuned robust multi-feature) that achieved high accuracy on images of real concrete bridge decks. Still, the method is sensitive to noise in the surveyed images, and its bulky process makes it unsuitable for real-world applications.

In 2017, [35] developed an autonomous robotic system for inspecting bridge decks. The system uses multiple processes to segment cracks from bridge deck surface images, including gradient crack detection, crack cleaning and linking, and noise removal. The approach depends on image processing, which is prone to being time-consuming and highly sensitive to the quality of the captured images. Hence, it might not be suitable for identifying cracks on intricate surfaces with significant noise levels and high-resolution bridge deck images. In 2021, Zhang et al. proposed an approach for detecting cracks on bridge decks using an

integrated one-dimensional convolutional neural network (1D-CNN) and long short-term memory (LSTM) method in the image frequency domain [36]. The authors demonstrated that their approach achieves high accuracy in crack detection. However, their study did not take into account the crack width and the analysis time.

Previous research indicates that accurately and efficiently detecting and segmenting cracks in bridge decks from a complex background remains a challenging task, as highlighted in Fig. 1. Moreover, a comparison presented in Table 1 reveals that past studies focused only on crack detection without considering crack width or analysis time for high-resolution bridge deck images. To address this gap in the literature, the current study conducted a comprehensive study that optimizes various object detection and segmentation algorithms to overcome these limitations.

Object detection algorithms can be divided into two and one-stage methods. Faster RCNN is recognized as a representative of a two-stage object detection model that builds upon the Fast R-CNN by integrating a region proposal network (RPN) with the CNN model. Unlike its predecessor, the RPN in Faster R-CNN shares the full-image convolutional features with the detection network, significantly reducing computational costs for region proposals. Moreover, it is a fully convolutional network that concurrently predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which Fast R-CNN utilizes for object detection. By sharing their convolutional features, the RPN and Fast R-CNN are merged into a unified network that enables the RPN component to direct the unified network toward the regions of interest [9].

In contrast, one-stage object detection algorithms are RetinaNet and YOLO. RetinaNet utilizes a multi-scale feature pyramid network to simultaneously predict object classes and locations, blending the benefits of both one-stage and two-stage object detection. RetinaNet used focal loss, which applies a modulation term to the cross-entropy loss to down-weight the loss for well-classified examples. It focuses on hard examples, making it more effective for detecting challenging objects. Based on that RetinaNet is one of the best single-stage object detection models that work well with dense and small objects [11].

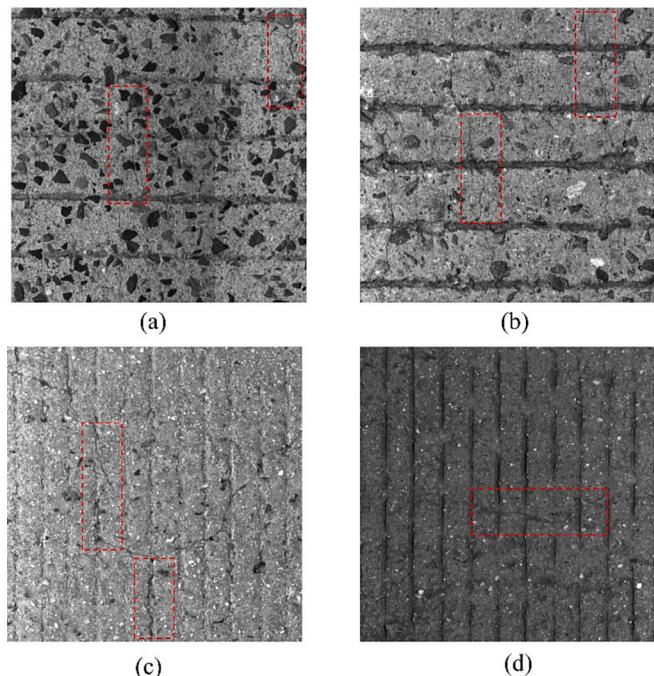


Fig. 1. Nature complex of bridge deck data. a) the background with many noises, b) noise and hairy cracks, c) cracks on the scratch line, d) cracks similar to the background.

At the same, YOLO predicts the bounding boxes and class probabilities for objects in an image through a single neural network. Currently, YOLOv7 is a state-of-the-art, real-time object detector. Similar to previous versions, it also includes a backbone, head, and neck. The backbone extracts key features of the image, fed to the head through the neck, and the head consists of the output layers with final detections. However, one of the key improvements in YOLOv7 is that it also uses focal loss to improve accuracy in detecting small objects. Additionally, the most significant improvement in YOLOv7 is using extended efficient layer aggregation networks (E-ELAN), which control the shortest and longest gradient path for effective learning and convergence [12].

In terms of crack segmentation, U-Net [30] and pix2pix [31] are popular algorithms for image-to-image processing. However, selecting a suitable model for a given dataset often necessitates extensive research and experimentation. Moreover, the accuracy of the outcomes is heavily influenced by the appropriate selection of parameters and layer structures for each model. In this study, the requirement is to detect cracks on bridge decks from high-resolution images with high accuracy and efficient time. Therefore, state-of-the-art object detection algorithms, such as Faster R-CNN (Resnet50 and Resnet101 backbones), RetinaNet (Resnet50 and Resnet101 backbones), and YOLOv7 were used to detect cracks and test their speed and accuracy performance. Moreover, the main objective of this study is to accurately segment cracks on the surface of a bridge deck to determine their width based on earlier detection. As shown in Fig. 1, the bridge deck surface is highly complex and contains significant noise, while the cracks themselves are quite narrow. To achieve this goal, the authors optimized segmentation algorithms and proposed a model that meets the agency's specific needs.

3. Methodology

Currently, no object detection network is capable of handling images with a resolution of $18,000 \times 10,000$ pixels. To address this challenge, two solutions have been proposed. The first option involves scaling down the original image, which could result in the loss of critical crack features. Therefore, this study adopts the second solution, which involves dividing the original image into smaller patches. Finding the optimal size of patches is critical to achieving a balance between analysis time and accuracy. To accomplish this goal, the validation step involved using different patch sizes. Based on the results, the optimal patch size was determined in terms of both speed and accuracy.

After this step, cracks on the bridge deck can be detected with rectangle bounding boxes. However, this box cannot explain the exact crack length and width. Due to this reason, the segmentation techniques were then used to precise crack length as well as the width of cracks from the detected bounding box. Specifically, U-Net and pix2pix networks were optimized for segmenting cracks from the detected zone. Then, the best crack segmentation model was suggested to segment cracks to obtain the exact crack length and width on the bridge deck. The whole process is shown in Fig. 2.

4. Data collection

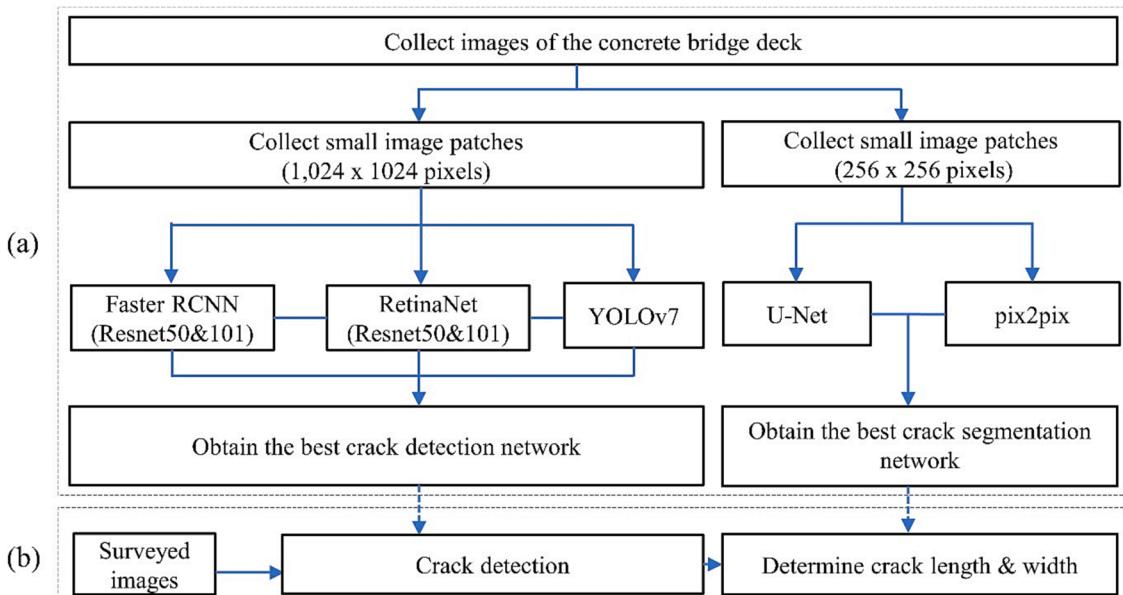
Several bridge decks in South Korea were surveyed using a 3D-mounted vehicle camera to collect surface images. The collected images have a resolution of $18,000 \times 10,000$ pixels, corresponding to $3,600 \times 2,000$ mm in the width and length directions, respectively. In other words, one pixel in each surveyed image represents 0.2 mm in real scale. After the survey, a total of 100 images with the presence of cracks were selected from different survey locations to create the dataset.

As mentioned earlier, in the crack detection step, the original images with $18,000 \times 10,000$ pixels were divided into smaller patches measuring $1,024 \times 1,024$ pixels for training and validation. Only the small patches containing cracks were selected to create the labels for training and validation using an annotation tool called LabelImg [37]. The labeled data will be used to train and validate various object

Table 1

Related works for detecting cracks on the bridge deck surface.

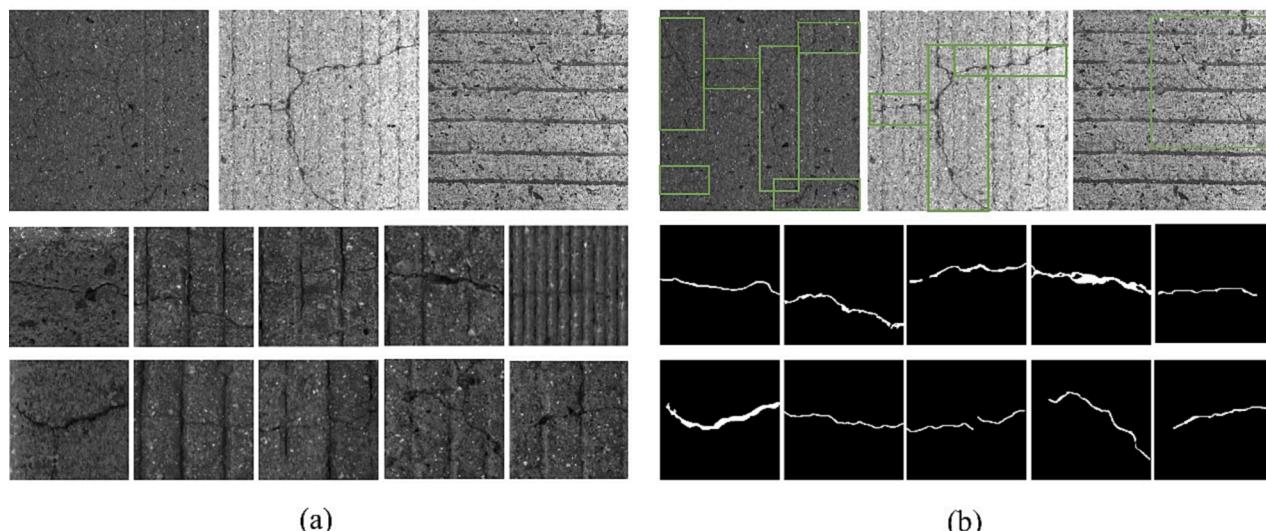
Research	Image processing	Deep learning	Crack detection	Crack segmentation	Crack width	Analysis Time
Lim, Ronny Salim, et al. (2011)	yes		yes			
Adhikari, R. S., O. Moselhi, and A. Bagchi. (2014)	yes		yes			
Prasanna, Prateek, et al. (2014)	yes		yes			
La, Hung M., et al. (2017)	yes		yes			
Zhang, Qianyun, et al. (2021)		yes	yes			
Proposed Method		yes	yes	yes	yes	yes

**Fig. 2.** The process of the proposed method for detecting and segmenting cracks from the high-resolution image of the bridge deck. a) Training and validation, b) Testing Process.

detection networks. In this study, a total of 1,155 and 286 images were used for the training and validation crack detection process.

Moreover, in order to train the crack segmentation model using U-Net, the process began with selecting 800 images based on the crack object detection results. Each image was cropped to 256 × 256 pixels and manually labeled using the Image Labeler app in Matlab software [38]. The resulting dataset was split into a training set of 700 images and

a validation set of 100 images. Otherwise, the input and output data were normalized by dividing each image pixel value by 255, standardizing the range to [0,1] to improve training stability and prevent the model from disproportionately affecting large input values. This preprocessing step was crucial to ensure the accuracy of the resulting crack segmentation. An example of the dataset used in this study is illustrated in Fig. 3.

**Fig. 3.** Sample data for training crack detection and segmentation networks a) original data, b) labeled data.

After training and validating crack detection and segmentation models, the authors collected testing data from various bridge decks across South Korea to test the trained models' performance, as detailed in the Application section.

5. Model evaluation

5.1. Crack detection

To evaluate the model in deep learning, three metrics were usually used: precision, recall, and F1-score. Among them, precision measures the proportion of true positives out of all the positive predictions made by the model. The number of false positives is low if a model has a high precision value. On the other hand, if the recall value is high, the number of false negatives is low, and the proportion of true positives is high compared to all actual positive samples. While F1-score provides a single score that balances the trade-off between precision and recall. It should be noted that these metrics are determined using an intersection over union (IoU). Typically, a value of 0.5 is used for IoU in real-world applications. Eqs. (1)–(4) are utilized to determine precision, recall, F1-score, and IoU, respectively. Moreover, the mean average precision (MAP) is a common evaluation metric used to evaluate the overall performance of a model. In summary, MAP is the average of the average precision (AP) values for all classes. Where AP is computed as the average of the maximum precision values at each recall level.

$$\text{Precision} = \frac{TP_{obj}}{TP_{obj} + FP_{obj}} \quad (1)$$

$$\text{Recall} = \frac{TP_{obj}}{TP_{obj} + FN_{obj}} \quad (2)$$

$$F_{1score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

$$IoU = \frac{AO}{AU} \quad (4)$$

5.2. Crack segmentation

Regarding crack segmentation, the labeled image can be represented as a binary image, where positive points indicate the presence of cracks (with a value of 1), and negative points indicate the absence of cracks (with a value of 0). It is evident that the crack segmentation problem is an imbalanced classification task, as the positive points account for only a small fraction (typically less than 5 percent) of the total image area. As a result, employing accuracy as a metric for evaluating model performance, as indicated by Eq. (5), does not adequately reflect the model's performance since an all-black prediction would still yield an accuracy greater than 95 percent or even 98 percent.

In order to overcome this issue, the study assesses model performance using precision, recall, F1-score, and IoU as defined by Eq. (6), (7), (8), and (9), respectively [39]. By employing these metrics, the model's performance is evaluated based on its ability to accurately identify and correctly classify the positive points (i.e., cracks) in the labeled image. This approach provides a more comprehensive evaluation of model performance and is better suited for the imbalanced nature of the crack segmentation problem.

$$\text{Accuracy} = \frac{TP_{pixels} + TN_{pixels}}{TP_{pixels} + FP_{pixels} + TN_{pixels} + FN_{pixels}} \quad (5)$$

$$\text{Precision} = \frac{TP_{pixels}}{TP_{pixels} + FP_{pixels}} \quad (6)$$

$$\text{Recall} = \frac{TP_{pixels}}{TP_{pixels} + FN_{pixels}} \quad (7)$$

$$F_{1score} = \frac{2 \times \text{Precision}_{pixel} \times \text{Recall}_{pixel}}{\text{Precision}_{pixel} + \text{Recall}_{pixel}} \quad (8)$$

$$IoU = \frac{TP_{pixels}}{TP_{pixels} + FP_{pixels} + FN_{pixels}} \quad (9)$$

6. Crack detection based object detection algorithms

6.1. Model training and result

A single computing system powered by a GeForce GTX 1080Ti GPU was used to comprehensively train and evaluate the effectiveness of three unique neural networks built using the PyTorch platform [40]. Optimizing the models' parameters is essential to achieve optimal performance; however, this process can be time-consuming. Therefore, the authors focused specifically on the learning rate parameter, which is known to impact the performance of the model significantly. By carefully adjusting learning rates within a range of 0.000001 to 0.001, the authors sought to optimize the performance of each network. The training process was conducted with 100 epochs for each learning rate and network. As discussed earlier, the original image has a high resolution; hence, different image sizes were used in the validation step to investigate the optimum image size to improve speed and still ensure high accuracy. Detailed information regarding the network training setups is presented in Table 2, while other training parameters were held constant at their default values as prescribed by the respective network configurations [41–43].

During the training process, a model was generated at the end of each epoch, and its performance was evaluated using the validation dataset. It is worth noting that the detection of cracks is different from the detection of typical objects since cracks are not isolated entities. Therefore, the accuracy of crack detection cannot be fully reflected by conventional evaluation metrics. However, in training and validation steps, the metrics can be used to roughly compare each network as well as select the best model for each network. In this study, the mean average precision at a threshold of 0.5 (MAP@0.5) was considered to compare the object detection networks. The results of evaluating five networks for crack detection are summarized in Table 3. It demonstrates that Faster RCNN and RetinaNet with the ResNet101 backbone outperformed those with ResNet50. However, it should be noted that the inference time increased with the use of ResNet101 as compared to ResNet50. Among the object detection networks tested, YOLOv7 achieved the highest MAP@0.5 score of 0.748, surpassing Faster RCNN-ResNet101 (0.728) and RetinaNet-ResNet101 (0.727). Besides, similar results were obtained when evaluating the performance of each model in terms of MAP (0.5:0.95) across a range of thresholds of IOU from 0.5 to 0.95. YOLOv7 has the highest value of MAP(0.5:0.95) of 0.41, while Faster RCNN-Resnet101 and RetinaNet-Resnet101 have lower scores of 0.36 and 0.313, respectively. The result of MAP(0.5:0.95) shows that the YOLOv7 also can fit crack better than other networks.

In addition, YOLOv7 significantly improves speed compared to Faster RCNN-Resnet101 and RetinaNet-Resnet101, being 8.2 and 3 times faster, respectively. To analyze a 2-meter stretch of the bridge deck, Faster RCNN and RetinaNet necessitate around 32 and 10 s, respectively, for evaluation. On the other hand, YOLOv7 only needs less

Table 2
Networks training parameters.

Network	Learning Rate	Training Image Size	Batch Size	Backbone
Faster RCNN	1E-3	600 × 600	1	ResNet50, ResNet101
RetinaNet	1E-5	608 × 608	2	ResNet50, ResNet101
YOLOv7	1E-4	640 × 640	8	CSPDarknet

Table 3

The comparison of five object detection networks for crack detection on the bridge deck surface.

Network	MAP@0.5	MAP@0.5–0.95	Speed1(s)	Speed2(s)
Faster R-CNN-resnet50	0.712	0.346	0.164	29.5
Faster R-CNN-resnet101	0.728	0.360	0.180	32.4
RetinaNet-resnet50	0.720	0.310	0.055	9.90
RetinaNet-resnet101	0.727	0.313	0.065	11.7
YOLOv7	0.748	0.410	0.022	3.96

Note: Speed1 is obtained with image size of 1024×1024 pixels; speed2 is calculated for the original image with the size of $18,000 \times 10,000$ pixels.

than 4 s to analyze the same stretch and guarantees superior accuracy. Based on these results, YOLOv7 was selected as the most appropriate network for detecting cracks on the bridge deck surface. Fig. 4 illustrates the relationship between the F1-score and confidence levels for a model of the YOLOv7 network. YOLOv7 achieves peak performance at a confidence level of 0.33. Based on this graph, the optimum confidence levels for any trained models can be chosen for the application. This result highlights the significant impact of confidence level on object detection network performance, underscoring the need to carefully consider this parameter in the context of the specific application to maximize the network's performance.

Besides, the authors aimed to decrease the analysis time for crack detection by increasing the image size during validation. To investigate the effect of image size on the model's performance, different sizes were used, and Fig. 5 shows the resulting MAP@0.5 and speed for each size tested. The MAP@0.5 was slightly reduced when the image size changed from $1,024 \times 1,024$ to $3,072 \times 3,072$ but dramatically decreased after $4,096 \times 4,096$. This suggests that YOLOv7 cannot recognize small cracks from an image size of $4,096 \times 4,096$. Based on the results, an image size of $2,048 \times 2,048$ achieved almost the same accuracy as $1,024 \times 1,024$, but at the cost of a quarter reduction in analysis speed, and was thus chosen for the testing phase. The next chapter will focus on optimizing the crack segmentation algorithms to determine the width of the detected cracks.

7. Crack width assessment based segmentation algorithms

7.1. U-net model

7.1.1. U-net architecture

The U-Net architecture comprises an encoder network that progressively reduces the spatial resolution of the input image and a decoder

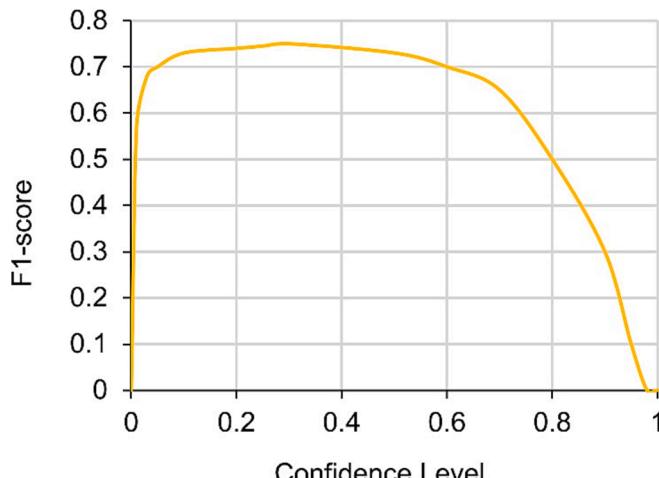


Fig. 4. Relationship between F1-score and confidence level of YOLOv7 model.

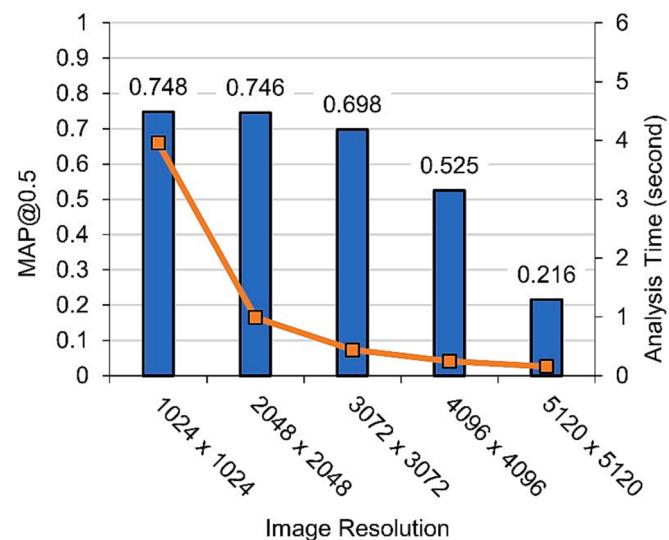


Fig. 5. Relationship between validation's image size versus MAP@0.5 and analysis time.

network that, conversely, increases the resolution of the encoded feature maps, as illustrated in Fig. 6. Like a conventional CNN architecture, the encoder network extracts relevant features from the input image and compresses them into a compact representation. The decoder network then takes this representation and performs a reverse downsampling operation to generate a segmentation mask of the exact resolution as the input image.

A significant innovation of the U-Net architecture is the introduction of bypass connections, which enable the decoder to access feature maps from the encoder at corresponding spatial resolutions. This is a critical feature of the architecture as it preserves high-resolution details that may otherwise be lost during encoder downsampling, resulting in more accurate image segmentation. Additionally, each block of the U-Net architecture includes a CNN and an activation and batch normalization layer to help the network learn more non-linear relationships in the data and reduce the internal covariate shift, ultimately improving the convergence speed of the network.

7.1.2. Improving crack segmentation with optimized u-net model

The optimization of the U-Net model was achieved through experimentation with a comprehensive set of parameters. These parameters included the depths of the network, the choice of activation functions, the selection of loss functions, and the application of data augmentation. The depth of the network refers to the number of layers in the encoder and decoder networks. In this study, three models, shown in Table 4, were used to determine the optimal number of layers that resulted in the best segmentation performance. The choice of activation function also plays a vital role in the overall performance of the U-Net model. Therefore, different activation functions, such as ReLU [44,45], LeakyReLU [46], ELU [47], and Swish [48], were used to determine the optimal choice for the model. In addition to activation functions, the selection of loss functions was also investigated. Various loss functions were tested, including binary cross-entropy [49], L2 loss, and Kullback-Leibler divergence loss [50], to determine the best one for the U-Net model. Finally, data augmentation techniques, such as rotation, flipping, random brightness, and random contrast, were applied to the training dataset to increase its size and diversity. This step helped improve the U-Net model's robustness and reduce overfitting.

The Adam optimizer was employed to optimize the model's parameters, and the learning rate was adjusted between 0.00001 and 0.001 to identify the optimal value. In order to enhance the model's generalization performance, dropout regularization was utilized, with the dropout

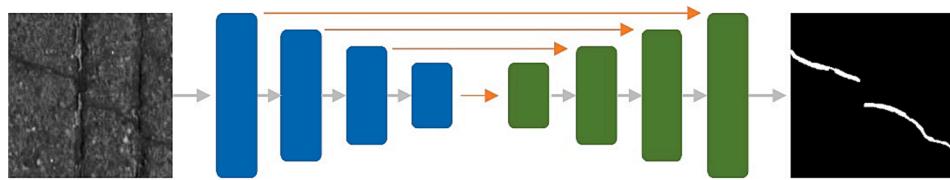


Fig. 6. U-net architecture.

Table 4
The summary of three U-Net models.

	U-Net-1	U-Net-2	U-Net-3
Layers	38	46	54
Params	16,656,641	66,996,481	117,336,321

rate ranging from 0.1 to 0.5. The training process was carried out over a total of 100 epochs.

The U-Net model was optimized in this study through an extensive parameter experimentation process, as shown in Table 5. The findings reveal that increasing the number of layers led to higher model accuracy. Still, there was a point where the accuracy became negligible, and the model's performance began to deteriorate while taking up more memory and becoming slower. Therefore, U-Net-2 was chosen because it balanced the accuracy with the structure's complexity. The activation layer was experimented with, and it was discovered that ReLU performed better than ELU, LeakyReLU, and Swish on the bridge deck crack segmentation dataset.

Furthermore, the Kullback-Leibler divergence loss function was observed to outperform binary cross-entropy and L2 loss in consistently achieving high accuracy during training. Additionally, data augmentation was found to enhance the U-Net model's performance. Overall, the optimized U-Net-2 model, as shown in Fig. 7, using ReLU activation, data augmentation for pre-processing, and Kullback-Leibler divergence loss, as Eq. (10), for training, demonstrated the best results on the bridge deck crack segmentation dataset. These findings shed light on the importance of parameter optimization and the selection of appropriate techniques for achieving optimal model performance in image segmentation tasks.

$$Loss_{DKLD} = \sum_i label_i [\log(label_i) - pred_i] \quad (10)$$

7.2. pix2pix model

7.2.1. pix2pix architecture

The pix2pix architecture is a widely used deep learning model for image-to-image translation tasks, the same as the U-Net. However, it consists of two distinct components: a generator (G-Net) and a

Table 5
The performance of U-Net models on validation data.

Model	F1-score	Precision	Recall	IoU
U-Net-1 ReLU L2loss	0.716	0.843	0.623	0.558
U-Net-1 ReLU BCEloss	0.734	0.843	0.650	0.580
U-Net-1 ReLU KLDloss	0.687	0.821	0.590	0.522
U-Net-2 ReLU L2loss	0.734	0.802	0.676	0.579
U-Net-2 ReLU BCEloss	0.649	0.876	0.516	0.481
U-Net-2 ReLU KLDloss	0.745	0.835	0.672	0.593
U-Net-2 LeakyReLU KLDloss	0.722	0.816	0.648	0.565
U-Net-2 ELU KLDloss	0.673	0.835	0.564	0.507
U-Net-2 Swish KLDloss	0.717	0.801	0.649	0.559
U-Net-2 ReLU KLDloss Aug	0.778	0.835	0.729	0.637
U-Net-3 ReLU L2loss	0.737	0.802	0.681	0.583
U-Net-3 ReLU BCEloss	0.738	0.803	0.682	0.584
U-Net-3 ReLU KLDloss	0.741	0.812	0.681	0.588

discriminator (D-Net), as illustrated in Fig. 8. The generator takes an input image and produces an output image in the target domain, like the U-Net. On the other hand, the discriminator is designed to differentiate between the synthesized output image and the genuine output image from the target domain.

In this study, the author utilized the G-Net architecture proposed in the U-Net model and optimized it according to the parameters and hyperparameters identified in the previous experiments. A simple structure of convolutional neural networks (CNNs) was used for the discriminator (D-Net). It takes in a pair of images (input image and corresponding target image) and distinguishes between the generated and actual output images from the target domain. The discriminator is trained to maximize the probability of correctly classifying the real and fake images.

The CNN architecture of D-Net includes several convolutional and pooling layers, with batch normalization and LeakyReLU activation functions applied after each layer to improve the model's stability and performance, as shown in Fig. 9. The output of the last layer is a scalar value that indicates the probability that the input image is real or fake. The discriminator is trained using the binary cross-entropy loss function to update its parameters and improve its classification accuracy.

7.2.2. Performance of pix2pix model

Model training was conducted using the PyTorch platform with a single GeForce GTX 1080Ti GPU. The training process was run for 100 epochs. The Adam optimizer, with a beta1 value of 0.5, was employed to control the weight update rate. Additionally, dropout regularization was applied with a rate of 0.1 to prevent overfitting, and batch normalization was set to a value of 0.00001 for normalization purposes. Two types of loss functions, namely Kullback-Leibler divergence loss and binary cross-entropy loss, were combined to achieve optimal results. The former is more effective in preserving high accuracy over a prolonged training period, while the latter helps deal with imbalanced data distributions. Overall, this training setup aimed to balance the benefits of various techniques to ensure the best performance of the model.

Based on the analysis presented in Table 6, it is evident that the proposed U-Net model outperforms pix2pix in terms of precision, recall, and F1-score. Furthermore, as depicted in Fig. 10, the U-Net model achieves better results for both crack length and width than pix2pix. Although pix2pix occasionally offers superior results than U-Net, the difference is insignificant, as illustrated in Fig. 10a. This suggests that a more complex model may not necessarily yield better performance for a specific problem.

Moreover, the state-of-the-art U-Net models seem to offer a slight advantage for this application, despite minor performance gains achieved by adding more layers, which comes at the cost of increased complexity and reduced speed [51]. Thus, U-Net-based models with ResNet, DensNet, EfficientNet, InceptionResNet, or Attention do not consistently outperform the standard UNet models. Furthermore, the accuracy of UNet-based models tends to be lower because the encoder is too shallow or too deep [52]. Based on the findings, the proposed U-Net model is deemed the most suitable for the crack segmentation problem on the bridge deck, achieving precision, recall, F1-score, and IoU accuracies of 0.835, 0.729, 0.778, and 0.637, respectively.

In practical scenarios such as agency projects, the precise determination of crack width is critical in determining the appropriate

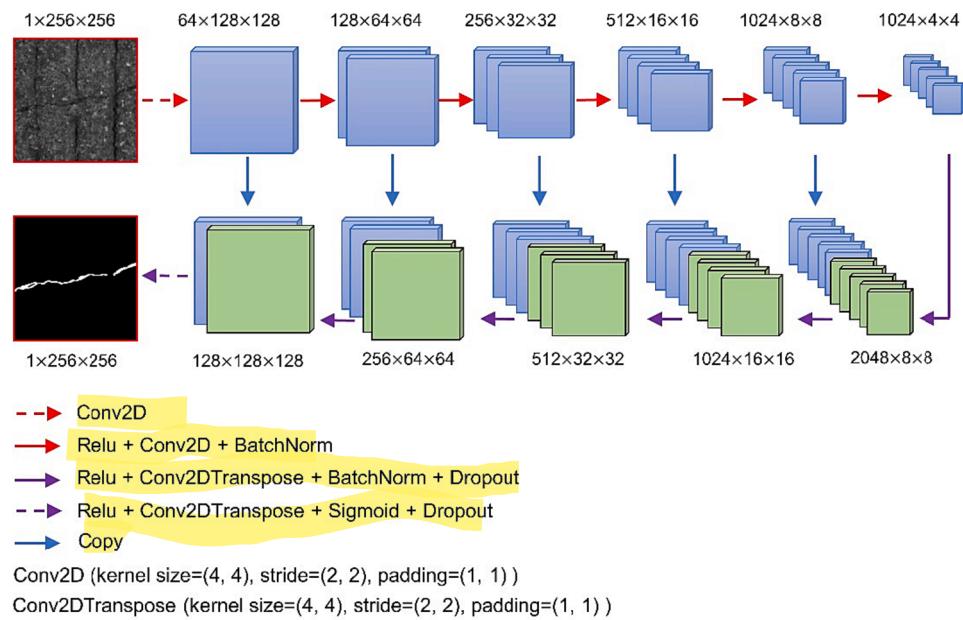


Fig. 7. The proposed U-Net with the best performance for crack segmentation on the bridge deck surface data.

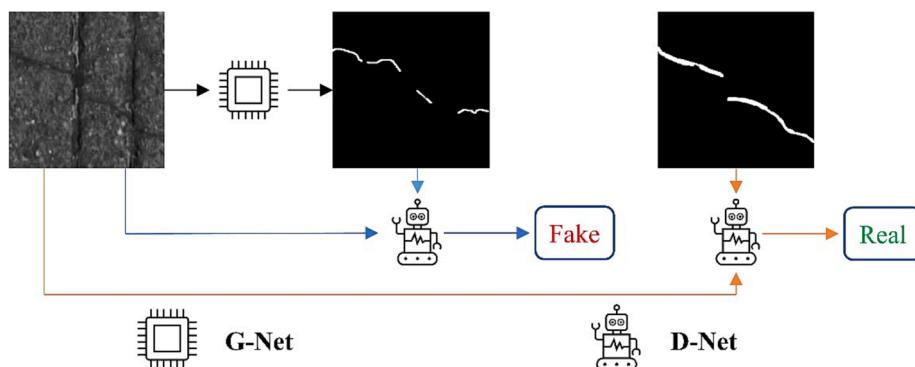


Fig. 8. Pix2pix architecture.

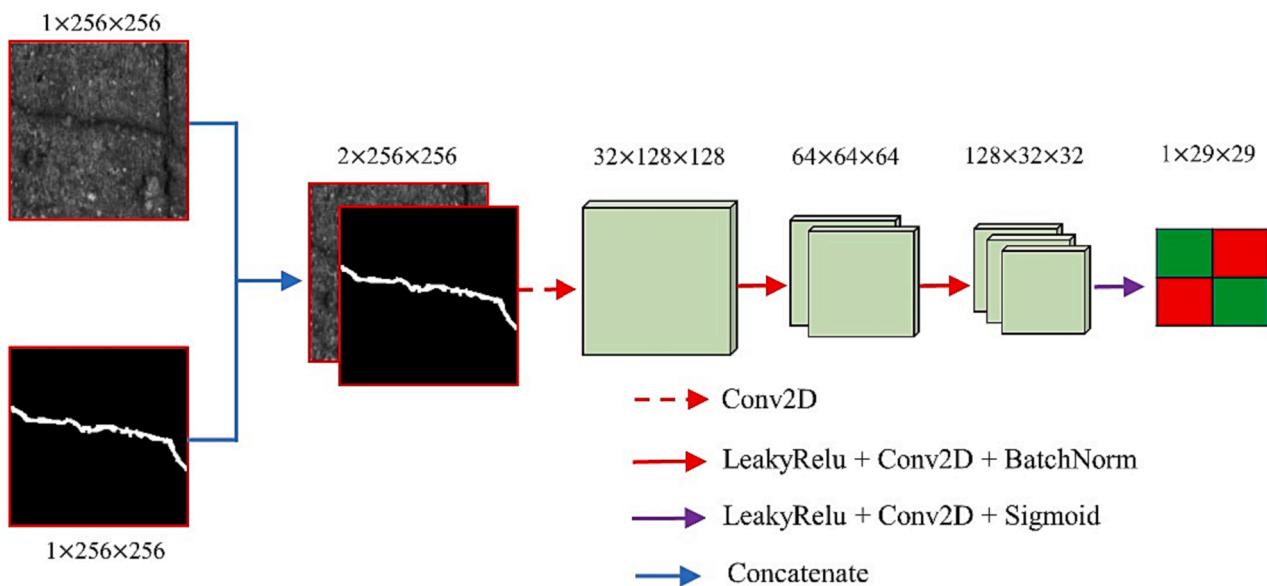


Fig. 9. Discriminator network.

Table 6

Comparison between the proposed U-Net model and the pix2pix model.

Model	F1-score	Precision	Recall	IoU
Proposed U-Net	0.778	0.835	0.729	0.637
pix2pix	0.747	0.811	0.691	0.595

rehabilitation approach. Thus, the maximum and average crack widths were processed using post-processing techniques. The results of the proposed U-Net model were compared against pix2pix in terms of accuracy for crack width, as shown in Fig. 11. The comparison between the proposed U-Net and pix2pix models showed that the proposed U-Net model achieved higher accuracy in predicting both the maximum and average ground truth crack widths, with R^2 values of 0.85 and 0.86, respectively. In contrast, the pix2pix model achieved R^2 values of 0.76 and 0.78 for the maximum and average ground truth crack widths, respectively. These findings suggest that the proposed U-Net model outperforms pix2pix. Additionally, the use of average crack width is deemed more appropriate for representing cracks on bridge deck surfaces.

8. Application

The proposed process for detecting and segmenting cracks on bridge decks was tested on two bridges in South Korea: The Daepocheon bridge in Busan (420 m \times 12.1 m) and the Naechoncheon bridge in Seoul (171.0 m \times 24.3 m), using a comprehensive survey. The survey data included a total of 50 images selected based on the presence of cracks on their surface. From that, 9,000 images with the size of 1,024 \times 1,024 were created, and the authors randomly selected 300 images from these data for the testing process. As shown in Table 7, the YOLOv7 model shows an impressive performance on testing data, which closely mirrors

its performance on the validation data. This compelling result indicates that the trained model exhibits both excellence and stability, making it a reliable choice for the application of diverse datasets.

In terms of crack detection, MAP@0.5 is only a reference index used to choose the best model for object detection networks and does not represent the actual accuracy of the model. It can be seen in Fig. 12a, a small detected box was assigned as a false positive. However, this false positive box can be removed by removing the overlap for detected bounding boxes. In addition, in Fig. 12b two boxes were assigned as false positives in comparison with the ground truth. However, in the judgment of crack detection, this case can be assigned as a true positive. Therefore, in real applications, the accuracy of crack detection was obtained by manually checking the detection and quantifying the over or miss detection of cracks. From the manual checking process, the authors noted that errors in crack detection mainly occurred between small image divisions, leading to miss-detections. To address this issue, the authors increased the sliding window twice in horizontal and vertical directions during testing. Based on this, the performance of the proposed method is shown in Table 8, where the results demonstrate an average accuracy of 92.38 percent among the 50 images. The minimum accuracy is 89.22 percent, while the maximum accuracy is 95.58 percent. The accuracy standard deviation is 1.62 percent, indicating the stability of the proposed method for detecting and segmenting cracks on bridge decks.

As mentioned in the paper, creating ground truth for crack segmentation is time-consuming and requires careful attention. To compare the crack segmentation results, the authors obtained 100 images from the crack segmentation step to create ground truth. Table 9 illustrates the performance of the proposed U-Net model on testing data. The impressive outcome shows that the trained model is good and stable, making it a reliable choice for applying to other datasets. Fig. 13a shows a good result in comparison between the predicted crack width and

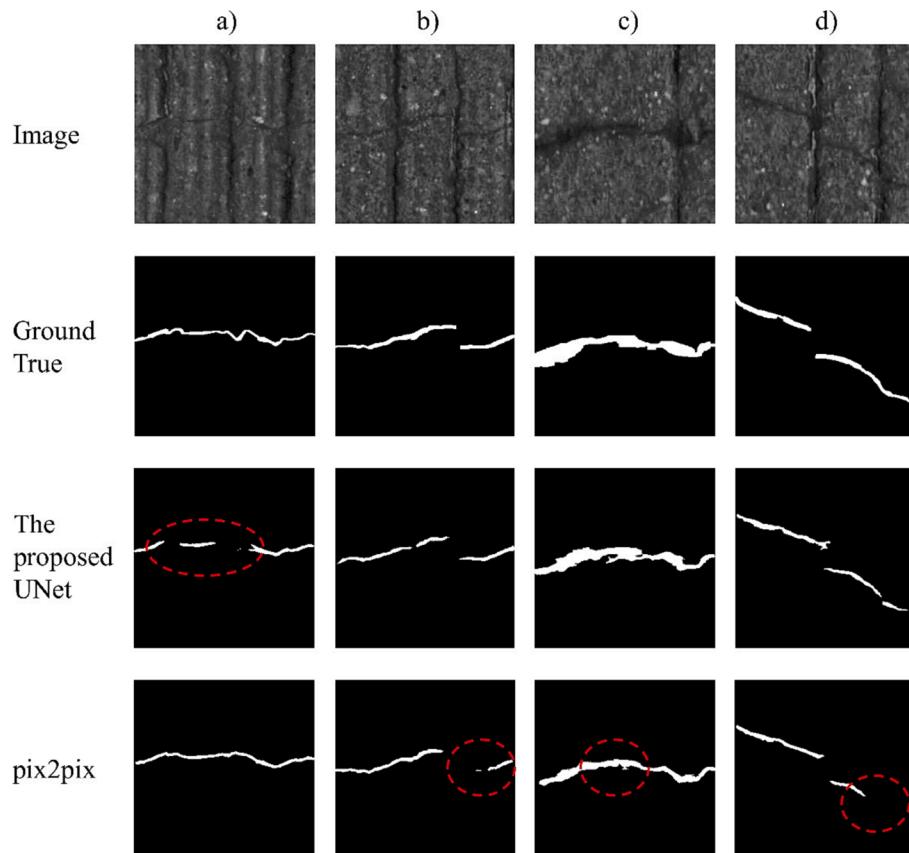


Fig. 10. Comparison of model's performance in crack segmentation.

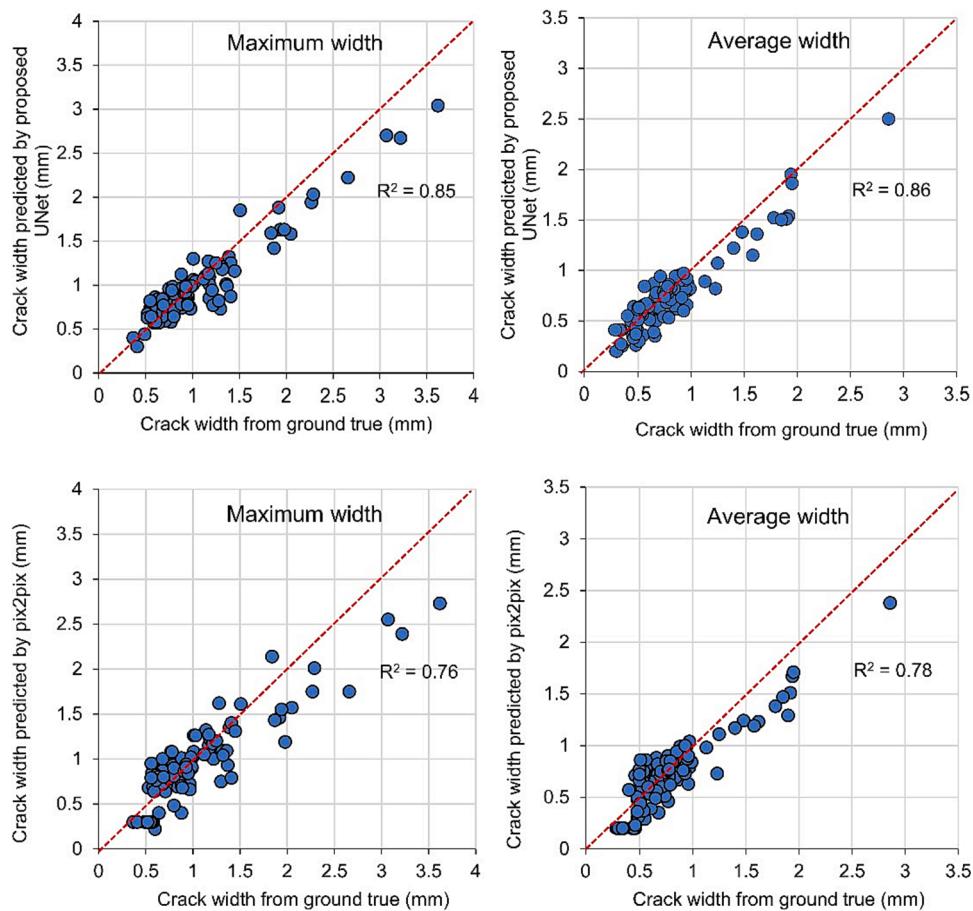


Fig. 11. Performance of segmentation models on crack width.

Table 7

Performance of YOLOv7 on testing data (300 images with the size of $1,024 \times 1,024$ pixels).

Model	MAP@ 0.5	MAP@ 0.5–0.95
YOLOv7	0.745	0.408

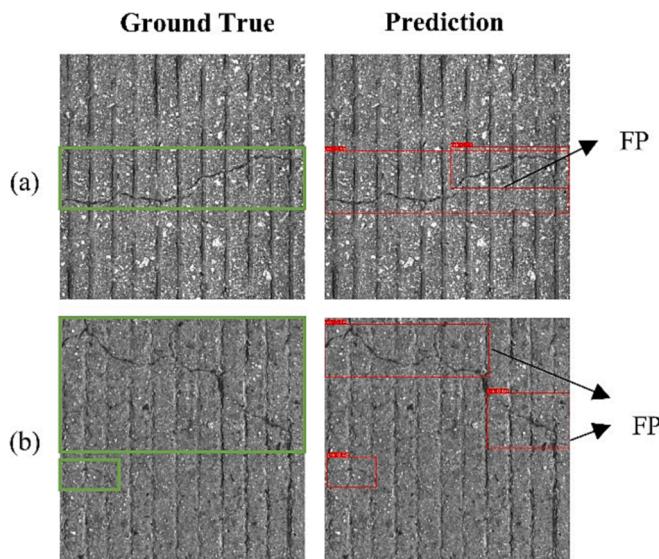


Fig. 12. Sample of crack detection by YOLOv7.

Table 8

Performance of the proposed method on testing data (50 original images with the size of $18,000 \times 10,000$ pixels).

Metric	Length (m)	Absolute Error (m)	Accuracy
Mean	25.08	1.42	92.38
Standard Deviation	4.74	0.52	1.62
Minimum	16.85	0.55	89.22
Maximum	38.71	3.40	95.58

Table 9

Performance of the proposed U-Net on testing data.

Model	F1-score	Precision	Recall	IoU
Proposed U-Net	0.781	0.846	0.725	0.640

ground true crack width with an R^2 value of 0.87. Moreover, the requirement of this project is to classify crack width to propose an appropriate rehabilitation method for the bridge deck. Therefore, the authors also check the performance of the proposed method in terms of crack width classification. The result shows that the proposed method can classify crack width with an average accuracy of 91 percent, as shown in Fig. 13b. Additionally, as shown in Fig. 13b, false classification only occurred between two nearby classes. This error is mainly due to the confusion in classification at the boundaries of each class.

The findings demonstrate the effectiveness of the proposed method in accurately identifying and analyzing cracks on bridge decks, which can contribute to enhancing the safety and longevity of infrastructure systems. Additionally, the crack length of detected cracks can be precisely determined from the segmented images using interpolate function

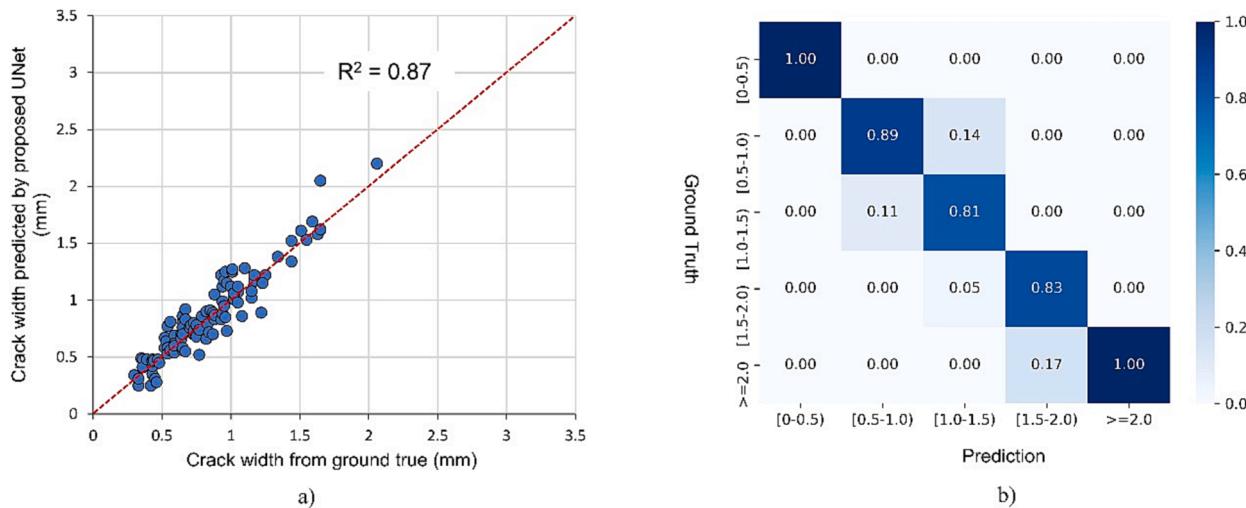


Fig. 13. The performance of the proposed U-Net in predicting crack width.

(interp1d) in the SciPy library [53]. Finally, Fig. 14 shows an example of the output result using the proposed method to detect crack length and classify crack width on the bridge deck surface.

9. Conclusion and recommendation

This study proposed a novel two-stage method for detecting and segmenting cracks on the bridge deck surfaces. Based on the conducted research, the following findings are withdrawn:

- Five different object detection networks were trained and tested to evaluate their performance in detecting cracks on the complex background of the bridge deck surface. The investigation revealed that YOLOv7 outperformed the other four networks in terms of accuracy and analysis speed. Specifically, YOLOv7 showed an increase

in MAP@0.5 and MAP(0.5:0.95) compared to Faster RCNN-ResNet101 and RetinaNet-ResNet101. Moreover, YOLOv7 was 8.2 and 3 times faster than Faster RCNN-ResNet101 and RetinaNet-ResNet101, respectively. Therefore, YOLOv7 is the preferred option for detecting cracks on the bridge deck surface due to its superior performance and efficiency.

- YOLOv7 has the ability to detect cracks on the higher-resolution image with the size of $2,048 \times 2,048$ pixels. With this, the analysis time can be reduced by four times compared to using the same training image size of $1,024 \times 1,024$ pixels. It can be concluded that YOLOv7 is very appropriate for detecting cracks on high-resolution bridge deck surfaces image.
- The proposed U-Net model outperforms pix2pix and the other U-Net for crack segmentation on the bridge deck in terms of precision, recall, F1-score, and IoU. In the validation dataset, the proposed U-

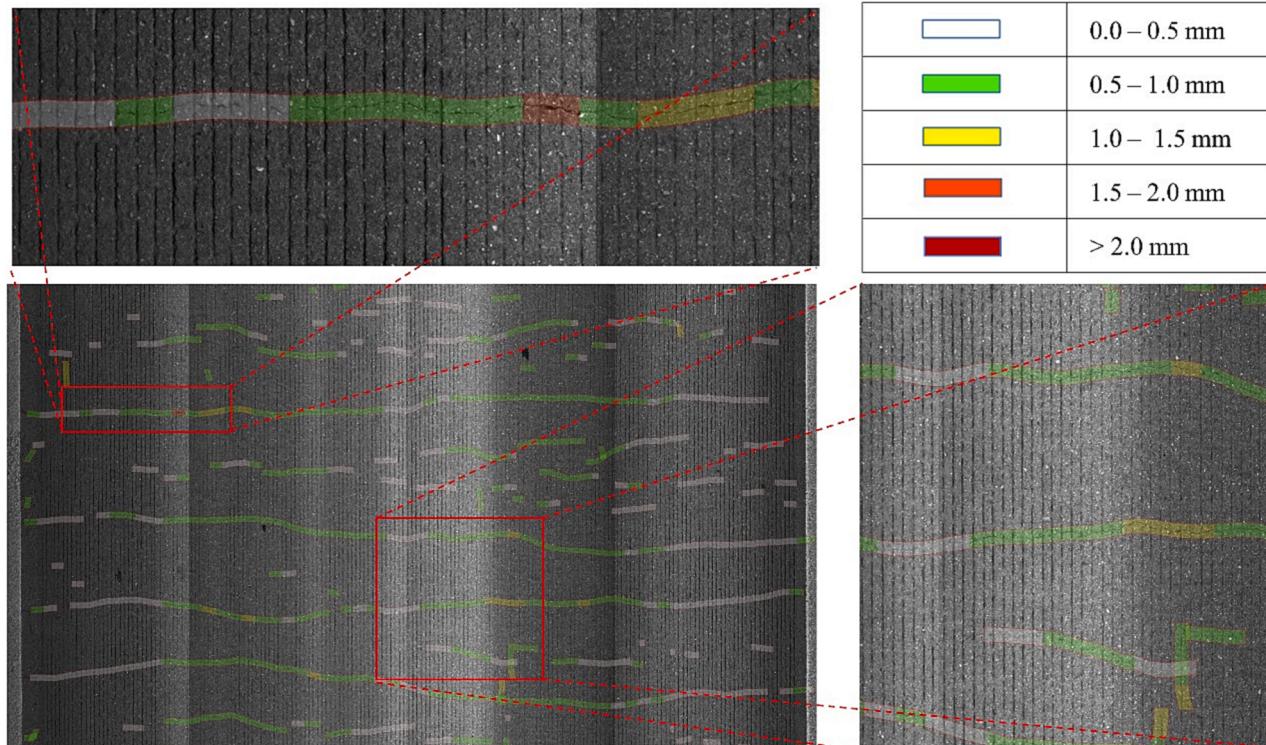


Fig. 14. Crack length detection and width classification on the bridge deck surface.

- Net network with the optimum parameters delivers precision, recall, F1-score, and IoU of 0.835, 0.729, 0.778, and 0.637, respectively. The optimum U-Net model also achieved R^2 of 0.85 and 0.86 for maximum and average crack width, respectively. These findings suggest that the use of average crack width is deemed more appropriate for representing cracks on bridge deck surfaces.
- Additionally, the proposed U-Net model exhibits impressive performance on the testing data, with precision, recall, F1-score, and IoU reaching 0.846, 0.725, 0.781, and 0.64, respectively. Furthermore, the optimized U-Net model achieves an R^2 value of 0.87 when comparing the predicted crack width to the ground truth data.
 - The YOLOv7 integrated with the proposed U-Net model was successfully applied to two bridges in South Korea. The application shows that the proposed method has the ability to detect and segment cracks on the bridge deck to ensure both fast and high accuracy.

Although the proposed approach offers several benefits, it does have some limitations:

- The current optimal segmentation model lacks the ability to accurately segment very small cracks. To address this limitation, it is essential for future research to focus on exploring novel techniques or enhancing existing segmentation networks. By incorporating these advancements, significant improvements can be achieved in the approach's segmentation results, particularly in detecting minute cracks.
- Dividing the high-resolution image into smaller patches presents a potential solution to the problem. However, it comes with a limitation, as more sliding windows are required to avoid missing crack detections, leading to increased analysis time. In the future, it is crucial to explore methods that can handle high-resolution images without the need to divide them into smaller patches.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant No. 23RS-2019-KA152342, RS-2023-00243421) and Sejong University.

References

- [1] L. Zhang, F. Yang, Y. D. Zhang, Y. J. Zhu, Road crack detection using deep convolutional neural network, in: 2016 IEEE international conference on image processing (ICIP), IEEE, 2016, pp. 3708–3712.
- [2] N. T. H. Nguyen, T. H. Le, S. Perry, T. T. Nguyen, Pavement crack detection using convolutional neural network, in: Proceedings of the 9th International Symposium on Information and Communication Technology, 2018, pp. 251–256.
- [3] B. Li, K.C. Wang, A. Zhang, E. Yang, G. Wang, Automatic classification of pavement crack using deep convolutional neural network, *Int. J. Pavement Eng.* 21 (4) (2020) 457–463.
- [4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L.I. Fei-Fei, Imagenet large scale visual recognition challenge, *Int. J. Comput. Vis.* 115 (3) (2015) 211–252.
- [5] F. Liu, J. Liu, L. Wang, Deep learning and infrared thermography for asphalt pavement crack severity classification, *Autom. Constr.* 140 (2022), 104383.
- [6] F. Liu, J. Liu, L. Wang, Asphalt pavement fatigue crack severity classification by infrared thermography and deep learning, *Autom. Constr.* 143 (2022), 104575.
- [7] T.S. Tran, V.P. Tran, H.J. Lee, J.M. Flores, V.P. Le, A two-step sequential automated crack detection and severity classification process for asphalt pavements, *Int. J. Pavement Eng.* 23 (6) (2022) 2019–2033.
- [8] S.D. Nguyen, T.S. Tran, V.P. Tran, H.J. Lee, M.J. Piran, V.P. Le, Deep learning-based crack detection: a survey, *Int. J. Pavement Res. Technol.* (2022) 1–25.
- [9] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, *Adv. Neural Inf. Proces. Syst.* 28 (2015).
- [10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, 2016, pp. 21–37.
- [11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
- [12] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, *arXiv preprint arXiv:2207.02696* (2022).
- [13] W. Wang, B. Wu, S. Yang, Z. Wang, Road damage detection and classification with faster r-cnn, in: 2018 IEEE international conference on big data (Big data), IEEE, 2018, pp. 5220–5223.
- [14] C. Gou, B. Peng, T. Li, Z. Gao, Pavement crack detection based on the improved faster-rcnn, in: In: 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), IEEE, 2019, pp. 962–967.
- [15] J. Tang, Y. Mao, J. Wang, L. Wang, Multi-task enhanced dam crack image detection based on faster r-cnn, in: 2019 IEEE 4th international conference on image, vision and computing (ICIVC), IEEE, 2019, pp. 336–340.
- [16] R. Li, J. Yu, F. Li, R. Yang, Y. Wang, Z. Peng, Automatic bridge crack detection using unmanned aerial vehicle and faster r-cnn, *Constr. Build. Mater.* 362 (2023), 129659.
- [17] M. Nie, C. Wang, Pavement crack detection based on yolo v3, in: 2019 2nd international conference on safety produce informatization (IICSPI), IEEE, 2019, pp. 327–330.
- [18] V.P. Tran, T.S. Tran, H.J. Lee, K.D. Kim, J. Baek, T.T. Nguyen, One stage detector (retinanet)-based crack detection for asphalt pavements considering pavement distresses and surface objects, *Journal of Civil Struct. Health Monit.* 11 (1) (2021) 205–222.
- [19] Y. Du, N. Pan, Z. Xu, F. Deng, Y. Shen, H. Kang, Pavement distress detection and classification based on yolo network, *Int. J. Pavement Eng.* 22 (13) (2021) 1659–1672.
- [20] Y. Zhang, J. Huang, F. Cai, On bridge surface crack detection based on an improved yolo v3 algorithm, *IFAC-PapersOnLine* 53 (2) (2020) 8205–8210.
- [21] S. Zhu, X. Xia, Q. Zhang, K. Belloulat, An image segmentation algorithm in image processing based on threshold segmentation, in: 2007 third international IEEE conference on signal-image technologies and internet-based system, IEEE, 2007, pp. 673–678.
- [22] C. Liu, C.-S. Tang, B. Shi, W.-B. Suo, Automatic quantification of crack patterns by image processing, *Comput. Geosci.* 57 (2013) 77–80.
- [23] A.M.A. Talab, Z. Huang, F. Xi, L. HaiMing, Detection crack in image using otsu method and multiple filtering in image processing techniques, *Optik* 127 (3) (2016) 1030–1033.
- [24] S.L. Lau, E.K. Chong, X. Yang, X. Wang, Automated pavement crack segmentation using u-net-based convolutional neural network, *IEEE, Access* 8 (2020) 114892–114899.
- [25] W. Wang, C. Su, Automatic concrete crack segmentation model based on transformer, *Autom. Constr.* 139 (2022), 104275.
- [26] F. Liu, J. Liu, L. Wang, Asphalt pavement crack detection based on convolutional neural network and infrared thermography, *IEEE Trans. Intell. Transp. Syst.* 23 (11) (2022) 22145–22155.
- [27] N.H.T. Nguyen, S. Perry, D. Bone, H.T. Le, T.T. Nguyen, Two-stage convolutional neural network for road crack detection and segmentation, *Expert Syst. Appl.* 186 (2021), 115718.
- [28] Y. Jiang, D. Pang, C. Li, Y. Yu, Y. Cao, Two-step deep learning approach for pavement crack damage detection and segmentation, *Int. J. Pavement Eng.* (2022) 1–14.
- [29] K. Chen, G. Reichard, X. Xu, A. Akanmu, Automated crack segmentation in close-range building facade inspection images using deep learning techniques, *Journal of Building Engineering* 43 (2021), 102913.
- [30] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
- [31] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1125–1134.
- [32] R. S. Lim, H. M. La, Z. Shan, W. Sheng, Developing a crack inspection robot for bridge maintenance, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 6288–6293.
- [33] R. Adhikari, O. Mosehli, A. Bagchi, Image-based retrieval of concrete crack properties for bridge inspection, *Autom. Constr.* 39 (2014) 180–194.
- [34] P. Prasanna, K.J. Dana, N. Gucunski, B.B. Basily, H.M. La, R.S. Lim, H. Parvardeh, Automated crack detection on concrete bridges, *IEEE Trans. Autom. Sci. Eng.* 13 (2) (2014) 591–599.
- [35] H.M. La, N. Gucunski, K. Dana, S.-H. Kee, Development of an autonomous bridge deck inspection robotic system, *J. Field Rob.* 34 (8) (2017) 1489–1504.

- [36] Q. Zhang, K. Barri, S.K. Babanajad, A.H. Alavi, Real-time detection of cracks on concrete bridge decks using deep learning in the frequency domain, *Engineering* 7 (12) (2021) 1786–1796.
- [37] Tzutalin, 'labelimg', free software: Mit license. (2015).
- [38] Matlab, 2018. version 9.7.0.1190202 (r2019b). the mathworks inc.
- [39] A. Garcia-Garcia, S. Orts-Escalano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, J. Garcia-Rodriguez, A survey on deep learning techniques for image and video semantic segmentation, *Appl. Soft Comput.* 70 (2018) 41–65.
- [40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch (2017).
- [41] <https://github.com/luolzaaa/faster-rcnn-pytorch>.
- [42] <https://github.com/yhenon/pytorch-retinanet>.
- [43] <https://github.com/wongkinyiu/yolov7>.
- [44] R. Hahnloser, H.S. Seung, Permitted and forbidden sets in symmetric threshold-linear networks, *Adv. Neural Inf. Proces. Syst.* 13 (2000).
- [45] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, *Commun. ACM* 60 (6) (2017) 84–90.
- [46] J. Xu, Z. Li, B. Du, M. Zhang, J. Liu, Reluplex made more practical: Leaky ReLU. In: 2020 IEEE Symposium on Computers and Communications (ISCC), 2020.
- [47] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (elus) (2015).
- [48] P. Ramachandran, B. Zoph, Q. V. Le, Searching for activation functions (2017).
- [49] Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 31, Curran Associates, Inc., 2018.
- [50] J. M. Joyce, Kullback-leibler divergence, in: *International Encyclopedia of Statistical Science*, Springer Berlin Heidelberg, 2011, pp. 720–722.
- [51] J. Kugelman, J. Allman, S.A. Read, S.J. Vincent, J. Tong, M. Kalloniatis, F.K. Chen, M.J. Collins, D. Alonso-Caneiro, A comparison of deep learning u-net architectures for posterior segment OCT retinal layer segmentation, *Sci. Rep.* 12 (1) (Sep. 2022).
- [52] F. Liu, L. Wang, UNet-based model for crack detection integrating visual explanations, *Constr. Build. Mater.* 322 (2022), 126265.
- [53] P. Fabian, Scikit-learn: Machine learning in python, *J. Mach. Learn. Res.* 12 (2011) 2825.