Journal of Computers Vol. 30 No. 2, 2019, pp. 165-172
doi:10.3966/199115992019043002015

One-Shot Learning Method Based on Convolutional Neural Network for Intelligent Robot

Kan Wang[1], Wen-bai Chen[1*], Chang Liu[1], Ding-qian Liu[1], Hu Han[2]

[1] School of Automation, Beijing Information Science & Technology University, Beijing 100192, China
chenwb@bistu.edu.cn

[2] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China

Received 6 December 2018; Revised 6 February 2019; Accepted 6 March 2019

Abstract. In order to solve the insufficient of large-scale data processing ability for intelligent robot, and improve the ability of image classification, one-shot learning method based on convolutional neural network is proposed in this paper. Firstly, a sample is augmented with data augmentation techniques to construct a small target dataset, so that the neural network can capture the key features of the target task as much as possible. Then a pre-training model that performs well on large-scale datasets will be transferred to target task more or less. Due to its existing pre-training model weights and underlying features of the image, the layer freezing technique can be used to freeze the simple geometric shape features. The training and fine-tune will only be performed on the fully-connection layer and the last few conventional networks, which preserve the combined features, finally achieving the identification and classification on the target sample set. The experimental results show that the method has high accuracy and flexibility in the single-sample classification problem, meanwhile, can save a lot of time and computing resources.

Keywords: convolutional neural networks, data augmentation, intelligent robots, one-shot learning, transfer learning

1 Introduction

With the continuous development and wide application of artificial intelligence, deep learning has been highly valued by the industry at present [1]. Deep learning has improved the accuracy of image classification on large-scale datasets, which provides great convenience for people's life and work. However, as more application scenarios emerge, we are increasingly facing the problem of insufficient number of samples. Therefore, small sample learning has become an important research direction.

Small sample learning is to solve the problem of insufficient computing power or sample data, and use a small number of samples to approximate the classification and recognition accuracy of networks trained on large data sets. Small sample learning has received more and more attention. In order to achieve small sample learning, Flood [2] proposed the Relation Network. The feature is extracted by embedding module, classification is performed by the distance between the two features through relational model. Peng et al. [3] proposed that the first domain adaptation and sensor fusion method, which learn from no task-relevant target-domain data to solve the problem of less data. Chen [4] proposed a novel auto-encoder network dual TriNet to realize feature augmentation. Wang and Hebert [5] proposed that novel categories can learn from few annotated examples. Training a regression network with a large collection of model pairs can get a better classifier for a small number of samples. Chelsea [6] proposed a meta-learning algorithm that is model-agnostic to quickly learn a new task by training the model's initial parameters for a large number of different tasks.

[*] Corresponding Author

According to the above analysis, this paper, towards intelligent robots, presents one-shot learning method based on convolutional neural network. Using deep network learning methods, a single sample is augmented by data expansion technology to obtain more annotation data. Migrate the pre-training model and better fit the target domain by fine-tuning. Then the well-performing pre-trained model is migrated to the target dataset for training, and the sample classification is finally realized.

2 One-Shot Transfer Learning

2.1 Data Augmentation

The quality of the model is extremely dependent on the quality of the training data, and in the absence of sufficient data, the priority method is to increase the size of the dataset. Data augmentation techniques can increase the number of samples, so as to increase the extraction rate of effective information and increase the generalization ability of neural networks. There are many ways to extend the data of natural images. In terms of geometric transformation, including: rotation transformation, flip transformation, translation transformation and scale transformation, ect; in the aspect of pixel transformation, it includes: channel overlap, contrast transformation, color jitter, noise disturbance and so on. In the data augmentation, in order to avoid over-fitting, a picture is scaled, cropped, flipped, rotated random angle and other geometric operations. When the image is irregular due to random transformation, discard parts outside the size of the standard window and fill the missing parts with adjacent pixels. Finally, the image is ZCA whitened [7]. The whitening transformation is a linear transformation, the purpose is to eliminate redundancy and simplify input. It transforms a random variable vector with known covariance matrix into a new set of vectors, its covariance is the unit matrix, which means that they are irrelevant and each variable has a variance of 1. By mapping the original data onto the principal component axis, the decoupling of adjacent features accomplished.

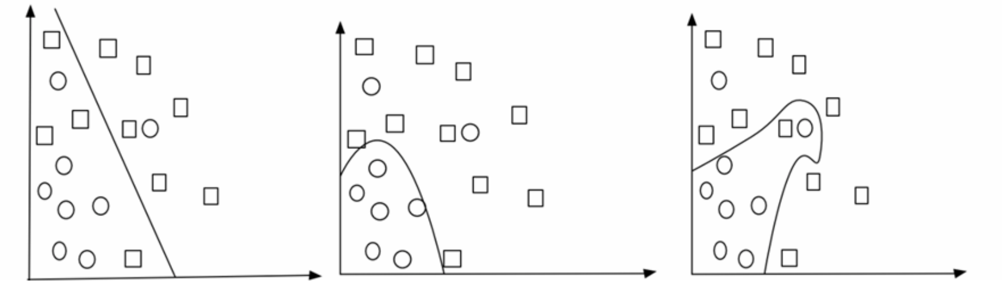

Fig. 1. Under-fitting, high deviation (left); Appropriate fit (middle); Overfitting, high variance (right)

2.2 Transfer Learning

Transfer learning is a machine learning method that makes full use of prior knowledge. It extends the mature technology and knowledge in a field to related fields, so as to solve the learning problem in the target field with only a small amount of tagged sample data or even no [8]. The core is to establish a link between existing knowledge and new knowledge.

In this experiment, we use a model-based transfer learning method (Fig. 2). First, preserve the convolution layer structure of the model, subtract the part after the fully-connection layer. Then do a simple training on the training set, record the feature map obtained on the last layer. Finally, the fully-connection layer or global average pooling is used to combine the deep features learned. Then map spatial correlation on each output channel separately.

2.3 Xception Neural Network Structure

Xception represents the strongest inception structure, which holds that the correlation between cross-channel and cross-space in convolution neural network [9] feature graph can be completely separated, so do not map them together. First, use a 1x1 convolution to map across-channel channel correlation (Fig. 3). This structure is called the deep separable convolution structure [10].

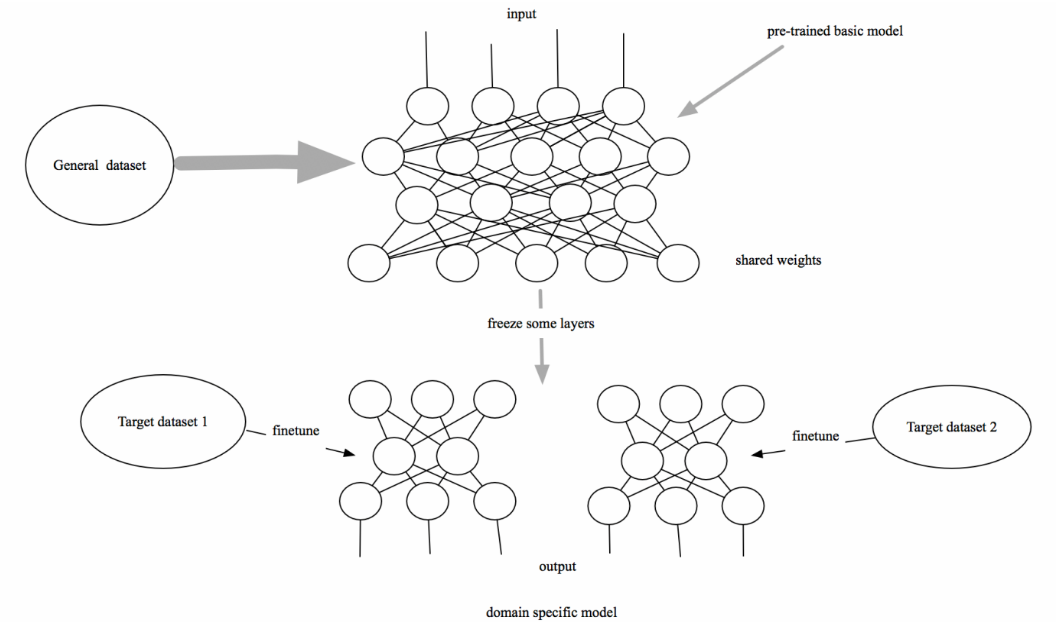

Fig. 2. Model-based transfer learning process

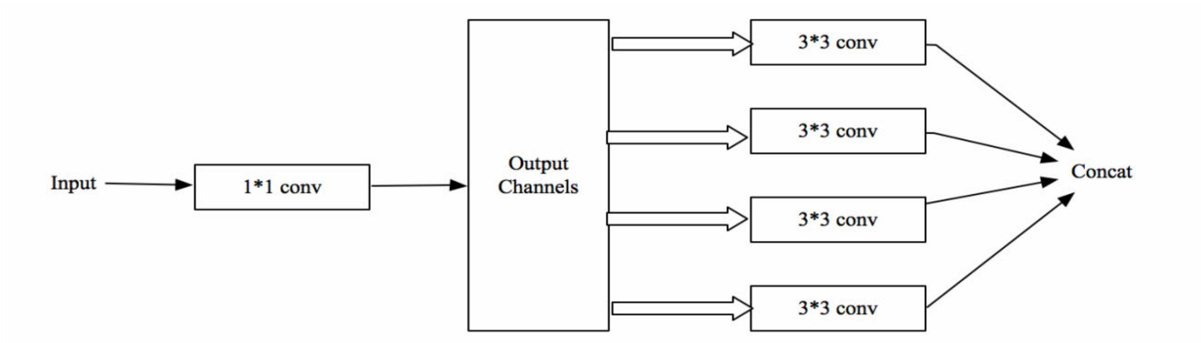

Fig. 3. Xception basic component module

The selection of the pre-training model directly affects the accuracy of the classification results. Due to the limited number of open source models with pre-training weights, after comparisons including VGG-16, ResNet50, DenseNet121, InceptionV3, Xception and MobileNet, the Xception network [11] is selected as the pre-training model (Fig. 4).

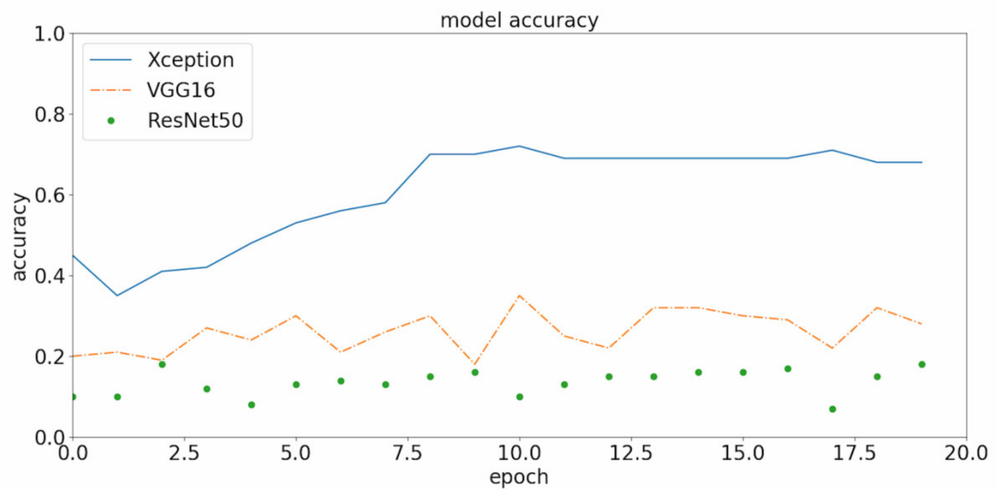

Fig. 4. Validation accuracy of various pre-training models after 20 iterations

The VGG-16 model has only 23 layers, but it has several times more than the parameters of other models, and most of them focus on the fully-connection layer. The effect is not obvious because the network depth is too shallow, and the depth of the network is crucial [12-13]. The deep network naturally integrates low-level, intermediate, and advanced features in an end-to-end manner, and the level of features can be augmented by stacking layers. So the deeper the network theory has more powerful expression ability.

The ResNet50 model and the Xception model are similar in terms of network depth, structure and the number of parameters, and all contain local residual structures (Table 1). In the information transmission, the deep network often has the problem of effective information loss or partial deletion, and it is easy to have the phenomenon of gradient disappearing in the backward propagation, which leads to the decrease of the training effect. The residual structure overlays the input information directly into the output through a linear forward channel. When the full information of the input is ensured to reach the output, the content that the network needs to learn is also changed into the residuals of input and output, which reduces the difficulty of the learning process.

Table 1. Pre-training model comparison

Model	Time/ms	Training set loss function	Training accuracy	Verification set loss function	Verification set accuracy
Xception	25	3.2236	0.8000	4.9617	0.6800
VGG16	6	0.0560	0.9900	4.9515	0.2800
ResNet50	0.768	1.5228	0.4500	2.3877	0.1800

3 One-Shot Learning Classification Analysis

3.1 Experimental Dataset

In this experiment, the generic dataset is ImageNet. The other two datasets are the target domain, one is a small dataset containing 15 types of pictures, and the other is a Caltech-101 dataset that contains 101 types of pictures, to compare the performance of the migrated network in different sizes of target domains.

Small datasets are mainly used to evaluate the performance of different pre-training models, as well as to adjust some parameters, such as learning rate, activation functions, regularization parameters, value evaluation functions and so on (Fig. 5).

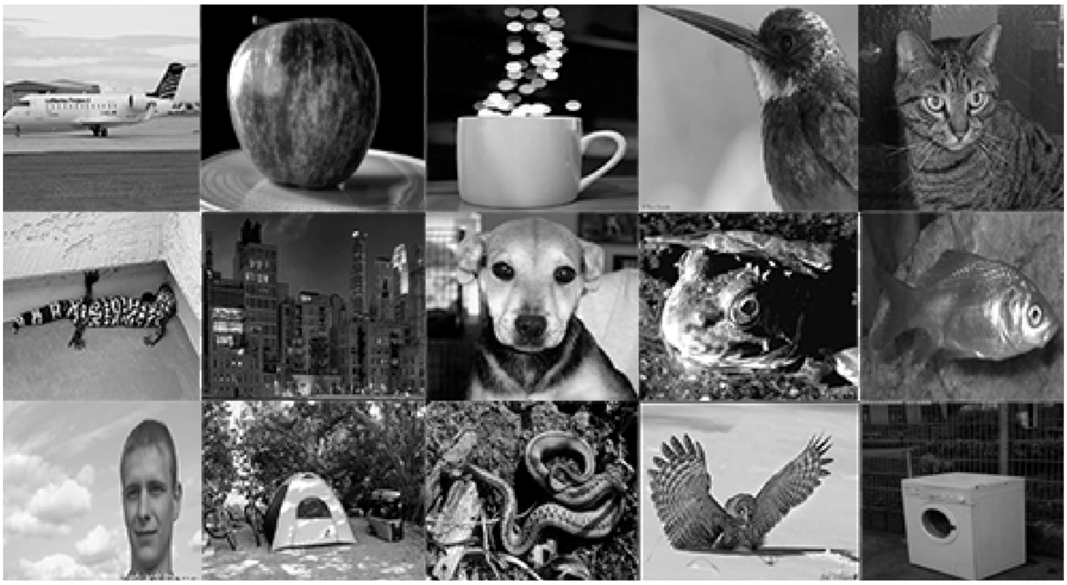

Fig. 5. Small dataset

3.2 Neural Network Training Methods

Because of the depth problem of the pre-training network and the lack of the number of samples, a smaller sample dataset can easily produce the problem of over-fitting [14]. So when training on a small sample dataset, we use the following two methods:

First, add appropriate redundant pictures during training.

Secondly, discard the method of training the fully-connection layer, and the global average pooling method is used to output 15 feature maps, finally obtain their weighted average value. These averages are the confidence probability of the corresponding category, and then enter the output category information in the classifier. Because the number of parameters is greatly reduced, the method of using feature graphs to represent the confidence graph of a class directly reduces the over-fitting effect to some extent.

The fully-connection layer approach is still used when training on the Caltech dataset. The fully-connection layer re-compresses the high-dimensional data from the previous convolutional neural network into a vector form. Operations such as the convolutional layer, the pooling layer, and the activation function map raw data to the hidden feature space, and the fully-connection layer maps the learned distributed feature representation to the sample tag space.

3.3 Neural Network Setup

The softmax activation function is used for multi-classification problems. It performs an equal-dimensional shrinkage projection. Each element in the new vector group also becomes the real number between (0, 1). The common loss function of softmax is Categorical.Cross-entropy, that is, multi-class logarithmic loss function.

In the two stages of this experiment, because the target task is different, we use different learning rate algorithms. The RMSProp algorithm is used to train the new fully-connection layer, and the SGD (stochastic gradient descent) algorithm is used to fine-tune the weights when layer freezes. Fine-tuning should be done at very low learning rates. The method of using random gradient descent is more effective. This is to ensure that the weight of the update has a slower speed, so as to avoid excessive changes and destroy the features learned before.

In traditional machine learning, by evaluating the performance of multiple machine learning models on the same sample, look for models that perform well and combine them to improve overall performance. In deep networks, dropout makes a neuron no longer overly dependent on other neurons that are closely connected to it, but learns to work with other neurons in the entire network. Through the method of random elimination, the joint adaptability of local neuron clusters is reduced, and the generalization ability of the network is enhanced to some extent.

4 Experimental Results and Analysis

4.1 Data Augmentation Sample

The result of data augmentation basically ensures that the same pixel does not appear in the same area, avoiding over-learning of a certain feature, thereby minimizing the local over-fitting effect of the neural network. At the same time, we should pay attention to the logic when the data augmentation, try to make the results conform to the observation angle in daily life. If there is an unconventional angle, it may turn into harmful noise to the model, which will interfere with the convergence and accuracy of the model.

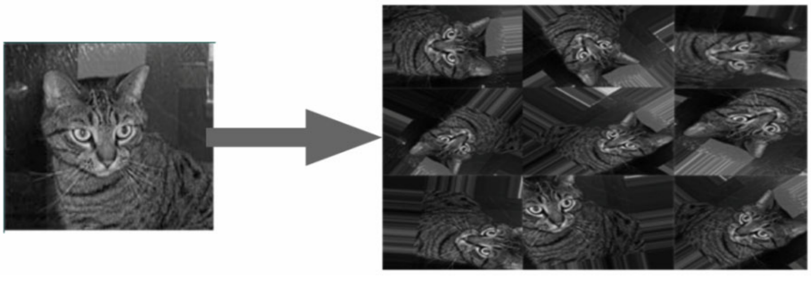

Fig. 6. Source sample (left) and data expansion sample (right)

4.2 Xception Network Training Effect

The results below take the accuracy and loss function values on a data set as a set of images, while a dataset is also divided into the results of the pre-tuning network and the fine-tuned network.

It can be seen from the Fig. 7 that the accuracy on small dataset can reach around 80% by 30 iteration updates. In the layer freeze, the accuracy can exceed 90% by fine-tuning the last convolution block. The experiment shows that the methods we used above, including the addition of redundant images, data expansion, regularization and other means can effectively prevent over-fitting. However, by observing the trend of the curve, the robustness of the network is poor during training. The curve does not rise slowly, but has a higher starting point, and there are many violent shocks in the process. This is related to the prior knowledge in the pre-training model. Many categories can be identified when the network is pre-trained, and their characteristics have been memorized by the network. However, in the one-shot learning, due to the rarity of the data, the weight of some features is magnified or ignored, which leads to the collision and conflict between the knowledge of existing networks and prior knowledge. After a certain iteration, the curve rises slowly and the trend tends to be stable.

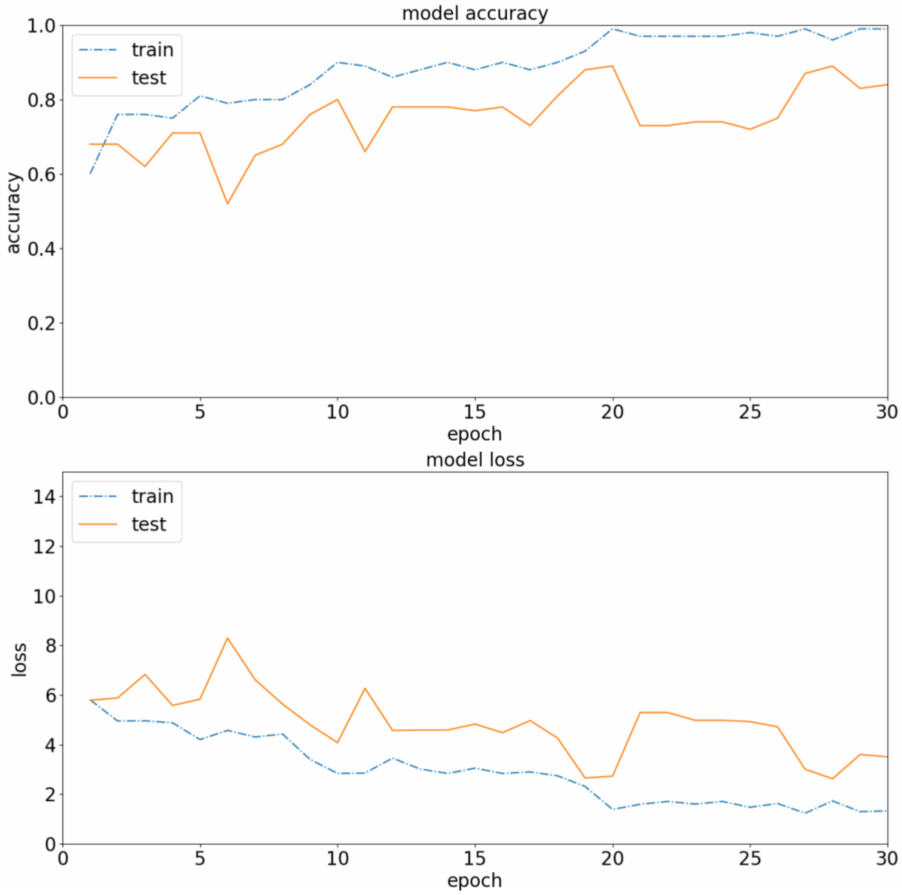

Fig. 7. Verification accuracy (top) and loss function before fine-tuning on small datasets (bottom)

Table 2. Small dataset training effect

	Iteration round	Training set accuracy	Verification set accuracy	Over-fitting rate
Before fine-tuning	30	0.99	0.78	1.270
After fine-tuning	30	0.96	0.91	1.055

In order to make the training features more diverse, we can fine-turn on large datasets. The process of fine-tuning from the original task set to a specific task set is a task-oriented process for generalization features. The underlying features are mostly basic geometries such as edges and corners, while high-level features may have some special changes that directly reflect specific tasks. When the category increases, the time spent on fine-tuning will increase greatly. It costs more than just training a new fully-connection layer, so the fine-tuning process can also be omitted on the mobile terminal.

As we can see from the Fig. 8, the accuracy of the Caltech-101 dataset is around 75% by 120 iteration updates. While after the layer freezes, accuracy can reach about 80% by fine-tune the last convolution block. Compared with small datasets, not only is the accuracy of the classification decreased, but the effect of fine-tune also decreased significantly, with a growth rate of around 5%. As we can see from the curve, when the number of categories increases, there is no such situation as a sharp concussion on a small dataset.

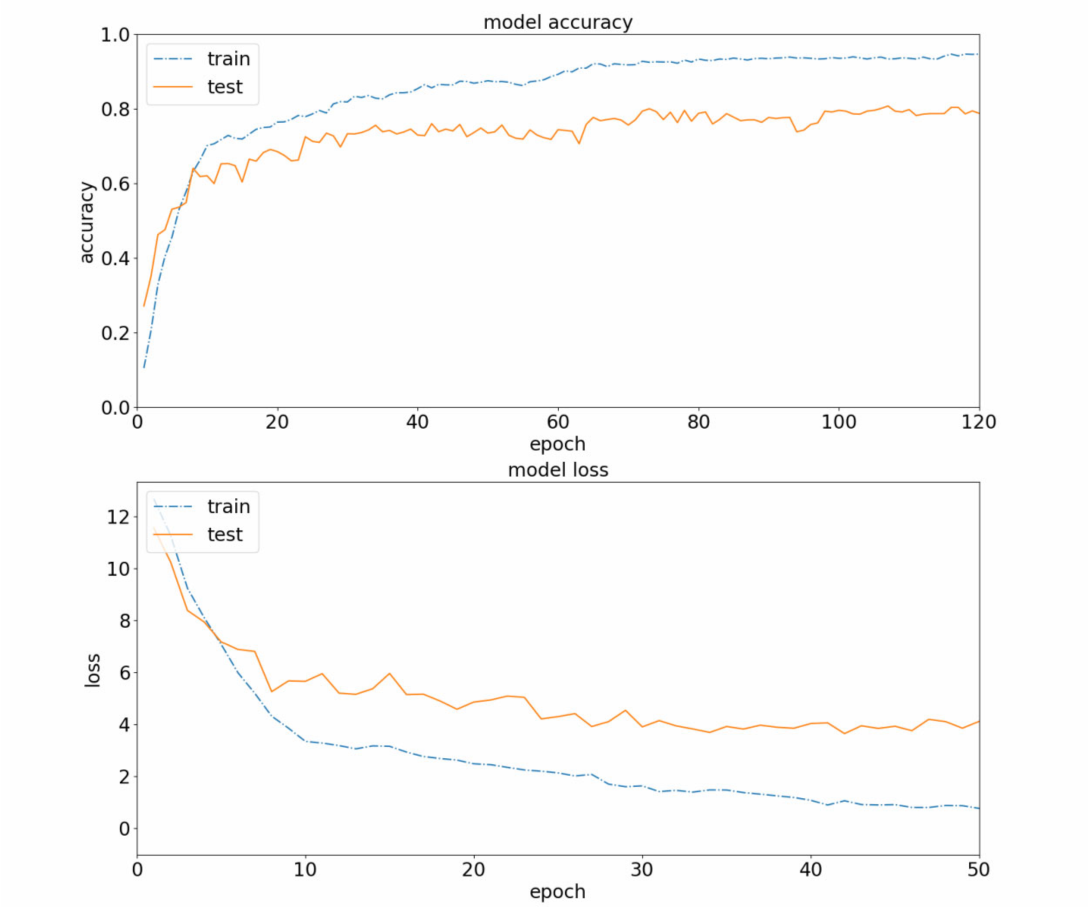

Fig. 8. Verifies accuracy before fine-tune on the Caltech-101 datasets (top) and loss function (below)

The curve in the figure shows that at the same time as the dataset increases, the accuracy decrease is more obvious. Analyze the reasons: first, in order to speed up training, no particularly large fully-connection layer was built when training on the Caltech dataset. Compared to the VGG network, which has two layers of 4096-node fully-connection layers, it uses only a single layer with a 256-node fully-connection layer. This is slightly smaller for datasets with 101 classes; secondly, the number of convolution blocks frozen does not change during fine-tune, which means that the increase in the number of categories does not change the number of parameters. In theory, freezing more layers will have a better training effect.

Table 3. Caltech101 dataset training effect

	Iteration round	Training set accuracy	Verification set accuracy	Over-fitting rate
Before fine-tuning	120	0.95	0.75	1.267
After fine-tuning	120	0.96	0.81	1.185

5 Conclusion

One-shot learning method based on convolutional neural network is presented in this paper to solve the insufficient of large-scale data processing ability for intelligent robot. The method utilized the data augmentation techniques to construct a small target dataset with a single sample. Migrate the pre-training model and better fit the target domain by fine-tuning. Freezing the underlying simple geometric shape features using the layer freezing technology. The training and fine-tune will only be performed on the fully-connection layer and the last few convolutional networks which preserve the combined features. The experimental results show that the method has high accuracy and flexibility in the single-sample classification problem, meanwhile, can save a lot of time and computing resources. It is suitable for mobile robots and other mobile terminal equipment.

Acknowledgement

This work is supported by Cross-Training Plan of High Level Talents and University Student Innovation Science and Technology Project of Beijing.

References

[1] G. Montavon, W. Samek, K. Muller, Methods for interpreting and understanding deep neural networks, Digital Signal Processing 73(2018) 1-15.

[2] F. Sung, Y.X. Yang, L. Zhang, T. Xiang, P.H.S. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, in: Proc. Computer Vision and Pattern Recognition (CVPR) 2018.

[3] K.C. Peng, Z.Y. Wu, J. Ernst, Zero-Shot deep domain adaptation, in: Proc. European Conference on Computer Vision, 2018.

[4] J. Lv, X. Shao, J. Huang, X. Zhou, X. Zhou, Data augmentation for face recognition, Neurocomputing 230(2017) 184-196.

[5] Y.X. Wang, M. Hebert, Learning to learn: model regression networks for easy small sample learning, in: Proc. European Conference on Computer Vision, 2016.

[6] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks., in: Proc. International Conference on Machine Learning, 2017.

[7] K. Banerjee, T.V. Dinh, L. Levkova, Velocity estimation from monocular video for automotive applications using convolutional neural networks, in: Proc. IEEE Intelligent Vehicles Symposium, 2017.

[8] J. Snell, K. Swersky, R.S. Zemel, Prototypical networks for few-shot learning, in: Proc. Advances in Neural Information Processing Systems, 2017.

[9] Q.S. Zhang, Y.N. Wu, S.C. Zhu, Interpretable convolutional neural networks, in: Proc. Computer Vision and Pattern Recognition (CVPR), 2018.

[10] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: Proc. Computer Vision and Pattern Recognition (CVPR), 2017.

[11] Z. Zhen, P. Gao, S.L. Sun, License plate recognition system based on transfer learning, in: Proc. International Conference Signal and Information Processing, Networking and Computers, 2018.

[12] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, D. Quillen, Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, in: Proc. The International Journal of Robotics Research, 2018.

[13] E. Hoffer, R. Banner, I. Golan, D. Soudry, Norm matters: efficient and accurate normalization schemes in deep networks, in: Proc. Neural Information Processing Systems, 2018.

[14] L. Bottou, F.E. Curtis, J. Nocedal, Optimization methods for large-scale machine learning, Siam Review 60(2)(2018) 223-311.