How Does the Data Set and the Number of Categories Affect CNN-Based Image Classification Performance?

Chao Luo[1], Xiaojie Li[1*], Jing Yin[2], Jia He[1], Denggao[1], Jiliu Zhou[1]
[1] Chengdu University of Information Technology, Chengdu, China.
[2] Chongqing University of Technology, Chongqing, China

* Corresponding author. Tel.: 15108373207; email: lixiaojie000000@163.com
Manuscript submitted January 16, 2019; accepted March 8, 2019.
doi: 10.17706/jsw.14.4.168-181

Abstract: Convolution neural network(CNN) has been widely applied in many fields and achieved excellent results, especially in image classification tasks. As we all know, many factors affect the performance of image classification. In particular, the size of training data sets and the number of categories are important factors affecting performance. While for most people, a large number of training data set are difficult to obtain or need to do a classification task with a large number of categories. Thus, we consider two questions of this approach: How does the size of the data set affect performance? How does the number of categories affect performance? In order to figure out these two questions, we constructed two types of experiment: Experiment 1, changing the number of categories and exploring how the number of categories affects performance in image classification task. There are 7 groups experiment performed by increasing the number of categories and performed 5 times experiment in each group (35 times experiment in total). Observe the change in accuracy to analyze the impact of the number of categories on performance. Experiment 2, changing data set size and exploring how the data set size affect performance. For each k-classification experiment, we do 5 groups by increasing the size of the training set. There are 35 groups experiment performed 5 times experiment in each group (175 times experiment in total). Observe changes in accuracy to analyze the effect of data set size on performance. For the CNN-based network, the results of experiment 1 show that the more categories, the worse the performance, and the less categories, the better the performance. In addition, when the number of categories to be classified is large, sometimes better accuracy can be obtained. The results of experiment 2 show that the larger the training set, the higher the test accuracy. When the training data set are insufficient, better results can be obtained. Therefore, in classification experiment, when the data set size is small or the number of categories is large, we can do more experiments and retain the best results. Results of this paper not only can guide us to do experiments on image classification, but also have important guiding significance for other experiments based on deep learning.

Key words: Multi-classification, CNN, ResNet.

1. Introduction

Image classification is the most classical task in machine learning [1]-[3]. It has important significance in many fields, such as object recognition, face recognition, image retrieval, computer vision and so on [4]-[6]. Image classification methods mainly include traditional machine learning methods and deep learning methods. Support Vector Machine (SVM) and K-nearest neighbor(KNN) algorithm are the most commonly used methods in traditional machine learning and have achieved good results[7]-[11]. However, for large

training sets, both SVM method and KNN method have very large computational complexity, which requires a large amount of memory. Moreover, the classical support vector machine algorithm only gives two classes of classification algorithm, but in the practical application, it is generally necessary to solve the classification problem of multiple classes. Therefore, the method based on deep learning is widely used in image classification tasks, and breakthrough progress has been made [12], [13].

In recent years, with the development of hardware technology, especially the improvement of GPU computing power, deep learning has achieved rapid development in the field of image classification, especially convolutional neural networks have a significant effect on improving the accuracy of image classification. Currently, CNN-based image classification networks widely used include AlexNet [14], VGG [15], GoogLeNet [16], ResNet [17]. However, for neural networks, it is well known that if we increase the number of layers, it will lead to gradient dispersion or gradient explosion, so that the accuracy of the training set will decline [18]. It is very harmful for classification network. Therefore, He Kaiming and his team proposed a residual network structure. Residual Network (ResNet) is a well-known CNN-based network structure for image classification [19], [20]. The main advantage of ResNet is that the gradient can be transmitted through a shortcut connection, so that gradient dispersion or gradient explosion will not occur. The network structure can achieve excellent performance. However, in addition to the performance influenced by the network structure itself, the data set and the number of categories are also the influencing factors that cannot be ignored.

For image classification experiments, the number of categories and data set size have a great impact on the performance. In general, large data sets can improve classification accuracy, while small data sets can reduce classification accuracy. The more categories, the lower the accuracy. However, for most people, a large number of training data set are difficult to obtain and need to do a classification task with a large number of categories. In order to figure out the impact of the data set size and the number of categories on classification performance, we propose the following questions: How does the number of categories affect accuracy? When the number of categories is fixed, how does the size of the data set affect accuracy?

In order to answer these questions, we construct two types of multi-classification experiments. Based on the advantages of the ResNet described above, a 32-layer convolution network structure based on ResNet is constructed for image classification in our experiment. The data set we used was based on Cifar10 [21]. It includes 60,000 images for a total of 10 categories (each category with 6,000 images), and 50,000 images are used as training sets, 10,000 images are used as test sets [22]. We reclassified the Cifar10 data set into different size of training data sets.

- For experiment 1, We do 7 groups experiment by increasing the number of categories k ($k \in \{3, \dots, 9\}$). In each group, 5 times experiment were performed, and the average results of the five experiments was taken as the experimental result of the group. A total of 35 experiments were performed. Experimental results of the average accuracy show that the more the categories of classification, the worse the accuracy, the less the number of categories, the higher the accuracy. Furthermore, when the number of categories to be classified is large, sometimes better accuracy can be obtained.
- For experiment 2, for each k ($k \in \{3, \dots, 9\}$), 5 groups experiment were performed by increasing the size of the training set for each k-classification, 35 groups experiment in total. In each group, 5 times experiment were performed and the average results of the five times experiment was taken as the experimental result of the group. There are 175 experiments performed in total. Experimental results of average accuracy illustrate that the larger the training set, the higher the accuracy. However, when the training data set are insufficient, better results can be obtained.

2. Related Work

2.1. Image Classification Based on Convolution Neural Network

As the most successful machine learning algorithm of today, Convolution Neural Network (CNN) has also widely adopted in image classification as the core algorithm [23]-[25]. The difference between deep convolutional neural network and traditional neural network is that convolutional neural network consists of several feature extractors composed of convolution layer and pool layer, and a neuron is only connected with some adjacent neurons [26]. Convolution neural network has the following properties: Firstly, convolution neural network has the characteristics of weight sharing. There are multiple feature maps in each convolution layer, and weight sharing is achieved through convolution kernel between the feature maps [27]. The direct benefit of sharing weights is to reduce the connectivity between layers of the network, while reducing the risk of over-fitting. Secondly, the subsampling layer is a very important layer in convolution neural networks. Subsampling, also known as pooling, usually includes mean pooling and max pooling. Subsampling can be regarded as a special convolution process [28]. Convolution and subsampling greatly simplify the complexity of the model and reduce the parameters of the model.

In general, the advantages of CNN are as follows: First of all, the convolutional network has the characteristics of local perception [29]. In simple terms, the size of the convolution kernel is generally smaller than the size of the input image, so the features extracted by the convolution will pay more attention to locality [30]. In fact, each neuron does not need to perceive the global image, but only needs to perceive the local image, and then the local information is integrated to get the global information. Second, the convolutional network has the characteristic of parameter sharing, which can greatly reduce the amount of computation. Third, the convolutional network uses multiple convolution kernels for convolution operations. Generally, people will not only use a single convolution kernel to filter the input image, because the parameters of a kernel are fixed, the extracted features will also be simple. It is like that we look at things, people must analyze things from multiple perspectives, so as to avoid prejudice to the thing as much as possible. Therefore, people also need multiple convolutional kernels to convolve the input image. Finally, pooling layer in convolutional network is very important. Adding a pooling layer behind the convolutional layer can well aggregate the features and reduce the dimension to reduce the amount of computation.

2.2. ResNet

Deep convolutional neural network has achieved breakthrough results in image classification [31]. Recent evidence shows that the depth of the network is crucial [32]. The team of the Shang Tang Company used the 1207-layer deep neural network to get the best performance in the 2016 ImageNet image classification. Most special visual recognition tasks also benefit greatly from the depth model.

However, blindly increasing the depth of the network may lead to gradient disappearance or gradient explosion, which will degrade the network performance. The main reason is the gradient dispersion caused by Back Propagation [33]. As a result, gradients cannot be delivered to all network layers. Therefore, the performance of the network will be degraded. In order to solve the problems raised above, He Kaiming and his team proposed Residual Network [34], through the residual network structure, the layer can be very deep, and the accuracy is also very excellent. The authors point out that if a saturated accuracy has been learned (or when the error in the next layer is found to be larger), then the next learning goal is to turn to identical mapping learning, that is, the input x is approximated to the output H(x) to keeping the layers behind will not cause a drop in accuracy. To put it plainly, the input x is reintroduced into the output result, so that the weight of the stacking layer tends to zero, learning will be simple, and it will be more convenient to approach identity mapping. The classical method is to map x to $F(x) + x$ through the network, then the network's mapping $F(x)$ naturally tends to 0.

$$H(x) = \sigma(F(x)) + x$$

where $H(x)$ represents the output of the residual unit, x represents the input of residual unit, $F(x)$ is residual function, σ represents ReLU activation function. It learns parameters through intermediate function $F(x)$. The formula $F(x) + x$ can be implemented by a fast connection feed forward neural network. ResNet is equivalent to changing the learning goal, instead of learning a complete output, the training goal is to approximate the residual result $F(x)$ ($F(x) = H(x) - x$) to zero. Therefore, as the network layer increases, the gradient will not disappear.

Fig. 1 shows the structure of the residual unit, which stacked by conv-BN-ReLU-conv-BN layer. Fl−1 represents the output of the upper layer, and F represents the output of this layer.

This structure breaks the convention that the output of n − 1 layer of traditional neural network can only be input to n layer, so that the output of a certain layer can directly skip several layers as the input of the latter layer [35]. Its significance lies in providing a new direction for the problem of stacking multi-layer networks leading to an increase in error rate. Therefore, the number of the neural network layers can exceed the previous constraints, reaching dozens of layers, hundreds of layers or even thousands of layers, which provides feasibility for high-level semantic feature extraction and classification.

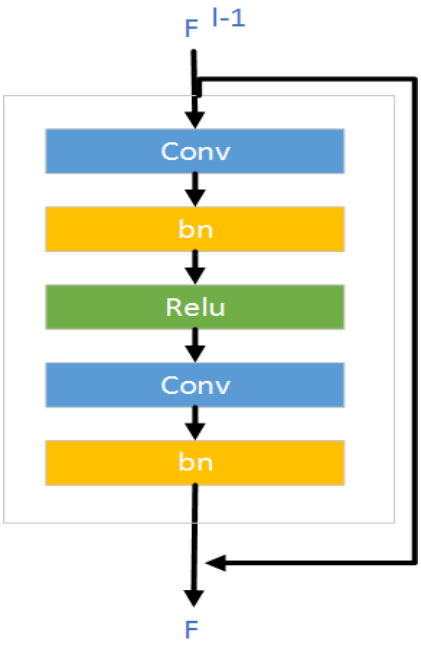


Fig. 1. Residual unit.

3. Method

3.1. Data Acquisition and Preprocessing

In our experiment, we construct the training data set based on Cifar10. It has a total of 60,000 color images. These images are 32 × 32, divided into 10 categories, 6,000 images per category. For each category, randomly select 5,000 pictures as a training set and 1,000 pictures as the test set. We performed 2 types of classification tasks and made the corresponding different training data sets. The experiment data set is constructed as follows:

Experiment 1: 7 groups (g) experiment are performed by increasing of categories ($k \in \{3, \ldots, 9\}$) and we have done 5 (t) times experiment in each group.

1. For g=1, k=3, we take 5,000 pictures of each category as the training set, the training data set size is 5,000×3=15,000.
2. For g=2, k=4, we take 5,000 pictures of each category as the training set, the training data set size is 5,000×4=20,000
3. …
4. Repeat the above steps until g=7, k=9.

Formally, for the k-th ($k \in \{3, \ldots, 9\}$), the training data size is denoted as $set1_k^{g,t}$, then

$$set1_k^{g,t} = 5,000 \times k$$

where g ($g \in \{1, \ldots, 7\}$) group and t ($t \in \{1, \ldots, 5\}$) time. For example, if k = 5, g = 3, $t \in \{1, \ldots, 5\}$, the training set is:

$$set1_5^{3,t} = 5,000 \times 5 = 25,000$$

Experiment 2: For the k-th ($k \in \{3, \ldots, 9\}$) classification experiment, 5 groups (g) experiment are performed by increasing the size of the training data set and we have done 5 (t) times experiment in each group.

1. For k=3, g=1, we take 1,000 pictures of each category as the training set, the training data set size is 1,000×3=3,000.
2. For k=3, g=2, we take 2,000 pictures of each category as the training set, the training data set size is 2,000×3=6,000.
3. …
4. Repeat the above steps until k=3, g=5.

Formally the k-th ($k \in \{3, \ldots, 9\}$) and g ($g \in \{1, \ldots, 5\}$) group, the training data size is denoted as $set2_k^{g,t}$, then

$$set2_k^{g,t} = 1,000 \times k \times g$$

where $t \in \{1, \ldots, 5\}$. For example, if k = 3, g = 2, $t \in \{1, \ldots, 5\}$, the training set is:

$$set2_3^{2,t} = 1,000 \times 3 \times 2 = 6,000$$

3.2. ResNet-32 Network

In order to analyze the impact of data set size and the number of categories on the performance, a 32 layers convolution neural network based on residual unit is constructed. The structure of our network model is illustrated in Figure 2. The network is composed of residual units, the activation function uses ReLU, and the complement zero is set to SAME. Each residual block is stacked by conv-BN-ReLU-conv-BN layer followed by Pool layer and FC layer.

Network structure: We know that when the depth of the neural network is continuously increased, the accuracy of the model will rise first and reach saturation, and then continue to increase the depth will lead to a decrease in accuracy. Therefore, we should choose the appropriate depth when constructing the network. In order to determine how many layers of the network should be constructed we have done some experiments. We construct different depths of network structure (including 20 layers, 32 layers, 44 layers, 56 layers, and 110 layers), and train them separately to obtain different prediction accuracy (Table 1). From Table 1, we can see that when the network is 20 layers, the prediction accuracy is 91.65%. When the network layer is 32 layers, the prediction accuracy is 92.41%. Until the network layer is 110 layers, the

prediction accuracy is 93.35%. From the above results, we can know that when the network layer is 110 layers, the prediction performance is best. However, the deeper the network, the higher the hardware configuration required. Moreover, the deeper the network, the longer time it takes to wait for training results. Furthermore, in this paper, the main research is the impact of the training set size and the number of categories on performance, and do not need a very deep a network. Based on the above reasons, we finally construct a 32-layer network to complete this experiment.

Learning rate strategy: Inspired by [36] and [37], we set the initial learning rate (lr = 0.1), and the learning rate is reduced by 10 times per iteration. Because the initial learning rate is set too low, the network will converge slowly since you have made very few adjustments to the weight of the network. While if your learning rate is set too high, the network will not converge. Therefore, we set the initial learning rate to 0.1 and adopt the "Step Decay" strategy, so that the learning rate is reduced by 10 times per iteration.

Experiment configurations: To ensure the consistency of the experiment, we use accuracy as the quantization metric, 1000 epochs are trained for each experiment (batch_size = 100). All experiments are implemented in python2.7 by using Caffe framework. We train the networks on a NVIDIA Tesla M40 GPU and the model that performs the best on test data set are saved for further analysis.

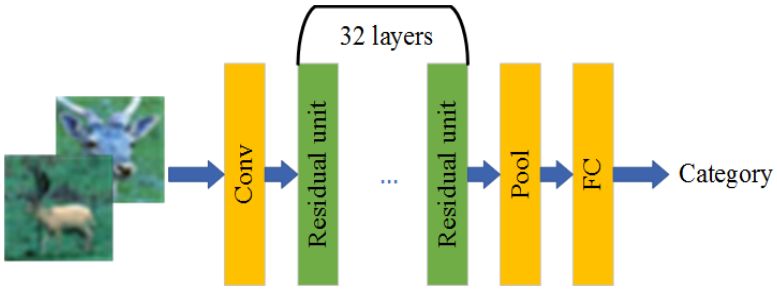


Fig. 2. 32-layer residual network.

This experiment uses the cifar10 dataset, and 50000 images are used as a training set and 10000 images are used as a test set

Table 1. The Accuracy of Different Layers of the Network

Model	Accuracy
ResNet-20	91.65%
ResNet-32	92.41%
ResNet-44	92.52%
ResNet-56	93.16%
ResNet-110	93.35%

3.3. Training Classification Network

We completed 2 types of experiments. The effects of the number of categories on performance and the impact of data set size on classification performance are studied separately.

Experiment 1: In experiment, 7 groups (g) experiment are performed in total with k (k $\in$ {3, … , 9}). For each group, the training data set size is 5,000*k. The details are as follows:

1. For g=1, k=3, the training set $set1_3^{1,t}$=15,000. 5 times ($t \in \{1, \dots, 5\}$) experiment are performed and carefully record each result and calculate the average performance in this group.
2. For g=2, k=4, the training set $set1_4^{2,t}$=20,000. 5 times ($t \in \{1, \dots, 5\}$) experiment are performed and carefully record each result and calculate the average performance in this group.

3. …

4. Repeat the above steps until g=7, k=9.

Experimental 2: In experiment, 5 groups (g) experiment are performed with a fixed k ($k \in \{3, \ldots, 9\}$) and 5 times experiment in each group, 35 groups experiment are done in total. For each group, the training data set size grows by a gradient of $1{,}000 \times k \times (g\text{-}1)$. The details are as follows:

Due to $k \times 5{,}000$ as training data size, $k \times 1{,}000$ as test data size, so when k=3, there are 18,000 images. 15,000 images can be used as training sets and 3,000 images can be used as test sets. For each 3-classification:

1. For g=1, the training set $set2_3^{1,t}$=3,000. 5 times ($t \in \{1, \ldots, 5\}$) experiment are performed and carefully record each test result and calculate the average performance in this group.
2. For g=2, the training set $set2_3^{2,t}$=6,000. 5 times ($t \in \{1, \ldots, 5\}$) experiment are performed and carefully record each test result and calculate the average performance in this group.

3. …

4. Repeat the above steps until g=5.

Table 2. The Accuracy of Each Classification Task with different *K*-classifier. All Experiments were Performed 5 Times with Fixed k

k-classifier	g	t=1th	t=2th	t=3th	t=4th	t=5th
3	1	0.9610	0.9386	0.9700	0.9746	0.9703
4	2	0.9610	0.9392	0.9252	0.9217	0.9620
5	3	0.9296	0.9376	0.9234	0.9574	0.9622
6	4	0.9575	0.9251	0.9571	0.9606	0.9552
7	5	0.9517	0.9292	0.9191	0.9321	0.9196
8	6	0.9355	0.9418	0.9473	0.9367	0.9386
9	7	0.9210	0.9250	0.9290	0.9255	0.9356

4. Results and Discussion

4.1. Experiment 1

For experiment 1, 7 groups experiment were performed with k ($k \in \{3, \ldots, 9\}$) and 5 times experiment for each group. We conducted a total of 35 (7×5) experiments and results are shown in Table 2. For comparison, we take the average precision and maximum precision of each group and plot the curve in Fig. 3, the horizontal axis corresponds to the number of classification categories, and the vertical axis corresponds to the quantization metric (accuracy). Calculate the average value of the 5 times experiment results for each group and plot them accordingly with the red curve, the blue point represents the maximum accuracy of each group.

From Fig. 3 and Table 2, it is clear that all accuracy is above 0.9000. For $k \in \{3, \ldots, 9\}$, $g \in \{1, \ldots, 7\}$ and $t \in \{1, \ldots, 5\}$.

1. When $set1_3^{1,t}$=15,000, the highest accuracy is 0.977 and the average accuracy is 0.9629.
2. When $set1_4^{2,t}$=20,000, the highest accuracy is 0.9675 and the average accuracy is 0.9418.
3. …

4. When $set1_9^{7,t}$=45,000, the highest accuracy is 0.9356 and the average accuracy is 0.9290.

From the above experimental results, we can know that as the number of categories increases, the test accuracy is gradually decreasing. Although there are some fluctuations, it is generally declining. The number

of categories is inversely proportional to the performance of the classification.

Furthermore, by analyzing the results of k = 5 and k = 7 in Table 2, we can see that when the number of categories is large, sometimes we can also achieve better results. For $k(k \in \{3, \ldots, 9\}), g \in \{1, \ldots, 7\}$ and $t \in \{1, \ldots, 5\}$.

1. When k = 5, g = 3 and t = 4, training set size $set1_5^{3,4}$ =25,000, the accuracy is 0.9574.

2. When k = 7, g = 5 and t = 1, training set size $set1_7^{5,1}$=35,000, the accuracy is 0.9517.

From the above results, we can see that the accuracy of k=7 can sometimes reach the accuracy of k=5. Therefore, when a large number of categories, sometimes we can get better results.

4.2. Experiment 2

For experiment 2, fix k ($k \in \{3, \ldots, 9\}$) and perform 5 groups experiment for each classification(35 groups experiment in total), Fig. 4 to 10 show the corresponding accuracy, the horizontal axis corresponds to the size of the training set, and the vertical axis corresponds to the test accuracy. Calculate the average of each group's accuracy and connect it with red curve.

From the trend of the curve, we can know that the accuracy is constantly increasing as the training set increases. For example, from Fig. 4, we can see that for k = 3, and $g, t \in \{1, \ldots, 5\}$.

1. When $set2_3^{1,t}$=3,000, the average precision is 0.7351.

2. When $set2_3^{2,t}$=6,000, the average precision is 0.9715.

3. When $set2_3^{3,t}$=9,000, the average accuracy is 0.9746.

4. When $set2_3^{4,t}$=12,000, the average accuracy is 0.9597.

5. When $set2_3^{5,t}$=15,000, the average accuracy is 0.9711.

From the above data, we can know that when the training set size=3,000, the classification performance is poor. However, when the training set size=6,000, the classification performance has been greatly improved, and with the increase of the training set, the classification performance is constantly increasing.

Similarly, by analyzing Fig. 5, 6, 7, 8, 9, 10 we can see that, for average accuracy, the larger the training set, the higher the accuracy. As the training set increases, the performance of the classification is gradually increasing. The performance of the classification is proportional to the size of the training set.

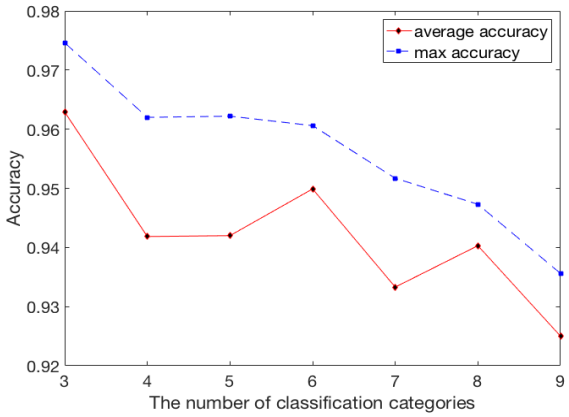


Fig. 3. The average accuracy with $k(k \in \{3, \ldots, 9\})$ for experiment 1. The horizontal axis corresponds to the number of classification categories, and the vertical axis corresponds to the accuracy. Red solid lines are joined by the average accuracy of each group, and blue dotted lines are joined by the maximum accuracy of each group.

Furthermore, analyzing 5 groups experiment results for each k-classification, we can know that when the training samples are insufficient, while sometimes better results can be obtained. For example, when k = 5, the better accuracy obtained by $set2_5^{1,5}$=5,000 is approach to the accuracy obtained by 5 $set2_5^{2,3}$=10,000. When k = 6, the better accuracy obtained by $set2_6^{1,2}$=6,000 is approach to the accuracy obtained by $set2_6^{2,1}$=12,000. In each group experiment results, when k = 3, 4, 5, the small difference exists in accuracy, but when k = 6, 7, 8, 9, the big difference exists in accuracy, so we can know that for the same network, the more categories that are classified, the more obvious the performance difference.

Similarly, From Fig. 5, we can see that for k = 4, and $g, t \in \{1, \ldots, 5\}$.

1. When $set2_4^{1,t}$=4,000, the average precision is 0.7866.

2. When $set2_4^{2,t}$=8,000, the average precision is 0.9334.

The average accuracy of the 8,000 training set is much higher than the average accuracy of the 4,000 training set. However, the best result of a training set with 4,000 is approach to the poor result of a training set with 8,000.

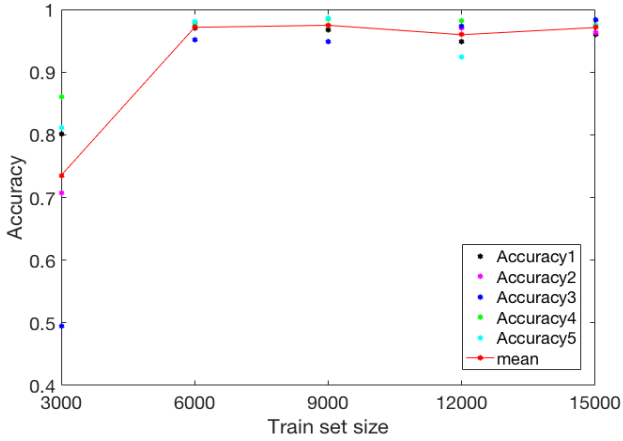


Fig. 4. Set k = 3, and g, t $\in$ {1, 2, … , 5}. Each square point represents the t-th experiment results in g-th group, and the red circle points represent the average of the results of each group. We connect the average of each group result with a polyline to obtain a trend graph of experimental results.

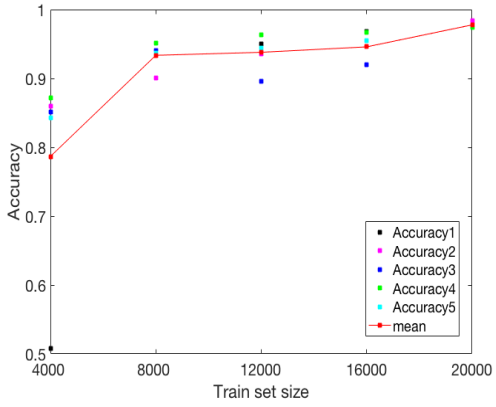


Fig. 5. Set k = 4, and g, $t \in \{1, 2, \ldots, 5\}$. Each square point represents the t-th experiment results in g-th group, and the red circle points represent the average of the results of each group. We connect the average of each group result with a polyline to obtain a trend graph of experimental results.

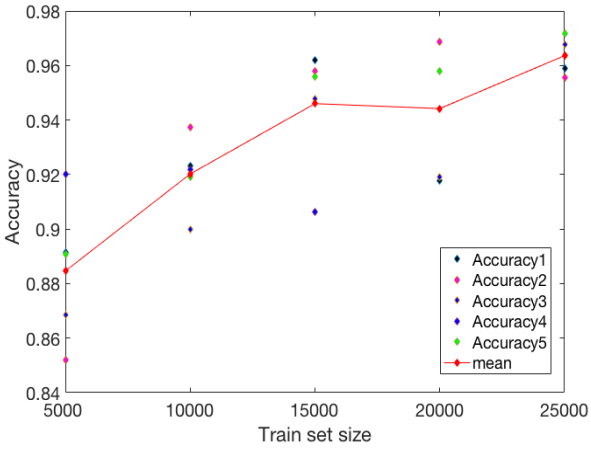


Fig. 6. Set k = 5, and $g, t \in \{1, 2, \ldots, 5\}$. Each square point represents the t-th experiment results in g-th group, and the red circle points represent the average of the results of each group. We connect the average of each group result with a polyline to obtain a trend graph of experimental results.

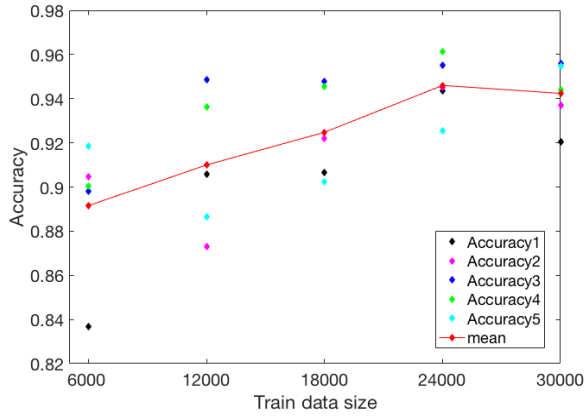


Fig. 7. Set k = 6, and $g, t \in \{1, 2, \ldots, 5\}$. Each square point represents the t-th experiment results in g-th group, and the red circle points represent the average of the results of each group. We connect the average of each group result with a polyline to obtain a trend graph of experimental results.

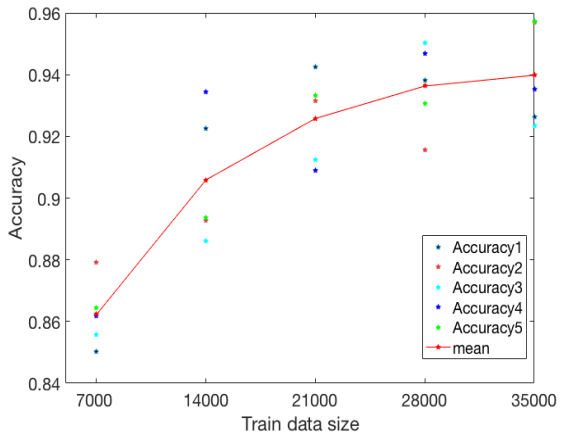


Fig. 8. Set k = 7, and $g, t \in \{1, 2, \ldots, 5\}$. Each square point represents the t-th experiment results in g-th group, and the red circle points represent the average of the results of each group. We connect the average of each group result with a polyline to obtain a trend graph of experimental results.

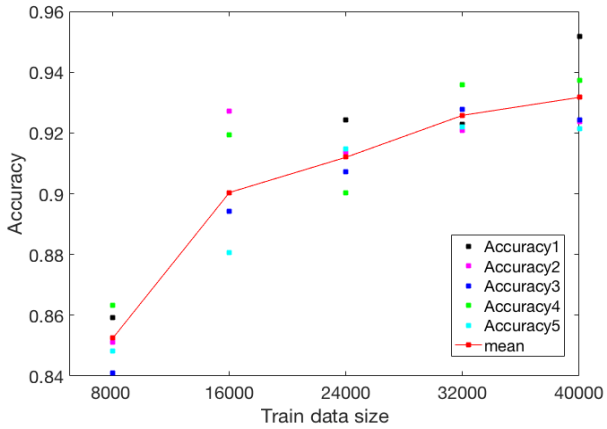


Fig. 9. Set k = 8, and $g, t \in \{1, 2, \ldots, 5\}$. Each square point represents the t-th experiment results in g-th group, and the red circle points represent the average of the results of each group. We connect the average of each group result with a polyline to obtain a trend graph of experimental results.

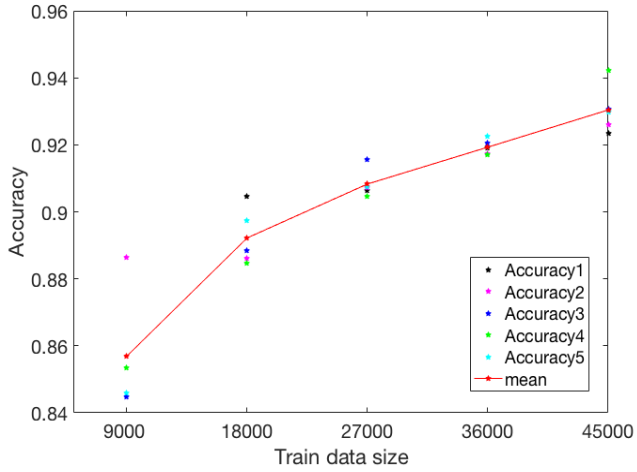


Fig. 10. Set k = 9, and $g, t \in \{1, 2, \ldots, 5\}$. Each square point represents the t-th experiment results in g-th group, and the red circle points represent the average of the results of each group. We connect the average of each group result with a polyline to obtain a trend graph of experimental results.

5. Conclusion

In this paper, we study the effect of data set size and the number of categories on performance in multi-classification experiments. For experiment 1, 7 groups experiment are performed with different category numbers and 5 times are performed in each group (35 times experiment in total). For experiment 2, we do 5 groups experiment by increasing the size of the training set for each category, and the data set of each group was increased by a gradient of $1000 \times k \times (g\text{-}1)$. There are 35 groups experiment performed with 5 times experiment in each group (175 times experiment in total). The results of experiment 1 show that, for the average accuracy, the more categories, the worse the performance, and the less categories, the better the performance. Therefore, in future experiments, we can reduce the number of categories in order to improve accuracy. In addition, when the number of categories to be classified is large, sometimes higher accuracy can be obtained. So, for classification experiments with a large number of categories, we can train more times to get the most accurate results. The results of experiment 2 show that, for the average accuracy of each group, the larger the training set, the better the classification performance. However, when the

number of training data is insufficient, sometimes better classification results can be obtained. Therefore, when you do classification experiments, if you can't get a large data set, you can do more experiments and choose the best results. Furthermore, in each group experiment results, the more categories that are classified, the more obvious the performance difference. In the future, the results of this paper not only can guide us to do experiments on image classification, but also have important guiding significance for other experiments.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61602066) and by the Project Supported by the Scientific Research Foundation of the Education Department of Sichuan Province(17ZA0063 and 2017JQ0030) and the Scientific Research Foundation (KYTZ201608) of CUIT, and partially supported by Sichuan International Science and Technology Cooperation and Exchange Research Program (2016HH0018), and Sichuan Science and Technology Program (2018GZ0184), and the outstanding young scientists and technicians of the Education Department of Sichuan Province (No.2019JDJQ0002).

References

[1] Liu, A. J., Yu, H., Yang, W., & Sun, C. (2015). Combining active learning and semi-supervised learning based on extreme learning machine for multi-class image classification.

[2] Qassim, H., Verma, A., & Feinzimer, D. (2018). Compressed residual-VGG16 CNN model for big data places image recognition. *IEEE Computing and Communication Workshop and Conference*, 169–175.

[3] Shang, L., Yang, Q., Wang, J., Li, S., & Lei, W. (2018). Detection of rail surface defects based on CNN image recognition and classification. *Proceedings of the International Conference on Advanced Communication Technology*.

[4] Boehm, J. (2000). Curvature-based range image classification for object recognition. *Proceedings of SPIE - The International Society for Optical Engineering 2000*.

[5] Liu, C., & Wechsler, H. (2002). Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing a Publication of the IEEE Signal Processing Society*.

[6] Haraclick, R. M. (1973). Texture features for image classification. *IEEE Trans Smc*.

[7] Zamolotskikh, A., & Cunningham, P. (2007). An assessment of alternative strategies for constructing emd-based kernel functions for use in an SVM for image classification. *International Workshop on Content-Based Multimedia Indexing*.

[8] Lin, Y., Lv, F., Zhu, S., & Yang, M. (2011). Large-scale image classification: Fast feature extraction and SVM training. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[9] Foody, G. M., & Mathur, A. (2004). Toward intelligent training of supervised image classifications: Directing training data acquisition for SVM classification. *Remote Sensing of Environment*.

[10] Durand, T., Thome, N., & Cord, M. (2016). MANTRA: Minimum maximum latent structural SVM for image classification and ranking. *Proceedings of the IEEE International Conference on Computer Vision*.

[11] Amato, G., & Falchi, F. (2010). KNN based image classification relying on local feature similarity. *Proceedings of the International Conference on Similarity Search and Applications* (pp. 101–108).

[12] Chaib, S., Yao, H., Gu, Y., & Amrani, M. (2017). Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models. *Proceedings of the International Conference on Digital Image Processing*.

[13] Mccoppin, R., & Rizki, M. (2014). Deep learning for image classification. *SPIE Defense + Security*, 90790T.

[14] Li, H., Tao, C., Wu, Z., Chen, J., Gong, J., & Deng, M. (2017). RSI-CB: A large scale remote sensing image classification benchmark via crowdsource data 2017.

[15] Wang, L., Guo, S., Huang, W., & Qiao, Y. (2015). Places205-VGGNet models for scene recognition. computer science.

[16] Singla, A.; Yuan, L., & Ebrahimi, T. (2016). Food/non-food image classification and food categorization using pre-trained GoogLeNet model. *International Workshop on Multimedia Assisted Dietary Management*, 3–11.

[17] Mahmood, A., Bennamoun, M., An, S., & Sohel, F. (2016). ResFeats: Residual network based features for image classification.

[18] Gupta, H., Jin, K. H., Nguyen, H. Q., Mccann, M. T., & Unser, M. (2017). CNN-based projected gradient descent for consistent CT image reconstruction. *IEEE Transactions on Medical Imaging*.

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition.

[20] Bae, W., Yoo, J., & Ye, J. C. (2017). Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. *Computer Vision and Pattern Recognition Workshops*.

[21] Krizhevsky, A. (2010). Convolutional deep belief networks on CIFAR-10 2010.

[22] Carvalho, E. F., & Engel, P. M. (2013). Convolutional sparse feature descriptor for object recognition. *Proceedings of the Brazilian Conference on Intelligent Systems*.

[23] Qin, H., Yan, J., Li, X., & Hu, X. (2016). Joint training of cascaded CNN for face detection. *Computer Vision and Pattern Recognition*, 3456–3465.

[24] Hou, R., Chen, C., & Shah, M. (2017). Tube convolutional neural network (T-CNN) for action detection in videos. *IEEE International Conference on Computer Vision*, 5823–5832.

[25] Chen, Y. N., Han, C. C., Wang, C. T., Jeng, B. S., & Fan, K. C. (2006). The application of a convolution neural network on face and license plate detection.

[26] Lo, S. C. B., Chan, H. P., Lin, J. S., Li, H., Freedman, M. T., & Mun, S. K. (1995). Artificial convolution neural network for medical image pattern recognition. *Neural Networks*, 1201–1214.

[27] Boulch, A. (2018). ShaResNet: Reducing residual network parameter number by sharing weights 2018.

[28] Graham, B. (2014). *Fractional Max-Pooling*.

[29] Kulkarni, P., Zepeda, J., Jurie, F., Perez, P., & Chevallier, L. (2015). Hybrid multi-layer deep CNN/aggregator feature for image classification.

[30] Roychowdhury, S., & Ren, J. (2017). Non-deep CNN for multi-modal image classification and feature learning: An Azure-based model. *IEEE International Conference on Big Data*, 2893–2812.

[31] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., & Bernstein, M. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

[32] Huang, J., Qian, F., Guo, Y., Zhou, Y., Xu, Q., Mao, Z. M., Sen, S., & Spatscheck, O. (2013). An in-depth study of LTE: Effect of network protocol and application behavior on performance. *Computer Communication Review*, *43*, 363–374.

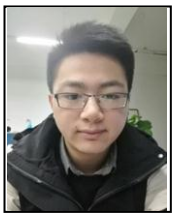
[33] Chen, C. H., & Lai, H. (2002). An empirical study of the gradient descent and the conjugate gradient backpropagation neural networks. *Oceans*, 132–135.

[34] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. *European Conference on Computer Vision*, 630–645.

[35] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.

[36] Wu, X., Ward, R., & Bottou, L. (2018). WNGrad: Learn the learning rate in gradient descent 2018.

[37] Gu, J., Wang, Y., & Heffernan, N. T. (2014). Personalizing knowledge tracing: Should we individualizeslip.



Chao Luo is a graduate student. He is currently studying at the School of Computer Science, Chengdu University of Information Engineering. His primary research directions are machine learning and deep learning.

Xiaojie Li received the Ph.D degree in computer science and engineering from the School of Computer Science, Sichuan University, Chengdu, China, in 2015. She is an associate professor at Chengdu University of Information Technology, Chengdu, China.

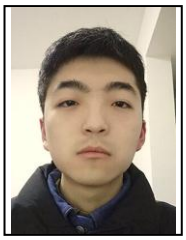
Her current research interests are in machine learning, medical image segmentation and deep learning.

Jing Yin received the her B.S. degree in information and computing science from Chongqing Three Gorges College, Wanzhou, China, in 2002 and the M.S. degree in computer software and theory from Southwest Jiaotong University, Chengdu ,China, in 2005. She is currently pursuing the Ph.D. degree in computer scinece and technology at Sichuan University, China. Her research interest includes machine learning, deep learning, and neural network

From 2005 to now, she is a teacher in college of computer science and engineering at Chongqing University of Technology, China.

Jia He received her BEng and Msc degrees in computer science and technology from Southwest Normal University of China, in 1989 and 1996, respectively, and received PhD degree in computer science from University of Electronic Science and Technology of China, in 2012. She is currently a professor of computer science at Chengdu University of Information Technology, China.

Her current research interests include are computer vision, artificial intelligence, and pattern recognition.



Denggao Li is pursuing the bachelor's degree in computer science and technology at Chengdu University of Information Technology, Chengdu, China.

His current research interests are in machine learning, medical image segmentation and deep learning.

Jiliu Zhou received his BSc degree in electronic and computer science from Sichuan University in 1985, the MSc degree in electronic and computer science from Tsinghua University in 1988, the PhD degree from Sichuan University in 1999. He was promoted full professor in 1999 at Sichuan University, and now associated with Sichuan University and Chengdu University of Information Technology (CUIT) as full professor.

He has published more than 200 journal papers, and now is the leader of Collaborative Innovation Center for Graph and Image Intelligent System.