coatings

MDPI

Article

Enhancing Pavement Distress Detection Using a Morphological Constraints-Based Data Augmentation Method

Zhengchao Xu [1,2,†], Zhe Dai [3,*,†], Zhaoyun Sun [1,†], Chen Zuo [3,*], Huansheng Song [1] and Changwei Yuan [3]

[1] School of Information Engineering, Chang'an University, Xi'an 710064, China
[2] School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China
[3] College of Transportation Engineering, Chang'an University, Xi'an 710064, China
* Correspondence: zhedai@chd.edu.cn (Z.D.); chenzuo789@outlook.com (C.Z.)
† These authors contributed equally to this work.

Abstract: Pavement distress data in a single section usually presents a long-tailed distribution, with potholes, sealed cracks, and other distresses normally located at the tail. This distribution will seriously affect the performance and robustness of big data-driven deep learning detection models. Conventional data augmentation algorithms only expand the amount of data by image transformation and fail to enlarge the data diversity. Due to such a drawback, this paper proposes a novel two-stage pavement distress image augmentation pattern, in which a mask is generated randomly according to the geometric features of the distress in the first stage; and in the second stage, a distress-free pavement image with the fused mask is transformed into a pavement distress image. Furthermore, two convolutional networks, M-DCGAN and MDTMN, are designed to complete the generation task in two stages separately. In comparison with other generation algorithms, the quality and diversity of the generation results of proposed algorithms are better than other algorithms. In addition, distress detection tests are conducted which indicate that the expanded dataset can raise the IoU from 48.83% to 83.65% at maximum, and the augmented data by the proposed algorithm contributes more to the detection performance.

Keywords: pavement image augmentation; long-tailed distribution data; image generation; pavement distress detection

Citation: Xu, Z.; Dai, Z.; Sun, Z.; Zuo, C.; Song, H.; Yuan, C. Enhancing Pavement Distress Detection Using a Morphological Constraints-Based Data Augmentation Method. *Coatings* **2023**, *13*, 764. https://doi.org/10.3390/coatings13040764

Academic Editor: Valeria Vignali

Received: 8 March 2023
Revised: 6 April 2023
Accepted: 10 April 2023
Published: 13 April 2023

1. Introduction

Detection networks based on deep convolutional neural networks have become the most popular algorithms among researchers in the area of pavement distress detection [1–12]. With the development of deep learning theory and the improvement of computer hardware performance, the depth and breadth of detection networks have been increasing to achieve superior accuracy, along with a rapid increase in the number of parameters. Currently, the most powerful Transformer-based neural network has reached 600 million parameters [13]. In order to avoid overfitting and to obtain a robust model, the training data must be sufficient in quantity and diversity. There are many strategies to prevent overfitting during the training of the detection model, such as batch normalization [14], drop out, drop connect [15], early stopping, weight decay, etc. There is another more effective and intuitive strategy that can be implemented before model training, which is the data augmentation, but the current data augmentation algorithm is very effective for datasets with balanced data distribution and has limited improvement for datasets with "long tail" distribution.

There are various pavement distresses on a road section; however, the quantity difference between pavement distress is often significant. For example, while the number of the common crack category is high, the number of sealed cracks or potholes is relatively small, and the data distribution of the distress shows a long-tail distribution, which is a

kind of unbalanced data performance. As shown in Figure 1, the frequency of different categories of distress is used as the basis for ranking, from high to low, with the head of the curve being the most common distress and the tail being the distress that occurs less frequently, forming a curve similar to a "long tail" shape. As the detection model undergoes more training epochs, it eventually converges on the training set losses. However, this improvement does not extend well to the test set, as it exhibits weak generalization and performs poorly, particularly for tail data.

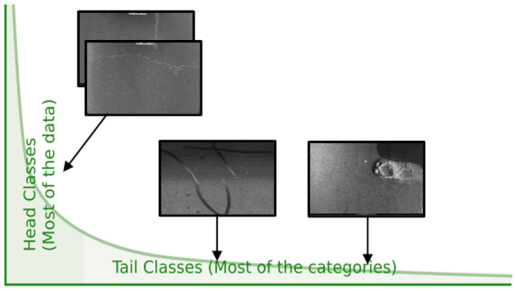


Figure 1. Long-tailed distribution of pavement distress data on a single section.

To improve the problem of long-tailed datasets, the most intuitive means of tackling the problem is to augment the tailed data. The conventional methods are to apply affine transformations to the data, such as rotation, translation, mirroring, etc., or to change the image hue, add Gaussian noise [16] or image blurring, in addition to other means of augmentation [17], all of which can expand the number of tail data and bridge the problem of insufficient samples in the tail. However, conventional methods for data augmentation require the reuse of all or part of the image content of the original data, and do not introduce new perspectives or content, with limited improvement in intra-class diversity for the tail data [18,19]. Therefore, this enhancement algorithm is more suitable for head data, or datasets with more balanced data distribution. However, for tail data, although the quantity can be boosted using such methods, the diversity enhancement is rather weak and cannot effectively avoid model overfitting.

Deep learning-based image generation algorithms can generate random noise into a specified category of images or edit certain features of an image, and this kind of uncertain image generation can enhance the diversity of images. At present, the dominant image generation algorithms are VAE [20], GAN [21], Diffusion Model [22], and others. Some scholars have carried out research on the pavement image augmentation based on the above algorithms and have attained some progress, but there are still some drawbacks. For example, the generated image quality is poor with low resolution, the training is difficult to converge, the model collapses, or the complexity of the model is too high.

In this work, we propose a novel two-stage approach for pavement distress image augmentation, as shown in Figure 2. In this approach, in the first stage, we propose a self-attentive mechanism-based distress mask generation network, M-DCGAN (Mask-DCGAN), for learning distress features from a small number of labelled distress images and autonomously generating synthetic distress masks that fit the distress geometries. In the second stage, the mask images generated in the first stage are fused with the distress-free pavement images to become prototype images for pavement distress image generation. In addition, a texture synthesis network with multiple receptive fields, MDTMN (Mask-to-Distress Texture Mapping Network), is designed to generate a prototype image into a synthetic image with relatively realistic pavement distress texture.

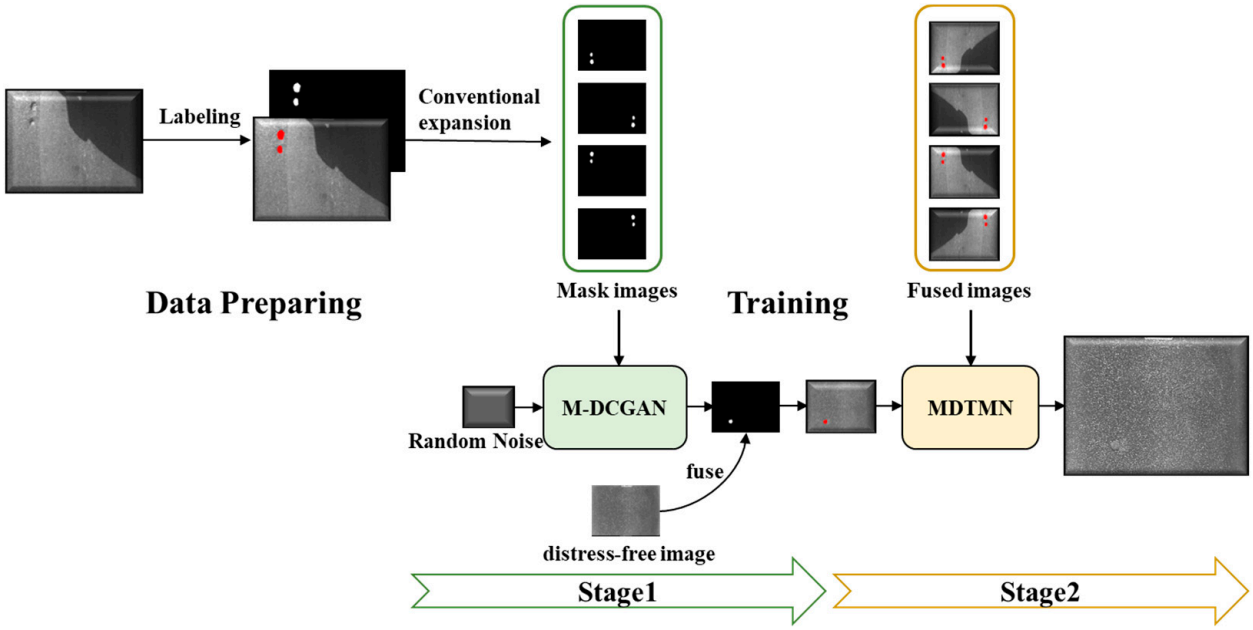


Figure 2. Flowchart of a two-stage pavement distress image augmentation.

To demonstrate the effectiveness of the proposed MDTMN, three different types of generation algorithms, namely Pix2Pix [23], CPN [24], and CycleGAN [25], are selected for comparison experiments, which are divided into two parts: qualitative and quantitative evaluations. The comparison experiment verified that the images generated by the proposed method outperformed those generated by other algorithms in terms of image quality and diversity.

Furthermore, to prove the effectiveness of the data augmentation model proposed in this paper, pixel-level detection of pavement distress was conducted for the tail data. The experiments were carried out with the dataset not augmented, the dataset augmented by the conventional method, and the dataset augmented by the proposed method, and finally, the detection model achieved the best detection accuracy on the dataset augmented by the proposed algorithm.

The contributions of this paper are summarized as follows:

- We propose a two-stage pavement distress augmentation approach and accompany each stage with a specific task generation algorithm (M-DCGAN and MDTMN). The M-DCGAN generates a mask that matches the geometric characteristics of the distress in one stage, and in the second stage the generated mask will be filled by the MDTMN with a context-sensitive distress texture. This two-stage pavement distress image generation algorithm allows the generation of high-quality images while enhancing image diversity;
- Our proposed image augmentation algorithm can simultaneously obtain a distress mask image with pixel-level labelling (one-stage output) and a corresponding pavement distress image (two-stage output), which can be utilized directly as a dataset for pavement distress semantic segmentation algorithms. It reduces the cost of manual labelling and provides more accurate labelled data as well;
- The proposed method focuses on diversifying the original dataset, which effectively improves the performance of the detection model.

2. Related Work

The size of network structures in novel deep learning networks is constantly being increased to extract more effective features, resulting in a significant increase in the number of parameters. In comparison with previous small-sized neural networks, the new struc-

tures require more data to be involved in the training. Therefore, how to perform effective data augmentation has also become an active research issue in recent years.

In previous studies, the common means used were random rotation, translation and mirroring, which were used to enhance the geometric invariance of the CNN-based models by applying affine transformations to the images [26]. Based on the above approaches, Auto-Augment [27] uses reinforcement learning algorithms to learn how to combine basic image augmentation algorithms to form more complex image augmentation strategies. The most critical drawback of this combination of algorithms for data augmentation strategies is the high computational cost of the search combination. Inspired by the drop out strategy in the model training, an augmentation strategy by masking some of the information in the image can also improve the robustness of the model [28].

The YOLO family [29–31] of detection algorithms has been well known to researchers for its fast and accurate detection, not only in terms of network structure, but also the accompanying data augmentation algorithms. YOLOv4 introduces the Mosaic data augmentation method, which combines four different images into one large image and then splits the large image into four parts as part of the training set. YOLOv5 introduces the Cut-Mix data augmentation method, which overlays a part of one image onto another to create a new hybrid image. This method expands the training set and can better handle inter-target occlusions, improving the robustness of the model. Both YOLOv4 and YOLOv5 use the Mix-Up data augmentation approach to generate a new training image by randomly selecting two training images and mixing their pixel and label information in a certain ratio. This approach can extend the training set and improve the detection accuracy of the model for small targets.

The output of the above image data augmentation methods all contains part or entire representations of the original data, and the diversity of the inter-class data is not augmented. Considering the above drawbacks, the researcher used a GAN-based image generation algorithm to generate synthetic images that conform to the prior distribution by learning the feature distribution of known samples [32–35]. The generation of synthetic images is random in nature and can be used as supplementary data to enhance the diversity of the dataset. However, the drawbacks are also significant; the quality of the generated images is poor, the image size is too small, model collapse tends to occur when training is inadequate, and the representation of synthetic images does not conform to natural patterns. Training with such synthetic images does not effectively improve the accuracy and robustness of the detection model. Despite this, there are already many algorithms that can generate high quality images, such as Big-GAN [36] and diffusion model. However, these algorithms generally require high-performance computers, especially in terms of requiring high numbers of high-performance GPUs to support training and are not cost-effective if only used for data augmentation tasks. In addition, these algorithms require large amounts of high-quality training data, which is in conflict with the original purpose of image data augmentation. Most importantly, the time to generate a single high-quality image often exceeds 10 s, which is not suitable for high-volume data augmentation. This paper addresses the shortcomings of these algorithms and redesigns the image generation model to obtain better quality augmented data that can be used directly for training.

3. Dataset

The pavement distress data in this paper comes from the dataset HRSD in our previous work [37], with a total of 3200 pavement images in the entire dataset, of which 850 are crack images, 180 are sealed cracks, only 102 are potholes, and the rest are distress -free images, with a long-tailed distribution of data. In this paper, we select the potholes images at the tail end as the object of augmentation and carry out the corresponding algorithm design.

Image Labeling

The semantic in the asphalt pavement images is mostly the pavement context, and the distress semantic only a minority. In this subsection, the pavement semantic is manually

annotated to separate the pavement semantic and the distress semantic in the pavement distress image, and the first stage training data—mask images—and the second stage training data—fused images—are generated simultaneously. As shown in Figure 3, the image of the pavement distress will be fed into the image annotation tool and the distressed area will be manually annotated as a single-color block, with no additional annotation of the pavement.

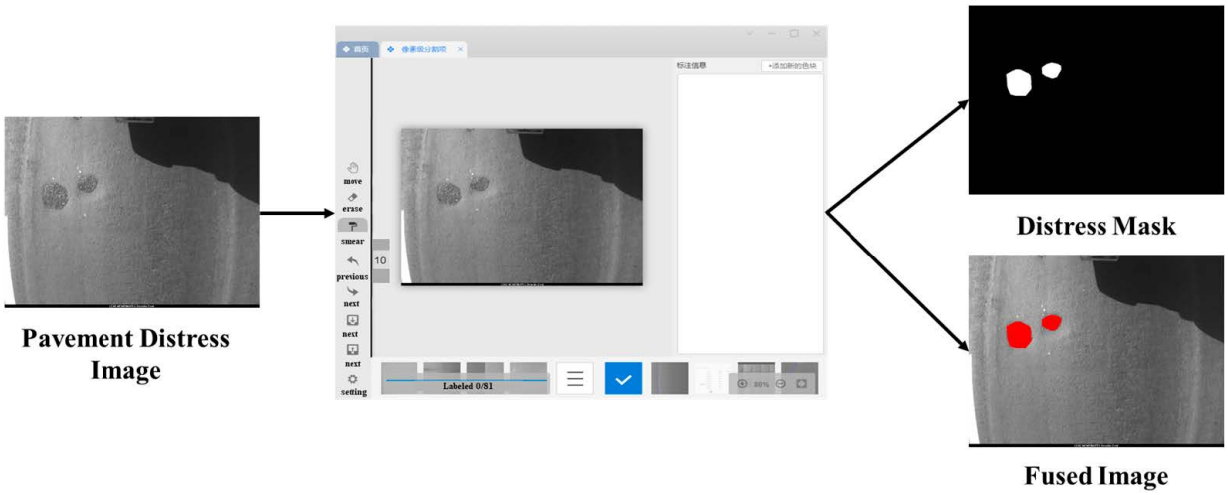


Figure 3. Pothole images labelling procedure.

The 102 pothole images are divided into an initial training set, an initial validation set, and an initial test set in the ratio of 7:2:1. Nevertheless, the number of pothole images is insufficient for the model training. Therefore, this paper adopts the conventional image augmentation algorithm, which transforms the images of potholes, using rotation, translation, mirroring, scaling, and random erasing to expand the initial training sets only, as shown in Figure 4. After data augmentation, the number of images in the training set increased from 70 to 600.

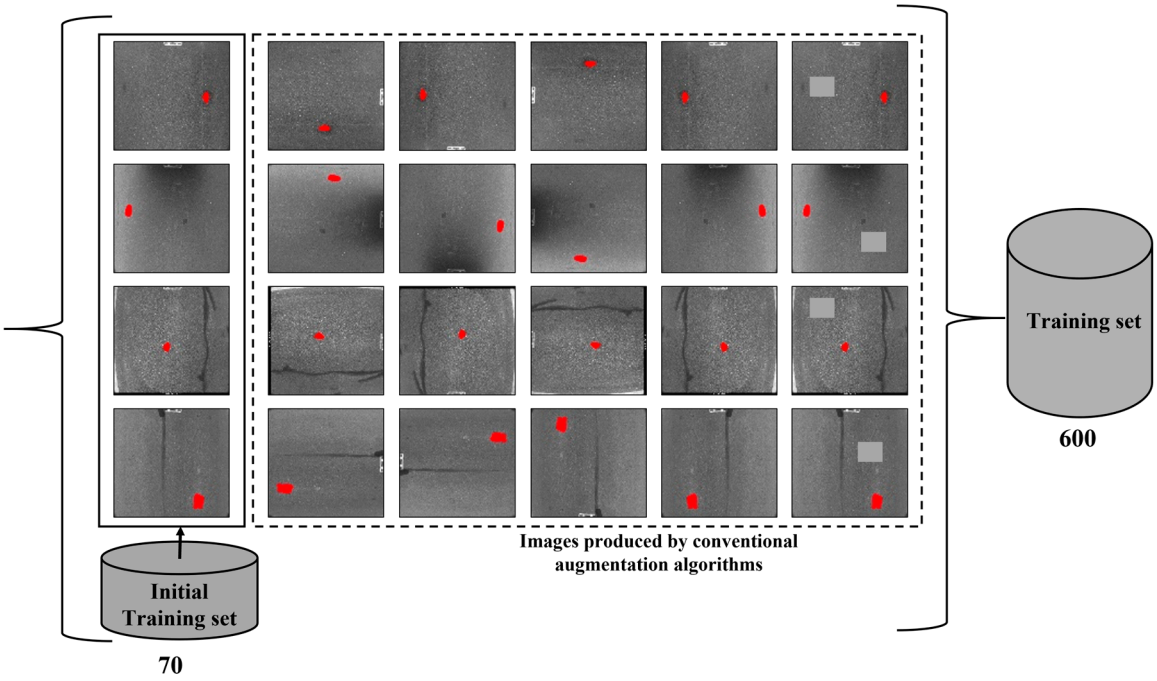


Figure 4. Training set after augmentation by the conventional algorithm. The red marks illustrate the labeled pit area.

4. Two-Stage Augmentation Approach and Algorithms

This paper divides the complex pavement distress image enhancement procedure into two stages, and in each stage the relevant algorithm is designed to complete the corresponding processing task. As shown in Figure 2, this section describes in detail the algorithms designed in each stage.

4.1. Pavement Distress Mask Generation Network (M-DCGAN)

In the first stage, a self-attention-based generative adversarial network, M-DCGAN, is designed for distress mask image generation in this study. Since the training data of M-DCGAN is a simple pavement distress mask image, it will not be disturbed by other approximate semantics in the pavement image.

4.1.1. An Improved Generator and Discriminator

M-DCGAN is based on the DCGAN framework and is designed to generate two-dimensional random noise matrices into distress mask images. DCGAN is an image generation algorithm that employs unsupervised representational learning with a combination of deep convolutional neural networks and generative adversarial networks internally, as shown in Figure 5. It is an improved algorithm to the vanilla GAN and can output better high-quality images. However, the following drawbacks remain:

- Limited number of transposed convolutional layers and only images sized 64 × 64 can be generated;
- Correlation between channels in the feature map is not fully exploited;
- Gradient disappearance still occurs during training.

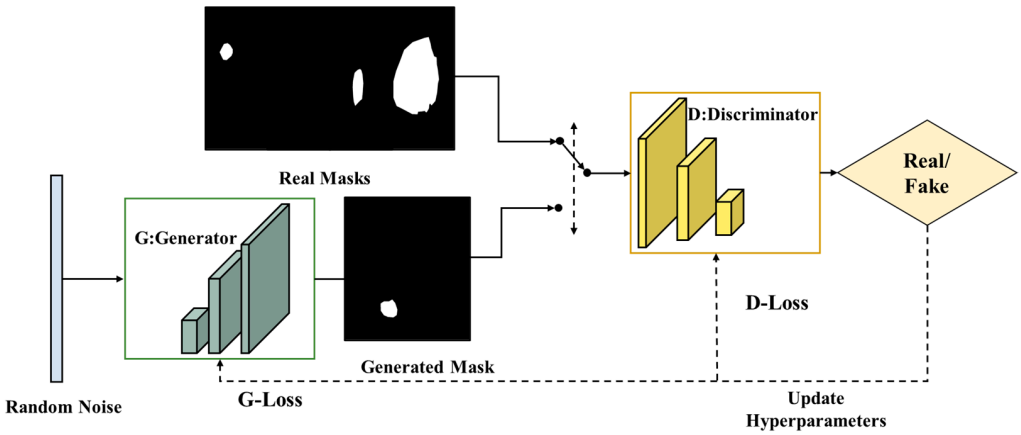


Figure 5. DCGAN architecture.

To address the above drawbacks, this paper makes a slight modification to DCGAN for the task of distress mask generation by redesigning the network structure and loss function to build Mask-DCGAN (M-DCGAN). The redesigned generator and discriminator are shown in Figure 6a,b. In the generator network, the green part is the network structure of the original DCGAN. The random noise is transformed into a feature map of size 1024 × 4 × 4 by transposed convolution. The final image of size 128 × 128 is generated by transposed convolution with step size 2, transposed convolution kernel size 4 × 4 and padding operation 1, which completes the step-by-step scale-up operation of two times the original size. The output size of the transposed convolution is calculated from Equation (1):

$$H_{out} = \text{stride} \times (H_{in} - 1) + \text{kernel} - 2 \times \text{padding} \quad (1)$$

where stride and kernel denote the step size and convolution kernel size, respectively.

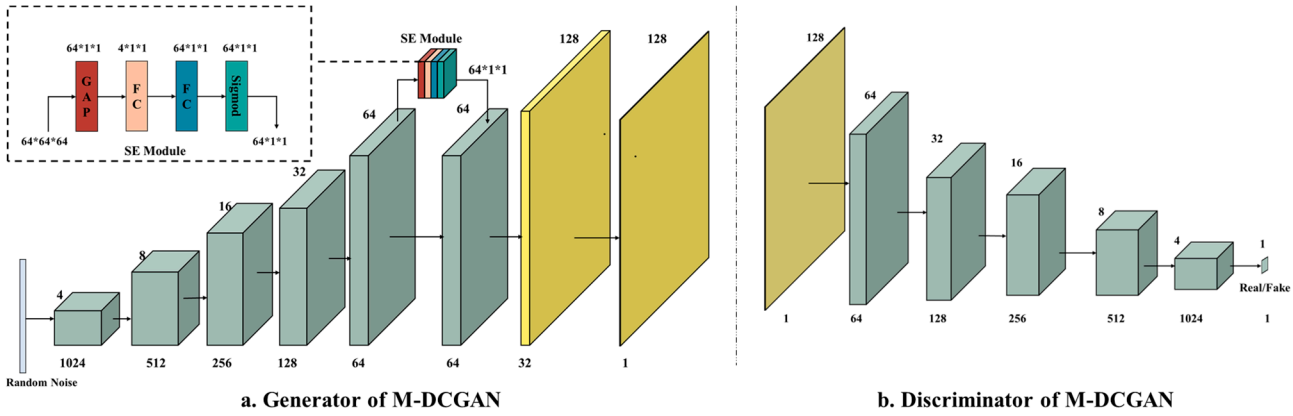


Figure 6. M-DCGAN architecture.

As illustrated in Figure 6a, a self-attention-based SE module is added to the M-DCGAN generator to exploit the weight relationships of different channels in the features. This module, which is placed between the fifth transposed convolutional layer and the sixth, can autonomously calculate the weight coefficients of each of the 64 channels and then perform a weighted dot multiplication operation with the feature map to assign the weights to each channel, enhancing the features with high contributions and weakening the irrelevant features.

As illustrated in Figure 6b, as the size of the generated image was increased by a factor of two, the discriminator was adapted accordingly, increasing the size of the input image to 128 × 128 pixels, while the trailing fully connected layer was replaced with a convolutional layer.

4.1.2. Model Training

The deep learning models involved in this paper are trained on a GPU server with the following configuration: CPU: Intel Xeon, GPU: Rtx2080 × 8, 64 G of RAM, 5 TB of hard disk, and Ubuntu 14 as the operating system. The model was trained using the WGAN (Wasserstein GAN) loss function to address training instability and to avoid model collapse. The rest of the configuration items are as follows: the optimizer is Adam, the learning rate is set to 0.0002, the cut-off parameter is set to 0.01, the batch size is set to 64, the number of rounds of discriminator training per batch is set to 5, and the overall number of training epochs is set to 200.

In order to compare the WGAN loss function and vanilla GAN loss function, the two loss functions are used for training, as shown in Figure 7. The Figure 7a shows the loss function of vanilla GAN, and the generator loss and discriminator loss have not changed since the first few rounds, and the training is basically at a halt, which is due to the disappearance of the gradient during back propagation. In Figure 7b, the loss function of WGAN is used, and the loss curve oscillates during training but conforms to the training rule of generative adversarial network, and the loss also tends to converge. What is more concerning is that the gradient passed in the network is always maintained and the training does not stagnate. In addition, the images generated by the model with the same number of iterations can also reflect the training effect of the model. With respect to the images in the red box in Figure 7a, it can be seen that the background and foreground are not generated correctly, and still present a random noise appearance, while the green dashed box in Figure 7b shows that the background has all been generated into a black background, and the distress masks of different shapes and sizes in the foreground have been generated properly.

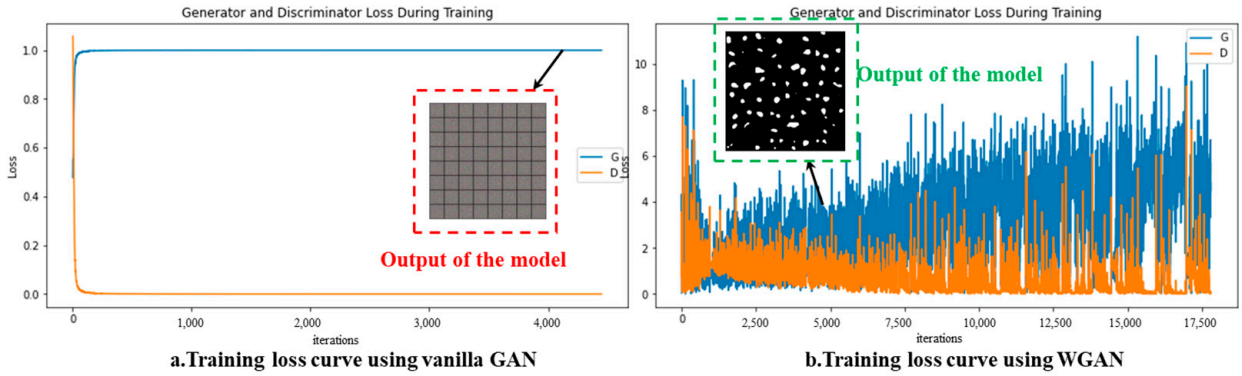


Figure 7. Training loss curve using GAN vs. WGAN. (**a**): The training process of M-DCGAN with original GAN. (**b**): The training process of M-DCGAN with WGAN.

4.1.3. Evaluation of the Generation Results of M-DCGAN

A batch quantity of random noise can be generated into the same number of distress mask images using the trained M-DCGAN model. In order to show the complete distribution of the generated images of a batch and to facilitate the evaluation of the generated results, both the generated images and the training data will be presented and evaluated in an 8 × 8 patchwork, as shown in Figure 8. It can be observed from Figure 8 that after the 15th epoch, the generated distress masks are already very close to the real masks, but after 100 epochs, the angularity of the masks gradually increases and image artefacts appear, which is due to model over-fitting.

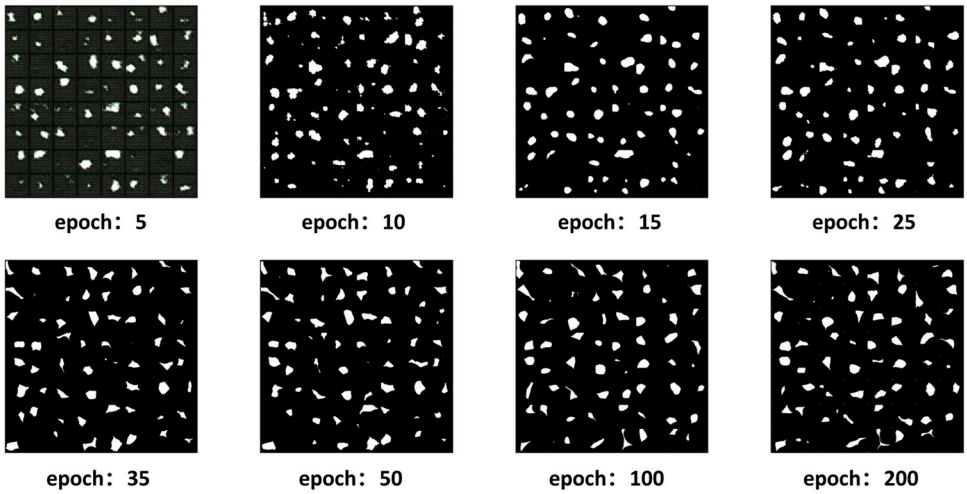
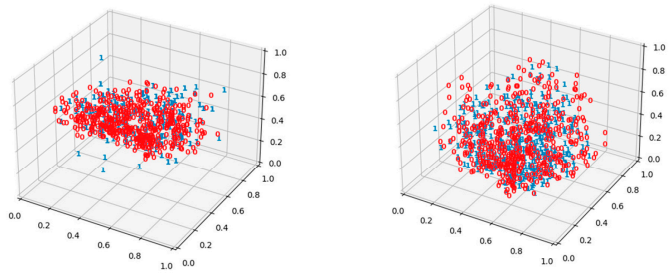


Figure 8. Pothole mask images generated by increasing training epochs.

Generative adversarial generative network itself is an unsupervised learning algorithm, and its generation results are not unique. In addition, the real pavement pothole morphology is not constant, and it is difficult to evaluate the generation results with quantitative metrics. To verify whether the distribution of M-DCGAN generated images and training samples are close, this subsection uses 500 generated images and 500 real mask images to form a validation dataset, and assigns "0" and "1" labels, respectively, and then uses t-SNE [38] (T-distributed Stochastic Neighbor Embedding). The visualization results are shown in Figure 9. The two types of data have an overall distribution that is similar to each other, which proves that the M-DCGAN generator can effectively learn the image distribution of the distress masks and generate the images with similar distribution and appearance of the distress masks. Moreover, to verify the effectiveness of the SE module, Figure 9a,b show the distribution of the generated results with and without the SE module,

respectively. Evidently, the generated results of the model with the SE module are closer to the real data.



(a)M-DCGAN **(b)M-DCGAN(w/o SE Module)**

Figure 9. t-SNE visualization of the morphological distribution of generated images and real mask images.

4.2. Mask-to-Distress Texture Mapping Network (MDTMN)

The most of pavement images are distress-free pavement images, and such images are usually eliminated from the dataset before the training of detection models, which will result in wasted cost of collection and storage. The MDTMN designed in this paper can effectively use distress-free images or non-target class pavement distress images to generate images with the specified class of distress.

4.2.1. Input Data of MDTMN

The mask generated by M-DCGAN needs to be fused with the distress-free image before it can be used as an input for distress texture generation, because the distress mask image and the original image have the same size. The fusion of the two images is to map the coordinates of the white pixels in the distress mask region to the original image, and then replace the pixels in the corresponding coordinate region of the original image with red pixels with pixel values of (255, 0, 0), as shown in Figure 10. The reason for using pure red pixels is to distinguish them from the original pixels in the image and highlight the distress region to be generated.

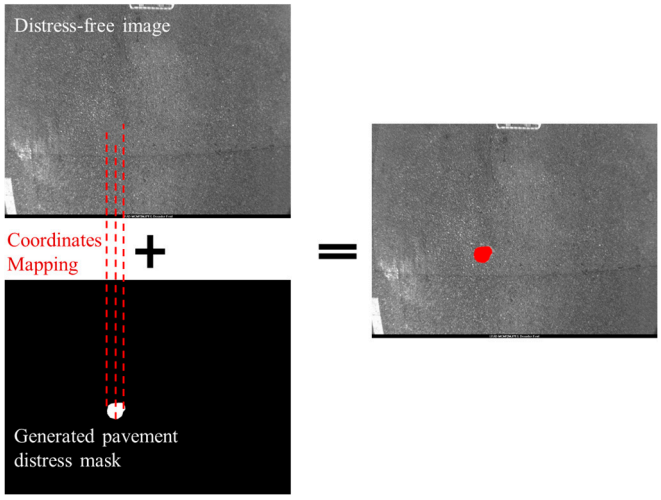


Figure 10. Image fusion procedure. The red mark shows the generated pit mask.

4.2.2. MDTMN Architecture

The overall structure of MDTMN is illustrated in Figure 11. At the input of MDTMN, the mask fusion image is channel fused with the mask image to form a four channels image, and then sent to the encoder for feature extraction and compression. After compression,

the features are extracted at multiple scales under different receptive fields by the dilated convolution group in the middle section, and the extracted features are concatenated and sent to the decoder again for level-by-level feature and image size recovery. In order to ensure that the generated image can maintain more image details, skip connections are also added in the first and third convolutional layers of the network to introduce more low-level semantic information into the high-level features. The final output is a pavement image of the masked area generated as pavement distress.

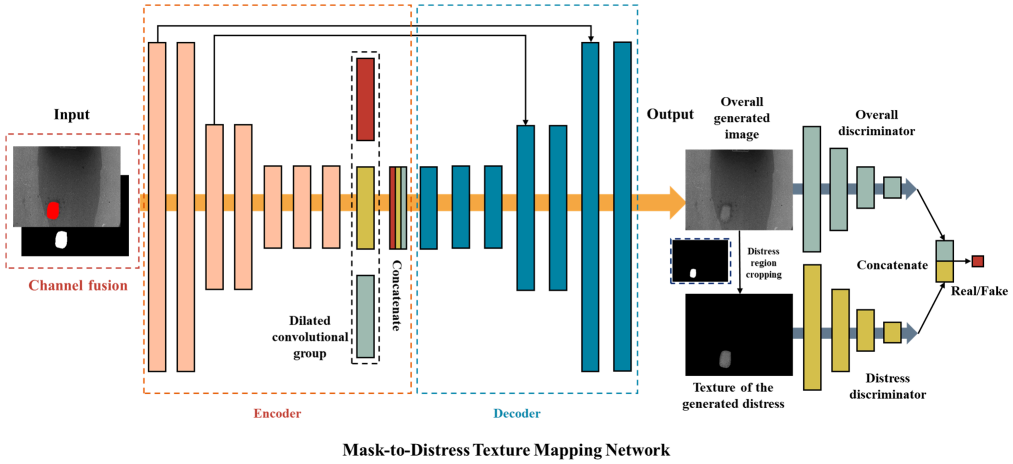


Figure 11. The Overall Architecture of MDTMN.

The discriminator is divided into two sub-discriminators: the overall discriminator and the distress discriminator, respectively. They extract the features of the complete generated image and the distress image with only the distress area simultaneously, and then output the individual image features for feature concatenation. After this, the concatenated features pass through the fully connected layer and the sigmoid function activation, and finally, are converted into the probability of the image being fake or real. In the initialization of the training discriminator, the real image and the distress features of the real image are labeled as "real", and the generated image and the generated distress are labeled as "fake". The structural parameters of the overall network are shown in Table 1a–c.

Table 1. a: MDTMN's Generator Network Details. **b**: Overall\Partial Distress Discriminator Network Details. **c**: Features Concatenation Layer.

a				
Operation	**Kernel Size**	**Dilated Rate**	**Step**	**Output Channel**
Conv	5 × 5	1	1	64
Conv	3 × 3	1	2	128
Conv	3 × 3	1	1	128
Conv	3 × 3	1	2	256
Conv	3 × 3	1	2	256
Conv	3 × 3	1	2	256
Dilated-conv Group	3 × 3	2,6,12	1	256
Conv	3 × 3	1	2	256
Conv	3 × 3	1	2	256
Conv	3 × 3	1	2	256
Trans-conv	4 × 4	1	1/2 × 1/2	128
Conv	3 × 3	1	2	128
Trans-conv	4 × 4	1	1/2 × 1/2	64
Conv	3 × 3	1	2	64
Conv	3 × 3	1	2	3

Table 1. *Cont.*

b				
Operation	**Kernel Size**	**DilatedRate**	**Step**	**Output Channel**
Conv	5 × 5	1	2	64
Conv	5 × 5	1	2	128
Conv	3 × 3	1	2	256
Conv	3 × 3	1	2	512
FC	3 × 3	-	-	1024
c				
Operation	**Kernel Size**	**DilatedRate**	**Step**	**Output Channel**
Concat	-	-	-	2048
FC	-	-	-	1

From the details of the network structure in Table 1a, it can be seen that the encoder in the red dashed box in Figure 10 employs a step-by-step convolution to extract the features from the original image, which directly responds to the exponentially increasing number of output feature map channels. There is a special parallel step of the dilated convolution operation group. MDTMN uses a total of three different dilated rates of 2, 6, and 12, and the receptive field can reach 5 × 5 pixels, 12 × 12 pixels, and 25 × 25 pixels, respectively. The outstanding advantage of the dilated convolution is to enlarge the receptive field of the network without introducing additional parameters by filling 0-valued pixels at specified positions to the convolution kernel.

The two discriminators described in Table 1b gradually compress the image into a 1024-dimensional feature vector by a convolution operation with a step size of 2. The two vectors are output as the probability of the real image by the feature concatenation and full connection operation in the table in Table 1c.

4.2.3. Loss Function

The input content of MDTMN in this section is the fused pavement image, which differs from M-DCGAN, where the input is random noise. The difference between the two network structures is also large, especially given that the MDTMN uses two discriminators, so although they are the same generative adversarial network, the loss function is different.

Since two discriminators are being used, two loss functions are introduced simultaneously: the Mean Squared Error (MSE) loss function and the vanilla GAN loss function. MSE is defined as the mean of the 2-Norm squared of n samples x in a batch and the corresponding n outputs y. It can be used to compare the difference between two types of images in a batch, as in Equation (2).

$$MSE(x, y) = \frac{1}{n}(\|x - y\|_2)^2 \tag{2}$$

The common MSE loss is modified by converting it into a loss function for the MDTMN generator so that it focuses on the quality of distress generation in the masked region, as shown in Equation (3).

$$L(x, Mk) = \|M_k \odot (G(x, M_k) - x)\|_2^2 \tag{3}$$

where M_k is the distress mask image, $G(x, M_k)$ denotes the generated image, and x denotes the original image. By matrix dot product of the mask with the generated image and the filled image, respectively, the region filled by the generator and the original distress region

can be obtained, and by comparing the gap between these two regions, the loss of the generator can be obtained.

The loss of the discriminator is calculated by the vanilla GAN Loss formula, as in Equation (4):

$$\min_{G} \max_{D} V(D, G) = \mathrm{E}[\log D(x, M_k) + \log(1 - D(G(x, M_k)))] \quad (4)$$

where D denotes the discriminator network of two branches that discriminate the overall image and partial distress generation, respectively.

When training is performed using a two-stage training model, the first stage training generator that uses a loss function of Equation (3), the second stage using the overall synchronization of training requires the combination of two loss functions as Equation (5).

$$L_{total}(x, M_k) = \min_{G} \max_{D} E(\alpha V(D, G) + L(x, M_k)) \quad (5)$$

where α is the hyperparameter set to 0.1.

4.2.4. Training Parameters Configuration

The training data are fused images, and the image size is resized to 256 × 256 in order to improve the training speed, the optimizer is selected as Adam, where the learning rate is set to 0.001, the number of generator training epochs is set to 100, the total epochs is set to 200, and the number of batches is set to 64. A total of 1600 images are used for the training, and the remaining 200 images in the dataset are used as the test set. The results of the periodic generation of the model in training are shown in Figure 12. The outline of the potholes already appeared from the 5th epoch, and the internal structure of the potholes gradually became clear as the training epochs increased, and the generated images are very close to the real potholes after 50 epochs.

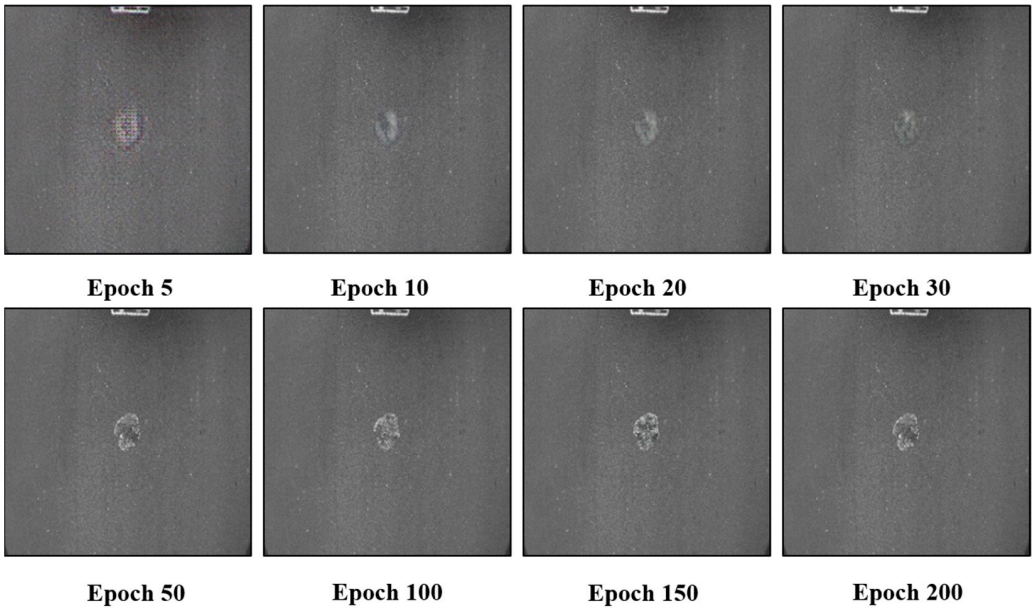


Figure 12. Generated results by different epochs models.

4.2.5. Qualitative Evaluation

For the comparison of network performance, in addition to the MDTMN model in this paper, three more deep learning algorithms, CPN, Cycle-GAN, and PIX2PIX, are selected for comparison. Two types of test data are adopted: one is the test set image with real potholes, as shown in Figure 13, and the other is the image without potholes in the pavement image, which needs to be generated by the model autonomously, as

shown in Figure 13. Figure 14 shows that among the four generation models, only CPN does not generate pavement distress. This is primarily because its training and network design do not generate semantically for the mask of the specified region. Instead, it uses the global random mask generation method. It can be seen that since there are many pavements context in the figure, the CPN focuses more attention on the generation of pavement context, and its generated images also generate the mask into pavement context. Cycle-GAN is an unsupervised generative adversarial network because it does not strictly restrict the generation semantic and the hidden variables are not sufficiently separated. Although the cyclic bidirectional generation is used for training, its generation semantic is still disturbed by other hidden variables, and even though the contours of the potholes in the generated images can be seen, the overall image quality is poor and blurred, and there is more additional noise. Only MDTMN and PIX2PIX can generate the pits correctly, among which MDTMN has the best quality of pits generation, closest to the original pothole image, the clearest contours, and the most details inside the pothole. The last row shows an image with two potholes masks. Although both MDTMN and PIX2PIX generate the double potholes correctly, the former is slightly more realistic than the latter. For the small pothole at the bottom right, the interior generated by PIX2PIX is close to off-white, but the actual interior color is gray-black, and only MDTMN is closer.

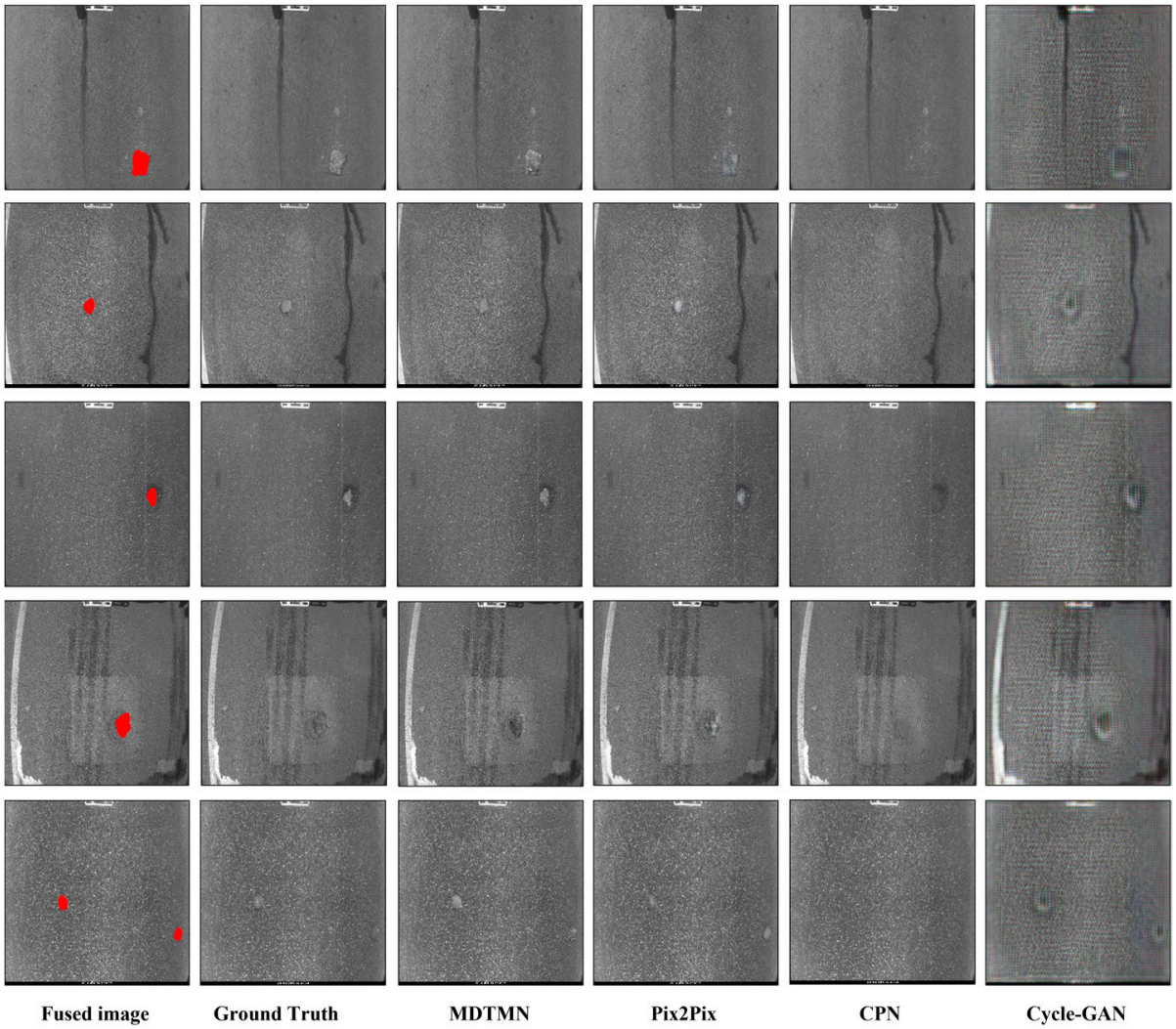


Figure 13. Pothole images generation with Ground Truth images reference. The red marks illustrate the labeled pit area.

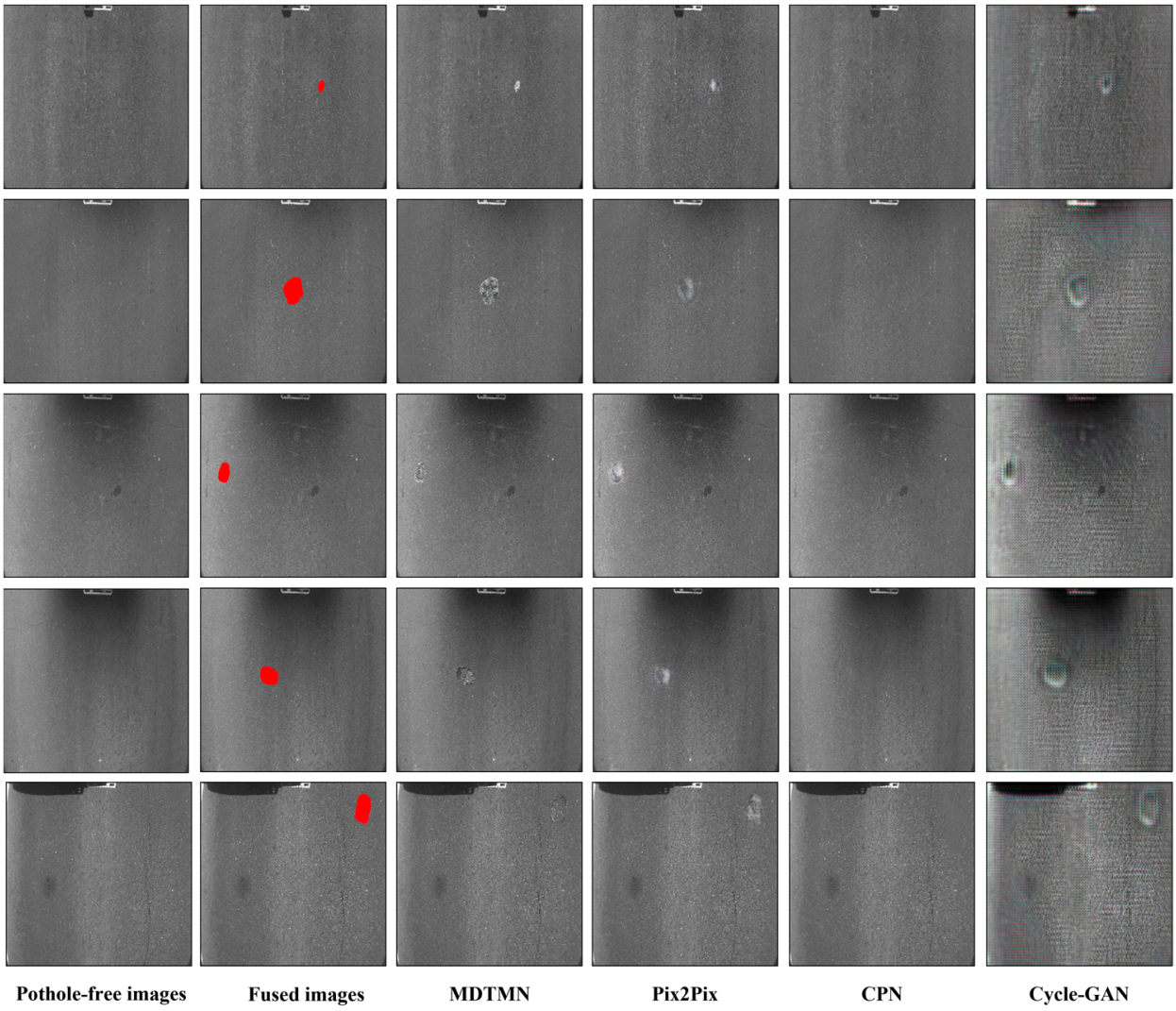


Figure 14. Pothole images generation with pothole-free images. The red marks show the generated pit mask.

Figure 14 shows the results of generating the original pavement image without pothole lesions by each model. The MDTMN still generates good results, and although there is no GT image for comparison, the details inside the generated potholes are still rich, and especially, there is no "model collapse" phenomenon. The MDTMN model can generate different details of potholes distress according to different masks. The best performance in the test of freely generated distress images is MDTMN, whereas CPN still fails to generate potholes, and Cycle-GAN's still suffers from implicit semantic interference generating images that still contain a lot of noise.

4.2.6. Quantitative Evaluation

The generated results of MDTMN are already very close to the real pavement distress images in terms of visual perception, but still need to be evaluated quantitatively. In this section, four evaluation indexes, PSNR, SSIM, FID, and LPIPS, are used to evaluate them in terms of SNR, structural similarity, and visual perception, respectively.

The LPIPS (Learned Perceptual Image Patch Similarity) of the four metrics uses depth features for the perceptual metric of two images. The calculation procedure is as follows, the reference image and the generated image $\boldsymbol{x}$ and $\boldsymbol{x_0}$ are fed into the underlying network $\boldsymbol{F}$ for feature extraction of a total of $\boldsymbol{L}$ layers, and then feature normalization is performed, where the features of the lth layer can be expressed as $\hat{y}^l, \hat{y}^l{}_0 \in \mathbb{R}^{H_l \times W_l \times C_l}$, the scaling

channel is activated using vector $w_l \in \mathbb{R}^{C_l}$ and then the L2 distance is calculated, mean in space, and summed over the lth layer, as in Equation (6).

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \| \omega_l \odot (\hat{y}^l{}_{hw} - \hat{y}^l_{0hw}) \|_2^2 \quad (6)$$

The evaluation of the generated results of these four models is shown in Table 2.

Table 2. Evaluation of the generation results of each algorithm under the reference of Ground Truth images.

Models	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓
MDTMN	28.35	0.633	0.038	35.713
PIX2PIX	28.17	0.619	0.053	38.230
CPN	28.40	0.603	0.050	38.332
Cycle-GAN	22.61	0.282	0.1813	202.703

The higher the metric, the better the performance, as indicated by ↑, and the lower the metric, the better the performance, as indicated by ↓.

The performance metrics of each model with reference images are shown in Table 2, in which MDTMN leads all models. However, in PSNR, it is only slightly ahead of PIX2PIX. Likewise, slightly behind are CPN and Cycle-GAN, which have the worst visual effect, and have the lowest scores for all evaluation metrics. It is worth noting the score of PSNR. Although MDTMN achieves a better generation effect, the highest score in this item is indeed the result of CPN algorithm, because it only focuses on the difference of pixel values and is not sensitive to the image structure and semantic distribution, so this score cannot be used to evaluate the quality of generated images alone.

To evaluate the gap between autonomously generated distress and real pavement distress, since there are no paired GT images, this paper follows the batch images from the perspective of visual perception, and the comparison results in Table 3 show that MDTMN is optimal in both visual perception proximity scores, proving that its generated images are closest to the real distress.

Table 3. The comparison results between the autonomous generation of pavement distress images by each algorithm and the real images.

Models	LPIPS	FID
MDTMN	0.176	225.118
PIX2PIX	0.179	230.928
CPN	0.175	245.520
Cycle-GAN	0.237	283.782

4.3. Pixel-Wise Pavement Distress Detection Test

To demonstrate the effectiveness of data augmentation, this paper uses the semantic segmentation network BiseNet to perform pixel-level distress detection on the original 102 pothole images (90 images for training) and 850 crack images (770 images for training) at the beginning. Then, the pothole images augmented by the conventional algorithm, the synthetic pothole images augmented by the proposed algorithm, and different combinations between them will can be inserted into the training set at different training stages. More specifically, to quantify the improvement of detection performance by the number of augmented images, the original images were augmented to 300 images by the conventional and proposed algorithms, respectively. Following each round of model training, the training set was progressively expanded by incorporating 100 pothole images generated by various augmentation techniques. This procedure was iteratively performed until all augmented images in the training set were used. The results are reported in Table 4.

Table 4. Pothole detection results under different training data, original data (O), data augmented by conventional (C), and proposed algorithms (P).

Data	Amount (Crack/Pothole)	Pixel-Acc	Recall	Kappa	IoU
O	952 (850/102)	0.8889	0.521	0.6515	0.4883
O + C(100)	1052 (850/202)	0.8816	0.7302	0.7954	0.665
O + P(100)	1052 (850/202)	0.8933	0.7433	0.8003	0.686
O + C(100) + P(100)	1152 (850/302)	0.925	0.7466	0.8234	0.704
O + C(200)	1152 (850/302)	0.897	0.7646	0.8226	0.7029
O + P(200)	1152 (850/302)	0.9412	0.8212	0.8750	0.7812
O + C(300)	1252 (850/402)	0.9031	0.8106	0.8663	0.7622
O + P(300)	1252 (850/402)	0.9501	0.8517	0.8964	0.8152
O + C(200) + P(100)	1252 (850/402)	0.9260	0.8551	0.8867	0.7997
O + C(100) + P(200)	1252 (850/402)	0.9104	0.8651	0.8902	0.8033
O + C(300) + P(300)	1552 (850/702)	0.9208	0.9013	0.9093	0.8365

From the detection performance evaluation in Table 4, we obtain the highest IoU from the model using all 600 augmented images, reaching 0.8365, which is 1.7 times higher than the performance of the model only using the original images. The IoU boost is fastest when the number of images is boosted to twice the original images, and then the IoU boost becomes slower as the amount of data gradually increases.

In addition, when the training data is the result of a single augmentation algorithm, the detection model performs better on the proposed datasets of different magnitudes. When the training data is a mixed dataset, the detection model with a larger proportion of images augmented by the proposed algorithm in the training set has a better IoU performance of the model. Accordingly, it can be concluded that the model performance is not simply positively correlated with the number of images in the dataset, and enhancing the diversity of augmented images is a more effective means to improve the performance and robustness of detection models.

5. Conclusions

In this paper, a new two-stage image augmentation approach is proposed, in which the augmentation of pavement distress mask images is conducted in the first stage, and in the second stage, the distress-free images are first fused with the mask images, and then the distress masks in their fused images are generated to generate pavement distress to complete the image augmentation of semantic content. Moreover, M-DCGAN and MDTMN are designed for different tasks in the two phases. These two generation networks can work together in combination or separately, resulting in better performance than using the same type of algorithms. In addition, since the generation approach of the pavement distress images is based on the distress mask generation, it can be naturally used as the well-labeled training data for the semantic segmentation model, which saves a lot of labor. This approach can leverage the full range of data in the dataset, including images without pavement distress. When comparing the datasets augmented by traditional algorithms for detection experiments, the augmented datasets using the proposed algorithm can lead to better performance of the detection model. This image augmentation model with associated generative model is generic and can be used to augment other similar datasets as well.

Furthermore, data augmentation is a performance-demanding task for algorithms. Augmentation algorithms need to find a balance between computational complexity and efficiency. The proposed algorithms are based on the SE module, DCGAN, and U-net. All three network structures are well known for their low computational complexity and high efficiency. Therefore, the proposed algorithms possess these advantages as well.

Author Contributions: Data curation, Z.D.; Formal analysis, H.S.; Funding acquisition, Z.D. and Z.S.; Investigation, Z.D.; Methodology, Z.X.; Resources, Z.S.; Supervision, Z.S. and C.Y.; Validation, Z.X. and Z.D.; Writing—original draft, Z.X.; Writing—review and editing, Z.D. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key projects of Shaanxi Provincial Department of Science and Technology (No. 2022JBGS3-08), by the Postdoctoral Science Foundation of China (No. 2022M710482), by the Central Universities Basic Research Special Funds (No. 300102342107, No. 300102242901), by the Natural Science Foundation of Shaanxi Province (No. 2022JQ-527), by the National Natural Science Foundation of China(No. 51978071), and by the Department of Transportation science and technology project of Zhejiang Province (No. 2023016).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data related to this manuscript can be available on reasonable request from corresponding authors.

Conflicts of Interest: The authors have no conflict of interest regarding this manuscript.

References

1. Shang, J.; Xu, J.; Zhang, A.A.; Liu, Y.; Wang, K.C.P.; Ren, D.; Zhang, H.; Dong, Z.; He, A. Automatic Pixel-level pavement sealed crack detection using Multi-fusion U-Net network. *Measurement* **2023**, *208*, 112475. [CrossRef]
2. Zhu, J.Q.; Zhong, J.T.; Ma, T.; Huang, X.M.; Zhang, W.G.; Zhou, Y. Pavement distress detection using convolutional neural networks with images captured via UAV. *Autom. Constr.* **2022**, *133*, 103991. [CrossRef]
3. Yuan, G.J.; Li, J.B.; Meng, X.L.; Li, Y.N. CurSeg: A pavement crack detector based on a deep hierarchical feature learning segmentation framework. *IET Intell. Transp. Syst.* **2022**, *16*, 782–799. [CrossRef]
4. Tang, Y.Z.; Zhang, A.A.; Luo, L.; Wang, G.L.; Yang, E.H. Pixel-level pavement crack segmentation with encoder-decoder network. *Measurement* **2021**, *184*, 109914. [CrossRef]
5. Tang, W.H.; Huang, S.; Zhao, Q.M.; Li, R.; Huangfu, L.W. An Iteratively Optimized Patch Label Inference Network for Automatic Pavement Distress Detection. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 8652–8661. [CrossRef]
6. Sun, Y.J.; Yang, Y.; Yao, G.; Wei, F.J.; Wong, M.P. Autonomous Crack and Bughole Detection for Concrete Surface Image Based on Deep Learning. *IEEE Access* **2021**, *9*, 85709–85720. [CrossRef]
7. Song, L.; Wang, X.C. Faster region convolutional neural network for automated pavement distress detection. *Road Mater. Pavement Des.* **2021**, *22*, 23–41. [CrossRef]
8. Shim, S.; Kim, J.; Lee, S.W.; Cho, G.C. Road surface damage detection based on hierarchical architecture using lightweight auto-encoder network. *Autom. Constr.* **2021**, *130*, 103833. [CrossRef]
9. Cao, M.T.; Chang, K.T.; Nguyen, N.M.; Tran, V.D.; Tran, X.L.; Hoang, N.D. Image processing-based automatic detection of asphalt pavement rutting using a novel metaheuristic optimized machine learning approach. *Soft Comput.* **2021**, *25*, 12839–12855. [CrossRef]
10. Mei, Q.P.; Gul, M. A cost effective solution for pavement crack inspection using cameras and deep neural networks. *Constr. Build. Mater.* **2020**, *256*, 119397. [CrossRef]
11. Liu, J.W.; Yang, X.; Lau, S.; Wang, X.; Luo, S.; Lee, V.C.S.; Ding, L. Automated pavement crack detection and segmentation based on two-step convolutional neural network. *Comput.-Aided Civ. Infrastruct. Eng.* **2020**, *35*, 1291–1305. [CrossRef]
12. Kalfarisi, R.; Wu, Z.Y.; Soh, K. Crack Detection and Segmentation Using Deep Learning with 3D Reality Mesh Model for Quantitative Assessment and Integrated Visualization. *J. Comput. Civ. Eng.* **2020**, *34*, 04020010. [CrossRef]
13. Hu, R.; Singh, A. Unit: Multimodal multitask learning with a unified transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 1439–1449.
14. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
15. Wan, L.; Zeiler, M.; Zhang, S.; Le Cun, Y.; Fergus, R. Regularization of neural networks using dropconnect. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 17–19 June 2013; pp. 1058–1066.
16. Moreno-Barea, F.J.; Strazzera, F.; Jerez, J.M.; Urda, D.; Franco, L. Forward Noise Adjustment Scheme for Data Augmentation. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 728–734.
17. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
18. Wang, Y.L.; Huang, G.; Song, S.J.; Pan, X.R.; Xia, Y.T.; Wu, C. Regularizing Deep Networks with Semantic Data Augmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 3733–3748. [CrossRef] [PubMed]

19. Liu, S.; Zhang, J.; Chen, Y.; Liu, Y.; Qin, Z.; Wan, T. Pixel level data augmentation for semantic image segmentation using generative adversarial networks. In Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 1902–1906.
20. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
21. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
22. Dhariwal, P.; Nichol, A. Diffusion models beat gans on image synthesis. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 8780–8794.
23. Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
24. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph. (ToG)* **2017**, *36*, 107. [CrossRef]
25. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
26. Taylor, L.; Nitschke, G. Improving deep learning with generic data augmentation. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1542–1547.
27. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. Autoaugment: Learning augmentation policies from data. *arXiv* **2018**, arXiv:1805.09501.
28. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 13001–13008.
29. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
31. Glenn, J. YOLOv5 Release v6.1. 2022. Available online: https://github.com/ultralytics/yolov5/releases/tag/v6.1 (accessed on 8 March 2023).
32. Gao, Z.; Peng, B.; Li, T.; Gou, C. Generative adversarial networks for road crack image segmentation. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
33. Jackson, P.T.; Abarghouei, A.A.; Bonner, S.; Breckon, T.P.; Obara, B. Style augmentation: Data augmentation via style randomization. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 15–20 June 2019; pp. 10–11.
34. Mazzini, D.; Napoletano, P.; Piccoli, F.; Schettini, R. A novel approach to data augmentation for pavement distress segmentation. *Comput. Ind.* **2020**, *121*, 103225. [CrossRef]
35. Pei, L.; Sun, Z.; Xiao, L.; Li, W.; Sun, J.; Zhang, H. Virtual generation of pavement crack images based on improved deep convolutional generative adversarial network. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104376. [CrossRef]
36. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096.
37. Xu, Z.; Sun, Z.; Huyan, J.; Li, W.; Wang, F. Pixel-level pavement crack detection using enhanced high-resolution semantic network. *Int. J. Pavement Eng.* **2022**, *23*, 4943–4957. [CrossRef]
38. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.