Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 15, No. 3, 2024

Research on energy consumption optimization of RetinaNet model based on lightweight improvement in edge intelligent terminals in power system

Yi Mu*

School of Electrical and Electronic Engineering, North China Electric Power University, Baoding, 071003, China
* **Corresponding author Email:** muyi354464107@163.com

Abstract: With the increasing application of edge computing in power systems, intelligent terminal devices are facing new challenges in processing capabilities and energy optimization. As a highly efficient target detection model, RetinaNet's potential application in edge intelligent terminals of power systems has not been fully explored. This study proposes a lightweight model based on the convolutional neural network RetinaNet. Multiple lightweight models are used to replace the original backbone network ResNet for comparison. The best model is selected while ensuring accuracy. Redundant connections in the model network are pruned through channel pruning to reduce model size and improve detection speed. The results show that compared with the original algorithm, the RetinaNet model proposed in this study reduces parameter count by 73%, decreases computational load by 41.8%, reduces model size by 72.7%, and only decreases the mean average precision (mAP) value by 1.8 percentage points.

Keywords: Edge computing; RetinaNet; Structural design; Model pruning.

1. Introduction

In March 2020, State Grid Corporation of China established the strategic goal of building a "Chinese-characteristic, internationally leading energy internet company" as its long-term development objective, and accelerated the construction of smart grids and power IoT. With the advancement of power IoT and digital transformation of energy, numerous electrical quantity sensors, status sensors, and intelligent video monitoring systems, as well as other power edge devices, have joined the power IoT, generating a large amount of heterogeneous data[2-3].

2. Literature Review

Currently, both domestically and internationally, electric edge intelligent terminals are widely used to monitor and control the operating status of the power grid. Reference designed an Internet of Things and PLC programmable logic controller system, which utilizes sensors to collect information from various regions and manages and transmits the data intensively. Reference designed an embedded video surveillance system based on SOPC technology, and testing shows that this system can basically meet the visual management requirements of power grid construction. Reference designed an intelligent power terminal, which can collect and monitor the power parameters and power quality in building power distribution, upload the data to the upper computer of the terminal through power line communication, and also send warning messages to the mobile terminal. Reference proposed a temperature monitoring system for energy storage battery compartments, which can realize comprehensive monitoring of multiple temperature-sensitive points through real-time wireless data transmission. The electric edge terminals mentioned above typically use traditional methods, which mainly have the functions of data collection and transmission control but lack the ability to perform real-time intelligent perception and analysis at the edge of the power IoT. The traditional centralized data processing model based on cloud computing already lacks real-time performance. Therefore, constructing electric edge intelligent terminals with the capability of intelligent perception and analysis at the edge has become an urgent demand and inevitable trend in the development of digital power grids. Target detection is an unavoidable topic in this process. Through target detection, intelligent terminals can perform preliminary processing on inspection images, greatly relieving the pressure on the cloud. Currently, the use of aircraft for line inspection has been widely adopted as a routine inspection method. Through target detection technology, automated analysis and processing of aerial photography images and videos can be carried out, thereby achieving automatic positioning of potential faults in transmission line fittings. High-complexity deep learning target detection models have high detection accuracy, but due to the limited resources of electric edge intelligent terminals, their speed may be affected and they are usually difficult to deploy. Therefore, researching and developing a detection model suitable for electric edge intelligent terminals that balances accuracy and lightweight has significant practical application implications.

This paper combines the actual needs of target detection in the power IoT and conducts research on fitting detection algorithms. In response to the high accuracy and real-time requirements in the power IoT, this paper proposes a lightweight target detection algorithm designed specifically for deployment on electric edge intelligent terminals based on improvements to the RetinaNet network. The RetinaNet network consists mainly of four parts: the backbone network, detection head, sample allocation strategy, and loss function. The lightweight improved model designed in this paper for the power grid mainly focuses on improving the backbone network of the RetinaNet network, as well as model size,

parameter quantity, computational load, and complexity. The RetinaNet network is a single-stage target detection model that maintains both fast detection speed and high detection accuracy.

3. Method

3.1. Materials and Equipment

The failure of transmission lines is a major cause of large-scale power outages in the power grid. Maintaining and operating these lines is crucial to ensuring the safety and stability of the power system. Various metallic accessories made of iron or aluminum, commonly referred to as fittings, are widely used on transmission lines. These fittings serve the primary functions of supporting, fixing, and connecting bare conductors, conductors, insulators, and other components.

Currently, there are many publicly available image datasets related to power IoT devices, with a significant proportion being aerial datasets. Most of these public datasets are used for transmission line fault or defect detection, with few datasets focused on fitting detection. Therefore, to detect fittings under limited conditions, it is necessary to first locate a clear and annotated dataset of fittings.

3.2. Experimental datasets

The experimental dataset used is the transmission line hardware dataset. The hardware includes insulators, bolted pins, anti-vibration hammers, and tension clamps. To ensure the comprehensiveness and accuracy of the data during the creation of the dataset, the hardware was placed on blue and white mats respectively, photographed at different heights, and due to the influence of light, multiple angles were captured. Under the illumination of light, the hardware exhibited different shadows, as shown in the original images in Figures 1 and 2. The captured images were annotated using the LabelImg annotation software.

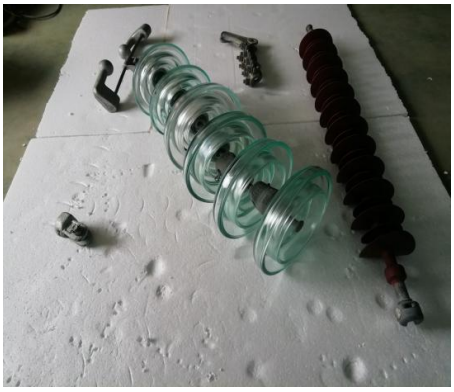

Figure 1. Dataset picture example1

Figure 2. Dataset picture example2

3.3. RetinaNet Object Detection Network Improvements

3.3.1. RetinaNet algorithm structure

The RetinaNet network model structure mainly consists of the backbone network part (Backbone), the feature pyramid part (FPN), and the Head module, as shown in Figure 3.

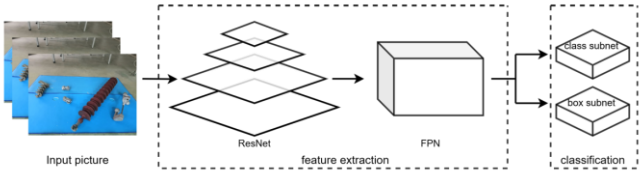

Figure 3. Structure diagram of the RetinaNet network model

The core responsibility of the backbone network is feature extraction, while the feature pyramid takes on the task of feature fusion. In addition, the classification module functions to detect and identify features processed as mentioned earlier. The core architecture of the RetinaNet model adopts the Deep Residual Networks (ResNet), a concept proposed by Kaiming He et al. in 2016. The main purpose is to simplify the structure of convolutional neural networks and address the potential issues of gradient vanishing or exploding as the network depth increases, which may result in decreased model performance. In this study, the concept of residual modules was introduced for the first time, assuming the existence of an optimal solution mapping $H(x)$ and proposing to fit a residual mapping, namely $F(x) = H(x) + x$. This approach is implemented by adding shortcut connections in the convolutional feedforward network, as illustrated in Figure 4.

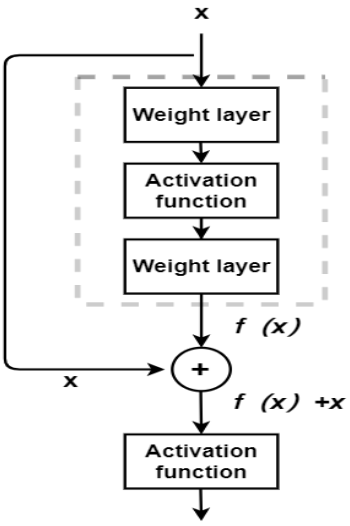

Fig.4 Structure diagram of the Residual Block

By directly integrating the mapping result x of the network itself into the output of the superimposed layer, this process not only avoids adding extra network parameters but also maintains the stability of computational complexity.It significantly enhances the network's capability to represent image features.

RetinaNet is the first model to introduce Focal Loss as its classification loss function. The proposal of Focal Loss originates from the issue of sample imbalance in object

detection tasks in the field of image processing. The imbalance mentioned here is distinct from the usual understanding, as it also emphasizes the difficulty of samples. Focal loss is based on binary cross-entropy (CE). It is a dynamically scaled cross-entropy loss that, through a dynamic scaling factor, can dynamically reduce the weights of easily distinguishable samples during the training process, thus quickly focusing the attention on those difficult-to-distinguish samples.

Cross Entropy Loss: Cross-entropy loss based on binary classification, which takes the form shown in Eq. (1).

$$CE(p,y) = \begin{cases} -\log(p) & if\ y=1 \\ -\log(1-p) & otherwise \end{cases} \quad (1)$$

Where y takes values {1, -1}, representing positive and negative samples respectively, p denotes the probability of the model's predicted label, with a range from 0 to 1. Usually, when p > 0.5, it is classified as a positive sample, otherwise as a negative sample. For the sake of illustration, redefining p_t as shown in equation (2).

$$p_t = \begin{cases} p & if\ y=1 \\ 1-p & otherwise \end{cases} \quad (2)$$

By synthesizing the aforementioned equations, we can der ive a simplified formula as shown in Equation (3).

$$CE(p,y) = CE(pt) = -\log(pt) \quad (3)$$

Introduce a modulation factor to focus on challenging samples, as shown in Equation (4).

$$FL(p_t) = -(1-p_t)^{\gamma}\log(p_t) \quad (4)$$

γ is a parameter that ranges from [0,5]. When γ equals 0, it transforms into the original CE loss function. The term $(1-p_t)^{\gamma}$ reduces the loss contribution of easily distinguishable samples, thereby increasing the loss ratio for hard-to-distinguish samples. As p_t approaches 1, this indicates that the sample is easily distinguishable; at this point, the modulation factor $(1-p_t)^{\gamma}$ approaches 0, suggesting a minimal contribution to the loss and thus reducing the loss ratio for easily recognizable samples. The effects of different values of γ on loss performance are illustrated in Figure 5.

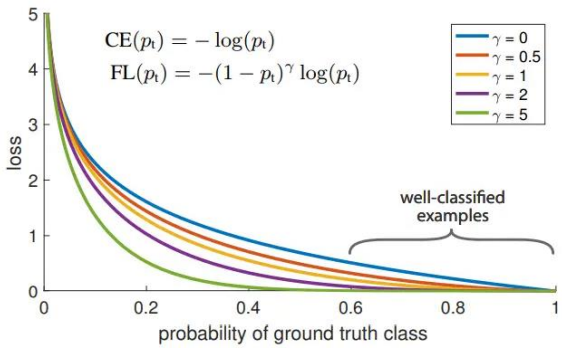

Figure 5. The classification performance of CE and FL for different probability distributions

It is evident that as p_t increases, the distribution of easily distinguishable samples improves significantly, leading to a smaller corresponding loss. Through the aforementioned balance between positive and negative samples, as well as easy and hard samples, we can derive the final form of Focal loss as shown in equation (5).

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t) \quad (5)$$

By using α_t, one can suppress the imbalance in the number of positive and negative samples. Meanwhile, 'γ' allows control over the imbalance in the quantity of easily versus difficultly distinguishable samples.

3.3.2. Improvements of Backbone Networks

Traditional convolutional neural networks (CNNs) theoretically exhibit greater expressive power with increased depth. However, once CNNs reach a certain depth, further increasing layers does not enhance classification performance. Instead, it leads to slower convergence and decreased accuracy. Even augmenting the dataset to address overfitting fails to improve classification performance and accuracy. ResNet50 tackles this issue using residual networks, but deployment on resource-constrained platforms proves challenging. Therefore, this paper focuses on enhancing the backbone through lighter models, such as ShuffleNetV2 and MobileNetV3.

The ShuffleNetV2 model represents a lightweight convolutional neural network architecture. Its main features include the use of grouped convolutions and channel shuffle operations to reduce computational load and parameter count, thereby improving inference speed and efficiency. The structure of ShuffleNetV2 comprises an input layer, multiple residual blocks, and an output layer. Each residual block consists of a 1x1 convolution, a 3x3 depthwise separable convolution, another 1x1 convolution, and a channel shuffle operation. By stacking several such residual blocks, one can construct a deep neural network with multiple layers. ShuffleNetV2 achieves a faster inference speed and lower computational complexity while maintaining high accuracy, outperforming other lightweight architectures. The basic module of ShuffleNetV2 is illustrated in Figure 6.

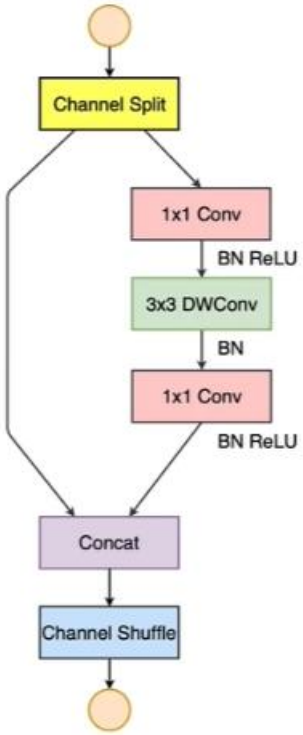

Figure 6. Basic Module of ShuffleNetV2

MobileNetV3 is also a lightweight convolutional neural network that employs a technique called "Depthwise Separable Convolution." This technique breaks traditional convolution operations into two steps: depth convolution and pointwise convolution. Such decomposition reduces computational load and parameter count, enhancing the model's inference speed and efficiency. MobileNetV3 introduces a new network structure known as the "Inverted Residual Block." This structure increases network depth by utilizing smaller input feature maps and more layers, thereby further boosting performance. Additionally, MobileNetV3 employs an adaptive Expansion Factor to automatically adjust the network's width and depth based on the size of the input data. The MobileNetV3 architecture is shown in Figure 7.

Both lightweight networks excel in resource-constrained scenarios like mobile and embedded devices, finding widespread application in computer vision tasks such as image classification, object detection, and semantic segmentation. The goal of this study is to propose a target detection model that is low-power, low-parameter, low-complexity, and preserves accuracy, enabling deployment on edge intelligent terminals or resource-constrained mobile devices. Therefore, the original backbone network of RetinaNet, ResNet50, is replaced with ShuffleNetV2 and MobileNetV3, demonstrating the effectiveness and reliability of the replacements through experiments.

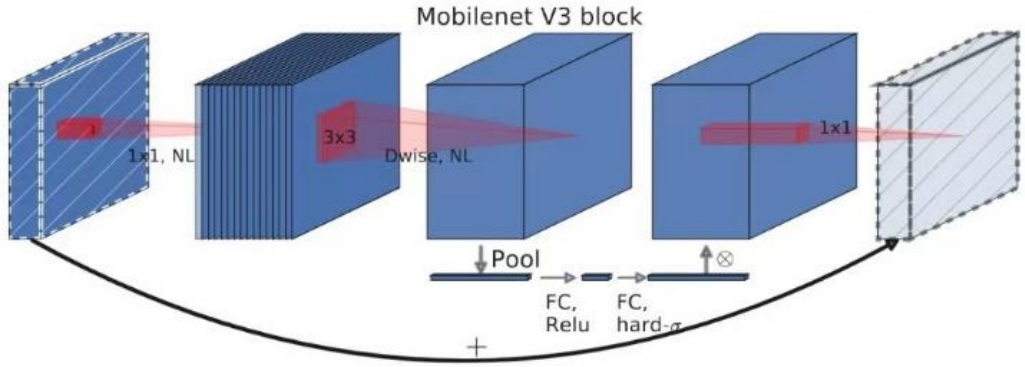

Figure 7. MobileNetV3 Network Architecture

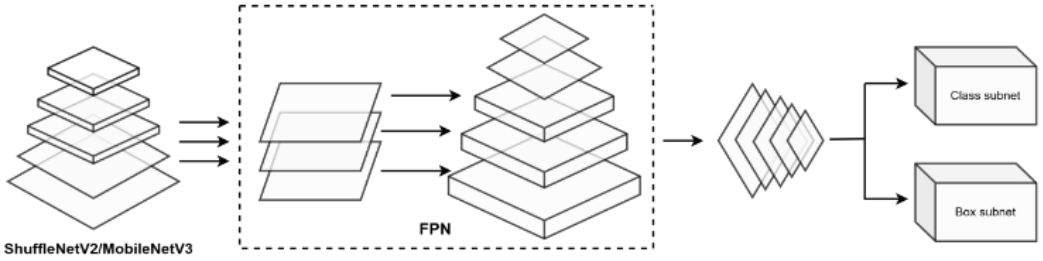

Figure 8. Improved Structure of the RetinaNet Model

3.3.3. Channel Pruning

This study simplified the model's network structure. However, the model still contains a significant amount of redundant information. This redundancy reduces detection speed and limits deployment on resource-constrained devices. To optimize the model while maintaining accuracy, it is essential to prune unnecessary parameters in the network. This reduction will enhance detection speed [17-19].

Pruning methods mainly fall into two categories: structured pruning and unstructured pruning. The core idea of unstructured pruning is to eliminate individual weights in the network. This approach decreases the model's storage and computational requirements. Although it does not alter the overall structure of the network, it produces a sparse matrix. Efficient computation of sparse matrices typically requires specialized software libraries or hardware support. A typical algorithm for structured pruning is channel pruning. Its primary principle involves removing certain convolutional layer channels to reduce the number of feature maps, thus decreasing computational load and memory usage. Compared to unstructured pruning techniques, channel pruning significantly lowers the computational resources needed for convolution operations, greatly boosting processing speed. Moreover, the pruned network architecture not only has lower memory and storage requirements but is also more suitable for deployment in embedded systems and mobile devices. This study conducted channel pruning using a standard pruning method with a sparsity rate of 0.

4. Results and Discussions

4.1. Experimental Setup

4.1.1. Dataset and Data Augmentation

This experiment utilized a dataset of transmission line fittings. We annotated it using LabelImg, placing blue and white pads and capturing images from multiple angles, resulting in a total of 2,511 images, which include insulators, bolts, anti-vibration hammers, and tension clamps, as detailed in Section 3.2 on the dataset introduction.

Data augmentation involves applying a series of transformations or processing techniques to the original data during training to generate new training samples. This technique enhances the model's generalization ability and robustness by transforming and processing the original dataset. It creates more samples, increasing the diversity of data, which improves adaptability to various environments and scenarios, thereby reducing the risk of model overfitting on the training set. Data augmentation can also introduce

noise or interference, pushing the model to focus more on feature extraction rather than specific details, enhancing its generalization capacity. Datasets may present various changes and anomalies, such as differing light conditions, angles, and scales. Through data augmentation, we can simulate these variations and anomalies, enabling the model to adapt better. For images in the dataset, we employed horizontal flipping as the data augmentation method. Subsequently, we partitioned this dataset into training and validation sets in a 70:30 ratio. The improvements and comparative experiments discussed later were conducted using this dataset.

4.1.2. Experimental Environment and Evaluation Metrics

The operating system of the experimental platform is Ubuntu 20.04. The CPU model is InterCore i9-12600k at 3.2GHz. The total RAM is 32GB, and the GPU model is NVIDIA GeForce RTX 4070, with a memory capacity of 16GB. The training utilizes the PyTorch 2.0.3 framework with CUDA 12.1. During model training, input images are uniformly resized to 640x640, with a batch size of 4 and a worker count of 8. The training spans 40 epochs, starting with a learning rate of 0.01 and employing Stochastic Gradient Descent (SGD) as the optimization algorithm. This paper employs Mean Average Precision (mAP) as the evaluation metric for detection accuracy of the algorithm model. mAP is one of the most common evaluation metrics to measure model performance in object detection. It uses the P-R curve to illustrate the relationship between precision and recall, calculated as follows:

(1) The average precision (AP) calculates the area under the curve, expressed in formula (6), where P represents precision (Pre) and R denotes recall (Rec).

(2) Compute the AP values for all categories and average them to obtain the final mean average precision (mAP), represented in formula (7), where C indicates the number of categories.

$$AP = \int_0^1 P(R)\,dR = \sum_{k=0}^{n} P(k)\,\Delta R(k) \tag{6}$$

$$mAP = \frac{1}{C}\sum_{c \in C} AP(c) \tag{7}$$

Additionally, using parameters (Params) and Multiply-Accumulate Operations (MACs) assesses model complexity and computational efficiency. Fewer Params indicate lower computational and storage demands, making them suitable for resource-limited environments. The number of MACs determines the computational load required by the network, specifically how many multiplications and additions the network must perform. A higher MAC count implies a greater computational burden, requiring more resources and time for training and inference, consequently increasing energy consumption due to the additional computations needed.

4.2. Analysis of Results Based on Lightweight Improved RetinaNet Model

4.2.1. Comparison of Performance across Different Backbone Network Models

Research shows that as the depth of deep neural networks increases, the performance of recognition models improves, leading to higher detection accuracy . However, greater depth does not always guarantee better outcomes; alternative conclusions exist [25-27]. This variation arises because factors such as dataset, network depth, network structure, and parameter settings also significantly impact model performance. This study substitutes the backbone network employed by the original RetinaNet and investigates the influence of various backbone networks on the model's performance, including ShuffleNetV2, MobileNetV3, and the original ResNet50. The experimental results are presented in Table 1.

Table 1. Detection Results of Different Backbone Networks

Backbone Networks	mAP@0.5/%	mAP@0.5-0.95/%
ResNet50	71.07	39.18
ShuffleNetV2	52.08	22.95
MobileNetV3	71.99	41.42

4.2.2. Analysis of Channel Pruning Experimental Results

From section 4.2.1, it is clear that after replacing the backbone, the model with MobileNetV3 as the backbone achieves the highest accuracy. Furthermore, MobileNetV3 is inherently a lightweight model, thus pruning other models holds little significance. To pursue lightweight design, this study employs the torch-pruning library to conduct channel pruning on the model after replacing it with MobileNetV3, setting the sparsity rate to 0 and the pruning rate to 50%. The results of this pruning are shown in Table 2.

Table 2. Comparison of RetinaNet-MobileNetV3 Model Before and After Pruning

RetinaNet-MobileNetV3	Params/M	MACs/G	mAP@0.5%
Pruning Before	11.16	77.19	71.99
After pruning	8.82	74.98	69.23

The figure demonstrates that the model, after channel pruning using the torch-pruning library, eliminated redundant channels from the original lightweight model, resulting in a model with fewer parameters and reduced computational load. Consequently, this model can operate faster and consume less computational resources. To better illustrate the lightweight nature and performance of the processed model, we compared our optimized lightweight model with the original unmodified model while keeping the experimental conditions constant. The results of this comparison are presented in Table 3.

Table 3. Comparison of the Original Model and the Pruned Model

Model	Params/M	MACs/G	mAP@0.5%	Model Size/MB
RetinaNet	32.33	128.94	71.07	247
Pruned-RetinaNet-MobileNetV3	8.82	74.98	69.23	67.5

The comparative experimental results above were calculated based on images sized 800*1333. As shown in Table 3, when compared to the unmodified RetinaNet model, the proposed lightweight improved model reduced parameter count by 23.51M, which is about 73% of the original MB model's parameters. The computation load decreased by 53.96G, representing 41.8% of the original model's workload. Additionally, the model size reduced by 179.5MB, approximately 72.7% of the original model’s size, while the mAP value only declined by 1.84%. The significant reductions in computation load, parameter count, and model size led to a major decrease in power consumption. Therefore, this model is particularly suitable for resource-constrained edge intelligent terminals, effectively demonstrating the improvements made to the model.

4.2.3. Comparison of Performance with Other Similar Grid Models

Maintaining a consistent experimental setup, this study compares its model against widely used target detection algorithms for grid applications. The comparison results are presented in Table 4.

Table 4. Comparison of Performance with Other Similar Models

Model	mAP@0.5/%	mAP@0.5-0.95/%	Params/M	Model Size /M
Faster R-CNN	68.88	35.69	41.58	318.5
SSD	41.44	19.84	13.81	105.9
The Present Research	69.23	39.1	8.82	67.5

Among them, the SSD model in Table 4 is implemented through ResNet50. Compared with classic object detection algorithms such as SSD and Faster R-CNN, the improved RetinaNet model proposed in this paper demonstrates significant advantages. These advantages are not only reflected in the increase of mean average precision (mAP) value, but also in the reduction of parameters (Params) and the decrease in model size. Furthermore, while maintaining high accuracy, this model also achieves lightweight, effectively balancing the relationship between model performance and resource consumption, once again demonstrating the effectiveness of model improvements. Overall, this lightweight RetinaNet improved model is particularly suitable for deployment on endpoint detection devices with limited computing capabilities or constrained resources due to its small size, low computational requirements, and high real-time characteristics.

5. Conclusion

This study combines the issues of target detection technology and limited resources of intelligent edge terminals in the power industry, proposing a lightweight improved model based on the RetinaNet model. By replacing the backbone network of the original RetinaNet model multiple times, lightweight models such as ShuffleNetV2 and MobileNetV3 were selected as replacement networks. After the replacement, an evaluation of performance metrics was conducted on the transmission line hardware dataset, identifying the most suitable RetinaNet model for the power grid. This model was then pruned using the torch-pruning library to remove a significant amount of redundant channels, greatly reducing the model's parameters, computational load, and size while maintaining accuracy, thus lowering the model's power consumption during operation. This outcome demonstrates that the proposed model in this study exhibits high real-time performance, low power consumption, and suitability for the power grid. It can be deployed on edge detection devices with limited computing capabilities or restricted resources, significantly alleviating the pressure on cloud services and providing reliable technical support for practical monitoring and management of power transmission lines. In future research, the author plans to deepen and refine existing algorithms, explore more cutting-edge deep learning technologies, and enhance the intelligence level of power edge intelligent devices by integrating advanced technologies such as edge computing and the Internet of Things. Through these efforts, the author hopes to make a greater contribution to the development of smart grids and power IoT.

References

[1] Cui Hengzhi, Jiang Chengling, Miao Weiwei, et al. Design and implementation of intelligent electric power IoT system based on edge computing[J].Electric power information and communication technology,2020, 18(4): 33-41. CUI Hengzhi, JIANG Chengling, MIAO Weiwei, et al. Design and implementation of power intelligent IoT system based on edge computing[J]. Electric Power Information and Communication Technology, 2020, 18(4): 33-41(in Chinese).

[2] SUN Haoyang, ZHANG Jichuan, WANG Peng, et al. Edge computing technology for power distribution Internet of Things[J].Power Grid Technology, 2019, 43(12): 4314-4321. SUN Haoyang, ZHANG Jichuan, WANG Peng, et al. Edge computation technology based on distribution internet of things[J]. Power System Technology, 2019, 43(12): 4314-4321(in Chinese).

[3] QIU Shuguang,PANG Chengxin,JIA Jia. Research on the application of LPWAN and edge computing fusion in power Internet of Things[J].Internet of Things technology,2019, 9(7): 63-66.

[4] LI Yuke. Design and application of wireless power monitoring and control system based on Internet of Things and PLC[J]. Electrical technology and economy,,2024,(06):376-379.

[5] Duan Jian,Wang Xinchao,He Xiaoyang. Power grid visualization construction based on embedded video surveillance technology [J]. Information technology, 2019, 43(11):169-172+176.DOI:10.13274/j.cnki.hdzj.2019.11.035.

[6] YANG Caiwei. Design of intelligent power terminal[D]. Changchun Institute of Technology, 2023. DOI: 10.27834 / d.cnki.ggccc.2023.000110.

[7] Gu Jian,Xin Zhanqiang,Luo Zhixiong. Application of temperature monitoring system in energy storage battery compartment[J]. Mass electricity,2024,39(01):44-45.

[8] LI Xin, LAI Ji, CHEN Zhongtao, et al. Intelligent fault detection system for state grid IoT equipment based on edge computing[C]//2019 IEEE 3rd International Electrical and Energy Conference (CIEEC). Beijing: IEEE, 2019 : 1434-1439.

[9] Qi Yincheng, Jiang Aixue, Zhao Zhenbing, et al. Transmission line inspection image fitting detection method based on improved SSD model[J].electrical measurement and instrumentation,2019,56(22):7-12+43. DOI:10.19753/j.issn1001-1390.2019.022.002.

[10] PENG Ziyang, CHEN Nuotian, YI Junfei, et al. Transmission line power devices and abnormal target detection based on improved RetinaNet algorithm[J].Hunan Electric Power, 2023, 43(05):79-84.

[11] ZHAO Qiang. Theory and application of transmission line fittings[M].Beijing: China Electric Power Press, 2013. 2-3.

[12] ZHOU Chunxin,HUO Yizhi,DU Youhai,JIANG Minlan, ZENG Lingguo, ZHANG Changjiang, SHI Xiaowei. Nondestructive testing of soybean appearance quality based on improved RetinaNet[J/OL]. Chinese Journal of Cereals and Oils..

[13] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 770-778

[14] Lin T Y , Goyal P , Girshick R ,et al.Focal Loss for Dense Object Detection[J].IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, PP(99):2999-3007. DOI: 10.1109 / TPAMI.2018.2858826.

[15] Ma N , Zhang X , Zheng H T ,et al.ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design[C]//European Conference on Computer Vision. Springer, Cham, 2018. DOI:10.1007/978-3-030-01264-9_8.

[16] Howard, A., Zhu, M., Chen, B., & Kalenichenko, D. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[17] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming[C]// Proceedings of the 2017 IEEE international conference on computer vision. Piscataway: IEEE, 2017: 2736-2744.

[18] WU D, LV S, JIANG M, et al. Using channel pruning-based YOLOv4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments[J]. Computers and Electronics in Agriculture. 2020, 178: 105742.

[19] WANG D, HE D. Channel pruned YOLO V5s-based deep learning approach for rapid and accurate apple fruitlet detection before fruit thinning[J]. Biosystems Engineering, 2021, 210: 271-281.

[20] FANG G, MA X, SONG M, et al. Depgraph: Towards any structural pruning[C]// Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2023: 16091-16101.

[21] CHEN X, ZHU J, JIANG J, et al. Tight compression: compressing CNN model tightly through unstructured pruning and simulated annealing based permutation[C]// Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC). Piscataway: IEEE, 2020: 1-6.

[22] SITU Z, TENG S, LIAO X, et al. Real-time sewer defect detection based on YOLO network, transfer learning, and channel pruning algorithm[J]. Journal of Civil Structural Health Monitoring, 2024, 14(1): 41-57.

[23] YANG F. An improved YOLOv3 algorithm for remote Sensing image target detection[C]// Proceedings of the 2021 Journal of Physics: Conference Series. Bristol: IOP Publishing, 2021, 2132(1): 012028.

[24] CHENG X , ZHANG Y , CHEN Y ,et al. Pest identification via deep residual learning in complex background[J].Computers and Electronics in Agriculture, 2017, 141:351-356

[25] SUH H K , JORIS I , WILLEM H J ,et al. Transfer learning for the classification of sugar beet and volunteer potato under field conditions[J].Biosystems Engineering, 2018, 174:50-65

[26] MILELLA A , MARANI R , PETITTI A ,et al. In-field high throughput grapevine phenotyping with a consumer-grade depth camera[J]. Computers and Electronics in Agriculture, 2019, 156:293-306

[27] YAHYA A,ZAFER C, KOCAMAZ A F .Identification of haploid and diploid maize seeds using convolutional neural networks and a transfer learning approach[J].Computers and Electronics in Agriculture, 2019, 163(2019):1-11.