

## Research Article

# Insulator Segmentation for Power Line Inspection Based on Modified Conditional Generative Adversarial Network

Zishu Gao,<sup>1,2</sup> Guodong Yang<sup>1</sup>, En Li<sup>1</sup>, Tianyu Shen,<sup>1,2</sup> Zhe Wang,<sup>1,2</sup> Yunong Tian,<sup>1,2</sup> Hao Wang,<sup>1,2</sup> and Zize Liang<sup>1</sup>

<sup>1</sup>*Institute of Automation, Chinese Academy of Sciences, The State Key Laboratory of Management and Control for Complex Systems, 95 Zhongguancun East Road, Beijing 100190, China*

<sup>2</sup>*University of Chinese Academy of Sciences, No. 19(A) Yuquan Road, Beijing 100049, China*

Correspondence should be addressed to Guodong Yang; [guodong.yang@ia.ac.cn](mailto:guodong.yang@ia.ac.cn)

Received 7 August 2019; Accepted 12 October 2019; Published 12 November 2019

Academic Editor: Grigore Stamatescu

Copyright © 2019 Zishu Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There are a large number of insulators on the transmission line, and insulator damage will have a major impact on power supply security. Image-based segmentation of the insulators in the power transmission lines is a premise and also a critical task for power line inspection. In this paper, a modified conditional generative adversarial network for insulator pixel-level segmentation is proposed. The generator is reconstructed by encoder-decoder layers with asymmetric convolution kernel which can simplify the network complexity and extract more kinds of feature information. The discriminator is composed of a fully convolutional network based on patchGAN and learns the loss to train the generator. It is verified in experiments that the proposed method has better performances on mIoU and computational efficiency than Pix2pix, SegNet, and other state-of-the-art networks.

## 1. Introduction

Insulators are widely used in the power transmission system. Once cracked, there would be great failure of power grid system, causing significant economic loss and social chaos [1]. Therefore, it is very necessary to detect the insulators for power line inspection. With the continuous improvement of robotics and image processing technologies, the manual inspection is being replaced by inspection robots or UAVs capable of autonomous inspection, mounted cameras as the sensors for environment perception or defect detection. However, it is very difficult to extract and identify the insulator components from the insulator images, because the insulators have different colour textures, resolution, and spectrum, also with various positions and postures [2]. In addition, the images are always with cluttered background, which makes the insulators difficult to be recognized [3]. Besides, the insulator images may be blurred due to jitter during the movement of the inspection robot [4].

Segmentation of insulators in the aerial images has been a basic problem of insulator inspection. Various researches

have focused on this area. Traditional methods usually leverage various features for insulator inspection. Zhao et al. [5] adopt a localization approach of insulators based on shape points and equidistant model. They use the orientation angle detection and the binary shape prior knowledge to detect different kinds of insulators. The method of [6] benefits from the saliency and adaptive morphology, which fuses the colour and gradient features to detect the insulators. But this method cannot be applied to locate various insulators with inconspicuous colour. Zhai et al. [7] present bunch-drop fault detection to determine the coordinates of insulators, but this method can only be used for glass and ceramic insulators. In [8], the multiscale and multifeature descriptor is proposed to represent the local features. They obtain spatial order features from the local features, then the region of insulators is determined using spatial order features. These methods have similar disadvantage. They present undesirable results when the insulator is very close to the background environment or the background is complex.

Compared with traditional methods, machine learning approaches are robust and accurate for target detection.

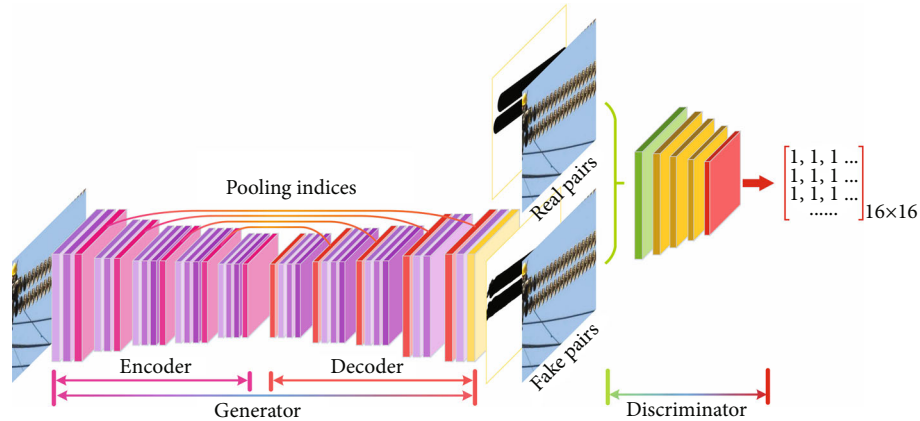


FIGURE 1: Network architecture.

Shang et al. [9] locate the insulators' position based on the maximum between cluster variance and the Adaboost classifier. But this method requires independence between the insulators. The studies in [10] extract the features based on Local Directional Pattern (LDP). A classification model based on Support Vector Machine (SVM) is integrated into sliding window framework for locating insulators. In [11], Binary Robust Invariant Scalable Keypoints (BRISK) and Vector of Locally Aggregated Descriptors (VLAD) are adopted to detect the insulators. These mixed features are classified by SVM. But this method is limited to infrared images. Yan et al. extract the histogram of oriented gradients (HOG) and local binary pattern (LBP) and use sliding window method and SVM to realize the insulator detection [12]. These approaches are basically designed for a specific type of insulators, leading to a lack of adaptability.

While moving ahead with deep learning technology, the above algorithms are gradually replaced. Deep learning has achieved very efficient results in various tasks such as detection, recognition, and segmentation. The studies in [13] construct the saliency area detection framework based on generative adversarial network. However, they use synthetic insulator samples in the training processing and real images in the test experiments, which lack sufficient reliability. In [14], the single shot multibox detector (SSD) combined with a strategy of two-stage fine-tuning is adopted for identifying the insulators. But this method is only used for porcelain insulators and composite insulators. Siddiqui et al. propose a rotation normalization and ellipse detection method. The proposed Convolutional Neural Network- (CNN-) based detection framework achieves detecting 17 different types of insulators [15]. In [16], authors improve the anchor generation method and nonmaximum suppression (NMS) in the region proposal network (RPN) of the faster R-CNN model, which enhance the accuracy and efficiency. But these methods cannot realize real-time detection. Arnab et al. propose that high-order consistency occurs in the CNN-based segmentation method [17]. In [18], authors show that semantic segmentation based on GAN can solve the high-order consistency problem.

In summary, current insulator segmentation methods all have some deficiencies. Feature-based traditional methods

cannot deal with various types of insulators with different scales or shapes. CNN-based segmentation networks lead to high-order consistency that cannot be used in real-time situation. To address these issues, a more adaptive method needs to be devised. In this paper, we use an end-to-end GAN network to achieve pixel-level insulator segmentation. The trained model can achieve segmenting insulators without manually set parameters. It is verified in experiments that the network can produce high-quality pixel-level segmentation of insulators in real time on embedded devices in the routine inspection.

The contributions of this paper are the following: Firstly, a lightweight end-to-end generator with asymmetric convolution kernel is devised to produce pixel-level segmentation of insulators with the original RGB image as input. Secondly, we explore the patchGAN classifier in the discriminator, presenting a punishing function at the scale of image patches.

The rest of this paper is organized as follows: Section 2 discusses the pipeline of our modified conditional generative adversarial network. Section 3 presents the dataset establishment. The experimental evaluation and discussions are proposed on Section 4, and we conclude this paper in Section 5.

## 2. Modified Conditional Generative Adversarial Network

**2.1. Modified Model.** In this section, we introduce the overall description of the proposed network. As shown in Figure 1, the framework is a fully convolutional GAN, which is constituted by two components: a lightweight generator based on encoder-decoder network and a discriminator with classification model based on patchGAN. The generator produces fake segmentation result for a given image. The discriminator takes in both the fake segmentation images and ground truth real images and tries to discriminate real images from fake generated images. During the training process, the generator model is concurrently trained to generate more realistic images, which are hard to discriminate from the ground truth real images.

**2.2. Generator.** The generator follows the encoder-decoder architecture and the details are listed in Table 1. It is composed

TABLE 1: Generator architecture.

Set	Layer name	Type of layers	Output size
Input	RGB image		$256 \times 256 \times 3$
	Conv1	Conv+BN+ReLU, fs = (3 ; 1)	$256 \times 256 \times 64$
Encoder1	Conv2	Conv+BN+ReLU, fs = (1 ; 3)	$256 \times 256 \times 64$
	MP1	Max pooling (window $2 \times 2$ )	$128 \times 128 \times 64$
Encoder2	Conv3	Conv+BN+ReLU, fs = (3 ; 1)	$128 \times 128 \times 128$
	Conv4	Conv+BN+ReLU, fs = (1 ; 3)	$128 \times 128 \times 128$
	MP2	Max pooling (window $2 \times 2$ )	$64 \times 64 \times 128$
Encoder3	Conv5	Conv+BN+ReLU, fs = (3 ; 1)	$64 \times 64 \times 256$
	Conv6	Conv+BN+ReLU, fs = (1 ; 3)	$64 \times 64 \times 256$
	Conv7	Conv+BN+ReLU, fs = (3 ; 3)	$64 \times 64 \times 256$
	MP3	Max pooling (window $2 \times 2$ )	$32 \times 32 \times 256$
Encoder4	Conv8	Conv+BN+ReLU, fs = (3 ; 1)	$32 \times 32 \times 512$
	Conv9	Conv+BN+ReLU, fs = (1 ; 3)	$32 \times 32 \times 512$
	Conv10	Conv+BN+ReLU, fs = (3 ; 3)	$32 \times 32 \times 512$
	MP4	Max pooling (window $2 \times 2$ )	$16 \times 16 \times 512$
Encoder5	Conv11	Conv+BN+ReLU, fs = (3 ; 1)	$16 \times 16 \times 512$
	Conv12	Conv+BN+ReLU, fs = (1 ; 3)	$16 \times 16 \times 512$
	Conv13	Conv+BN+ReLU, fs = (3 ; 3)	$16 \times 16 \times 512$
	MP5	Max pooling (window $2 \times 2$ )	$8 \times 8 \times 512$
Decoder1	UP1	UpSampling	$16 \times 16 \times 512$
	Deconv1	Deconv+BN+ReLU, fs = (3 ; 1)	$16 \times 16 \times 512$
	Deconv2	Deconv+BN+ReLU, fs = (1 ; 3)	$16 \times 16 \times 512$
	Deconv3	Deconv+BN+ReLU, fs = (3 ; 3)	$16 \times 16 \times 512$
Decoder2	UP2	UpSampling	$32 \times 32 \times 512$
	Deconv4	Deconv+BN+ReLU, fs = (3 ; 1)	$32 \times 32 \times 512$
	Deconv5	Deconv+BN+ReLU, fs = (1 ; 3)	$32 \times 32 \times 512$
	Deconv6	Deconv+BN+ReLU, fs = (3 ; 3)	$32 \times 32 \times 256$
Decoder3	UP3 UpSampling		$64 \times 64 \times 256$
	Deconv7	Deconv+BN+ReLU, fs = (3 ; 1)	$64 \times 64 \times 256$
	Deconv8	Deconv+BN+ReLU, fs = (1 ; 3)	$64 \times 64 \times 256$
	Deconv9	Deconv+BN+ReLU, fs = (3 ; 3)	$64 \times 64 \times 128$
Decoder4	UP4	UpSampling	$128 \times 128 \times 128$
	Deconv10	Deconv+BN+ReLU, fs = (3 ; 1)	$128 \times 128 \times 128$
	Deconv11	Deconv+BN+ReLU, fs = (1 ; 3)	$128 \times 128 \times 64$
Decoder5	UP5	UpSampling	$256 \times 256 \times 64$
	Deconv12	Deconv+BN+ReLU, fs = (3 ; 3)	$256 \times 256 \times 64$
	Deconv13	Deconv+tanh, fs = (4 ; 4)	$256 \times 256 \times 3$

of 5 layers of encoding and 5 layers of decoding. Each encoding layer consists of convolutional layer, batch normalization (BN), rectified linear units (ReLU), and max pooling layer. BN is adopted to stabilize training, speed up the convergence, and

regularize the model [19]. Max pooling with a  $2 \times 2$  window and the stride of 2 is inserted between two encoding layers, which achieves subsampling the feature map by a factor of 2. Furthermore, we store the max pooling indices to capture

TABLE 2: Discriminator architecture.

Set	Layer name	Type of layers	Output size
Input	RGB and generated		$256 \times 256 \times 6$
Encoder1	Conv1	Conv+LeakyReLU, fs = (4, 4), s = 2	$128 \times 128 \times 64$
Encoder2	Conv2	Conv+LeakyReLU+BN, fs = (4, 4), s = 2	$64 \times 64 \times 128$
Encoder3	Conv3	Conv+LeakyReLU+BN, fs = (4, 4), s = 2	$32 \times 32 \times 256$
Encoder4	Conv4	Conv+LeakyReLU+BN, fs = (4, 4), s = 2	$16 \times 16 \times 512$
Encoder5	Conv5	Conv+LeakyReLU+BN, fs = (4, 4), s = 2	$16 \times 16 \times 1$

the image's boundary information in the encoder feature maps. In particular, we use two asymmetric spatial filters of  $3 \times 1$  and  $1 \times 3$  instead of  $3 \times 3$ , which deepen the network structure and increase the degree of its nonlinearity. In addition, the  $3 \times 1$  and  $1 \times 3$  filters reduce the number of parameters and yield a more compact generator model, which helps in improving its computational efficiency [20]. The encoder layers predict both low-level and high-level feature maps, which have excellent feature expression capability.

Each decoding layer has a corresponding encoder layer. UpSampling layer is applied to upsample the input feature map utilizing the max pooling indices. As one of the most successful methods in segmentation, the max pooling indices that are stored by the corresponding encoder feature map pass to decoder feature maps, which preserves the boundary details and leads to high segmentation accuracy. BN is inserted between the deconvolution and ReLU. The asymmetric spatial filters are also used to each of these maps. In the absence of asymmetric spatial filters, the entire network parameters have increased by more than 19M, which has a great impact on processing speed.

The generator was built as a lightweight network, but the number of layers is a comprehensive trade-off between time-consuming and segmentation accuracy. The final output of the generator is a  $256 \times 256 \times 3$  segmentation result, which is fed to discriminator model with the input image.

**2.3. Discriminator.** The discriminator model structure is presented in Table 2. The concatenation of the generated image and ground truth real image is the input of the discriminator. The discriminator model has 5 blocks and consists of convolutional layer, LeakyReLU, and BN. The convolutional filter is  $4 \times 4$ , with the stride of 2. BN is added to this model except the first block, which is leveraged to accelerate the network convergence process. LeakyReLU is used to guarantee that neurons will not die when the input is less than 0.

It is well-known that the  $L_1$  loss produces blurry results in the generator, which help to force low-frequency correctness [21].  $L_1$  loss can be defined as follows:

$$L_1(G) = \int_{x,y,z} |ky - G(x,z)k_1|. \quad (1)$$

Hence, the discriminator is motivated to model the high-frequency structure. For this end, the patchGAN is adopted as the discriminator structure. Based on insulator segmentation experiment, we choose patch size  $16 \times 16$  instead of 70

$\times 70$  in [22], which are verified in the effect in the experimental section. The patchGAN maps from  $256 \times 256$  image to a  $16 \times 16$  array of outputs  $X$ , where each  $X_{ij}$  signifies whether the patch  $ij$  in the image is real or fake. It is worth noting that we only use the discriminator during the training phase, so the efficiency is not primary in the experiments.

**2.4. Objective.** The objective function of the network can be defined as follows:

$$G = \arg \min_G \max_D L_{\text{GAN}}(G, D) + \lambda L_1(G),$$

$$L_{\text{GAN}}(G, D) = \int_{x,y} [\log D(x, y)] + \int_{x,z} [\log (1 - D(x, G(x, z)))] \quad (2)$$

where  $\lambda$  is the weight parameter,  $G(x, z)$  is the predicted segmentation image,  $y$  is the ground truth, and  $L_1(G)$  stands for  $L_1$  loss.

As the formula shows, it has two parts. First,  $G$  tries to minimize the accuracy of the discriminator  $D$  that tries to maximize it. In addition, the generator is trained to achieve both fooling the discriminator and producing more realistic image which is similar to the ground truth in an  $L_1$  sense.

### 3. Establishment of Our Dataset

**3.1. Data Collection UAV System.** To accomplish this task, a UAV data acquisition system is designed and shown in Figure 2. The data acquisition system is composed of a Pan-Tilt camera of Zenmuse and a DJI M200 UAV platform and an insulator segmentation method to be proposed. The camera captures the images of insulators on the transmission line, including various types like porcelain insulators and composite insulators.

**3.2. Datasets and Implementation Details.** The insulator datasets are acquired in two ways: the UAV data acquisition system and the Internet. Samples are enhanced by random rotation, mirroring, colour perturbation, and blurring and resized to  $256 \times 256 \times 3$  before training. The datasets consist of 6000 insulator images with more than 6 types, and each image contains 1 to 10 insulators, with an average of 4 insulators per image, adding up to a total of 24,000 insulators. They are divided into a training set of 5000 images, a validation set of 500 images, and a test set of 500 images. It is worth mentioning that a whole strip of connected domains covering the insulator is used as the insulator label, ignoring its edge



FIGURE 2: UAV data acquisition system.

details because for the insulator identification there is no need to mark the shape. Besides, this labeling method not only reduces network complexity but also improves the processing efficiency.

#### 4. Experiments

In this section, we carry out several experiments to demonstrate and validate the following goals. First, we describe the evaluation metrics used in the experiments. Next, we demonstrate the improvement of segmentation accuracy and efficiency comparing our model with state-of-the-art methods. Then, we conduct some experiments to verify the capacity of our generator. Besides, we compare the segmentation results of different patch sizes in the discriminator. Furthermore, the influence of training set image number is evaluated. Finally, we analyse the segmentation results of insulators in different sizes.

All the networks are implemented based on Keras framework using TensorFlow backend. The network is checked out on NVIDIA Tesla V100 server. During the training, we set batch size of 8, Adam optimizer with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ , and learning rate of 0.0001.

**4.1. Evaluation Metrics.** Mean Intersection over Union (mIoU) is a standard for defining the segmentation accuracy. mIoU evaluates the prediction precision of the segmentation. mIoU can be formulated as

$$\text{mIoU} = \frac{1}{n_c} \sum_i \frac{n_{ii}}{\sum_i n_{ij} + \sum_j n_{ji} - n_{ii}}, \quad (3)$$

where  $n_c$  is the number of the dataset classes and  $n_{ii}$  is the calculated number of pixels of class  $i$  predicted to class  $i$ .  $n_{ij}$  is the number of pixels of class  $i$  predicted to class  $j$ , and  $n_{ji}$  is the number of pixels of class  $j$  predicted to class  $i$ .

The average segmentation time of different models is compared in this paper, which is very important for the real-time performance.

**4.2. Analysis of Architecture.** To verify the superiority of the modified network, we compare our method with Pix2pix [22], SegNet [23], Unet [24], and FCN [25]. FCN uses a fully convolutional network to transform image pixels to pixel categories for semantic segmentation. The segmentation-equipped VGG16 net [26] is adopted as the front structure in this experiment. Figure 3 illustrates the segmentation performance of the five models. Table 3 shows the quantitative comparison results. We can see that Unet performs as good as SegNet, and it has the lowest time consumptions. FCN has a slight increase of mIoU, but it has the most parameters and the longest processing time. Pix2pix performs relatively well due to the adoption of GAN, which is similar to our model. The GAN model can correct the higher order inconsistencies between the generated segmentation image and ground truth real image. Our method is superior to other methods with the highest mIoU, the fewest parameters, and the lowest time consumption. It shows that our model with asymmetric spatial filters and patchGAN boosts the performance.

**4.3. Influence of Generator Architecture.** To show the time consumption and segmentation accuracy of our model, we compare several generator models. In this experiment, the same discriminator model with patch size  $16 \times 16$  is leveraged. We call the model that used the spatial filters  $3 \times 3$  as 33 patch16 for convenience. The asymmetric spatial filters  $1 \times 3$  and  $3 \times 1$  are adopted in our models. In addition, we use the same generator as the Unet network, which we call Unet patch16. The difference between the Unet patch16 and Pix2pix is that they have different patch sizes. The comparison results are shown in Table 4. This experiment demonstrates that our method has a little advantage over mIoU, and the parameters are much less than them. It can be seen that the encoder-decoder architecture with asymmetric spatial filters in the generator plays an important role in it.

**4.4. Comparison of Patch Size in the Discriminator.** The patch size of our discriminator influences the segmentation performance. Table 5 shows the qualitative results. We can see that patchGAN with a  $16 \times 16$  patch size is used in all our

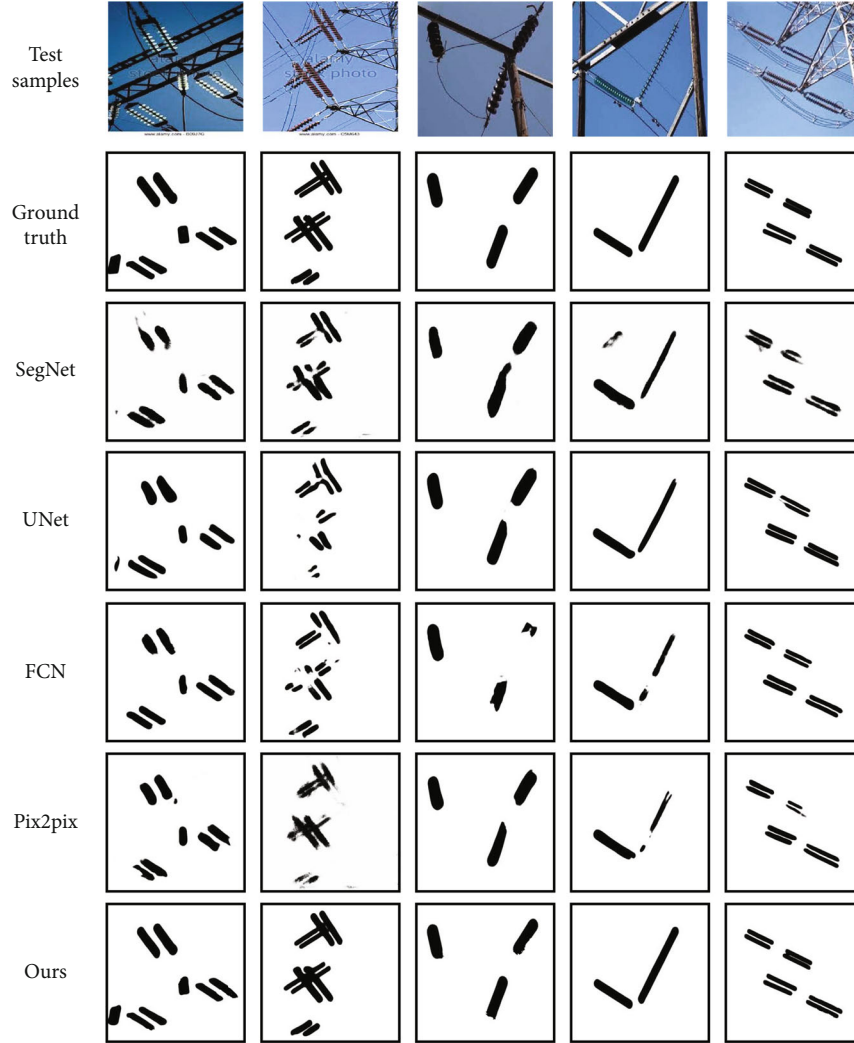


FIGURE 3: Segmentation results of different models.

TABLE 3: Results of different models.

Model	mIoU	Para (M)	Average time (ms)
Pix2pix	0.862	30.81	28
SegNet	0.831	32.44	29
Unet	0.836	28.03	18
FCN	0.853	134.56	266
Ours	0.902	18.73	17

TABLE 4: mIoU and parameters for different generators.

Model	mIoU	Para (M)
Unet patch16	0.882	33.57
33 patch16	0.868	38.01
Ours	0.902	18.73

experiments. Obviously,  $1 \times 1$  means PixelGAN and  $256 \times 256$  means GAN. The  $1 \times 1$  PixelGAN and  $256 \times 256$  GAN obtain results that are not very satisfactory. The  $16 \times 16$  patch

size performs as good as the  $64 \times 64$  patch size, but the  $64 \times 64$  patch size has more parameters.

*4.5. Influence of the Training Set Image Number on Segmentation Results.* To evaluate the influence of training set image number, 1000, 2000, 3000, 4000, and 5000 images are randomly selected to constitute different training datasets. We train the model using these datasets and verify its performance on the same test datasets. Figure 4 shows the mIoU results. The results show that the more training set number, the higher mIoU. But mIoU grows slowly when the training set reaches 3000 or more.

*4.6. Analysis of Segmentation Results of Insulators in Different Sizes.* To verify the ability of our model about detecting various insulators with different scale in the insulator images, Figure 5 shows the segmentation results. The result demonstrates that although the objects in the background are larger than the insulators, our model can still segment the insulators with high quality. Our model has the ability to realize the

TABLE 5: mIoU for different patch size.

Patch size	$1 \times 1$	$16 \times 16$	$64 \times 64$	$256 \times 256$
mIoU	0.836	0.902	0.901	0.842

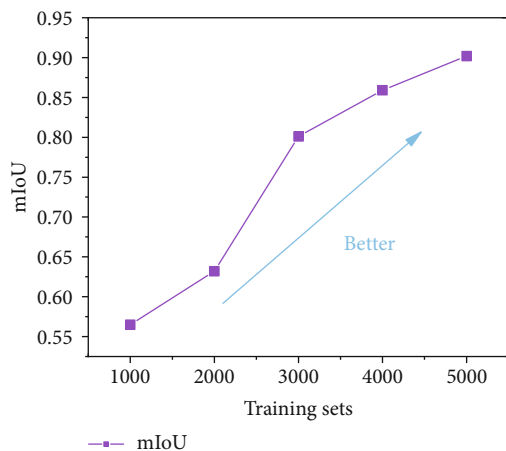


FIGURE 4: mIoU results of different training set numbers.

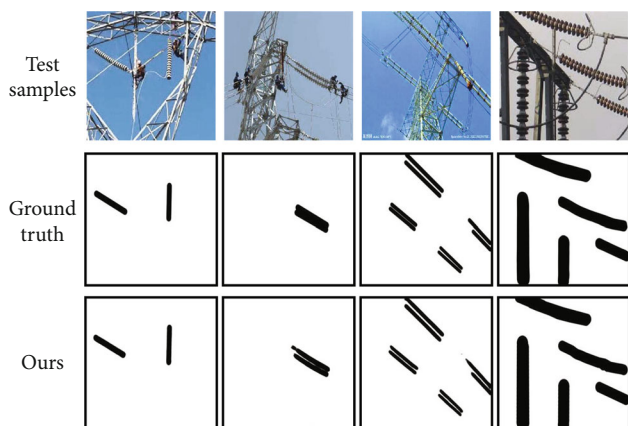


FIGURE 5: Segmentation results of insulators in different sizes.

segmentation of both the near insulators and the distant insulators during the actual detection process.

**4.7. Influence of Noise on Segmentation Results.** To simulate the different weather conditions, we add the salt and pepper noise to the insulator images. In this experiment, three kinds of training datasets are designed: all noisy dataset, half noisy and half noise-free dataset, and noise-free dataset, respectively. We train the three models which are called model noise, model half noise, and model no noise for convenience. Then, we verify the segmentation performance on the same test datasets which are images with salt and pepper noise. Table 6 shows the quantitative comparison results. Figure 6 illustrates the performance of segmentation. We can see that the noisy datasets used in the training process boost the segmentation performance. Therefore, the diversity of training datasets has an important impact on the segmentation results.

TABLE 6: mIoU for different training datasets.

Model	mIoU
Model noise	0.766
Model half noise	0.689
Model no noise	0.302

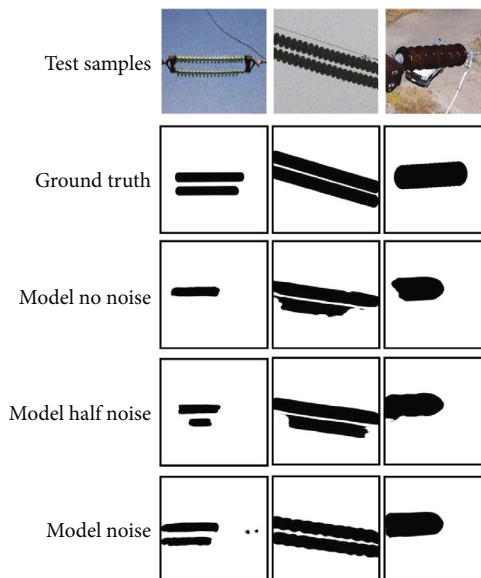


FIGURE 6: Segmentation results of different training datasets.

## 5. Conclusion

In this paper, we introduce a pixel-level insulator segmentation network with modified conditional generative adversarial network. Asymmetric spatial filters are adopted in the generator to reduce network parameters and improve computing efficiency. In addition, we explore the patchGAN classifier in the discriminator to model the high-frequency structure. The network can produce high-quality segmentation of insulators with high mIoU and less time cost compared with the existing end-to-end segmentation methods. Furthermore, the trainable parameters are restricted, which makes the proposed network applicable to real-time segmentation on embedded devices in the future. Additionally, the approach also can be applied to other detection tasks in power inspection.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This project was supported by the National Key Research and Development Plan (2017YFC0806501) and the National Natural Science Foundation (U1713224).

## References

- [1] W. Wang, Y. Wang, J. Han, and Y. Liu, "Recognition and drop-off detection of insulator based on aerial image," in *2016 9th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 162–167, Hangzhou, China, 2016.
- [2] D. Zuo, H. Hu, R. Qian, and Z. Liu, "An insulator defect detection algorithm based on computer vision," in *2017 IEEE International Conference on Information and Automation (ICIA)*, pp. 361–365, Macau, China, 2017.
- [3] Y. Wang, R. Wang, S. Wang, M. Tan, and J. Yu, "Underwater bio-inspired propulsion: from inspection to manipulation," *IEEE Transactions on Industrial Electronics*, p. 1, 2019.
- [4] Y. Han, Z. Liu, D.-J. Lee, G. Zhang, and M. Deng, "High-speed railway rod-insulator detection using segment clustering and deformable part models," in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016.
- [5] Z. Zhao, N. Liu, and L. Wang, "Localization of multiple insulators by orientation angle detection and binary shape prior knowledge," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 22, no. 6, pp. 3421–3428, 2015.
- [6] Y. Zhai, D. Wang, M. Zhang, J. Wang, and F. Guo, "Fault detection of insulator based on saliency and adaptive morphology," *Multimedia Tools and Applications*, vol. 76, no. 9, pp. 12051–12064, 2017.
- [7] Y. Zhai, R. Chen, Q. Yang, X. Li, and Z. Zhao, "Insulator fault detection based on spatial morphological features of aerial images," *IEEE Access*, vol. 6, pp. 35316–35326, 2018.
- [8] S. Liao and J. An, "A robust insulator detection algorithm based on local features and spatial orders for aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 5, pp. 963–967, 2014.
- [9] J. Shang, C. Li, and L. Chen, "Location and detection for self-explode insulator based on vision," *Journal of Electronic Measurement and Instrument*, vol. 31, no. 6, pp. 844–849, 2017.
- [10] T. Jabid and M. Z. Uddin, "Rotation invariant power line insulator detection using local directional pattern and support vector machine," in *2016 International Conference on Innovations in Science, Engineering and Technology (IC-SET)*, Dhaka, Bangladesh, 2016.
- [11] Z. Zhao, G. Xu, and Y. Qi, "Representation of binary feature pooling for detection of insulator strings in infrared images," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 23, no. 5, pp. 2858–2866, 2016.
- [12] Y. Tiantian, Y. Guodong, and Y. Junzhi, "Feature fusion based insulator detection for aerial inspection," in *2017 36th Chinese Control Conference (CCC)*, pp. 10972–10977, Dalian, China, 2017.
- [13] W. Chang, G. Yang, J. Yu, and Z. Liang, "Real-time segmentation of various insulators using generative adversarial networks," *IET Computer Vision*, vol. 12, no. 5, pp. 596–602, 2018.
- [14] X. Miao, X. Liu, J. Chen, S. Zhuang, J. Fan, and H. Jiang, "Insulator detection in aerial images for transmission line inspection using single shot multibox detector," *IEEE Access*, vol. 7, pp. 9945–9956, 2019.
- [15] Z. Siddiqui, U. Park, S.-W. Lee et al., "Robust powerline equipment inspection system based on a convolutional neural network," *Sensors*, vol. 18, no. 11, article 3837, 2018.
- [16] Z. Zhao, Z. Zhen, L. Zhang, Y. Qi, Y. Kong, and K. Zhang, "Insulator detection method in inspection image based on improved faster r-cnn," *Energies*, vol. 12, no. 7, article 1204, 2019.
- [17] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr, "Higher order conditional random fields in deep neural networks," in *Computer Vision – ECCV 2016. ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., vol. 9906 of Lecture Notes in Computer Science, pp. 524–540, Springer, Cham, 2016.
- [18] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," 2016, <https://arxiv.org/abs/1611.08408>.
- [19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, USA, 2016.
- [20] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," 2015, <https://arxiv.org/abs/1502.03167>.
- [21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: feature learning by inpainting," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, Las Vegas, NV, USA, 2016.
- [22] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1125–1134, Honolulu, HI, USA, 2017.
- [23] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds., vol. 9351 of MICCAI 2015. Lecture Notes in Computer Science, pp. 234–241, Springer, Cham, 2015.
- [25] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3431–3440, Boston, MA, USA, 2015.
- [26] K. Simonyan and A. Zisserman, "Very deep convolutional Vnetworks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.





**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

