

# Toward a Cluttered Environment for Learning-Based Multi-Scale Overhead Ground Wire Recognition

Wenkai Chang<sup>1,2</sup>  · Guodong Yang<sup>1</sup> · En Li<sup>1</sup> · Zize Liang<sup>1</sup>

Published online: 10 February 2018  
© Springer Science+Business Media, LLC, part of Springer Nature 2018

**Abstract** In this paper, we propose a learning-based real-time method to recognize and segment an overhead ground wire (OGW) from an image, which is mainly applied to the multi-scale features in a cluttered environment. The recognition and segmentation are implemented under the framework of conditional generative adversarial nets. The generator is an end-to-end convolutional neural network (CNN) with skip connection. The discriminator is a multi-stage CNN and learns the loss function to train the generator. The OGW is recognized and shown in the downsampling visual saliency map. Thus, the location and existence of OGW are verified, which is crucial for the detection in the cluttered environment with structural lines. Detailed experiments and comparisons are performed on real-world images to demonstrate that the proposed method significantly outperforms the traditional method. Additionally, the optimized network achieves approximately 200 fps on a graphics card (GTX970) and 30 fps on an embedded platform (Jetson TX1).

**Keywords** Power line recognition · Conditional generative adversarial nets · Power line inspection · Hybrid robot

## 1 Introduction

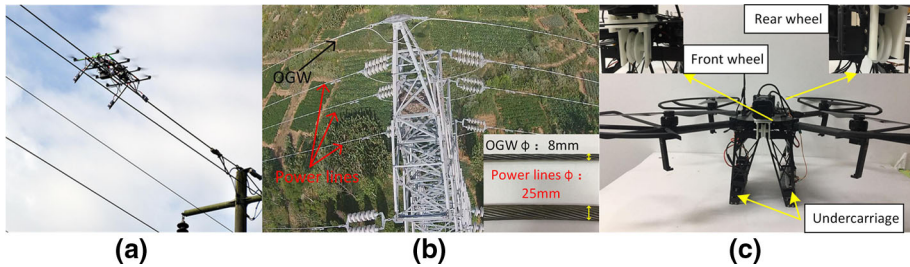
Regular power line inspection is crucial for the normal operation of a power grid system. It is generally carried out by specialized workers with great risks and high labor intensity. During the past decades, many power line inspection robots have been investigated. The improvement in automation level makes them great potential for large-scale autonomic inspection. The wires should be robustly and accurately detected, irrespective of the robot type, e.g., climbing robots [1–3], flying robots [4, 5] and hybrid robots [6–8] (see Fig. 1). Traditional line-based

---

✉ Wenkai Chang  
changwenkai2013@ia.ac.cn

<sup>1</sup> Institute of Automation Chinese Academy of Sciences, Beijing 100190, China

<sup>2</sup> University of Chinese Academy of Sciences, Beijing 100049, China



**Fig. 1** The hybrid robot and pylon. **a** The SKIVE [7]. **b** The pylon. **c** Our prototype machine [8]. These aerial robots move on the power line using wheels most of the time and flies over obstacles when necessary

wire-detection methods have been extensively studied. Other detection methods are available based on the strip texture. However, none of these methods can handle a significant change in the wire features when robots approach to them.

A wire in an image is considered to be rectangular area when a robot travels along the conductors. Clean background (e.g., the sky) facilitates extraction of the wires using binarization-based segmentation algorithm. Reference [9] utilized the adaptive binarization process to pre-divide the area of the wires and extract the corresponding lines using Hough transformation. The lines of the metal structure are eliminated by a prior knowledge of the robot direction. The requirements for clean background is reduced in [10]. The strip texture of the wires (see Fig. 1b) is investigated using an edge operator and the wire area is segmented out using a gray-scale distribution feature. However, this process is only suitable for wires with a specific scale and direction. Moreover, laser range finder (LRF) is employed to detect the wires. LineScout [11, 12] installed a 2D LRF. The wire was extracted by the gradient-based feature.

The wire is treated as a line when it is detected from a distance. In some cases, various special sensors are utilized. Such as, the millimeter-wave radar [13] and synthetic aperture radar [14]. However, the installation may be costly in terms of weight, space, and power, all of which present constraints for mobile robots, especially aerial robotics. Vision-based systems are more suitable for deployment of aerial robot. The applicable methods generally assume that the lines in the image correspond to the wires to be detected [15–17]. Therefore, extracting the lines is the first step. The edge extraction using Hough transformation is the basic method [18]. Furthermore, the line segment detector (LSD) [19] is utilized to process the images with indefinite number of lines [15]. However, this extracted lines need subsequent processing to delete the fake wires with additional information. Such as the pylons are detected as context information [16]. The near-infrared images is used in Ref. [15] because of the obvious difference between the wire and air in the near-infrared image. However, when the robot worked close to the wire, the special targets were sparsely captured with a narrow view. Artificial structures (e.g., roads, buildings, and pylons) in cluttered background can produce many lines, and the wire could be barely distinguished without additional information.

For the power line inspection robots, a faster detection method is urgently needed. However, the detection of the thin OGW (8 mm) is a challenge. At present, relevant research is very scarce. Unlike observers at a fixed distance, the scale of the wire gradually becomes larger when the robot approaches it. Such as, the OGW is more suitable to be considered as a line when the robot flies from a distance. In another case, the OGW diameter cannot be ignored when the robot lands on the wire. Instead, the wire should be treated as a quadrilateral region with spiral strips. Therefore, the algorithms that independently utilize the strip and line-based

features can not be applied throughout the whole landing stages. Additionally, landing on the OGW as soon as possible is beneficial to reduce energy consumption. Therefore, the real-time detection is crucial in landing.

In view of the above shortcomings, in this paper, the OGW is recognized as a linear object with spiral strips, which combines the line and strip-based features. The recognition is implemented using an end-to-end inference. An encode-decode network with skip connection is proposed as a generator that creates the OGW visual saliency maps using color images as input. In this framework, the existence of OGW is combined with the detection of its location. cGAN has been vigorously studied in the last 2 years, and many techniques have been previously proposed. Nonetheless, this present study focuses on the fast inference and robustness of the generator in this specific application, especially achieves real-time process in an embedded platform.

The remainder of this paper is organized as follows. Section 2 describes the network architecture. The loss functions and the training samples are discussed in Sect. 3. Section 4 provides the comparison with other methods and multiple variants. Finally, Sect. 5 ends the paper by drawing the main conclusions.

## 2 The Structure of the Network

### 2.1 cGAN Framework

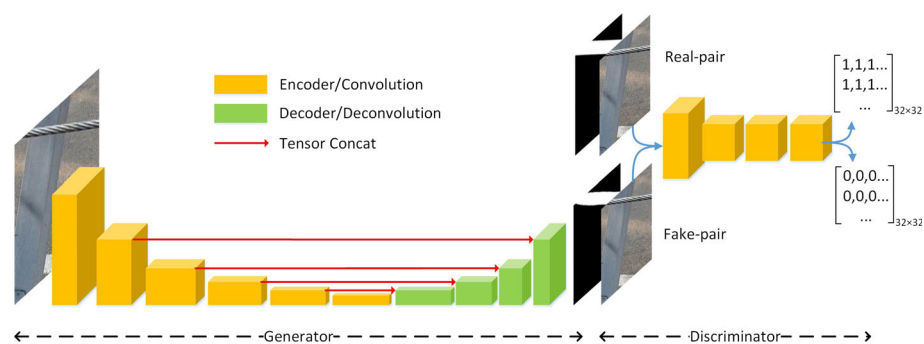
CNN is widely used in object detection, semantic segmentation and visual saliency prediction [20–22]. In general, the OGW passes through the whole image. A slender object is inappropriately labeled as rectangular, although it is widely used in learning-based object detection methods [23–25]. We consider the OGW recognition as the prediction of OGW visual saliency maps. Thus, the pose and the existence of an OGW are clear when the wire is highlighted in the saliency map.

The network should be processed in real time. The pre-trained networks (e.g., VGG [26] and ResNet [27]) are inappropriate for fast processing in an embedded platform. Inspired by the symmetrical neural network architecture in [22] and [28], we propose a compact neural network under the cGAN framework, which is composed of a generator and a discriminator. We call this process as OGW recognition network (ORNet) for convenience.

### 2.2 Generator

As shown in Fig. 2, the generator follows the encoder-decoder architecture and the details are listed in Table 1 for the input image resolution of  $256 \times 256$ . The generator is responsible for generating the OGW visual saliency maps for the given images.

The generator input is a three channel color image that is normalized at  $[-1, 1]$ . In spite of the pooling layer, the filter with interval stride implements the downsampling. The filter size (fs) is  $3 \times 3$  and stride (s) is 2. Between each convolutional layer and following non-linearity, we use the batch normalization (BN) [29], which readjusts the distribution of the feature maps and accelerates the convergence of the module. The PReLU [30] is applicable for the range  $[-1, 1]$ , and responds to both positive and negative values. The dropout layers individually connect to the BN layers to enrich the extracted features and improve the generalization ability. The deconvolution layer is the transpose of convolution. The bias are not used in any layer to reduce processing time. Because the cuDNN [31] uses separate kernels for convolution and bias addition [21]. The output of the generator is the OGW visual saliency



**Fig. 2** Architecture of ORNet

**Table 1** Generator architecture

Name	Type	Size
Input	Color image	$256 \times 256 \times 3$
$encoder_1$	conv, fs = (3,3), s = 2	$128 \times 128 \times 64$
$encoder_2$	PReLU + conv + BN, fs = (3,3), s = 2	$64 \times 64 \times 128$
$encoder_3$	PReLU + conv + BN, fs = (3,3), s = 2	$32 \times 32 \times 256$
$encoder_4$	PReLU + conv + BN + dropout, fs = (3,3), s = 2	$16 \times 16 \times 256$
$encoder_5$	PReLU + conv + BN + dropout, fs = (3,3), s = 2	$8 \times 8 \times 256$
$encoder_6$	PReLU + conv + BN, fs = (3,3), s = 2	$4 \times 4 \times 256$
$decoder_5$	ReLU + deconv + BN, fs = (3,3), s = 2	$8 \times 8 \times 256$
$decoder_4$	ReLU + deconv + BN, fs = (3,3), s = 2	$16 \times 16 \times 128$
$decoder_3$	ReLU + deconv + BN, fs = (3,3), s = 2	$32 \times 32 \times 64$
$decoder_2$	ReLU + deconv + BN, fs = (3,3), s = 2	$64 \times 64 \times 32$
$decoder_1$	ReLU + deconv + tanh, fs = (3,3), s = 2	$128 \times 128 \times 1$ (the output)

map with the size of  $128 \times 128$ , which is less than the original image resolution. Thus, only the relative position and direction of the wire in the image are considered. The purpose is to reduce the number of layers to improve the inference speed. This strategy does not appear in the previous saliency prediction or semantic segmentation methods.

In the absence of skip connection, the generator requires that all information flow passes through all layers, including the bottleneck. For precise OGW location, the low-level information should be considered, and directly shuttling this information across the network would be desirable. Skip connections provide the generator a means to circumvent the bottleneck of the information. Therefore, skip connections are added between each  $encoder_i$ – $decoder_i$ , where  $i = 2, 3, 4, 5$ , which concatenates all channels at  $encoder_i$  with those at  $decoder_i$ .

### 2.3 Discriminator

The discriminator is only utilized in the training stage. Table 2 lists the layer configuration of the discriminator, which is a full convolution network. The half sampled color image and the single channel saliency map are concatenated as input of the discriminator. The output of the discriminator is divided into  $M \times M$ . Each element corresponds to a small patch in the

**Table 2** The discriminator architecture

Name	Type	Size
Input	Saliency map and color image	$128 \times 128 \times 4$
<i>encoder</i> <sub>1</sub>	LReLU + conv + BN, fs = (3,3), s = 2	$64 \times 64 \times 128$
<i>encoder</i> <sub>2</sub>	LReLU + conv + BN, fs = (3,3), s = 2	$32 \times 32 \times 256$
<i>encoder</i> <sub>3</sub>	LReLU + conv + BN, fs = (3,3), s = 1	$32 \times 32 \times 256$
<i>encoder</i> <sub>4</sub>	LReLU + conv + sigmoid, fs = (3,3), s = 1	$32 \times 32 \times 1$ (the output)

input. It tries to classify if each patch is real or fake by inspecting the local texture of the synthetic image [28]. The discriminator shows how the binary classification between real and synthetic samples significantly benefits the accuracy without needing to specify a tailored loss function.

### 3 Loss Function and Training Samples

#### 3.1 Loss Function

The ORNet is trained by the combination of the content and adversarial losses. The content loss follows the  $L_I$  loss. The predicted OGW visual saliency map is pixel-wise compared with the ground truth. Combined with the structure of the network, given image  $I^{256 \times 256}$ , the saliency map  $S^{128 \times 128}$  is represented as a matrix of probabilities, where  $S_j \in [0, 1]$  is the probability of patch  $I_j^{2 \times 2}$  to belong to the OGW. The content loss  $\mathcal{L}_{cont}$  is defined as

$$\mathcal{L}_{cont} = \frac{1}{N} \sum_{j=1}^N (S_j - \hat{S}_j) \quad (1)$$

where  $\hat{S}_j$  is the output of the generator and the  $S$  is the ground-truth saliency map. The resolutions both are  $128 \times 128$ .

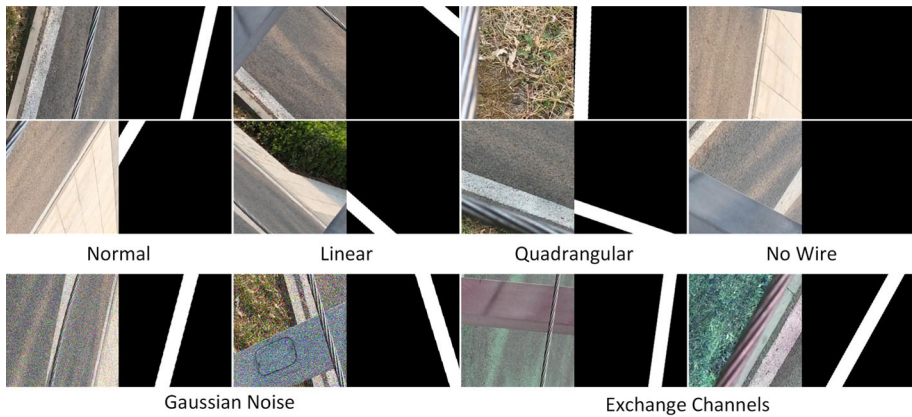
An appropriate loss for the discriminator is the average of the binary cross entropies across the  $M \times M$  determination.

$$\mathcal{L}_{dis} = -\frac{1}{N} \sum_{k=1}^N \left[ \log(P_k) + \log(\mathbf{1}^{M \times M} - \hat{P}_k) \right] \quad (2)$$

where the  $N = M \times M$ . The  $P_k$  is the output of the discriminator that considers a real pair as input, and the  $\hat{P}_k$  is calculated by the discriminator using a fake pair as input. To realize the adversarial training between generator and discriminator, the generator is trained by keeping the discriminator weights constant and back-propagating the error through the discriminator to update the generator weights.

$$\mathcal{L}_{GAN} = -\frac{1}{N} \sum_{k=1}^N \log(\hat{P}_k) \quad (3)$$

$$\mathcal{L}_{gen} = \mathcal{L}_{cont} \cdot \beta + \mathcal{L}_{GAN} \cdot \gamma \quad (4)$$



**Fig. 3** The samples for training

The  $\mathcal{L}_{gen}$  is the adversarial losses and the  $\beta$  and  $\gamma$  are the coefficients. In practice, the training of generator and discriminator are carried out at the same time. In the initial training stage,  $\mathcal{L}_{cont}$  is used to train the generator to make the subsequent training more stable.

$$\arg \min_{W_G} \{ \mathcal{L}_{cont} | batch[I, S] \} \quad (5)$$

where the  $W_G$  represents the generator weights. Thus, function (5) yields  $S_j = \hat{S}_j$ . In the adversarial training stage

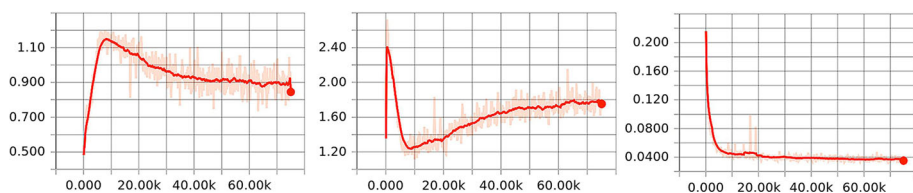
$$\arg \min_{W_G} \{ \mathcal{L}_{gen} | batch[I, S], \overline{W_D} \} \quad (6)$$

$$\arg \min_{W_D} \{ \mathcal{L}_{dis} | batch[I, S], \overline{W_G} \} \quad (7)$$

where the  $\overline{W_D}$  and  $\overline{W_G}$  are respectively the parameters of discriminator and generator in the previous stage. Therefore, the function (6) yields  $\hat{P}_k = \mathbf{1}^{M \times M}$ , which aims to make the saliency map more realistic to confuse the discriminator. The function (7) tries to obtain  $P_k = \mathbf{1}^{M \times M}$  and  $\hat{P}_k = \mathbf{0}^{M \times M}$ , which aims to improve the ability of identification of the discriminator.

### 3.2 Samples

The typical samples are shown in the Fig. 3. Each sample pair is composed of a color image (input) and the OGW visual saliency map (output). The *Normal* samples contain the wire with the clear strip texture. They are mainly influenced by the lines that correspond to the artificial structure. The *Linear* samples are slightly farther than the *Normal* ones. The strip texture is illegible but still exhibit special pattern. Moreover, some of the samples only contain a short segment with other messy lines interference. The *Quadrangular* samples emphasize the strip texture by close observation. The *No Wire* samples contain nothing but the background, which is a challenge to distinguish the wires from cluttered environment. Two types of additional samples are present. Some of them add Gaussian random noise, and the others randomly exchange the color channels (see *Gaussian Noise* and *Exchange Channels* in Fig. 3).



**Fig. 4** The losses. From left to right:  $\mathcal{L}_{dis}$ ,  $\mathcal{L}_{GAN}$  and  $\mathcal{L}_{cont}$

## 4 Experiments

The proposed ORNet model was assessed and compared from different perspectives. The generated saliency maps was analyzed in detail, including the intermediate maps at different training epoch and the imperfections without the discriminator. The ORNet was also compared with other methods and the variants.

### 4.1 Training and Generator Outputs

The network was implemented using TensorFlow and trained using an Adam solver. The learning rate was  $l = 2 \times 10^{-3}$  and decayed 0.85 every epoch. There were 5000 training samples and 500 testing samples. This testing samples are *Test Set1*. Moreover, another 500 samples of different lighting conditions are added to form the *Test Set2*, which focuses on the generalization of the methods. The generator was trained using the content loss for one epoch as the initial weight. In 85 min, 15 epochs of adversarial training were implemented on a graphics card (GTX970) and the CPU (i7-4770k).

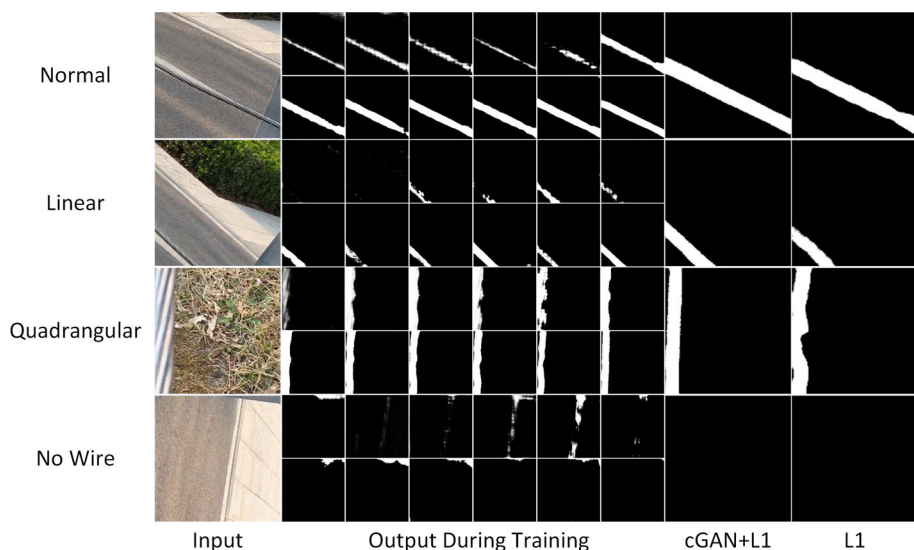
The losses are shown in Fig. 4. The content loss ( $\mathcal{L}_{cont}$ ) rapidly decreased in the initial stage. The GAN-loss ( $\mathcal{L}_{GAN}$ ) of the generator and the discriminator loss ( $\mathcal{L}_{dis}$ ) were all large beat. As shown in Fig. 5, a clear saliency map was gradually formed in the initial epochs, the generator quickly learned the pattern OGW. In the following training,  $\mathcal{L}_{dis}$  continued to decrease, which meant the performance of the discriminator steadily improved until the Nash equilibrium. Meanwhile, the  $\mathcal{L}_{cont}$  was barely reduced, and the generator network was refined for the unclear samples (see Fig. 5 *Linear* samples). The *No Wire* sample in Fig. 5 indicated that the generator attempted to extract the lines in the initial epochs, which was corrected later.

Compared with the generated saliency maps using only the content loss ( $\gamma = 0$ ), the  $\mathcal{L}_{GAN}$  sharpened the edges (see L1 and cGAN+L1 in Fig. 5). The generator produces blurry saliency map [32], which prompts the restriction of the discriminator to model high-frequency information to sharpen the edges in the saliency map.

### 4.2 Evaluation Standards

The Intersection over Union (IOU), the basic evaluation standard of image segmentation, which is used to evaluate the output of different methods. The pixels that are greater than 0.3 in the visual saliency map belong to the wire area. It is worth noting that the higher IOU is not the purpose of this article, but rather the accuracy rate  $r$  of total test samples is more suitable to reflect the detection effect. Combined with the actual situation of wire inspection, the output is considered to be correct when the IOU is greater than 0.4.





**Fig. 5** The generator output of some typical samples

### 4.3 Comparison of Different Methods

The structure-based feature methods, local-feature method and network-feature method are used for comparison.

*The structure-based feature* The LSD [19], an effective line extractor, was used in the line-based method. The HOG feature was used in the strip-based method [10]. Both the outputs of line-based and strip-based methods are messy. Therefore, image erosion was added as the post process to filter the noise after lines extraction by LSD, which improved the outputs of line-based and made the comparison more meaningful.

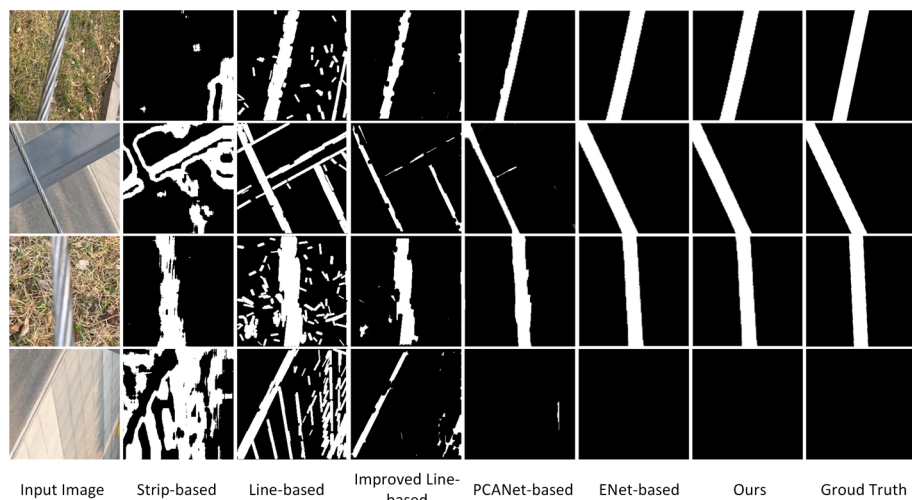
*The local-feature* The PCANet [33] is based on the principal components analysis (PCA), which is an interpretable feature extraction method in mathematics. The parameters of PCANet are set to  $L_1 = L_2 = 8$  and  $k_1 = k_2 = 7$ . The block size is  $8 \times 8$ . The original PCANet can only classify the entire image. The histogram of *block size* that surrounds every pixel in the output layer of  $L_2$  is calculated. The histograms of the corresponding positions are concatenated, which is used to determine whether the pixel is the wire.

*The network-feature* The ENet [21] is a fast segmentation network that has been verified on the embedded platform (Jetson TX1). In this paper, the final softmax layer of ENet is replaced by the probability output layer (the visual saliency maps). The processing speed and generalization ability will be compared in the subsequent experiments.

The outputs of different methods are shown in Fig. 6 and Table 3. The traditional structural-based features are susceptible to the cluttered environments (see *Strip-based* and *Line-based* in Fig. 6). In particular, the strip-based method is only valid for some special scale wires. The line-based method can not distinguish the real wires when the environment contains many lines, which is further validated in the *No Wire* sample (the last row in Fig. 6). In contrast, not only our method, but also the other methods that based on higher-level features have better accuracy.

Compared with the full-image feature obtained by the multi-layer convolution, the PCANet represents a local feature. After careful adjustment of the parameters, the network has a good





**Fig. 6** Qualitative comparison of the different methods

**Table 3** The accuracy and IOU of different method

Method	Test Set1		Test Set2	
	IOU	$r$	IOU	$r$
Strip-based	0.157	0.185	0.162	0.202
Line-based	0.278	0.269	0.254	0.235
Improved line-based	0.620	0.705	0.593	0.657
PCANet-based	0.854	0.899	0.739	0.837
ENet-based	0.954	0.976	0.908	0.931
Ours	0.942	0.966	0.897	0.930

detection accuracy on most samples. But there are two issues that can not be ignored. One problem is the ambiguity of the edge. The observation field of PCANet is limited, which essentially equivalent to judging whether the patch cropped from the image belongs to the wire. Thus, the extracted feature is unstable when the observed conductor is at the edge of the patch. This can be verified from the rough edges in *PCANet-based* Fig. 6. The second issue is the limited generalization. As shown in Table 3, The accuracy and IOU of PCANet is declined. However, the network-feature methods, ENet and ORNet, maintain a good generalization.

The ENet is an effective semantic segmentation network. As shown in Fig. 6 and Table 3, The output, accuracy and IOU of ENet are close to the method in this paper. However, as mentioned in the introduction session, the wire must be detected in real time. We will compare the processing speed of ORNet and ENet in the next session.

#### 4.4 Comparison of Different Architecture

The number of layers and the number of feature maps are the two main factors of the change in the architecture. For the 12-layer architecture (12-1 and 12-2 in Table 4), a decoder module ( $256 \times 256 \times 64$ ) was added after the *decoder*<sub>1</sub> of the generator, and an encoder module ( $256 \times 256 \times 64$ ) was added before the *encoder*<sub>1</sub> of the discriminator. In addition, skip

**Table 4** Performance of variants

Net	Time(ms)		$r$	Parameters
	TX1	GTX970		
ORNet	33	5	0.930	4.360M
11-2	119	12	0.932	16.835M
12-1	40	6	0.931	4.313M
12-2	130	14	0.934	16.650M
ENet	49	7	0.931	0.374M

connections between the encoders and decoders in the generator were added, and the training samples were adjusted to fit the size of the input and output. We set *Number-1* to mean the  $N$ -layer architecture with the same feature maps as the ORNet and *Number-2* to mean the double feature maps.

As listed in Table 4, the accuracy  $r$  is slightly improved whether to increase the number of layers and feature maps. However, the time consumed is significantly increased, which seriously reduced the real time. Because of skip connections, the number of parameters in 11-1,2 was slightly more than that in 12-1,2. Nevertheless, the processing speed of 11-1,2 is faster than that of 12-1,2, which shows that the consumed time between layers at the code-level was not negligible in tiny network architecture. It is worth noting that ENet contains the least parameters, however, consuming more time than ORNet due to the branches in the structure.

## 5 Conclusion

In this paper, the OGW is perceived by the end-to-end networks under the cGAN framework. The low-resolution visual saliency maps accurately show the location and existence of OGW. The threshold setting and complex treatment scheme in the traditional methods were removed. The multi-scale features that fuse the line and strip-based features are learned by the generator in the adversarial training. At the same time, the discriminator learns the abstract loss function to sharpen the edges in saliency maps. We are certain that this is the first time the empty samples (*No Wire* images) are clearly fitted into the detection framework. The experiments demonstrated the great advantages of our method. Additionally, the optimal architecture is verified on embedded processor (Jetson TX1) in real time. Thus, this wire detection method can be used in various power line inspection robots without much additional labor.

**Acknowledgements** This work is supported by the National Natural Science Foundation of China (Grant No. 61403374)

## References

1. Jaka K, Pernus F, Likar B (2010) A survey of mobile robots for distribution power line inspection. *IEEE Trans Power Deliv* 25(1):485–493
2. Pagnano A, Höpf M, Teti R (2013) A roadmap for automated power line inspection maintenance and repair. *Procedia Cirp* 12:234–239
3. Pouliot N, Richard PL, Montambault S (2015) LineScout technology opens the way to robotic inspection and maintenance of high-voltage power lines. *IEEE Power Energy Technol Syst J* 2(1):1–11

4. Matikainen L, Lehtomäki M, Ahokasa E, Hyypää J, Karjalainen M, Jaakkola A, Kukkoa A, Heinenenb T (2016) Remote sensing methods for power line corridor surveys. *ISPRS J Photogramm Remote Sens* 119:10–31
5. Jones DI, Earp GK (2001) Camera sightline pointing requirements for aerial inspection of overhead power lines. *Electr Power Syst Res* 57(2):73–82
6. Hydro-Québec (2017) Drone used to inspect power transmission systems. YouTube. <http://mir-innovation.hydroquebec.com/mir-innovation/en/medias-news.html>. Accessed 10 Oct 2017
7. SKIVE AVIATION AIRBORNE ROBOTICS (2015) SKIVE powerline robot. YouTube. <http://www.skive.aero/index.php/powerline-inspection>. Accessed 10 Oct 2017
8. Chang W, Yang G, Zhi J, Liang Z, Cheng L, Zhou C (2017) Development of a power line inspection robot with hybrid operation modes. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 973–978
9. Zhao D, Yang G, Li E, Liang Z (2013) Design and its visual servoing control of an inspection robot for power transmission lines View Document. In: *2013 IEEE international conference on robotics and biomimetics (ROBIO)*, pp 546–551
10. Song Y, Wang H, Zhang J (2014) A vision-based broken strand detection method for a power-line maintenance robot. *IEEE Trans Power Deliv* 29(5):2154–2161
11. Pouliot N, Richard P, Montambault S (2012) LineScout power line robot: characterization of a UTM-30LX LIDAR system for obstacle detection. In: *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp 4327–4334
12. Richard P, Pouliot N, Montambault S (2014) Introduction of a LIDAR-based obstacle detection system on the LineScout power line robot. In: *2014 IEEE/ASME international conference on advanced intelligent mechatronics (AIM)*, pp 1734–1740
13. Ziegler V, Schubert F, Schulte B, Giere A, Koerber R (2013) Helicopter near-field obstacle warning system based on low-cost millimeter-wave radar technology. *IEEE Trans Microw Theory Tech* 61(1):658–665
14. Deng S, Li P, Zhang J, Yang J (2012) Power line detection from synthetic aperture radar imagery using coherence of co-polarisation and cross-polarisation estimated in the Hough domain. *IET Radar Sonar Navig* 6(9):873–880
15. Luo X, Zhang J, Cao X, Yan P, Li X (2014) Object-aware power line detection using color and near-infrared images. *IEEE Trans Aerosp Electron Syst* 50(2):1374–1389
16. Shan H, Zhang J, Cao X (2015) Power line detection using spatial contexts for low altitude environmental awareness. In: *Integrated communication navigation and surveillance conference (ICNS)*, w2–1–w2–10
17. Yetgin ÖE, Sentürk Z, Gerek ÖN (2015) A comparison of line detection methods for power line avoidance in aircrafts. In: *International conference on electrical and electronics engineering (ELECO)*, pp 241–245
18. Song B, Li X (2014) Power line detection from optical images. *Neurocomputing* 129:350–361
19. Gioi RG, Jakubowicz J, Morel J, Randall G (2012) LSD: a line segment detector. *Image Process Line* 2:35–55
20. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: *European conference on computer vision (ECCV)*, pp 21–37
21. Paszke A, Chaurasia A, Kim S, Culurciello E (2016) ENet: a deep neural network architecture for real-time semantic segmentation. [arXiv:1606.02147](https://arxiv.org/abs/1606.02147)
22. Pan J, Ferrer C, McGuinness K, Connor NE, Torres J, Sayrol E, Nieto X (2017) SalGAN: visual saliency prediction with generative adversarial networks. [arXiv:1701.01081](https://arxiv.org/abs/1701.01081)
23. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. In: *Advances in neural information processing systems (NIPS)*, pp 91–99
24. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: *The IEEE conference on computer vision and pattern recognition (CVPR)*, pp 779–788
25. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. [arXiv:1703.06870](https://arxiv.org/abs/1703.06870)
26. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
27. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *The IEEE conference on computer vision and pattern recognition (CVPR)*, pp 770–778
28. Isola P, Zhu J, Zhou T, Efros A (2016) Image-to-image translation with conditional adversarial Networks. [arXiv:1611.07004](https://arxiv.org/abs/1611.07004)
29. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. [arXiv:1502.03167](https://arxiv.org/abs/1502.03167)
30. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp 1026–1034

31. Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B, Shelhamer E (2014) cuDNN: efficient primitives for deep learning. [arXiv:1410.0759](#)
32. Larsen ABL, Sønderby SK, Larochelle H, Winther O (2015) Autoencoding beyond pixels using a learned similarity metric. [arXiv:1512.09300](#)
33. Chan T, Jia K, Gao S, Lu J, Zeng Z, Ma Y (2014) PCANet: a simple deep learning baseline for image classification? *IEEE Trans Image Process* 50(2):1374–1389