Article

Transmission Line Vibration Damper Detection Using Multi-Granularity Conditional Generative Adversarial Nets Based on UAV Inspection Images

Wenxiang Chen 1,2, Yingna Li 1,2,* and Zhengang Zhao 1,2

1 Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China; 20182204169@stu.kust.edu.cn (W.C.); 20192204211@stu.kust.edu.cn (Z.Z.)
2 Computer Technology Application Key Lab of the Yunnan Province, Kunming 650500, China
* Correspondence: liyingna@kust.edu.cn

Abstract: The vibration dampers can eliminate the galloping phenomenon of transmission lines caused by the wind. The detection of vibration dampers based on visual technology is an important issue. Current CNN-based methods struggle to meet the requirements of real-time detection. Therefore, the current vibration damper detection work has mainly been carried out manually. In view of the above situation, we propose a vibration damper detection-image generation model called DamperGAN based on multi-granularity Conditional Generative Adversarial Nets. DamperGAN first generates a low-resolution detection result image based on a coarse-grained module, then uses Monte Carlo search to mine the latent information in the low-resolution image, and finally injects this information into a fine-grained module through an attention mechanism to output high-resolution images and penalize poor intermediate information. At the same time, we propose a multi-level discriminator based on the multi-task learning mechanism to improve the discriminator’s discriminative ability and promote the generator to output better images. Finally, experiments on the self-built DamperGenSet dataset show that the images generated by our model are superior to the current mainstream baselines in both resolution and quality.

Keywords: power transmission lines; vibration dampers detection; unmanned aerial vehicle (UAV); conditional generative adversarial nets (CGAN); Monte Carlo search (MCS)

1. Introduction

The galloping phenomenon is prone to occur when high-voltage transmission lines are subjected to wind forces. Long-term vibration leads to metal fatigue damage at the suspension, which leads to serious accidents such as wire breakage and tower collapse. The use of vibration dampers on high-voltage transmission lines can reduce the vibration of the wires caused by the wind, thereby reducing the probability of accidents. The aim of a vibration damper detection task is to provide the position of the vibration damper in the image. It is the basis for other tasks such as vibration damper corrosion inspection and mobile inspection. At present, this problem has attracted the attention of researchers studying smart grids and other fields [1,2].

In recent years, UAV-related technology has become one of the fastest developing directions in the field of science and technology. UAVs have the advantages of simple operation, easy portability, and low use cost that other aircraft such as helicopters do not have [3]. The UAV array network composed of sensor networks [4] can quickly complete tasks such as object recognition [5] and crop yield estimation [6]. While drones bring convenience to human activities, they also bring some negative effects [7]. Examples include the communication security issues of UAVs [8] and the noise pollution generated during flight, which threatens the safety of birds [9]. Current transmission line inspection tasks are still mainly performed manually. Therefore, the use of UAVs for power line inspection is an
issue worth researching, and there have been a number of relevant studies [10]. This paper focuses on the detection of line vibration dampers using aerial images obtained by UAVs.

In early work, the use of image processing algorithms to improve the visual perception of vibration dampers in images was the most common method. Usually, researchers use an appropriate feature extraction operator to detect vibration dampers [11]. In addition, there are studies that combine machine learning algorithms to improve the level of automation [1], and such methods were also the main direction of early research. However, traditional methods generally have the problem of a low detection accuracy.

In recent years, with the renewed popularity of deep learning technology, multiple algorithms represented by convolutional neural networks (CNNs) have emerged, providing new energy to the task of vibration damper detection. Such methods can obtain an end-to-end model through iterative training on the dataset, and only one input is required to obtain an output with excellent detection accuracy in subsequent use. Although CNN-like algorithms have high detection accuracy, they still have the problem of a long training time, and the effect is limited by the size of the dataset. We hope to obtain a method with low resource requirements and a high detection effect.

Benefiting from the rapid development of deep learning, image generation algorithms represented by GAN [12] have become one of the research hotspots in the field of computer vision. CGAN [13] through additional auxiliary conditions and the use of common data annotations in the field of object detection can generate vibration damper detection images.

In view of the research status of the field of vibration damper detection, as shown in Figure 1, we proposed a model for line vibration damper detection image generation that is based on a conditional generative adversarial network. The main contributions of this paper are as follows:

- On the basis of an improved conditional generative adversarial network, we proposed a framework for vibration damper detection image generation named DamperGAN. The framework contains a two-stage generator and a multi-scale discriminator.
- In order to generate high-resolution vibration damper detection images, we used a coarse-to-fine generation method. At the same time, an attention mechanism and a penalty mechanism based on Monte Carlo search were introduced into the generator to provide enough semantic information for image generation to improve the edge information of the image.
- In order to improve the high-resolution images generated by the discriminator, we proposed a multi-level discriminator structure based on the parameter sharing mechanism, so that the entire discriminator pays attention to both semantic information and underlying feature information.
- Aiming to address the problem of no public dataset in the field of vibration damper detection, we established a dataset named DamperGenSet that is based on real optical images from UAV aerial photography. Through comparison with the experimental results of multiple state-of-the-art models on the DamperGenSet dataset, we prove that our proposed DamperGAN has certain advantages.

![Figure 1](image-url)

Figure 1. Test examples of our model on DamperGenSet.

The remainder of this article is organized as follows. Section 2 briefly introduces related work on vibration damper detection and image generation. Section 3 introduces the
2. Related Work

In this section, we focus on related work on vibration damper detection using image technology, including research using traditional techniques and deep learning techniques.

2.1. Damper Detection

In recent years, there has been a certain amount of research on vibration dampers for transmission lines. In the early days, the research on vibration dampers was mainly based on image processing algorithms and edge detection models. In recent years, the convolutional neural network represented by AlexNet [14] has developed rapidly, providing new solutions for the field of object detection, and there have been increasingly more studies on CNN-based vibration damper detection.

The simplest method in the traditional algorithm is to use the statistical properties of basic colors in an image, such as the use of histogram technology for vibration damper corrosion detection [15]. This is also a detection method that improves the visibility of the vibration damper in an image via color space transformation [16]. Huang et al. [17] used grayscale processing, edge detection, threshold segmentation, morphological processing, and other technologies to calculate the rusted area ratio of the vibration damper to determine the degree of corrosion of the vibration damper and carried out displacement detection. Pan et al. [11] used the edge extraction operator to estimate the damage degree of the vibration damper. Extracting the edge of the vibration damper is also an effective detection method. For example, the Canny operator and Hough transform were used to detect the vibration damper, and the displacement distance of the vibration damper was calculated on the basis of the edge information [18]. Miao et al. [19] used the wavelet transform on the vibration damper detection problem. In a study by Wu et al. [2], on the basis of the helicopter aerial image, the authors used the Snake model to extract the edge of the vibration damper. Edge detection operators combined with machine learning algorithms were the most robust method before CNNs. Jin et al. [20,21] performed vibration damper detection and classification tasks using Harr features and cascaded Adaboost classifiers based on UAV aerial images.

At present, CNN-based research has become the main work conducted in the field of vibration damper detection of transmission lines, for example, YOLOv4 [1,22], based on the one-stage class method; Faster R-CNN [16,23], based on the two-stage class method; and research based on Cascade R-CNN [24]. However, the amount of work on vibration dampers is currently relatively small. Additionally, there are CNN-based studies on other power line components and foreign objects. For example, Faster R-CNN is used to detect the shape change of insulator strings [25], insulators [26], and icing on power lines [27]. Mask R-CNN is used to detect foreign objects such as bird nests and balloons on transmission lines [28], insulator defect detection of high-speed rail power transmission lines [29], and infrared imaging equipment used to detect the humidity of insulators [30].

Usually, these studies are merely simple applications of power component datasets; most of the studies lack targeted modification for specific environments and scenarios, and the solutions provided are mostly skill stacking. However, these studies usually only use CNNs to perform model iteration on the dataset, without targeting the particularity of the environment in which the object is located. In addition, the performance of the model is very dependent on the size of the training set.

2.2. Image Generation

With the rise of deep learning, the field of image generation has also been given new research directions. Variational autoencoder (VAE) [31] is the earliest model, which is based on probability maps in the field of image generation. Some researchers [32] added
an attention mechanism to VAE and proposed a high-quality image generation model based on DRAW. In [33], researchers proposed a mechanism for generating images using visual features named Attribute2Image, which is able to synthesize images with a separate foreground and background.

In recent years, the advent of adversarial generative networks (GAN) [12] has provided new vitality to image generation. GAN uses the mechanism of generator and discriminator adversarial learning and is trained to generate and input images that are very similar. The conditional generative adversarial network (CGAN) came into being in order to ensure the safety of staff during image generation, and a large amount of research [34–36] based on different conditions was proposed. There have also been many studies on image generation that are based on different conditions, such as text-based [37,38], label-based [39], and image-based [40–42] conditions.

However, the generated image output by the above research work is generally distorted and the texture is not clear. Most models only repeat the content in the training set, and it is difficult to mine the semantic space. These models were trained on the Damper-GenSet dataset, as the complex background information in DamperGenSet results in a mediocre quality of the generated images.

2.3. Research Summary

From the above research work, we can see that there is still room for improvement in vibration damper detection and image generation algorithms of overhead transmission lines. We summarize the following characteristics of these studies:

- Detection using traditional image processing algorithms is limited by the quality of the input image and the rationality of operator selection. If the background information in the image is too complex, the vibration damper will not be obvious enough, and importantly, the plated feature information will be weakened by the background. This makes it difficult for the feature operator to fully output the completed vibration damper information. The advantage of the traditional method is that the calculation speed is fast and the resource occupancy rate is low. In simple scenarios with high real-time requirements, it is still the most effective detection method.

- CNN-based methods are currently the most accurate solutions in the field of vibration damper detection. We only require a sufficient amount of training data to obtain an excellent end-to-end detection model. However, there is not yet a fully public vibration damper dataset. Nevertheless, such methods require higher computing power for operating equipment. We want to provide a solution that runs in real time on edge devices such as drones.

- The image generation algorithm led by GAN provides us with new solutions. CGAN uses the idea of adversarial learning to output high-quality images with a simple network structure. At the same time, auxiliary conditions can help us define the semantic information in the image. However, the current research work still has the problem of the image not being delicate enough, and there is no research work on the CGAN-based class of vibration dampers for transmission lines.

Therefore, combining the features summarized in the above research, we hope to obtain an end-to-end deep neural network model, one that can run in real time on edge devices such as drones. However, due to the difficulty of reproducing the actual scene of a transmission line in the data, the training data that the model relies on cannot be too high. Therefore, in this follow-up research work, we propose a model for generating vibration damper detection results on the basis of conditional generative adversarial networks.

3. Basic Knowledge of GAN

The generative adversarial networks (GANs) [12] are based on max–min game theory and consist of two sub-networks: a generator and a discriminator. The goal of the generator is to obtain a generated image that is very close to the original image with a d-dimensional noise. The discriminator scores images to decide whether the input is from a real image or a
generated image. In a way, the discriminator exists for the generator to output high-quality images. On the other hand, the fidelity of the images output by the generator is constantly improving, and it is also constantly challenging the discriminative ability. The objective function of the entire GAN is as follows:

$$\min_{G} \max_{D} V_{GAN}(G, D) = E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_{z}(z)}[\log(1 - D(G(z)))]$$

where \( x \) represents the real image obtained from the dataset and \( b \) represents a d-dimensional noise obtained using a normal distribution \( p_{z} \).

On the basis of GAN, CGAN [13] introduces additional auxiliary variables to control the generation results of the model output. In CGAN, the generator uses auxiliary conditions to output generated images, and the discriminator uses auxiliary conditions as the basis to determine whether the input image is true or false. The objective function of CGAN is as follows:

$$\min_{G} \max_{D} V_{CGAN}(G, D) = E_{s,x \sim p_{s}(s,x)}[\log D(s,x)] + E_{s,x^{'},z \sim p_{s}(s,x^{'})}[\log(1 - D(s,x^{'},z))]$$

where \( s \) represents the auxiliary condition and \( x^{'},z = G(s,z) \) represents the output generated image.

In addition to the generator and discriminator losses, previous work [40,43] has shown that minimizing the distance between real and fake images can help the generator output higher-quality images.

Compared with the L2 distance, the L1 distance can more directly describe the difference between a real image and a fake image, and can also help generate images to reduce distortion and blurring problems.

Therefore, we also introduced the L1 distance in the model. The formula for calculating the L1 distance is as follows:

$$\min_{G} V_{L1}(G) = E_{s,x^{'},z \sim p_{s}(s,x^{'})}[\| x - x^{'},z \|_{1}]$$

Therefore, the objective function of our model DamperGAN is the sum of (2) and (3).

4. DamperGAN

4.1. Overall Framework

DamperGAN consists of a generator \( G(I_{o}; \theta_{G}) \) and a discriminator \( D((I_{o}, I_{s}); \theta_{D}) \). We hope to use the generator \( G \) in DamperGAN to obtain a vibration damper detection image \( I_{s} \) on the basis of the original image \( I_{o} \) obtained by the drone. The overall structure of DamperGAN is shown in Figure 2, where \( I_{o} \) represents the real dataset distribution, \( G \) represents the generator, \( D \) represents the discriminator, \( I_{o} \) represents the original image, \( I_{s} \) represents the generated image output by the generator, \( S \) represents the real image output by the discriminator degree score, and \( V_{G}^{R} \) represents the calculated penalty value fed back to the generator.

![Figure 2. The framework of DamperGAN.](image)

The training of DamperGAN is divided into the learning of the generator and the learning of the discriminator, and the training of the two is a process of confrontation. The training goal of the generator is to output a high-quality fake image that is good
enough to match the real image, while the goal of the discriminator is to discriminate between real and fake input images.

4.2. Multi-Granularity Generator

The traditional generator in DamperGAN is decomposed into two sub-generators: $G_1$ and $G_2$, where $G_1$ is defined as a global generator and $G_2$ is defined as a local generator. The overall structure of the generator is shown in Figure 3. The local generator can effectively improve the resolution of the generated image. For example, the image with a resolution of $512 \times 512$ is input into the global generator, and the output resolution of the local generator is $1024 \times 1024$.

![Figure 3. The framework of the generator containing three parts: the global module, the Monte Carlo search, and the local module. First, we used the global module to obtain the LR image; then, we used the Monte Carlo for detail mining; and finally, we used the local module to obtain the HR image.](image)

The global generator consists of three parts: front-end convolution layer $G_1^{(F)}$, residual block $G_1^{(R)}$, and transposed convolution back-end $G_1^{(B)}$. The original real images with a resolution of $512 \times 512$ are sequentially passed through these three parts, and the final output image with a resolution of $512 \times 512$ contains the dampers.

The local generator also contains three parts: front-end convolutional layer $G_2^{(F)}$, residual block $G_2^{(R)}$, and transposed convolution back-end $G_2^{(B)}$. The input of the local generator is the original image with a resolution of $1024 \times 1024$. Unlike the global generator, the input of the residual block of the local generator is the sum of the feature maps of the $G_1$ and the $G_2^{(F)}$, and thus, the extracted feature information from the global generator $G_1$ is passed to the local generator $G_2$.

In the training process, the original image of $1024 \times 1024$ was initially a downsampling operation to obtain an image with a resolution of $512 \times 512$ to pre-train the residual network block of $G_1$ and then to train the local generator $G_2$, mainly because the resolution of two input images is different.

Additionally, in order to improve the semantic guidance for the generated images, we proposed an attention mechanism and a penalty mechanism based on Monte Carlo search.
4.2.1. Penalty Mechanism

To further improve the details of the generated images and the positional accuracy of the boxes, we proposed a penalty mechanism based on a Monte Carlo search (MCS). Monte Carlo searches can mine latent spatial information from the data, and we can obtain the semantic details with clear texture on the basis of the results of the Monte Carlo search and the score of the discriminator.

We performed a Monte Carlo search on the basis of the results generated by the global generator $G_1$, and the search process can be expressed by the following equations:

$$\{I^1_S, \ldots, I^N_S\} = MC^{G_S}(I'_S, N)$$

$$I^i_S = G_\beta(I'_S, z)$$

where $I^i_S (i = 1, \ldots, N)$ represents the $N$ images obtained on the basis of the results generated by the global generator $G_1$, and $MC^{G_S}$ represents the state we simulated using MCS. $G_\beta$ represents the generation model based on MCS virtualization technology that shares parameters with the local generator $G_2$, $z \sim N(0, 1)$ represents the noise variable introduced during input that guarantees diversity in the search results.

In order to reward the search process with better results, we fed the $N$ Monte Carlo search results and the generated images of the global generator $G_1$ output to the discriminator $D$, and we were able to obtain a penalty value $V_D^{G}$ according to the score output by the discriminator. The whole calculation process is shown as Equation (6).

$$V_D^{G} = \frac{1}{N} \sum_{i=1}^{N} \left( 1 - D(I^i_S; \theta_d) \right) + \| I'_S - I_S \|_1$$

where $D(I^i_S; \theta_d)$ represents the score of the image output from $D$, and the higher the score, the more likely it is to be a true image.

4.2.2. Attention Mechanism

After obtaining multiple intermediate results through a Monte Carlo search, we hoped to use these results as the basis for the next high-resolution image generation task. We therefore introduced an attention mechanism to feed the positive effects of the Monte Carlo search into the local generator. We constructed a feature set with $N$ intermediate results obtained by sampling and used the convolution operation to obtain different weight matrices for different results. Finally, the result obtained was fed into a local generator to obtain a higher-resolution target image.

We performed a convolution operation on the basis of $N$ intermediate images to obtain the attention weight matrix. The results were calculated as shown in Equation (7).

$$I^i_A = \text{Softmax}(I^i_S W^i_A + b^i_A), \text{ for } i = 1, \ldots, N$$

where $I^i_S$ represents the input image, $\{W^i_A, b^i_A\}_{i=1}^{N}$ represents the parameter of the convolution operation, and Softmax(·) represents the SoftMax function used for normalization. We multiplied the resulting attention weight matrix with the corresponding input image to obtain the final output:

$$I'_S = (I^1_A \otimes I^1_S) \oplus \ldots \oplus (I^N_A \otimes I^N_S)$$

where $I'_S$ represents the final output result of the attention mechanism, $I^i_S$ represents the input images; and the symbols $\otimes$ and $\oplus$ represent the multiplication and addition elements of the matrix, respectively.
4.2.3. Objective Function

The loss function of the entire generator network is written as Equation (9).

\[ J_G(\theta_G) = E_{l_o \sim P(I_o), l_t \sim P(I_t)}[\log(1 - D(l_o, G(l_o, I_t)))] + V_G^D \]  

(9)

where \( D(l_o, G(l_o, I_t)) \) represents the score of the discriminator \( D \) for the image output by the generator \( G \), which reflects the discriminator’s ability to discriminate against false images; \( \theta_G \) represents the parameters constituting the generator; and \( V_G^D \) represents the penalty value for the generated result output by Equation (6).

4.3. Multi-Level Discriminator

The structure of the discriminator is crucial for generating high-resolution images. The discriminator requires a deeper network or a larger-scale convolution kernel to extract feature information in order to distinguish between high-resolution real samples and generated samples, but this inevitably leads to a surge in network capacity and even overfitting. In addition, an overly complex discriminator will consume a large amount of memory resources, which are very scarce when generating high-resolution images.

4.3.1. Multitasking Mechanism

In order to improve the model’s ability to output high-resolution images, we proposed a multi-scale discriminator structure based on a multi-task learning mechanism. As shown in Figure 4, the network uses three discriminators—\( D_1 \), \( D_2 \), and \( D_3 \)—with the same structure to deal with input images of different resolutions.

![Figure 4](image)

**Figure 4.** The discriminator consists of three parts with the same structure: \( D_1 \), \( D_2 \), and \( D_3 \). First, we used the shared convolutional layer to obtain the feature map of input; then, the feature map was down-sampled for 2 times and 4 times and output to \( D_1 \), \( D_2 \), and \( D_3 \); finally, we obtained the discriminator scores.

We utilized a multi-task learning strategy to train the discriminator on the basis of a parameter sharing mechanism [44]. First, we utilized the shared convolutional layers to obtain the feature maps of the samples. The feature map was then down-sampled with factors of 2 and 4 to obtain feature maps of the other two scales. Finally, three discriminators were used to process feature maps of three different scales.

Although the three discriminators use the same network structure, different inputs can bring about different discrimination capabilities to the entire discriminator network. The discriminator for small input can process more high-level semantic information, while the discriminator for large input can process more low-level feature information. Therefore, the structure of the multi-layer discriminator is very beneficial for improving the
discrimination level of the entire discriminator network. When we deal with different identification requirements, we only need to increase or decrease the number of discriminators on the basis of the original model and do not need to retrain completely from scratch.

4.3.2. Objective Function

The formula for the entire discriminator network is as shown in Equation (10).

$$J_D(\theta_d) = - \sum_{k=1,2,3} \mathbb{E}_{I_o \sim P_r(I_o), I_g \sim P_g(I_g)} \log D_k((I_o, I_g); \theta_d)$$  \hspace{1cm} (10)

where $P_r$ represents the real dataset, $P_g$ represents the sample set of generated images, $I_o$ represents the original image, $I_g$ represents the generated image, $\theta_d$ represents the parameters of the discriminator network, and $D_k$ represents one of the discriminators.

On the basis of the entire multi-granularity generator and multi-scale discriminator proposed in this paper, we conducted adversarial learning of the two. The entire learning process is shown in Algorithm 1.

Algorithm 1. The training process of DamperGAN.

Input: Real damper image dataset $I_o = \{I_{o1}, \ldots, I_{oN}\}$; Generator $G$; Discriminator $\{D_i\}_{i=1}^{l=k}$; g-steps, the training step of the generator; d-steps, the training step of the discriminators.

Output: $G$, generator after training.

1: Initialize generator $G$ and discriminator $\{D_i\}_{i=1}^{l=k}$ with random weights;
2: repeat
3: for g-steps, perform
4: $G$ generate fake images;
5: Calculate the penalty value $V_G^D$ via Equation (6);
6: Minimize Equation (9) to update the parameters of the generator $G$;
7: end for
8: for d-steps, perform
9: Use $G$ to generate fake images $I_g = \{I_{g1}, \ldots, I_{gN}\}$;
10: Use real images $I_o = \{I_{o1}, \ldots, I_{oN}\}$ and fake images $I_g = \{I_{g1}, \ldots, I_{gN}\}$ to update the discriminator parameters by minimizing Equation (10);
11: end for
12: until DamperGAN completes convergence
13: return

4.4. Network Structure

As we generated images on the basis of annotated images, there were a large number of low-level features between them. To improve the restoration of these features, we used a simple U-net [45] as the main infrastructure of the generator and discriminator.

The structure of the entire generator network is shown in Table 1. CONV stands for convolution operation, N-m indicates that the number of convolution kernels in each convolution layer is m; K-mxm indicates that the size of the kernel is mxm; S-m indicates that the stride of the kernel is m; P-m indicates that the size of the boundary expansion of the input image during the convolution operation is m, and IN, ReLU indicates that the current loss function is InstanceNorm-ReLU [46].

The structure of the entire discriminator network is shown in Table 2. Unlike the generator, InstanceNorm was not used for normalization in the first layer of the network, Convolution-InstanceNorm-LeakyReLU [47] was used as the loss function, and the slope of Leaky ReLU was 0.2. The convolutional layer used in the last layer produced a one-dimensional output, and the discriminator of the three-layer network architecture was the same.
| Network | Layer Information | Input | Output |
|---------|------------------|-------|--------|
| Down-Sample | CONV-(N64, K7 × 7, S1, P3), IN, ReLU | (512, 512, 3) | (512, 512, 64) |
|           | CONV-(N128, K3 × 3, S2, P1), IN, ReLU | (512, 512, 64) | (256, 256, 128) |
|           | CONV-(N256, K3 × 3, S2, P1), IN, ReLU | (256, 256, 128) | (128, 128, 256) |
|           | CONV-(N512, K3 × 3, S2, P1), IN, ReLU | (128, 128, 256) | (64, 64, 512) |
|           | CONV-(N1024, K3 × 3, S2, P1), IN, ReLU | (64, 64, 512) | (32, 32, 1024) |
| Residual Block | CONV-(N1024, K3 × 3, S1, P1), IN, ReLU | (32, 32, 1024) | (32, 32, 1024) |
|           | CONV-(N1024, K3 × 3, S1, P1), IN, ReLU | (32, 32, 1024) | (32, 32, 1024) |
|           | CONV-(N1024, K3 × 3, S1, P1), IN, ReLU | (32, 32, 1024) | (32, 32, 1024) |
|           | CONV-(N1024, K3 × 3, S1, P1), IN, ReLU | (32, 32, 1024) | (32, 32, 1024) |
|           | CONV-(N1024, K3 × 3, S1, P1), IN, ReLU | (32, 32, 1024) | (32, 32, 1024) |
|           | CONV-(N1024, K3 × 3, S1, P1), IN, ReLU | (32, 32, 1024) | (32, 32, 1024) |
|           | CONV-(N1024, K3 × 3, S1, P1), IN, ReLU | (32, 32, 1024) | (32, 32, 1024) |
| Up-Sample | CONV-(N512, K3 × 3, S0.5, P1), IN, ReLU | (32, 32, 1024) | (64, 64, 512) |
|           | CONV-(N256, K3 × 3, S0.5, P1), IN, ReLU | (64, 64, 512) | (128, 128, 256) |
|           | CONV-(N128, K3 × 3, S0.5, P1), IN, ReLU | (128, 128, 256) | (256, 256, 128) |
|           | CONV-(N64, K3 × 3, S0.5, P1), IN, ReLU | (256, 256, 128) | (512, 512, 64) |
|           | CONV-(N3, K7 × 7, S1, P3), IN, ReLU | (512, 512, 64) | (512, 512, 3) |

| Network | Layer Information | Input | Output |
|---------|------------------|-------|--------|
| Input Layer | CONV-(N64, K4 × 4, S2, P2), Leaky ReLU | (512, 512, 3) | (256, 256, 64) |
|           | CONV-(N128, K4 × 4, S2, P2), IN, ReLU | (256, 256, 64) | (128, 128, 128) |
|           | CONV-(N256, K4 × 4, S2, P2), IN, ReLU | (128, 128, 128) | (64, 64, 256) |
|           | CONV-(N512, K4 × 4, S2, P2), IN, ReLU | (64, 64, 256) | (32, 32, 512) |

5. Experiments and Analysis

5.1. Experiment Description

5.1.1. Dataset

A dataset of vibration dampers for overhead transmission lines is required for the proposed theoretical validation and experimental analysis. At present, although there is a lot of research work on vibration dampers, there is no completely public vibration damper detection dataset. Moreover, most of the vibration damper data in the article were obtained by geometric transformation methods such as flipping, cutting, and scaling. An insufficient number of vibration dampers would make it difficult to verify the correctness of the proposed theory. Therefore, we made a dataset for vibration damper detection based on the real UAV cruise video of overhead transmission lines and named it DamperGenSet. In the process of making the DamperGenSet dataset, we used CAD2018 as a data labeling tool to label the positions of all existing line vibration dampers in the original image. The callout box used magenta and was as close as possible to the smallest bounding rectangle of the target area.

DamperGenSet contained a total of 3000 images, each of which contained vibration dampers, and the types of vibration dampers were not unique, such as hippocampus an-tislip vibration dampers and hook wire vibration dampers. We randomly divided all 3000 images into a training set and a test set. The training set contained 2500 images and the test set contained 500 images. The ratio of training set to test set was 5:1. In addition, as the dataset was obtained by shooting with UAVs, the presentation angle of the vibration damper in the image was not unique, which also put forward higher requirements for the robustness of the model.
5.1.2. Experiment Configuration

In terms of hyperparameter settings in the experiment, we trained DamperGAN for a total of 200 epochs. The learning rate of the first 100 epochs remained unchanged, and the learning rate of the last 100 epochs gradually decreased to 0. In terms of experimental software settings, all our programs were written in Python language and integrated based on the PyTorch 1.4 platform. In the system environment of the experimental platform, we used Ubuntu18.04 as the operating system. In terms of the hardware environment of the experimental platform, we used an NVIDIA RTX 2080 GPU as the main equipment for training calculation, matched with an AMD R5-3600X CPU and 32 GB RAM.

5.2. The Baselines

In the following experiments, we use state-of-the-art methods of image generation as comparison methods.

Pix2Pix [40]: pix2pix is one of the most representative techniques in generative algorithms. The method uses an adversarial mechanism to learn the mapping between input and output, and has achieved excellent results in tasks such as image translation.

CRN [48]: This method is different from the adversarial training method in GAN, which just uses traditional convolutional layers to construct the entire network. It is worth noting that this method adds diversity loss between input and output to the loss function.

X-Fork [49]: X-fork is an image generation model that generates target images based on semantic segmentation maps and original images. The key to its effect is the quality of the semantic segmentation map.

X-Seq [49]: This method uses the idea of image generation to complete the task of semantic segmentation. The first generator outputs the image of the target viewpoint, and the second generator outputs the semantic segmentation map based on it.

SelectionGAN [50]: This method uses an attention mechanism to select the intermediate generated results of the model to improve the quality of the final generated pictures.

5.3. Qualitative Evaluation

To visually compare the difference between the detection effect of DamperGAN and other baselines, we conduct qualitative analysis and comparison experiments based on the DamperGenSet dataset. As can be seen from Figure 5, under the same test image, the detection effect of CRN is not stable enough. We believe that relying on convolutional layers for feature transfer alone is not enough to generate sufficiently realistic images. The performance of SelectionGAN is excellent. As a state-of-the-art framework based on CGAN, the attention mechanism provides a lot of reference information for the final result. The performance of pix2pix also has room for improvement. We argue that simple adversarial learning mechanisms still require other tricks to empower them. The entries for x-fork and x-seq are generally better. They all benefit from a multi-stage generation strategy, with different focuses in different generation stages, and task classification can make the whole network perform better. The image texture generated by the DamperGAN proposed in this paper is more refined, the position of the frame is more accurate, and the number of ghosts is lower. The overall performance of DamperGAN is better than other baselines.
5.4. Quantitative Evaluation

5.4.1. Inception Score (IS) and Fréchet Inception Distance (FID)

The Inception Score (IS) is a common standard used to evaluate the quality of the output of the generative model, and the higher the value, the higher the clarity of the image. Its calculation formula is shown in Equation (11).

\[
\text{IS}(G) = \exp(\mathbb{E}_{x \sim P_x} D_{KL}(p(y|x) \parallel p(y)))
\]

(11)

where \(x\) represents the generated image output by \(G\), \(y\) represents the category of the generated image, \(D_{KL}\) represents relative entropy, and \(P_x\) represents the sample space of the generated image.

As the object we detected, the anti-vibration hammer, does not belong to the ImageNet dataset [51], we want to improve the test speed. Therefore, we use AlexNet instead of the Inception framework to score DamperGAN.

The Fréchet Inception Distance (FID) is a metric used to measure the gap between the generated image and the real image. The larger the value of FID, the less realistic the image is. Its calculation formula is shown in Equation (12).

\[
\text{FID}(I_t, I_g) = \|\mu_t - \mu_g\|^2 + T_r \left(\sum_{i_t} + \sum_{i_g} - 2\left(\sum_{i_t} \sum_{i_g}\right)\right)^{\frac{1}{2}}
\]

(12)

where \(\mu_t\) represents the mean value of the feature map extracted from the original target image, \(\mu_g\) represents the mean value of the feature map extracted from the generated image, \(T_r\) represents the sum of the elements on the diagonal of the matrix, \(\Sigma_{t_i}\) represents the covariance matrix of the feature map of the original image, and \(\Sigma_{g_i}\) represents the covariance matrix of the image’s feature maps. We also use AlexNet instead of inception as the tool for extracting feature maps. We compute \(4096 \times 4096\) covariance matrices \(\Sigma_{t_i}\) and \(\Sigma_{g_i}\) using the \(1 \times 4096\) vector output from the last pooling layer.

As shown in Table 3, DamperGAN outperforms other baselines in IS and FID, and has the most improvement over classic pix2pix and CRN. This shows that the improvements in
our proposed model provide additional reference information for image generation, and the obtained images are closest to the original in quality and similarity.

Table 3. IS and FID of the different models.

| Model                | InsuGenSet IS | FID  |
|----------------------|--------------|------|
| Pix2Pix              | 3.25         | 57.45|
| CRN                  | 3.04         | 57.68|
| X-Fork               | 3.37         | 56.90|
| X-Seq                | 3.61         | 56.42|
| SelectionGAN         | 3.75         | 56.08|
| DamperGAN            | 3.83         | 55.31|

5.4.2. Structural Similarity (SSIM), Peak Signal-to-Noise Ratio (PSNR), and Sharpness Difference (SD)

After evaluating the generative ability of the model using IS and FID, based on the research work in [52], we adopt three pixel-level metrics: SSIM, PSNR, and SD, to further evaluate the generated images.

Structural Similarity (SSIM) is an index that evaluates the similarity between images based on brightness and contrast. The value of SSIM ranges from $-1$ to $1$, with larger values representing better performance. The formula for SSIM is shown in Equation (13).

$$SSIM(I_g, I_t) = \frac{(2\mu_I \mu_g + c_1)(2\sigma_I \sigma_g + c_2)}{(\mu_I^2 + \mu_g^2 + c_1)(\sigma_I^2 + \sigma_g^2 + c_2)}$$

(13)

where $I_g$ represents the output image, $I_t$ represents the standard image, $\mu_I$ represents the average value of the output image, $\mu_g$ represents the average value of the labeled image, $\sigma_I$ represents the standard deviation of the generated image, $\sigma_g$ represents the standard deviation of the labeled image, and $c_1, c_2$ are the adjustment factors.

The peak signal-to-noise ratio (PSNR) uses the peak signal of the noise ratio between the real image and the generated image for similarity comparison. Higher numbers represent better similarity. The calculation formula of PSNR is shown in Equations (14) and (15), and the specific meaning of the parameters is the same as above.

$$PSNR(I_g, I_t) = 10 \log_{10} \left( \frac{\max_I^2}{\text{mse}} \right)$$

(14)

$$\text{mse}(I_g, I_t) = \frac{1}{n} \sum_{i=1}^{n} (I_t[i] - I_g[i])^2, \max_I = 255$$

(15)

The sharpness difference (SD) is a metric used to describe the degree of sharpness loss during image generation, and we refer to the work in [52] to describe this criterion by computing the gradient change between the generated image and the original image. The calculation formula of SD is shown in Equations (16) and (17), with the specific meaning of the parameters being the same as above.

$$\text{SharpDiff.}(I_g, I_t) = 10 \log_{10} \left( \frac{\max_I^2}{\text{grads}} \right)$$

(16)

$$\text{grads} = \frac{1}{N} \sum_i \sum_j \left( |(\nabla I_0 + \nabla I_0) - (\nabla I_g + \nabla I_g)| \right)$$

(17)

where $\nabla I = |I_{i,j} - I_{i,j-1}|, \nabla I = |I_{i,j} - I_{i,j-1}|$. 


As can be seen from Table 4, under the same test picture, thanks to the attention mechanism, the performance of SelectionGAN is still stable, and its performance under various standards is in the forefront; however, its good score comes at the cost of great computation time. The X-Fork and X-Seq perform similarly, both of them having a certain degree of lead in terms of indicators. In addition, the calculation speed is faster than SelectionGAN. The Pix2pix and CRN have the lowest score. However, the advantage of them is that the calculation speed is much faster than other baselines, which is a significant advantage for scenarios with extremely high real-time requirements. DamperGAN outperforms other baselines in all three evaluation indicators, and the performance on FPS is also similar to SelectionGAN. We believe that two-stage generation, Monte Carlo search, etc. allow it to have better performance and use less additional computational cost.

Table 4. SSIM, PSNR, SD, and FPS of the different models.

| Model          | InsuGenSet | FPS |
|----------------|------------|-----|
|                | SSIM       | PSNR | SD  |
| Pix2Pix        | 0.29       | 15.91| 17.41| 160 |
| CRN            | 0.27       | 15.53| 17.12| 187 |
| X-Fork         | 0.38       | 16.37| 18.21| 85  |
| X-Seq          | 0.45       | 17.34| 18.58| 72  |
| SelectionGAN   | 0.63       | 26.83| 20.61| 66  |
| DamperGAN      | 0.70       | 28.14| 22.13| 63  |

5.5. Sensitivity Analysis

In this section, we perform multiple sets of sensitivity analysis on each component of DamperGAN, which includes the choice of backbone, edge extraction, attention mechanism, number of training iterations, and minimum amount of training data.

5.5.1. Two-Stage Generation

We conduct a sensitivity analysis on the generator used by DamperGAN while retaining other improvements. As shown in Table 5, the performance of using a two-stage generator is most balanced, the performance of a single-stage generator is average, and the time-consumption of a three-stage generator is too high.

Table 5. Comparison of the effectiveness of the generator networks.

|                  | IS   | FID  | FPS |
|------------------|------|------|-----|
| Single generator | 3.28 | 56.84| 82  |
| Two-stage generator | 3.83 | 55.31| 63  |
| Three-stage generator | 4.25 | 54.96| 37  |

5.5.2. Monte Carlo Search

The Monte Carlo search is the basis of the attention mechanism and the penalty mechanism. The introduction of Monte Carlo search allows us to further mine semantic information on the basis of low-resolution generated images and improve the basis for high-resolution generation. Therefore, the number of Monte Carlo searches that are introduced is critical to the performance impact. We conduct a comparative analysis on the number of Monte Carlo searches introduced. As shown in Table 6, when the number of Monte Carlo searches introduced is 5, the model can achieve a balance between the quality of the output results and the calculation speed.
Table 6. Introducing the Monte Carlo search time comparison.

| MCS          | SSIM | PSNR | SD   | FPS |
|--------------|------|------|------|-----|
| Not introduced | 0.57 | 26.28 | 19.30 | 75  |
| N = 1        | 0.63 | 26.84 | 20.46 | 70  |
| N = 3        | 0.68 | 27.60 | 21.37 | 67  |
| N = 5        | 0.70 | 28.14 | 22.13 | 63  |
| N = 7        | 0.72 | 28.47 | 22.54 | 58  |
| N = 9        | 0.73 | 28.62 | 23.02 | 51  |

5.5.3. Multi-Level Discriminator

In order to cope with the excellent performance of two-stage generation, we use the multi-task learning mechanism to propose a multi-level discriminator structure to improve the discriminator’s discriminative ability, which will further improve the effect of the generator’s output image. Therefore, the number of layers of the multi-layer discriminator affects the performance of the entire network. As shown in Table 7, when the number of comparators with the same structure is 3, a good result can be achieved between model performance and calculation speed.

Table 7. Comparison of the effectiveness of the discriminant networks.

| Discriminator Type          | SSIM | PSNR | SD   | FPS |
|-----------------------------|------|------|------|-----|
| Single discriminator        | 0.60 | 26.48| 19.84| 71  |
| Two-level discriminator     | 0.65 | 27.26| 20.62| 67  |
| Three-level discriminator   | 0.70 | 28.14| 22.13| 63  |
| Four-level discriminator    | 0.72 | 28.53| 22.79| 58  |

5.5.4. Number of Epochs

The number of epochs for experimental training will affect the performance of the model. As the number of training epochs is not enough, the model will be under-fitted, and the model has not yet fully learned to identify all the objects to be detected. Excessive training epochs will reduce the robustness of the model, the parameters are limited by the existing training data, and the realization of unfamiliar data in the test set will be reduced. Therefore, we conduct an evaluation test of the number of training times for the performance of the model, and the test results are shown in Table 8. When the training epoch is 200, the model performs the most balanced.

Table 8. The effect of different epoch numbers on the experimental results.

| Number of Epochs | SSIM | PSNR | SD   | FPS |
|------------------|------|------|------|-----|
| 50               | 0.32 | 16.72| 17.84| 65  |
| 100              | 0.58 | 18.56| 19.05| 64  |
| 150              | 0.64 | 24.47| 21.31| 63  |
| 200              | 0.70 | 28.14| 22.13| 63  |
| 250              | 0.68 | 27.92| 21.86| 61  |

5.5.5. Minimum Training Data Experiment

Changes in the amount of training data will also affect the final performance of the model. At the same time, by comparing the performance of the model under different amounts of data, we can determine the generation ability of the model. As shown in Table 9, we conduct experiments with the minimum amount of data. The model performance does not drop significantly until the test set drops to 1750. Moreover, DamperGAN has strong robustness and can still learn key feature information on small-scale datasets, which overcomes the shortcomings of the previous model’s poor generalization ability to a certain extent.
Table 9. Minimum training data experimental results.

| The Amount of Training Set | SSIM  | PSNR  | SD    |
|----------------------------|-------|-------|-------|
| 2500 (100%)                | 0.70  | 28.14 | 22.13 |
| 2250 (90%)                 | 0.68  | 27.82 | 21.94 |
| 2000 (80%)                 | 0.65  | 25.86 | 20.25 |
| 1750 (70%)                 | 0.62  | 25.15 | 19.93 |
| 1500 (60%)                 | 0.56  | 23.83 | 17.42 |

5.6. Computational Complexity

The network parameters and training time are recorded to evaluate the space and time complexity of the networks. As shown in Table 10, compared with SelectionGAN, DamperGAN has a similar performance, but its parameters and training time are reduced. Overall, our model outperforms other baselines, which translates to increased temporal and spatial occupancy. However, the consumption of these resources is worth it, because we obtain the highest test results, and the FPS can support the requirements of real-time operation.

Table 10. Network parameters (Param.) and training time of the different models.

| Model          | Param. | Training Time (h) |
|----------------|--------|-------------------|
| Pix2Pix        | 47 M   | 14.92             |
| CRN            | 36 M   | 10.88             |
| X-Fork         | 62 M   | 16.30             |
| X-Seq          | 70 M   | 18.57             |
| SelectionGAN   | 78 M   | 20.06             |
| DamperGAN      | 82 M   | 22.68             |

6. Conclusions

This paper proposes a power line vibration damper detection image-generation model named DamperGAN based on CGAN, which can detect the position of the vibration damper in drone inspection aerial images. DamperGAN first generates low-resolution images based on coarse-grained modules, uses Monte Carlo search to mine latent information in low-resolution images, and then uses the attention mechanism to introduce positive information into fine-grained modules to output high-resolution images, using the penalty mechanism to evaluate the state of the mined intermediate information to improve the convergence effect of the model. We draw the following conclusions through qualitative and quantitative experiments on the self-built DamperGenSet dataset. The detection images generated by DamperGAN are closest to the ground truth in detail texture. Our model outperforms other baselines under multiple evaluation metrics. Sensitivity analysis experiments show that the two-stage generation, Monte Carlo search, and multi-level discriminator utilized in the model all have a positive impact on the final performance. All experimental results show that DamperGAN has the ability to detect vibration dampers in real time using a UAV, which provides a solid foundation for tasks such as corrosion and displacement detection of vibration dampers. In the future, we will continue to explore feasible optimization schemes for DamperGAN and incorporate more power device detection tasks.

Author Contributions: Conceptualization, W.C. and Y.L.; methodology, W.C., Y.L. and Z.Z.; validation, W.C.; formal analysis, W.C. and Y.L.; investigation, W.C., Y.L. and Z.Z.; resources, W.C., Y.L. and Z.Z.; writing—original draft preparation, W.C.; writing—review and editing, W.C., Y.L. and Z.Z.; visualization, W.C.; supervision, Y.L.; project administration, Y.L. All authors have read and agreed to the published version of the manuscript.
Funding: This research was supported by the National Natural Science Foundation of China, grant number 61962031; the National Natural Science Foundation of China, grant number 51667011; and the Applied Basic Research Project of Yunnan province, grant number 2018FB095.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data in this paper are undisclosed due to the confidentiality requirements of the data supplier.

Acknowledgments: We thank the Yunnan Electric Power Research Institute for collecting the transmission line UAV inspection data, which provided a solid foundation for the verification of the model proposed in this paper. At the same time, we thank the reviewers and editors for their constructive comments to improve the quality of this article.

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

References

1. Bao, W.; Ren, Y.; Wang, N.; Hu, G.; Yang, X. Detection of Abnormal Vibration Dampers on Transmission Lines in UAV Remote Sensing Images with PMA-YOLO. Remote Sens. 2021, 13, 4134. [CrossRef]

2. Wu, H.; Xi, Y.; Fang, W.; Sun, X.; Jiang, L. Damper detection in helicopter inspection of power transmission line. In Proceedings of the 2014 4th International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Harbin, China, 18–20 September 2014; pp. 628–632. [CrossRef]

3. Ma, Y.; Li, Q.; Chu, L.; Zhou, Y.; Xu, C. Real-Time Detection and Spatial Localization of Insulators for UAV Inspection Based on Binocular Stereo Vision. Remote Sens. 2021, 13, 230. [CrossRef]

4. Hinas, A.; Roberts, J.M.; Gonzalez, F. Vision-Based Target Finding and Inspection of a Ground Target Using a Multirotor UAV System. Sensors 2017, 17, 2929. [CrossRef]

5. Huang, S.; Han, W.; Chen, H.; Li, G.; Tang, J. Recognizing Zucchinis Intercropped with Sunflowers in UAV Visible Images Using an Improved Method Based on OCRNet. Remote Sens. 2021, 13, 2706. [CrossRef]

6. Popescu, D.; Stoican, F.; Stamatescu, G.; Chenaru, O.; Ichim, L. A Survey of Collaborative UAV–WSN Systems for Efficient Monitoring. Sensors 2019, 19, 4690. [CrossRef]

7. Zhi, Y.; Fu, Z.; Sun, X.; Yu, J. Security and Privacy Issues of UAV: A Survey. Mob. Netw. Appl. 2020, 25, 95. [CrossRef]

8. Coluccia, A.; Fascista, A.; Schumann, A.; Sommer, L.; Dimou, A.; Zarpalas, D.; Méndez, M.; de la Iglesias, D.; González, I.; Mercier, J.-P.; et al. Drone vs. Bird Detection: Deep Learning Algorithms and Results from a Grand Challenge. Sensors 2021, 21, 2824. [CrossRef]

9. Hassanzadeh, A.; Zhang, F.; van Aardt, J.; Murphy, S.P.; Pethybridge, S.J. Broadacre Crop Yield Estimation Using Imaging Spectroscopy from Unmanned Aerial Systems (UAS): A Field-Based Case Study with Snap Bean. Remote Sens. 2021, 13, 3241. [CrossRef]

10. Zhang, Y.; Yuan, X.; Li, W.; Chen, S. Automatic Power Line Inspection Using UAV Images. Remote Sens. 2017, 9, 824. [CrossRef]

11. Pan, L.; Xiao, X. Image recognition for on-line vibration monitoring system of transmission line. In Proceedings of the 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, China, 16–19 August 2009; pp. 3-1078–3-1081. [CrossRef]

12. 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]

13. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. arXiv 2014, arXiv:1411.1784. Available online: https://arxiv.org/abs/1411.1784 (accessed on 5 February 2022).

14. Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural network. In Proceedings of the 25th Neural Information Processing Systems, Lake Tahoe, Nevada, 3–6 December 2012; pp. 1097–1105.

15. Song, W.; Zuo, D.; Deng, B.; Zhang, H.; Xue, K.; Hu, H. Corrosion defect detection of earthquake hammer for high voltage transmission line. Chin. J. Sci. Instrum. 2016, 37, 113–117.

16. Yang, H.; Guo, T.; Shen, P.; Chen, F.; Wang, W.; Liu, X. Anti-vibration hammer detection in UAV image. In Proceedings of the 2017 2nd International Conference on Power and Renewable Energy (ICPRE), Chengdu, China, 20–23 September 2017; pp. 204–207. [CrossRef]

17. Huang, X.; Zhang, X.; Zhang, Y.; Zhao, L. A method of identifying rust status of dampers based on image processing. IEEE Trans. Instrum. Meas. 2020, 69, 5407–5417. [CrossRef]

18. Liu, Y.; Wen, S.; Chen, Z.; Zhang, D. Research of the Anti-vibration Hammer Resetting Robot Based on Machine Vision. In Proceedings of the 2020 Chinese Control and Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 3708–3712. [CrossRef]

19. Miao, S.; Sun, W.; Zhang, H. Intelligent visual method based on wavelet moments for obstacle recognition of high voltage transmission line deicer robot. Jiqiren (Robot) 2010, 32, 425–431. [CrossRef]
20. Jin, L.; Yan, S.; Liu, Y. Vibration damper recognition based on Haar-Like features and cascade AdaBoost classifier. J. Syst. Simul. 2012, 24, 60–63.

21. Liu, Y.; Jin, L. Vibration Damper Recognition of Transmission System Based on Unmanned Aerial Vehicles. In Proceedings of the 2011 Asia-Pacific Power and Energy Engineering Conference, Wuhan, China, 25–28 March 2011; pp. 1–3. [CrossRef]

22. Guo, J.; Xie, J.; Yuan, J.; Jiang, Y.; Lu, S. Fault Identification of Transmission Line Shockproof Hammer Based on Improved YOLO V4. In Proceedings of the 2021 International Conference on Intelligent Computing, Automation and Applications (ICCAA), Nanjing, China, 25–27 June 2021; pp. 826–833. [CrossRef]

23. Zhang, K.; Hou, Q.; Huang, W. Defect Detection of Anti-vibration Hammer Based on Improved Faster R-CNN. In Proceedings of the 2020 7th International Forum on Electrical Engineering and Automation (IFEEA), Hefei, China, 25–27 September 2020; pp. 889–893. [CrossRef]

24. Bao, W.; Ren, Y.; Liang, D.; Yang, X.; Xu, Q. Defect Detection Algorithm of Anti-vibration Hammer Based on Improved Cascade R-CNN. In Proceedings of the 2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Sanya, China, 4–6 December 2020; pp. 294–297. [CrossRef]

25. Zhao, Z.; Zhen, Z.; Zhang, L.; Qi, Y.; Kong, Y.; Zhang, K. Insulator Detection Method in Inspection Image Based on Improved Faster R-CNN. Energies 2019, 12, 2104. [CrossRef]

26. Liu, C.; Wu, Y.; Liu, J.; Sun, Z. Improved YOLOv3 Network for Insulator Detection in Aerial Images with Diverse Background Interference. Electronics 2021, 10, 771. [CrossRef]

27. Sadykova, D.; Pernebeyeva, D.; Bagheri, M.; James, A. IN-YOLO: Real-Time Detection of Outdoor High Voltage Insulators Using UAV Imaging. IEEE Trans. Power Deliv. 2020, 35, 1599–1610. [CrossRef]

28. Chen, W.; Li, Y.; Li, C. A Visual Detection Method for Foreign Objects in Power Lines Based on Mask R-CNN. International J. Ambient. Comput. Intell. (IJACI) 2020, 11, 34–47. [CrossRef]

29. Kang, G.; Gao, S.; Yu, L.; Zhang, D. Deep Architecture for High-Speed Railway Insulator Surface Defect Detection: Denoising Autoencoder With Multitask Learning. IEEE Trans. Instrum. Meas. 2019, 68, 2679–2690. [CrossRef]

30. Cheng, L.; Liao, R.; Yang, L.; Zhang, F. An Optimized Infrared Detection Strategy for Defective Composite Insulators According to the Law of Heat Flux Propagation Considering the Environmental Factors. IEEE Access 2018, 6, 38137–38146. [CrossRef]

31. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. ICLR. arXiv 2014, arXiv:1312.6114. Available online: https://arxiv.org/abs/1312.6114v10 (accessed on 5 February 2022).

32. Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D.; Wierstra, D. DRAW: A Recurrent Neural Network for Image Generation. In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), Lille, France, 6–11 July 2015; Volume 37, pp. 1462–1471. Available online: http://proceedings.mlr.press/v37/gregor15.html (accessed on 30 August 2021).

33. Yan, X.; Yang, J.; Sohn, K.; Lee, H. Attribute2Image: Conditional Image Generation from Visual Attributes. In Computer Vision–ECCV 2016. Lecture Notes in Computer Science; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Amsterdam, The Netherlands, 2016; Volume 9908; 2016, Volume 9908. [CrossRef]

34. Tang, X.L.; Du, Y.M.; Liu, Y.W.; Li, J.X.; Ma, Y.W. Image recognition with conditional deep convolutional generative adversarial networks. Zidonghua Xuebao/Acta Autom. Sin. 2018, 44, 855–864. [CrossRef]

35. Dumoulin, V.; Belghazi, I.; Poole, B.; Mastropietro, O.; Lamb, A.; Arjovsky, M.; Courville, A. Adversarially Learned Inference. arXiv 2016, arXiv:1606.00704. Available online: https://arxiv.org/abs/1606.00704v3 (accessed on 5 February 2022).

36. Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I.; Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS16), Barcelona, Spain, 5–10 December 2016; pp. 2180–2188.

37. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2019, 41, 1947–1962. [CrossRef]

38. Johnson, J.; Gupta, A.; Li, F.-F. Image Generation from Scene Graphs. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1219–1228. [CrossRef]

39. Chiarion, F.; Rahal, M.; Hueber, N.; Dufaux, F. Hallucinating A Cleanly Labeled Augmented Dataset From A Noisy Labeled Dataset UsingGAN. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 3616–3620. [CrossRef]

40. 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 (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5967–5976. [CrossRef]

41. Choi, Y.; Choi, M.; Kim, M.; Ha, J.; Kim, S.; Choo, J. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8789–8797. [CrossRef]

42. Dosovitskiy, A.; Springenberg, J.T.; Tatorchenko, M.; Brox, T. Learning to Generate Chairs, Tables and Cars with Convolutional Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 692–705. [CrossRef]

43. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2223–2232. [CrossRef]

44. Zhang, Y.; Yang, Q. A Survey on Multi-Task Learning. IEEE Trans. Knowl. Data Eng. 2021, 29, 2367–2381. [CrossRef]
45. Ronneberger, O. Invited Talk: U-Net Convolutional Networks for Biomedical Image Segmentation. In Bildverarbeitung für die Medizin 2017, Informatik aktuell; Maier-Hein, G., Fritzschke, K., Deserno, G., Lehmann, T., Handels, H., Tolsdorff, T., Eds.; Springer Vieweg: Berlin/Heidelberg, Germany, 2017; Volume 12. [CrossRef]

46. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv 2016, arXiv:1607.08022. Available online: https://arxiv.org/abs/1607.08022v3 (accessed on 5 February 2022).

47. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. Comput. Sci. 2015, 14, 38–39. [CrossRef]

48. Chen, Q.; Koltun, V. Photographic image synthesis with cascaded refinement networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1520–1529. [CrossRef]

49. Regmi, K.; Borji, A. Cross-view image synthesis using conditional gans. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3501–3510. [CrossRef]

50. Tang, H.; Xu, D.; Sebe, N.; Wang, Y.; Corso, J.J.; Yan, Y. Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2412–2421. [CrossRef]

51. Deng, J.; Dong, W.; Socher, L.; Li, L.; Li, K.; Li, F. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]

52. Mathieu, M.; Couprie, C.; LeCun, Y. Deep Multi-Scale Video Prediction Beyond Mean Square Error. arXiv 2016, arXiv:1511.05440. Available online: https://arxiv.org/abs/1511.05440v6 (accessed on 5 February 2022).