Research on object detection algorithm based on deep learning for mobile Terminal

Jie Luo *, Jufeng Ye, Lan Shan, Qunying He
College of Mechanical and Electrical Engineering, Zhejiang Tongji Vocational College of Science and Technology, Hangzhou, China

Abstract. In order to improve object detection ability of robot, this study introduces an object detection algorithm which is based on deep learning. Firstly, a neural network that contains Convolution Layer, Pooling Layer and Fully Connection Layer is designed to recognize whether the target object in image. Secondly, the Faster-RCNN, as a typical object detection algorithm, is used to detect the position of target object in image. Finally, the algorithm is transplanted on Raspberry Pi for deployment and testing. Results show that this algorithm can work well on Raspberry Pi, and the mAP (mean average precision) and FPS (frames per second) reach 97% and 3. Meanwhile, the position of target object is accurately marked in image. Therefore, this study implies that the object detection algorithm based on deep learning can be implemented on mobile terminal and is helpful to improve object detection ability of mobile terminal.

Key words: Deep Learning; Object Detection; Mobile Terminal.

1. Introduction
With the development of science and technology, robot is widely used in our daily lives and expected to complete complex tasks as diverse as elderly care, communication or delivery. Therefore, it is necessary to make robot have the same visual ability as human being that can detect object in image. The traditional object detection method is to extract hand-crafted features from objects manually, such as SIFT [1] and HOG [2]. However, these hand-crafted features are low-level and very sensitive to environment changes, so that these methods can not work well in complex environment.

In order to solve this problem, the deep learning technique is applied to the field of object detection and shows better performance than SIFT [3] and HOG [4]. At present, the popular object detection algorithms based on deep learning technique are YOLO and Faster-RCNN. YOLO adopts one stage method and can figure out which object is in image and where it is by taking the entire image through its network for only once [5]. YOLO has fast speed, but its precision is relatively low. On the contrary, Faster-RCNN adopts two stage method which includes Region Proposal Network and Roi Pooling [6]. Faster-RCNN has high precision, but it runs very slow due to its large computation. Considering that in this paper the robot on which the mobile terminal is mounted is not required to have high real-time performance but be able to precisely mark the position of object in image. Therefore, this paper adopts Faster-RCNN for completing object detection task.
In the past few years, many researchers have studied Faster-RCNN object detection algorithm in various fields. Sun et al. designed an improved face detection algorithm based on Faster-RCNN framework and obtained the state-of-the-art face detection performance on the FDDB benchmark [7]. Han et al. proposed a real-time small traffic sign detection approach based on revised Faster-RCNN and had higher mAP than the original object detection algorithm [8]. Harish et al. introduced an enhanced Faster-RCNN algorithm based on recurrent convolution neural network architecture for crop diseases detection and classification, which can detect crop diseases in early stages [9]. However, these studies are all based on condition that the object category is known, which have failed to verify the feasibility of these algorithms in the presence of unknown objects. It has been proved by practice that the Fast RCNN can only detect object which is contained in training set, and it is very easy to make mistakes when unknown objects with similar shape appear. Therefore, this paper studies the object detection algorithm under the interference of unknown objects.

In this paper, the following methods are used to solve this problem. Firstly, a recognition neural network is designed to screen unknown objects, which includes Convolution Layer, Pooling Layer and Fully Connected Layer. Secondly, the object detection is carried out by Faster-RCNN algorithm. Finally, the object detection system is deployed on Raspberry Pi to complete testing.

2. Method

2.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the most important branches of deep learning, and which is widely used in segmentation and object detection. Convolutional Neural Network is a multilayer neural network and composed of Input Layer, Convolution Layer, Pooling Layer, Fully Connected Layer and Output Layer.

2.1.1. Convolution Layer. Convolution Layer (Conv) is the core of Convolutional Neural Network and complete the process of feature extraction. The basic unit responsible for feature extraction is kernel, which is connected with local receptive field of the upper layer. For pixels nearby have high correlation in the image but pixels far away are not, so neuron in this layer do not need to connect with each neuron in the upper layer and global features is achieved by integrating all local features. Meanwhile, the kernel used in one position of the image is also applicable in other positions, which is called weight sharing. Therefore, the whole feature maps are achieved by sliding kernel through the image, and the calculation process of kernel is shown in Figure 1. As a result, the number of parameters is reduced effectively.

![Figure 1. Calculation process of Convolution Layer.](image-url)
2.1.2. Pooling Layer. Pooling Layer (Pool) is an important concept of Convolutional Neural Network and is a form of non-linear down-sampling. Pooling Layer is usually inserted behind the Activation Function Layer and can reduce space size represented by features and number of parameters for calculation. The two main methods of Pooling Layer are Max Pooling Layer and Average Pooling Layer. For Max Pooling Layer, each feature map is divided into separate blocks on which nonlinear function is used to calculate result. The calculation process of Max Pooling Layer is shown in the Figure 2.

![Figure 2. Calculation process of Max Pooling Layer.](image)

2.1.3. Fully Connected Layer. Fully Connected Layer (FC) plays the role of classifier in the Convolutional Neural Network and is used to map the distributed feature representation to label space of sample. As shown in Figure 2, each node in one layer connects all nodes in next layer. The output of the last Fully Connected Layer is passed to softmax regression for classification, also known as Softmax Layer.

![Figure 3. Structure of Fully Connected Layer.](image)

2.1.4. Batch Normalization Layer. Batch Normalization Layer (BN) [10] is used in this paper to make data obey the standard Gaussian distribution and located between Convolution Layer and Activation Function Layer.

Firstly, the mean and covariance of each batch can be calculated as follows:

\[
\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i
\]

\[
\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2
\]
Where $\mu_B$ and $\sigma_B^2$ are the mean and covariance of mini-batch B, respectively; $m$ and $x_i$ are the sample size and the $i$th data of sample, respectively.

Secondly, the process of normalization is implemented by the following equation:

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\mu_B} + \varepsilon}$$

Where $\varepsilon$ is a small number to prevent the denominator from being zero when $\mu_B$ equals to zero.

Finally, parameters for scale and shift are obtained by the following equations, which is used to restore the expression ability of after-normalized data by implementing linear transformation.

$$y_i \leftarrow \gamma \hat{x}_i + \beta = BN_{\gamma, \beta}(x_i)$$

Where $\gamma$ and $\beta$ are parameters learned by training, respectively.

2.1.5. Recognition neural network. The recognition neural network in this paper includes Convolution Layer, Max Pooling Layer, Fully Connected Layer, Batch Normalization Layer and Activation Function Layer. The whole image is taken as input and processed by different layers mentioned above. The famous recognition neural network named VGG16 [11] is used as a reference, and the construction of neural network used in this paper is shown in Figure 3. The size of kernels used in Convolution Layer and Max Pooling Layer are $3 \times 3$ and $2 \times 2$, and the step of kernels are 2 and 2, respectively. Besides, the activation function of Relu is used in Activation Function Layer to increase calculation speed and prevent gradient from disappearing. Moreover, it can be found that the size of feature map becomes half of input size after Max Pooling Layer, but the channel doubles. It is helpful to enhance the fitting ability of neural network in nonlinear mapping and reduce the number of parameters.

![Figure 4](image_url)

(a) Network_1. (b) Network_2

**Figure 4.** Construction of recognition neural network: (a) Network_1, (b) Network_2.
2.2. Faster-RCNN object detection algorithm

The Faster-RCNN object detection algorithm [6] is composed of three components, and they are feature extraction network, Region Proposal Network (RPN) and RoI Pooling. Firstly, image with any size is taken as input and scaled to a size predetermined, and the feature maps of image are obtained by neural network of VGG16 [11]. Then, these feature maps are feed into Region Proposal Network to obtain anchor boxes. The Softmax Layer is used to determine whether the anchor box belongs to foreground or background. Meanwhile, the anchor boxes are modified by Bounding Box Regression in another branch to generate Region Proposals which have higher accuracy. Finally, the proposal feature maps with fixed size are obtained by RoI Pooling Layer and feed into Fully Connected Layer afterwards. The classification and accurate position of object are obtained by Softmax Layer and Smooth L1 Loss, respectively. The construction of Faster-RCNN object detection algorithm is shown in Figure 4.

![Figure 5. Construction of Faster-RCNN object detection algorithm.](image)

2.3. Dataset preparation

In order to train Convolutional Neural Network, dataset composed of enough images is needed which has a great influence on the final training result. In this paper, images are extracted from videos which are taken from different distance, exposure, background and visual angle, as depicted in Figure 5. The objects labeled from 01 to 06 are shown in Figure 5(b). For recognition neural network, images with size of 256×256 are used, and each object has about 1200 images. For Faster-RCNN object detection algorithm, the position information of object is calibrated and saved to the corresponding XML file, and each object has about 250 images which has the size of 400×400. We employ 70% of the dataset as a training set, and the remaining 30% as a testing set.

![Figure 6. Dataset preparation for training Convolutional Neural Network: (a) Images of object with different distance, exposure, background and visual angle; (b) Objects labeled from 01 to 06.](image)
2.4. Deployment and test on Raspberry Pi
The recognition neural network and Faster-RCNN object detection algorithm are trained on deep learning architecture of Tensorflow which has the advantages of flexibility, usability and high efficiency. All programs used in this paper are implemented by Python programming. The after-training model is deployed on Raspberry Pi which is installed on operating system of Raspbian based on Linux. Raspberry Pi has 4 GB of memory and quad core CPU with 1.5GHz, and it has powerful computational ability to complete tasks mentioned above.

3. Results and discussion

3.1. Influence analysis of Batch Normalization Layer
In order to analyze the influence of Batch Normalization Layer (BN) on neural network, two neural networks are designed, one with BN and the other without. The results of two neural networks are listed in Table 1. We can see that the performance of two neural networks are alike in training accuracy but differ greatly in testing accuracy. In particular, the neural network with BN obtains average training accuracy of 98.9% and average testing accuracy of 97.7% on all objects. However, the neural network without BN obtains average training accuracy and average testing accuracy of 95.8% and 94.3%, respectively. Meanwhile, the neural network with BN has the advantage of faster convergence rate over neural network without BN.

The results in Table 1 clearly indicate that the BN has a great influence on testing accuracy and convergence rate, and it is helpful to improve the generalization and convergence speed of neural network. This is due to the fact that the problem of gradient disappearance and gradient explosion is solved by normalizing output of each layer based on mean and covariance, resulting the same distribution of the input of each layer. Moreover, BN has an effect on regularization to some extent and can reduce over-fitting. Therefore, the convergence speed is accelerated and the result obtained is better.

3.2. Influence analysis of number of layers
In order to analyze the influence of number of layers on neural network, two neural networks are designed and shown in Figure 3. We can see that the network_1 in Figure 3(a) have more layers than network_2 in Figure3 (b). The training accuracy and testing accuracy of two neural networks on different objects are listed in Table 2. Although the network_1 has the advantage of faster convergence rate over network_2, the training accuracy and testing accuracy of network_1 are lower than that of network_2. In particular, the network_1 obtains average training accuracy of 95.6% and average testing accuracy of 94.1% on all objects, and the network_2 obtains accuracies of 98.8% and 97.5%, respectively.

The results in Table 2 clearly indicate that the number of layers has a great influence on testing accuracy and convergence rate. The neural network with more layers can obtain better result compared with the neural network with fewer layers. This is due to the fact that the deeper the neural network, the
more neurons in neural network. Moreover, the neural network with more layers have stronger abstract ability on image to complete the classification task. Therefore, the accuracy obtained is better.

3.3. Performance analysis of Faster-RCNN
Considering that in this paper the robot on which the mobile terminal is mounted is not required to have high real-time performance but be able to precisely mark the position of object in image, and the Faster-RCNN is well known for high mAP (mean average precision) but low FPS (frames per second), this work thus adopts Faster-RCNN to implement object detection task. From Table 3, we can see that the mAP and speed of used Faster-RCNN model are 98.2% and 3 FPS. The result of object detection on specified object is shown in Figure 6. We can see that the green box marked by Faster-RCNN is very close to the edge of object, and the score of object is nearly 99%. Therefore, we can conclude that the Faster-RCNN object detection algorithm can complete the task of object detection in this work.

| Faster-RCNN model | mAP (%) | Speed (FPS) |
|-------------------|---------|-------------|
|                   | 98.2    | 3           |

Table 3. The mAP and speed of used Faster-RCNN model.

Figure 7. The result of object detection on specified object

3.4. Future work
In this paper, we only studied how to recognize the kinds of objects. However, in order to grasp objects for robot, we also need to get the position information of objects in the real world. Therefore, further research should be undertaken to investigate the transformation between the position in the image and the position in the real world.

4. Conclusion
In this paper we presented deep-learning based algorithms for object recognition and object detection, respectively. For object recognition, we design a neural network and improve its performance by increasing the number of layers and using Batch Normalization Layer. For object detection, the Faster-RCNN is used to precisely mark the position of object in image. These algorithms are deployed on mobile terminal of Raspberry Pi to complete tasks. The experimental results show that the deep learning technique can be used to complete object recognition and object detection.

Acknowledgments
This work was financially supported by the General Scientific Research Project of Zhejiang Provincial Department of Education (No. Y202044985)
References

[1] D.G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.

[2] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2005.

[3] P. Fischer, A. Dosovitskiy, T. Brox, Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT, Computer Science, 2014.

[4] E. Cengil, A. Cinars, Comparison Of Hog (Histogram of Oriented Gradients) and Haar Cascade Algorithms with a Convolutional Neural Network Based Face Detection Approaches, International Journal of Advance Research, Ideas and Innovations in Technology, Vol. 3, No. 5, pp. 244-255, 2017.

[5] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[6] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Proceedings of the Advances in Neural Information Processing Systems, pp. 91–99, 2015.

[7] X.D. Sun, P.C. Wu, S.C.H. Hoi, Face detection using deep learning: an improved Faster-RCNN approach, Neurocomputing, Vol. 299, pp. 42-50, 2018.

[8] C. Han, G.Y. Gao, Y. Zhang, Real-time small traffic sign detection with revised faster-RCNN, Multimedia Tools & Applications, Vol. 78, No. 10, pp. 13263-13278, 2019.

[9] M. Harish, A.V.L.N. Sujith, K. Santhi, An enhanced faster-RCNN based deep learning model for crop diseases detection and classification, International Journal of Engineering and Advanced Technology, Vol. 8, No. 6, pp. 4714-4719, 2019.

[10] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 32nd International Conference on Machine Learning (ICML), 2015.

[11] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, International Conference on Learning Representations (ICLR), 2015.