

# Article U-Net-Based CNN Architecture for Road Crack Segmentation

Alessandro Di Benedetto 💿, Margherita Fiani 💿 and Lucas Matias Gujski \*

Department of Civil Engineering (DICIV), University of Salerno, 84084 Fisciano, Italy \* Correspondence: lgujski@unisa.it

**Abstract:** Many studies on the semantic segmentation of cracks using the machine learning (ML) technique can be found in the relevant literature. To date, the results obtained are quite good, but often the accuracy of the trained model and the results obtained are evaluated using traditional metrics only, and in most cases, the goal is to detect only the occurrence of cracks. Particular attention should be paid to the thickness of the segmented crack since, in road pavement maintenance, the width of the crack is the main parameter and is the one that characterizes the severity levels. The aim of our study is to optimize the crack segmentation process through the implementation of a modified U-Net model-based algorithm. For this, the Crack500 dataset is used, and then the results are compared with those obtained from the U-Net algorithm, which is currently found to be the most accurate and performant in the literature. The results are promising and accurate, as the findings on the shape and width of the segmented cracks are very close to reality.

Keywords: computer vision; deep learning; CNN; U-net; road crack

# 1. Introduction

Every year, government authorities all around the world check and analyze the quality and performance of roads in order to detect possible road safety issues [1]. Good road conditions are the most important factor for safe driving and traffic conditions. The deterioration of road conditions in the form of cracks significantly degrades the quality and performance of roads and causes a significant decrease in traffic safety [2].

Cracks are a typical type of pavement distress that can compromise the safety of roads and highways. Localizing and repairing cracks is a critical task for the transportation maintenance department in order to keep the roads in excellent condition. Crack identification is an important part of the work.

The manual, labor-intensive, and subjective aspects of the traditional task of road crack segmentation by expert professionals makes it highly time-consuming. Therefore, designing and building an automated and effective road fracture segmentation system is extremely useful and valuable [3].

The identification of cracks can be carried out on 3D or 2D data and is aimed at assessing their severity levels. Based on the severity levels, the management entity can draw up an efficient maintenance plan [4].

The acquisition of 3D data involves using expensive and complex systems based on LiDAR technology and, in most cases, requires time-consuming and costly processing, particularly when the number of datapoints acquired is large [5]. The acquisition of 2D data, such as images, involves the use of lower-cost systems and significantly reduced processing times [6]. This has driven many researchers to study high-performance algorithms for identifying and segmenting cracks from images [7].

The automated road crack segmentation technique efficiently identifies road cracks and helps qualified technicians to evaluate the road's performance objectively, as well as helps the relevant departments to maintain roads that are in good condition and extend their service life. Given that automated survey systems can identify different types of road cracks at fast speeds in various situations, even in adverse weather conditions, it



Citation: Di Benedetto, A.; Fiani, M.; Gujski, L.M. U-Net-Based CNN Architecture for Road Crack Segmentation. *Infrastructures* 2023, *8*, 90. https://doi.org/10.3390/ infrastructures8050090

Academic Editor: Linh Truong-Hong

Received: 7 February 2023 Revised: 26 April 2023 Accepted: 3 May 2023 Published: 6 May 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). seems clear that introducing an intelligent pavement repair technology might lead to more effective outcomes.

In recent years, there have been many achievements related to road crack image segmentation algorithms based on computer vision (CV) techniques [8]. A machine can learn from the qualities of digital photos and videos thanks to computer vision. Using visual data, it helps to recognize characteristics and patterns better [9].

However, traditional CV techniques have weak generalization and adaptation abilities and heavily rely on the quality of the pictures. Additionally, the complex environment of the road surfaces results in poor camera settings with issues including low contrast, inconsistent lighting, and significant noise, making it challenging to build an efficient detection model using conventional CV approaches [10].

Progress in the identification of cracks was made thanks to deep learning (DL) techniques. Deep learning, a branch of artificial intelligence (AI), has been very successful in semantic segmentation. The semantic segmentation technique can meet the goals of crack segmentation, as it predicts a classification label for each pixel. Convolutional neural networks (CNNs), a key subfield in DL, provide promising results in the pixel-level detection of target objects in noisy images [11,12].

The CNN's end-to-end segmentation method is another benefit, as the segmentation process requires much less human involvement. Compared to the manually created features utilized in conventional approaches, the features obtained by convolutional neural networks offer a better rendering performance. By using this deep learning capability, certain efforts are devoted to developing reliable feature representations that can be utilized for segmenting images of road cracks. Deep neural networks use characteristics to identify whether or not there are cracks in the patches of a picture by combining the pixel-level classification confidence from several frames with various lighting conditions. The literature shows how well deep networks have worked for detecting pavement cracks using computer vision.

Lei Zhang et al. [6] proposed a crack detection method in which the discriminative features are learned directly from raw image patches using the ConvNets. Allen Zhang et al. [13] used two different CNNs for segmentation. Their approach was to first extract the relevant features and then input the same data into a different CNN. Haifeng Li et al. [14] developed a model employing a windowed intensity path technique to segment the extracted candidate cracks using a multivariate hypothesis test.

Tong et al. [15] also developed another two-stage CNN-based model to detect asphalt pavement crack length. The results presented show that the training strategy used produces an increase in accuracy. Fan et al. [16] used a trained CNN to model crack detection as a multi-label classification problem. Jenkins et al. [17] and Nguyen et al. [18] used a U-Netbased architecture for a semantic pixel-wise segmentation of road and pavement surface cracks. The CNNs proposed by Baoxian Li et al. [19] were used to classify crack patches into five categories based on 3D pavement images. They used WayLink's PaveVision3D Ultra to acquire 3D images.

König et al. [20] presented a novel surface crack segmentation method using an encoder-decoder-based DL architecture based on a U-Net-based network, and it was reported that its performance increases the effects when inserting pretrained encoder networks.

Fan et al. [21] demonstrated the use of Deep CNNs to detect and recognize cracks as defects with quantifiable properties in applications for crack detection on pavement surfaces (e.g., crack length and size). In a separate paper, the authors proposed a modified version of the U-Net in which two modules were added to the overall architecture to increase the performance of crack segmentation: the dilation module and the convolution and hierarchical feature learning module [22].

Almost all papers found in the relevant literature confirmed the performance of DL models under mostly ideal conditions with the absence of noise, obstacles, shadows, and overexposed areas, and sometimes without considering any rolling shutter effects.

However, such effects can definitely affect the final results of the segmentation process; thus, a growing research interest lies in those DL models and methodologies that maximize the accuracy of segmentation to have an appropriate confidence margin even in cases of poor-quality images analyzed by the trained model.

As a case in point, An et al. [23] highlighted that the integrated use of optical and thermal images in DL models improves crack detection in the presence of shadow, rust, dust, etc.

Other approaches are based on region-based classification or object detection [24–26], enabling improved classification capabilities in cases where images are acquired under poor conditions. An in-depth and detailed review of the scientific literature on the use of DL techniques for the analysis of distress and cracks in structures and infrastructures is found in [27].

All this confirms that the focus on developing a methodology that maximizes the accuracy of the DL model is a current and widely studied issue in the research.

This study suggests a technique for semantic crack image segmentation based on a residual structure developed with the architecture of the U-Net model and a ResNet50 encoder. Without first identifying the region of interest, this approach can automatically segment the cracks in a paved pavement image with a complex background by independently learning the characteristics of the cracks and obtaining additional feature information.

## 2. Methods and Dataset

The methodology we propose here aims to carry out the segmentation of cracks on road pavement from images acquired with commercial cameras using deep learning CNN techniques implemented in a Python environment.

The dataset used is the one proposed by Yang et al. [28]; the model implemented uses a modified U-Net to improve the automatic crack segmentation process. The proposed methodology is based on the following steps:

- Use of the Crack500 dataset [28];
- Data augmentation on the dataset;
- Implementation of the U-Net model with ResNet50 encoder pretrained with ImageNet;
- Model training;
- Metrics analysis.

A comparison will be made with the main results reported in the literature, particularly with those obtained by Lau et al. [29].

Figure 1 shows the workflow of the proposed methodology.



#### Figure 1. Workflow.

#### 2.1. Dataset

The Crack500 dataset was proposed by Yang et al. [28] and contains images acquired around the main campus of Temple University (Philadelphia, PA, USA) using mobile phones. It initially consisted of 500 images of varying sizes, which are around  $2000 \times 1500$  pixels. Each crack image has a pixel-level annotated binary map. In this study, it was divided into 250 training samples, 200 test samples, and 50 validation samples. To make the training process more efficient, a set of data augmentation procedures were conducted on the data, a technique that has been developed to reduce overfitting [30]. Using Python, each image was cropped into six image regions that were overlapped by 80 pixels, flipped horizontally, and then rotated in 90-degree steps, starting from 0 degrees and increasing up to 270 degrees (Figure 2). As a result, all images and the corresponding masks were obtained at a size of  $512 \times 512$  pixels.



Figure 2. Example of dataset and data augmentation.

## 2.2. The Algorithm

The network architecture we propose is a U-Net-based architecture with a ResNet50 encoder. The U-Net was presented by Ronneberger at the MICCAI conference in 2015 [31]. The U-Net is a U-shaped convolutional neural network that was originally used in the field of medical image segmentation. It has two symmetrical branches and is considered to be an encoder-decoder network structure. The architecture of the U-net is explained in Figure 3.



Figure 3. U-net.

A ResNet50 encoder trained on the ImageNet dataset [32] was utilized in this encoderdecoder-based design. The model quickly converges thanks to the usage of a pretrained encoder. The input picture is passed into the pretrained ResNet50 encoder, whose fundamental building blocks are a set of residual blocks. The relevant features from the input picture are extracted by the encoder with the assistance of these residual blocks, and these characteristics are then sent to the decoder. A transpose convolution is started by the decoder to upscale the input feature maps into the proper form. The specified shape feature maps from the pretrained encoder are then concatenated with these upscaled feature maps using skip connections. By assisting the model in obtaining all the low-level semantic data from the encoder, these skip connections enable the decoder to produce the necessary feature maps. The two  $3 \times 3$  convolution layers are then added after that, with a batch normalization layer and a ReLU non-linearity layer coming after each layer. The output of the final decoder block is sent into a  $1 \times 1$  convolution layer, which is then fed into a sigmoid activation function to produce the appropriate binary mask. The architecture of the proposed algorithm is shown in Figure 4.



Figure 4. Implemented model.

The model uses the Adam optimizer [33] with an initial learning rate set to 0.0001, which is reduced by a factor of 0.1 in every 4 epochs, and cross-entropy loss is established as its loss function. The network converges in 20 epochs. We implemented the network in Python using Tensorflow/Keras. The specifications of the workstation used to train the neural network are: TITAN X GPU (12 GB VRAM), Intel Core i7 processor, and 32 GB RAM.

In our work, the net was trained by using the training pair D = x(i), y(i), where x(i) is the i-th image patch and y(i) 0, 1 is the corresponding class label.

#### 2.3. Evaluating the Segmentation Model

Two common evaluation metrics were utilized to assess the suggested approach in order to objectively estimate the performance of the network model. The  $F_1$  score and Intersection over Union (*IoU*) are the conventional quantitative evaluation metrics utilized in our research.  $F_1$  is the combination of Precision and Recall and is computed as the harmonic mean of the two quantities [34]. Precision (*P*) is the proportion of correctly classified observations per predicted class, whereas Recall (*R*) or Sensitivity is used to measure the percentage of actual positives which are correctly identified.

Often, there is an inverse relationship between Precision and Recall: when precision increases, model sensitivity worsens and vice versa. For these reasons, it is important to find the golden mean, meaning a balance between the two indicators, to obtain a model that best fits the input data. The formulas used are:

$$F_1 = \frac{2PR}{P+R}; P = \frac{TP}{TP+FP}; R = \frac{TP}{TP+FN}$$
(1)

where *TP* is the true positive (samples correctly classified as positive), *FP* is the false positive (samples incorrectly classified as positive), and *FN* is the false negative (samples incorrectly classified as negative). We do not consider the transitional areas (0-pixel distance) between non-crack and crack pixels.

The  $F_1$  score is a combination of Precision and Recall and is a robust indicator for both balanced and unbalanced datasets. In general,  $F_1$  values greater than 0.9 are indicative of a very accurate classification; below 0.5, the classification may be considered inaccurate and therefore unsuitable. Analysis of  $F_1$  is necessary when a balance between Precision and Recall is desired.

Intersection over Union (*IoU*) is a geometric type of evaluation metric. It describes the closeness of the predicted results to the ground-truth bounding boxes and is expressed as:

$$IoU = \frac{Area(B_P \cap B_g)}{Area(B_P \cup B_g)}$$
(2)

where  $B_p$  is the predicted bounding box and  $B_q$  is the ground-truth bounding box.

In this case, the predicted bounding box represents the mask obtained with the proposed model (prediction output) and the ground-truth bounding box represents the mask used to train the implemented model (target mask).

Thus, the overlap occurs between the two masks, the predicted mask (prediction output) and the original mask (target mask). This means that the *IoU* is equal to the number of pixels that are common between the target mask and prediction output divided by the total number of pixels in both masks. The higher the overlap, the higher the score; values close to one indicate an excellent overlap, and values below 0.5 indicate a poor overlap. In other words, should the predicted mask be identical to the mask used for training, the *IoU* would be one. These metrics are commonly used in crack detection, but they do not consider the subjectivity of manually labeled ground truth [35].

## 3. Results

The trained model was applied to the Crack500 dataset. The original dataset consisted of 250 training samples, and then the training dataset was artificially increased with the data augmentation technique to improve performance during the training phase. A total of 12,000 images were used for training, 2400 images were used for validation, and 9600 images for testing. Figure 5 shows that both accuracy curves of the training and the validation set increase and stabilize at high and similar values with a gap of 0.006, indicating that the model is learning correctly and generalizing well to unseen data.



**Figure 5.** The trend of the accuracy curve during training. The epochs are on the X-axis, and the Y-axis is the prediction accuracy. The training curve is in blue, and the validation curve is in orange.

Our model has a total of 20.6 million parameters, and the number of FLOPS is 236G; the inference speed on our hardware is 6.7 frames per second on images of  $512 \times 512$ . As a result, this strategy might not be advantageous for real-time applications and looks more suited for batch processing.

The metrics used for evaluating the proposed model are Precision (P), Recall (R),  $F_1$ , and IoU, as described in Section 2.3.

Table 1 shows the results obtained by applying the model trained with the proposed methodology and the results obtained by using the models reimplemented by both Lau et al. [29] and others [18,36] that achieve high enough and comparable accuracies.

Method	Р	R	F <sub>1</sub>	IoU
U-Net by Nguyen	0.6954	0.6744	0.6895	0.5261
U-Net by Yu	0.6988	0.7619	0.7290	0.5736
U-Net by Lau	0.7426	0.7285	0.7327	0.5782
Proposed U-Net	0.8534	0.6813	0.7577	0.6248

**Table 1.** Testing results of our model compared with other U-Net-based models on the same Crack500 dataset.

It should be noted that the results listed in Table 1 are derived from the application of the same U-Net architecture on the same training dataset (Crack500). Comparing the results shown in the table, those of Lau et al. emerged as the most accurate, resulting in it being the method working best at present; thus, this method is used by us as a benchmark.

Compared to the results of Lau et al., our model produced an increase in Precision, a slight decrease in Recall, and an increase in  $F_1$  and IoU. The increase in Precision means that our model is more reliable, i.e., there are few false positives. However, the model may not predict all events by being less selective or sensitive (low Recall), i.e., there may be many false negatives even if the model is accurate.

Lau et al. implemented a model with the goal of balancing Precision and Recall since it is valuable that the model be precise, but it is also important to have adequate sensitivity (Recall). For us, the increase in Precision and the slight decrease in Recall still produced a balanced result, as can be seen by looking at the  $F_1$  score, which is close to 0.76.

The increase in IoU compared to that of the model of Lau et al. means that the predicted masks are more similar to the original ones (target masks). This is important when estimating crack widths; we assume that the masks used as training datasets are consistent with reality and, consequently, with crack width. Crack width is the main parameter used to derive severity levels [4].

Figure 6 shows examples of the results of the model trained on images referring to some major crack types. The images given as the input to the model, shown in columns a and a1, belong to the test dataset but are definitely not the ones used to train the model, which are others that are not shown. Columns c and c1 show the results we obtained on those input images, while columns b and b1 show the hand-drawn masks available for the dataset for visual comparison with our outputs. Please note that in all cases, our output is more accurate than the hand-drawn masks. In detail, panels in column "a" show: (1–3) transverse cracking, (4–6) longitudinal cracking, and (7–9) block cracking, while panels in column "a1" show: (1–3) portions of alligator cracking, (4–5) edge cracking, and (7–9) portions of non-cracked pavement.

It should be pointed out that the portions of the pavement shown in Figure 6 belong to different types of wear layers, which can be distinguished by different levels of adherence. Indeed, the images display very heterogeneous color scales, which are sometimes marked by the presence of very evident stains due to the presence of aggregates of different types. Columns b and b1 display the target masks used to train the model, and columns c and c1 display the masks obtained by applying the trained model, which are the predicted masks.

In almost all cases, the masks predicted are better than those hand-drawn, particularly concerning the crack width, i.e., in row 5, panel c1(5), the crack was better delineated than the target mask shown in panel b1(5), which is less compliant, in terms of width, than the real configuration, shown in panel a1(5). This aspect is crucial since, in the design of the maintenance plans, the main parameter regulating the severity levels of the different cracks is their width, in addition to the linear development and the area of extension [37].



Figure 6. Output of the implemented model; (a,a1) input images, (b,b1) target masks, (c,c1) predicted masks by our U-net.

# 4. Discussion

To further test the performance of the implemented model, the authors applied it to an orthorectified image not belonging to the Crack500 dataset. The image was acquired with a UAV (unmanned aerial vehicle) equipped with a Zenmuse P1 camera with a flight height of about 30 m. It portrays a short stretch of a provincial road with one carriageway and two lanes in each direction (Figure 7). The crack pattern visible in the figure mainly affects one lane. The crack widths were also measured in situ with a caliper and compared with those derived from the mask obtained.



**Figure 7.** (a) Orthorectified image of a distressed road pavement used to test our model. Reference system: UTM33/RND2008. (b) Map of Italy, the red dot marks the test site.

Figure 8 shows the results of using the proposed model on the section of road pavement characterized by different types of cracking (mainly block, longitudinal, and fatigue cracking) and a surface where the wear layer is of a texture composition different from that present in the Crack500 dataset. The three boxes, (1), (2), and (3), show cracks of different severity levels that correspond to high, moderate, and low, respectively. Again, the panels in column (a) show the predicted masks obtained in column (b), the predicted and classified masks according to crack width presented in column (c), and the measurements made with the caliper.

To quantify the width of the cracks and to assess the sensitivity of the model on width segmentation, the trained model was applied to the orthophoto with a pixel size of 3 mm.

The crack width was calculated using a few functions implemented in Matlab and applied to the model output mask. In particular, the "bwmorph" function removes pixels inside the cracks and keeps only the border pixels (https://www.mathworks.com/help/images/ref/bwmorph.html, accessed on 1 April 2023). The same function allows us to obtain the skeleton of the cracks. On the other hand, the "bwdist" function calculates the distance between the skeleton and the border pixels (https://www.mathworks.com/help/images/ref/bwdist.html, accessed on 1 April 2023). Thus, the distance calculated was used to produce a raster containing the width of the cracks.

The colors in Figure 8a are assigned according to the severity levels of the cracks, as reported by the Distress Identification Manual of the Federal Highway Administration Research and Technology program (https://www.fhwa.dot.gov/publications/research/infrastructure/pavements/ltpp/13092/001.cfm, accessed on 1 April 2023). The examples highlight the good sensitivity of the model in estimating crack width, which is a key aspect for classification in terms of severity levels.



**Figure 8.** Results of a test carried out on an orthorectified image of a distressed road pavement: pixel size 3 mm; (a) segmented cracks superimposed on the orthophoto; (**1a**,**2a**,**3a**) excerpt of the output mask; (**1b**,**2b**,**3b**) excerpt of the orthophoto with the overlaid raster containing the crack width, where the blue arrow points to the position of the caliper; (**1c**,**2c**,**3c**) in situ measurements with a caliper.

Looking at Figure 8a, one can notice that our model does not segment cracks with severity levels lower than the low level, mainly because the resolution of the orthophoto does not allow for the detection of cracks with an amplitude less than or slightly greater than the pixel size. Some cracks characterized by a low severity level were not segmented because the color difference between the crack and the pavement was not sharp enough, but this is plausible given that the images were taken at a flight height of about 30 m. Cracks with medium/high levels of severity were all segmented, and were also observed by traditional surveys carried out in situ.

In particular, a high severity level crack is shown in panel 1c, which is congruent with the crack width estimated from the mask (panels 1a–1b). In panel 2c, a moderate severity level crack is shown; again, the implemented model accurately segmented the crack as the severity level inferred from the mask (panels 2a–2b) is congruent with that measured in situ. Finally, panel 3c shows a case halfway between low and moderate severity, with the crack width being just over 6 mm. In panel 3b, the part considered falls at the medium severity level (>6 mm) at the node of the crack junction; as one moves away from the junction (toward the right), the severity level turns low (<6 mm). Again, the model was able to segment the crack width with sufficient accuracy; the severity level is congruent with that measured in situ.

The performance of segmentation is closely related to the resolution, quality, and somewhat to the exposure of the image. To segment cracks with low or lesser severity levels, it is advisable to use images taken very close to the pavement, preferably from a mobile system mounted on a car and using a medium/high quality camera capable of returning a pixel size of at least half the width of the crack. The key point is that cracks with medium/high severity levels are identified and segmented in almost all cases, which is a major aspect for decision making and drafting pavement management plans.

#### 5. Conclusions

The use of convolutional neural networks for pavement crack detection was the main goal of this study. The structure of our network is a U-Net with a ResNet50 encoder pretrained with the ImageNet dataset. The encoder component has proven to effectively extract image crack features. Even when these have irregular shapes and intricate textures, they can be handled due to the model's ability to accurately represent global context information. On the Crack500 dataset, the suggested crack segmentation model performs well. The results of the mask prediction show that the model can accurately segment cracks.

Even though the approach followed in this study demonstrated good performance, there is still much work to be conducted before pavement cracks can be automatically detected. A drawback of our proposed approach is that it requires a large number of manually drawn pixel-level crack pictures to build effective and accurate models. This is generally a well-known issue in the literature and is true for almost all ML approaches. The performance of the model is directly related to the dataset, and the manual annotation process is time-consuming and subjective. Collecting and labeling data samples takes time and must be performed accurately; synthetic data can be introduced in the model learning process.

The validation of the performance of the models is generally carried out on images acquired in nearly optimal conditions with no noise, obstacles, shadows, or overexposed areas; thus, applying the model on images of lower quality or with the presence of noise could lead to inaccurate results. Nonetheless, the application of our model on a real test area has led to very promising results, as the severity levels resulting from the crack calculation on the resulting masks are in line with those calculated from traditional surveys in situ. The proposed methodology aims to improve a DL model, referred to in the scientific literature as one of the most accurate models for crack segmentation based on a CNN, by modifying its architecture. The improvements made in our model also affect the segmentation of crack width; this aspect is relevant because the width is the key parameter for estimating

severity levels that mainly affect the identification of stretches that should be prioritized for intervention.

Going forward, we aim to optimize the proposed model and test its performance on another dataset type. We hope to build a more sophisticated crack dataset that includes cracks in buildings or bridges to improve crack segmentation algorithms.

Author Contributions: Conceptualization, M.F. and L.M.G.; methodology, M.F. and L.M.G.; software, L.M.G.; validation, A.D.B. and L.M.G.; formal analysis, A.D.B., M.F. and L.M.G.; investigation, A.D.B. and L.M.G.; resources, M.F.; data curation, A.D.B. and L.M.G.; writing—original draft preparation, A.D.B. and L.M.G.; supervision, M.F.; project administration, M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

# References

- 1. Tighe, S.; Li, N.; Falls, L.C.; Haas, R. Incorporating road safety into pavement management. *Transp. Res. Rec.* 2000, 1699, 1–10. [CrossRef]
- Gransberg, D.D.; Tighe, S.L.; Pittenger, D.; Miller, M.C. Sustainable pavement preservation and maintenance practices. In *Climate Change, Energy, Sustainability and Pavements*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 393–418.
- 3. Mohan, A.; Poobal, S. Crack detection using image processing: A critical review and analysis. *Alex. Eng. J.* **2018**, *57*, 787–798. [CrossRef]
- 4. Ragnoli, A.; De Blasiis, M.; Di Benedetto, A.; Blasiis, M.D.; Benedetto, A.D. Pavement Distress Detection Methods: A Review. *Infrastructures* **2018**, *3*, 58. [CrossRef]
- 5. Barbarella, M.; Di Benedetto, A.; Fiani, M. A Method for Obtaining a DEM with Curved Abscissa from MLS Data for Linear Infrastructure Survey Design. *Remote Sens.* **2022**, *14*, 889. [CrossRef]
- Zhang, L.; Yang, F.; Zhang, Y.D.; Zhu, Y.J. Road crack detection using deep convolutional neural network. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Pheonix, AZ, USA, 25–28 September 2016; pp. 3708–3712.
- 7. Wang, W.; Wang, M.; Li, H.; Zhao, H.; Wang, K.; He, C.; Wang, J.; Zheng, S.; Chen, J. Pavement crack image acquisition methods and crack extraction algorithms: A review. *J. Traffic Transp. Eng.* **2019**, *6*, 535–556. [CrossRef]
- Huang, J.; Liu, W.; Sun, X. A pavement crack detection method combining 2D with 3D information based on Dempster-Shafer theory. *Comput. Aided Civ. Infrastruct. Eng.* 2014, 29, 299–313. [CrossRef]
- Lorusso, A.; Messina, B.; Santaniello, D. The Use of Generative Adversarial Network as Graphical Support for Historical Urban Renovation. In Proceedings of the ICGG 2022-Proceedings of the 20th International Conference on Geometry and Graphics, Virtual, 15–9 August 2022; pp. 738–748.
- Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Cambridge, MA, USA, 20–23 June 2015; pp. 1395–1403.
- Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
- 12. Yu, S.; Jia, S.; Xu, C. Convolutional neural networks for hyperspectral image classification. *Neurocomputing* **2017**, *219*, 88–98. [CrossRef]
- Zhang, A.; Wang, K.C.; Li, B.; Yang, E.; Dai, X.; Peng, Y.; Fei, Y.; Liu, Y.; Li, J.Q.; Chen, C. Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. *Comput. Aided Civ. Infrastruct. Eng.* 2017, 32, 805–819. [CrossRef]
- 14. Li, H.; Song, D.; Liu, Y.; Li, B. Automatic pavement crack detection by multi-scale image fusion. *IEEE Trans. Intell. Transp. Syst.* 2018, 20, 2025–2036. [CrossRef]
- 15. Tong, Z.; Gao, J.; Han, Z.; Wang, Z. Recognition of asphalt pavement crack length using deep convolutional neural networks. *Road Mater. Pavement Des.* **2018**, *19*, 1334–1349. [CrossRef]
- 16. Fan, Z.; Wu, Y.; Lu, J.; Li, W. Automatic pavement crack detection based on structured prediction with the convolutional neural network. *arXiv* **2018**, arXiv:1802.02208.
- Jenkins, M.D.; Carr, T.A.; Iglesias, M.I.; Buggy, T.; Morison, G. A deep convolutional neural network for semantic pixel-wise segmentation of road and pavement surface cracks. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 2120–2124.
- Nguyen, N.T.H.; Le, T.H.; Perry, S.; Nguyen, T.T. Pavement crack detection using convolutional neural network. In Proceedings of the Ninth International Symposium on Information and Communication Technology, Danang City, Vietnam, 6–7 December 2018; pp. 251–256.

- 19. Li, B.; Wang, K.C.; Zhang, A.; Yang, E.; Wang, G. Automatic classification of pavement crack using deep convolutional neural network. *Int. J. Pavement Eng.* 2020, 21, 457–463. [CrossRef]
- Jacob, K.; Mark, D.; Mike, M.; Peter, B.; Gordon, M. Optimized deep encoder-decoder methods for crack segmentation. *Digit.* Signal Process. 2020, 108, 102907.
- 21. Fan, Z.; Li, C.; Chen, Y.; Di Mascio, P.; Chen, X.; Zhu, G.; Loprencipe, G. Ensemble of deep convolutional neural networks for automatic pavement crack detection and measurement. *Coatings* **2020**, *10*, 152. [CrossRef]
- 22. Fan, Z.; Li, C.; Chen, Y.; Wei, J.; Loprencipe, G.; Chen, X.; Di Mascio, P. Automatic crack detection on road pavements using encoder-decoder architecture. *Materials* **2020**, *13*, 2960. [CrossRef]
- An, Y.-K.; Jang, K.; Kim, B.; Cho, S. Deep learning-based concrete crack detection using hybrid images. In Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, Denver, CO, USA, 5–8 March 2018; pp. 273–284.
- 24. Dorafshan, S.; Thomas, R.J.; Maguire, M. Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Constr. Build. Mater.* **2018**, *186*, 1031–1045. [CrossRef]
- Ali, R.; Gopal, D.L.; Cha, Y.-J. Vision-based concrete crack detection technique using cascade features. In Proceedings of the Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems, Denver, CO, USA, 5–8 March 2018; pp. 147–153.
- Yeum, C.M.; Dyke, S.J.; Ramirez, J. Visual data classification in post-event building reconnaissance. *Eng. Struct.* 2018, 155, 16–24. [CrossRef]
- Azimi, M.; Eslamlou, A.D.; Pekcan, G. Data-driven structural health monitoring and damage detection through deep learning: State-of-the-art review. Sensors 2020, 20, 2778. [CrossRef]
- 28. Yang, F.; Zhang, L.; Yu, S.; Prokhorov, D.; Mei, X.; Ling, H. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Trans. Intell. Transp. Syst.* 2019, 21, 1525–1535. [CrossRef]
- 29. Lau, S.L.; Chong, E.K.; Yang, X.; Wang, X. Automated pavement crack segmentation using u-net-based convolutional neural network. *IEEE Access* **2020**, *8*, 114892–114899. [CrossRef]
- 30. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. J. Big Data 2019, 6, 60. [CrossRef]
- Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Minich, Germany, 5–9 October 2015; pp. 234–241.
- Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 2012, 25, 1097–1105. [CrossRef]
- 33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. arXiv 2014, arXiv:1412.6980.
- Lipton, Z.C.; Elkan, C.; Naryanaswamy, B. Optimal thresholding of classifiers to maximize F1 measure. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Nancy, France, 14–18 September 2014; pp. 225–239.
- 35. Tsai, Y.-C.; Chatterjee, A. Comprehensive, quantitative crack detection algorithm performance evaluation system. *J. Comput. Civ. Eng.* **2017**, *31*, 04017047. [CrossRef]
- Yu, G.; Dong, J.; Wang, Y.; Zhou, X. RUC-Net: A Residual-Unet-Bpased Convolutional Neural Network for Pixel-Level Pavement Crack Segmentation. Sensors 2022, 23, 53. [CrossRef]
- D6433-18; Standard Practice for Roads and Parking Lots Pavement Condition Index Surveys. ASTM International: West Conshohocken, PA, USA, 2018.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.