Asphalt Pavement Crack Detection Using Image-to-Image Translation

Alireza Sepidbar and Mohammadreza Sabouri, Ph.D., P.E.*

Department of Civil Engineering, Sharif University of Technology, Tehran, Iran;

Department of Civil Engineering, Sharif University of Technology, Tehran, Iran

Abstract

Pavement plays a crucial role in transportation because it is a permanent surface for use in road networks. The health of the pavement ensures the safety and convenience of drivers and passengers. In the past few decades, pavement management systems have encountered challenges which often have produced solutions with excessive demand for resources, but low-accuracy results. New approaches must be developed in order to quickly and economically identify pavement failure, especially cracks. This paper proposes a fast and accurate method for segmentation of all types of cracks in asphalt pavement images based on generative adversarial networks (GANs). The proposed model learns the mapping between two domains of pavement images and images of segmented cracks. This approach does not necessitate any preprocessing or post-processing tasks, and the model generates new images without the need to classify each pixel. It detects cracks with high accuracy using a conditional image-to-image translation. In this study, the model took an average of 0.29 s to identify the cracks in each image. This outstanding crack identification had a precision of 85.76%, a recall of 89.81% and an F1-score of 87.72%.

Key Words: Pavement failure-Transportation-Cracks-Generative adversarial Networks-Image-to-image Translation

* Corresponding author. Tel.: +982166164229
Mobile: +989121727256
Email Address: Sabouri@sharif.edu
(A. Sepidbar); alireza.sepidbar97@sharif.edu
Mobile: +989387774370

1. Introduction

Roads have a direct effect on mobility and communications and are continually used to meet the needs of the users [1]. Pavement conditions can affect travel safety, vehicle operating costs, travel time and greenhouse gas emissions [2]. In the past, pavements were repaired but not managed. However, technological advances have provided the necessary tools for managing and maintaining pavements. A pavement management system determines management and maintenance requirements by prioritizing and determining the optimal times for repair and by predicting future conditions [3].

One of the most important aspects of pavement management systems is failure detection. As a lack of timely maintenance can incur costs to governments, the application of manual or automatic failure detection methods can help to reduce these costs. Manual methods are performed by trained experts who have the ability to assess the condition of a pavement [4]; however, manual evaluations can be time-consuming and dangerous. In addition, the manual method depends on the experience and knowledge of the surveyor and can produce different results based on the experiences of different experts. The advent of automatic methods of assessment have been introduced and welcomed [5,6].

Crack detection can be challenging in some cases because of difficulties such as shadows, changes in image contrast, oil stains and environmental conditions. Zou et al. [7] tried to address these challenges by developing a fully automatic method for crack detection. In their method, the shadows are removed prior to crack detection.

The importance of crack detection in pavement management systems have led to the introduction of several methods. In general, automatic methods based on images can be divided into three categories. The first category includes thresholding, edge detection and region-growing methods. The second category includes machine learning methods that use algorithms such as support vector machines for crack detection. Other methods are based on deep learning and have made significant progress in relation to performance and accuracy [8]. Finally, the third category of methods uses 3D images to detect cracks.

Many studies have been conducted in each of these categories. Oliveira and Correia [9] used dynamic thresholding followed by entropy calculation for crack detection. Zhao et al. [10] tried to improve flaws in the Canny algorithm for detecting weak edges to identify cracks. Safaei et al. [11] introduced a tile-based image processing method, applying localized thresholding to detect cracked tiles by analyzing the spatial distribution of crack pixels. Subsequently, for longitudinal and transverse cracking, a curve is fitted to connect the cracked tiles.

With advancements in machine learning algorithms, numerous research efforts have been undertaken to enhance crack detection and segmentation. Hoang et al. [12] used the artificial bee colony algorithm to optimize the support vector machine model for crack detection and classification. Sabouri and Mohammadi present a new method involving two distinct techniques. A supervised learning approach utilizing Local Binary Pattern (LBP) and Random Forest for crack detection from annotated images, and a density-based technique relying on thresholding and spatial-geometric properties for iterative crack object identification [13]. Another study used an artificial neural network to identify cracks in the images taken by a drone after thresholding and removing noise in the images [14]. Chen et al. [15] presented a crack detection model based on deep learning and used fully convolutional architecture based on SegNet to detect concrete pavement cracks, asphalt cracks and bridge deck cracks.

Another type of deep learning model is convolutional neural networks with U-Net architecture. This model has been used to detect cracks pixel-wise. Huyan et al. [16] used a U-shaped architecture called CrackU-Net to accurately detect crack pixels and achieved high accuracy. Another study used U-Net to detect cracks on images taken by drone at different heights from an airport runway after determining the optimal height [17]. Sabouri and Sepidbar employed the U-Net model to segment cracks, complementing their approach with morphological operations to refine the segmentation results [18]. Li et al. employed a combination of the U-Net model and the ResNet neural network as the fundamental classification network to identify distressed areas within the images [19].

In a different category of deep learning models, object detection algorithms have been employed in numerous studies focusing on pavement distress detection, particularly cracks. YOLO stands out as one of the most renowned object detection algorithms due to its high performance, utilized across various research endeavors [20,21]. Jiang et al. introduced a two-stage approach integrating pavement crack detection and segmentation. Initially, they employed an optimized YOLOv4 for crack detection. Subsequently, cracks identified in the first stage underwent segmentation using a novel deeplabv3+ method [22].

Following the methods that were used for crack detection, the third category of methods was the use of 3D data [8]. Roberts et al. [23] used 3D models that they made using a drone and structure from motion technique to evaluate the condition of the pavement. Salameh et al. introduced a methodology employing high-resolution 3D scanners to establish a ground reference for field pavement cracking distress [24]. However, certain research endeavors focus on evaluating pavement cracks using 2D images. Sabouri and Sepidbar gathered a dataset of pavement cracks using a camera positioned at a specified distance from the pavement [25]. This distance can subsequently be utilized in conjunction with the specifications of the camera to calculate the dimensions of the cracks [26].

In general, methods based on deep learning have gained more popularity because of their superior performance. One of the deep learning models is Generative Adversarial Network (GAN), which was presented in 2014 by Goodfellow et al. [27]. GAN is one of the most common generative models that have been very successful and have the ability to produce real high-resolution images.

One of the applications of Generative Adversarial Network (GAN) is the translation of images from one style to another. Image segmentation and image painting are instances of image-to-image translation applications. Semantic segmentation requires a large number of labeled pixels for training and requires a lot of time to prepare the training data. Guo et al. [28] converted unreal images from video games into real images by using image-to-image translation and segmenting them. Another study used conditional GAN, called Pix2Pix, to map pixels to pixels and is useful for applications such as converting label images into real images in semantic segmentation, converting aerial images into maps and converting black and white images into color images [29].

This research, intended to use GAN and image-to-image translation to detect all types of asphalt pavement cracks across different conditions and shapes. By mapping pavement images to ground truth images, cracks were detected in the image-to-image translation. The image of the pavement surface, referred to as the "Input image," is transformed into an image depicting segmented pixels of cracks, known as the "Target image" or "Ground Truth image". This approach can perform well under different conditions and can identify and isolate cracks with high speed and accuracy.

In contrast to prior research employing pixel-wise segmentation methods, this approach is not reliant on the precision of ground truth images. Furthermore, this model does not necessitate any preprocessing or post-processing tasks, such as histogram equalization, feature extraction, morphological operations, and so forth. Here, the network learns to translate the input into the structure of the ground truth image. Furthermore, unlike other methods where networks predict the class of each pixel (crack or non-crack), this network generate a new image by incorporating the input image details and structure from the ground truth image.

2. Methods

The aim of this research was to develop a method for detecting all types of asphalt pavement cracks with high accuracy and speed using different photographic methods, environmental conditions, changes in brightness and image contrast. The proposed method does not require pre- or post-processing and can be widely used. This study aims to utilize a GAN for crack segmentation. GANs have been successful in different fields because they are based on game theory, while other generative models are based on optimization. In this framework, two models are taught simultaneously. The generative model (G) represents the distribution of the data. The second model is a discriminator (D) which estimates the probability that a sample of the data is from a training set or generated by G. The training aim for the G model is to maximize the error probability of the D. In the regard, p_g is the generating distribution of data x, $p_z(z)$ is the input noise variable and $G(z, \theta_g)$ is the mapping to the data space, where G is represented by a multi-layer perceptron with parameter θ_g. For the D model, the mapping is in the form of $D(x, \theta_d)$. D model is trained to maximize the probability of assigning the correct label to the samples coming from G and the training samples. G is simultaneously trained to the minimize $\log(1 - D(G(z)))$. In other words, D and G play a minimax game with the value of the V function as expressed in Equation 1 [27]:

$$min_G max_D V(D,G) = \mathrm{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathrm{E}_{z \sim p_z(z)}[\log(1-D(G(z)))] \qquad (1)$$

GANs have been used in different fields in different ways. One application is for image-to-image translation, which has as its goal to convert the input image from the first domain to the target image in the second domain with the aim of preserving the intrinsic content. Image conversion from one mode to another has solved many image processing and computer vision problems. The training process is carried out with the aim of mapping $G_{A \to B}$ from domain A to domain B and produce image x_{AB} [30,31]. The image-to-image translation can be expressed as Equation 2:

$$x_{AB} \in B : x_{AB} = G_{A \to B}\left(x_A\right) \tag{2}$$

During the learning phase, this method acquires the capability to transform the input image into a representation resembling the ground truth image. So, there are no constraints on segmenting all types of cracks across diverse circumstances. Despite the presence of ground truth images with low accuracy, this method can still be trained and applied.

In order to reduce computations and training time, prior to training, the sizes of the images have been reduced and the target images (i.e., ground truth) and the input images have been concatenated. Figure 1 shows an example of a concatenated image.

Figure 2 shows the concept of a GAN, which was used as the main model. As shown, the generative network produces a random distribution. Real samples and samples generated by the generator (i.e., fake samples) enter the discriminator. If the discriminator recognizes a sample as fake, the generative network will be updated until it can generate a fake sample that can receive a “real” label from the discriminator. In other words, the goal of the generator is to produce a fake sample that can deceive the discriminator and receive a “real” label. The purpose of the discriminator is to distinguish fake samples from real ones. These two networks compete to reach an optimal solution.

The pavement cracks were identified through the translation of the input image to the target image. Concatenated images consist of an input image combined with a target image and the network is trained using them. The aim of image-to-image translation is to map between input and output images by transforming an image between one domain and another. In this method, the pavement image is mapped to the ground truth image according to its structure and content. After the network is trained, every image from the test set that enters the network will be mapped to the structure of ground truth images. In the image that is mapped to the ground truth image structure, the pixels associated with the crack are isolated.

The network used was similar to UNIT, in the sense of an unsupervised image-to-image translation. UNIT is a GAN network model which includes a generator and two discriminators [30]. However, in the current research, one generator and one discriminator have been used because the goal was one-way mapping. The generator model in this method consists of a network with U-Net architecture which has an encoder and decoder. The encoder (Conv Block) includes the convolution layer and normalization layers and Leaky ReLU, which is the activation function. As this function has a positive slope on the side of negative values, it prevents backpropagation operation errors to allow the networks to correctly update. Equation (3) shows the Leaky ReLU function [32] as:

$$\begin{cases} f(x) = x & for\, x \geq 0 \\ f(x) = 0.01x & x < 0 \end{cases} \tag{3}$$

Among the components of the decoder (Transposed Conv Block) are a transposed convolution layer, batch normalization, dropout and ReLU. The ReLU function is equal to $f(x) = \max(0, x)$. If $x < 0$, the output will be zero and, if $x > 0$, a line with a slope of one will be produced [33]. This activation function is computationally efficient and can quickly cause the network to converge using a linear equation.
Seven encoders and six decoders form the generator. The discriminator consists of convolution, batch normalization, Zero Padding and leaky ReLU from the repeating layers. Figures 3 and 4 show the architecture of the generator and the discriminator, respectively.

In this study, Tensorflow (version 2.9.2) was used and the model was trained and tested in Google Colaboratory.

2.1. Determination of parameters and optimization:

The training parameters are initialized, then updated using the adaptive moment estimation (Adam) optimizer, which is first-order gradient-based optimization, and finally are replaced with their previous values [34]. The mini-batch size is a subset of the training data that are entered into the network for training in each epoch. The mini-batch size has been set as being equal to one because a large mini-batch size requires a large GPU array size.

As a result of defining loss functions, adversarial learning can effectively take place. The generator and the discriminator are updated using the loss functions and then compete in a minimax game which will result in production of high-precision images. The loss in this research is calculated through the functions for G and D. For the G model, a loss is a binary cross-entropy loss of the generated images and an array of ones. A second loss function is also available for the generator (Equation 4), which is the mean absolute error between the generated image and the ground truth image. In order to determine the loss function for the generator, these two functions are summed.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} abs\left(Ground\ truth - Generated\ image\right) \qquad (4)$$

The first part of the discriminator loss includes binary cross-entropy of the real images and an array of ones, as these are the real images. The second part is for generated images which is a binary cross-entropy loss of the generated images and an array of zeros. The discriminator loss function was obtained by summing these two parts [29,31].

2.2. Datasets:

Two common datasets were used for training and testing: Crack500 and CFD. Crack500 contains 500 images with a resolution of 2560×1440 [35] and CFD includes 118 images with a resolution of 480×320 [36]. After providing the concatenated images, the images were shuffled and each dataset was randomly split into training data (60%), validation data (10%) and test data (30%). Balancing training data for effective learning because of crack complexity, the number of images of the dataset, and assessing model performance on unseen test data were key considerations for dataset division. The test set for CFD comprises 35 images, and for Crack500, it consists of 150 images. The evaluation of this model's performance relied on the test images.

3. Results and Discussion

After preparing the network and dataset, the training data images were entered into the generator and the discriminator in pairs to train the style. The generator and the discriminator competed with each other to produce the desired result. The average training time required for the network for the Crack500 and CFD datasets was about one hour. GPU was utilized for training and the average test duration for each image was 0.29 s.

To evaluate the performance of the developed model, the generated images were compared with the ground truth images after forming a confusion matrix. This is a 2×2 matrix that considers the classes of pavement “cracks” and “non-cracks”. Table 1 shows the confusion matrix and the definition of its arrays. The confusion matrices for each dataset are shown in Tables 2 and 3, respectively.

The evaluation of the model in this research was pixel-wise. In most previous studies, each image was divided into a certain number of tiles and each tile was evaluated separately. This method can result in higher accuracy, but pixel-wise evaluation is a much more reliable method. The evaluation criteria are defined in Equations 5-8 as follows:

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$F1-score = \frac{2*Precision*Recall}{Precision+Recall} \tag{8}$$

Most of the pixels in pavement images relate to the pavement, and cracks account for a very small share of pixels. Due to the fact that the number of TN pixels is much greater than the number of pixels in the other parts, the accuracy is very close to 100 and is considered optimistic while the other three criteria are more reliable. Table 4 summarizes the evaluation results of the model for each dataset. CFD dataset scored higher on all criteria than Crack500 dataset, which can be explained by the complexity of images in Crack500 dataset. Considering the performance evaluation and the time spent during testing, this method can be considered fast and accurate for crack detection of asphalt pavements. Also, considering that the preparation of the dataset does not require special preprocessing and the image does not require post-processing after the test, it can be considered a relatively simple method. The duration of training depends directly on the hardware used in the training process. Table 5 presents a comparison of the results from this study with various pixel-wise segmentation methods that used CFD and Crack500 datasets in their research. Among the results, this research demonstrates the highest recall and F1-Score performance for CFD, with precision ranking second. Regarding the Crack500 dataset, recall performance is ranked first, while precision and F1-score rank second. The lower precision stems from misclassifying pavement pixels as crack pixels, often influenced by the presence of oil stains and shadows. Conversely, higher recall signifies a low false negative (FN) value, indicating minimal instances where crack pixels are incorrectly labeled as pavement pixels. Overall, this method proves to be reliable and accurate. By leveraging more powerful hardware and employing deeper networks for the generator, further enhancements in performance can be achieved.

Figure 5 and Figure 6 show examples of an input image, the generated result and the ground truth image from the Crack500 and CFD datasets, respectively.

4. Conclusion

In this study, generative adversarial networks (GANs) and image-to-image translation were used to detect and separate asphalt pavement crack images. In this approach, during the training phase, the model acquires the ability to convert the input image (depicting pavements with cracks) into instructions for generating the target image (ground truth image). Through this transformation, the model detects and segments the pixels corresponding to cracks. The original images and the ground truth of the training data were simultaneously entered into the generator in order to train the model, and based on the target structure, were used to detect cracks in other images. The images produced in the generator were fed into the discriminator to be evaluated. The generator and the discriminator then competed to produce the best result. The built model did not require time for labeling and dataset preparation. The performance of this model was evaluated using two datasets including CFD and Crack500 and it was found that it performed well, with an average precision of 85.76%, recall of 89.81% and F1-score of 87.72%.

Because of its high accuracy and speed, the proposed model can be implemented on different datasets. The method was found to be a reliable method for crack detection of asphalt pavements and could be useful for pavement management systems.

There exists a trade-off between the number of network layers and the time required for running and testing. Deeper networks can enhance model performance but also extend the duration of training and testing. This study aimed to devise a swift and precise method for crack detection. By leveraging robust hardware, improved and expedited performance can be attained.

It is important to note that this model was specifically trained and developed for crack detection and segmentation. With an extensive dataset containing ground truth data encompassing various types of pavement distress, this model is trainable. Our objective is to pursue this avenue for future research.

5. References

1. Lekshmipathy, J., Samuel, N. M., and Velayudhan, S., "Vibration vs. vision: best approach for automated pavement distress detection", *International Journal of Pavement Research and Technology*, **13**(4), pp. 402–410 (2020). https://doi.org/10.1007/s42947-020-0302-y

2. Ragnoli, A., De Blasiis, M., and Di Benedetto, A., "Pavement Distress Detection Methods: A Review", *Infrastructures (Basel)*, **3**(4), p. 58 (2018). https://doi.org/10.3390/infrastructures3040058

3. Shahin, M. Y., *Pavement Management for Airports, Roads, and Parking Lots* (2005).

4. Ahmadi, A., Khalesi, S., and Golroo, A., "An integrated machine learning model for automatic road crack detection and classification in urban areas", *International Journal of Pavement Engineering*, **23**(10), pp. 3536–3552 (2022). https://doi.org/10.1080/10298436.2021.1905808

5. Mohan, A. and Poobal, S., "Crack detection using image processing: A critical review and analysis", *Alexandria Engineering Journal*, **57**(2), pp. 787–798 (2018). https://doi.org/10.1016/j.aej.2017.01.020

6. Chambon, S. and Moliard, J.-M., "Automatic Road Pavement Assessment with Image Processing: Review and Comparison", *International Journal of Geophysics*, **2011**, pp. 1–20 (2011). https://doi.org/10.1155/2011/989354

7. Zou, Q., Cao, Y., Li, Q., et al. "CrackTree: Automatic crack detection from pavement images", *Pattern Recognit Lett*, **33**(3), pp. 227–238 (2012). https://doi.org/10.1016/j.patrec.2011.11.004

8. Cao, W., Liu, Q., and He, Z., "Review of Pavement Defect Detection Methods", *IEEE Access*, **8**, pp. 14531–14544 (2020). https://doi.org/10.1109/ACCESS.2020.2966881

9. Oliveira, H. and Correia, P. L., "Automatic road crack segmentation using entropy and image dynamic thresholding", *17th European Signal Processing Conference*, IEEE, pp. 622–626 (2009).

10. Zhao, H., Qin, G., and Wang, X., "Improvement of canny algorithm based on pavement edge detection", *2010 3rd International Congress on Image and Signal Processing*, IEEE, pp. 964–967 (2010).

11. Safaei, N., Smadi, O., Masoud, A., et al. "An Automatic Image Processing Algorithm Based on Crack Pixel Density for Pavement Crack Detection and Classification", *International Journal of Pavement Research and Technology*, **15**(1), pp. 159–172 (2022). https://doi.org/10.1007/s42947-021-00006-4

12. Hoang, N.-D., Nguyen, Q.-L., and Tien Bui, D., "Image Processing–Based Classification of Asphalt Pavement Cracks Using Support Vector Machine Optimized by Artificial Bee Colony", *Journal of*

Computing in Civil Engineering, **32**(5) (2018). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000781

13. Sabouri, M. and Mohammadi, M., “Hybrid Method: Automatic Crack Detection of Asphalt Pavement Images Using Learning-Based and Density-Based Techniques”, *International Journal of Pavement Research and Technology* (2023). https://doi.org/10.1007/s42947-023-00356-1

14. Shatnawi, N., “Automatic Pavement Cracks Detection using Image Processing Techniques and Neural Network”, *International Journal of Advanced Computer Science and Applications*, **9**(9) (2018). https://doi.org/10.14569/IJACSA.2018.090950

15. Chen, T., Cai, Z., Zhao, X., et al. “Pavement crack detection and recognition using the architecture of segNet”, *J Ind Inf Integr*, **18**, p. 100144 (2020). https://doi.org/10.1016/j.jii.2020.100144

16. Huyan, J., Li, W., Tighe, S., et al. “CrackU-net: A novel deep convolutional neural network for pixelwise pavement crack detection”, *Struct Control Health Monit*, **27**(8) (2020). https://doi.org/10.1002/stc.2551

17. Jiang, L., Xie, Y., and Ren, T., *A DEEP NEURAL NETWORKS APPROACH FOR PIXEL-LEVEL RUNWAY PAVEMENT CRACK SEGMENTATION USING DRONE-CAPTURED IMAGES* (2020).

18. Sabouri, M. and Sepidbar, A., “U-Net-based integrated framework for pavement crack detection and zone-based scoring”, *International Journal of Pavement Engineering*, **25**(1) (2024). https://doi.org/10.1080/10298436.2024.2308183

19. Cheng, J., Xiong, W., Chen, W., et al. “Pixel-level Crack Detection using U-Net”, *TENCON 2018 - 2018 IEEE Region 10 Conference*, IEEE, pp. 0462–0466 (2018).

20. Wang, S., Chen, X., and Dong, Q., “Detection of Asphalt Pavement Cracks Based on Vision Transformer Improved YOLO V5”, *Journal of Transportation Engineering, Part B: Pavements*, **149**(2) (2023). https://doi.org/10.1061/JPEODX.PVENG-1180

21. Zhang, R., Shi, Y., and Yu, X., “Pavement crack detection based on deep learning”, *2021 33rd Chinese Control and Decision Conference (CCDC)*, IEEE, pp. 7367–7372 (2021).

22. Jiang, Y., Pang, D., Li, C., et al. “Two-step deep learning approach for pavement crack damage detection and segmentation”, *International Journal of Pavement Engineering*, **24**(2) (2023). https://doi.org/10.1080/10298436.2022.2065488

23. Roberts, R., Inzerillo, L., and Di Mino, G., “Using UAV Based 3D Modelling to Provide Smart Monitoring of Road Pavement Conditions”, *Information*, **11**(12), p. 568 (2020). https://doi.org/10.3390/info11120568

24. Salameh, R., (Lucas) Yu, P., Yang, Z., et al., “Evaluating Crack Identification Performance of 3D Pavement Imaging Systems Using Portable High-Resolution 3D Scanning”, *Transportation Research Record: Journal of the Transportation Research Board*, **2677**(1), pp. 529–540 (2023). https://doi.org/10.1177/03611981221100239

25. Sabouri, M. and Sepidbar, A., “SUT-Crack: A comprehensive dataset for pavement crack detection across all methods”, *Data Brief*, **51**, p. 109642 (2023). https://doi.org/10.1016/j.dib.2023.109642

26. Sabouri, M. and Mohammadi, M., “A novel approach for automatic asphalt pavement crack length estimation”, *Road Materials and Pavement Design*, pp. 1–19 (2024). https://doi.org/10.1080/14680629.2024.2323517

27. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. “Generative adversarial networks”, *Commun ACM*, **63**(11), pp. 139–144 (2020). https://doi.org/10.1145/3422622

28. Guo, X., Wang, Z., Yang, Q., Lv, W., Liu, X., Wu, Q., and Huang, J., “GAN-Based virtual-to-real image translation for urban scene semantic segmentation”, *Neurocomputing*, **394**, pp. 127–135 (2020). https://doi.org/10.1109/TMM.2021.3109419

29. Isola, P., Zhu, J.-Y., Zhou, T., et al. “Image-to-Image Translation with Conditional Adversarial Networks”, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, pp. 5967–5976 (2017).

30. Pang, Y., Lin, J., Qin, T., et al. “Image-to-Image Translation: Methods and Applications”, *IEEE Trans Multimedia*, **24**, pp. 3859–3881 (2022). https://doi.org/10.1016/j.neucom.2019.01.115

31. Liu, M.-Y., Breuel, T., and Kautz, J., “Unsupervised Image-to-Image Translation Networks”, *31st Conference on Neural Information Processing Systems* (2017).

32. Dubey, S. R., Singh, S. K., and Chaudhuri, B. B., “Activation functions in deep learning: A comprehensive survey and benchmark”, *Neurocomputing*, **503**, pp. 92–108 (2022). https://doi.org/10.1016/j.neucom.2022.06.111

33. Agarap, A. F., “Deep Learning using Rectified Linear Units (ReLU)” (2018).

34. Kingma, D. and Ba, J., “Adam: A Method for Stochastic Optimization”, *International Conference on Learning Representations* (2014).

35. Zhang, L., Yang, F., Daniel Zhang, Y., et al. “Road crack detection using deep convolutional neural network”, *2016 IEEE International Conference on Image Processing (ICIP)*, IEEE, pp. 3708–3712 (2016).

36. Cui, L., Qi, Z., Chen, Z., et al. “Pavement Distress Detection Using Random Decision Forests”, pp. 95–102 (2015).

37. Shi, Y., Cui, L., Qi, Z., et al. “Automatic Road Crack Detection Using Random Structured Forests”, *IEEE Transactions on Intelligent Transportation Systems*, **17**(12), pp. 3434–3445 (2016). https://doi.org/10.1109/TITS.2016.2552248

Biographies

Alireza Sepidbar earned his B.Sc. degree in Civil Engineering from Ferdowsi University of Mashhad in 2020. He then completed a M.Sc. degree in Road and Transportation Engineering at Sharif University of Technology in 2022. His research focuses on the application of image processing and deep learning models for detecting pavement distress.

Dr. Mohammadreza Sabouri obtained both his Bachelor's and master's degrees in Civil Engineering from Sharif University of Technology. He then earned his Doctoral degree in Pavement and Materials from North Carolina State University. He is currently an Assistant Professor at Sharif University of Technology. His research specializes in imaging techniques, characterizing and modeling the performance of pavement materials, and the design, analysis, and development of pavement systems. He has published numerous papers and has been invited to present at various national and international conferences. He is a licensed Professional Engineer in Texas and Minnesota, USA.

List of Figures

List of Tables

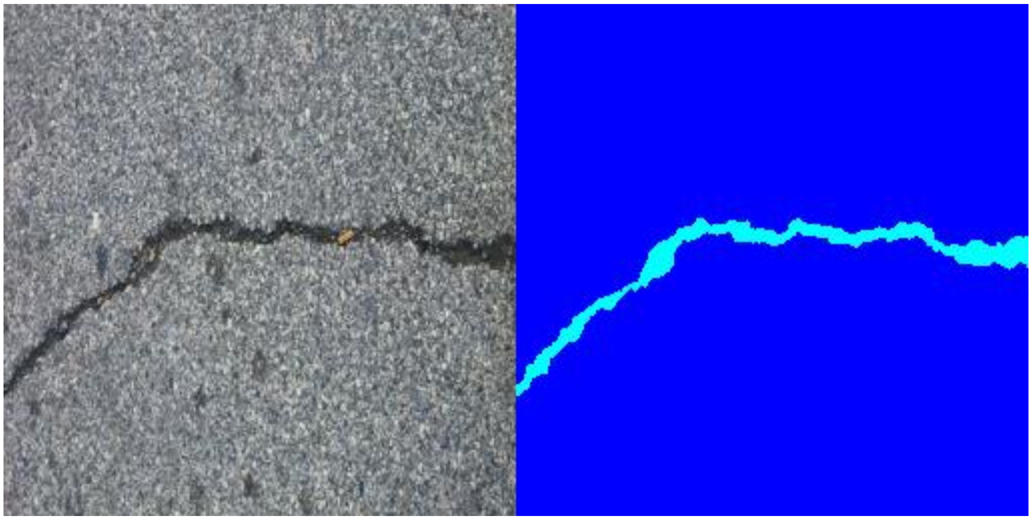

Figure 1. An example of concatenated Images

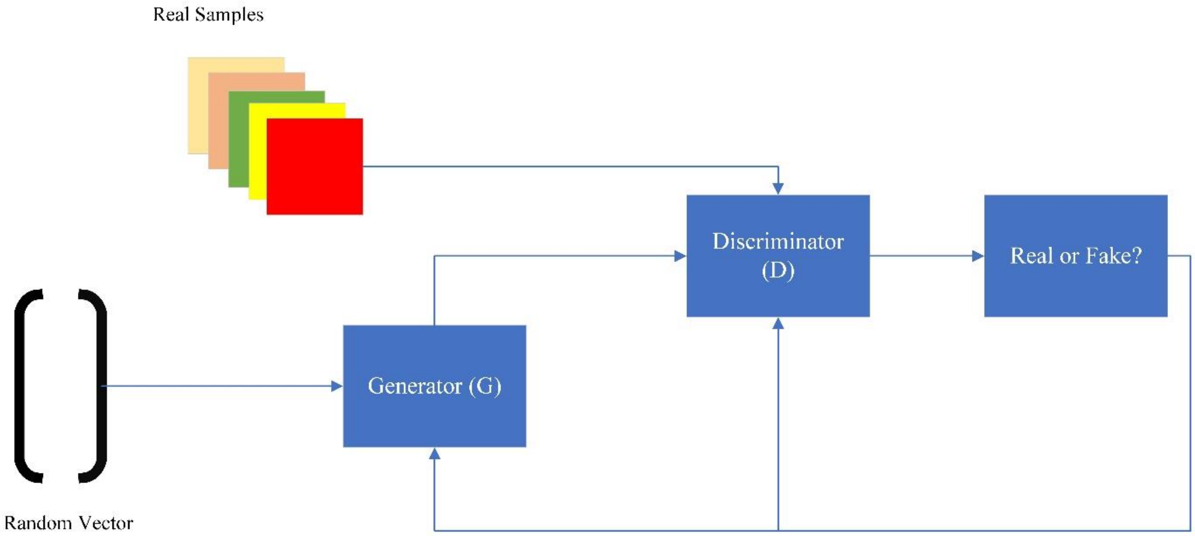

Figure 2. Concept of GAN

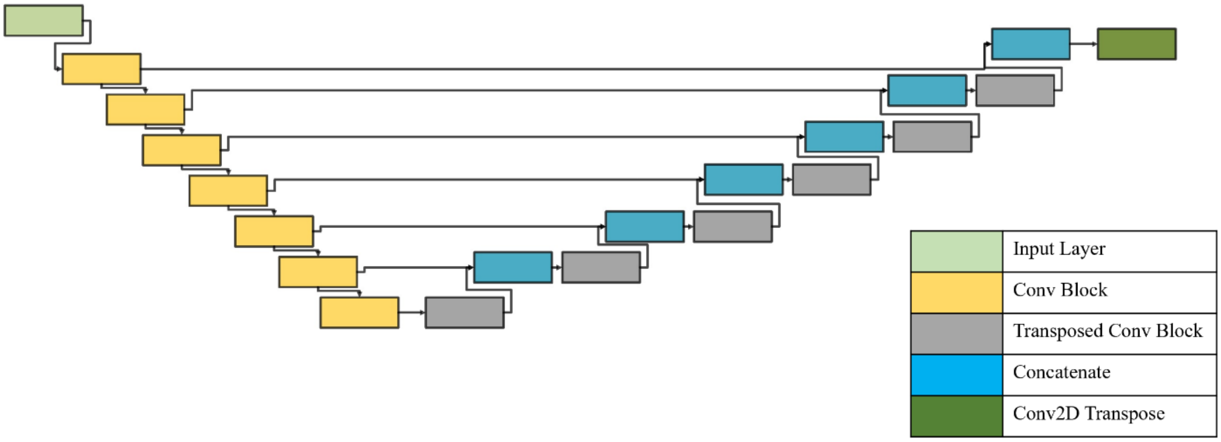

Figure 3. Architecture of generator

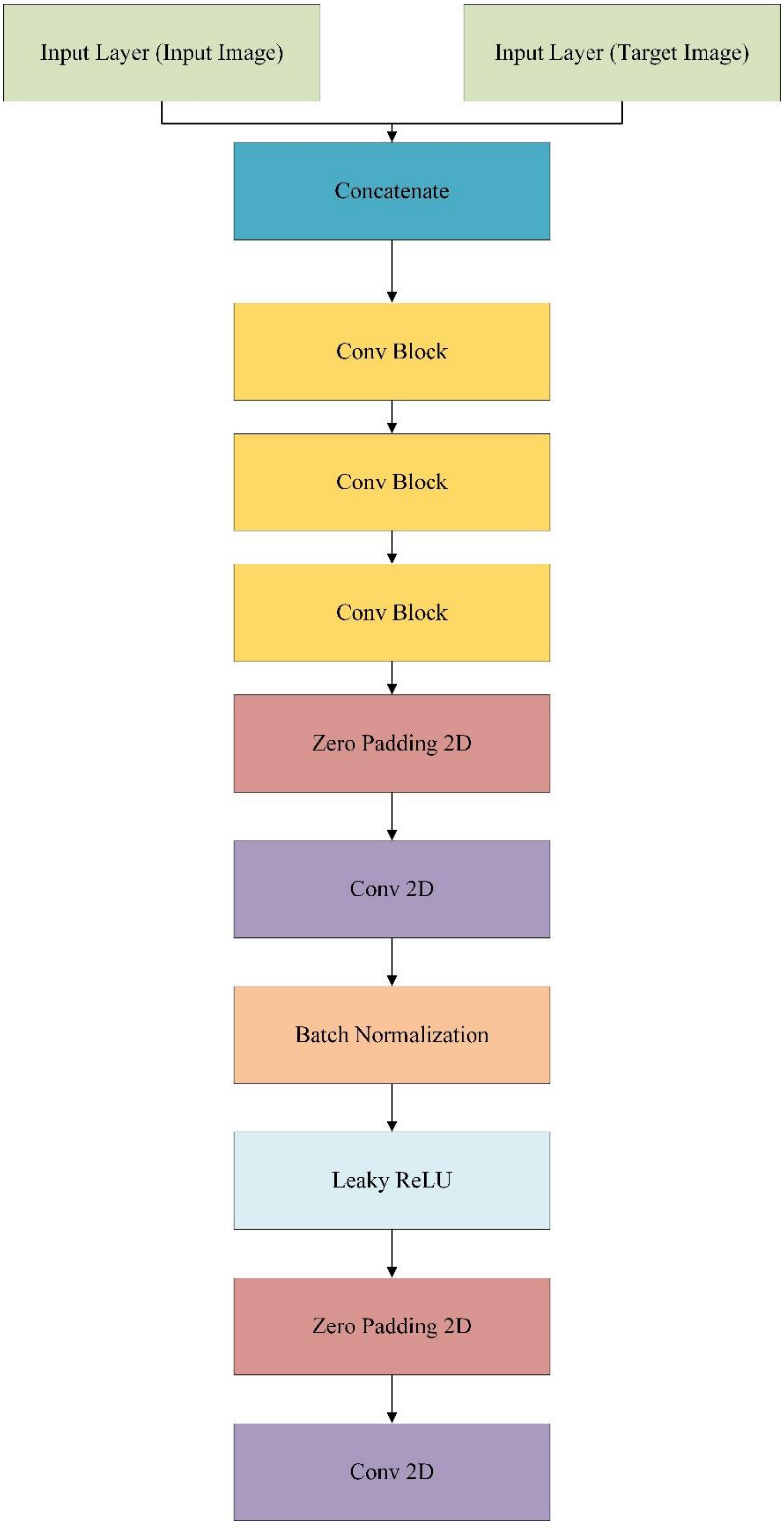

Figure 4. Architecture of discriminator

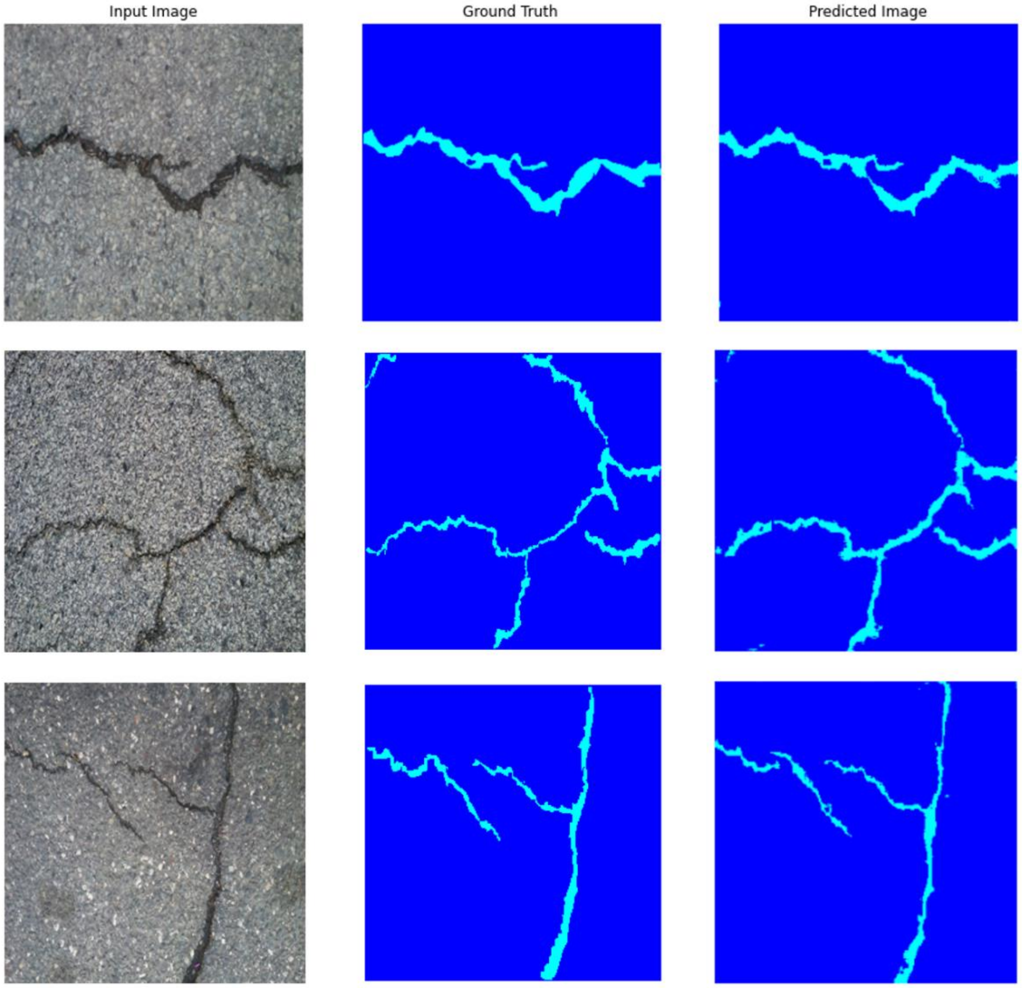

Figure 5. Examples of an input image, the generated result and the ground truth image from the Crack500 dataset

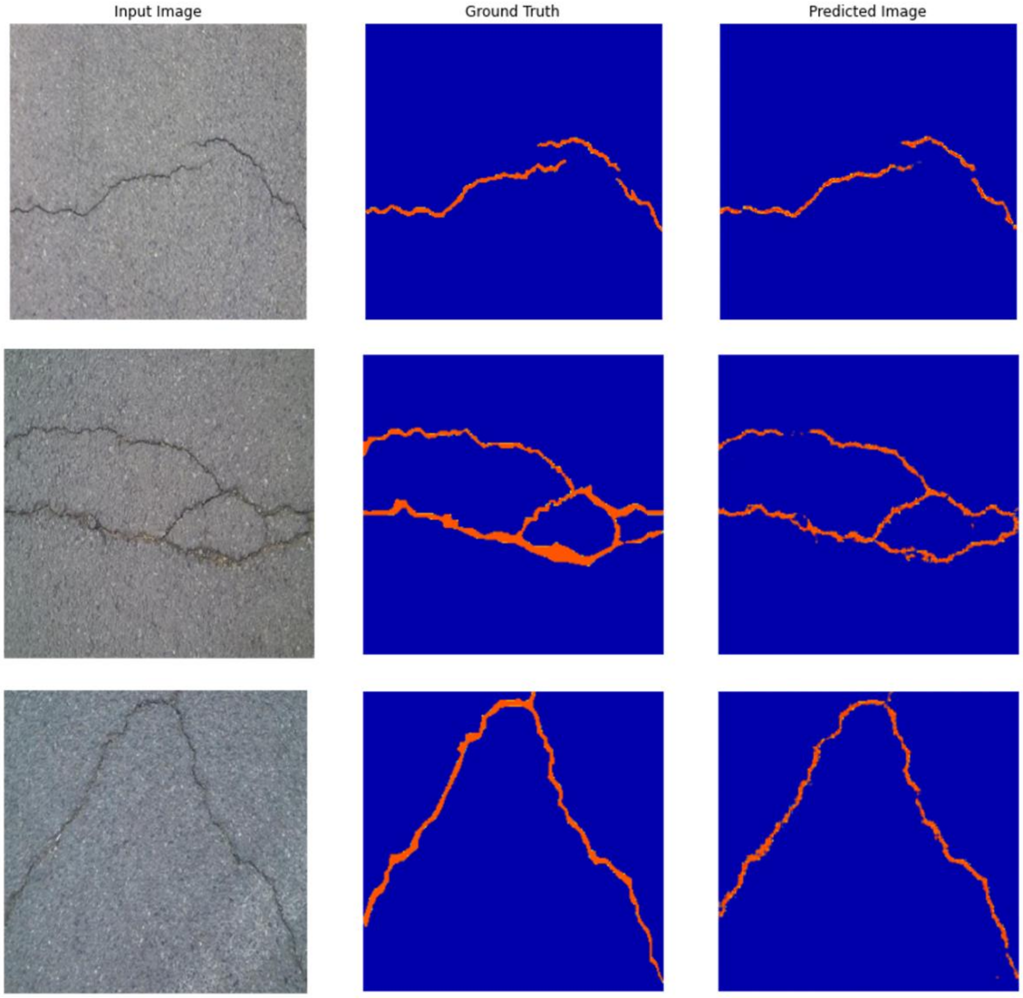

Figure 6. Examples of an input image, the generated result and the ground truth image from the CFD dataset

Table 1. Confusion matrix

	Crack	**Non-Crack**
Crack	TP	FN
Non-Crack	FP	TN

TP: The sample is a member of the crack category and is recognized as a member of the same class (True Positive).
FN: The sample is a member of the crack class and is recognized as a member of the non-crack class (False Negative).
TN: The sample is a member of the non-crack class and is recognized as a member of the same class (True Negative).
FP: The sample is a member of the non-crack class and is recognized as a member of the crack class (False Positive).

Table 2. Confusion matrix of test set of CFD

	Crack	**Non-Crack**
Crack	107880	16203
Non-Crack	7346	5244571

Table 3. Confusion matrix of test set of Crack500

	Crack	**Non-Crack**
Crack	11549092	2107483
Non-Crack	1881560	422221865

Table 4. Evaluation results of model for each dataset

	Precision (%)	**Accuracy (%)**	**Recall (%)**	**F1-score (%)**
CFD	86.94	99.56	93.62	90.16
Crack500	84.57	99.09	85.99	85.27

Table 5. Performance of different models

Dataset	Method	Precision	Recall	F1-Score
CFD	CrackIT	67.23	76.69	71.64
	Crack Forest-KNN	80.77	78.15	79.44
	Crack Forest-SVM	82.28	89.44	85.71
	Hybrid Method	**88.30**	83.66	85.92
	Our Method	86.94	**93.62**	**90.16**
Crack500	SVM	81.12	67.34	73.59
	Boosting	73.60	75.87	74.72
	Hybrid Method	**95.03**	78.64	**86.06**
	Our Method	84.57	**85.99**	85.27

Reference: [13,37]