Image recognition and detection based on fast area convolutional neural network

Hanwen Zhang\textsuperscript{1*}, Zhen Qin\textsuperscript{2} and Hua Xie\textsuperscript{1}

\textsuperscript{1} School of Automation Engineering, UESTC, Chengdu, China
\textsuperscript{2} Network and Data Security Key Laboratory of Sichuan Province, University of Electronic Science and Technology of China

*Corresponding author. Email: 201952060908@std.uestc.edu.cn

Abstract. Nowadays, image recognition and detection technologies based on traditional artificial neural networks and convolutional neural networks are slightly inadequate in terms of training and recognition time and accuracy, and are difficult to deploy on devices with limited hardware resources. Therefore, this article proposes a recognition and detection technology based on fast regional convolutional neural networks. We use RPN (Region Proposal Network) instead of Selective Search method to rebuild the network, and add a new ROI pooling layer before the fully connected layer of CNN. Determine the category. The average detection accuracy on our data set can reach 83.8\%, and the training time is only 0.34 hours.

Keywords: deep learning, image recognition and detection, convolutional neural network, fast area convolutional neural network, RPN

1. Introduction

With the continuous development of artificial intelligence technology, how to intelligently process massive image data on its basis has become one of the research hotspots in image recognition and detection technology. It is a challenge to achieve high-precision and rapid recognition and detection.

In order to process a large amount of image data and complete the task of image recognition and detection, traditional artificial neural networks use more solutions, which can achieve better recognition and detection results. However, this method usually requires a large number of parameters, each pixel has a small connection with a pixel located farther away, and the number of network layers is also subject to certain restrictions.

An effective way to reduce the parameters and build a deep network structure is to use convolutional neural networks. The network is locally connected, that is, each neuron is no longer connected to all neurons in the previous layer, and only partially connected to some of the neurons with larger connection. At the same time, shared weights are utilized, that is, a group of connections share a weight. Rather than each connection has one more weight.

Convolutional neural networks are the same as traditional artificial neural networks and are composed of neurons with learning weights and biases. Each neuron receives some input, makes a dot product, and optionally follows non-linear conditions. Each layer of the convolutional neural network converts one activation value to another through a differentiable function. Generally, the network structure has a convolutional layer, a pooling layer, and a fully connected layer. There will be a...
convolutional layer, a pooling layer, and a fully connected layer. An activation function, stacking these layers forms a complete network structure. In this structure, the input information can be assigned to different levels according to requirements. However, only using convolutional neural networks for image recognition and detection will create problems such as over-fitting, huge memory usage, and slow calculation speed.

The R-CNN algorithm was first put forward by Grishick et al in 2013. It uses Alexnet as the backbone network and combines CNN with the candidate frame recommendation method. The recognition and detection process is divided into 2 parts: feature extraction + SVM classification. Compared with CNN alone there is a big improvement, but there are still problems such as slow computer speed, pixel loss, and large space consumption.

In order to solve the above problems, we adopted the Faster R-CNN algorithm for image recognition and detection to achieve faster calculations and higher recognition and detection accuracy.

In summary, our major contributions are as follows:

- The improved CNN network structure is adopted, and the ROI Pooling layer is added after the last layer of the convolutional layer. It is no longer necessary to input and perform compression operations, which avoids the loss of pixels and solves the problem of scale scaling.
- Our method only is required to send one image to the network during training. Each image extracts CNN features and suggested regions at one time. The training data is directly entered into the loss layer in the GPU memory, so that the first few layers of features in the candidate region are not required. Repeat the calculation and no longer are required to store a large amount of data on the hard disk. And use the multi-task loss function (multi-task loss), the bounding box regression is directly added to the CNN network, so as to get faster and more accurate training.
- Use RPN (Region Proposal Network) instead of Selective Search to generate the recommendation window, so that the convolutional network can automatically generate the recommendation frame, and share the convolutional network with the recognition and detection network, thereby greatly reducing the number of recommendation frames and essentially improving the quality of the recommendation frame, To achieve high-speed and high-precision identification and detection tasks.

### Related work

#### 1.1. Image recognition and detection front technology

In recent years, deep learning technology has made major breakthroughs in both theory and basic framework. Deep learning frameworks mainly include Theano, Tensorflow, MXNet, Keras, Ptorch, and Caffe, among which TensorFlow is the most used deep learning framework at this stage. In image processing, most scenes need to first convert the image into a matrix or vector form, and then recognize and detect the image. Therefore, the numpy package is a very important pre-technology in the image recognition algorithm. The numpy matrix performs scientific calculations on pictures, simplifies the picture processing process into space vector calculations, and then realizes image recognition and detection. This article will use the TensorFlow deep learning framework and the numpy package for image recognition and detection technology.

#### 1.2. Network structure improvement

For the R-CNN algorithm, it is necessary to extract Region Proposal (about 2000) from the image and use each Proposal as an image for CNN feature extraction and SVM classification. In fact, an image has been extracted about 2000 features. And classification, because these 2000 or so Region Proposals are
all part of the same image, it is completely possible to perform feature extraction of the convolutional layer on the image, but because the scale of each Region Proposal is different, it cannot be directly connected to the fully connected layer. Because the input of the fully connected layer must be a fixed length. At this time, the SPP-Net algorithm proposed by He et al. in 2014 solves this problem. Its function is to correctly input to the network regardless of the scale of the input picture.

1.3. Network structure improvement
For the R-CNN algorithm, it is necessary to extract Region Proposal (about 2000) from the image and use each proposal as an image for CNN feature extraction and SVM classification. In fact, an image has been extracted from about 2000 features. And classification, because these 2000 or so Region Proposals are all part of the same image, it is completely possible to perform feature extraction of the convolutional layer on the image, but because the scale of each Region Proposal is different, it cannot be directly connected to the fully connected layer. Because the input of the fully connected layer must be a fixed length. At this time, the SPP-Net algorithm proposed by He et al. in 2014 solves this problem. Its function is to correctly input to the network regardless of the scale of the input picture.

This article uses the Faster R-CNN algorithm to add the ROI pooling layer to the final CNN network structure as a streamlined version of SPP-NET, which realizes the function of mapping from the original image area to the conv5 area and finally pooling to a fixed size. Just send a picture entering the network, there is no need for a lot of repeated calculations and no need to store a large amount of additional data on the hard disk.

1.4. Generate recommendation boxes quickly and efficiently
Having a fast and efficient method for seeking recommendation frames is very helpful for the improvement of image recognition and detection technology. This is also the bottleneck of many image recognition and detection algorithms. In this paper, the Faster R-CNN algorithm will add a neural network for edge namely RPN (Region Proposal Network) is used to replace the Selective search method to generate a recommendation box. It slides the window on the feature map to establish a neural network for object classification and regression of the box position. The position of the sliding window provides the general position information of the object, and the regression of the box provides a more accurate position of the box.

1.5. Bounding box regression
Image recognition and detection, not only need to locate the bounding box of the target object, but also need to identify the object in the target Bounding Box. Since the algorithm cannot perfectly match the manually labeled data. IOU evaluation and publicity are required. It defines the degree of overlapping of two Bounding Boxes.

A calculation formula for the coincidence degree IOU of rectangular boxes A and B is:

\[
IOU = \frac{(A \cap B)}{(A \cup B)}
\]  

(1)
Figure 3. Regression window.

As showed in Figure 3, the green is Ground Truth, and the red is Region Proposal. Even if the red box is recognized as an airplane by the classifier, the red box is not accurately positioned, that is, IOU<0.5, which is equivalent to a recognition error, so this article uses Bounding Box Regression to fine-tune the window to achieve higher precision image recognition and detection.

2. Approach
In this section, we will introduce the detailed algorithm design and function implementation.

2.1. Motivation
The current step of using R-CNN for image recognition and detection is to input the test image. Use the Selective search algorithm to extract about 2000 Region Proposal from top to bottom of the image, and scale each Region proposal to a size of 227*227 and enter it. To CNN, use the output of the fully connected layer of CNN as features, and then input the CNN features extracted by each Region Proposal into SVM for classification, and finally perform border regression on the classified Region Proposal, and use the Bounding Box regression value to correct the original. The recommended window to generate prediction window coordinates. Doing so will divide the training into multiple stages (fine-tuning network + SVM training + training frame regression), the steps are cumbersome; training is time-consuming, the disk takes up a lot of space, 5000 images will generate hundreds of G of feature files; slow, use GPU, VGG16 model requires 47s to process a picture; SVM and regression are follow-up operations, and CNN features are not learned and updated during SVM and regression. Therefore, we use the Faster R-CNN algorithm to achieve image recognition and detection tasks, greatly improving the calculation speed and accuracy, and saving a lot of storage space.

2.2. Network architecture design
We choose the improved algorithm Faster R-CNN to solve the problems of slow R-CNN and insufficient accuracy when performing image recognition and detection.

Our algorithm process can basically be divided into four parts, Conv Layers, the input is the picture, the output is the feature maps of the extracted picture, which are used as the shared features of the RPN and the fully connected layer; RPN, based on the feature maps to generate the region proposal, It is mainly to use softmax for anchors to determine the actual foreground or background, and perform Bounding Box regression on the anchors to obtain the ideal proposals; Roll Pooling Layer, the input is feature maps and Proposal, and the output is the extracted Proposal feature maps, which are used The category is determined at the fully connected layer; Classifier, based on the Proposal feature maps, calculates the Proposal category, and performs the Bounding box regression again to obtain the accurate position of the object detection frame.
Figure 4. The basic structure of Faster R-CNN.

- Region Proposal Networks (RPN). RPN takes pictures of any size as input and outputs a collection of rectangles of object Proposal. Each rectangle has an objective score. The RPN is implemented as follows: For the conv feature map output by the last shared conv layer, a small network is used to slide it horizontally. The network is fully connected to a 3*3 spatial window of the input conv feature map, followed by another ReLUs layer, in which each sliding window is mapped to a 512-dimensional fully connected feature, and then two branch fully connected layers-boundary regression layer (reg) and bounding box classification layer (cls) are generated after the feature. Among them, reg-layer is used to predict the coordinates x, y and width and height of the proposal corresponding to the central anchor point of the proposal; cls-layer is used to determine whether the proposal is the foreground or the background. The processing method of the sliding window ensures that the reg-layer and cls-layer are associated with all the feature space of the conv feature map.

At the position of each sliding window, k regional proposals need to be predicted. So reg-layer has 4k outputs, and cls-layer outputs 2k score, which is used to estimate the probability of the object or non-object of each proposal. k proposals are parameterized with respect to k reference boxes, which are recorded as anchors. Each anchor is at the center of the sliding window and is related to a scale and aspect ratio. We set the values of scales and aspect ratios to 3, so we get k=9 anchors at each sliding window position. For a W*H conv feature map, we will get W*H*k anchors, and all anchors are it has to scale invariance.

Figure 5. Region proposal network (RPN).
Figure 6. Example of using RPN Proposals.

- **Loss Functions.** Before calculating the loss value, set the anchor point calibration method. If the IOU value of the reference box corresponding to the Anchor and the ground truth is the largest, mark it as a positive sample or the reference box corresponding to the Anchor and the ground truth IOU > 0.7, mark it as a positive sample; If the IOU of the reference box and ground truth corresponding to the Anchor is less than 0.3, mark it as a negative sample, and the rest will not be used for final training.

The calculation formula is:

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

(2)

Where \( p_i \) is the predicted classification probability of a positive sample, When it is a positive sample, \( p^*_i = 1 \), otherwise, \( p^*_i = 0 \). The parameterized coordinates of the predicted Bounding Box when \( a = t_i \) positive sample, \( t^*_i \) is the parameterized coordinates of the Ground Truth Bounding Box of the positive sample, \( N_{cls} \) means mini-batch size, \( N_{reg} \) means anchor location, \( L_{reg} \) is the detection error, \( L_{cls} \) is the log loss of the two categories of classification error.

\[
L_{cls}(p_i, p^*_i) = -\log(p_i p^*_i + (1 - p_i) (1 - p^*_i))
\]  

(3)

For Bounding Box regression loss, 4 coordinates are used for parameterization.

**Table 1.** 4 Coordinate parameterized calculation formulas.

| \( t_x \) | \( t_{x^*} \) |
| \( t_y \) | \( t_{y^*} \) |
| \( t_w \) | \( t_{w^*} \) |
| \( t_h \) | \( t_{h^*} \) |

| \( t_x = \frac{x - x_a}{w_a} \) | \( t_{x^*} = \frac{x^* - x_a}{w_a} \) |
| \( t_y = \frac{y - y_a}{h_a} \) | \( t_{y^*} = \frac{y^* - y_a}{h_a} \) |
| \( t_w = \log(\frac{w}{w_a}) \) | \( t_{w^*} = \log(\frac{w^*}{w_a}) \) |
| \( t_h = \log(\frac{h}{h_a}) \) | \( t_{h^*} = \log(\frac{h^*}{h_a}) \) |

Among them, \( x, y, w, \) and \( h \) represent the two coordinates of the center of the box, width and height. \( x, x_a, \) and \( x^* \) are prediction box, anchor box and groundtruth box, respectively. It can be seen as the return of the bounding box from an anchor box to its nearby groundtruth box.

- **Rol Pooling.** Generally, the fully connected layer or classifier of CNN requires a fixed input size, so the input data have to be cropped or compressed. These preprocessing will cause the loss of data and geometric distortion, so SPP Net appears, which uses the pyramid idea realize the multi-scale input of data.
As showed in Figure 7, a SPP layer is added between the convolutional layer and the fully connected layer. At this time, the input of the network can be of any scale. Each pooling filter in the SPP layer will be realized according to the input, and the output scale of the SPP is always fixed. Therefore, using the characteristics of SPP Net. The feature map of the entire image can be obtained by convolving the original image only once. Rol pooling layer can actually be regarded as a single-layer SPP Net. Rol pooling layer only needs to be down sampled to a 7*7 feature maps. For the conv feature maps of the VGG16 network, there are 512 feature maps, so that all region proposals correspond to A 7*7*512 dimension feature vector is used as the input of the fully connected layer, which realizes the function of mapping from the original image area to the conv area and finally pooling to a fixed size.

3. Experiments
We completed the training of the model using Python language on the PyCharm platform and got the experimental results.

3.1. Implementation details
We use the TensorFlow deep learning framework to implement the algorithm.

When training RPN, SGD and BP are used for end-to-end training, image-centric sampling strategy, and each mini-batch is composed of a single image containing many positive and negative anchors. Randomly sampled 256 anchors of a picture to calculate the loss of the mini-batch, the ratio of positive anchors and negative anchors sampled is 1:1, if the positive anchors of a picture are less than 128, use negative anchors to complete mini-batch. Initialize the newly added layer of the network with a mean value of 0 and a variance of 0.01 de Gaussian distribution, with momentum = 0.9, weight = 0.005, on the verification data set, the first 60k mini-batches, leraning_rate = 0.001, and the last 20k iterations learning_rate = 0.0001.

We use alternate optimization to share convolutional layer features between regional proposal and target detection, use ImageNet training model for network initialization, and perform end-to-end fine-tuned for regional proposal tasks. Based on the proposals generated by RPN, Fast R-CNN is used to train a separate detection network. The detection network is also initialized with the ImageNet training model. At this time, the two networks do not share the convolutional layer. Immediately use the detection network to initialize the training of RPN, but the fixed shared convolutional layer, only the fine-tune RPN network layer, the two networks share the convolutional layer, and finally the fixed shared convolutional layer, the full-tune of the fine-tune Fast R-CNN Connection layer. So far, the two networks share the same convolutional layer, realizing a unified network.

At the same time, parameter settings are very important for deep learning tasks, which will affect the effect of training models. We use single-scale images to train and test the regional proposal and object network, and rescale the image to 600 pixels. For anchors, we use 3 types of scales. The area of the box is 1282, 2562, and 5122 pixels, and 3 aspect ratios, 1:1, 1:2, 1:3. For the processing of the anchor boxes across the imaginary boundary, during training, all cross-boundary anchors are ignored, so it will not affect the loss, and the full convolution RPN is used to process the entire image during the test. Due to the high overlap of RPN Proposals, NMS is used here for processing based on the cls scores of the anchors.
proposal region, and the IOU threshold of the fixed NMS is 0.7. After NMS processing, top-N proposal regions are used for target detection.

3.2. Experimental results
We tested our algorithm against the data set. Our data set includes 4,040 images in the training set and 800 images in the test set which includes 100 images are black-and-white, 100 images are blurred, 100 images are deformed. The test result is shown in Figure 8.

![Figure 8. Image inspection test results.](image)

We compared the model with R-CNN and Fast R-CNN. The results are shown in tables below.

**Table 2.** The detection accuracy of each algorithm for different types of images.

| Images Type           | R-CNN (%) | Fast R-CNN (%) | Faster R-CNN (%) |
|-----------------------|-----------|----------------|------------------|
| Black-and-white images| 65.4%     | 70.6%          | 75.9%            |
| Blurred images        | 62.7%     | 68.3%          | 73.5%            |
| Deformed images       | 63.5%     | 65.7%          | 71.2%            |
| Normal images         | 68.1%     | 71.6%          | 83.8%            |

Different results will be got from different algorithms when detecting various types of images. From Table 2 we can learn that no matter what kinds of images are detected, Faster R-CNN always have the highest accuracy.

**Table 3.** Comparison of calculation efficiency of various algorithms (average result of 800 image detection).

|                     | R-CNN | Fast R-CNN | Faster R-CNN |
|---------------------|-------|------------|--------------|
| Test time per image | 60s   | 5s         | 0.3s         |
| Speedup             | 1x    | 12x        | 200x         |
| Training Time       | 67 hours | 5.6 hours | 0.34 hours   |
| Speedup             | 1x    | 12x        | 197x         |

It can be seen quite intuitively from the Table 3 that our algorithm has faster calculation and processing speed while completing image recognition and detection tasks, and has no additional large-capacity disk space compared to R-CNN.

4. Conclusions
Depending on results of the experiments, an obvious result can be concluded. In the detection of different types of pictures, Faster R-CNN is the most prominent in terms of accuracy and computational efficiency, have been greatly improved compared with R-CNN and Fast R-CNN. This is enough to prove the superiority of our algorithm. Combined with contents of Table 2 and 3, we can easily say that Faster R-CNN can better complete the work of target detection.

Acknowledgment
This work was supported in part by the National Natural Science Foundation of China (No.62072074, No.62076054, No.62027827, No.61902054), the Frontier Science and Technology Innovation Projects.
of National Key R&D Program (No.2019QY1405), the Sichuan Science and Technology Innovation Platform and Talent Plan (No.2020JDJQ0020), the Sichuan Science and Technology Support Plan (No. 2020YFSY0010), and the Natural Science Foundation of Guangdong Province (No. 2018A030313354).

References

[1] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. PAMI, (2016)
[2] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis & Machine Intelligence, (2014)
[3] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]. Conference on Neural Information Processing Systems, (2015)
[4] Girshick R, Fast R-CNN[C]. International Conference on Computer Vision, (2015)
[5] He K, Gkioxari G, Dollar P, et al. Mask R-CNN[C]/Proceedings of the IEEE International Conference on Computer Vision, (2017)
[6] AGRAWAL, P, GIRSHICK R, MALIK J. Analyzing the performance of Multilayer Neural Networks for Object Recognition[J], Lecture Notes in Computer Science, (2014)
[7] Zhang, W. Shift-invariant pattern recognition neural network and its optical architecture. In Proceedings of annual conference of the Japan Society of Applied Physics, (1988)
[8] M. Yousefi, S. Golmohammady, A. Mashal, F. D. Kashani. Analyzing the propagation behavior of scintillation index and bit error rate of a partially coherent flat-topped laser beam in oceanic turbulence[J], J. Opte. Am, (2015)
[9] Viola P, Jones M. Robust real-time object detection[J], International Journal of Computer Vision, (2001)
[10] Wang G, Ren G, Wu Z, et al. A robust, coarse-to-fine traffic sign detection method[C]. Neural Networks(IJCNN), The 2013 International Joint Conference on. IEEE, Dallas, TX, USA, (2013)
[11] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition(CVPR, 2015)
[12] P. E. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models,” IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI, 2010)
[13] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated recognition, localization and detection using convolutional networks,” in International Conference on Learning Representations(ICLR, 2014)
[14] K. Lenc and A. Vedaldi, “R-CNN minus R,” in British Machine Vision Conference(BMVC, 2015)
[15] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in Neural Information Processing Systems(NIPS, 2012)
[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, and A. Rabinovich, “Going deeper with convolutions,” in IEEE Conference on Computer Vision and Pattern Recognition(CVPR, 2015)
[17] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” in European Conference on Computer Vision(ECCV, 2014)
[18] P. O. Pinheiro, R. Collobert, and P. Dollar, “Learning to segment object candidates,” in Neural Information Processing Systems(NIPS, 2015)
[19] B. Alexe, T. Deselaers, and V. Ferrari, “Measuring the object-ness of image windows,” IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI, 2012)
[20] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra, “Object-Proposal Evaluation Protocol is ‘Gameable’,” arXiv:1505.05836, 2015.