buildings

MDPI

Article

Deep-Learning- and Unmanned Aerial Vehicle-Based Structural Crack Detection in Concrete

Tao Jin [1,2,3], Wen Zhang [2], Chunlai Chen [1,*], Bin Chen [1,4], Yizhou Zhuang [2] and He Zhang [1,3]

[1] School of Engineering, Hangzhou City University, Hangzhou 310015, China; jintao@zucc.edu.cn (T.J.); zjuzhanghe@zju.edu.cn (H.Z.)
[2] College of Civil Engineering, Zhejiang University of Technology, Hangzhou 310014, China; wenzhang@zjut.edu.cn (W.Z.)
[3] Department of Civil Engineering, Zhejiang University, Hangzhou 310058, China
[4] Zhejiang Engineering Research Center of Intelligent Urban Infrastructure, Hangzhou 310014, China
* Correspondence: chencl@zucc.edu.cn

Abstract: Deep-learning- and unmanned aerial vehicle (UAV)-based methods facilitate structural crack detection for tall structures. However, contemporary datasets are generally established using images taken with handheld or vehicle-mounted cameras. Thus, these images might be different from those taken by UAVs in terms of resolution and lighting conditions. Considering the difficulty and complexity of establishing a crack image dataset, making full use of the current datasets can help reduce the shortage of UAV-based crack image datasets. Therefore, the performance evaluation of existing crack image datasets in training deep neural networks (DNNs) for crack detection in UAV images is essential. In this study, four DNNs were trained with different architectures based on a publicly available dataset and tested using a small UAV-based crack image dataset with 648 +pixel-wise annotated images. These DNNs were first tested using the four indices of precision, recall, mIoU, and F1, and image tests were also conducted for intuitive comparison. Moreover, a field experiment was carried out to verify the performance of the trained DNNs in detecting cracks from raw UAV structural images. The results indicate that the existing dataset can be useful to train DNNs for crack detection from UAV images; the TransUNet achieved the best performance in detecting all kinds of structural cracks.

Keywords: deep learning; crack detection; semantic segmentation; dataset; UAV

Citation: Jin, T.; Zhang, W.; Chen, C.; Chen, B.; Zhuang, Y.; Zhang, H. Deep-Learning- and Unmanned Aerial Vehicle-Based Structural Crack Detection in Concrete. *Buildings* **2023**, *13*, 3114. https://doi.org/10.3390/buildings13123114

Academic Editor: Elena Ferretti

Received: 18 October 2023
Revised: 4 December 2023
Accepted: 12 December 2023
Published: 15 December 2023

1. Introduction

Bridges are vital elements in transportation, and the harsh environments in which they are located usually accelerate the aging of such structures. Many bridges lose their operational functionality before reaching their desired service life; thus, inspection processes for determining the status of bridge conditions in service are a guarantee of operational safety. The main evaluation and detection methods for bridge cracks are manual visual observation and bridge inspection vehicle detection. In most cases, these traditional methods are feasible. However, they do have some deficiencies, such as cumbersome operations and high rates of missed detection. Moreover, it is difficult to reach the surfaces of high structures, such as bridge towers and the arch structures of large-span bridges. Thanks to the rapid development of deep learning, the field of detecting structural conditions has witnessed significant advancements in recent years [1]. This technology has opened up innovative pathways for the detection of structural cracks, which can replace the manual identification process. Many DNNs, such as U-Net [2] and DeepLab [3], have emerged, and are widely used for identifying cracks. More and more variants are also being developed for research and industrial purposes [4–7].

Meanwhile, UAVs mounted with high-resolution cameras can reach the surfaces of tall structures easily for crack image acquisition. As a new method for obtaining crack

images, UAVs are widely used [8,9]. Smaoui et al. [10] proposed a method for planning paths for autonomous scanning tasks of concrete structures based on onboard cameras, collecting and transmitting images to estimate the extent of damage caused by cracks. Ngo et al. [11] used drones to collect images of cracks and defects on the surface of concrete bridges, developing an effective method for assessing the structural integrity of concrete bridges. Compared with traditional cameras that require manual photography, UAVs have the advantage of being lightweight and convenient, and they feature automatic obstacle avoidance capabilities which enable them to reach previously inaccessible areas. This makes them an effective tool for image acquisition in challenging locations that human beings cannot easily access. UAVs are thus good platforms for taking pictures of cracks in tall bridges, thereby solving the problem of imaging cracks in extreme terrains. The capability and quality of capturing images through UAVs were verified and confirmed by Zhong et al. [12] using an IMETRUM non-contact detector. By constructing a bridge model and conducting a UAV test flight, Peng et al. [13] determined the optimal imaging distance for different switching conditions. They used this information to plan a UAV cruise route and subsequently conducted a real bridge flight to detect cracks on the deck. The resulting UAV crack dataset achieved a recognition accuracy rate of 90% on the R-FCN network model, demonstrating its ability to precisely identify cracks. Using UAV technology, Li et al. [14] developed a DenxiDeepCrack crack-recognition model. Using a dataset of UCrack 11 road cracks for model training, captured by UAVs in a vertical view, the model achieved an impressive AP value of 0.632. Research on crack image recognition based on UAVs has seen significant advancements, with notable progress in recent years. Due to the limited dataset of UAV-captured images, research on the semantic segmentation of cracks using this technology is still scarce.

While UAVs can obtain crack images for DNNs to detect cracks, training qualified DNNs with high performance is vital for the final results. Although the architecture of the DNNs is quite important for crack detection results, the crack image datasets that were used to train the DNNs have more fundamental influences. The dataset is the foundation of model training that provides the crack and background samples for the DNNs to learn high-dimensional features for identifying crack regions in images [15–18]. The numbers, diversity, and quality of the crack image dataset greatly influence or determine the crack detection performance. Many scholars have created high-quality crack image datasets. Kim and Cho. [19] used a web scraper to obtain 42,000 crack images from the Internet; Deng et al. [20] cropped captured concrete bridge cracks into small images of 500 $\times$ 375 and formed a bounding box with a crack dataset of 5009 images. Ye et al. [21] established the Bridge Crack Library with 7805 crack images and 3195 non-crack images, which includes a pixel-level annotation dataset of steel, concrete, and masonry bridge cracks. However, the crack image datasets mentioned above are mainly based on handheld camera images, whereas UAV-based crack image datasets are relatively scarce. The image conditions such as the resolution, lighting condition, and background features might be different. Considering the difficulty and complexity of establishing a crack image dataset, it is necessary to study the existing datasets in training DNNs for the detection of cracks in UAV images.

This paper evaluates the existing crack image datasets in training DNNs to detect cracks from UAV images. An open-sourced crack image dataset based on images from manual inspection was used to train four different DNN models, i.e., the U-Net, the DeepLab v3 (MobileNet v3), the DeepLab v3 (ResNet50), and the TransUNet. A UAV-based image dataset containing 648 images with pixel-wise annotation was established for testing purposes. In addition, raw UAV images from the bridge tower of a cable-stayed bridge were obtained in a field test to examine and illustrate the performance of the trained DNNs. The four DNN models with different architectures were used to compare the crack detection results in different models. Meanwhile, k-fold training based on the U-Net was applied to study the influence of the different compositions of crack images.

2. Introduction of U-Net, DeepLab v3, and the Crack Datasets

2.1. Related Network Introduction

Deep learning networks are crucial in identifying cracks as they are capable of inputting raw data into various layers of a network, interpreting essential information, and integrating semantic details obtained by both high- and low-level networks. This enables the network to accurately decipher the data and acquire specific information relevant to detecting cracks. In 1998, LeCun et al. [22] proposed the LeNet network, which was the first application of convolutional neural networks in image processing. In the subsequent developmental procedure, processing tasks of images were divided into three categories: image classification, object detection, and semantic segmentation.

Many image classification networks, such as AlexNet [23], GoogLeNet [24], VGG [25], and MobileNet [26], have been proposed for specific conditions and are often used as the backbone of subsequent networks. Inception [27] and Xception [28] were derived from VGG. Inception-ResNet [29], ResNeXt [30], and DenseNet [31] were derived from ResNet. On the basis of MobileNet, ShuffleNet [32], EfficientNet [33], and ConvNeXt [34] were derived. The target detection task started from the RCNN [35] with VGG as the backbone proposed in 2014, and Fast R-CNN [36], Faster R-CNN [37], SSD [38], and YOLO series [39–41] were derived. To effectively detect cracks, it is not sufficient to focus solely on research aimed at object detection or object classification. Realization of the pixel-level segmentation of crack morphology can be more helpful to evaluate the cracks for subsequent research. Thus, semantic segmentation networks such as DeepLab, FCN [42], U-Net, Mask R-CNN [43], and U2Net [44] are more suitable. In this study, the classic semantic segmentation model U-Net and the relatively new model TransUNet, as well as two versions of DeepLab v3, were selected as the training networks to train and verify the existing dataset.

2.2. Selected Models

Proposed in 2015, U-Net adopts an Encode–Decode structure that is made up of four down-sampling layers and four up-sampling layers, as illustrated in Figure 1. The input images have sizes of 256 × 256 × 3 (RGB). The U-Net undergoes two 3 × 3 convolutional layers and ReLu layers as initial steps. The down-sampling stage utilizes a 2 × 2 MaxPool layer, reducing the feature layer size to 128 × 128 × 64 before proceeding to the subsequent convolutional layer. Upon the completion of four down-sampling stages, the up1 up-sampling layer receives a 16 × 16 × 512 feature layer input. The feature layer size is then enlarged to 32 × 32 × 512 using bilinear interpolation and concatenated with the corresponding channel feature layer from the down-sampling stage. The concatenated feature layer is passed through a 3 × 3 convolutional layer before repeating the same process for subsequent up-sampling layers. Upon completion of the up-sampling stage, the final 1 × 1 convolutional layer produces a feature size of 256 × 256 × 64, leading to an output of 256 × 256 × 1 after further processing with a 1 × 1 convolutional layer.

There have been significant innovations in the formal structure of DeepLab v3. It introduces the concept of dilated convolution, increases the receptive field of the network, and can obtain more crack features on the basis of retaining the data information to the greatest extent. It exerts little pressure on the operating equipment and is relatively easy to realize. In rapid iterations of network updates, it still demonstrates an excellent training effect. DeepLab v3 improves the ASPP part on the basis of DeepLab v2, introduces the multi-grid aspect, and removes the post-processing structure of CRFs that have little impact on the training effect. In detail, the network structure of DeepLab v3 is shown in Figure 2. After inputting the image data, they pass through four residual structures, Block1~Block4, which derive from the 2~5 residual structures of the ResNet network. In Block4, the original ordinary convolution is replaced by dilated convolution. The introduction of dilated convolution increases the network receptive field and does not change the height and width of the feature map, which is more complete than directly entering the MaxPooling layer. After the above steps, the ASPP structure is input; the ASPP structure is shown in Figure 3. Compared with DeepLab v2, in v3, the BN layer and the ReLu layer are involved

in four parallel dilated convolutional layers. The four convolutional layers are composed of one 1 × 1 and three 3 × 3 dilated convolutional layers. Then, a global pooling layer is added to fuse the upper and lower feature information, and BI is the bilinear interpolation for adjusting the output size.

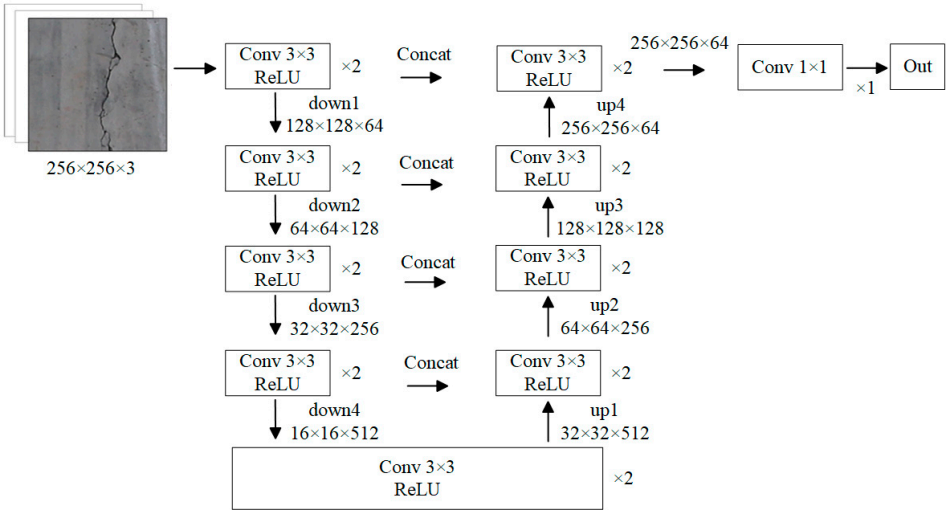


Figure 1. U-Net network structure diagram.

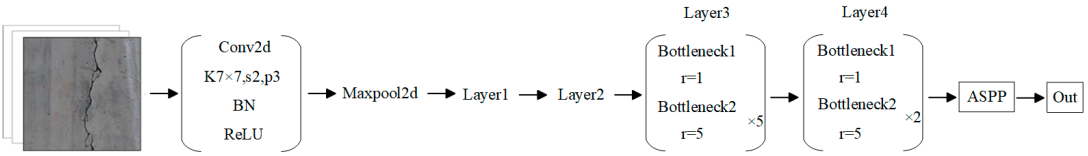


Figure 2. DeepLab v3 network structure diagram.

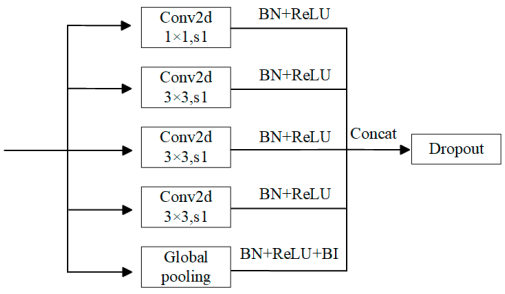


Figure 3. ASPP structure diagram.

2.3. Evaluation Indices

The main segmentation indicators of DeepLab v3 and U-Net are IoU, mIoU, and recall [45,46]. IoU is an indicator that evaluates the degree of overlap between the predicted and true boxes, which is the ratio of the overlap area between the predicted and true boxes to the union area. The mIoU calculates the average IoU for each class. The practical significance of recall is how many objects are detected among all objects. The calculation formula of IoU is shown in (1). The mIoU is the average of the sum of each type of IoU. The calculation formula of mIoU is shown in (2).

$$\mathrm{IoU} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN} + \mathrm{FP}) \quad (1)$$

$$\mathrm{mIoU} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN} + \mathrm{FP}) \quad (2)$$

In the formula, FP is the number of predicted positive results and negative actual results; FN is the number of predicted results that are negative and actually positive; and

TP is the number for which both the predicted result and the actual result are positive, i.e., the part included in both FP and FN. Detailed identification is shown in Table 1. Recall is the target ratio predicted by the model to be the correct sample in the correct sample, which is used to measure the recall ability of the model to the correct sample. The calculation formula is shown in (3).

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FP}) \quad (3)$$

Table 1. Identification of index elements.

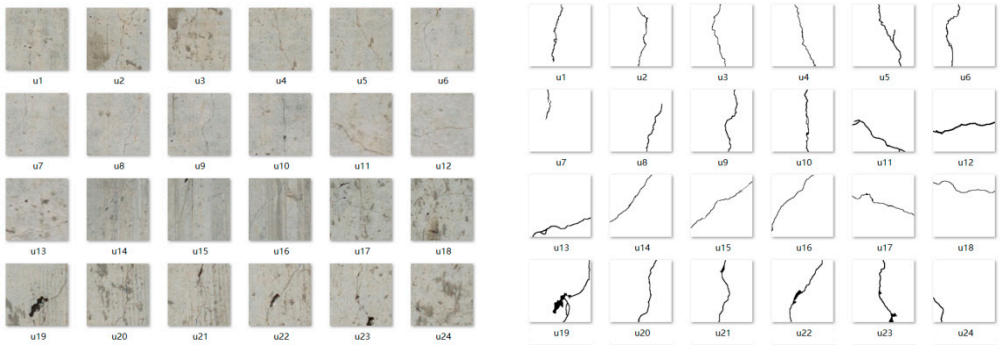
The Right Situation	Model Prediction	
	True	**False**
True	True prediction (TP)	False negative (FN)
False	False prediction (FP)	True negative (TN)

2.4. Data Collection

An open-source Bridge Crack Library was used as the main training dataset, which contained 7805 crack images and 3195 non-crack images; each image has a manually labeled PNG image. The testing dataset used bridge crack pictures taken by UAVs for cropping and labeling. The crack pixel area accounts for a small proportion of the whole image and the pixel labeling is relatively fine; therefore, in this study, Adobe Photoshop software (Version 21.0.1) was used to draw and mark the cracks. The UAV dataset utilized both original images of cracks captured by Benz et al. [47] and additional images of bridge cracks obtained by this study using UAVs, without any use of image-enhancing techniques. The dataset was composed entirely of unprocessed, raw images, with the aim of maintaining the authenticity and precision of the original captures. To create the dataset, the crack areas from the original images were cropped into 648 images of 256 × 256 pixels. Photoshop was used to draw the crack labels and perform binarization processing. An example of the completed UAV dataset is shown in Figure 4; the raw images were captured on overcast days. The UAV used in this study was the Phantom 4 RTK; the relevant parameters are shown in Table 2.

Table 2. Parameters of the Phantom 4 RTK.

General information			
Weight	Maximum rising speed	Maximum tilt angle	Hover accuracy
1391 g	6 m/s (Automatic flight) 5 m/s (Manual operation)	25° (Positioning mode) 35° (Attitude mode)	Enable RTK Vertical: ±0.1 m; Horizontal: ±0.1 m
Drawing function			
Ground sampling distance	Controllable rotation range	Height measurement range	Accurate hover range
(H/36.5) cm/pixel	Pitch: −90° to +30°	0–10 m	0–10 m
Camera			
Mechanical Shutter	Maximum photo resolution	Electronic shutter	Photo Format
8–1/2000 s	4864 × 3648 (4:3) 5472 × 3648 (3:2)	8–1/8000 s	JPEG
Intelligent flight battery			
Capacity	Specifications	Overall weight of battery	Maximum charging power
5870 mAh	PH4-5870 mAh-15.2 V	468 g	160 W
Remote control smart battery			
Capacity	Specifications	Type	Energy
4920 mAh	WB37-4920 mAh-7.6 V	LiPo 2S	37.39 Wh



(**a**) (**b**)

Figure 4. Example of the contents of the UAV crack dataset: (**a**) original image; (**b**) label.

3. Model Training

To evaluate the effectiveness of models trained on existing datasets for UAV images, four DNN models were trained and tested in this study. These models were developed using the following three-step process: (i) setting model hyperparameters; (ii) model training and UAV image dataset test; and (iii) on-site UAV crack image test. The UAV dataset was tested and evaluated using U-Net, DeepLab v3 (MobileNet v3), DeepLab v3 (ResNet50), and TransUNet [48]. Figure 5 shows the specific flowchart in this study.

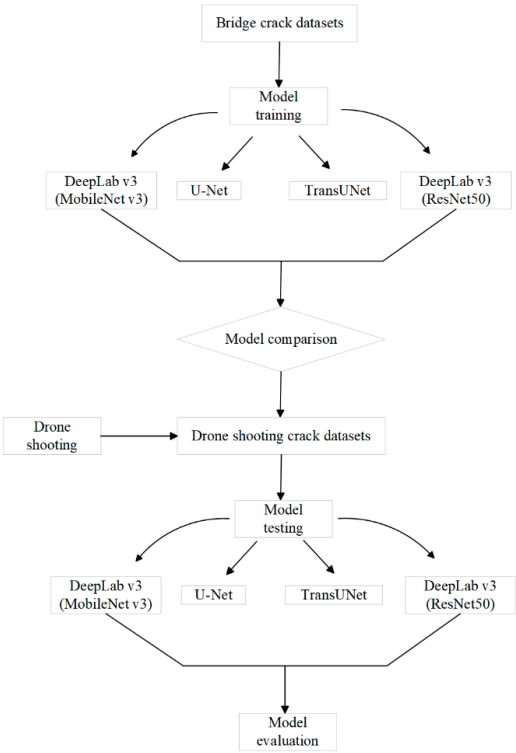


Figure 5. Flow chart of the evaluation process of crack detection.

3.1. Hyperparameter Tuning

The hyperparameter settings were as follows: batch size was set to 4; learning rate was set to 0.001; weight decay was 0.0001; and the epoch was set to 200. The training set used 6000 cracked and non-cracked images mixed at a ratio of 3:1; the validation set used 2000 images. The U-Net, TransUNet, and the backbone network of DeepLab v3 (ResNet50 and MobileNet v3-based) were trained. The optimal model hyperparameters obtained and experimental result indicators are shown in Table 3. In terms of the mIoU, which is quite important in semantic segmentation, TransUNet was slightly better than the other

models, with a value of 0.567. DeepLab v3 (ResNet50) with a value of 0.554 and U-Net with a value of 0.540 followed. In terms of precision, U-Net was the best, with a value of 0.797, and DeepLab v3 (ResNet50) and TransUNet were almost the same. Regarding the recall, DeepLab v3 (ResNet50) was the best and U-Net and TransUNet were quite similar. Generally, U-Net, DeepLab v3 (ResNet50), and TransUNet achieved similar crack detection performance, while DeepLab v3 (MobileNet v3) was not as accurate. The reason might be that the backbone of MobileNet v3 is much smaller than those of the other models, and was thus not good at feature learning in these specific cases. Besides, the train-loss was demonstrated in Figure 6 to show the training process.

Table 3. Model part of the training data.

Model Name	Precision	mIoU	Recall	F1
U-Net	0.797	0.540	0.684	0.736
DeepLab v3 (MobileNet v3)	0.546	0.415	0.691	0.610
DeepLab v3 (ResNet50)	0.780	0.554	0.708	0.742
TransUNet	0.781	0.567	0.675	0.724

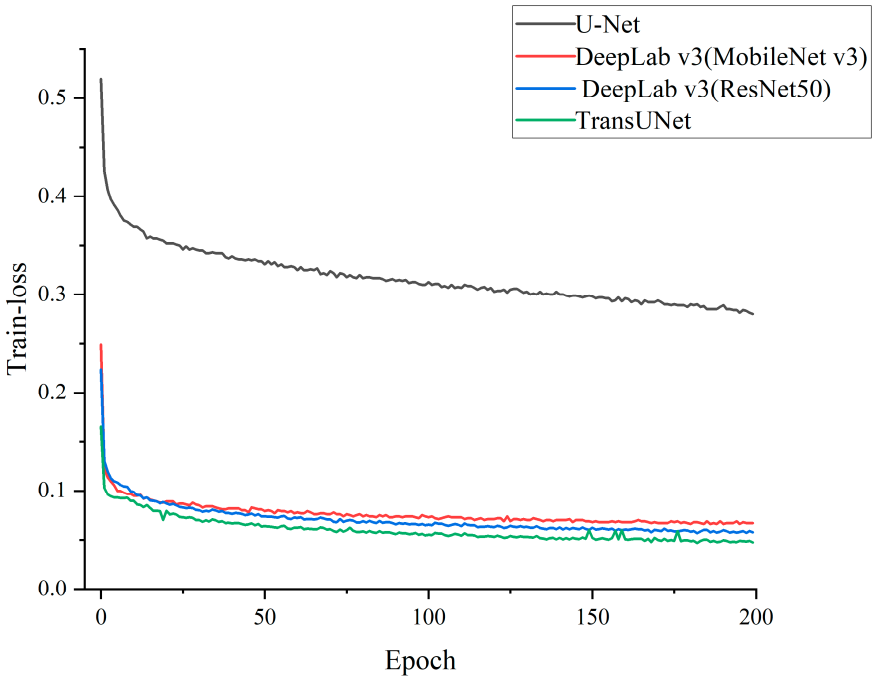


Figure 6. Training process loss curve.

To visually demonstrate the recognition performance of the DNNs on cracks, we used the trained models to predict six crack images; the results are shown in Figure 7. The first row of the table displays the original test images of the cracks; the second row shows the ground truth images that were manually marked, while the third to the sixth rows represent recognition outputs from the four DNN models.

Generally, all four of the models detected the cracks in Pic2 and Pic3 satisfactorily. Yet, there existed incorrect detection results, marked in red boxes. The U-Net network demonstrated good recognition results on all six crack images, correctly identifying the small features of the cracks and providing a detailed segmentation of the crack morphology that closely matched the original crack forms. However, it incorrectly identified some noise as cracks in Pic4, and experienced disconnection in the recognition of cracks with faint or large borders, such as in Pic6. In the DeepLab v3 model with MobileNet v3 as the backbone

network, the model did not produce good recognition results on the cracks in Pic1 and Pic5, with almost no effective recognition. In the DeepLab v3 model with ResNet50 as the backbone network, the model failed to identify the lower part of the crack in Pic5, but performed well on other images. TransUNet was accurate in detecting the crack regions, except for in Pic5.

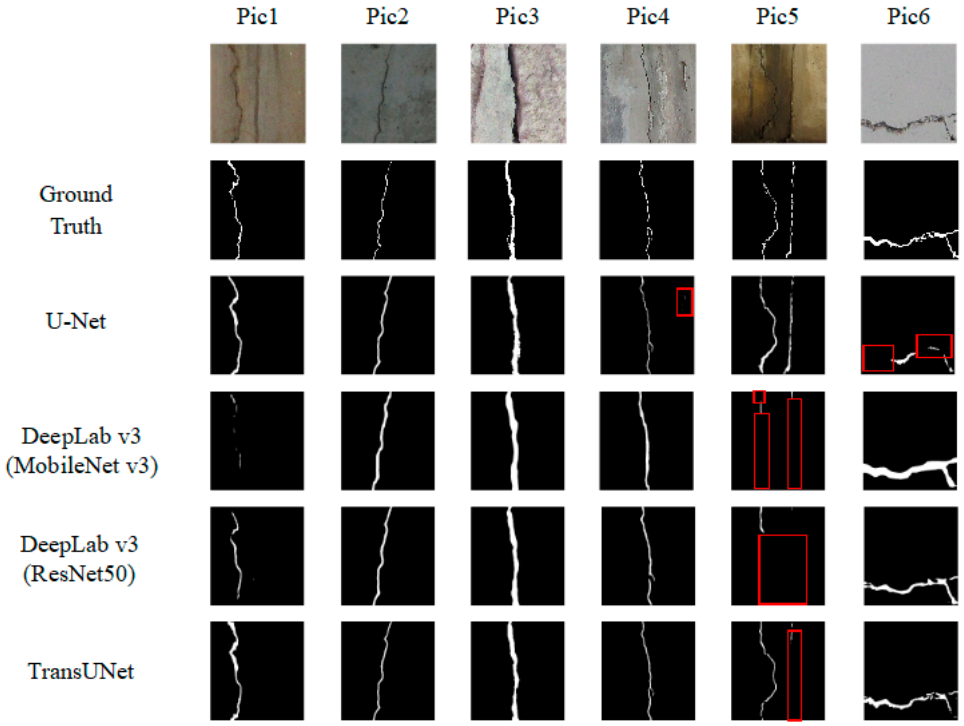


Figure 7. Crack detection based on four DNN models.

From the experimental image results, it can be concluded that the U-Net and TransUNet were good at identifying small crack branches and the segmentation morphology closely matched the original cracks. DeepLab v3 had weak recognition ability in areas where the light distribution was uneven and the contrast between the blank area and the crack area is low; the recognized images were wider than the original cracks. However, both DeepLab v3 models exhibited obvious advantages on cracks with indistinct borders and large widths. None of the four models identified the background ridges on Pic1 as cracks. The experiments demonstrated that different models have their own recognition advantages for cracks with different morphologies and backgrounds. Dim backgrounds, such as that in Pic5, can cause incorrect identification in crack detection.

3.2. UAV Image Testing

We conducted experiments on a UAV crack dataset using existing optimal models of the four DNN models; the results are summarized in Table 4. The precision of the U-Net network decreased by 9.7%, the mIoU value decreased by 6.9%, and the F1 decreased by 19.6%; however, the recall rate increased by 13.3%, reaching 0.775. Regarding DeepLab v3 with MobileNet v3 as the backbone, the precision value decreased by 2.7% and the mIoU value decreased by 4.8%, the recall rate decreased by 12.3% to 0.647, and the F1 value decreased by 25.7%. The DeepLab v3 network with ResNet50 as the backbone experienced a decrease in precision by 2.7%, a decrease in mIoU value by 3.2%, a decrease in recall by 8.6%, and a decrease in F1 by 15.4%. In terms of TransUNet, the precision value decreased drastically by 23.7% and the mIoU value decreased by 7.8%, the recall rate increased drastically by 19.4% to 0.806, and the F1 value decreased by 5.4%. Generally, the performance of the four models decreased by different intensities between 3.2% and 7.8% in terms of the mIoU. This showed that the crack detection performance did not decrease drastically, which might be

because the surface materials were all concrete, and in spite of the differences in lighting conditions, resolution, etc. Meanwhile, the specific indices of the models changed more considerably, especially the precision, which decreased by between 9.7% and 27.4%. The F1 value also experienced a large decrease. An interesting phenomenon was observed in the U-Net and TransUNet models: both of the recalls considerably increased. Similar to the test with the existing dataset, TransUNet achieved the best performance in terms of the mIoU; the performance ranking was the same, as shown in Table 3.

Table 4. Some training data for the model.

Model Name	Precision	mIoU	Recall	F1
U-Net	0.720	0.503	0.775	0.592
DeepLab v3 (MobileNet v3)	0.531	0.395	0.606	0.453
DeepLab v3 (ResNet50)	0.759	0.536	0.647	0.628
TransUNet	0.596	0.523	0.806	0.685

The proportion of training samples can affect the results; therefore, we considered the sensitivity in the choice of training and validation sets. The classic U-Net was used as the experimental DNN. We first employed k-fold cross-validation at a ratio of 3:1. The 8000 images in the crack dataset were divided into four equal parts: three-quarters of the 2000 images were randomly selected for training, and the remaining quarter, the other 2000 images, represented the validation set. In addition, different mix ratios of 4:1 and 5:1 were compared. The test image dataset comprised the UAV images; the results are summarized in Table 5. The crack image dataset proportions did have an effect on the crack detection performance. k-fold cross-validation-1 demonstrated the best mIoU of 0.543, which was better than the first one (0.503) at 7.4%. The change in mix ratio affected the performance, but did not exhibit obvious trends. The training image dataset was relatively large; therefore, the proportions of training and validation images did have an effect, but this was comparatively uncertain in the studied cases.

Table 5. Cross-validation of different mix ratios.

Model Name	Precision	mIoU	Recall	F1
U-Net	0.720	0.503	0.775	0.592
k-fold cross-validation-1	0.724	0.543	0.622	0.621
k-fold cross-validation-2	0.716	0.511	0.663	0.596
k-fold cross-validation-3	0.731	0.532	0.654	0.616
U-Net (Mix ratio: 4:1)	0.732	0.532	0.624	0.616
U-Net (Mix ratio: 5:1)	0.728	0.518	0.633	0.605

In order to visualize the trained DNNs in the detection of cracks in UAV images, tests of the sub-images were first conducted. Figure 8 shows the effects of the crack images identified by the four DNNs, and the incorrect detection results were marked in red boxes. Figure 8 shows that the four models achieved satisfactory recognition results for crack detection in Pic7, Pic8, and Pic12. Pic9 caused incorrect detection for the two DeepLab v3 versions and TransUNet. The dim background of Pic11 deceived these three models, but was detected successfully by TransUNet.

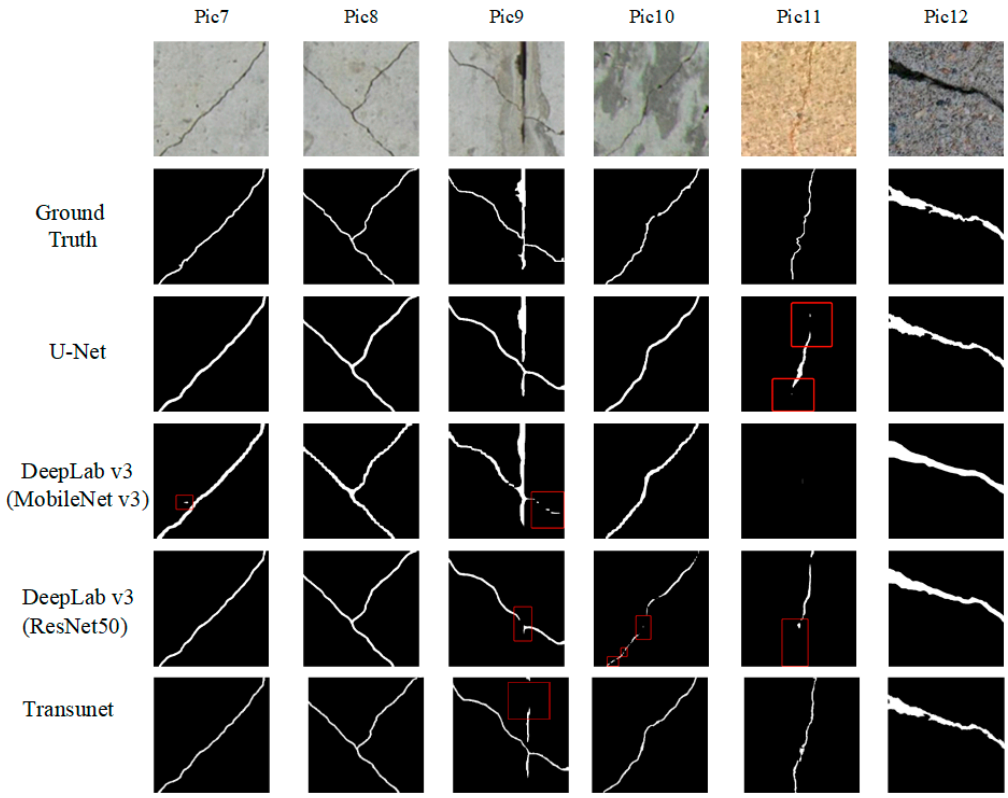


Figure 8. Crack detection on the UAV images.

Except for the perceptual comparison and demonstration of the crack detection performance, quantitative comparison regarding the crack area was conducted for the results of Pic7 to Pic12. Calculations of the crack area were based on OpenCV, as shown in Table 6.

Table 6. Comparison of the crack area in pixels.

Crack Area Source	Pic7	Pic8	Pic9	Pic10	Pic11	Pic12
Ground Truth	1565	2026	3189	1374	717	3812
U-Net	2301	3043	3761	2055	800	3677
DeepLab v3 (MobileNet v3)	2785	3254	3574	2272	0	4760
DeepLab v3 (ResNet50)	1767	2554	1564	608	583	4187
TransUNet	1457	1989	1835	1277	956	3577

For the area indicator of cracks, the same distribution was observed as for the first two indicators; however, there was a more obvious distinction between the U-Net and DeepLab (ResNet50) models. For the case of large cracks (Pic9 and Pic12), U-Net exhibited the smallest error. The specific values are shown in Table 6. U-Net achieved the smallest relative error, with a value of 3.5% for Pic12; the mean relative error of all six images was 30%. The smallest and mean relative errors for DeepLab v3 (MobileNet v3), DeepLab v3 (ResNet50), and TransUNet were 12.1% and 56.8%, 12.9% and 29.0%, and 6.2% and 16.3%, respectively. The TransUNet results were closer to the ground truth. In addition, DeepLab v3 (ResNet50) was similar to U-Net, and both were better than DeepLab v3 (MobileNet v3).

3.3. Field Crack Identification Experiment Based on UAV

In this study, we conducted UAV-based crack detection on an in-service bridge to validate whether the models trained on existing datasets can identify cracks from UAV images. The resolution of the UAV dataset images was 256 × 256. The drone used was

the DJI Phantom 4 RTK, a small multi-rotor high-precision surveying UAV (as shown in Figure 9). The UAV has a built-in RTK module and an obstacle-avoiding camera, which yields a greater anti-magnetic interference ability and accurate positioning ability, so as to improve the accuracy of image data and realize accurate data acquisition. The on-site detection process is shown in Figure 10. Our approach involved flying the UAV as close to the bridge superstructure as possible, at a distance of approximately 2.5 m. The environment was overcast during the shooting process. During the flight, any obstacles or unexpected distances from target objects resulted in an alarm and flight advice to ensure safe operations. We closely monitored the screen on the remote control in real time, capturing photographic evidence of cracks or areas where cracks were suspected. Due to the low resolution of the remote-control screen, it was difficult to accurately identify the areas with cracks on site. Therefore, it was necessary to screen clear images with more cracks and conduct subsequent recognition testing.

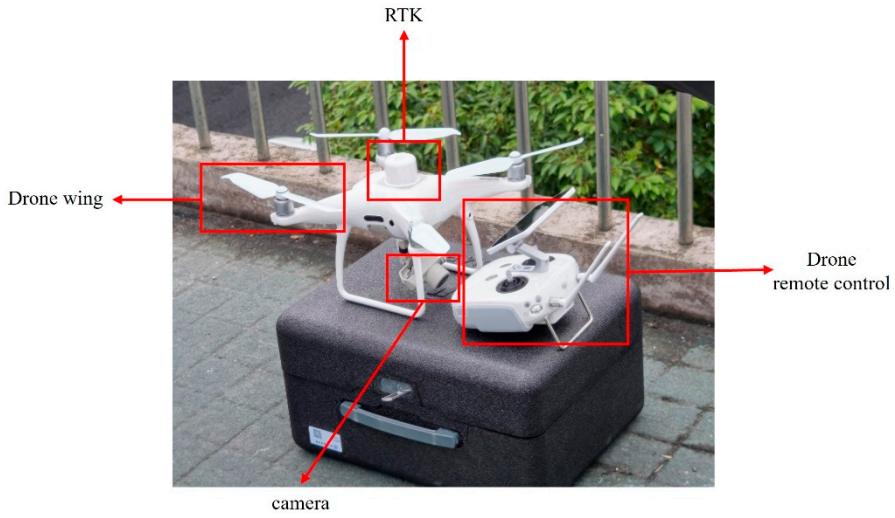
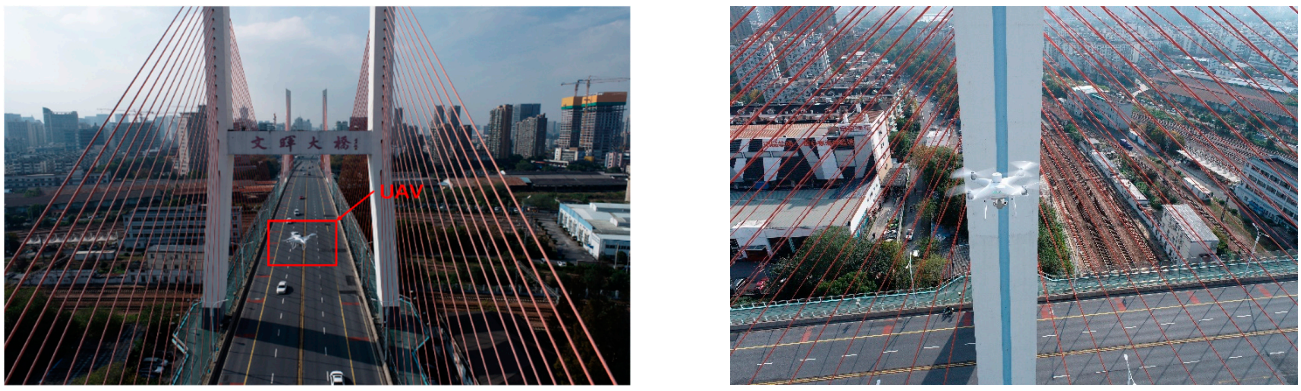


Figure 9. The UAV used in this study.



(**a**) (**b**)

Figure 10. UAV-based crack inspection: (**a**) overall scanning; (**b**) bridge tower inspection.

In Figure 11, we present the recognition results of crack detection from the raw images in our UAV detection of the bridge tower, and the incorrect detection was marked in red boxes. Our analysis included selecting raw images with various vertical and horizontal crack shapes to test the recognition performance of each model. Every large map depicted the main cracks, along with multiple secondary cracks (typically smaller in size). According to the identification results, U-Net could successfully identify the main cracks and find most of the secondary cracks. DeepLab V3 (MobileNet-V3-based) was weak in detecting cracks in this specific case.

DeepLab V3 (ResNet50-based) was able to identify the main crack, but the disconnection phenomenon marked in the red box in Figure 11 appeared. The ability to identify secondary cracks was weak; thus, it missed the small cracks detected by the U-Net. As for TransUNet, it performed well in the first and the third images, where the detected crack regions were more apparent and fine. However, the tiny cracks in the second image successfully deceived TransUNet. The results indicate that among the four models, TransUNet demonstrated a superior capacity to detect cracks, but tiny cracks could cause incorrect detection. Although U-Net performed similarly, the reliability of the results requires further investigation. The two versions of DeepLab v3 did not perform well in this case.

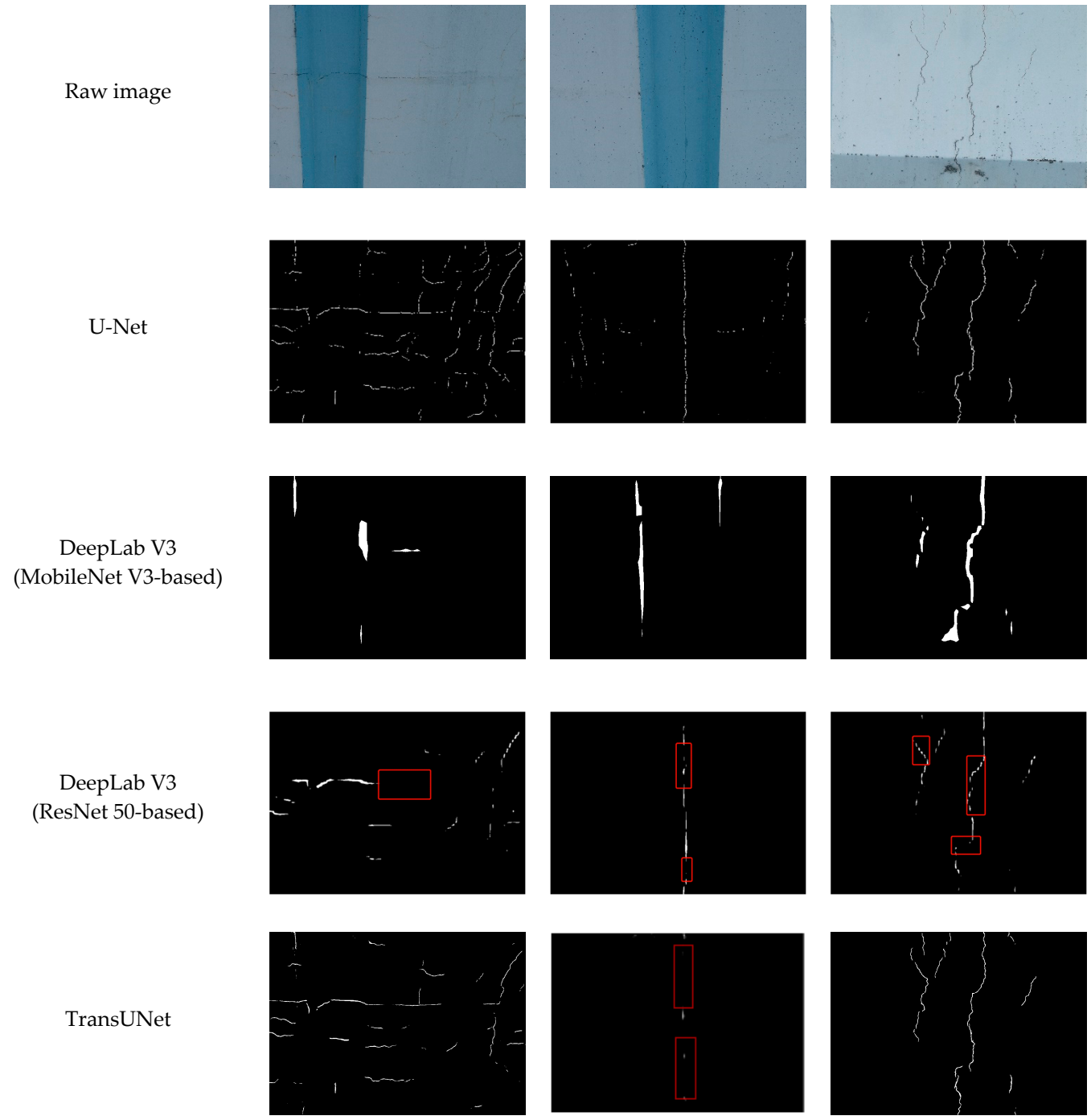


Figure 11. Field crack identification results of the UAV images.

4. Conclusions

This study investigated the performance of using existing crack image datasets to train the DNNs to detect cracks from UAV images. Four DNNs were adopted for the comparison study based on an existing crack image dataset and a newly established UAV image dataset. A field experiment was conducted to verify and intuitively demonstrate the ability of the trained DNNs in detecting cracks from raw structural images. This study has shown that models trained using the existing crack dataset can detect cracks in UAV images with a slight reduction in crack detection performance. Detailed conclusions can be drawn as follows:

(i) A publicly available crack image dataset with 11,000 images from handheld cameras was adopted for training purposes. Meanwhile, a small UAV image-based dataset with 648 images was newly established for testing. Four different DNNs—U-Net, DeepLab v3 (MobileNet v3), DeepLab v3 (ResNet50), and TransUNet—were adopted for comparison to study the performance of different DNNs.

(ii) The four DNN models were first trained on existing datasets and exhibited different features on different types of cracks. As demonstrated by the evaluation indices of precision, mIoU, recall, and F1, TransUNet was the best model and U-Net was a close second. DeepLab v3 (ResNet50) demonstrated similar performance to TransUNet and U-Net, while DeepLab v3 (MobileNet v3) was less accurate compared with the other three models. The image tests showed that the four models could successfully detect most of the crack regions, but dim and low-contrast backgrounds would cause incorrect detection.

(iii) The tests of the UAV image-based dataset indicated that the performance of all four models decreased. In detail, the mIoU reductions in U-Net, DeepLab v3 (MobileNet v3), DeepLab v3 (ResNet50), and TransUNet were 6.9%, 4.8%, 3.2%, and 7.8%, respectively. The two U-Net-related models exhibited an increase in the recall rate of more than 10%, while the precision and F1 dropped. Based on the classic U-Net model, the influence of the proportion of crack training samples was investigated based on k-fold cross-validation and two more mix ratios. It was found that the mix ratio did influence the crack detection performance; however, the influence was not certain and the largest change in all six groups was 7.4%. It showed that the large training dataset had rich diversity and could release the imbalance issue of the training sample.

(iv) The sub-UAV image test showed that the trained models could detect most of the crack regions, but a low-contrast background and fine cracks caused incorrect detection. Quantitative evaluation of the crack areas indicated that TransUNet was the best, with the smallest relative error of 6.2% and an average relative error of 16.3. The raw UAV image tests revealed that TransUNet and U-Net performed similarly. The TransUNet results were more continuous and smoother, but tiny cracks caused mistakes. DeepLab V3 (ResNet50) was better than DeepLab V3 (MobileNet V3), but they demonstrated problems of discontinuity and were less accurate than the two U-Net-based versions.

Author Contributions: Conceptualization, T.J., W.Z. and C.C.; methodology, T.J., W.Z., C.C. and Y.Z.; validation, B.C., Y.Z. and H.Z.; formal analysis, T.J. and W.Z.; investigation, T.J., Y.Z. and W.Z.; resource, T.J., B.C. and C.C.; data curation, T.J. and W.Z.; writing—original draft preparation, T.J. and W.Z.; writing—review and editing, T.J., W.Z. and C.C.; visualization, T.J., W.Z. and B.C.; supervision, T.J. and C.C. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper was jointly supported by the Youth Fund of the National Natural Science Foundation of China (No. 52308332), the China Postdoctoral Science Foundation (Grant No. 2022M712787), the National Natural Science Foundation of China (Grant No. 12272334), the Zhejiang Provincial Natural Science Foundation of China (Grant No. LGG21E080004), and the Scientific Research Foundation of Hangzhou City University (Grant No. J-202203).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare they have no conflict of interest.

References

1. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
2. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI*; Springer: Cham, Switzerland, 2015; pp. 234–241.
3. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
4. Liu, Z.; Cao, Y.; Wang, Y.; Wang, W. Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automat. Constr.* **2019**, *104*, 129–139. [CrossRef]
5. Qiao, W.; Zhang, H.; Zhu, F.; Wu, Q. A crack identification method for concrete structures using improved U-Net convolutional neural networks. *Math. Probl. Eng.* **2021**, *2021*, 6654996. [CrossRef]
6. Sun, X.; Xie, Y.; Jiang, L. DMA-Net: DeepLab with Multi-Scale Attention for Pavement Crack Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 18392–18403. [CrossRef]
7. Zou, Q.; Zhang, Z.; Li, Q.; Qi, X.; Wang, Q.; Wang, S. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Trans. Image Process.* **2019**, *28*, 1498–1512. [CrossRef]
8. Ghazali, M.H.M.; Rahiman, W. Vibration-based fault detection in drone using artificial intelligence. *IEEE Sens. J.* **2022**, *22*, 8439–8448. [CrossRef]
9. Nooralishahi, P.; Ramos, G.; Pozzer, S.; Ibarra-Castanedo, C.; Lopez, F.; Maldague, X.P.V. Texture analysis to enhance drone-based multi-modal inspection of structures. *Drones* **2022**, *6*, 407. [CrossRef]
10. Smaoui, A.; Yaddaden, Y.; Cherif, R.; Lamouchi, D. Automated Scanning of Concrete Structures for Crack Detection and Assessment Using a Drone. In Proceedings of the 2022 IEEE 21st international Ccnference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sousse, Tunisia, 19–21 December 2022; pp. 56–61.
11. Ngo, B.T.; Luong, C.X.; Ngo, L.; Luong, H. Development of a solution for collecting crack images on concrete surfaces to assess the structural health of bridges using drone. *J. Inf. Telecommun.* **2023**, *7*, 304–316. [CrossRef]
12. Zhong, X.G.; Peng, X.; Shen, M. Study on the feasibility of identifying concrete crack width with images acquired by unmanned aerial vehicles. *China Civ. Eng. J.* **2019**, *52*, 52–61.
13. Peng, X.; Zhong, X.; Zhao, C.; Chen, Y.F.; Zhang, T. The feasibility assessment study of bridge crack width recognition in images based on special inspection UAV. *Adv. Civ. Eng.* **2020**, *2020*, 8811649. [CrossRef]
14. Li, Y.; Ma, J.; Zhao, Z.; Shi, G. A Novel Approach for UAV Image Crack Detection. *Sensors* **2022**, *22*, 3305. [CrossRef]
15. Guo, F.; Qian, Y.; Liu, J.; Yu, H. Pavement crack detection based on transformer network. *Autom. Constr.* **2023**, *145*, 104646. [CrossRef]
16. Kao, S.P.; Chang, Y.C.; Wang, F.L. Combining the YOLOv4 deep learning model with UAV imagery processing technology in the extraction and quantization of cracks in bridges. *Sensors* **2023**, *23*, 2572. [CrossRef]
17. Jeong, E.; Seo, J.; Wacker, J.P. UAV-aided bridge inspection protocol through machine learning with improved visibility images. *Expert Syst. Appl.* **2022**, *197*, 116791. [CrossRef]
18. Baltacıoğlu, A.K.; Öztürk, B.; Civalek, Ö.; Akgöz, B. Is Artificial Neural Network Suitable for Damage Level Determination of Rc-Structures? *Int. J. Eng. Appl. Sci.* **2010**, *2*, 71–81.
19. Kim, B.; Cho, S. Automated Vision-Based Detection of Cracks on Concrete Surfaces Using a Deep Learning Technique. *Sensors* **2018**, *18*, 3452. [CrossRef]
20. Deng, J.H.; Lu, Y.; Lee, V.C.S. Concrete crack detection with handwriting script interferences using faster region-based convolutional neural network. *Comput. Aided Civ. Infrastruct. Eng.* **2020**, *35*, 373–388. [CrossRef]
21. Ye, X.W.; Jin, T.; Li, Z.X.; Ma, S.Y.; Ding, Y.; Ou, Y.H. Structural crack detection from benchmark data sets using pruned fully convolutional networks. *J. Struct. Eng.* **2021**, *147*, 04721008. [CrossRef]
22. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
23. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the 26th Annual Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012.
24. Szegedy, C.; Liu, W.; Jia, Y.Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
26. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
27. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
28. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.

29. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; p. 31.
30. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 492–1500.
31. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
32. Zhang, X.; Zhou, X.Y.; Lin, M.X.; Sun, R. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
33. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
34. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
35. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
36. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
37. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015.
38. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
39. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
40. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
41. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
42. Shelhamer, E.; Long, J.; Darrell, T. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern. Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef]
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
44. Qin, X.; Zhang, Z.; Huang, C.; Dehghan, M.; Zaiane, O.R.; Jagersand, M. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recogn.* **2020**, *106*, 107404. [CrossRef]
45. Liu, Y.; Bao, Y. Intelligent monitoring of spatially-distributed cracks using distributed fiber optic sensors assisted by deep learning. *Measurement* **2023**, *220*, 113418. [CrossRef]
46. Rosso, M.M.; Aloisio, A.; Randazzo, V.; Tanzi, L.; Cirrincione, G.; Marano, G.C. Comparative deep learning studies for indirect tunnel monitoring with and without Fourier pre-processing. *Integr. Comput. Aided Eng.* **2023**, *Pre-press*, 1–20. [CrossRef]
47. Benz, C.; Debus, P.; Ha, H.K.; Rodehorst, V. Crack Segmentation on UAS-based Imagery using Transfer Learning. In Proceedings of the 2019 International Conference on Image and Vision Computing New Zealand (IVCNZ), Dunedin, New Zealand, 2–4 December 2019.
48. Chen, J.N.; Lu, Y.Y.; Yu, Q.H.; Luo, X.D.; Adeli, E.; Wang, Y.; Lu, L.; Yuille, A.L.; Zhou, Y.Y. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv* **2021**, arXiv:2102.04306.