Application of Deep Convolutional Neural Network in Remote Protection of Transmission Lines

Wenjie Zheng¹*, Chengqi Li¹, Bo Yang² and Xuebin Lv²

¹State Grid Shandong Electric Power Research Institute, Jinan, 250003, China
²State Grid Shandong Electric Power Company, Jinan, 250001, China

*zhwjchww@163.com

Abstract. In order to alleviate the pressure of line operation and maintenance, it is necessary to use advanced image recognition methods to identify hidden line hazards. In this paper, we extract features from hidden targets and perform target recognition using deep convolutional neural network. Firstly, we collect a large number of photos with hidden dangers for transmission line. Secondly, we classify those samples according to the detection targets. Thirdly, the deep convolutional neural network model is trained and tuned by using the sample images of various detection objects. When deployed on the server, identification of hidden dangers such as cranes, tower cranes, excavators, and mountain fires reaches acceptable accuracy.

1. Introduction

Destruction caused by hidden dangers including construction machines, foreign objects on transmission lines, mountain fires and smoke are the main causes for line trips. In order to improve the capability to protect overhead transmission lines, an intelligent identification and early warning system for hidden dangers of transmission lines is proposed. To further reduce the workload of artificial identification of hidden dangers, we try to identify the types of hidden dangers automatically. The traditional target detection method is mainly divided into four steps: pre-processing, feature selection, target recognition and target location, whose research focuses on feature extraction and feature classification. Therefore, researchers have proposed various forms of features and classifiers.

In order to extract better features, Hinton use deep neural networks to automatically learn high-level features from large amounts of data in 2006 [1]. Compared with pre-designed features, learning features are more abundant and expressive. OverFeat mainly uses a multi-scale sliding window for classification, positioning and detection [2]. However, OverFeat uses a sliding window to generate candidate regions, which is essentially exhaustive and has a high time complexity. In 2014, Girshick et al. proposed the R-CNN model [3], which introduced a search algorithm instead of the traditional sliding window idea for target location. However, the process of R-CNN training model is very complicated, and the detection speed is very slow, which is a method that is not dominant in time and space. SPP-Net designed a spatial pyramid pooling layer in convolutional neural networks [4], which can extract feature vectors of the same length from different size feature maps. However, the fine-tuning weights in SPP-NET only update the fully-connected layer behind the network, and the previous convolutional layer is not updated. It is not advisable for deeper networks to adopt this method. Girshick proposed the Fast R-CNN model based on R-CNN and SPP-NET [5]. Compared with the R-CNN model, the feature is extracted for each candidate region. The Fast R-CNN only
extracts feature once for the detected image. The target detection and recognition accuracy are already high, but the speed of the model detection is affected by the process of extracting the candidate region. YOLO integrates target area prediction and target category prediction into a single neural network model to achieve rapid target detection and recognition with high accuracy [6]. However, each grid of YOLO only predicts two boxes, one category, which causes the model to predict the accuracy of adjacent targets. YOLO can quickly identify targets in an image, but accurately targeting a target is difficult. SSD takes into account both the accuracy and real-time requirements [7], but the shape of the Default Box and the size of the grid in the SSD are fixed in advance, resulting in the model not being good enough for the extraction of small targets in a particular picture. Ren et al. proposed the Faster R-CNN model [8]. Since RPN and Fast R-CNN share the feature extraction of the convolutional neural network, only one feature of the image to be detected is extracted, which speeds up the target detection.

This paper adopts deep convolutional neural network-based scheme to extract features and target recognition from hidden targets. On the basis of the accumulated image of a large number of transmission line hidden danger samples, the samples are classified according to the detection target, and the deep convolutional neural network model is trained and tuned by using the sample images of various detection objects, and the model is deployed in the system to achieve accurate identification of hidden dangers such as cranes, tower cranes, excavators, and mountain fires.

2. Labelling hidden danger targets and establishing sample library

In order to meet the normative nature of the hidden danger target label, this project requires the marking file recording format to be in accordance with the .xml conforming to the VOC2007 data set format. The marked content is a hidden target in transmission line corridors that is easy to cause failure of the transmission line, which are sown in figure 1.

![Sample Instance](image)

**Figure 1.** Sample Instance.

In order to improve the generalization ability of the model and avoid over-fitting, a large amount of training data needs to be marked. In this system, data is marked in a semi-automatic manner. That is, in the initial stage of system establishment, firstly, a batch of data is manually marked, and then the data is used to train the detection model, then the unlabelled picture is detected by the trained detection model to obtain the annotation information, finally a part of the mislabelled image is manually assisted. In the later stage, the result of the model detection can be directly used as the annotation information to realize the automatic labelling of data, which can effectively expand the transmission hidden danger sample image database.

In this project, according to the above marking requirements, after the sample is sorted and manually labelled, a training sample library is formed. The number of hidden target targets is shown in the following table 1.
### Table 1. Labelled hidden danger number.

| Primary classification       | Secondary classification | Quantity of sample photos | Quantity of hidden dangers |
|-----------------------------|--------------------------|---------------------------|----------------------------|
| Crane                       | Crane                    | 45000                     | 69000                      |
| Tower crane                 | Tower crane              | 36000                     | 43500                      |
| Construction machinery      | Bulldozer               | 31500                     | 37500                      |
|                             | Excavator               | 27000                     | 33000                      |
| Foreign objects on          | Kite                     | 300                       | 300                        |
| transmission line           | Plastic film            | 345                       | 345                        |
|                             | Dustproof net           | 285                       | 285                        |
| Mountain fire               | Mountain fire            | 18000                     | 21000                      |
| Smoke                       | Smoke                    | 30000                     | 39000                      |

### 3. Identification of hidden dangers for transmission lines

It has been determined above that the target detection in the transmission application scenario uses the Faster R-CNN recognition algorithm, and the CNN model needs to be trained using the labelled samples.

#### 3.1. Data preprocessing

There are many changes in the picture of the transmission hidden danger samples. Since photos taken by cameras installed on transmission towers have a large time span and complicated weather conditions, it is necessary to perform pre-processing on those photos in order to obtain a better recognition effect. The main operations include mean removal, normalization, and data enhancement.

1) **Mean removal.** The average of all training set images is subtracted from each image, after which the data of each dimension of the input data is centered to 0, by which we can enhance the robustness of image recognition to illumination, weather changes, etc.

2) **Normalization.** After mean removal, the data in each dimension is divided by the standard deviation of the data in this dimension to ensure that all data is normalized to between -1 and 1. Normalization ensures that the data in each dimension is in a range of changes.

3) **Data enhancement.** Among the specified types of hidden dangers, excavators and cranes are easily misjudged as trucks, buses and so on, but those misjudgements can be eliminated by increase data diversity using other car databases. In addition, each picture in the training set is shifted by 5 pixels in the up, down, left, and right directions, and then all the pictures are horizontally flipped to obtain an expanded training set.

#### 3.2. Training of Faster R-CNN model

Faster R-CNN combines the RPN network and the Fast R-CNN network, that is, the candidate area acquired by the RPN is directly connected to the ROI pooling layer. Target identification of hidden dangers using deep neural networks consists of two main phases including training and testing.

In the training phase, the hidden danger target appearing in the image is first marked, and the sample database is established. Then the error back propagation (BP) algorithm is used to train the network, and the network weight coefficient is updated with random gradient descent method, this process iterates many times until the model converges.

In the test phase, the convolutional neural network is used to learn the advanced features in the image, and then the RPN deep neural network is used to generate candidate regions for hidden targets such as cranes, tower cranes, excavators and mountain fires. Finally, SoftMax classification and Bbox regression are performed, giving the type and location information of each hidden target.

The specific steps of the loss function design, the Faster R-CNN training process and the BP training algorithm are described in detail below.
1) Loss function. RPN consists of two branches including a classification layer and a regression layer. The classification layer gives a two-category label, which means whether the area contains the target. The regression layer produces the position of the target. Current area is considered to contain the targets when its IOU with some ground-truth box is greater than 0.7, and it is considered to be a negative sample when its IOU with any ground-truth box is less than 0.3. The loss function is expressed as equation:

$$L\left(\{p_i\}\{t_i\}\right) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p^*_i) + \lambda \frac{1}{N_{reg}} \sum_i p^*_i L_{reg}(t_i, t^*_i)$$

(1)

Among them, $N_{cls}$ and $N_{reg}$ weighs classification and regression. Classification loss is two classification, while regression losses is non-category related. RPN is a full-convolution process, full connection of subsequent vectors is achieved by 1*1 convolution.

2) Back propagation training. The BP algorithm is the basic algorithm of neural network optimization, whose learning process includes two stages of forward propagation and back propagation.

3) Faster R-CNN training process. Training of faster R-CNN continues based on trained CNN models such as VGG16 and Resnet152, and the actual training process is shown in figure 2.

![Figure 2](image)

**Figure 2.** Training process of Faster R-CNN.

The training process is similar to an iterative process and iterated twice here.

4. Experiments

Since TensorFlow is a clear and efficient open source deep learning framework that can be used for computing on a variety of platforms, this paper uses TensorFlow for training.

In the actual training process, the verification set are generally used to check the effect of the current saved model after training N epoch ends, so that the model or parameter problem can be discovered in time. For example, when the model encounters divergence on the verification set, mean Average Precision does not grow, or grows slowly, it is necessary to terminate the training in time, re-adjust or adjust the model without waiting for the training to end. When the effect on the verification set is much worse than that on the training set, it is necessary to consider whether the model is over-fitting, and it is necessary to adjust the hyper-parameter in time to ensure the final training effect of the model, which is shown in table 2.
Table 2. Values of hyperparameters after adjusted.

| Hyperparameters              | Description                                           | Value  |
|------------------------------|-------------------------------------------------------|--------|
| Learning rate                | Learning rate determines the speed of weight update   | 0.01   |
| Weight decay                 | Influence on the loss function caused by adjusting model complexity | 0.05   |
| Iterations                   | The number of times the entire training set is input to the neural network for training | 400000 |
| Learning rate decay          | Reduced learning rate every n iteration                | 0.1 times |
| Weight initialization        | Initialize the weight of each network layer           | Evenly distributed |

In order to test the performance of the trained final model, it is necessary to test the trained model using the test set. The test set is completely independent of the training set, and there is no intersection between the two, and the test set data types and quantities are shown in table 3.

Table 3. Tag number of hidden dangers in test set.

| Hidden danger          | Crane | Tower crane | Excavator | Bulldozer | Kite & Plastic film & Dustproof net | Smoke | Fire |
|------------------------|-------|-------------|-----------|-----------|-------------------------------------|-------|------|
| Tag number             | 1626  | 864         | 1173      | 275       | 800                                 | 756   | 315  |

The test results of various hidden dangers are shown in table 4.

Table 4. Missed rate and false rate.

| Target               | Crane | Tower crane | Excavator | Bulldozer | Kite & Plastic film & Dustproof net | Smoke | Fire |
|----------------------|-------|-------------|-----------|-----------|-------------------------------------|-------|------|
| Missed rate          | 6.31% | 7.91%       | 7.82%     | 9.23%     | 12.45%                              | 4.37% | 4.61%|
| False rate           | 9.93% | 5.14%       | 9.78%     | 9.67%     | 14.20%                              | 10.32%| 9.61%|

The test results in the same scenario under the test set are shown in figure 3.

Figure 3. Detection results at different times in the same scene.

It can be seen from figure 3 that the algorithm can obtain better hidden danger target detection results at different times in the same scene, indicating that the algorithm is robust to complex conditions such as illumination changes and weather changes.

The test results in various scenario under the test set are shown in figure 4.
Figure 4. Detection results in various scenes.

It can be seen from figure 4 that the algorithm can obtain good detection results in different scenarios, indicating that the system has good generalization ability.

5. Conclusions
In order to take full advantage of artificial intelligence technology and achieve better protection for transmission lines, this paper uses Faster R-CNN to extract features from hidden targets and realize target recognition. On the basis of collecting a large number of hidden dangers on transmission lines, the samples are classified according to the detection target, and the deep convolutional neural network model is trained and tuned by using the sample images of various detection objects. When the model is deployed on the test set, test results show that the algorithm is robust to complex conditions such as illumination changes, weather changes and various scenarios, indicating that our algorithm has good generalization ability.

Acknowledgment
This paper was supported by State Grid Science and Technology Project (520626180067).

References
[1] Hinton G E and Salakhutdinov R 2006 Reducing the Dimensionality of Data with Neural Networks Science vol 313 pp 504-507
[2] Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R and Lecun Y 2013 Overfeat: integrated recognition, localization and detection using convolutional networks Eprint Arxiv:1312.6229
[3] Girshick R, Donahue J, Darrell T and Malik J 2014 Rich feature hierarchies for accurate object detection and semantic segmentation Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, CVPR 2014 pp 580-587
[4] K He, X Zhang, S Ren and J Sun 2015 Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition IEEE Transactions on Pattern Analysis and Machine Intelligence vol 37 pp 1904-1916
[5] Girshick R 2015 Fast R-CNN Proc. of the IEEE Int. Conf. on Computer Vision, ICCV 2015 pp 1440-1448
[6] Redmon J, Divvala S, Girshick R and Farhadi A 2016 You Only Look Once: Unified, Real-Time Object Detection 2016 Pro. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, CVPR 2016 vol 2016-December pp 779-788
[7] Liu W, Anguelov D and Erhan D, et al 2015 SSD: Single shot multibox detector Lecture Notes in Computer Science vol 9905 LNCS pp 21-37
[8] Shaoqing R, Kaiming H, Ross G, and Jian S 2017 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks IEEE Trans. on Pattern Analysis and Machine Intelligence vol 39 pp 1137-1149