Obstacle Detection for Power Transmission Line Based on Deep Learning

Yun Wang[1,2], Hongli Gao[1,2*], Yuekai Liu[1,2], Liang Guo[1,2], Caijiang Lu[1,2], Lei Li[1,2], Yu Liu[3], Xianyin Mao[3]
[1]College of mechanical engineering
Southwest Jiaotong University
Chengdu, China
[2]Engineering Research Center of the Ministry of Education of Advanced Driving Energy saving Technology
Southwest Jiaotong University
Chengdu, China
[3]Electric Power Research Institute of Guizhou Power Grid Co., Ltd
Guizhou, China
Hongli_Gao@home.swjtu.edu.cn

Abstract—**Power line inspection robots often work in the mountains, surrounded by trees and other background disturbances, while the field illumination changes significantly. The distance between the camera and the obstacles has a great influence on the scale changes of the obstacles in the image space. Therefore, accurate obstacle detection is very difficult. The deep learning methods such as Single Shot Multi-Box Detector (SSD) algorithm can be insensitive to illumination and scale changes in complex background with a much high accurate detection. But the amount of parameters of the SSD are so huge that make it very difficult to transplant to embedded systems. Aiming at the problem of stable and accurate detection of obstacles and easy to transplant to embedded systems, this paper proposes a method which significantly reduces the model parameters and improves the detection speed. In this method, the 23 convolution layers of the original SSD are simplified to 7 layers, and the Batch Normalization layer is added after each convolution layer to normalize the convolutional data. The result shows that the algorithm not only ensures the detection accuracy, but also greatly reduces the parameter quantities of the model and improves the detection speed significantly from 4.5 fps of original SSD to 15 fps of simplified SSD on Jetson TX2(an embedded system). Compared with some classical computer vision based detection algorithms, the method is more adaptable to complex environments.**

Keywords-Machine Vision; Inspection robot; Obstacle detection; Deep learning

I. INTRODUCTION

Today's society is increasingly dependent on the stable transmission of electricity, and the disruption of electricity will have a huge impact on social production and life. The transmission line, as a tool for remote power transmission, exposed to the sun, wind and rain, coupled with continuous mechanical tension will lead to it wear and break. These are serious threats to the stable power transmission[1].

At present, there are three main methods of patrol inspection. The first one is pedestrian patrol inspection. The electric power staff carries a telescope and other equipment to conduct manual inspections on the power line on the ground. This method is dangerous and labor intensive[2]. The second is to use helicopter-equipped equipment for inspection[3], which is costly. The third is to use drones to carry camera equipment, which is difficult to control under high wind conditions [3].

The development of mobile robot technology provides a new mobile platform for overhead power line inspection. The patrol robot that can climb the power line can work with electricity, and avoid obstacles such as dampers, suspension clamps, and insulators, which can improve the inspection condition[4]. To achieve full-time independent inspection, the detection of obstacles on the power line is the primary problem to be solved.

Vision-based power line obstacle detection has been extensively studied in the field of power line inspection robots. Le Huang, Gong-ping Wu[1] adopted adaptive homomorphic filtering method. This method can effectively deal with the influence of weak light, and the detection result is good, but the accuracy is still need to be improvement. Cai-shi Hu and Gong-ping Wu[5] used traditional image detection method to extract image primitives and applied structural constraints to realize the detection of obstacles. The algorithm is simple and fast. But because it is based on the geometry of obstacles, mainly circular features, and its structural constraints of the same power line, it is greatly affected by illumination and background environment, and the precision is not very high. Wen-ming Cao and Yao-nan Wang[6] proposed a method for extracting image edges using wavelet maximum value algorithm and calculating joint invariant moment features of obstacle image edges, which achieved good results. But this method is also sensitive to illumination and environment changes.

The current deep learning algorithm has made significant progress in the field of computer vision. This method effectively solves the influence of illumination, scale changes, and complex environments, and has high accuracy. In 2012, the Hinton team used the Alex-Net Deep Learning Network to win the championship in ImageNet[7], which led to a research boom in deep learning. Deep learning has two development lines in the image field. One is based on Object Proposals, called two stages method. Such as Fast R-CNN[8] and Faster R-CNN[9]. The other line is called one-stage detection method that embeds the generation of candidate boxes into the feature maps, so the detection speed is greatly improved. Typical

Supported by the Fundamental Research Funds for the Central Universities (no. VP21ZR1102Y17025) and Central University Basic Research Fund (no. 2682017ZDPY09, 2682019CX35, 2018GF02)

representative algorithms are YOLO[10], SSD[11]. At present, the method of deep learning has not been introduced into the field of obstacle detection for power line inspection robots. The traditional obstacle recognition algorithm has poor adaptability to illumination and background changes, and it is difficult to achieve multi-scale recognition. Although the YOLO and SSD algorithms are faster, the depths of the network are very deep and the parameters are huge. The computation is still large for embedded systems.

Based on the above problems, this paper simplifies the existing SSD model, reduces network complexity, the amount of parameters, and replaces the original VGG-16[12] network with fewer convolutional layers, making the algorithm more suitable for embedded systems.

II. POWER LINE ENVIRONMENT AND OBSTACLE DETECTION ALGORITHM

A. Power Line Environment

Power line environment is shown in Fig.1, (a) is a power line inspection robot. (b), (c), and (d) are main obstacles of the power transmission line, damper, insulator, and suspension clamp.

Figure 1. Inspection robot and Power line obstacle. (a) inspection robot (b) damper (c) suspension clamp (d) insulator

B. SSD Algorithm

SSD is an object detection algorithm proposed by Wei Liu. Its full name is Single Shot Multi-Box Detector. Single Shot indicates that the SSD algorithm belongs to the one-stage method. Multi-Box indicates that the SSD is multi-box prediction. SSD uses different feature maps to detect objects of different scales, and uses bounding boxes with different aspect ratios to detect objects[11] . The SSD network structure is shown in Figure 2.

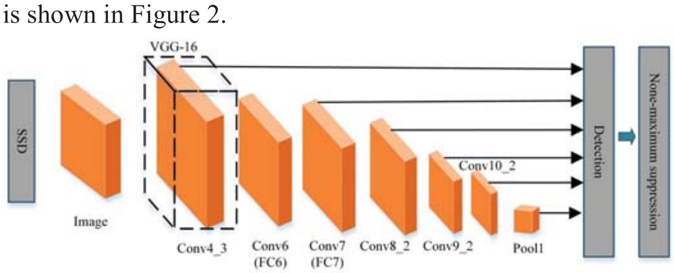

Figure 2. SSD algorithm structure

SSD algorithm converts the fully connected layer FC6 and FC7 to the convolution layer conv6 and conv7 of the base network VGG-16. At the same time, conv8, conv9, conv10, the three convolutional layer and pooling11, the pooling layer are added. Therefore, its network structure is still too complicated, and the parameters are too large, so it is difficult to apply to embedded systems.

C. Simplified SSD Algorithm

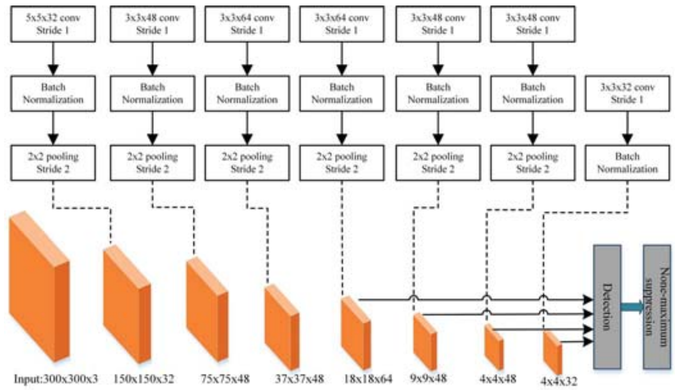

Figure 3. Simplified SSD algorithm structure

In order to improve the calculation speed of SSD, the original SSD model is simplified, as is shown in Figure 3. It is simplified into 7 blocks. The first 6 blocks are composed of convolution layers, Batch Normalization layers, and pooling layers. The last block is composed of convolution layer and Batch Normalization layer. The entire network has 7 convolution layers, 7 batch normalization layers, and 6 pooling layers. Each convolutional layer is followed by a Batch Normalization layer. Batch normalization is a training method proposed by Google[13], which can improve the data distribution and avoid gradient disappearance and gradient explosion. Batch Normalization is calculated as follows:

$$\frac{1}{m}\sum_{i=1}^{m} x_i = \mu_B \tag{1}$$

$$\frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 = \sigma_B^2 \tag{2}$$

$$\frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} = \hat{x}_i \tag{3}$$

$$\gamma \hat{x}_i + \beta = y_i \tag{4}$$

m is the number of samples; x_i is the sample point; μ_B is the sample mean; σ_B^2 is the sample variance; ε is a parameter, for preventing the denominator to be 0; γ and β are the parameters that need to be learned. In addition to avoiding gradient explosions and gradient disappearance, the Batch Normalization layer speeds up the learning of the model, while also prevent overfitting. Therefore, we removed the dropout layer after adding the Batch Normalization layer.

D. Loss Function Calculation

After the training sample is determined, the loss function is needed to calculate. The loss function is defined as the weighted sum of location loss and confidence loss[11].

$$L(x,c,l,g)=\frac{1}{N}(L_{conf}(x,c)+\alpha L_{loc}(x,l,g)) \quad (5)$$

N is the positive sample size of the priori boxes. $L_{conf}(x,c)$ is the classification loss, $L_{loc}(x,l,g)$ is the position loss, α is the weight coefficient, set α=1 here. For position loss $L_{loc}(x,l,g)$, it uses Smooth L1 loss, defined as follows:

$$L_{loc}(x,l,g)=\sum_{i\in Pos,m\in\{cx,cy,w,h\}}^{N}\sum x_{ij}^{k} smooth_{L_1}(l_i^m-\hat{g}_j^m) \quad (6)$$

$$smooth_{L_1}(x)=\begin{cases}0.5x^2 & when\,|x|<1\\ |x|=0.5 & others\end{cases} \quad (7)$$

N is the positive sample size of the priori boxes, $x_{ij}^k\in\{0,1\}$ is an indicator parameter, x_{ij}^k=1 indicating that the i priori box matches the j ground truth, and the ground truth category is k . $\hat{g}_j^m$ is the actual box position parameter after encoding, l_j^m indicates the predicted value of the a priori box.

Classification confidence $L_{conf}(x,c)$ uses the cross-entropy loss function[14] :

$$L_{conf}(x,c)=-\sum_{i\in Pos}^{N} x_{ij}^p\log(\hat{c}_i^p)-\sum_{i\in Neg}\log(\hat{c}_i^0) \quad (8)$$

$\hat{c}_i^0$ is the probability that category is the background and classification is correct. $\hat{c}_i^p$ is the probability value calculated by the Softmax function, and the calculation formula is:

$$\hat{c}_i^p=\frac{\exp(c_i^p)}{\sum_p\exp(c_i^p)} \quad (9)$$

III. Comparison of Complexity between simplified SSD and SSD

The number of operations of the model can be measured by FLOPS (floating-point Operations). The calculation formula of the complexity of the convolutional neural network is as follows[15]:

$$Time\sim O(\sum_{l=1}^{D}M_l^2\bullet K_l^2\bullet C_{l-1}\bullet C_l) \quad (10)$$

$$Space\sim O(\sum_{l=1}^{D}K_l^2\bullet C_{l-1}\bullet C_l+\sum_{l=1}^{D}M_l^2\bullet C_l) \quad (11)$$

$Time$ represents the time complexity, D is the number of layers of convolutional neural network, l is lth convolutional layer, C_l is the number of output channels of the l th convolutional layer, M_l is the length of the lth convolutional layer feature map. Here we compare the spatial complexity of the SSD model and the simplified SSD model, which mainly compares the weight parameters. The pooling layer has no weight parameters, and the Batch Normalization layer has only two parameters. Therefore, the weight parameters are mainly concentrated in the convolutional layer. The parameter statistics of the SSD model and the simplified SSD are shown in Table Ⅰ, and Table Ⅱ:

TABLE I. SSD PARAMETER STATISTICS

net name	*kernel*	*stride*	*channels*	*Parameter numbers*
Conv1_1	3x3	1	64	3*3*3*64=1728
Conv1_2	3x3	1	64	3*3*64*64=36864
Conv2_1	3x3	1	128	3*3*64*128=73728
Conv2_2	3x3	1	128	3*3*128*128=147456
Conv3_1	3x3	1	256	3*3*128*256=294912
Conv3_2	3x3	1	256	3*3*256*256=589824
Conv3_3	3x3	1	256	3*3*256*256=589824
Conv4_1	3x3	1	512	3*3*256*512=1179648
Conv4_2	3x3	1	512	3*3*512*512=2359296
Conv4_3	3x3	1	512	3*3*512*512=2359296
Conv5_1	3x3	1	512	3*3*512*512=2359296
Conv5_2	3x3	1	512	3*3*512*512=2359296
Conv5_3	3x3	1	512	3*3*512*512=2359296
Conv6	3x3	1	1024	3*3*512*1024=4718592
Conv7	1x1	1	1024	1*1*1024*1024=1048576
Conv8_1	1x1	1	256	1*1*1024*256=262144
Conv8_2	3x3	1	256	3*3*256*256=589824
Conv9_1	1x1	1	128	1*1*256*128=32768
Conv9_2	3x3	1	256	3*3*128*256=294912
Conv10_1	1x1	1	128	1*1*256*128=32768
Conv10_2	3x3	1	256	3*3*128*256=294912
Conv11_1	1x1	1	128	1*1*256*128=32768
Conv11_2	3x3	1	256	3*3*128*256=294912
total				21722816

TABLE II. SIMPLIFIED SSD PARAMETER STATISTICS

net name	*kernel*	*stride*	*channels*	*Parameter numbers*
Conv1	5x5	1	32	5*5*3*32=2400
Conv2	3x3	1	48	3*3*32*48=13824
Conv3	3x3	1	64	3*3*48*64=27648
Conv4	3x3	1	64	3*3*64*48=36864
Conv5	3x3	1	48	3*3*512*512=27648
Conv6	3x3	1	48	3*3*48*48=20736
Conv7	3x3	1	32	3*3*48*32=13824
total				142944

It can be seen from the above two tables that the parameter quantity of the simplified SSD model is only 0.66% (142944/21722816=0.66%) of the original model. The parameter quantity is greatly reduced.

IV. Experiment Analysis

A. Experimental Platform and Data Set

The experimental platform uses Jetson TX2, which is equipped with 8G memory and uses a Pascal-based GPU. Jetson is equipped with Ubuntu 18.04, and we use python and keras for programming. The data set uses the power line platform built by the laboratory to produce 400 training sets, 160 validation sets and 150 test sets for the damper, the suspension clamp and the insulator.

B. Network Training

The SSD and the simplified SSD algorithm were trained with an initial learning rate setting to 0.01 and an attenuation coefficient setting to 0.0005. At the same time, the data set is expanded by means of image rotation, increasing contrast, and increasing noise. The training loss value of SSD and simplified SSD algorithm is as shown in Figure 4:

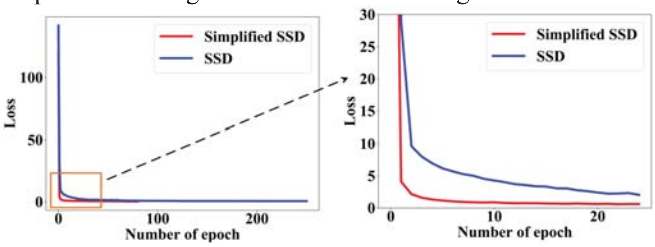

Figure 4. Simplified SSD and SSD loss value graph.

After 80 epochs, each epoch has 500 iterations, that is, a total of 125,000 iterations, the loss value of the simplified SSD drops to 0.25, while the SSD experiences 250 epochs, and each epoch has 500 iterations, and the loss value drops to 0.32. Thus, it can be seen that the simplified SSD is much faster than the SSD model training.

C. Network Accuracy Analysis

For multi-object recognition, the Mean Average Precision (mAP)[16] is used as the evaluation model. The correctly identified positive samples are called True positives. the correctly identified negative samples are called True negatives. the incorrectly identified positive samples are called False positives, and the incorrectly identified negative samples are called False negatives. The precision is defined by the following formula:

$$precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{12}$$

The recall is defined as follow:

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{13}$$

With the precision and recall rate, the Precision-Recall (P-R) curve can be drawn. The area under the curve is the AP value. The specific calculation formula is as follow:

$$AP = \int_0^1 p(r)dr \tag{14}$$

The ratio of the prediction box to the real box (IoU) > 0.5 is as the threshold value to determine the category confidence of prediction box.

Of the 150 photos in the test set, there were tally 310 obstacles of 3 categories. The precision-recall rate (P-R) calculated by equations (12) and (13) is shown in Figure 5:

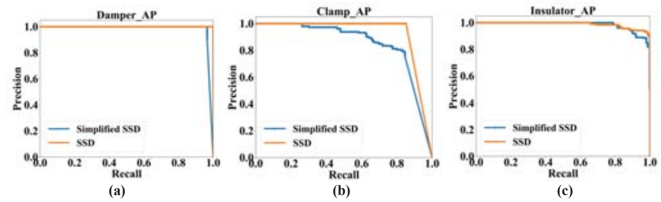

Figure 5. P-R curve of three obstacles. (a) Damper (b) Clamp (c) Insulator

The area under the curve in Fig. 5 is the AP value. The AP value, mAP value of the obstacles of the SSD algorithm and the simplified SSD algorithm are calculated and shown in Table III. The model sizes and detection speed of the two algorithms are also shown in Table III:

TABLE III. AP MAP MODEL SIZE AND FRAME RATE STATISTICS

Method	AP/%			mAP/%	Model Size	Speed
	Damper	*Clamp*	*Insulator*			
SSD	100	85.26	98.77	94.7	301Mb	4.5fps
Simplified SSD	96.55	79.59	98.40	91.51	2.5Mb	15fps

As can be seen from Table III, the mAp of the simplified SSD model is not much lower than that of the SSD, but the parameter amount has dropped a lot. The frame rate test on the Jetson TX2 has increased from 4.5FPS of SSD to 15FPS of simplified SSD.

The detection results of simplified SSD and SSD on the test set are shown in Figure 5, (a) is the detection results of Simplified SSD, (b) is the detection result of SSD, both algorithms correctly detect the dampers, suspension clamps and insulators.

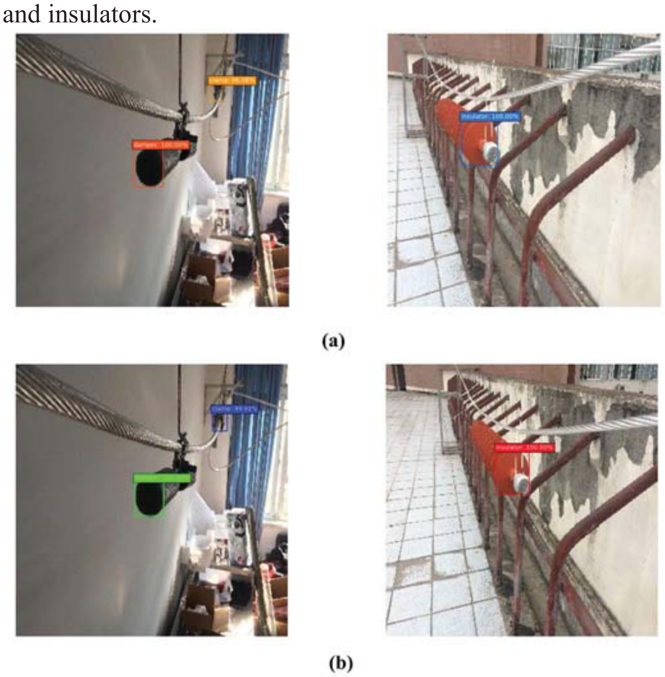

Figure 6. Results of simplified SSD and SSD on test set. (a) Simplified SSD (b) SSD

V. Conclusion

In this paper, a simplified SSD algorithm is proposed. It reduces the convolutional layer of SSD from 23 layers to 7 layers, which greatly reduces the parameter quantities and reduces the model size from 301Mb to 2.5Mb. The Batch Normalization layer has also been added after each convolution layer, effectively avoiding gradient disappearance and gradient explosion. While the model is lightweight, the mAP is not reduced too much, and it can basically be used on the embedded JETSON TX2. Moreover, the algorithm realizes the recognition stability of obstacles in complex environments under multi-scale, and has certain significance for the obstacle recognition of power transmission line. Although the parameter has dropped a lot in the algorithm, it still does not fully realize real-time performance. In the future, we will optimize the network structure further and promote the calculation speed more.

References

[1] J. Katrasnik, F. Pernus, and B. Likar, "A survey of mobile robots for distribution power line inspection," IEEE Transactions on Power Delivery, vol. 25, pp. 485-493, January 2010.

[2] L. Huang, G. P. Wu, and X. H. Ye, "Obstacle identification under low-light conditions of transmission line inspection robot," Acta Optica Sinica, vol. 38, 2018, pp. 291-299.

[3] B. Avidar, "Electronic airborne inspection method for overhead transmission power-lines," Proceedings of the 6th International Conference on Transmission &Distribution Construction & Live Line Maintenance, pp. 12-17, September 1993.

[4] Y. C. Zhang, Z. Z. Liang, and M. Tan, "Mobile Robot for Overhead Powerline Inspection-a Review," ROBOT, vol. 26, 2004, pp. 467-473.

[5] C. S. Hu, G. P. Wu, H. Cao, and G. W. Jiang, "Research of Obstacle Recognition Based on Vision for High Voltage Transmission Line Inspection Robot," China Journal of Sensors and Actuators, vol. 21, 2008, pp. 2092-2096.

[6] W. M. Cao, Y. N. Wang, F. Yin, X. Wu, and S. Miao, "Research on obstacle recognition based on vision for deicing robot on high voltage transmission line," China Journal of Sensors and Actuators, vol.32, 2011, pp. 2049-2056.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al, "ImageNet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, 2015, pp. 211-252.

[8] R. Girshick, "Fast R-CNN," IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 1440-1448.

[9] S. Ren, K. He, R. Girshick, J, Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, pp. 1137-1149.

[10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016, pp. 779-788.

[11] W. Liu, D, Anguelov, D. Erhan, S. Dumitru, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, "SSD: Single Shot MultiBox Detector," European Conference on Computer Vision. Amsterdam, Springer International Publishing, 2016, pp. 21-37.

[12] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015 International Conference on learning Representations. Lille: University of Oxford, 2015, pp. 1-14.

[13] S. Ioffe, C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 32nd International Conference on Machine Learning, ICML 2015. Lille, France: ICML, 2015, pp. 448-456.

[14] P. T. De Boer, D. P. Kroese, and S. Mannor, "A tutorial on the cross-entropy method," Annals of Operations Research, vol. 134, 2005, pp. 19-67.

[15] K. M. He, J. Sun, "Convolutional neural networks at constrained time cost," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, pp. 5353-5360, June 7-12, 2015.

[16] M. Everingham, L. V. Gool, and C. K. I. Williams, "The pascal visual object classes (voc) challenge," International Journal of Computer Vision, vol.88, pp. 303-338, 2010.