ASM Sc. J., 13, Special Issue 1, 2020 for ICFICE, 114-122

An Improved Deep Learning Method for Classification of Animal Image Patches Based on Small Datasets

Guangxing Wang[1*] and Seong-Yoon Shin[2]

[1]*Department of Information Technology Center, Jiujiang University*
Jiujiang 332005, China
[2]*School of Computer Information and Communication Engineering, Kunsan National University*
Gunsan 54150, Korea

In recent years, convolutional neural networks (CNNs) technology based on deep learning method has made remarkable achievements in the field of computer vision, mainly applied to face recognition, image classification, natural language processing, and so on. However, CNNs require sufficient data samples for training to improve prediction accuracy. If the necessary optimization measures are not taken for small data sets, the prediction accuracy of the CNN often fails to achieve the expected goal. This paper proposes an improved deep learning method based on small data sets for animal image classification. First, a CNN to build a training model for small data sets is used, while data augmentation is utilized to expand the data samples of the training set. Second, using the pre-trained network on large-scale datasets, such as VGG16， the bottleneck features in the small dataset are extracted to be stored in two NumPy files as new training datasets and test datasets, finally training a fully connected network with the new datasets. In this paper, Kaggle famous Dogs vs Cats dataset is used as the experimental dataset, which is a two-category classification dataset. On this dataset, CNNs were performed and CNN prediction experiments were improved. The experimental results show that the improved method has a prediction accuracy of 92%, the prediction time is greatly shortened, and the improved method can be extended to the multi-classification case.

Keywords: deep learning; convolutional neural network; VGG16; data augmentation; Kaggle Dogs vs Cats dataset

I. INTRODUCTION

In recent years, convolutional neural networks (CNNs) technology based on deep learning method (Lemley *et al.*, 2017) has made remarkable achievements in the field of computer vision, mainly applied to face recognition, image classification, natural language processing, and so on (Krizhevsky *et al.*, 2012; Karen Simonyan & Andrew Zisserman 2014; Christian Szegedy *et al.*, 2015; Kaiming He *et al.*, 2016). However, CNNs require sufficient data samples for training to improve prediction accuracy (Joseph Lemley, 2017). In particular, if the necessary optimization measures are not taken for small data sets, the prediction accuracy of the CNN often fails to achieve the expected goal. In order to overcome the problems of overfitting and low prediction accuracy caused by insufficient training samples, the commonly used method is Dropout (Srivastava *et al.*, 2014) and Batch Normalization (BN) (Ioffe and Szegedy, 2015). Data augmentation is a widely used method of augmenting data sets (Simard, 2003). For image classification, data augmentation is based on an image processing process; i.e., a process of generating a corresponding series of new images by performing a series of operations such as shifting, cropping, scaling, and flipping the original image (Richard M. Zur, 2009). This method is commonly used in convolutional neural networks to extend the training set for training, thereby improving prediction accuracy and suppressing overfitting (Kingma *et al.*, 2014).

*Corresponding author's e-mail: wanggxrs@163.com

Another way to improve prediction accuracy is by using networks that are streamlined for large-scale data sets. Such networks can achieve good features on most computer vision problems. Using such features to predict new samples can not only achieve higher accuracy, but also greatly shorten training time and improve efficiency (Karen Simonyan & Andrew Zisserman, 2014; Christian Szegedy *et al.*, 2015; Kaiming He *et al.*, 2016). Typical representatives of trained CNNs on large-scale datasets are AlexNet, VGG, ResNet, GooLeNet, and others. These pre-networks can be used not only for the prediction of two-category samples, but also for the prediction of multi-class samples, and the classification results are excellent. However, when these pre-trained networks are used directly, the degree of improvement in prediction accuracy is not significant. Through the pre-processing of small data, the improved network prediction accuracy can be greatly enhanced, and the training time is greatly shortened.

In 2013, Kaggle released the Dogs vs Cats image classification algorithm competition which attracted the attention of the academic community (Kaggle, 2013). At that time, most of the participating teams used the Support Vector Machine (SVM) algorithm of machine learning method, k Nearest Neighbor (kNN) algorithm, Decision Tree (DT) algorithm, and other methods to predict the classification problem of dog and cat images, achieving good results. Although the competition is over, discussions and algorithm improvements for the two-category problem continue. In particular, the research and development of deep learning CNNs in recent years, using CNNs to solve the two-category classification problems, can achieve better results. Based on the Kaggle Dogs vs Cats dataset as an example, this paper proposes an improved CNN prediction method. First, a CNN is used to build a training model based on small datasets, and data augmentation techniques are employed to suppress the overfitting of the training results by improving the number of data samples in the training set. Second, using the pre-trained network on large-scale datasets, such as VGG16，the bottleneck features in the small dataset are extracted and stored in two NumPy files as new training datasets and test datasets. Finally, a fully connected network is activated with the new datasets. Based on the experimental dataset, the experimental results are compared with the improved methods. The experimental results show that the improved CNN method has a prediction accuracy of 92% and the prediction time is greatly shortened.

The remainder of this paper is organized as follows. Section II introduces related work and experimental dataset. Section III explains the details of the proposed method. Section IV presents the experimental results. The last chapter contains conclusions and future work.

II. RELATED WORK

Due to the characteristics of feature learning and highly nonlinear transformation, the deep learning model has a strong potential to acquire the complex patterns of data in data processing. In order to achieve better generalization capabilities, the deep learning model relies on the availability of a large amount of training data. Therefore, since deep learning has become popular in computer vision, many well-marked image datasets have been introduced (Deng *et al.*, 2009; Zhou *et al.*, 2014). However, for small datasets, the sample data used to train the network is limited, the prediction results are not accurate, and over-fitting occurs due to too small data volume. Therefore, data augmentation, regularization, and the Dropout methods are often used to suppress the over-fitting of prediction results.

Data augmentation is the process of generating a new image containing features of the original image, such as offset, crop, zoom, rotate, mirror, etc., by a series of transformations on the original image. In this way, the number of dataset samples can be increased further (Simard *et al.*, 2003). Regularization is also known as weight decay. Commonly used regularizations have L1 and L2 regularizations, and L1 can obtain more sparse parameters than L2. Regularization can reduce the dimensionality and complexity of the data and facilitate the convergence of the prediction results (Bishop, 1995). Dropout is a more effective method to prevent overfitting. In the process of CNN training, the weight of some neurons is reset to 0; that is, some neurons are invalidated, which can reduce parameters and avoid overfitting (Srivastava *et al.*, 2014). The methods used to eliminate overfitting and improve accuracy are different for different classification problems and application areas. On the other hand, predicting samples of small datasets by using pre-trained networks on large-scale datasets is undoubtedly a sensible choice in the current research on image classification and object classification detection. This method can not only improve the accuracy of prediction, but also shorten the prediction time and greatly improve the efficiency of data processing. For example, the VGG16 convolutional neural network is used in this paper to solve the two-category problem, which can greatly improve the prediction accuracy, with the data processing time greatly shortened.

The dataset used in this paper is Kaggle Dogs vs Cats Dataset. It is a competition of the 2013 Kaggle Big Data Competition, which uses algorithms to achieve dog and cat identification using a given dataset including cat and dog images. The dataset consists of training data and test data. The training dataset contains 12,500 images for each of the dogs and cats. The test dataset contains images of 12,500 dogs and cats. These images vary in size, background, and posture. The training dataset picture name contains the dog or cat label, and the test dataset picture name does not contain the dog or cat label. Therefore, the designed model must first be trained through the training data set, and then tested with the test data set to further verify the accuracy of the model classification. In this paper, 1,000 pictures of each dog and cat set are randomly selected from the training dataset to create the experimental training dataset, whereas 400 pictures for each of the dog and cat sets from the test dataset were selected to compose the experimental test dataset.

III. PROPOSED METHOD

In this chapter, the understanding of convolutional neural networks and brief introduction of the basic definition and geometric meaning of convolution are the focus. What follows is the basic structure of the convolutional neural network and the structure of the well-known large-scale data set training model VGG16 convolutional neural network. Combined with the research of this thesis, the model design of the two-category classifications and the regularization of the model are presented in detail. The model regularization methods used mainly include data augmentation and dropout.

A. Convolutional Neural Networks

Before introducing the CNN, the focus is on the concept of convolution. Convolution is an important operation in analytical mathematics; that is, a mathematical operator that generates a third function through two functions f and g. Its geometric meaning is the area of the overlap of the functions f and g through flipping and translation. Convolution can be defined as:

let f(x) and g(x) be two integrable functions in the R1 space, with the integral operation performed, as in (1).

$$S(x) = \int_{-\infty}^{\infty} f(\tau)\, g(x-\tau) d\tau \qquad (1)$$

It can be shown that for almost all real numbers or x, the above integral exists. Thus, with the different values of x, this integral defines a new function S(x), which is called the convolution of the function f and g, which is recorded, as in (2).

$$S(x) = (f * g)(x) \qquad (2)$$

Here, "*" means convolution, which is a matrix form of convolution, converted into discrete forms, as in (3).

$$S(x) = \sum_{\tau} f(\tau) g(x-\tau),\ \ \tau \in (-\infty, \infty) \qquad (3)$$

It is easy to verify that (f*g)(x)=(g*f)(x) and (f*g)(x) is still an integrable function. This means replacing the multiplication with convolution (Liming, 2012). Equation (3) represents the result of multiplying two functions f(τ) and g(τ) and summing them in the interval τ ∈ (-∞, ∞). When x = 0, g(-τ) is the result of the inverse of τ by g(τ). Inverting makes g(τ) flip 180 degrees around the vertical axis, so the calculation method of summation after multiplication is called convolution sum, which is simply called convolution. Further, x is an amount by which g(-τ) is displaced, and different x corresponds to a different convolution. Let X = x(τ) = f(τ), W = w(x - τ) = g(x - τ), then the two-dimensional convolution obtained by equation (3), as is (4).

$$\begin{aligned} S(i,j) &= (X * W)(i,j) \\ &= \sum_{m}\sum_{n} x(i-m, j-n) w(m,n) \end{aligned} \qquad (4)$$

The convolution formula in convolutional neural networks is slightly different from the definition in a strictly mathematical viewpoint. For example, for two-dimensional convolution, the convolution in a convolutional neural network is defined, as in (5).

$$s(i,j) = (X * W)(i,j) = \sum_{m}\sum_{n} x(i+m, j+n) w(m,n) \qquad (5)$$

Here, X denotes an input, and W denotes a convolution kernel. If X is a two-dimensional input matrix, then W will be a two-dimensional matrix. If X is a multidimensional tensor, then W is also a multidimensional tensor. Based on the mathematical knowledge of the above convolution, the convolutional neural network is introduced next.

As a part of machine learning, deep learning methods have achieved remarkable results in the field of computer vision, especially in complex data processing, such as object recognition, image classification, and natural language processing. CNNs based on deep learning methods are inspired by the human visual system (LeCun *et al.*, 1998; Fukushima, 1980). The application of convolutional neural networks based on deep learning methods is increasingly widespread (Schmidhuber, 2015; Deng and Yu, 2014). AlexNet network technology, one of the CNNs, won in the 2012 image network competition (Krizhevsky *et al.*, 2012). Since then, CNN technology has been widely used to solve various image segmentation and classification problems. Unlike other classification algorithms, CNN algorithms include feature extraction and classification. A basic convolutional neural network model consists of five parts: input layer, convolutional layer, pooling layer, full connection, and output layer, as shown in Figure 1. The input layer is responsible for processing the input image into a uniform size format. The processed image is sent to the convolutional layer for feature learning and convolution processing using shared weights to extract significant features of the image. The pooling layer then splits the obtained image and tries to retain the main feature information. Next, the extracted feature information is given weights and connected at the fully connected layer.

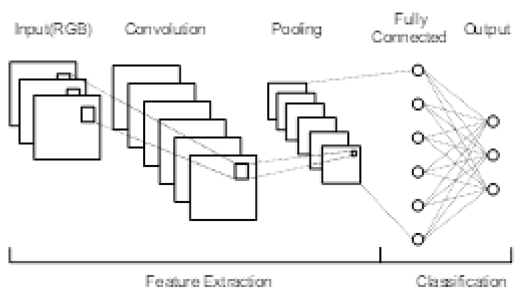

Figure 1. Convolutional block structure of a CNN (CNN consists of five parts, with feature extraction and classification performed during model training)

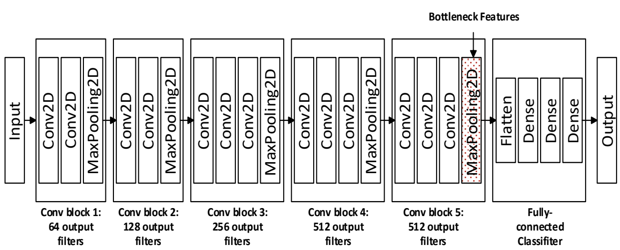

Figure 2. VGG16 CNN structure diagram

Finally, an output neuron image corresponding to the classification is output at the output layer. In the five parts of the CNN, the input layer, the convolutional layer, and the pooling layer are the processes of extracting the feature information of the image. The subsequent fully connected layer and output layer are used to perform image classification operations. Based on the CNN, neural network models such as VGG16 and VGG19 are derived. The VGG16 model is a CNN model with five convolutional blocks and a fully connected layer, as shown in Figure 2. In the largest pooling layer of the fifth convolution block, the bottleneck features of the input data are extracted. The last layer is the fully connected layer.

B. Model Design

The CNN mentioned above can be expressed as follows:

$$\begin{aligned} IN \; &=> \; CONV - ReLU \; => \; POOL \\ &=> \; FC - ReLU \; => \; OUT \end{aligned} \tag{6}$$

Here, IN denotes an input layer, CONV denotes a convolutional layer, and ReLU (Rectified Linear Units) denotes an activation function derived from a neural network, which can greatly shorten the learning period and improve learning efficiency. POOL represents the pooling layer, FC represents the fully connected layer, OUT represents the output layer, and "-" means follow. The stacking of these levels constitutes the convolutional neural network structure in deep learning. In practical applications, there are multiple CONV and POOL layers in the convolutional neural network structure, which are designed to reduce image size and extract finer features. It is then classified using the fully connected layer and finally output at the output layer. Therefore, the expression of the commonly used convolutional neural network model is as shown in (7).

$$\begin{aligned} IN \; &=> \; [CONV - ReLU \; => \; POOL?] * M \\ &=> \; [\,FC] * N - ReLU \; => \; OUT \end{aligned} \tag{7}$$

Here, "*" indicates repetition, and "?" indicates an optional pooling layer. In theory, the more layers of convolution and pooling (the larger the M value) there are, the more fine-grained input image features can be obtained, but this requires a lot of learning time. In this paper, fewer convolution and pooling layers are chosen, and data augmentation methods are combined to improve the model's production efficiency (M=3 and N=2 are selected). Thus, the feature extraction part of the convolutional neural network model consists of three convolutional blocks consisting of three convolutional layers and one pooling layer. The classification part consists of one fully connected layer and an output layer. Each convolutional layer and fully connected layer are followed by an activation function or ReLU.

The proposed method is to use the VGG16 CNN to extract the bottleneck feature by using the convolutional layer part of VGG16 after removing the fully connected layer and saving the extracted features in the NumPy (NPY) file of the training set and the NPY file of the test set. The new feature set file is then used to create a fully connected network. The expression of the optimized CNN model is as illustrated in (8) and (9).

$$IN => [CONV - ReLU => POOL] * M \\ => OUT^{\#} \tag{8}$$

$$IN^{\#} => [FC] * N - ReLU => OUT \tag{9}$$

In these formulae, the "#" mark indicates an NPY file with a bottleneck feature NPY file, which is a vector group file with image feature vectors. Let M=5 and N=3, so the expression (8) indicates that the convolutional layer part of the VGG16 CNN is used, and the top three connected layers are removed to train the data (the input is the bottom layer, the top layer is the output). The bottleneck feature in the largest pooling layer of the fifth layer is extracted, and finally the obtained bottleneck feature data in two NPY files are saved. General speaking, only the convolution block of the VGG16 network is used except the fully connected layer to extract the bottleneck feature. Expression (9) represents the use of an NPY format file that extracts the bottleneck feature data as an input file, trains a fully connected network, predicts new NPY test samples, and outputs the results.

C. Model Regularization

In this paper, because a very small dataset is used consisting of a training data set of 2000 images and a test data set of 800 images. Therefore, a data augmentation technology is needed to extend the training set, and then a dropout method is necessary to reduce the parameters and suppress the overfitting again in the fully connected phase of the improved CNN.

Data Augmentation: Data augmentation is a method that can effectively suppress overfitting. The method can generate new samples with original image data features by performing operations such as rotation, scaling, shifting, mirroring, etc. within a certain range of values of the original image, thereby achieving the purpose of increasing the number of data samples in the training set. The data augmentation method has certain universal applicability. The data augmentation is shown in Figure 3, in which the image size is uniformly processed to 150 × 150 pixels.

In this paper, a random transformation is used to augment the data of the training samples. The parameter configuration table of the data augmentation random transformation is shown in Table 1. The original image is randomly rotated over a range of 40°. The width and height of the image are randomly shifted within a range of 20%. Random trimming is performed within 20%. Random scaling is performed over a 20% range. The image is randomly flipped horizontally and, finally, it is filled with the nearest model. After a series of random transformations, the data set samples with data augmentation are trained. After the data-augmented training samples, each picture becomes different.

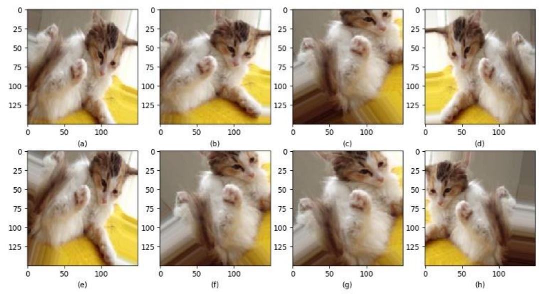

Figure 3. Data augmentation example: (a) original image, (b) horizontal offset, (c) vertical offset after rotation, (d) mirror image, (e) cropping, (f) vertical offset, (g) rotation, (h) mirroring and scaling

Table 1. Data augmentation parameters

No.	Parameter	Augmentation
1	Rotation (°C)	40
2	Width_shift (%)	20
3	Height_shift (%)	20
4	Shear (%)	20
5	Zoom (%)	20
6	Horizontal_flip	TRUE
7	Fill_mode	Nearest

Dropout: Data augmentation is only an increase in the number of training samples and to some extent can suppress overfitting. However, due to the lack of diversity in the training samples after data augmentation, it is insufficient to eliminate overfitting in the deep learning model. Therefore, the dropout method is also needed to further eliminate overfitting. The dropout method was proposed by Srivastava *et al.* (2014). It refers to the random reduction of neural units by probability p during the training of neural networks, thereby reducing the number of parameters of the neural network and eliminating the problem of over-fitting of training results. In this paper, the dropout method is applied only at the last fully connected layer and the random probability is set to p=0.5, along with the activation functions ReLU and Sigmoid. The improved fully connected layer network structure is shown in Figure 4, where "Input#" represents an input file with a bottleneck feature. The convolutional neural network and the optimized fully connected layer network structure are shown in Table 2 and Table 3, respectively.

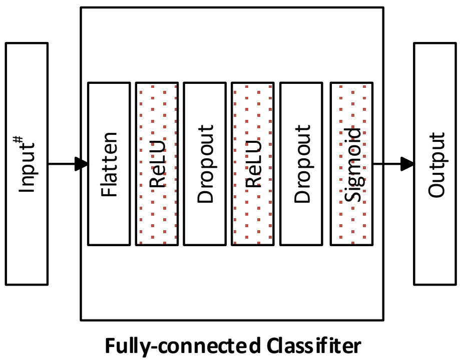

Figure 4. Improved fully connected layer network structure diagram

Table 2. Architecture of CNN

No.	Layer	Output size	Filter/ stride size	Dropout
1	Input	150×150×3	—	—
2	Convolution2D	148×148×32	3×3	—
3	ReLU	148×148×33	—	—
4	Max Pooling2D	74×74×32	2×2	—
5	Convolution2D	72×72×32	3×3	—
6	ReLU	72×72×32	—	—
7	Max Pooling2D	36×36×32	2×2	—
8	Dropout	36×36×32	—	0.25
9	Convolution2D	34×34×64	3×3	—
10	ReLU	34×34×64	—	—
11	Max Pooling2D	17×17×64	2×2	—
12	Flatten	1×1×18496	—	—
13	Fully connected	1×1×64	—	—
14	ReLU	1×1×64	—	—
15	Dropout	1×1×64	—	0.5
16	Fully connected	1×1×65	—	—
17	Sigmoid	1×1×65	—	—

Table 3. Architecture of implemented CNN fully connected layer

No.	Layer	Output size	Dropout
1	Fatten	1×1×8192	—
2	Fully connected	1×1×256	—
3	ReLU	1×1×256	—
4	Dropout	1×1×256	0.5
5	Fully connected	1×1×256	—
6	ReLU	1×1×256	—
7	Dropout	1×1×256	0.5
8	Fully connected	1×1×257	—
9	Sigmoid	1×1×257	—

IV. EXPERIMENT

In this section, small datasets will be used to train and test convolutional neural networks and improved convolutional neural networks, respectively. The experimental results will be compared based on prediction accuracy, loss rate, and training time.

A. Experimental Environment and Procedure

The experimental hardware environment is a desktop computer that can connect to the Internet and is equipped with Intel Core i7-4790, with the graphics processor NVIDIA GeForce GTX 960 equipped with 2GB memory. The computer's RAM memory is 8GB, and the hard drive capacity is 2TB. The software environment is developed using the Python programming language and the Keras deep learning library is installed with TensorFlow as backend.

The experimental steps are divided into data pre-processing, fully connected network training and testing, classification, and output. First, the data augmentation technique is used to expand only the training set, and the picture is uniformly processed into a size of 150×150 pixels. The bottleneck feature of the training sample and the test sample are extracted by using the convolution part of the VGG16 network except for the fully connected layer while the NPY format files are saved as training and test sets, respectively. Second, the newly generated NPY format file is used, containing the features of the bottleneck feature as an input file, training a fully connected network, and adding the dropout method and the ReLU activation function to the fully connected layer to suppress overfitting. Finally, the training model is used to predict the NPY format file of the test set sample containing the bottleneck feature, while the two-category activation function Sigmoid output is used to obtain the prediction accuracy and loss rate.

B. Experimental Results and Discussions

All experimentation in this research is carried out with Kaggle Dogs vs Cats dataset. Experiments were carried out before and after the improvement of the CNN. A CNN batch size of 32 image is used, and the RMSProp optimizer is used, with 100 steps per iteration and a total of 30 epoch, the loss rate and the best accuracy of the test being 0.4423 and 0.8094, respectively. Training and prediction total time is 3274.3s. The improved CNN is configured with an iterative batch size of 16, and each epoch being 64 iterations in 50 epochs. The loss rate and the best accuracy of the obtained test are 1.3210 and 0.9175, respectively. In addition, the data pre-processing time is 333.2s. Training and prediction time is 253.5s. The loss rate and accuracy statistics are shown in Figure 5 and Figure 6, respectively. It can be concluded from the experimental results that the CNN prediction results are overfitting, and the accuracy of the test is only 0.8094, which takes a long time. The improved method test results are well-converged, the test accuracy is up to 0.9175 (an increase of 13.4%), and the training and testing speed is greatly accelerated, with the time consumption greatly reduced, only accounting for 18% of the time before the improvement. The improved method test-loss rate is improved compared to before the improvement, but it is generally convergent and much better than before the improvement.

In the above experiment, the trained model is saved in the training of small data sets. The model is called to predict the test and obtain the actual forecast. Eight example pictures of cats and dogs are randomly extracted from the test dataset, respectively, calling the CNN training model and the improved model for prediction. The predicted results are shown in Table 4. It can be seen from the classification prediction results that the CNN model prediction results are generally low. For example, Picture No. 4 predicts that the animal type probability of the cat is 5%, but the number of pictures with the No. 8 is 100%. The improved CNN model predicts better results, and the prediction probability is over 68%, preferably 71.9%. Therefore, it is shown that the model trained by the improved method has much better prediction results for animal pictures than before the improvement. Of course, the prediction probability of the CNN model is not high and may be related to the background of the image selected and the posture of the animal. This is also the influencing factor mentioned earlier. The reason that the prediction result of the CNN model is not ideal may be that the selection of the sampled picture of this prediction is not highly appropriate, or the number of samples selected is too small. However, the prediction results for the improved CNN model are generally better and, therefore, the improved CNN method has higher prediction accuracy for the two-category problem.

The above experimental results verify the two following aspects. First, the data augmentation and dropout methods can effectively suppress overfitting and improve the accuracy of prediction results. Second, training and predicting new samples using neural network models trained on large-scale data sets can greatly reduce the time for training and predicting samples, improve work efficiency, and further improve the accuracy of prediction. The experimental results are a good proof that using a pre-trained convolutional neural network model on a large-scale data set is a good method if one wants to use the deep learning method to predict a limited number of training samples. It can greatly improve the accuracy of prediction and the work efficiency.

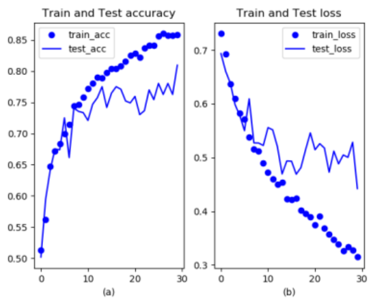

Figure 5. CNN experiment results: (a) Accuracy statistics for training and testing, (b) Loss rate statistics for training and testing

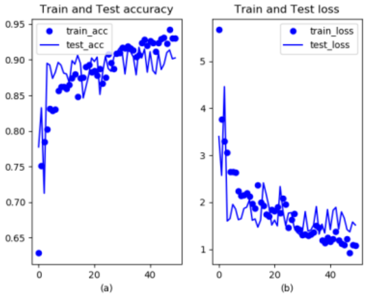

Figure 6. Improved CNN experiment results: (a) accuracy statistics for training and prediction, (b) loss rate statistics for training and testing

Table 4. Comparison between CNN training model and improved model classification prediction results

No.	**Test images**	**CNN method** Classification prediction & probability (%)	**Proposed method** Classification prediction & probability (%)
1		cat: 6.8	cat: 70.7
2		cat: 20.3	cat: 68.3
3		cat: 36.6	cat: 69.6
4		cat: 5	cat: 71.9
5		dog: 83	dog: 68.9
6		dog: 73.3	dog: 69.2
7		dog: 6.8	dog: 70.2
8		dog: 100	dog: 70.1

V. CONCLUSION AND FUTURE WORK

This paper presented an improved deep learning method based on small datasets for animal image classification. A two-category experiment was performed in the Kaggle Dogs vs Cats dataset. The Kaggle Dogs vs Cats dataset consists of 25,000 training sets for pictures of dogs and cats with labels, and a 12500-test set for pictures of dogs and cats without labels. 1,000 pictures were randomly selected from each of the training sets, the experimental training set was built, and 400 experimental samples were selected randomly from each of the test sets. First, the experimental training set was extended by using data augmentation technology. The convolutional layer of the VGG16 CNN was used to extract the bottleneck feature from experimental training dataset and the experimental test dataset and saved as the NPY format file of the training set and test set, respectively. The pre-processing operation of the data was completed. Second, the generated training dataset of NPY format file was used as an input file to train a fully connected network, and the dropout method and the ReLU activation function were added in the fully connected layer to suppress the overfitting of prediction results, reduce parameters, and improve prediction accuracy. Finally, the training model was used to predict the NPY format file test dataset and compared with the test results of the CNN.

The experimental results show that the improved CNN has a prediction accuracy of up to 92%, which is 13.4% higher than before the improvement. The time taken was shortened to 253.5.s, which accounted for only 18% of CNN training and testing time; the production efficiency was also greatly improved. Through the prediction and classification experiments of some pictures, it can be concluded that the classification effect of the improved method is generally better than CNN. Of course, the accuracy of the improved CNN prediction can be further improved. It can also be improved from the network structure of the CNNs, the parameter setting of the convolution block, and the more flexible application of the dropout method to improve the prediction accuracy and reduce the loss rate. These are the contents that may be studied in the future.

VI. REFERENCES

Alex Krizhevsky,Ilya Sutskever,Geoffrey E. Hinton 2012, ‘ImageNet Classification with Deep Convolutional Neural Networks’, *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp.1-9.

B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva 2014, ‘Learning deep features for scene recognition using places database’, in *Advances in neural information processing systems*, pp. 487-495.

C.M. Bishop 1995, ‘Training with noise is equivalent to Tikhonov regularization’, *Neural computation*, vol. 7, no. 1, pp. 108-116.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich 2015, ‘Going Deeper With Convolutions’, *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

D.P. Kingma, D.J. Rezende, S. Mohamed, and M. Welling 2014, ‘Semisupervised Learning with Deep Generative Models’, *In Advances in Neural Information Processing Systems 27*, pp.3581–3589.

J. Deng, W. Dong, R. Socher, L.-J.Li, K. Li, and L. Fei-Fei, 2009, ‘Imagenet: A large-scale hierarchical image database’, in *Proceedings of the IEEE Conference on Computer Visio and Pattern Recognition*, pp. 248–255.

J. Lemley, S. Bazrafkan, and P. Corcoran 2017, ‘Deep learning for consumer devices and services: Pushing the limits for machine learning’, artificial intelligence, and computer vision,” *IEEE Consumer Electronics Magazine*, vol. 6, no. 2, pp. 48-56.

J. Schmidhuber 2015, ‘Deep learning in neural networks: An overview’, *Neural networks*, vol. 61, pp. 85-117.

Joseph Lemley, Shabab Bazrafkan and Peter Corcoran 2017, ‘Smart Augmentation Learning an Optimal Data Augmentation Strategy,’ *IEEE Access*, vol. 5, pp. 5858-5869.

K. Fukushima 1980, ‘Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position’, *Biological Cybernetics*, vol. 36, no. 4, pp. 193-202.

Kaggle 2013, Competition of creation an algorithm to distinguish dogs from cats homepage. [Online]. Available: https://www.kaggle.com/c/dogs-vs-cats

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun 2016, ‘Deep Residual Learning for Image Recognition’, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778.

Karen Simonyan and Andrew Zisserman 2014, ‘Very Deep Convolutional Networks for Large-Scale Image Recognition’, *Computer Science*, pp. 1-14.

L. Deng and D. Yu 2014, ‘Deep learning: methods and applications’, *Foundations and Trends in Signal Processing*, vol. 7, no. 3-4, pp. 197-387.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov 2014, ‘Dropout: a simple way to prevent neural networks from overfitting’, *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958.

P.Y. Simard, D. Steinkraus, and J.C. Platt 2003, ‘Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis’, *Proceedings of the International Conference on Document Analysis and Recognition*, pp.958-963.

Richard M. Zur, Yulei Jiang, Lorenzo L. Pesce, and Karen Drukker 2009, “Noise injection for training artificial neural networks: A comparison with weight decay and early stopping,” Medical Physics, pp. 4810-4818.

S. Ioffe and C. Szegedy 2015, ‘Batch normalization: Accelerating deep network training by reducing internal covariate shift’, *Proceedings of Machine Learning Research*, vol. 37, pp. 448-456.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner 1998, ‘Gradient-based learning applied to document recognition’, *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324.

Y. Liming 2012, *Digital Signal Processing*, Mechanical Industry Press, Issued in 2012.