



InsulatorGAN: A Transmission Line Insulator Detection Model Using Multi-Granularity Conditional Generative Adversarial Nets for UAV Inspection

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: Insulator detection is one of the most significant issues in high-voltage transmission line inspection using unmanned aerial vehicles (UAVs) and has attracted attention from researchers all over the world. The state-of-the-art models in object detection perform well in insulator detection, but the precision is limited by the scale of the dataset and parameters. Recently, the Generative Adversarial Network (GAN) was found to offer excellent image generation. Therefore, we propose a novel model called InsulatorGAN based on using conditional GANs to detect insulators in transmission lines. However, due to the fixed categories in datasets such as ImageNet and Pascal VOC, the generated insulator images are of a low resolution and are not sufficiently realistic. To solve these problems, we established an insulator dataset called InsuGenSet for model training. InsulatorGAN can generate high-resolution, realistic-looking insulator-detection images that can be used for data expansion. Moreover, InsulatorGAN can be easily adapted to other power equipment inspection tasks and scenarios using one generator and multiple discriminators. To give the generated images richer details, we also introduced a penalty mechanism based on a Monte Carlo search in Insulator-GAN. In addition, we proposed a multi-scale discriminator structure based on a multi-task learning mechanism to improve the quality of the generated images. Finally, experiments on the InsuGenSet and CPLID datasets demonstrated that our model outperforms existing state-of-the-art models by advancing both the resolution and quality of the generated images as well as the position of the detection box in the images.

Keywords: unmanned aerial vehicle (UAV); power transmission line inspection; insulator detection; conditional generative adversarial nets; image generation

1. Introduction

Insulators provide electrical insulation and mechanical support for overhead transmission lines. Insulator detection refers to the process of locating the position of the insulator in inspection images. This process serves as the foundation for other tasks such as insulator defect detection and power line extraction. Recently, insulator detection has attracted significant interest from researchers in many fields such as smart grid and computer vision, and some progress has been made [1,2].

At present, the inspection of insulators mainly depends on the visual observation of the staff, which can easily cause omissions, unmanned aerial vehicles (UAV), with the advantages of low cost, small size, and multiple cruise modes [3], combined with other systems such as wireless sensor networks (WSNs) [4], which have become important auxiliary equipment in many fields such as crop identification [5] and yield estimation [6]. On the other hand, UAVs bring safety hazards and privacy issues, such as threats to the flight activities of birds [7]. In addition, the communication between the drone and Ground Control Station (GCS) is vulnerable to attacks, which can cause data leakage [8]. However,



Article

Citation: Chen, W.; Li, Y.; Zhao, Z. InsulatorGAN: A Transmission Line Insulator Detection Model Using Multi-Granularity Conditional Generative Adversarial Nets for UAV Inspection. *Remote Sens.* **2021**, *13*, 3971. https://doi.org/10.3390/ rs13193971

Academic Editor: Paolo Tripicchio

Received: 2 September 2021 Accepted: 28 September 2021 Published: 4 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). in general, the convenience brought by drones is higher. Therefore, we believe that it is feasible to use drones for power line inspections. Currently, researchers have used drones for power line identification and monitoring tasks [9], This paper will focus on the research of insulator detection algorithms using images acquired by UAVs.

In past studies, researchers have generally used feature extraction combined with classifiers for insulator detection [10,11]. This type of method usually uses artificially designed operators to traverse the image to obtain the edge and texture of the insulator. This process is followed by insulator detection based on feature information. However, such methods rely heavily on the completeness of the features. As the features used cannot cover all situations, it is difficult to obtain good detection results. In addition, the feature classifier requires significant manual work, and the level of automation is low, making it difficult to apply it on a large scale in reality.

In recent years, some researchers have proposed the use of convolutional neural networks (CNNs) for insulator detection. A power line insulator detection model was proposed in [12] for different environmental backgrounds based on YOLO. In [13], a frame-work based on Faster R-CNN was proposed for different types of insulators. Currently, significant redundant information remains in the detection results from the state-of-the-art models used for object detection. An improvement in precision comes at the cost of an increase in the quantity of data and parameters. For situations with high real-time requirements and small amounts of data, it is difficult for CNNs to achieve excellent detection results. In addition, overhead transmission lines are mostly located in mountains, lakes, and suburbs. Those environments are complex, which increases the difficulty of model convergence.

One way to solve the above problems is to use a conditional Generative Adversarial Nets (CGAN) [14]. CGAN uses aligned images to enable the framework to learn the relevant mapping from input to output. In this way, the original image and ground truth can be used to generate an insulator-detection image. Some researchers have achieved good results using CGAN in image conversion tasks. For example, a network model named pix2pix [15], proposed by Isola et al., was used to complete the conversion between images. The pix2pix model achieved an excellent performance, but the resolution of the generated image was only 256 * 256. The researchers in [16] noted that image generation using an adversarial network was not reliable enough; instead, the authors used perceptual loss to obtain the image. The resolution of the generated image was improved, but the performance of the semantic details was poor.

The researchers in [17] sought to introduce semantic guidance in image generation and proposed two multi-view image-generation models, X-Fork and X-Seq. The authors inputted an image of a known perspective and the semantic segmentation map of the target perspective together into the model and provided semantic guidance to make the generated image more realistic. Inspired by the work in [17], the researchers in [18] proposed a SelectionGAN model based on a multi-channel attention mechanism. The SelectionGAN model further expanded the semantic generation space. This model improved the semantic details in the image by referring to the intermediate results generated and achieved advanced results in the translation of satellite images and ground images. However, the above techniques have poor compatibility with transmission line scenarios because not all scenarios have sufficient semantic segmentation maps for model training.

To solve the above problems, this paper proposes an insulator-detection imagegeneration model called InsulatorGAN based on an improved conditional generation confrontation network. Our model can be flexibly adapted to a variety of power component inspection tasks and other application scenarios. InsulatorGAN contains a generator and multiple discriminators. As shown in Figure 1, our model generates images through two steps: first, it uses a coarse image module to generate a low-resolution image (LR image); next, a fine image module is used to improve the semantic information in the LR image and to generate a high-resolution image (HR Image).



Figure 1. Test examples of InsulatorGAN on InsuGenSet and CPLID. First, the LR Image is generated, then the details are enriched to obtain the HR Image.

The differences between InsulatorGAN and previous models are as follows. First, the generator of InsulatorGAN uses multiple stages, which ensures that InsulatorGAN can not only generate high-resolution and realistic images but can also be applied to other fields, such as image translation and style conversion. Second, to strengthen the semantic constraints in the image-generation process, we used the Monte Carlo search (MCS) [19] to sample low-resolution target images multiple times and calculate the corresponding penalty value according to the sampling results. The penalty mechanism can force the generator to produce images with richer semantics to avoid mode collapse [20]. Third, to improve the ability of the discriminator, based on the multi-task learning strategy of parameter sharing [21], we proposed a discriminator framework based on a multi-scale structure. Although all the discriminators use the same network structure, the input of different resolutions allows the discriminators to cooperate with each other, extract feature maps at different abstraction levels, and accelerate the training of the model. In addition, to solve the problem where the public dataset CPLID [1] features less data and a simple background, we used the images obtained via UAV to build the insulator imagegeneration dataset, InsuGenSet. Moreover, the insulator detection results output by the InsulatorGAN model can be used for data expansion. We compared InsulatorGAN with several mainstream image-generation models and achieved the best results on the CPLID and InsuGenSet, which demonstrates that our model can generate high-resolution and high-quality images. The main contributions of this paper are as follows:

- This paper proposes an insulator-detection image-generation model, InsulatorGAN, based on an improved conditional Generative Adversarial Nets. This model includes a generator and multiple discriminators. Moreover, we used a two-stage method from coarse to fine to generate high-resolution insulator inspection images that can be flexibly adapted to other scenes;
- To improve the constraints on the insulator image-generation process, a penalty mechanism based on the Monte Carlo search was introduced into the generator. This mechanism enables the generator to obtain sufficient semantic guidance and add more semantic details to the generated image;
- Based on the parameter sharing mechanism, we propose a multi-scale discriminator structure that enables the entire discriminator network to use feature information at different levels of abstraction to determine whether the input image is true or false;
- To solve the small scale of the public insulator dataset CPLID, we established a dataset called InsuGenSet for insulator-detection image generation based on real images. We conducted many comparative experiments between the InsulatorGAN and state-of-the-art models on InsuGenSet, and the results demonstrated the effectiveness and flexibility of InsulatorGAN.

The rest of this article is arranged as follows. Section 2 introduces the related work on insulator detection and image generation. Section 3 introduces the knowledge on GAN. The architecture of InsulatorGAN is illustrated in Section 4. In Section 5, we present several sets of experiments that determined the effectiveness of our framework. Finally, the conclusions and future work are outlined in Section 6.

2. Related Work

2.1. Insulator Detection

Insulators provide electrical insulation and mechanical support for overhead transmission lines. Insulator detection refers to the process of locating the position of the insulator in an inspection image. This process provides the foundation for other tasks such as insulator defect detection and power line extraction. Recently, insulator detection has attracted significant interest from researchers in many fields such as smart grids and computer vision, and some progress has been made [1,2].

Insulator detection is a major issue in power line inspection and provides a reference for insulator defect detection, foreign object detection, wire detection, and robot path planning. With the rise of deep convolutional neural networks in the image field, insulator detection has also been given new life. The researchers in [1] proposed a cascaded architecture for insulator defect detection.

Sadykova et al. [12] used the YOLOv3 neural network model to train a real-time insulator detection classifier under varying image resolutions and different lighting conditions to assess the presence of ice, water, and snow on the insulator. Zhao et al. [13] improved the anchor generation method and the non-maximum suppression algorithm in the Faster R-CNN model for use with different sizes and aspect ratios and the mutual occlusion of insulators in aerial images. To enhance the reuse and spread of insulator features, Liu et al. [22] added a multi-level feature mapping module based on YOLOv3 and Dense Blocks and proposed an insulator detection network called YOLOv3-dense. Focusing on the interference of complex backgrounds and small targets, the researchers in [23] proposed two insulator detection models, Exact R-CNN and CME-CNN. CME-CNN added an encoder–decoder based on Exact R-CNN to obtain pure insulators.

However, the excellent performance of the above models have generally been obtained at the cost of a complex network structure. In this case, either the real-time performance of the network is difficult to guarantee, or the detection precision is not sufficient. The model in this paper is an improved conditional generative adversarial network with a simple framework and clear training methods. With the support of a small amount of data, high detection precision can be obtained in a short time, and the generated images can be used for data expansion, which solves the problems of the aforementioned work.

2.2. Image Generation

Image generation has become a hot research topic due to the rise of deep learning. Variational autoencoder (VAE) [24] is a generative model based on the probability graph model. In [25], the researchers proposed an Attribute2Image model that can generate images from visual features by synthesizing the foreground and background. In addition, researchers in [26] introduced an attention mechanism into VAE and proposed a DRAW model to improve the image quality.

Recently, researchers have achieved favorable performance in image generation using GAN [27]. The training goal of the generator and the discriminator is for the two to defeat each other. A large number of GAN-based frameworks were subsequently proposed, such as conditional GANs [28], Bi-GANs [29], and InfoGANs [30]. GAN can also generate new images based on labels [31], text [32,33], or images [15,34,35].

However, the images generated by the above models generally have problems such as blurring and distortion. The model does not learn how to generate images but simply repeats the content of the images in the training set. InsulatorGAN is also an image-based CGANs, but our model introduces more sufficient semantic guidance with a novel penalty mechanism, further improves the semantic architecture of the image, and overcomes the problem of image distortion.

3. Basic Knowledge of GAN

The Generative Adversarial Network [27] includes two adversarial learning subnetworks, a discriminator, and a generator, which are trained using the maximum–minimum game theory. The generator G obtains an image via a d-dimensional noise vector and produces a generated image as close as possible to the real image. On the other hand, the discriminator D is used to determine whether the input is a fake image from the generator or a real image from a real dataset. The loss function of the entire generative adversarial network is as follows:

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

where *x* represents the real image sampled from the real data distribution p_r , and *z* represents the d-dimensional noise vector sampled from the Gaussian distribution p_g .

CGANs [14] control the results of model generation by introducing auxiliary variables. In the CGANs, the generator generates images based on auxiliary conditions, and the discriminator makes judgments based on auxiliary conditions and images (false images or real images). The loss function is as follows:

$$\min_{G} \max_{D} V_{CGAN}(G, D) = E_{s, x \sim p_r(s, x)} [\log D(s, x)] + E_{s, x' \sim p_r(s, x')} [\log(1 - D(s, x'))]$$
(2)

where *s* represents the auxiliary variable, and x' = G(s, z) represents the image generated by the generator.

In addition to fighting against loss, previous works [34,36] have also sought to minimize the L1 or L2 distance between the real and the fake images to help the generator synthesize images with greater similarities to the real images. Previous research has proven that, compared with the L2 distance, the L1 distance can help the model reduce blur and distortion in the image. Therefore, the L1 distance was also introduced into InsulatorGAN. The formula for minimizing the L1 distance is as follows:

$$\min_{G} V_{L1}(G) = E_{x,x' \sim p_r(x,x')}[\|x - x'\|_1]$$
(3)

The loss function for this type of CGANs is the sum of Equations (2) and (3).

4. InsulatorGAN

4.1. Task Definition

First, we must define the task of the insulator-detection image generation. Suppose we have a clear image of a line insulator taken by a drone, and we mark that image as the original image I_o . After the insulators in the image are marked, the image is recorded as the ground truth I_t . The insulator-detection image-generation task refers to the generated image I_g containing the box, which is very similar to I_t in terms of visual effects.

4.2. Overall Framework

The framework of InsulatorGAN proposed in this paper is shown in Figure 2. Our goal was to generate an insulator-detection image I_g . For this purpose, we used a generator $G(I_o; \theta_g)$ and a discriminator $D((I_o, I_g); \theta_d)$, where θ_g and θ_d , respectively, represent the parameters of the generator and discriminator; I_o represents the image sampled from the real data distribution; and I_g represents the image generated by the generator G.



Figure 2. The framework of InsulatorGAN.

The training of the entire model can be divided into two adversarial learning processes: generator learning and discriminator learning. The target of generator G is to generate an image from the insulator detection results and make the generated images cheat the discriminator D. In other words, the target of the generator is to minimize the distance between the fake image and the target image. Correspondingly, the goal of the discriminator is to accurately determine whether the input is a fake image generated by the generator or a ground truth and to calculate a penalty value for the fake image.

4.3. Multi-Granularity Generator

As shown in Figure 3, the process of image generation is divided into three parts: First, input the original image I_o and the target image I_t . Here, the generator uses coarse-grained modules to obtain low-resolution images I'_g . Then, the Monte Carlo search method is introduced to sample the low-resolution image N times. Finally, the attention mechanism is introduced to extract the features of the N intermediate result images, and the output of the attention mechanism and the original image I_o are input into the fine-grained output module. Finally, we obtain the high-resolution target image I_g . We hoped to improve the quality of the image through the two-stage generation method, use the Monte Carlo search to mine the hidden spatial information of the samples output by the coarse-grained module, combine the penalty mechanism and the attention mechanism to constrain the position of the generated box, and to improve image resolution and detail performance.



Figure 3. The framework of the generator containing three parts: the coarse-grained module, the Monte Carlo search, the and fine-grained module. First, the coarse-grained module is used to obtain the LR image; then, the Monte Carlo search is used for detail mining; lastly, the attention mechanism is used to input the detailed information into the Fine-grained module to obtain the HR Image.

7 of 20

4.3.1. Penalty Mechanism

To strengthen the semantic constraints during image generation and to further improve the semantic details in the image, we propose a penalty mechanism based on the Monte Carlo search (MCS) strategy, which makes the generated image more accurate in its semantics and details.

We performed Monte Carlo searches on the low-resolution target image I'_g output by the coarse-grained generation module. The specific process can be outlined as follows:

$$\left\{I_g^1, \dots, I_g^N\right\} = M C^{G_\beta}(I'_g; N) \tag{4}$$

$$I_g^i = G_\beta(I'_g, z) \tag{5}$$

where $I_g^i(i = 1, \dots, N)$ represents the image obtained based on I'_g sampling, and MC^{G_β} represents the state of the MCS simulation. G_β represents the generation module virtualized by MCS technology and shares parameters with the fine-grained generation module. This module generates N intermediate result images continuously for N times. Here, $z \sim N(0, 1)$ is the noise vector introduced to ensure the diversity of the sampling results. We use the sum of the output of the encoder and the noise vector as the input of the decoder. The noise vector ensures that each sampling model focuses on the feature vector differently and can extract richer semantic information from the low-resolution target image.

Next, *N* intermediate result images and images with a known angle of view are sent to the discriminator. Then, we can calculate the penalty value based on the output of the discriminator. The calculation process is as follows:

$$V_D^G = \frac{1}{N} \sum_{i=1}^N \left(\left(1 - D(I_g^i; \theta_d) \right) + \|I_g - I_t\|_1 \right)$$
(6)

where $D(I_g^i; \theta_d)$ represents the probability that the image I_g^i output by the discriminator is a real image.

4.3.2. Attention Mechanism

After obtaining N intermediate result images through sampling, we refer to the intermediate result images to provide sufficient semantic guidance for the next stage of generation. For this process, we propose a multi-channel attention mechanism. This mechanism is different from the methods used in previous work, where the synthesized image was generated only from the RGB three-channel space. We use sampling to obtain N intermediate results as feature sets to build a larger semantic generation space, and the model then extracts more fine-grained information by referring to the information from different channels. Then, the calculation result is input into the fine-grained generation module to obtain the high-resolution target image.

Next, we must perform a convolution operation on N intermediate result images to obtain the corresponding attention weight matrix. The results are calculated as follows:

$$I_A^i = \text{Softmax}(I_g^i W_A^i + b_A^i), \text{ for } i = 1, \dots, N$$
(7)

where $\{W_A^i, b_A^i\}_{i=1}^N$ represents the parameter of the convolution operation and Softmax(·) represents the softmax function for normalization. We then multiply the resulting attention weight matrix with the corresponding intermediate result image to obtain the final output:

$$I''_{g} = (I_{A}^{1} \otimes I_{g}^{1}) \oplus \dots \oplus (I_{A}^{N} \otimes I_{g}^{N})$$

$$\tag{8}$$

where I''_g in the formula represents the final output result of the attention mechanism, and the symbols \oplus and \otimes represent the addition and multiplication elements of the matrix, respectively.

The generator continues optimization by minimizing the following objective function:

$$J_G(\theta_g) = \mathbb{E}_{I_o \sim P_r(I_o), I_t \sim P_t(I_t)} [\log(1 - D(I_o, G(I_o, I_t)))] + V_G^D$$
(9)

where $D(I_o, G(I_o, I_t))$ represents the score of the generated image obtained by the generator *G* using the original image and the ground truth (I_o, I_t) . In other words, this represents the probability that the discriminator *D* considers the generated image to be the ground truth, θ_g represents the parameters of the generator, and V_G^D represents the penalty value for generator *G* obtained using Formula (6).

4.4. Multi-Level Discriminator

The architecture of the discriminators is shown in Figure 4. The discriminator is composed of a deep convolutional network with a simple architecture. Widening and deepening the network to improve the discrimination ability of the discriminator will inevitably lead to a sharp increase in the model's parameters, which will increase the model's training time.



Figure 4. The framework of discriminators. The discriminator consists of three parts with the same structure: D_1 , D_2 , and D_3 . First, the shared convolutional layer is used to process the input image to obtain the feature map; after that, the map is down-sampled 2 times and 4 times and input into D_1 , D_2 , and D_3 to obtain the discriminator scores.

4.4.1. Multitasking Mechanism

A discriminator network based on a multi-scale architecture is proposed in this paper. This network uses three discriminators, D_1 , D_2 , D_3 , with the same structures to process images at three different scales. Based on the parameter sharing mechanism [21], a multi-task learning strategy is used in this paper for the training of the discriminator. We first extracted the basic characteristics of the real sample and the generated sample through the convolutional network and obtained the corresponding feature map. Then, we used 2 and 4 as sampling factors to down-sample the feature maps of the real and generated images to obtain feature maps at three different scales. Moreover, we used the three discriminators D_1 , D_2 , D_3 , to process the feature maps.

Although each discriminator has the same structure, a discriminator with a smaller input can enhance the semantic information of the generated image, while a larger input is superior at adding details to the generated image. Thus, the design of a multi-level discriminator is very conducive to model training. When a high-resolution model is desired,

we only need to add discriminators to the original model instead of training the model from zero.

4.4.2. Objective Function

The formula for the above multi-task learning process is as follows:

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

where P_r and P_g represent the sample distribution of the real images and generated images, respectively; I_o represents the original image; I_g represents the generated image obtained using the original image and ground truth; θ_d represents the parameters of the discriminator; and D_k represents one of the multiple discriminators.

We then conducted adversarial learning between the generator and the discriminators. The optimization algorithm is shown in Algorithm 1.

Algorithm 1. The Training Process of InsulatorGAN.

Input: Real insulator image dataset $I_0 = \{I_{o_1}, \dots, I_{o_N}\}$; Generator *G*; Discriminator $\{D_i\}_{i=1}^{i=k}$; 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 $\{D_i\}_{i=1}^{i=k}$ with random weights;

- 2: repeat
- 3: for g-steps do
- 4: *G* generate fake images;
- 5: Calculate the penalty value V_D^G via Formula (6);
- 6: Minimize Formula (9) to update the parameters of the generator *G*;
- 7: end for
- 8: for d-steps do
- 9: Use *G* to generate fake images $I_g = \{I_{g_1}, \dots, I_{g_N}\}$;

10: Use real images $I_o = \{I_{o_1}, \dots, I_{o_N}\}$ and fake images $I_g = \{I_{g_1}, \dots, I_{g_N}\}$ to up-date the discriminator parameters by minimizing Formula (10);

11: end for

- 12: until InsulatorGAN completes convergence
- 13: return

4.5. Network Structure

For the problem of insulator-detection image generation, there is a large amount of underlying feature sharing between the input and output. To allow these features to be transmitted stably in the network, we used U-net [37] as the basic structure for the generator and discriminators.

The structure of the generator is shown in Table 1. CONV represents a convolutional layer; N-m indicates that each convolutional layer contains m convolution kernels; K-mxm indicates that the scale of the convolution kernel is mxm; S-m indicates that the moving step size of the convolution kernel is m; P-m represents the number of boundary expansions required for the convolution kernel to iterate on the image; and IN, ReLU indicates that the loss function of the current convolution layer is the InstanceNorm-ReLU layer [38].

The structure of the discriminator is shown in Table 2. Before the image is sent to the discriminator, it is necessary to use a convolution kernel with a size of 3×3 , an offset step of 1, and an edge padding of 1 to extract the primary features in the image. After the feature map with the same size as the original sample image is obtained, that map is down-sampled by 2 times and 4 times. Then, the feature map obtained after down-sampling is sent to the discriminator of the corresponding size. Notably, the first layer uses Convolution-InstanceNorm-LeakyReLU [39] as the activation function without normalization. The slope of Leaky ReLU is 0.2. After the last convolutional layer of the discriminators, a fully connected layer is used to generate a one-dimensional output. The discriminators with different input resolutions are the same in terms of network structure.

Network	Layer Information	Input	Output
	CONV-(N64, K7x7, S1, P3), IN, ReLU	(512, 512, 3)	(512, 512, 64)
	CONV-(N128, K3x3, S2, P1), IN, ReLU	(512, 512, 64)	(256, 256, 128)
Down-Sample	CONV-(N256, K3x3, S2, P1), IN, ReLU	(256, 256, 128)	(128, 128, 256)
	CONV-(N512, K3x3, S2, P1), IN, ReLU	(128, 128, 256)	(64, 64, 512)
	CONV-(N1024, K3x3, S2, P1), IN, ReLU	(64, 64, 512)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
Residual Block	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N1024, K3x3, S1, P1), IN, ReLU	(32, 32, 1024)	(32, 32, 1024)
	CONV-(N512, K3x3, S0.5, P1), IN, ReLU	(32, 32, 1024)	(64, 64, 512)
	CONV-(N256, K3x3, S0.5, P1), IN, ReLU	(64, 64, 512)	(128, 128, 256)
Up-Sample	CONV-(N128, K3x3, S0.5, P1), IN, ReLU	(128, 128, 256)	(256, 256, 128)
	CONV-(N64, K3x3, S0.5, P1), IN, ReLU	(256, 256, 128)	(512, 512, 64)
	CONV-(N3, K7x7, S1, P3), IN, ReLU	(512, 512, 64)	(512, 512, 3)

 Table 1. The architecture of the generator.

Table 2. The architecture of the discriminators.

Network	Layer Information	Input	Output
	CONV-(N64, K4x4, S2, P2), Leaky ReLU	(512, 512, 3)	(256, 256, 64)
Input Lavor	CONV-(N128, K4x4, S2, P2), IN, ReLU	(256, 256, 64)	(128, 128, 128)
input Layer	CONV-(N256, K4x4, S2, P2), IN, ReLU	(128, 128, 128)	(64, 64, 256)
	CONV-(N512, K4x4, S2, P2), IN, ReLU	(64, 64, 256)	(32, 32, 512)

5. Experiments and Analysis

5.1. Experiment Description

5.1.1. Dataset

First, we needed an insulator dataset with sufficient images to analyze and evaluate the model. The current public dataset for insulator detection is only CPLID, but most of the insulator entities are obtained through image augmentation methods such as rotation and cropping. The real samples included are limited and cannot fully reflect the characteristics of insulator samples from power lines. Therefore, we used the UAV to record a 500 KV overhead power line inspection video in a location in China as the data source and produced a dataset to be used in generating insulator-detection images. Moreover, we used AutoCAD as a labeling tool to draw the smallest external polygonal frame around all the insulator entities. To improve the separation between the insulator and the background, we used magenta to draw the frame. This color is very rare in nature, which improves the model's focus on insulation during training. We named this dataset InsuGenSet, and the training set and test set each contained 2500 and 500 image pairs composed of real images and ground-truth values.

5.1.2. Experiment Configuration

InsulatorGAN training involved a total of 200 epochs. The first 100 epochs were maintained at a 0.0001 learning rate, while the next 100 epochs gradually shrunk to 0. Before starting the training, the parameters of the model were initialized using a normal distribution with a mean value of 0 and a standard deviation of 0.02. An NVIDA RTX 2080 GPU with a memory capacity of 8 GB was used for training and testing. The operating system was Ubuntu 18.04, and the PC had a memory capacity of 32 GB. All algorithms

were built using Pytorch1.4. The number of Monte Carlo searches was 5, and the loss proportion of the control discriminator and feature extraction matching was 10.

5.2. The Baselines

In this study, we compared InsulatorGAN with the current mainstream models of image generation, image translation, and object detection.

Pix2Pix [15]: This method uses anti-loss learning to map from x to y, where x and y represent images in different domains; this method has achieved excellent results in image-translation tasks.

CRN [16]: Unlike the method used by GAN for image generation, the CRN model uses a CNN to output a corresponding generated image based on an input semantic layout image. This method creatively uses the technique of calculating the matching loss between images and calculates the diverse loss in the generated and semantic-segmentation images.

X-Fork [17]: Similar to pix2pix, this generator produces multi-view images by learning the mapping of G: { $I_a \rightarrow I_b$, S_b }. I_a and I_b represent the images under perspective a and perspective b, respectively, and S_b represents the semantic segmentation map of perspective b.

X-Seq [17]: X-Seq uses two CGANs (G1 and G2) together as a model, where G1 synthesizes the image of the target perspective, and G2 synthesizes the semantic segmentation map of the target perspective based on the output image of G1.

SelectionGAN [18]: This model introduces a multi-channel attention mechanism based on X-seq to selectively learn the intermediate results of the model, thereby generating cascading images from coarse to fine.

Cascade R-CNN [40]: Cascade R-CNN is one of the baselines in object detection. The detection speed of this method is slow, but the algorithm offers high precision and robustness.

YOLOv4 [41]: Based on the YOLO structure, this is the most recent optimization strategy and was adopted to achieve a balance between FPS and precision.

CenterNet [42]: Unlike Faster R-CNN and YOLOv4, this model converts object detection into key-point detection based on a heatmap.

5.3. Quantitative Evaluation

In this section, the inception score and Fréchet inception distance are used to evaluate the quality of the generated image. Then, we use AP50-AP90 to evaluate the accuracy of the box position in the generated image. Finally, some pixel-level indicators are used to evaluate the similarity between the fake image and ground truth.

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

The Inception Score (IS) is a universal model index for generative models that can measure the clarity and diversity of images generated by the model. A higher *IS* indicates a higher quality. The calculation formula is as follows:

$$IS(G) = \exp\left(\mathbb{E}_{x \sim p_g} D_{KL}(p(y|x) \parallel p(y))\right)$$
(11)

where *G* represents the generator, *x* represents the generated image, *y* represents the predicted label of the generated image, and D_{KL} represents the *KL* divergence (Kullback–Leibler Divergence), also called relative entropy.

Because the CPLID and InsuGenSet datasets contain insulators that are not included in the ImageNet dataset [43], we could not directly use the pre-trained Inception model. Moreover, the Inception model features a large number of parameters, so we instead used AlexNet to score the InsulatorGAN. The Fréchet Inception Distance (FID) is an index used to calculate the distance between a real image and a fake image. A lower *FID* value for an image indicates a higher image quality. The calculation formula is as follows:

$$FID(I_t, I_g) = \|\mu_{I_t} - \mu_{I_g}\|^2 + T_r \left(\sum_{I_t} + \sum_{I_g} -2\left(\sum_{I_t} \sum_{I_g}\right)^{\frac{1}{2}}\right)$$
(12)

where T_r represents the sum of the elements on the diagonal of the matrix μ_{I_t} , μ_{I_g} represents the mean of the feature map of the ground truth I_t and the generated image I_g , and Σ_{I_t} and Σ_{I_g} represent the covariance matrix of the feature map of I_t and I_g . In order to increase the calculation speed and prevent the occurrence of underfitting due to the small scale of dataset, similar to IS, we used AlexNet instead of InceptionV3 to obtain the basic features of the image. We used the output of the last pooling layer before the fully connected layer, that is, a 1 * 1 * 4096 vector as the basic feature of the input image. On this basis, we could obtain the covariance matrix of 4096 * 4096, which was Σ_{I_t} and Σ_{I_g} .

Table 3 outlines the experimental results for IS and FID. Here, the score of Insulator-GAN was higher than the scores of the other baseline architectures, which indicates that the images generated by InsulatorGAN are superior, in terms of quality and diversity, to those generated by other baseline models.

Table 3. IS and FID of the different models.

Model –	InsuC	GenSet	СР	LID
	IS	FID	IS	FID
Pix2Pix	3.36	57.34	3.19	58.25
CRN	3.02	57.58	2.92	58.45
X-Fork	3.48	56.89	3.26	57.93
X-Seq	3.73	56.53	3.69	57.68
SelectionGAN	3.86	56.17	3.77	57.31
InsulatorGAN	4.05	55.23	3.98	56.83

5.3.2. Precision of Box

Here, we evaluate the generation position of the detection frame based on the Average Precision (AP) of the COCO dataset [44]. The result of this standard must be based on the Intersection of Union (IoU) of the fake image and the Ground truth, which is used as the basis for the threshold setting of the correct standard. The formula is shown below:

$$IoU = \frac{DefaultBox \cap GroundTruthBox}{DefaultBox \cup GroundTruthBox}$$
(13)

where *DefaultBox* represents the position of the predicted frame, and *GroundTruthBox* represents the location of the ground truth. The ratio of the intersection and the union of the two is used to indicate the precision of the predicted frame.

As shown in Table 4, although the precision of InsulatorGAN was not significantly improved compared with that of Cascade R-CNN, the precision was still higher than that of Yolov4 and CenterNet. This result indicates that InsulatorGAN can locate the position of the insulator in the generated image effectively and achieve the goal of insulator detection. In addition, when the IoU threshold is 0.9, the detection precision was significantly higher than that of other baselines because the detection frame generated by InsulatorGAN was the smallest external polygon, which is a natural advantage of our model.

Nr. 1.1		InsuGenSet	:		CPLID	
Model	AP ₅₀	AP ₇₀	AP ₉₀	AP ₅₀	AP ₇₀	AP ₉₀
CenterNet	85.38	78.34	70.53	85.74	77.90	67.92
YOLOv4	89.14	81.58	74.97	87.83	80.46	71.71
Cascade R-CNN	92.26	89.52	80.34	91.63	88.25	78.94
InsulatorGAN	96.44	93.58	90.78	95.81	92.07	89.40

Table 4. APs of the different models.

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

Based on the work in [45], we used the structural similarity (SSIM), peak signal-tonoise ratio (PSNR), and sharpness difference (SD) to measure the pixel-level similarity between the generated image and the Ground Truth.

Structural Similarity (SSIM) evaluates the similarity between images based on attributes such as the brightness and contrast of the image, with a value range of [-1, 1]. When the value of SSIM is larger, the similarity of the input image pair is higher. The formula is shown below:

$$SSIM(I_g, I_t) = \frac{(2\mu_{I_g}\mu_{I_t} + c_1)(2\sigma_{I_g}I_t + c_2)}{(\mu_{I_g}^2 + \mu_{I_t}^2 + c_1)(\sigma_{I_g}^2 + \sigma_{I_t}^2 + c_2)}$$
(14)

where μ_{I_g} , μ_{I_t} represent the mean value of the generated image I_g and ground truth I_t ; σ_{I_g} , σ_{I_t} represent the standard deviation of I_g and I_t , respectively; and c_1 , c_2 represent a constant introduced to ensure that the denominator is not 0.

The peak signal-to-noise ratio (PSNR) evaluates the quality of the generated image relative to the ground truth by measuring the peak signal that reaches the noise ratio. Here, a high value indicates a better image quality. The formula is as follows:

$$PSNR(I_g, I_t) = 10\log_{10}\left(\frac{\max_{I_g}^2}{mse}\right)$$
(15)

$$mse(I_g, I_t) = \frac{1}{n} \sum_{i=1}^{n} (I_t[i] - I_g[i])^2, \max_{I_g} = 255$$
 (16)

The sharpness difference can be obtained during image generation by calculating the loss of sharpness. In this paper, we used the concepts outlined in [45] to calculate the gradient change between the fake image and the ground truth:

SharpDiff.
$$(I_g, I_t) = 10 \log_{10} \left(\frac{\max_{I_g}^2}{grads} \right)$$
 (17)

$$grads = \frac{1}{N} \sum_{i} \sum_{j} \left(\left| \left(\nabla_{i} I_{v} + \nabla_{j} I_{v} \right) - \left(\nabla_{i} I_{g} + \nabla_{j} I_{g} \right) \right| \right)$$
(18)

where $\nabla_i I = |I_{i,j} - I_{i-1,j}|, \nabla_j I = |I_{i,j} - I_{i,j-1}|$ and the Sharpness Difference in Formula (17) can be regarded as the reciprocal of the gradient.

As shown in Table 6, the score of InsulatorGAN was higher than the scores of all baselines. The results indicate that InsulatorGAN can learn to generate high-quality insulator-detection images for complex environments such as mountains, forests, and towns. The proposed penalty mechanism can further improve the semantic details of the image, make the generated image more realistic, and significantly reduce forgery traces in the image. In addition, the similarity values for the fake image and the ground truth were high, indicating that the generated results can be used as samples to expand the insulator dataset.

To compare the speed of all models, we conducted a comparative speed test. As shown in Table 5, the test speed of the InsulatorGAN proposed in this paper was lower than that of all the baselines. This result was due to the two-stage generation method adopted by the model from coarse to fine, which inevitably increased the quantity of calculations. The Monte Carlo search also required significant time. Nevertheless, the speed difference between InsulatorGAN and SelectionGAN was relatively small, and 64FPS was enough to meet the needs of practical applications.

Table 5. SSIM, PSNR, and SD of the different models. FPS is the number of images processed per second during testing.

M. 1.1	I	insuGenSet			CPLID		FDC
Model	SSIM	PSNR	SD	SSIM	PSNR	SD	FPS
Pix2Pix	0.30	15.89	17.52	0.28	15.63	17.37	162
CRN	0.28	15.62	17.15	0.25	15.48	16.84	188
X-Fork	0.39	16.26	18.27	0.36	16.04	17.94	86
X-Seq	0.47	17.23	18.69	0.45	16.94	18.36	74
SelectionGAN	0.64	26.72	20.72	0.62	26.14	20.29	68
InsulatorGAN	0.71	28.25	22.15	0.68	27.51	21.50	64

5.4. Qualitative Evaluation

The qualitative experimental results are shown in Figures 5 and 6. The resolution of the test images was 512 * 512. It can be seen that the image generated by InsulatorGAN in this paper was clearer, and the details of the object or scene were richer. When using the InsuGenSet and CPLID datasets, previous models are prone to generate fuzzy and distorted images. InsulatorGAN instead learned how to generate images of insulators with a strong diversity, producing more semantic details at the edges and connections of insulators. The wires, trees, and towers in the non-insulator area were observed to be realistic and similar to those of the Ground Truth.



Figure 5. Test examples of each model on the InsuGenSet dataset.



Figure 6. Test examples of each model on the CPLID dataset.

5.5. Sensitivity Analysis

In this section, we conducted five experiments on sensitivity analysis, including the two-stage generation method, the number of introductions of Monte Carlo search, the multi-level discriminator, the number of training iterations, and the minimum training data experimental.

5.5.1. Two-Stage Generation

In order to verify the impact of two-stage generation on model performance, we conducted experiments on the number of stages of InsulatorGAN introduced on the InsuGenSet dataset. The experimental results are shown in Table 6. It can be seen from the table that when using two-stage generation, InsulatorGAN had the most balanced performance on various indicators.

Table 6. Comparison of the effectiveness of the generator networks.

	IS	FID	FPS
Single generator	3.49	56.95	82
Two-stage generator	4.05	55.23	64
Three-stage generator	4.21	55.02	37

5.5.2. Monte Carlo Search

In order to verify the influence of the number of times the Monte Carlo search was introduced on the performance of the model, we conducted experiments on the number of introductions of InsulatorGAN's Monte Carlo search on the InsuGenSet dataset. The experimental results are shown in Table 7. From the experimental results, it can be seen that when the number of times of using Monte Carlo search was 5, the performance of InsulatorGAN on various indicators was the most balanced.

MCS	SSIM	PSNR	SD	FPS
Not introduced	0.58	26.39	19.30	75
N = 1	0.64	26.95	20.46	70
N = 3	0.69	27.71	21.37	67
N = 5	0.71	28.25	22.15	64
N = 7	0.73	28.58	22.54	58
N = 9	0.74	28.73	23.02	51

 Table 7. Introducing the Monte Carlo search time comparison.

5.5.3. Multi-Level Discriminator

To verify the impact of the multi-level discriminator on the performance of the model, we conducted experiments on the number of times the discriminator of InsulatorGAN was introduced on the InsuGenSet dataset. The experimental results are shown in Table 8. It can be seen from the experimental results that when the three-level discriminator was used, the performance of InsulatorGAN on various indicators was the most balanced.

Table 8. Comparison of the effectiveness of the discriminant networks.

	SSIM	PSNR	SD	FPS
Single discriminator	0.61	26.59	19.82	72
Two-level discriminator	0.66	27.38	20.74	68
Three-level discriminator	0.71	28.25	22.15	64
Four-level discriminator	0.73	28.61	22.88	59

5.5.4. Number of Epochs

To verify the impact of the number of epochs on the performance of the model, we conducted experiments on the number of iterations of InsulatorGAN on the InsuGenSet dataset. The experimental results are shown in Table 9. From the experimental results, it can be seen that when the epoch was 200, InsulatorGAN had the most average performance on various indicators.

Table 9. The effect of different epoch numbers on the experimental results.

Number of Epochs	SSIM	PSNR	SD	FPS
50	0.33	16.73	17.92	65
100	0.59	18.58	19.06	63
150	0.67	24.56	21.30	63
200	0.71	28.25	22.15	64
250	0.69	27.94	21.85	61

5.5.5. Minimum Training Data Experiment

In order to verify the impact of the training set size on the model's performance, we conducted experiments on the minimum training sample size of InsulatorGAN on the InsuGenSet dataset. The experimental results are shown in Table 10. It can be seen from the results that the performance of InsulatorGAN did not drop significantly with the reduction of the training set at the beginning. Until the training set size is reduced to 70%, InsulatorGAN's scores on various indicators are similar to SelectionGAN, which shows that InsulatorGAN has strong robustness and can still learn key feature information even on a small-scale dataset. To a certain extent, it overcomes the shortcomings of the previous model's poor generalization ability.

The Amount of Training Set	SSIM	PSNR	SD
2000 (100%)	0.71	28.25	22.15
1800 (90%)	0.69	27.93	21.85
1600 (80%)	0.66	25.97	20.36
1400 (70%)	0.63	25.04	19.82
1200 (60%)	0.57	23.92	17.31

Table 10. Minimum training data experimental results.

5.6. Ablation Analysis

To analyze the functions of the different components in InsulatorGAN, we performed an ablative analysis on InsuGenSet. As shown in Table 11, model B had better indicators than model A, which demonstrates that the two-stage generation method from coarse to fine was able to better improve the clarity of the image. Model C promoted the learning of the generator by introducing a multi-level discriminator, which enhanced the quality of the generated image while improving stability. The score of Model D shows that the penalty mechanism significantly improved the performance of InsulatorGAN, allowing the model to obtain sufficient semantic constraints during the generation process and increasing the clarity and realism of the generated image.

Table 11. The results of the ablation analysis.

Model	Architecture	SSIM	PSNR	SD
А	Pix2Pix	0.30	15.89	17.52
В	A + two-stage generation	0.52	21.35	19.21
С	B + multi-level discriminator	0.61	24.82	20.80
D	C + Monte Carlo search	0.71	28.25	22.15

5.7. Computational Complexity

The network parameters and training time were counted to evaluate the space complexity and time complexity of the networks. The results are shown in Table 12. It can be seen that, compared with Cascade R-CNN, InsulatorGAN had a smaller number of network parameters, but it obtained a better test performance. Compared with generative models such as SelectionGAN, although the parameter amount and training time of InsulatorGAN increased slightly, the performance improvement it brings was worthwhile.

Table 12. Network parameters (Param.) and training time of the different models.

Model	Param.	Training Time (h)
CenterNet	14 M	4.05
YOLOv4	28 M	6.38
Cascade R-CNN	184 M	49.84
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
InsulatorGAN	82 M	22.73

6. Conclusions

This paper proposed a power-line-insulator-detection image-generation model called InsulatorGAN, which can generate insulator-detection images based on aerial images taken by drones. This model first uses a coarse-grained module to generate low-resolution target images and then uses a Monte Carlo search to mine the hidden information of the intermediate results. Finally, this method uses a fine-grained model to synthesize high-resolution target images based on the search results. Quantitative and qualitative experiments on the public CPLID dataset and the self-built InsuGenSet dataset provided the following results. Compared with current mainstream models, the model in this paper can generate clearer and more diversified insulator-detection images. The ablative analysis experiment demonstrated that the penalty mechanism proposed in this paper based on a Monte Carlo search significantly improved the image quality. In addition, the model in this paper can be flexibly transferred to multiple scenarios. In the future, we will further explore the application of this model to other power-line components, image-style transfer, image translation, and other fields.

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.

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. Tao, X.; Zhang, D.; Wang, Z.; Liu, X.; Zhang, H.; Xu, D. Detection of Power Line Insulator Defects Using Aerial Images Analyzed with Convolutional Neural Networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *50*, 1486–1498. [CrossRef]
- 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]
- 3. 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]
- 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]
- 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]
- 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]
- Coluccia, A.; Fascista, A.; Schumann, A.; Sommer, L.; Dimou, A.; Zarpalas, D.; Méndez, M.; de la Iglesia, 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] [PubMed]
- 8. Zhi, Y.; Fu, Z.; Sun, X.; Yu, J. Security and Privacy Issues of UAV: A Survey. Mob. Netw. Appl. 2020, 25, 95. [CrossRef]
- 9. Zhang, Y.; Yuan, X.; Li, W.; Chen, S. Automatic Power Line Inspection Using UAV Images. Remote Sens. 2017, 9, 824. [CrossRef]
- 10. Yin, J.; Lu, Y.; Gong, Z.; Jiang, Y.; Yao, J. Edge Detection of High-Voltage Porcelain Insulators in Infrared Image Using Dual Parity Morphological Gradients. *IEEE Access* 2019, *7*, 32728–32734. [CrossRef]
- Iruansi, U.; Tapamo, J.R.; Davidson, I.E. An active contour approach to water droplets segmentation from insulators. In Proceedings of the 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, 14–17 March 2016; pp. 737–741. [CrossRef]
- 12. Sadykova, D.; Pernebayeva, D.; Bagheri, M.; James, A. IN-YOLO: Real-Time Detection of Outdoor High Voltage Insulators Using UAV Imaging. *IEEE Trans. Power Deliv.* 2020, *35*, 1599–1601. [CrossRef]
- 13. 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*, 1204. [CrossRef]
- 14. Mirza, M.; Osindero, S. Conditional generative adversarial nets. arXiv 2014, arXiv:1411.1784.

- 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]
- 16. 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]
- 17. 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]
- 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]
- 19. Browne, C.B.; Powley, E.; Whitehouse, D.; Lucas, S.M.; Cowling, P.I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S. A Survey of Monte Carlo Tree Search Methods. *IEEE Trans. Comput. Intell. Ai Games* **2012**, *4*, 1–43. [CrossRef]
- Srivastava, A.; Valkov, L.; Russell, C.; Gutmannet, M.U.; Sutton, C.A. VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 3308–3318. [CrossRef]
- 21. Zhang, Y.; Yang, Q. A Survey on Multi-Task Learning. IEEE Trans. Knowl. Data Eng. 2021, 29, 2367–2381. [CrossRef]
- 22. 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]
- Wen, Q.; Luo, Z.; Chen, R.; Yang, Y.; Li, G. Deep Learning Approaches on Defect Detection in High Resolution Aerial Images of Insulators. Sensors 2021, 21, 1033. [CrossRef]
- 24. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. ICLR. arXiv 2014, arXiv:1312.6114.
- Yan, X.; Yang, J.; Sohn, K.; Lee, H. Attribute2Image: Conditional Image Generation from Visual Attributes. In *Computer Vision–* ECCV 2016. ECCV 2016. Lecture Notes in Computer Science; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Amsterdam, The Netherlands, 2016; Volume 9908. [CrossRef]
- 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).
- 27. 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]
- 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]
- 29. Dumoulin, V.; Belghazi, I.; Poole, B.; Mastropietro, O.; Lamb, A.; Arjovsky, M.; Courville, A. Adversarially Learned Inference. *arXiv* **2016**, arXiv:1606.00704.
- 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 (NIPS'16), Barcelona, Spain, 5–10 December 2016; pp. 2180–2188.
- Chiaroni, F.; Rahal, M.; Hueber, N.; Dufaux, F. Hallucinating A Cleanly Labeled Augmented Dataset from A Noisy Labeled Dataset Using GAN. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 3616–3620. [CrossRef]
- 32. 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] [PubMed]
- Johnson, J.; Gupta, A.; L, 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]
- 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]
- 35. Dosovitskiy, A.; Springenberg, J.T.; Tatarchenko, M.; Brox, T. Learning to Generate Chairs, Tables and Cars with Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 692–705. [CrossRef] [PubMed]
- 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]
- Ronneberger, O. Invited Talk: U-Net Convolutional Networks for Biomedical Image Segmentation. In *Bildverarbeitung für die* Medizin 2017, Informatik aktuell; Maier-Hein, G., Fritzschej, K., Deserno, G., Lehmann, T., Handels, H., Tolxdorff, T., Eds.; Springer Vieweg: Berlin/Heidelberg, Germany, 2017; Volume 12. [CrossRef]
- 38. Ulyanov, D.; Vedaldi, A.; Lempitsky, V. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv* 2016, arXiv:1607.08022.
- 39. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. Comput. Sci. 2015, 14, 38–39. [CrossRef]
- 40. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162. [CrossRef]

- 41. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.
- 42. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as Points. arXiv 2016, arXiv:1904.07850.
- 43. 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]
- Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In European Conference on Computer Vision-ECCV 2014, Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8693. [CrossRef]
- 45. Mathieu, M.; Couprie, C.; LeCun, Y. Deep multi-scale video prediction beyond mean square error. arXiv 2016, arXiv:1511.05440.