Research on Insulator Defect Detection Based on an Improved MobilenetV1-YOLOv4

Shanyong Xu 1, Jicheng Deng 1,*, Yourui Huang 1,2, Liuyi Ling 1 and Tao Han 1

1 School of Electrical & Information Engineering, Anhui University of Science and Technology, Huainan 232001, China
2 School of Electrical and Opto Electronic Engineering, West Anhui University, Lu’an 237012, China
* Correspondence: 2021200770@aust.edu.cn

Abstract: Insulator devices are important for transmission lines, and defects such as insulator bursting and string loss affect the safety of transmission lines. In this study, we aim to investigate the problems of slow detection speed and low efficiency of traditional insulator defect detection algorithms, and to improve the accuracy of insulator fault identification and the convenience of daily work; therefore, we propose an insulator defect detection algorithm based on an improved MobilenetV1-YOLOv4. First, the backbone feature extraction network of YOLOv4 ‘Backbone’ is replaced with the lightweight module Mobilenet-V1. Second, the scSE attention mechanism is introduced in stages of preliminary feature extraction and enhanced feature extraction, sequentially. Finally, the depthwise separable convolution substitutes the $3 \times 3$ convolution of the enhanced feature extraction network to reduce the overall number of network parameters. The experimental results show that the weight of the improved algorithm is 57.9 MB, which is 62.6% less than that obtained by the MobilenetV1-YOLOv4 model; the average accuracy of insulator defect detection is improved by 0.26% and reaches 98.81%; and the detection speed reaches 190 frames per second with an increase of 37 frames per second.

Keywords: insulator defect detection; YOLOv4; Mobilenet-V1; scSE attention mechanism; depthwise separable convolution

1. Introduction

Insulators are unique insulating devices that are used on overhead transmission lines to protect the power transmission lines. There are mainly three types of insulators: ceramic, glass, and composite. Due to long-term exposure to the outdoors, short-circuit faults between lines are likely to occur when there are insulator problems such as bursting, string loss, and contamination, which result in significant harm to the normal power supply and service life of transmission lines. Therefore, insulator defect detection is important to ensure normal operation of power lines.

With the development of artificial intelligence and machine learning technology, many scholars have conducted research on target algorithms for insulator defect detection that can be divided into two main categories: One category is the two-stage target detection algorithm based on candidate regions and CNN extraction; the other category is the one-stage target detection algorithm that directly completes feature extraction, classification, and regression prediction [1]. The first category includes the RCNN [2], Fast RCNN [3], and Faster RCNN [4] two-stage target detection algorithms; the second category includes YOLO (You Only Look Once) [5], SSD (Single Shot Multibox Detector) [6], YOLOv2 [7], YOLOv3 [8], YOLOv4 [9], etc. The authors of [10] proposed an improved YOLOv5 insulator breakage detection algorithm; this algorithm improved the accuracy of the algorithm by adding the attention mechanism ECA-NET (efficient channel attention) and by using the Soft NMS algorithm to reallocate the original candidate frames. The detection effect of overlapping targets was strengthened, which could meet the engineering application. In a study by [11], an insulator defect detection method was proposed that adopted an RPN
network for feature extraction to realize the detection of tiny defects on insulators. The advantage of this method was that it had high detection accuracy. However, it had low detection speed, i.e., only 12.8 frames per second and it could not be deployed on the mobile end. A study by [12] adopted ResNet as a backbone network, and equipped it with a batch of normalized convolutional attention modules (BN-CBAM) that could better utilize channel information and could enhance the ability of different channels to feature maps. In this case, this model had a large number of parameters. Mobile deployment requires more computing resources and memory. The authors of [13] proposed a defect detection algorithm deployed on FPGA. The algorithm had high efficiency, power loss was only 10 W, and hardware deployment was realized. Ref. [14] proposed an insulator fault detection algorithm based on a convolutional neural network, which, first, cascaded the detection network and the fault classification network to realize the position detection and fault classification of insulators, then detected faulty insulator(s) in images with complex background and high resolution, and finally, realized the fault detection. Unfortunately, there were few datasets and the identified faults were limited. Ref. [15] proposed a YOLOv5 insulator defect detection method; although it realized fast detection, this method sacrificed accuracy for speed, with only 81% accuracy, and therefore, could not be used in practical applications. In [16], the authors proposed an insulator detection method based on YOLOv2 that could complete the detection of 25 images per second, however, with an accuracy of only 88%, and therefore, may have missed detection. Ref. [17] proposed an insulator defect detection method based on YOLOv5s and designed a new residual module to reduce network parameters and to extract more useful features. The method had high detection speed but at the cost of network accuracy. Ref. [18] proposed an improved YOLOv3-based insulator detection with a new feature pyramid network, which had high detection accuracy for insulator defects. However, the network could not learn independently. Ref. [19] proposed an aerial image insulator detection based on an improved YOLOv3, which utilized multi-level feature mapping modules in the network. This model had better detection accuracy, but increased the complexity of the network. Ref. [20] proposed an insulator fault detection model based on an improved YOLOv3 with the utilization of an SPP network and a multi-scale prediction network. The performance of this method for insulator detection was relatively good, but the weight of the improved network model was up to 255 MB with significant use of memory. Ref. [21] proposed an insulator fault detection model based on an improved Faster RCNN that adopted a feature pyramid network to complete the insulator image location under a complex background and insulator identification. However, this method had high requirements for hardware. Although the detection accuracies of the proposals by [19–21] were improved, they were difficult to deploy to actual models and practical applications. In [22], an image texture segmentation algorithm was proposed for aviation insulators. However, the recognition effects were unsatisfactory as the texture features of the insulators were very similar to leaves, though different from most backgrounds. Ref. [23] proposed a multi-scale feature insulator detection algorithm that, first, extracted the local features of insulators, and then trained local features to obtain spatial order features to improve the robustness of the algorithm. However, this algorithm was highly complex, time-consuming, and incapable of satisfying real-time applications. Ref. [24] proposed a multi-scale residual neural network-based insulator surface damage identification method that could effectively detect insulators in single backgrounds while ignoring the identification of insulators in complex backgrounds. Ref. [25] proposed an improved U-Net insulator image segmentation method based on attention mechanism, which embedded the ECA-Net attention mechanism in the encoding stage of U-Net to improve the semantic feature extraction ability of the model and further the accuracy of the insulator image segmentation. Ref. [26] proposed an insulator self-explosion detection method based on YOLOv4 that, first, fused the shallow feature map into the feature pyramid, and then adopted the SENet structure to improve the recognition accuracy of network. This method had high accuracy and a slight decrease in detection speed. Ref. [27] proposed a new insulator defect detection algorithm using deep learning and
morphological detection that adopted a residual network to extract the morphological features of insulators, and then the image segmentation pixel clustering method to establish a mathematical model of insulator defects. The detection accuracy of this method was good, but the network parameters were more complicated. Ref. [28] proposed an insulator string detection method based on an improved YOLOv5 that adopted an EIOU loss function and the AFK-MC2 anchor point generation method to detect insulators. Ref. [29] proposed an insulator defect detection method based on an improved lightweight YOLOv4, reaching a detection accuracy of 93.81% and a detection speed of 53 frames per second. Ref. [30] proposed an improved insulator detection algorithm based on YOLOvX that adopted an improved Siou loss function to speed up model convergence, and was embedded with the ECA attention mechanism. The research results showed that the detection accuracy of this method reached 97.18%, and the detection speed was 71 frames per second. Although both two-stage and one-stage target detection algorithms have achieved great success in insulator recognition, due to the limitations of storage space and power consumption, many scholars have begun to study a lightweight model with reduced complexity and improved detection speed that can be deployed on tiny devices. A traditional CNN has high memory and computational demands, which makes it unable to run on mobile and embedded devices. MobileNet [31–33] was proposed by the Google team, in 2017, with a focus on lightweight CNN networks in mobile terminals or embedded devices. It used depth convolution in the network to reduce the amount of computations and parameters. Using the reciprocal residual structure in the network could reduce the memory consumption during reasoning. Wang [34] proposed a CSPNet lightweight network, in 2019, that integrated gradient changes into a feature map from beginning to end, and thus, reduced the amount of computations and ensured accuracy. This method reduced the amount of calculation and improved the running speed of the model without reducing the accuracy of the model and could be used for mobile deployment. In 2020, Huawei proposed a new lightweight network, i.e., GhostNet [35]. We knew that the redundancy of the feature map was very important, therefore, we designed a Ghost module that realized the operation with fewer computations to generate the redundant feature maps, and deployed the network to a mobile terminal. In order to make the algorithm model smaller, we applied it to insulator detection.

To further improve insulator detection accuracy and speed, to reduce algorithm complexity, and to make the algorithm easy to be deployed on hardware equipment (such as drones), in this paper, we propose a lightweight insulator defect detection algorithm based on an improved MobileNetV1-YOLOv4 that reconstructs the backbone feature extraction network of YOLOv4 through the lightweight module MobileNet-V1, introduces the scSE attention mechanism to enhance feature extraction, and adopts depthwise separable convolution to reduce the overall number of parameters in the network. The effectiveness of this proposed algorithm was verified through simulation and experimental analysis.

2. Fundamentals of MobileNetV1-YOLOv4

YOLOv4 is a high-accuracy, one-stage target detection algorithm that has been developed from YOLOv3. In 2020, Bochkovskiy et al. published the YOLOv4 model and explained it. The MobileNetV1-YOLOv4 algorithm is a lightweight network model obtained from YOLOv4. It consists of the following three parts: backbone feature extraction network, feature pyramid network (PANet, SPP), and classification regression layer (YOLO Head). Figure 1 shows the network structure of MobileNetV1-YOLOv4.
Figure 1. The network structure of MobilenetV1-YOLOv4.

2.1. Backbone Feature Extraction Network

The backbone feature network of YOLOv4 is CSPDarkNet53. It adopts the CSPnet structure to split the original residual block into two parts: the main body and the branch. The main body is still a residual block, and the branch is a large residual edge that is connected to the main body through a small amount of processing. The benefit of this is that gradients do not disappear as the network depth complexity increases. To reduce network complexity, CSPDarkNet53 is replaced by MobilenetV1. The network structure adopts depthwise separable convolution that decomposes standard convolution into two steps, i.e., depthwise convolution and point-by-point convolution. This can significantly reduce the size of the algorithmic model. Finally, to avoid information loss of image features due to multiple convolutions, output is still in three dimensions, i.e., $13 \times 13 \times 1024$, $26 \times 26 \times 512$, and $52 \times 52 \times 256$.

2.2. Feature Pyramid Network

A feature pyramid network enhances feature extraction for three dimensions of the initial feature network output, which can be divided into two main parts: SPP and PANet. The result obtained by the convolution of the last feature layer of the MobilenetV1 network is subjected to SPP max pooling, in which there are four different compositions of convolutional kernels; the pooling kernel sizes for max pooling are $13 \times 13$, $9 \times 9$, $5 \times 5$, and $1 \times 1$, respectively. Finally, the splicing of the quantity of channels is realized and sent to the PANet network. The PANet structure is an instance segmentation algorithm that can be used for both top-to-bottom and bottom-to-top feature extractions. The size of the feature map is changed through upsampling and downsampling to realize dimension splicing and complete repeated feature extraction.
2.3. Classification and Regression Layer

In the PANet network, $3 \times 3$ convolution is first used to convolve the result of the feature pyramid, and then the $1 \times 1$ convolution, and finally three prediction results of different sizes are obtained. However, the positions of the prediction results and the final prediction frame do not correspond, and therefore, decoding is demanded to finally acquire the prediction information.

3. Improved MobilenetV1-YOLOv4 Algorithm

In this paper, the YOLOv4 algorithm is adopted as the basic network. The original CSPdarknet53 network is replaced with a lighter network Mobilenet-V1 to make the detection speed of the model faster. The $3 \times 3$ ordinary convolutions in the PANet network and SPP are replaced by depthwise separable convolution to further reduce the number of network parameters. To maintain network accuracy, the scSE attention mechanism is added to the three dimensions of the preliminary feature extraction, and also to the results of upsampling in the enhanced feature extraction network. Finally, the convolution layer after SPP is modified to five layers. The improved network structure is shown in Figure 2.

![Figure 2. Improved MobilenetV1-YOLOv4 network.](image)

As shown in Figure 2, the convolution size in the CBR module is $1 \times 1$, and later, normalization is used to avoid the disappearance of gradient and to speed up the convergence; the activation function can effectively increase the nonlinearity of the network and make it learn better; the DBRCBR module contains two types of convolutions, i.e., depthwise convolution and point convolution, to form depthwise separable convolution. Therefore, the traditional $3 \times 3$ convolution module is replaced with this depthwise separable convolution. Conv2D in the DBRCBR module is point convolution, and it is actually a $1 \times 1$ convolution. Its function is to freely change the number of output channels. Next, it performs channel fusion on the feature map of depth convolution output. After the convolution operations of depth convolution and point convolution, this is the operation process of depth separable...
convolution. The scSE attention mechanism can enhance the feature information that is needed and can suppress the feature information that is not needed.

3.1. scSE Attention Mechanism

The spatial and channel squeeze and channel excitation (scSE) attention mechanism module is a combination of two modules, spatial squeeze and channel excitation (cSE) and channel squeeze and spatial excitation (sSE).

The cSE attention module is shown in Figure 3. First, global pooling is performed on the feature map to compress the feature image information. It can generate a Z vector (dimension is $1 \times 1 \times C$) whose dimension changes from $[C, H, W]$ to $[C, 1, 1]$, and compresses the height and width of the feature image to $1 \times 1$. Then, two fully connected layers are used to process feature information; the fully connected neuron quantity is less than the input feature layer the first time, while it is equal the second time. After two times of full connection, the $C$-dimension vector is obtained, and the Sigmoid activation function is used for normalization to fix its value between zero and one. At this time, the weight of each channel of the input feature layer is known. Finally, multiply the obtained weights with the original feature map to obtain the calibrated results, which improves the ability of the network to extract channel features.

![Diagram](image)

**Figure 3.** Spatial squeeze and channel excitation (cSE).

Figure 4 displays the sSE attention mechanism module whose role is to squeeze the feature map along the channel and spatially motivate. First, operate on the input feature map. The channel compression is performed using a $1 \times 1 \times 1$ convolutional method, which changes the feature map from the initial $[C, W, H]$ to $[1, H, W]$. Then, a new spatial feature map is obtained by the Sigmoid activation function method, and finally the obtained new spatial feature map is multiplied with the original feature map to achieve spatial information calibration. This operation has a significant effect on the spatial position of related features, thereby, omitting irrelevant features and improving the learning ability of spatial feature information.

The scSE attention mechanism module adds the new features obtained by both cSE and sSE attention modules to complete the stacking of information. The scSE attention mechanism can enhance the channel and spatial information, and can improve the detection accuracy of the whole network, as shown in Figure 5.
After depthwise and point convolutions, we can finally acquire new features. In depthwise convolution, the amount of these new feature maps is consistent with that of the convolutional kernels. After depth convolution is consistent with the number of channels in the input layer, so the convolution kernel is responsible for calculating one channel. The quantity of convolution kernels is similar to that of the channels in the previous-layer feature map. Through this operation, three characteristic graphs are obtained. However, the number of feature maps after depth convolution is consistent with the number of channels in the input layer, so more and more effective feature maps cannot be obtained. Meanwhile, this operation also requires a separate convolutional computation for each channel of the input layer, which cannot effectively utilize the feature information of different channels at the same spatial position. Thus, point convolution is adopted to convolve these feature maps again to generate new feature maps. The computation mode of point convolution is very similar to that of conventional convolution. The size of the convolution kernel of point convolution is 1 × 1 × C, where C represents the quantity of channels after depthwise convolution. Each convolutional kernel has three channels. Then, multiply and accumulate the convolutional kernel with the three channels of the input image to obtain new feature maps. The amount of these new feature maps is consistent with that of the convolutional kernels. After the above depthwise and point convolutions, we can finally acquire new features. In addition, the number of network parameters is significantly reduced, and the detection speed is quickened.

3.2. Depth Separable Convolutional Module

Depthwise separable convolution is a combination of depthwise convolution and point convolution. It can reduce the number of parameters calculated, and can reduce the model size. The process of depthwise separable convolution is shown in Figure 6. First, the first convolutional operation is performed on the input RGB image, and each convolutional kernel is responsible for calculating one channel. The quantity of convolution kernels is similar to that of the channels in the previous-layer feature map. Through this operation, three characteristic graphs are obtained. However, the number of feature maps after depth convolution is consistent with the number of channels in the input layer, so more and more effective feature maps cannot be obtained. Meanwhile, this operation also requires a separate convolutional computation for each channel of the input layer, which cannot effectively utilize the feature information of different channels at the same spatial position. Thus, point convolution is adopted to convolve these feature maps again to generate new feature maps. The computation mode of point convolution is very similar to that of conventional convolution. The size of the convolution kernel of point convolution is 1 × 1 × C, where C represents the quantity of channels after depthwise convolution. Each convolutional kernel has three channels. Then, multiply and accumulate the convolutional kernel with the three channels of the input image to obtain new feature maps. The amount of these new feature maps is consistent with that of the convolutional kernels. After the above depthwise and point convolutions, we can finally acquire new features. In addition, the number of network parameters is significantly reduced, and the detection speed is quickened.

Figure 4. Channel squeeze and spatial excitation (sSE).

Figure 5. Concurrent spatial and channel squeeze and channel excitation (scSE).
the number of network parameters is significantly reduced, and the detection speed is quickened.

Channel Input $\times$ Filters$\times$ Maps$\times$ Filters$\times$ Maps$\times$ 4 ($a$) Depthwise convolution

The First Process of Depth Separable Convolution

Maps$\times$ 3 Filters$\times$ 3 Maps$\times$ 3 ($b$) Point convolution

The Second Process of Depth Separable Convolution

Figure 6. Depthwise separable convolution.

For ordinary convolution, the calculation formula of its parameter quantity is Formula (1) as follows:

$$N_{pc} = D_F \times D_F \times C_i \times C_o$$  \hspace{1cm} (1)

where $D_F$ refers to the dimension of convolutional kernel, $C_i$ refers to the quantity of input channels, $C_o$ refers to the dimension of output, and $N_{pc}$ refers to the quantity of parameters.

The formula of network computation amount is Formula (2) as follows:

$$N_{cc} = H_o \times W_o \times C_o \times D_F \times D_F \times C_i$$  \hspace{1cm} (2)

where $H_o$ and $W_o$ stand for the height and width of the input feature map, respectively, and $N_{cc}$ stands for the network computation amount.

For depthwise separable convolution, its parameter quantity is calculated by adding the total parameters obtained by depthwise convolution and point convolution. Formula (3) can be used to calculate the parameter quantity of depthwise convolution as follows:

$$N_{p1} = D_F \times D_F \times C_i$$  \hspace{1cm} (3)

Formula (4) can be used to calculate its network computation amount as follows:

$$N_{c1} = H_o \times W_o \times C_i \times D_F \times D_F$$  \hspace{1cm} (4)

Formula (5) can be used to calculate the parameter quantity of point convolution as follows:

$$N_{p2} = C_i \times C_o$$  \hspace{1cm} (5)

Formula (6) can be used to calculate its network computation amount as follows:

$$N_{c2} = H_o \times W_o \times C_o \times 1 \times 1 \times C_i$$  \hspace{1cm} (6)

Thus, the total parameters for depthwise separable convolution can be obtained by Formula (7):

$$N_{pd} = N_{p1} + N_{p2}$$  \hspace{1cm} (7)
The total network computation amount can be obtained by Formula (8):

$$N_{cd} = N_{c1} + N_{c2}$$  \hspace{1cm} (8)

Further, the ratio of the parameter quantity of depthwise separable convolution to that of ordinary convolution can be obtained as per Formula (9), from which we can know the number of parameters under depthwise separable convolution is \( \frac{1}{C_o} + \frac{1}{D_F^2} \) of the original one:

$$\frac{N_{pd}}{N_{pc}} = \frac{D_F \times D_F \times C_i + C_i \times C_o}{D_F \times D_F \times C_i \times C_o} = \frac{1}{C_o} + \frac{1}{D_F^2}$$  \hspace{1cm} (9)

In the improved MobilenetV1-YOLOv4 algorithm network, the ordinary \( 3 \times 3 \) convolutions in the PANet network and the SPP network are replaced by the depthwise separable convolution, and the weight after training is 57.9 MB. As compared with the weight of 244 MB trained by the YOLOv4 network, the size of the weight is reduced by 76.3%.

3.3. Mobilenet-V1 Network

In this paper, the original backbone network CSPdarknet53 is replaced by the Mobilenet-V1 lightweight network. The Mobilenet model is a lightweight network proposed by Google for devices with low computing power such as embedded devices. Its core operation is to form the main network by using depthwise separable convolutional blocks. Three preliminary effective features can be obtained through the Mobilenet-V1 network, which are \( 52 \times 52 \times 256, 26 \times 26 \times 512, \) and \( 13 \times 13 \times 1024 \). Table 1 demonstrates the Mobilenet-V1 structure, in which the ordinary convolution is represented by Conv, the depthwise separable convolution is represented by Conv dw, and the step size in the convolutional process is represented by S. Three layer structures of the original Mobilenet-V1 are removed, namely global average pooling, fully connected layer, and Softmax.

| MobileNet-V1 Body | Type/Stide      | Filter Shape                  | Output Size         |
|-------------------|----------------|------------------------------|---------------------|
| Input             |                |                              | 416 \times 416 \times 3 |
| Conv/s2           |                | \( 3 \times 3 \times 3 \times 32 \) | 208 \times 208 \times 32 |
| Conv dw/s1        |                | \( 3 \times 3 \times 32 \) dw | 208 \times 208 \times 32 |
| Conv/s1           |                | \( 1 \times 1 \times 32 \times 64 \) | 208 \times 208 \times 64 |
| Conv dw/s1        |                | \( 3 \times 3 \times 64 \) dw | 104 \times 104 \times 64 |
| Conv/s1           |                | \( 1 \times 1 \times 64 \times 128 \) | 104 \times 104 \times 128 |
| Conv dw/s1        |                | \( 3 \times 3 \times 128 \) dw | 104 \times 104 \times 128 |
| Conv/s1           |                | \( 1 \times 1 \times 128 \times 128 \) | 104 \times 104 \times 128 |
| Conv dw/s2        |                | \( 3 \times 3 \times 128 \) dw | 52 \times 52 \times 128 |
| Conv/s1           |                | \( 1 \times 1 \times 128 \times 256 \) | 52 \times 52 \times 256 |
| Conv dw/s1        |                | \( 3 \times 3 \times 256 \) dw | 52 \times 52 \times 256 |
| Conv/s1           |                | \( 1 \times 1 \times 256 \times 256 \) | 52 \times 52 \times 256 |
| Conv dw/s2        |                | \( 3 \times 3 \times 256 \) dw | 26 \times 26 \times 256 |
| Conv/s1           |                | \( 1 \times 1 \times 256 \times 512 \) | 26 \times 26 \times 512 |
| 5 \times Conv dw/s1, |                | \( 3 \times 3 \times 256 \) dw, | 26 \times 26 \times 512 |
| Conv/s1           |                | \( 1 \times 1 \times 512 \times 512 \) | 26 \times 26 \times 512 |
| Conv dw/s2        |                | \( 3 \times 3 \times 512 \) dw | 13 \times 13 \times 512 |
| Conv/s1           |                | \( 1 \times 1 \times 512 \times 1024 \) | 13 \times 13 \times 1024 |
| Conv dw/s1        |                | \( 3 \times 3 \times 1024 \) dw | 13 \times 13 \times 1024 |
| Conv/s1           |                | \( 1 \times 1 \times 1024 \times 1024 \) | 13 \times 13 \times 1024 |

4. Experimental Platform Construction and Training

4.1. Experimental Platform

This experiment is completed under Windows11 system with a computer configuration as follows: the 12th Gen Intel(R) Core(TM) i7-12700KF@3.61 GHz CPU, 32 GB memory,
NVIDIA GeForce RTX 3080 Ti GPU, 12GB video memory, software Anaconda3, version 3.7 Python, and Tensorflow2.5 deep learning framework.

4.2. Data Collection and Processing

The dataset of this paper comes from the Baidu open source dataset, including 4147 pictures of diversified and defective insulators, and various types of insulator defect data. In order to enable the network model to learn the characteristics and location information of insulators and to improve the accuracy of model output, insulator defects in the image can be marked. The .xml label files are formed by using the labeling software to mark insulator defects. The boundary box is minimized as far as possible during the image marking of insulator defects to reduce the influence of background. The annotation results are saved in PASCAL VOC format, and the generated XML format file is saved in the pre-created folder. The ratio of training set plus validation set to test set is 9:1, and the ratio of training set to validation set is 9:1. Some dataset images are shown in Figure 7.

![Figure 7. Several sample images.](image)

4.3. Model Training

A total of four groups of comparative experiments were performed in this study. The first group contains YOLOv4, MobilenetV1-YOLOv4, and improved MobilenetV1-YOLOv4; the second group is the comparison with other lightweight models, namely MobilenetV1-YOLOv4, MobilenetV2-YOLOv4, MobilenetV3-YOLOv4, and Ghostnet-YOLOv4; the third group is the comparison with improved networks, respectively, improved MobilenetV1-YOLOv4, improved MobilenetV2-YOLOv4, improved MobilenetV3-YOLOv4, and improved Ghostnet-YOLOv4. The fourth group of experiments was conducted in a deeply separable convolutional network by adding the scSE attention mechanism. It is compared with the current algorithm YOLOv5.

The idea of migration learning is applied in the training of the network, and the pretrained weights are obtained by using the training VOC dataset. During the training process, each network is trained for a total of 250 times. For the first 50 times of training, the backbone network of the model is frozen by the frozen training to improve the training efficiency and avoid the destruction of weights. In addition, the occupied video memory is also small in the process. The batch size is set to 32 during frozen training. Afterwards, unfrozen training is performed to unfreeze the backbone, which increases the occupation of the network video memory. At this time, the batch size is set to 16. By judging the current batch size, the learning rate of the model can be adaptively adjusted from the minimum 0.0001 to the maximum 0.01. Three different YOLO Heads are acquired by improving the training of the network, and finally the prediction frame is obtained by using the non-maximum suppression method. The essence of the non-maximum suppression method is to search local maxima and suppress nonmax elements. First, set the confidence threshold...
of the target box. The threshold set in this study is 0.5. Then, arrange the list of candidate boxes in descending order according to the confidence. In the identified target categories, select and retain the bounding box A with the highest confidence, and then calculate the IoU of bounding box A and the remaining boxes B. If the IoU value is greater than the threshold, remove B. Repeat this step until the iteration of a target class is completed and the required target box is finally output.

In the training process, the mosaic data enhancement method is used, and four pictures are randomly read at one time from the dataset for scaling or cropping to get a new picture after combination. The new image has a different background, which enriches the spatial semantic information and enhances the generalization ability of the model. By using the label smoothing training method, the data can be better calibrated and can be more accurately predicted. The cosine annealing algorithm is used to avoid the network falling into the local optimal solution. Formula (10) illustrates the principle of using the cosine annealing algorithm to prevent the network from falling into the local optimal solution:

$$\eta_t = \eta_{\text{min}} + \frac{1}{2}(\eta_{\text{max}} - \eta_{\text{min}}) \left(1 + \cos \left(\frac{T_{\text{cur}}}{T_i} \pi\right)\right)$$  \hspace{1cm} (10)

where $i$ represents the ordinal of the index value; $\eta_{\text{min}}$ and $\eta_{\text{max}}$ stands for the minimum and maximum values of learning rate, respectively; $T_{\text{cur}}$ refers to the current number of cycles; and $T_i$ means the total number of cycles under the current operating environment.

4.4. Loss Function

The loss function of the improved MobilenetV1-YOLOv4 consists of three parts, which are, in turn, position regression loss, object confidence loss, and classification loss, as shown in Formula (11):

$$Loss = Loss_{\text{loc}} + Loss_{\text{obj}} + Loss_{\text{cls}}$$  \hspace{1cm} (11)

Formula (12) shows the position regression loss:

$$Loss_{\text{loc}} = \lambda_{\text{coord}} \sum_{i=0}^{s \times s} \sum_{j=0}^{M} t_{ij}^{\text{obj}} (2 - W_i \times h_i) (1 - \text{CIOU})$$  \hspace{1cm} (12)

where the computations for CIOU, $\beta$, and $\nu$ follow Formulas (13)–(15), respectively:

$$\text{CIOU} = \text{IOU} - \frac{\rho^2 (b_{\text{obj}}^t)}{c^2} - \beta \nu$$  \hspace{1cm} (13)

$$\beta = \frac{\nu}{1 - \text{IOU} + \nu}$$  \hspace{1cm} (14)

$$\nu = \frac{4}{\pi^2} \left(\arctan \frac{\omega^t}{h^t} - \arctan \frac{\omega}{h}\right)^2$$  \hspace{1cm} (15)

Formula (16) displays the object confidence loss:

$$Loss_{\text{obj}} = -\sum_{i=0}^{s \times s} \sum_{j=0}^{M} t_{ij}^{\text{obj}} \left[\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log \left(1 - C_i\right)\right] - \lambda_{\text{noobj}} \sum_{i=0}^{s \times s} \sum_{j=0}^{M} t_{ij}^{\text{noobj}} \left[\hat{C}_i \log(C_i) + \left(1 - \hat{C}_i\right) \log \left(1 - C_i\right)\right]$$  \hspace{1cm} (16)

Formula (17) displays the classification loss function:

$$Loss_{\text{cls}} = -\sum_{i=0}^{s \times s} \sum_{j=0}^{M} t_{ij}^{\text{obj}} \sum_{c \in \text{classes}} \left[\hat{P}(c) \log(P(c)) + \left(1 - \hat{P}(c)\right) \log(1 - P(c))\right]$$  \hspace{1cm} (17)
From (12) to (17), \( \lambda_{\text{coord}} \) and \( \lambda_{\text{noobj}} \) represent the weight coefficient; \( s \times s \) represents the grid number; \( M \) represents each anchor box; \( w_i \) and \( h_i \), respectively, represent the width and height of the prediction box center point; \( \text{IOU} \) represents the ratio of the predicted value to the true value; \( \rho^2 (b, b_{\text{gt}}) \) represents the Euclidean distance between the center points of prediction box and true value; \( C \) represents the diagonal distance of the smallest closure region that can contain both the prediction box and the true value; \( \beta \) is a parameter to measure the consistency of length-width ratio; \( V \) is a trade-off parameter; \( \hat{C}_i \) is the predicted confidence degree and \( C_i \) is the actual confidence degree; \( I_{ij}^{\text{obj}} \) refers to the jth prediction box of the ith grid that is responsible for predicting the target; and \( I_{ij}^{\text{noobj}} \) refers to the jth prediction box of the ith grid that contains no prediction of target. Figure 8 demonstrates the improved MobilenetV1-YOLOv4 loss function. The smaller the loss function, the better the robustness of the model.

![A Loss Curve](image)

Figure 8. Loss function.

As shown in Figure 8, the loss function of the training set and the verification set has a significant downward trend. After 200 times of training, the loss function gradually converges to a fixed value. After reaching 250 times, the loss function is basically unchanged. This indicates that the improved network has high convergence speed and the trainings work well.

5. Experimental Results and Discussion

This paper takes precision \( P \), recall rate \( R \), average precision (mAP), and frame rate (FPS) as the main evaluation indicators of the proposed algorithm in the experiment. Their formulae are, respectively, as follows:

\[
P = \frac{TP}{TP + FP}
\]

(18)

\[
R = \frac{TP}{TP + FN}
\]

(19)

\[
AP = \int_{0}^{1} P(R)dR
\]

(20)
\[ mAP = \frac{1}{n} \sum_{i=1}^{n} AP \]  
\[ f_{FPS} = \frac{N}{t} \]

In these formulae, \( TP \) refers to the number of correctly predicted positive samples, \( FP \) the number of incorrectly predicted positive samples, and \( FN \) the number of incorrectly predicted negative samples; \( n \) represents the number of object detection categories, \( N \) represents the number of detected pictures, and \( t \) represents the detection time. An intersection ratio, which is the ratio of the intersection of the predicted frame and the real frame to their union, is introduced and set to 0.5 in this experiment. Table 2 provides the experimental results of the first group containing YOLOv4, MobilenetV1-YOLOv4, and the improved MobilenetV1-YOLOv4.

Table 2. Original contrast experiment (the first group).

| Algorithm             | Recall Rate (R)/\% | mAP/% | FPS Frame/s | Model Weight/MB |
|-----------------------|--------------------|-------|-------------|-----------------|
| YOLOv4                | 77.59              | 94.41 | 74          | 244             |
| MobilenetV1-YOLOv4    | 92.29              | 98.55 | 153         | 155             |
| Improved              | 93.25              | 98.81 | 190         | 57.9            |

As can be seen from Table 2, the recall rate of the improved MobilenetV1-YOLOv4 is 21.7% higher than that of YOLOv4, and it reaches a detection speed of 190 frames/s, which may be due to the addition of the scSE attention mechanism and the depthwise separable convolution. In addition, as compared with the YOLOv4 algorithm, the detection speed of the improved MobilenetV1-YOLOv4 also increases by 2.6 times, and the detection precision improves by 4.14%; as compared with MobilenetV1-YOLOv4, the improved MobilenetV1-YOLOv4 also has a higher detection speed and a higher detection precision; the model weight of the improved MobilenetV1-YOLOv4 is significantly reduced as compared with YOLOv4 and MobilenetV1-YOLOv4. This is because scSE can automatically learn the effective features of an image from both image space and feature channel, can suppress useless redundant features, and can better retain image edge information. Finally, more useful features are obtained through feature splicing and convolution. The improved algorithm can identify insulator defects better.

Table 3 presents the comparisons among the improved MobilenetV1-YOLOv4, the MobilenetV1-YOLOv4, the lightweight algorithms MobilenetV2-YOLOv4, MobilenetV3-YOLOv4, and Ghostnet-YOLOv4.

Table 3. Comparison experiment of mainstream lightweight networks (the second group).

| Algorithm             | Recall Rate (R)/\% | mAP/% | FPS Frame/s | Model Weight/MB |
|-----------------------|--------------------|-------|-------------|-----------------|
| MobilenetV1-YOLOv4    | 92.29              | 98.55 | 153         | 155             |
| MobilenetV2-YOLOv4    | 92.53              | 98.95 | 139         | 148             |
| MobilenetV3-YOLOv4    | 90.12              | 97.74 | 129         | 152             |
| Ghostnet-YOLOv4       | 88.92              | 98.02 | 118         | 150             |
| Improved              | 93.25              | 98.81 | 190         | 57.9            |

As can be seen from Table 3, the recall rate of the improved MobilenetV1-YOLOv4 is higher than that of other lightweight algorithms, and its mAP value is higher than that of other lightweight algorithms except that of MobilenetV2-YOLOv4. In addition, the detection speed of the improved MobilenetV1-YOLOv4 is optimal. Although MobilenetV1-YOLOv4, MobilenetV2-YOLOv4, MobilenetV3-YOLOv4, and Ghostnet-YOLOv4 are
lightweight networks, among the enhanced feature extraction networks, $3 \times 3$ convolution is still an ordinary convolution. The large number of network parameters leads to slower detection speed and greater model weight. The improved network is basically composed of deeply separable convolution and $1 \times 1$ convolution composition. This convolution can improve the real-time performance of the algorithm, and under the action of the scSE attention mechanism, it can ensure that the network accuracy will not decline.

Table 4 shows the third group of comparative experiments, which all have the added scSE attention mechanism. Finally, there are five convolution layers behind the SPP, and the $3 \times 3$ convolution is replaced by a depthwise separable convolution.

### Table 4. Comparative experiment after network improvement (the third group).

| Algorithm                  | Recall Rate (R)/% | mAP/% | FPS Frame/s | Model Weight/MB |
|----------------------------|-------------------|-------|-------------|-----------------|
| Improved Ghostnet-YOLOv4   | 91.33             | 97.98 | 135         | 43.8            |
| Improved MobilenetV2-YOLOv4| 92.77             | 98.13 | 168         | 45.5            |
| Improved MobilenetV3-YOLOv4| 90.84             | 97.98 | 164         | 48.9            |
| Improved MobilenetV1-YOLOv4| 93.25             | 98.81 | 190         | 57.9            |

As can be seen from Table 4, as compared with other improved lightweight networks, the improved MobilenetV1-YOLOv4 still has the best detection accuracy and detection speed. In Table 4, all the lightweight networks have been improved. Because the improved Ghostnet-YOLOv4, improved MobilenetV2-YOLOv4, and improved MobilenetV3-YOLOv4 have a small number of network model parameters, the accuracy is lower than that of the improved MobilenetV1-YOLOv4. Through experimental verification, the detection speed of the improved MobilenetV1-YOLOv4 is optimal, with detection accuracy of 98.81% and detection speed of 190 frames/s, therefore, achieving significant improvement in its detection of insulator defects.

As can be seen from Table 5, first, scSE attention mechanism (2 scSEs) is only added after the upsampling results. Secondly, scSE (3 scSEs) is added after preliminary feature extraction. In this paper, we argue that embedding scSE attention modules into different parts of the network produces different experimental results. The semantic information of the feature map initially extracted is not rich, but it still retains the medium and shallow texture information and contour information of the target in the feature map. This information is very important for target detection. After the initial extraction of the three dimensions, embedding the scSE attention module can better enhance the spatial features and channel features of the target in the feature map. In the PANet structure, its feature map shows richer semantic features, larger receptive field, and smaller feature map scale. The scSE attention module can no longer effectively distinguish important spatial and channel features from the highly fused small-scale feature map.

### Table 5. Comparative experiments (the four group).

| Algorithm                  | scSE | Recall Rate (R)/% | mAP/% | FPS Frame/s | Model Weight/MB |
|----------------------------|------|-------------------|-------|-------------|-----------------|
| Improved MobilenetV1-YOLOv4| 2    | 90.84             | 98.07 | 206         | 52              |
| Improved MobilenetV1-YOLOv4| 3    | 90.84             | 98.18 | 196         | 57.6            |
| Improved MobilenetV1-YOLOv4| 5    | 93.25             | 98.81 | 190         | 57.9            |
| YOLOv5                     |      | 98.55             | 99.67 | 109         | 81.8            |
In the experiment, with a gradual increase in scSE, the accuracy gradually increases, but the detection speed becomes smaller and smaller, because, in the same network, the network parameters are also increased. Finally, as compared with YOLOv5, although the accuracy is reduced, the detection speed and model weight are still better than YOLOv5.

Figure 9a–d show the P-R curves of the above four groups of experiments. The insulator defect label name is 1.

In a P-R curve, P represents precision and R represents recall. With recall as the abscissa axis and precision as the ordinate axis, a P-R curve can intuitively display the overall accuracy and recall of the classification algorithm. We can illustrate the superiority of the algorithm by comparing the area size under the P-R curve. It can be seen from Figure 9a–c that the area of the improved MobilenetV1-YOLOv4 curve is large, indicating that its detection accuracy is good. The green line represents YOLOv5, which occupies the largest area due to its highest accuracy.

Figure 10 is a chart of insulator defect prediction results. Regardless of whether the insulator defects in the figure are large or small, they can be effectively detected without missed detection. In addition, in each picture, the confidence level of the improved algorithm is higher than YOLOv4, indicating that the detection accuracy of this algorithm is higher. When the defect is at the edge of the image, the improved YOLOv4 detection effect is better.
Figure 9. P-R curves based on YOLOv4.

In a P-R curve, P represents precision and R represents recall. With recall as the abscissa axis and precision as the ordinate axis, a P-R curve can intuitively display the overall accuracy and recall of the classification algorithm. We can illustrate the superiority of the algorithm by comparing the area size under the P-R curve. It can be seen from Figure 9a-c that the area of the improved MobilenetV1-YOLOv4 curve is large, indicating that its detection accuracy is good. The green line represents YOLOv5, which occupies the largest area due to its highest accuracy.

Figure 10 is a chart of insulator defect prediction results. Regardless of whether the insulator defects in the figure are large or small, they can be effectively detected without missed detection. In addition, in each picture, the confidence level of the improved algorithm is higher than YOLOv4, indicating that the detection accuracy of this algorithm is higher. When the defect is at the edge of the image, the improved YOLOv4 detection effect is better.

6. Conclusions

In this paper, we put forward an insulator defect detection algorithm based on an improved MobilenetV1-YOLOv4. Comparative experiments with other algorithms prove the superiority of the proposed algorithm model, which has good detection accuracy while ensuring the speed.
(1) The accuracy of the insulator defect detection algorithm based on an improved MobilenetV1-YOLOv4 is up to 98.81% with a detection speed of 190 frames/s, which is 116 frames/s, 37 frames/s, 51 frames/s, and 72 frames/s faster than that of YOLOv4, MobilenetV1-YOLOv4, MobilenetV2-YOLOv4, MobilenetV3-YOLOv4, and Ghostnet-YOLOv4, respectively. As compared with the original YOLOv4, the detection accuracy of the proposed insulator defect detection algorithm is 4.14% higher.

(2) We obtain a brand new algorithm structure by replacing the backbone network, adding the scSE attention mechanism, and using depthwise separable convolution to reduce the quantity of network parameters. The experimental results show that the proposed algorithm can effectively detect insulator defects with highly improved detection accuracy and speed.

(3) We put forward a lightweight detection algorithm that is significantly advantageous for real-time detection of insulator defects. This is a direction worthy of research. In the future, we plan to further research insulator faults such as bursting and contamination. With the emergence of a new target detection algorithm, the algorithm proposed in this paper needs to be further optimized. At present, it will take some time to complete the hardware deployment. In the future, we plan to consider applying this model to other detections to achieve the real-time performance and generalization ability of this algorithm.

Author Contributions: Methodology, S.X., J.D., Y.H., L.L. and T.H.; conceived and designed the experiments, S.X. and J.D.; performed the experiments, J.D. and Y.H.; conceptualization and investigation, J.D. and L.L.; analyzed the data, J.D. and T.H.; wrote the paper, S.X. and J.D.; funding acquisition, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61772033 and the Research and the Development Fund of Institute of Environmental Friendly Materials and Occupational Health, Anhui University of Science and Technology, grant number ALW2021YF03.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: The data that support the findings of this study are available on request from corresponding author. The paper has obtained the written informed consent of the patient to publish the paper.

Data Availability Statement: The data that support the findings of this study are available on request from corresponding author.

Acknowledgments: The authors want to thank the editor and anonymous reviewers for their valuable suggestions for improving this paper.

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

References

1. Dong, W.; Liang, H.; Liu, G.; Hu, Q.; Yu, X. Review of Deep Convolution Applied to Target Detection Algorithm. J. Front. Comput. Sci. Technol. 2022, 16, 1025–1042.
2. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision & Pattern Recognition IEEE Computer Society, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
3. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 2016, 39, 1137–1149. [CrossRef] [PubMed]
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Cheng-Yang, F.; Alexander, C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
7. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
8. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767.

9. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. arXiv 2020, arXiv:2004.10934.

10. Han, G.; He, M.; Gao, M.; Yu, J.; Liu, K.; Qin, L. Insulator Breakage Detection Based on Improved YOLOv5. Sustainability 2022, 14, 6066. [CrossRef]

11. Wang, S.; Liu, Y.; Qing, Y.; Wang, C.; Lan, T.; Yao, R. Detection of Insulator Defects with Improved ResNeSt and Region Proposal Network. IEEE Access 2020, 8, 184841–184850. [CrossRef]

12. Gao, Z.; Yang, G.; Li, E.; Liang, Z. Novel Feature Fusion Module-Based Detector for Small Insulator Defect Detection. IEEE Sens. J. 2021, 21, 16807–16814. [CrossRef]

13. Yu, L.; Zhu, J.; Zhao, Q.; Wang, Z. An Efficient YOLO Algorithm with an Attention Mechanism for Vision-Based Defect Inspection Deployed on FPGA. Micromachines 2022, 13, 1058. [CrossRef]

14. Wang, Z.; Liu, X.; Peng, H.; Zheng, L.; Gao, J.; Bao, Y. Railway Insulator Detection Based on Adaptive Cascaded Convolutional Neural Network. IEEE Access 2021, 9, 115676–115686. [CrossRef]

15. Feng, Z.; Guo, L.; Huang, O.; Li, R. Electrical insulator defects detection method based on yolov5. In Proceedings of the 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), Suzhou, China, 14–16 May 2021; IEEE: Manhattan, NY, USA, 2021; pp. 979–984.

16. 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. 2019, 35, 1599–1601. [CrossRef]

17. Liquan, Z.; Mengjun, Z.; Ying, C.; Yanfei, J. Fast Detection of Defective Insulator Based on Improved YOLOv5s. Comput. Intell. Neurosci. 2022, 2022, 8955292. [PubMed]

18. Zhang, X.; Zhang, Y.; Liu, J.; Zhang, C.; Xue, X.; Zhang, H.; Zhang, W. InsuDet: A Fault Detection Method for Insulators of Overhead Transmission Lines Using Convolutional Neural Networks. IEEE Trans. Instrum. Meas. 2021, 70, 1–12. [CrossRef]

19. 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]

20. Liu, J.; Liu, C.; Wu, Y.; Xu, H.; Sun, Z. An Improved Method Based on Deep Learning for Insulator Fault Detection in Diverse Aerial Images. Energies 2021, 14, 4365. [CrossRef]

21. Zhao, W.; Xu, M.; Cheng, X.; Zhao, Z. An Insulator in Transmission Lines Recognition and Fault Detection Model Based on Improved Faster RCNN. IEEE Trans. Instrum. Meas. 2021, 70, 1–8. [CrossRef]

22. Wu, Q.; An, J.; Lin, B. A Texture Segmentation Algorithm Based on PCA and Global Minimization Active Contour Model for Aerial Insulator Images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2012, 5, 1509–1518. [CrossRef]

23. Liao, S.; An, J. A Robust Insulator Detection Algorithm Based on Local Features and Spatial Orders for Aerial Images. IEEE Geosci. Remote Sens. Lett. 2014, 12, 963–967. [CrossRef]

24. She, L.; Fan, Y.; Wang, J.; Cai, L.; Xue, J.; Xu, M. Insulator Surface Breakage Recognition Based on Multiscale Residual Neural Network. IEEE Trans. Instrum. Meas. 2021, 70, 1–9. [CrossRef]

25. Han, G.; Zhang, M.; Wu, W.; He, M.; Liu, K.; Qin, L.; Liu, X. Improved U-Net based insulator image segmentation method based on attention mechanism. Energy Rep. 2021, 7, 210–217. [CrossRef]

26. He, H.; Huang, X.; Song, Y.; Zhang, Z.; Wang, M.; Chen, B.; Yan, G. An insulator self-blast detection method based on YOLOv4 with aerial images. Energy Rep. 2022, 8, 448–454. [CrossRef]

27. Zhang, Z.; Huang, S.; Li, Y.; Li, H.; Hao, H. Image Detection of Insulator Defects Based on Morphological Processing and Deep Learning. Energies 2022, 15, 2465. [CrossRef]

28. Ding, J.; Cao, H.; Ding, X.; An, C. High Accuracy Real-Time Insulator String Defect Detection Method Based on Improved YOLOv5. Front. Energy Res. 2022, 10, 889. [CrossRef]

29. Qiu, Z.; Zhu, X.; Liao, C.; Shi, D.; Qu, W. Detection of Transmission Line Insulator Defects Based on an Improved Lightweight YOLOv4 Model. Appl. Sci. 2022, 12, 1207. [CrossRef]

30. Han, G.; Li, T.; Li, Q.; Zhao, F.; Zhang, M.; Wang, R.; Yuan, Q.; Liu, K.; Qin, L. Improved Algorithm for Insulator and Its Defect Detection Based on YOLOv. Sensors 2021, 22, 6186. [CrossRef]

31. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv 2017, arXiv:1704.04861.

32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

33. Howard, A.; Sandler, M.; Chu, G.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; Quoc, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.

34. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.

35. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More Features from Cheap Operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1580–1589.