Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.024405
Article

Tech Science Press

Automated Crack Detection via Semantic Segmentation Approaches Using Advanced U-Net Architecture

Honggeun Ji[1,2], Jina Kim[3], Syjung Hwang[4] and Eunil Park[1,4,*]

[1]Department of Applied Artificial Intelligence, Sungkyunkwan University, Seoul, 03063, Korea
[2]AI Research Team, Scalawox, Seoul, 08589, Korea
[3]R&D Team, Raon Data, Seoul, 03073, Korea
[4]Department of Interaction Science, Sungkyunkwan University, Seoul, 03063, Korea
*Corresponding Author: Eunil Park. Email: eunilpark@skku.edu
Received: 15 October 2021; Accepted: 05 January 2022

Abstract: Cracks affect the robustness and adaptability of various infrastructures, including buildings, bridge piers, pavement, and pipelines. Therefore, the robustness and the reliability of automated crack detection are essential. In this study, we conducted image segmentation using various crack datasets by applying the advanced architecture of U-Net. First, we collected and integrated crack datasets from prior studies, including the cracks in buildings and pavements. For effective localization and detection of cracks, we used U-Net-based neural networks, ResU-Net, VGGU-Net, and EfficientU-Net. The models were evaluated by the five-fold cross-validation using several evaluation metrics including mean pixel accuracy (MPA), mean intersection over union (MIoU), and confusion matrix. The results of the integrated dataset showed that ResU-Net (68.47%) achieves the highest MIoU with a relatively low number of parameters compared to VGGU-Net (67.71%) and EfficientU-Net (68.07%). In addition to the performance, ResU-Net showed the lowest test runtime, 40 milliseconds per single image, and the highest true positive rate of 45.00% in the pixel-wise recognition test. As the models were trained and validated with diverse surfaces, the proposed approach can be used as a pre-trained model in the task with relatively few data sources. Furthermore, both practical and managerial implications are discussed herein.

Keywords: Crack detection; semantic segmentation; deep learning; fully convolutional network; U-Net

1 Introduction

In industrial infrastructures, cracks are repeatedly generated due to various factors such as corrosion, poor construction, and/or loading [1]. Over the past few years, several locations have suffered notable physical damage owing to cracks occurring in roads and transport pipes [2]. To prevent potential accidents caused by such cracks, significant efforts have been devoted toward detecting abnormalities using numerous different technical and administrative approaches. Traditionally, when well-trained experts conduct comprehensive investigations of specific structures, such as pipes, they utilize various

datasets collected via diverse pathways from the structures (e.g., sensors) to compute and estimate the exact location or extent of specific cracks, as well as to determine the need for structural reinforcements [3,4]. However, this approach has significant limitations. In general, a majority of traditional approaches require labor-intensive and time-consuming tasks [5–7].

To address this issue, a number of remarkable techniques based on computer vision and machine learning theories have been introduced for realizing efficient crack detection and classification tasks [8–10]. In-line with this trend, a majority of previous studies have proposed the use of deep-learning-based architectures with a convolutional neural network (CNN) to detect specific cracks [11,12]. For instance, Fan et al. [13] and Zhang et al. [12] introduced CNN-based models with supervised learning approaches for detecting cracks in pavements and roads, respectively. In addition to deep learning models applying a CNN for road/pavement crack detection, several researchers have explored semantic segmentation approaches with models to more accurately classify whether each pixel is included in the crack dimensions [14]. Zhang et al. [2] proposed a vision-oriented crack detection system with a deep semantic segmentation network. Additionally, Lee et al. [15] argued that a crack detection network involving image segmentation approaches can be helpful for robust crack detection tasks.

Based on the findings of previous studies, the current study aims to explore whether comprehensive image segmentation architectures can be employed to detect cracks in diverse surfaces, regardless of both objects and environments. With this aim, we collected integrated datasets including diverse crack images. Subsequently, three deep learning models based on the U-Net network architecture were employed to localize and detect crack positions. The remainder of this paper is organized as follows: Section 2 introduces several examples of crack detection via image-based deep learning techniques. Sections 3 and 4 describe the study methodologies and performances of the employed deep learning models, respectively. Both the implications and limitations of this study, as well as the scope for future research, are presented in Section 5.

2 Related Studies

Deep learning approaches using image datasets constitute an important research topic in the fields of computer science and artificial intelligence. Several previous studies have employed both digital images and deep learning architectures to solve specific industrial problems [16]. This indicates that deep learning models can be more effective and efficient when employing image datasets, as compared with traditional machine learning models [17]. Deep learning-based representation approaches employing a neural network architecture have been widely used in image classification, object detection, image captioning, and semantic segmentation [18]. In addition to deep learning approaches, the collection of large-scale image datasets (e.g., ImageNet [19]) has enabled several researchers and practitioners to successfully complete numerous large-scale visual recognition tasks.

In general, a CNN is organized in three dimensions (width × height × depth) with three main layers (convolutional, pooling, and connected layers). Based on this architecture, a number of CNN-based image processing models have been introduced to utilize image datasets and extract valuable features [20]. Similar to other deep learning models applied to image datasets, the detection of cracks in specific structures (so-called crack detection) is one of the adaptation domains using deep learning models. Thus, several prior studies have focused on crack detection using deep learning networks; this approach is classified into three techniques: image classification-, bounding-box-, and image segmentation-based techniques [21]. Tab. 1 summarizes prior studies.

The traditional application of a CNN architecture involves an image classification task to determine whether an image includes a distinguishable crack. Using this approach and task, the classifier is generally organized into an input layer, convolutional layers (for extracting specific characteristics), pooling layers (for reducing the number of dimensions), and connected layers. Wang et al. [22] used a CNN architecture and component analyses to classify pavement cracks, achieving accuracies of 97.2%, 97.6%, and 90.1% for longitudinal, diagonal, and alligator cracks, respectively. Chen et al. [23] introduced a deep learning model with both CNN and naïve Bayes-based fusion schemes for detecting cracks on the surface of a nuclear power plant, achieving a hit rate of 98.3% per frame.

One of the notable approaches using a CNN architecture for crack detection is a bounding box task. This task aims to specify certain sliding windows for crack locations. Cha et al. [24] proposed a visual inspection method with a faster region-based CNN (Faster R-CNN) for detecting cracks in concrete and steel surfaces. Based on 2,366 collected images, the proposed method yielded precision rates of 90.6%, 83.4%, 82.1%, 98.1%, and 84.7% when detecting five different types of damage in concrete and steel surfaces, respectively.

Although deep learning models for image classification and bounding box tasks have achieved excellent performance, the models for these tasks have notable limitations in terms of their practical applicability. As one of the most representative limitations, the models for these tasks are unable to precisely indicate crack regions. In addition, large crack shapes occur on diverse surfaces, which needs to be addressed and considered in crack detection tasks [2]. To address the abovementioned limitations, a semantic segmentation task that focuses on both crack detection and image classification, while also considering each pixel, is required. In general, a fully convolutional network (FCN) is employed for pixel-wise classification. For instance, Yang et al. [25] utilized the FCN architecture for detecting diversely categorized cracks with an accuracy of 97.96%. In addition, Dung et al. [1] proposed an FCN-based model using a VGG network as the backbone. To successfully detect concrete cracks, the implemented FCN with pre-trained VGG-16 models was employed, achieving a precision ratio of 90%. Zhang et al. [2] employed both an FCN and a dilated convolutional layer to observe concrete cracks; this model, validated on 600 images, resulted in a mean pixel accuracy (MPA) of 92.55% and a mean intersection over union (MIoU) of 86.05%. Pan et al. [26] proposed SCHNet, a segmentation model for concrete crack based on VGG-19 with self-attention mechanism. With the data augmentation method, SCHNet achieved MIoU of 85.31%.

Encoder-decoder architectures have been employed to achieve better performance than general deep learning models [27]. Thus, researchers have focused on U-Net, which is an FCN with an encoder-decoder architecture, for image segmentation [28]. Ji et al. [29] applied the DeepLabV3+ with the encoder-decoder architecture to asphalt pavements. The model, trained for 250 images, recorded 0.8342 MIoU for validation data and 0.7331 for an external validation set with 80 images. Liu et al. [30] proposed a detection and segmentation method using ResNet-34 based U-Net with a YOLO detector. The proposed two-step approaches scored 90.58%, 95.75% f-1 score in detection and segmentation. In general, U-Net can be employed to present more accurate image locations by delivering feature matrices from the encoder to the corresponding decoder. Therefore, this study employs U-Net-based approaches to explore whether comprehensive image segmentation architectures can be employed to detect cracks in diverse surfaces, regardless of both objects and environments.

3 Methods

In this study, to explore crack outlines, three U-Net-based models designed for semantic segmentation tasks were employed. Fig. 1 provides an overview of the proposed architecture. Each U-Net model employed was trained using the datasets of crack images and subsequently validated using test datasets.

Table 1: Summary of prior studies on crack detection using deep learning models (FWIoU: Frequency Weighted Intersection over Union)

Sources	Method	Dataset	Results
Wang et al. [22]	CNN and principal component analysis	30,000 pavement crack images	Correct rate: 97.2% (longitudinal crack), 97.6% (transverse crack), 90.1% (alligator crack)
Chen et al. [23]	CNN and Naïve Bayes-based fusion model	147,344 crack images, 149,460 non-crack images	Hit rate: 98.3%
Cha et al. [24]	Faster R-CNN	2,366 crack images	Mean average precision: 87.8%
Yang et al. [25]	FCN	More than 800 crack images	Accuracy: 97.96%, Precision: 81.73%, Recall: 78.97%, F1 score: 79.95%
Dung et al. [1]	VGG network, FCN with VGG backbone	40,000 (classification), 600 crack images (segmentation)	Accuracy: 99.9% (classification), Average precision 89.3% (segmentation)
Zhang et al. [2]	FCN with dilated convolution	600 crack images	Pixel accuracy: 96.84%, Mean pixel accuracy: 92.55%, MIoU: 86.05%, FWIoU: 94.22%
Pan et al. [26]	SCHNet (Spatial-Channel Hierarchical Network)	11,000 pavement and concrete images	MIoU: 85.31%
Ji et al. [29]	DeepLabv3+	300 pavement crack images	MIoU: 73.31%
Liu et al. [30]	ResNet based U-Net with YOLO detector	7,104 pavement crack images	F1 score: 90.58% (detection), 95.75% (segmentation)

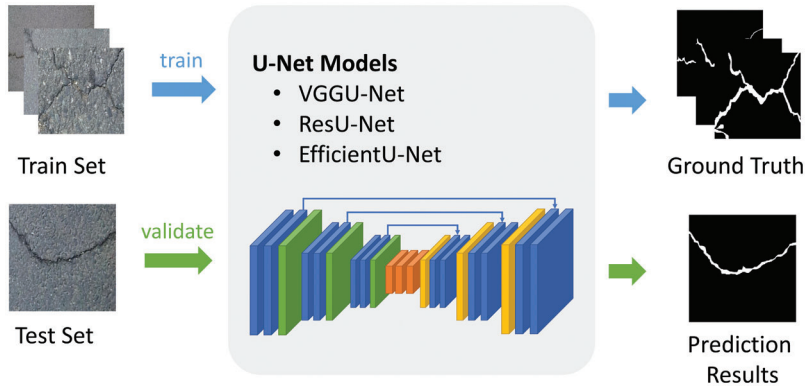


Figure 1: Overview of the proposed architecture

3.1 Dataset

We employed crack image datasets of diverse surfaces such as pavements or concrete[1]. This dataset includes more than 11,200 RGB and annotated masked crack images [12,31–35]. We also used the additional road-crack image dataset [1], which is organized into 11,449 images in total (pavement: 4650, concrete: 6,799). All the images were then resized to 224 × 224, and each pixel was normalized to between zero and 1 (Fig. 2).

[1] https://github.com/khanhha/crack_segmentation

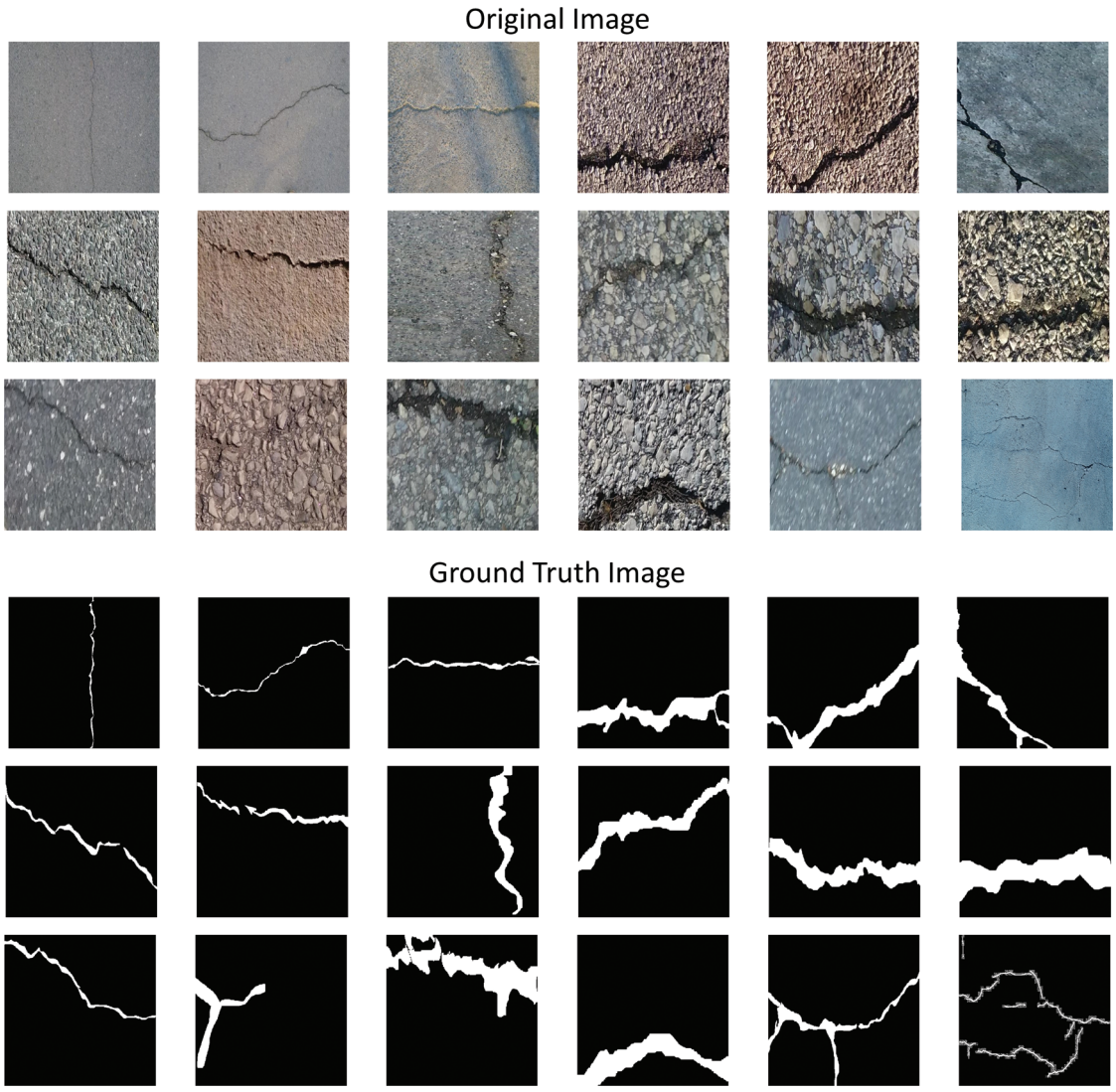


Figure 2: Examples of original and ground truth images from combined dataset

3.2 Semantic Segmentation Models

To detect cracks in diverse materials using a large-scale dataset, we employed three U-Net based models: VGGU-Net, ResU-Net [36], and EfficientU-Net [37]. Moreover, Tab. 2 shows the number of parameters and layers in this study.

Table 2: Comparison of parameters and layers of each model

Model	Number of parameters	Number of layers
VGGU-Net	3,903,489	39
ResU-Net	4,723,057	95
EfficientU-Net	11,472,361	443

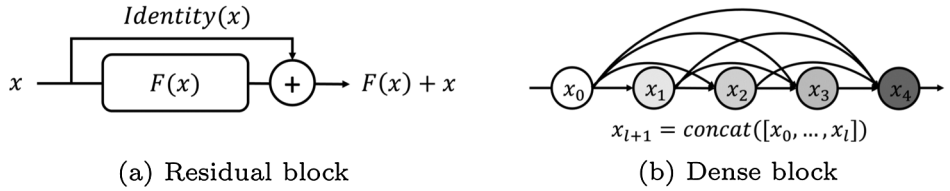
3.2.1 VGGU-Net

VGGU-Net was proposed by Simonyan et al. [37]. This model is organized using U-Net architecture and a VGG network as the encoder. As a feature extraction component, the VGG network reduces a set of high-dimensional features in raw images to low-dimensional features with multiple convolutional and pooling layers. In general, the extracted features in the VGG network are linked to a fully connected layer. Thereafter, the output of the layer is estimated and obtained via the activation function.

However, in this study, the feature extraction component of the VGG network, i.e., the multiple convolutional and pooling layers, were used. Subsequently, the decoding components, including the de-convolutional, convolutional, and up-sampling layers, were added. The architecture of the proposed VGGU-Net is presented in Fig. 4a.

3.2.2 ResU-Net

ResU-Net, which includes a residual block to exclude issues regarding training degradation in deeper hidden layers [38], was employed [36]. The residual block was based on the assumption that, for the input x, it is more efficient and beneficial to optimize residual mapping issues, $F(x) := H(x) - x$, as compared to optimizing the original mapping, $H(x)$. As presented in Fig. 3a, the operation in the residual block is represented as $F(x) + x$ (*shortcut connection*). The residual block in ResU-Net is organized as two convolutional layers with 2 kernels of 3×3 with batch normalization in the main flow, and a single convolutional layer with 1 kernel of 1×1 in *shortcut connection*. Fig. 4b depicts the model architecture.



(a) Residual block (b) Dense block

Figure 3: Residual and dense blocks

3.2.3 EfficientU-Net

EfficientU-Net, which is an advanced model of UNet++ [26], and EfficientNet B4 [39] as the backbone encoder are applied. UNet++ is a refined architecture of U-Net, achieved by updating the *skip connection* with a dense block of DenseNet [40]. Moreover, DenseNet is connected among all the layers (Fig. 3b). Thus, the vanishing gradient problem was addressed using the model with dense blocks.

Fig. 4c presents the model architecture of EfficientU-Net. In particular, the architecture of EfficientU-Net is organized by the enhanced *skip connection* (indicated in green blocks) between the encoding component as a contracting path and the decoding component as an expansive path (indicated in black arrows). Specifically, the skip connections (indicated by green and blue) integrate the feature maps of all previous blocks. An equation of the layer connection rule is described in the following Eq. (1), where H indicates the operation of block X, x stands for the output of block X, u denotes the up-sampling, and [ ] represents a concatenating operation. Moreover, i and j refer to the vertical and horizontal orders of blocks. Each block X mainly consists of two $3 \times$ convolution layers with batch normalization and a leaky ReLU activation function.

$$\mathrm{x}^{i,j} = \begin{cases} H(x^{i-1,j}), & \text{if } j = 0 \\ H([[x^{i,k}]_{k=0}^{j-1}, \; u(x^{i+1,j-1})]), & \text{if } j > 0 \end{cases} \tag{1}$$

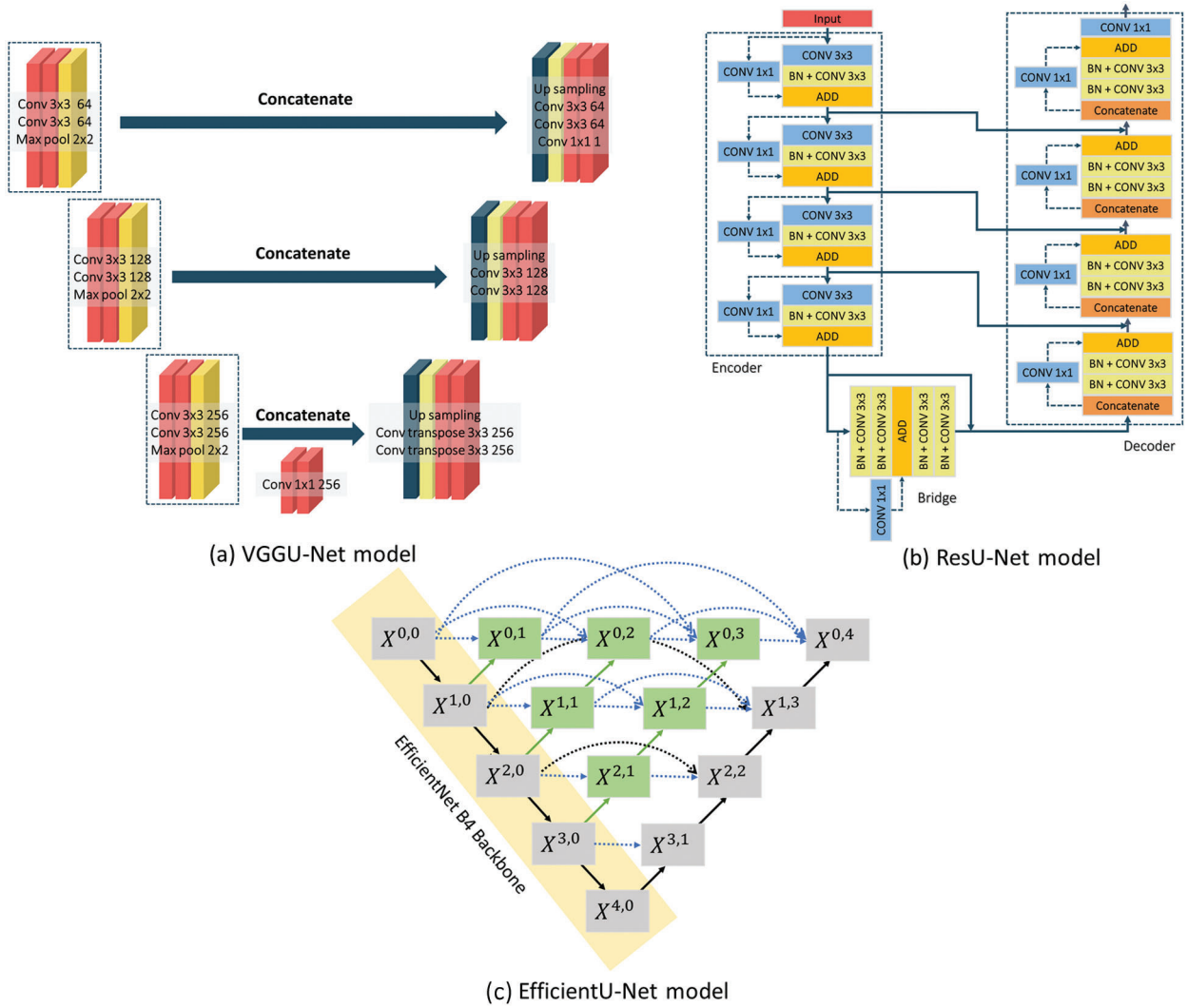


Figure 4: Models employed in this study. (a) It represents the structure of VGGU-Net, which utilizes the VGG network as an encoder. (b) It represents the structure of ResU-Net with a residual block in the encoder and decoder. (c) It represents the structure of EfficientU-Net, which comprises an EfficientNet encoder, a decoder, and an enhanced skip connection

3.3 Loss Function and Optimization

We employed a binary cross entropy (BCE) as a loss function, which is widely used for pixel-wise binary classification (Eq. (2)). Here, θ is the model parameter, and n, i and j are the pixel locations. In addition, N, W, and H denote the batch size, width, and height of the input, respectively; $y_{nij} \in \{0, 1\}$ represents the presence of the crack, and $\hat{y}$ denotes the probability of the predicted class.

$$J(\theta) = -\frac{1}{N \cdot W \cdot H} \sum_{n=1}^{N} \sum_{i=1}^{W} \sum_{j=1}^{H} y_{nij} \log(\hat{y}_{nij}) + (1 - y_{nij}) \log(1 - \hat{y}_{nij}), \tag{2}$$

We used the adaptive moment estimation (Adam) optimization algorithm with a learning rate η of 0.001, β_1 of 0.9, and β_2 of 0.999 [41]. The batch size was set to 16.

3.4 Metrics

Two widely used evaluation metrics for semantic segmentation tasks were employed to evaluate the proposed models: MPA and MIoU [42]. MPA is the mean of the ratio of pixels correctly classified for each class (Eq. (3)), where n_c is the number of classes, t_i is the total number of pixels in a specific class i, n_{ii} is the number of correctly classified pixels, and n_{ji} is the number of pixels incorrectly classified.

$$MPA = \frac{1}{n_c}\sum_i \frac{n_{ii}}{t_i}, \tag{3}$$

In addition, MIoU is the mean of the overlapping area over the union between the prediction and ground truth images (Eq. (4)).

$$MIoU = \frac{1}{n_c}\sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}, \tag{4}$$

Also, typical evaluation metrics for classification tasks were utilized to evaluate pixel-wise crack recognition performance of the proposed models; confusion matrix, TPR (true positive rate) and FPR (false positive rate). The confusion matrix is a table layout that can visualize the performance of the model. Each row of the matrix represents a real class, and each column represents a prediction class. The confusion matrix consists of the number of *TN* (true negative), *FP* (false positive), *FN* (false negative), and *TP* (true positive). The TPR is used to measure the proportion of actual positives which are correctly identified (Eq. (5)).

$$TPR = \frac{TP}{TP + FN}, \tag{5}$$

The FPR is a measure of false positives for whole positive predictions (Eq. (6)).

$$FPR = \frac{FP}{FP + TN}, \tag{6}$$

The FNR is the measure that true positive is omitted by the test (Eq. (7)).

$$FNR = \frac{FN}{TP + FN}, \tag{7}$$

Also, the TNR is the ratio that the true negative is predicted to be negative (Eq. (8)).

$$TNR = \frac{TN}{FP + TN}, \tag{8}$$

3.5 Model Training

We applied five-fold cross-validation procedures to validate the performance and robustness of the proposed models. The following procedures were conducted [43]: First, the entire dataset was divided into five folds, four of which were used for the training set. The last fold was employed as the testing dataset. Thereafter, the testing datasets were changed to other folds that were not previously included as the testing datasets. The average results of the five evaluations were then computed. As the current study employed 11,449 images, each fold had 2290 images; the last fold included 2289 images. We used Python language to implement all U-Net-based models based on deep learning frameworks, Tensorflow and Keras. We trained the model on an Ubuntu 18.04 machine equipped with RTX2080Ti GPU.

4 Results and Discussion

Fig. 5 shows the results of the cross-validation procedures. ResU-Net achieved the highest convergence speed, followed by VGGU-Net and EfficientU-Net (Fig. 5a). Although the BCE loss of VGGU-Net was greater than that of EfficientU-Net during the early stages of the training procedures, the BCE (Binary Cross Entropy) loss rapidly decreased.

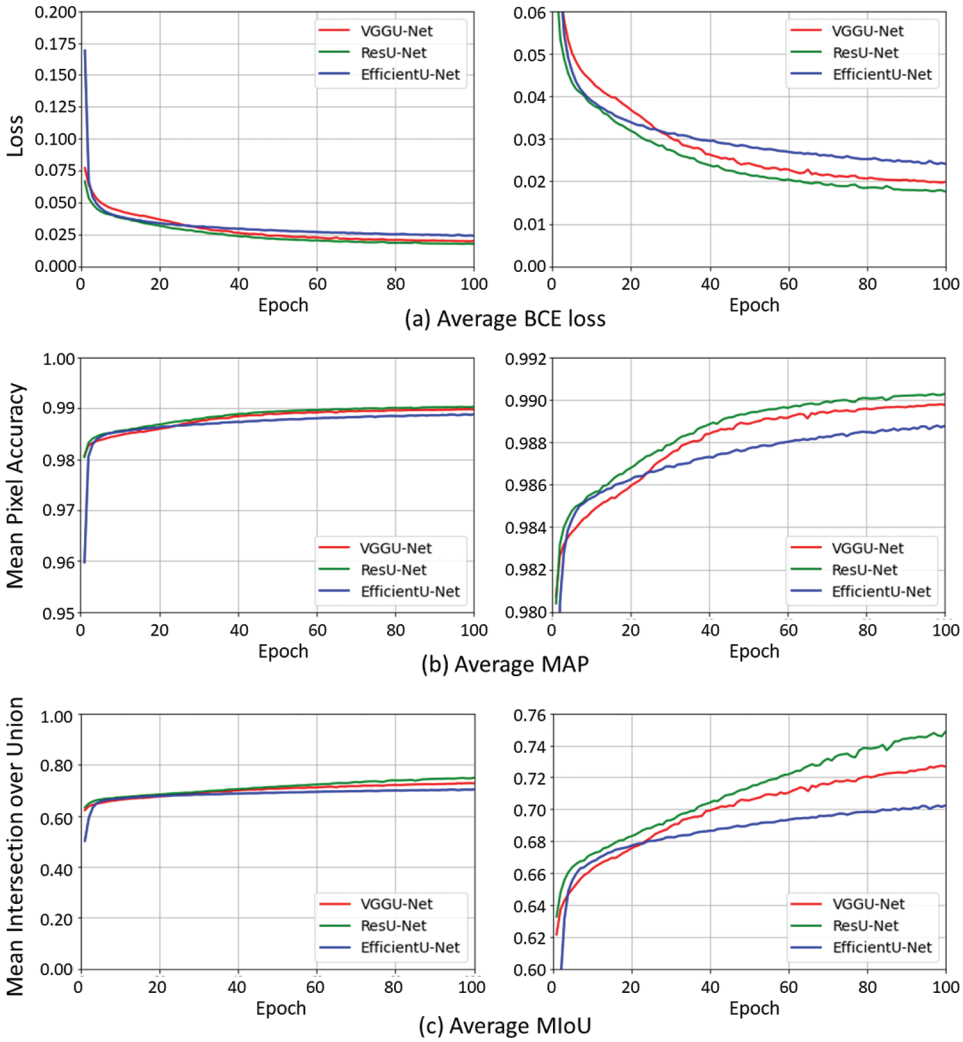


Figure 5: Performance of each model with respect to training epochs. Each graph was obtained by averaging the results of the five-fold cross-validation for the employed models. (a) It depicts the variations in BCE loss. (b) It shows the changes in MPA. (c) It shows the changes in MIoU

Fig. 5b shows the average MPA. ResU-Net demonstrated the highest accuracy (99.02%), followed by VGGU-Net (98.97%) and EfficientU-Net (98.87%). Although VGGU-Net showed the lowest training accuracy among the models during the early stage of training, after 30 epochs, it achieved accuracies higher than those of EfficientU-Net.

Fig. 5c shows the MIoU during training. The overall trend was similar to those of the BCE loss and MPA. ResU-Net showed the highest MIoU (74.86%), followed by VGGU-Net (72.67%) and EfficientU-Net

(70.24%). The difference in performance among the models was not significant at the beginning of training; however, as training progressed, significant differences were observed in the results of the models.

Tab. 3 shows the average results of the five-fold cross-validation procedures for concrete images, pavement images, and the integrated images. During testing for the all images, EfficientU-Net showed the lowest BCE loss (0.04), followed by VGGU-Net (0.06) and ResU-Net (0.07). Moreover, EfficientU-Net exhibited the highest MPA (98.55%), as compared with the other models (ResU-Net: 98.47%, VGGU-Net: 98.46%). However, ResU-Net achieved superior performance in terms of the MIoU (68.47%), as compared with the other models (EfficientU-Net: 68.07%; VGGU-Net: 67.72%).

Table 3: Average test performance determined via five-fold cross-validation

Dataset	# of images	Model	BCE loss	MPA (%)	MIoU (%)
Pavement images	4,650	VGGU-Net	0.1339	97.3183	64.9058
		ResU-Net	0.1565	97.3618	65.9357
		EfficientU-Net	0.0837	**97.5103**	**66.6365**
Concrete images	6,799	VGGU-Net	0.0224	99.2424	69.6318
		ResU-Net	0.0233	99.2396	**70.2162**
		EfficientU-Net	0.0168	**99.2666**	69.0531
All images concrete & pavement images	11,499	VGGU-Net	0.0676	98.4621	67.7175
		ResU-Net	0.0773	98.4777	**68.4787**
		EfficientU-Net	0.0422	**98.5541**	68.0750

In the case of 4,650 pavement images, the EfficientU-Net showed the highest performance, achieving 66.63% in MIoU and 97.51% in MPA, followed by ResUNet achieving 65.93%, 97.36% in terms of MIoU and MPA. However, for the concrete with 6,799 images, ResU-Net achieved 70.21% in MIoU, showing significantly higher performance than other models (VGGU-Net: 69.63%, EfficientU-Net: 69.05%). Meanwhile, we could confirm that there is no significant performance gap between the three models in MPA (VGGU-Net: 69.63%, ResU-Net: 99.23%, EfficientU-Net: 69.05%).

Fig. 6 depicts the original, ground truth, and predicted images. Overall, the results of the semantic segmentation tasks using ResU-Net and EfficientU-Net were similar. It indicates that both models can be employed to effectively detect cracks in various surfaces, whereas VGGU-Net can involve notable limitations in detecting cracks that are considerably long and thin. Moreover, VGGU-Net tends to be unsuitable for detecting cracks on diverse surfaces owing to the generalized issues of the results.

Tab. 4 shows the test runtime of each model. As ResU-Net is composed of the residual blocks, it requires 40.6250 milliseconds to process a single image, which is 1.23, 1.69 times faster than VGGU-Net (49.9998 ms) and EfficientU-Net (68.7599 ms).

Fig. 7 shows the confusion matrix calculated for each pixel of the ground truths and predicted images, and their TNR (true negative rate), FPR (false positive rate), FNR (false negative rate), and TPR (true positive rate). As shown in the first row of each confusion matrix, all models accurately predicted TN. It means that the non-crack pixels were appropriately classified as non-crack (VGGU-Net: 560,890,680, ResU-Net:560,484,708, EfficientU-Net: 561,072,993). Similarly, in TNR and FPR scores, there is no significant difference between the models. (VGGU-Net, 99.56%, 00.44%; ResU-Net, 99.49%, 00.51%; EfficientU-Net, 99.59%, 00.41%).

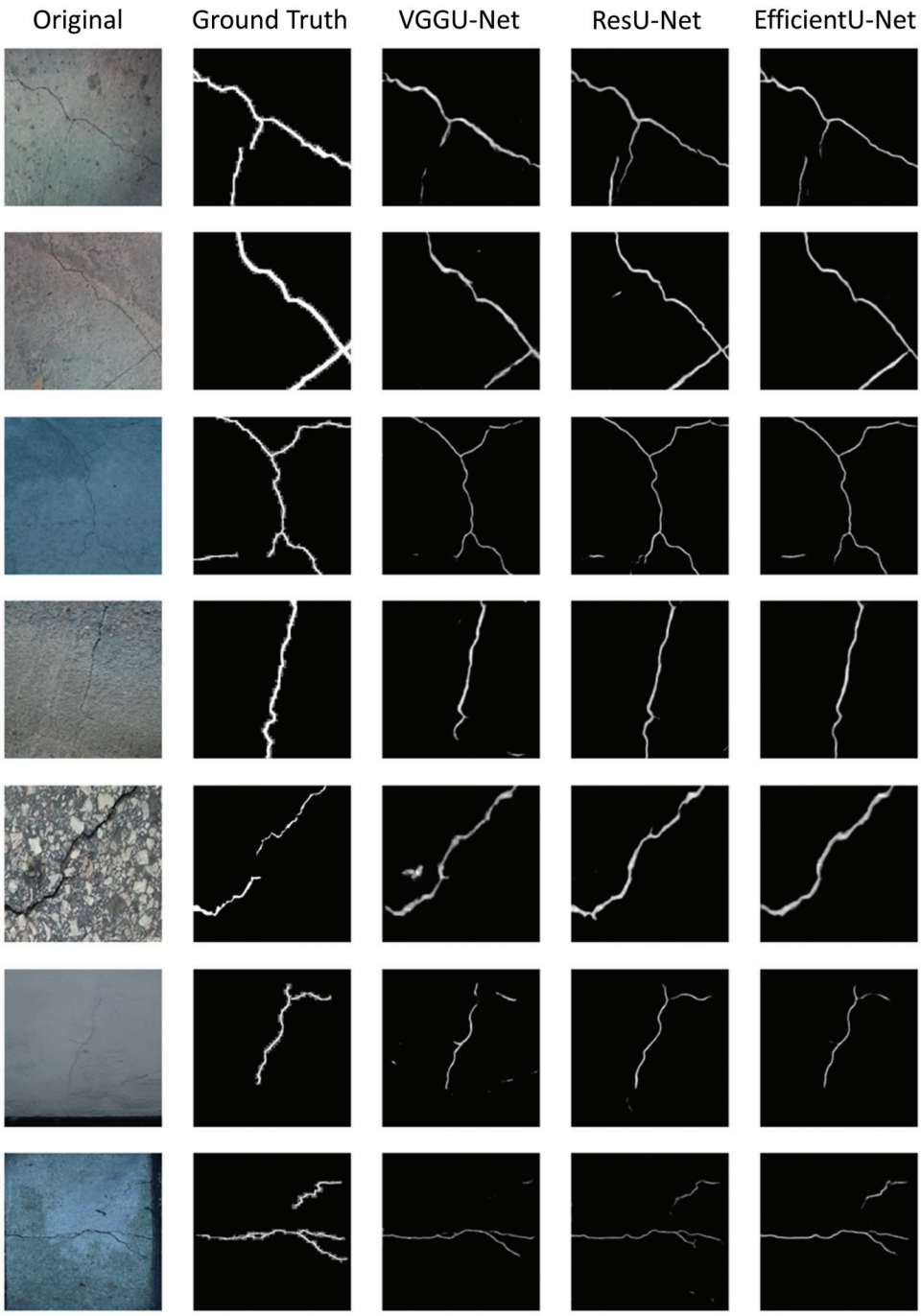


Figure 6: Segmentation results. The original image, ground truth image, and segmentation results were obtained by employing the last fold as the test set

Table 4: Test runtime of the employed models

Model	Test runtime (ms)
VGGU-Net	49.9998
ResU-Net	**40.6250**
EfficientU-Net	68.7599

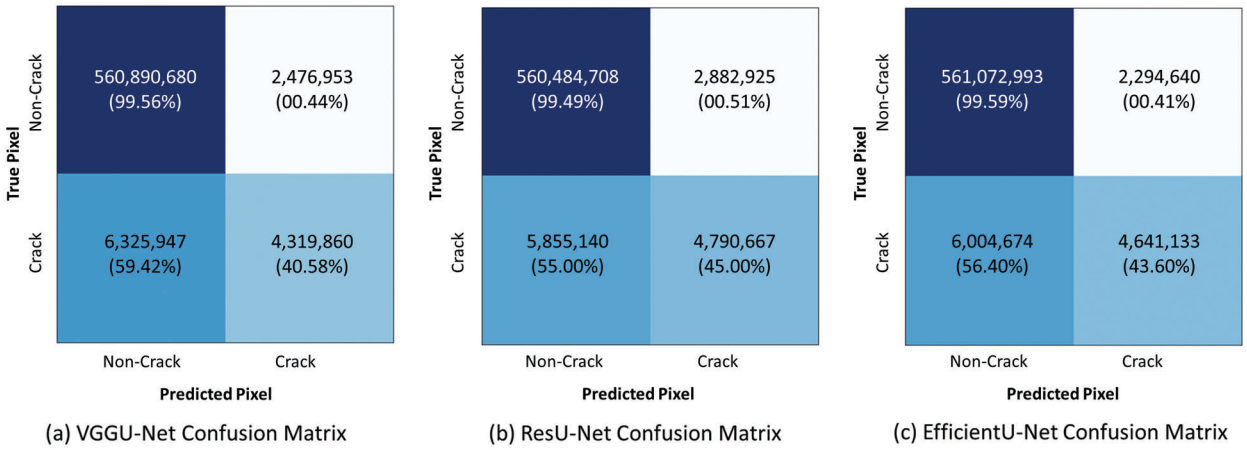


Figure 7: Confusion matrix for employed segmentation models (a) It represents the confusion matrix of VGGU-Net. (b) It indicates the confusion matrix of ResU-Net. (c) It shows the confusion matrix of EfficientU-Net

However, there is a huge performance gap in predicting crack pixels. As depicted in the second row of the confusion matrices, ResU-Net, which classified 4,790,667 and 5,855,140 pixels in terms of TP and FN, recognized the crack pixel accurately compared to other models. Also, ResU-Net achieved the highest TPR (45.00%) and the lowest FNR (55.00%) scores, which are crucial indicators in recognition of risk. It implies that ResU-Net is not only superior to other competing models in crack detection but also robust in misrecognition, i.e., judging crack pixels as non-crack.

EfficientU-Net yielded the highest MPA, as compared with the other U-Net-based models. In addition, ResU-Net presented the highest MIoU and TPR, as compared with the other models. Based on the experimental results, it was confirmed that ResU-Net is a more efficient and effective deep learning model for crack-related image segmentation tasks, compared with the other U-Net-based models.

To sum up the aforementioned results, the technical and experimental merits of using ResU-Net for various tasks can be detailed as follows:

- The number of required parameters for ResU-Net is approximately 51% less than that for EfficientU-Net.
- The processing speed of ResU-Net is 1.69 times faster than that of EfficientU-Net.
- The convergence time of ResU-Net is faster than that of the other U-Net-based models.
- The true positive rate of ResU-Net showed the highest rate, 45.00%.

5 Conclusion

The detection of cracks on specific surfaces is essential for efficiently maintaining and managing different types of structures. In this study, we integrated several datasets consisting of diverse surfaces such as concrete walls and pavements, in order to enhance the generalization ability of crack detection models. Three U-Net-based deep learning models, VGGU-Net, ResU-Net, and EfficientU-Net, were validated with five-fold cross-validation using several evaluation metrics including MPA, MIoU, and confusion matrix. Based on the findings of the current study, several practical and managerial implications are presented. As the proposed models are trained, implemented, and tested using datasets that are organized based on images of diverse surfaces, the models can be applied to different surfaces and structures, without being limited to specific environments such as concrete structures or pavements.

It implies that the proposed models are more flexible and comprehensive than other models, when addressing more general issues than crack detection. For example, the proposed model can be used as a pretrained model in the domain such as the pipeline transportation [16], where limited data is available [44].

Despite the contributions of this study, a few limitations remain unaddressed. First, this study did not consider the characteristics of surfaces (e.g., material) and cracks (e.g., depth and extent). Thus, the performance of the proposed models can be further improved if characteristics of specific surface images and cracks are extracted and reflected as key features in the models. Second, recurrent-oriented schemes were not employed when designing the advanced U-Net models. Future research should focus on addressing and resolving these issues.

Funding Statement: This research was supported by KICT (Korea Institute of Civil Engineering and Building Technology) Grant Number [KICT 2021-13]. This research was also supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport [21ATOG-C161932-01].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. V. Dung and L. D. Anh, "Autonomous concrete crack detection using deep fully convolutional neural network," *Automation in Construction*, vol. 99, pp. 52–58, 2019.

[2] J. Zhang, C. Lu, J. Wang, L. Wang and X. -G. Yue, "Concrete cracks detection based on FCN with dilated convolution," *Applied Sciences*, vol. 9, pp. 2686, 2019.

[3] M. Goktepe, Y. Ege, N. Bayri and S. Atalay, "Non-destructive crack detection using GMI sensor," *Physica Status Solidi*, vol. 1, pp. 3436–3439, 2004.

[4] T. Dogaru and S. T. Smith, "Edge crack detection using a giant magneto resistance based eddy current sensor," *Nondestructive Testing and Evaluation*, vol. 16, pp. 31–53, 2000.

[5] H. Cheng, J. -R. Chen, C. Glazier and Y. Hu, "Novel approach to pavement cracking detection based on fuzzy set theory," *Journal of Computing in Civil Engineering*, vol. 13, pp. 270–280, 1999.

[6] H. Cheng, J. Wang, Y. Hu, C. Glazier, X. Shi *et al.,* "Novel approach to pavement cracking detection based on neural network," *Transportation Research Record*, vol. 1764, pp. 119–127, 2001.

[7] P. Subirats, J. Dumoulin, V. Legeay and D. Barba, "Automation of pavement surface crack detection using the continuous wavelet transform," in *Proc. of 2006 Int. Conf. on Image Processing*, New York, NY, USA, pp. 3037–3040, 2006.

[8] S. Dorafshan, R. J. Thomas and M. Maguire, "Benchmarking image processing algorithms for unmanned aerial system-assisted crack detection in concrete structures," *Infrastructures*, vol. 4, pp. 19, 2019.

[9] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Advanced Engineering Informatics*, vol. 25, pp. 507–515, 2011.

[10] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[11] Y. -J. Cha, W. Choi and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, pp. 361–378, 2017.

[12] L. Zhang, F. Yang, Y. D. Zhang and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *Proc. of 2016 IEEE Int. Conf. on Image Processing*, New York, NY, pp. 3708–3712, 2016.

[13] Z. Fan, Y. Wu, J. Lu and W. Li, "Automatic pavement crack detection based on structured prediction with the convolutional neural network," 2018. [Online]. Available: https://arxiv.org/abs/1802.02208.

[14] H. Li, R. Zhao and X. Wang, "Highly efficient forward and backward propagation of convolutional neural networks for pixel wise classification," 2014. [Online]. Available: https://arxiv.org/abs/1412.4526.

[15] D. Lee, J. Kim and D. Lee, "Robust concrete crack detection using deep learning-based semantic segmentation," *International Journal of Aeronautical and Space Sciences*, vol. 20, pp. 287–299, 2019.

[16] W. Fang, L. Ding, H. Luo and P. E. Love, "Falls from heights: A computer vision-based approach for safety harness detection," *Automation in Construction*, vol. 91, pp. 53–61, 2018.

[17] D. Shen, G. Wu and H. -I. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering*, vol. 19, pp. 221–248, 2017.

[18] J. Ker, L. Wang, J. Rao and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access*, vol. 6, pp. 9375–9389. 2017.

[19] J. Deng, W. Dong, R. Socher, L. -J. Li, K. Li *et al.,* "Imagenet: A large-scale hierarchical image database," in *Proc. of the 2009 IEEE Conf. on Computer Vision and Pattern Recognition*, New York, IEEE, pp. 248–255, 2009.

[20] A. Voulodimos, N. Doulamis, A. Doulamis and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational Intelligence and Neuroscience*, vol. 2018, pp. 7068349, 2018.

[21] A. Mohan and S. Poobal, "Crack detection using image processing: A critical review and analysis," *Alexandria Engineering Journal*, vol. 57, pp. 787–798, 2018.

[22] X. Wang and Z. Hu, "Grid-based pavement crack analysis using deep learning," in *Proc. of the 2017 4th Int. Conf. on Transportation Information and Safety*, New York, NY, pp. 917–924, 2017.

[23] F. -C. Chen and M. R. Jahanshahi, "Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve Bayes data fusion," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 4392–4400, 2017.

[24] Y. -J. Cha, W. Choi, G. Suh, S. Mahmoudkhani and O. Büyüköztürk, "Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, pp. 731–747, 2018.

[25] X. Yang, H. Li, Y. Yu, X. Luo, T. Huang *et al.,* "Automatic pixel-level crack detection and measurement using fully convolutional network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, pp. 1090–1109, 2018.

[26] Y. Pan, G. Zhang and L. Zhang, "A Spatial-channel hierarchical deep learning network for pixel-level automated crack detection," *Automation in Construction*, vol. 119, pp. 103357, 2020.

[27] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Cham, Switzerland: Springer, pp. 3–11, 2018.

[28] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation, in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*, Cham, Switzerland: Springer, pp. 234–241, 2015.

[29] A. Ji, X. Xue, Y. Wang, X. Luo, and W. Xue, "An integrated approach to automatic pixel-level crack detection and quantification of asphalt pavement," *Automation in Construction*, vol. 114, pp. 103176, 2020.

[30] J. Liu, X. Yang, S. Lau, X. Wang, S. Luo *et al.,* "Automated pavement crack detection and segmentation based on two-step convolutional neural network," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, pp. 1291–1305, 2020.

[31] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei *et al.,* "Feature pyramid and hierarchical boosting network for pavement crack detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 1525–1535, 2019.

[32] M. Eisenbach, R. Stricker, D. Seichter, K. Amende, K. Debes *et al.,* "How to get pavement distress detection ready for deep learning? a systematic approach," in *Proc. of the Int. Joint Conf. on Neural Networks*, New York, NY, IEEE, pp. 2039–2047, 2017.

[33] Y. Shi, L. Cui, Z. Qi, F. Meng and Z. Chen, "Automatic road crack detection using random structured forests," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 3434–3445, 2016.

[34] R. Amhaz, S. Chambon, J. Idier and V. Baltazart, "Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2718–2729, 2016.

[35] Q. Zou, Y. Cao, Q. Li, Q. Mao and S. Wang, "Cracktree: Automatic crack detection from pavement images," *Pattern Recognition Letters*, vol. 33, pp. 227–238, 2012.

[36] Z. Zhang, Q. Liu and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, pp. 749–753, 2018.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014. [Online]. Available: https://arxiv.org/abs/1409.1556.

[38] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, New York, NY, IEEE, pp. 770–778, 2016.

[39] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1905.11946.

[40] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, New York, NY, IEEE, pp. 4700–4708, 2017.

[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980.

[42] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for se-mantic segmentation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, New York, NY, IEEE, pp. 3431–3440, 2015.

[43] J. D. Rodriguez, A. Perez and J. A. Lozano, "Sensitivity analysis of k-fold cross validation in prediction error estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 569–575, 2009.

[44] R. Jafari, S. Razvarz, A. Gegov and B. Vatchova, "Deep learning for pipeline damage detection: An overview of the concepts and a survey of the state-of-the-art," in *Proc. of the 2020 IEEE 10th Int. Conf. on Intelligent Systems (IS)*, New York, NY, IEEE, pp. 178–182, 2020.