**Review of Classification methods**

Deng Guoquan. Student number:19403844

Dong Shengbo. Student number:19441746

(Contribution percentage Deng Guoquan:55%, Dong Shengbo:45%)

December 2019

**Abstract**

*In this paper, we study the performance of different classifiers on the CIFAR-10 dataset. There are many different classification techniques that can solve the image classification problems. We choose some simple classifiers like* *Support Vector Machines (SVM),* *random forest, and some complex neural network like Inception network, Residual network to classify the dataset. Then we make a comparison and discuss the pros and cons of these models.*

***Keywords****: Image Classification, CIFAR-10, SVM, CNN, Random forest, Inception network, Residual network*

1. **Introduction**
   1. **Background**

Nowadays, AI-powered camera is one of the biggest selling points for the smartphones. An AI camera can do automatic scene recognition and then it can automatically adjust the settings like exposure, saturation, dynamic range and contrast. Then people can easily take a good photo like a professional photographer. So how it works? Smartphone manufacturers usually use deep learning to develop image recognition engines which use a model trained on millions of images. So it is necessary to find a classification technique that has great performance on image classification problems.

* 1. **Related Works**

Image classification is a fundamental and important topic in deep learning. When it comes to image classification problems, convolution network has been the mainstream of the state-of-art computer vision solution for a large variety of tasks. Since the 2012 ImageNet competition, the winning model called “AlexNet” appeared and quickly implied into a lot of other computer vision tasks, the past 10 years, researchers have been intending to develop larger and deeper neural networks to gain the better result and score although it also cause high computation cost and endless data need. In this paper, we will apply some of the most popular models include VGG Block, Inception Block and Residual Block in the image classification task and evaluate the efficiency of different network structure. Because the neural network such as VGG, GoogleNet and Resnet are usually very large and it will take a very long time to train, we will imply the mini structure of theses networks base on their initial ideas and the correspond data set. And we will try to gain the higher score with less model size as much as possible. To make a comparison, we also use some classic machine learning models like SVM, Random Forest to do the experiments.

* 1. **Dataset**

As a beginner, ImageNet is too large and even can not been used in personal computer. So we choose the mini data set CIFAR-10 as experimental data set. The CIFAR-10 small photo classification problem is a standard data set used in CV and deep learning. Although the data set has been effectively solved but it is not easy to reach the state-of-art 90% accuracy with a small model, and recently CIFAR-10 is used as the benchmark of DAWNBench competition which consider computation time and cost as the critical resources in building the network.

CIFAR-10 data set is comprised of 60000 32x32x3 pixel color photographs from 10 classes, such as airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. The whole data set is split into train set contain 50000 photos and test set consist of 10000 photos. The data set can be download from Google Tensorflow or Keras service.

### Methodology

#### 2.1 Network Selection and architecture details

Some Milestones of classical network structure are firstly considered. For example VGG 16 proposed by K.Simonyan and A.Zisserman, developed from AlexNet and with larger model size and achieve 92.7% top-5 accuracy in ImageNet competition. Residual network proposed in order to solve the questions on how to build deeper network. Due to the gradient is hard to transfer to earlier layers, the earlier layers become more like a random mapping layer and affect the optimized direction of the later layers which cause degradation in the deeper network. So residual block provides a high way for the gradient transform from end layers to root layers. And I will also test the inception network proposed from Google Net, inception block is a mix of different size of convolution kernel layers, so that it can catch more information from the image and it also reduce sparsity and increase correlation in an output layer base on the idea of Hebbian principle.

The network structures are shown below figure and each structure consists of a mini version and a deeper version:

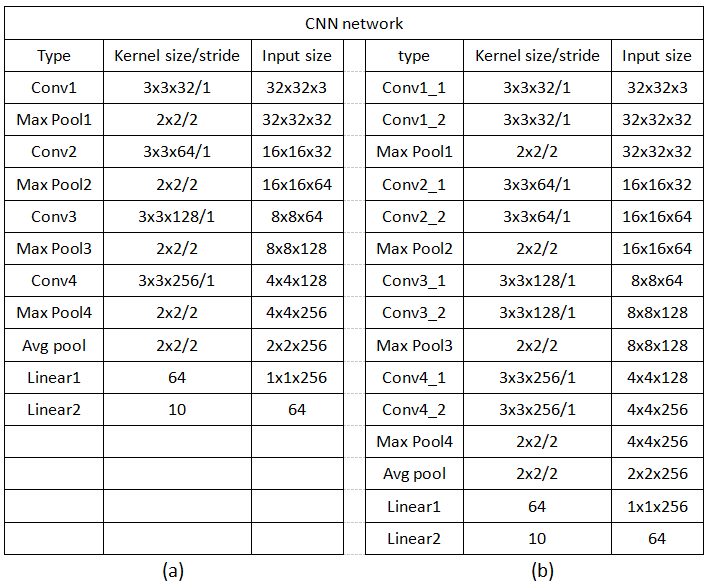


Figure 1: (a)mini version CNN network (b)deeper version CNN network

##### 2.1.1 Design of CNN Network

Figure 1 describe the structure of CNN network and I also summary design rules:

(a)the channel size of the later layer should be larger than earlier layer, because pool can cause information loss though it provide good generalization and can been seen as abstract layer. If larger channel size in the earlier layers will also cause information bottleneck and waste computation resources.

(b)Average pool in the last third layer can produce a better result than using full connect layer after a flatten layer. It also reduce the model size and speed training but the reason why it works still unclear and it will be discussed in the future.

(c)The idea of stacking two convolution layers comes from VGG block, it provides more sight of the network and improve nonlinear.

(d)Although VGG has achieve great success in a lot of tasks but I try to less use full connect layer.

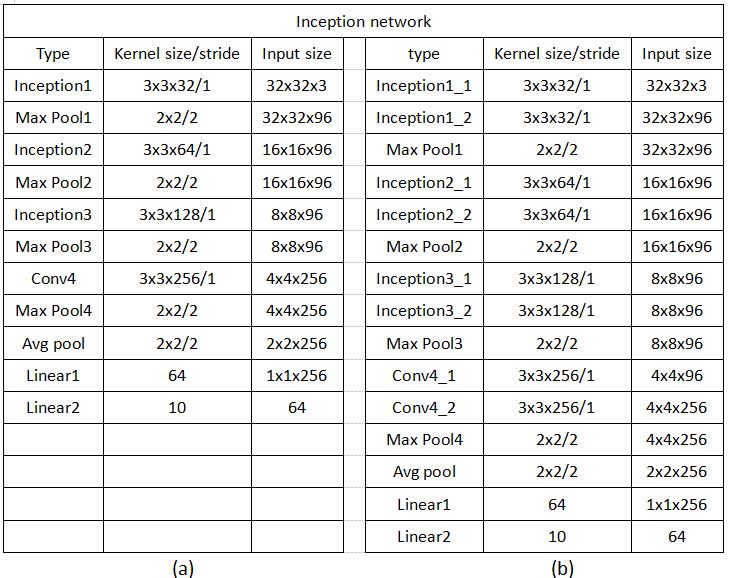


Figure 2:(a)mini version Inception network (b)deeper version Inception network

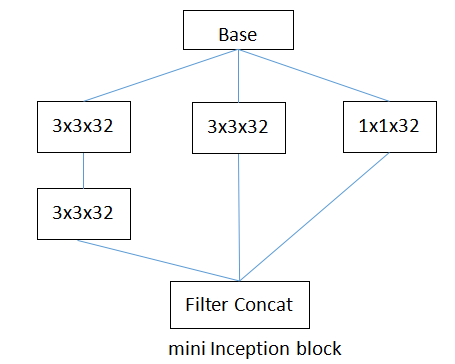


Figure 3:Inception block

##### 2.1.2 Design of Inception Network

Figure 2 describe the framework of Inception network and Figure 3 show the structure of inception block.

(a)The core of inception block is to increase the width of the sight of the network. Basing on the idea of Inception v2&v3 paper, replacing 5x5 layer into 2x3x3 in order to reduce model size and save computation resources. Different size of filter contact can be imaged as several mini networks working in the same time and con-cat the output of these mini networks can create a tight information with less sparsity. A similar idea is multi-head attention mechanism.

(b)The sight of inception block should be suitable correspond to the image size. If the image size is too small, the zero padding will destroy distribution of the image original distribution because more zero padding lead to difference between two image gradually disappear. In this experiment, inception should be utilized in the image size larger than 4x4.

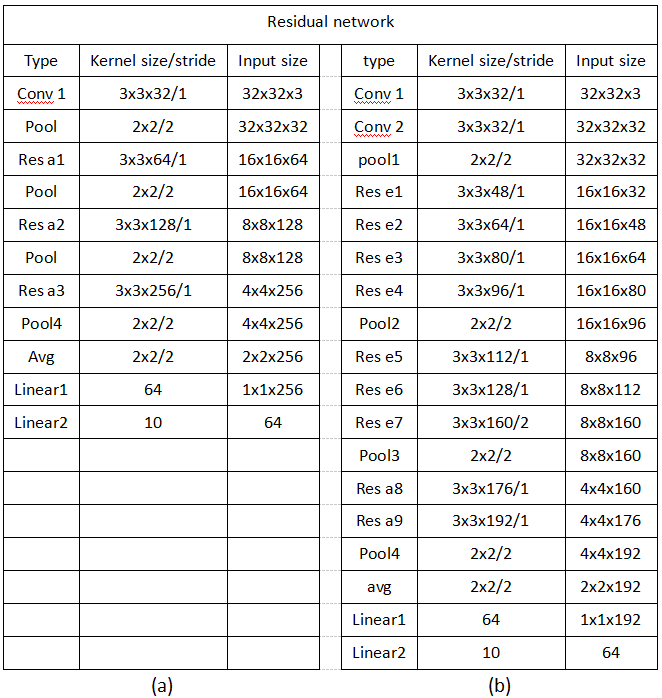


Figure 4:(a)mini version Residual network (b)deeper version Residual network

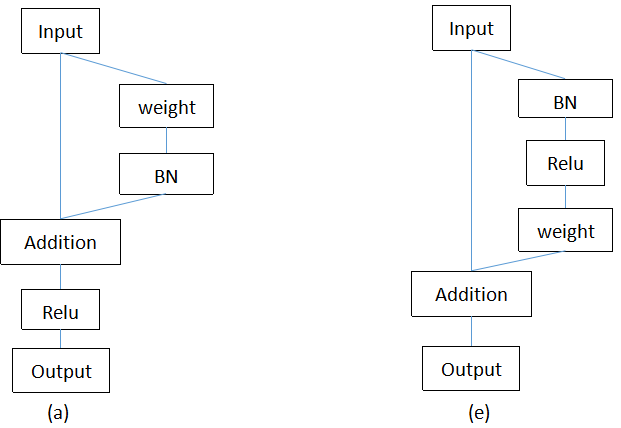


Figure 5:(a)Residual block a (e)Residual block e

##### 2.1.3 Design of Residual Network

The residual network is shown in figure 4. As shown in figure 5, I use two types of residual block. The residual block a is widely used in common condition and the residual block e publishing by the later paper which can gain higher accuracy.

(a)The core of residual block is to create the high way for the gradient transformed to the root layer. Two mini residual blocks are test in this experiment, the residual block a and residual block e. Although the original paper show that implying the batch normalization in the high way can destroy the distribution of the original high way and result in lower accuracy. But I still imply the batch normalization in the short cut when two near layers having different channel size because without batch normalization, the variance can increase to a unacceptable level. This question will be discuss in the future.

(b)I suppose if the distribution of original input is so larger or smaller than the batch normalization layer’s distribution in the residual path, the relu(nonlinear layer) will become meaning less, so I believe the z-score per-processing is strongly useful.

(c)To some extent, residual block e combine the advantages of (a)without batch normalization layer in the short cut (b) promise the nonlinear layer relu provide nonlinear effect after a batch normalization process.

#### 2.1.4 evaluation method and loss function

All the models will be created in similar model size, in a other word, the similar number of parameters so that models can be compared in a approximate condition. After training, the loss of training data set, the loss and accuracy of testing data set will be compared and analysis. The loss function is chose as cross entropy cost function after softmax process, the formula is 

#### 2.1.5 Activation Function

Activation function chose relu and there is no activation function in the end layer. Relu can be used to guarantee the gradient can be transfer to the earlier layers. It is widely used in most of deep learning tasks.

#### 2.2 Hyper Parameter and Setting

##### 2.2.1 Initialization method

Initialization method play an important role in network training process and a appropriate weight initialization means faster converge and less training time. In this report, all the weight initial methods are Xavier. The Xavier initialization solve the problem of the continued increasing variance scale in later layers. Imaging the network formula is . so the variance of the k+1 layer become . since it is linear mapping. Xavier force  so guarantee the variance between layers can be the same. Without the variance shift in the layers, the gradient descent can be utilize in a same scale between different layers so network can learn faster.

##### 2.2.2 regularization method

There are two regularization methods used in the experiment, The first is L2 weight regularization, which is normal used in deep learning, and .Label smoothing is another useful mechanism. Different from pure cross entropy cost function with one hot label, which we called hard classify target, label smoothing is a soft classify target function intend to maximize the correct class and persist the similar distance between the correct class and all other classes. It avoid two problems: (1)if the model learns to assign full probability to correct class it will loss generalize. (2) it encourage distance between correct class and all other classes even two classes may be similar etc Toy poodle and Mini poodle in the ImageNet, to some extent, it cause information loss. After utilize label smoothing, the points in the same class will become more tight. And the formula of label smoothing is also easily handled,.

#### 2.3 Batch Normalization

Batch normalization is a technique for improving speed, performance and stability of neural network. It is usually used between the output layer and the activation function by normalize the layer and re-scaling the layer into to other distribution. The effect of the batch normalization is avoid the internal covariation shift of the layers, Because it re-scale layers into appropriate distribution, different layers between two batch normalization layers, to some degree, become more independent that is the reason gradient descent algorithm can be more efficient in changing the weight and affect the finial result. And normalize process in batch normalization layer also promise the nonlinear when I use the relu activation by avoiding nearly most neural cells in a layer are positive which destroy the nonlinear characteristic of network and cause useless layers. In this experiment, in training period the normalize process is z-score process so the batch size larger than 32 will be better. In the testing period, using the moving average value of the mean and variance recorded in the training period because there is no mean and variance can be calculated, the moving average decay weight chose as 0.95 instead of 0.999 in the usual work which can be faster.

#### 2.4 Optimization Method

Instead of using normal SGD method, Adam delta optimization is strongly advised. Adam optimization is a combination of RMSProp and Momentum. The benefit of momentum is it can reduce the variance of model in the training period by restricting the change of optimize direction. And the benefit of the AdaGrad and the RMSProp is recording the accumulated gradient of each parameter of different weight layers aim to distribute larger learning rate to the parameter whose accumulated gradient is less and restrict the learning rate of the parameter which has been changed a lot, so that the weight of network can learn in a balance way and avoid extremely sparsity for example some weight are too large while other is near zero without changing.

**2.5 Principal Component Analysis(PCA)**

The Principal Component Analysis (PCA) is an image pre-processing algorithm which can reduce data dimensionalities. PCA can canter data points to mean location, then factorise the covariance matrix into Singular Vector Decomposition components. Then we can use non-zero eigenvectors to represent the dataset spanning over a lower dimensional space. It tries to preserve the essential parts which have more variation of the data and remove the non-essential parts with fewer variation. PCA can help us achieve a good balance between accuracy and time.

**2.6 Support Vector Machine(SVM)**

Support Vector Machine is a classic supervised learning model which often used for classification and regression. Its core idea is to find a hyperplane represents the maximum margin of separation between classes. If the data have many attributes, SVM can work very well. Besides SVM can change error term to avoid over-fit. But the training speed is very low while using non-linear classification in high-dimension space. When using Non-linear classification in high dimension, there will be many support vectors, so the training speed will become very low. In this experiment,we choose linear SVM to train CIFAR-10 and we also use PCA to pre-process the dataset and reduce the training time.

**2.7 Random Forest**

The random forest consists of many decision trees. And there is no correlation between the decision trees. When training, every tree in the forest learns from a random sample of the data. The random forest trains each tree on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree. It is a non-parametric model and it is easy to contain lots of numeric or categorical data layers. Once the rules are developed the classification is very fast. Besides random forests is an ensemble model which uses the results from different models to calculate the final result. And the result is usually better than the result from any one of the individual models.[11] Building a random forest needs 3 steps:

1. Draw n tree bootstrap samples from the original data.
2. For each samples create an un-pruned classification. Then at every node, random select features.
3. Predict new data by aggregating the predictions of the n trees.

### Experimental Study

**3.1 Random Forest**

In this experiment, we use Random Forest which provided by Scikit-learn to classify CIFAR-10. Firstly, we change the tree number in the forest and keep other configuration the same. And we found that when the tree number increases, the accuracy is increasing too，but it will take more time. Then we change the number of n\_components of PCA and keep the parameters of random forest the same. The accuracy is not very different when n\_components equal to 60, 110, 140. But if we don’t use PCA to reduce dimension, The computation time increases a lot and the accuracy becomes a little worse. So if we want to increase the accuracy , we can just increase the number of trees. But the training time will be longer. PCA can help us pre-process the data and reduce the computation time greatly.

TABLE 1: Random Forest Features

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Time(s) |
| 300tree + 100PCA | 0.4907 | 194.7 |
| 500tree + no PCA | 0.4872 | 1197.5 |
| 500tree + 60 PCA | 0.4932 | 274.1 |
| 500tree + 100PCA | 0.4992 | 345.2 |
| 500tree + 140PCA | 0.4946 | 367.4 |
| 1000tree + 100PCA | 0.5015 | 666.3 |
| 1000tree + 140PCA | 0.4959 | 748.4 |

**3.2 Support Vector Machine(SVM)**

In this experiments, SVM model is directly derived from Sklearn’s library. We change the number of n\_components of PCA and keep other configuration the same. We can see from the TABLE 2 if we don’t use PCA to pre-process the dataset, it will take much more time than using PCA. It seems that PCA is very appropriate for SVM classifier select features. Although PCA can reduce training time greatly. But if the n\_components is too large, the compression of data will be very low, and the computation time will increase a lot.

TABLE 2: SVM Features

|  |  |  |
| --- | --- | --- |
| Model | Accuracy | Time(s) |
| SVM + no PCA | 0.5437 | 8634.3 |
| SVM + 60 PCA | 0.529 | 319.1 |
| SVM + 100 PCA | 0.5401 | 467.9 |
| SVM + 140 PCA | 0.5394 | 583.7 |
| SVM + 180 PCA | 0.5413 | 703.0 |

**3.3 convolution network**

#### 3.3.1 How to Train Model Completely

There can be some confusions on how to train a neural model and bringing the machine learning algorithm into real world. Some questions are frequently needed to be considered: (1) how much is the initial learning rate, 0.1, 0.01 or 0.001 (2)what is the time to reduce learning rate as less learning rate can result in a further improvement of the model. (3)what is the time to stop learning, too much learning can cause over fitting while lack of learning will cause less accuracy.

In this experiment, I utilize the training system by following steps:

1. Finding the largest efficient learning rate. Testing the learning rate begin from 0.1 and if it can not work decrease it into one tenth until it works. I find that 0.1, to some degree, high probability cause NaN problem after the weights change out of the bound, 0.01 can work and I choose the one little less than the 0.01, so the initial learning rate is 0.001.
2. Batch size is 64 which is nearly two times than 30 samples and not too big because it can lead to higher computation cost and memory cost. Mini batch update is faster than full data set update but the gradient descent direction is not as efficient as full data update.
3. Using CIFAR-10 testing data set as indicator to stop learning or descent the learning rate. With the reason that batch update can cause loss and accuracy fluctuation, the more stable indicator curve is necessary. The experiment uses the moving average of cost curve of testing data set as indicator. The decay of moving average function is 0.99, it is better over than 0.95 in the real using.
4. There are two condition can stop the training: (1)the learning rate reach the minimum learning rate, in this experiment the minimum is 1e-7 (2)the moving average accuracy of training data set has reach a enough high level, in this experiment is 95%.
5. The condition that learning rate should be descent is when the moving average of cost curve of testing data set begin to increase, it means the model has over fitting or step into fluctuation stage in current learning rate.
6. In the experiment, the learning rate descent rate is 0.1. When the learning rate descent to its one tenth, the model obviously need some time to adapt the new learning rate, at that time, it will cause fluctuation or even degradation, so I force each learning rate period must at least update larger than 20 mini batch.
7. Print the loss curve and check whether the last period of the curve is enough flat in order to decide to continued training or not.

#### 3.3.2 Experiment Result

In the first section, I experiment three mini models with the CIFAR-10 data set without per-processing function and the figure 6 show the result. Three models has similar parameters: (1) mini CNN model with 406602 (2) mini inception model with 386538 (3) mini residual model with 450058. And the result display that Inception block has a great understanding with the raw image, residual network are little better than VGG block but it consist of more fluctuation.

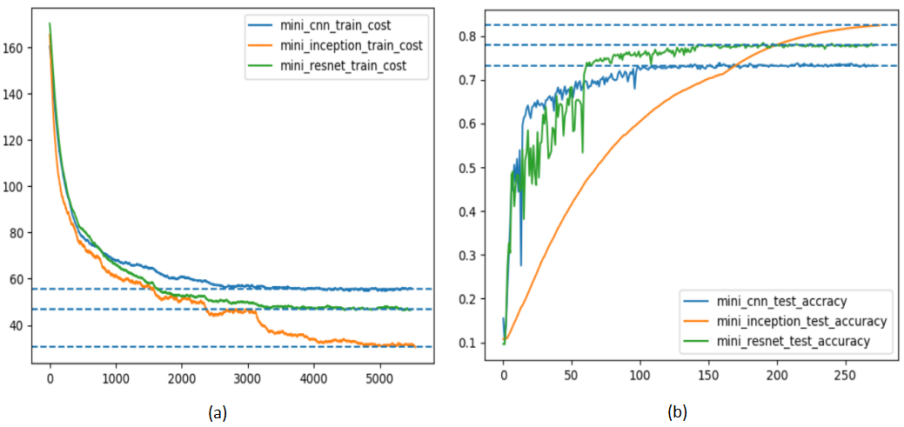


Figure 6: (a)Training Loss of three models (b)Testing Accuracy of three models

In the second section, I also experiment performance between different models after using z-score normalization and the result is shown in figure 7. After using the z-score normalization, residual network become similar with the CNN network with VGG block, and I suppose the advantage of residual network is it can be used in deeper network, so in the similar depth, their performance will be similar.

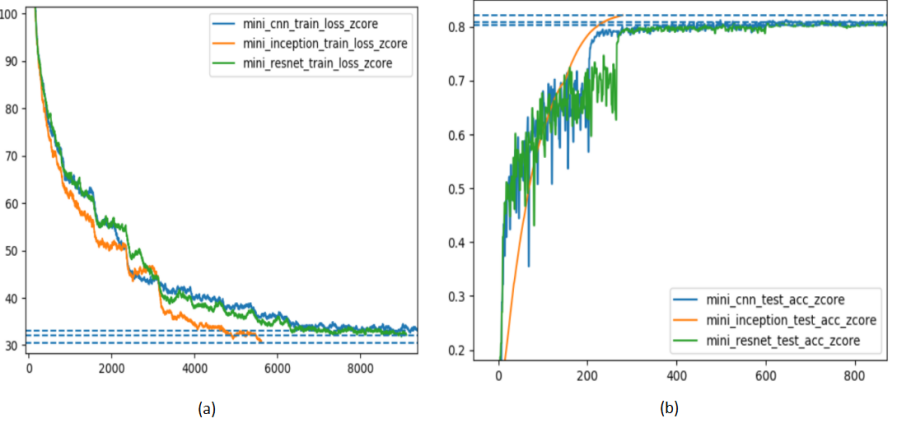


Figure 7: (a)Training Loss of three models after z-score pro-processing (b)Testing Accuracy of three models after z-score pro-processing

The problem of lack of generalization can been seen through figure 8 Even though the accuracy in the training data set for these models has reach 95%, the accuracy of testing data set for all three models still remain at the a little higher than 80% level, the gap between training accuracy and testing accuracy is very large. So I experiment the effect of data augmentation. Data augmentation is a useful method to improve the generalization of the model and narrow the gap between training and testing but it also lead to a question: data augmentation increase the distance between training set and the testing data set, what it means is the entropy of the mapping from training data set to testing data set become larger, so it is difficult for the model to converge after data augmentation. The figure 9 (a) show the condition that after using data augmentation, the result of models become worse decreasing to 75% while the accuracy gap between training data set and testing data set become very small as shown in (b). Before using data augmentation, the generalization ability from training set to real world(testing set) is limit since we gain a enough high accuracy in the training set and need to avoid over fitting. However after using data augmentation, we get a potential improve space for the model to gain higher accuracy.

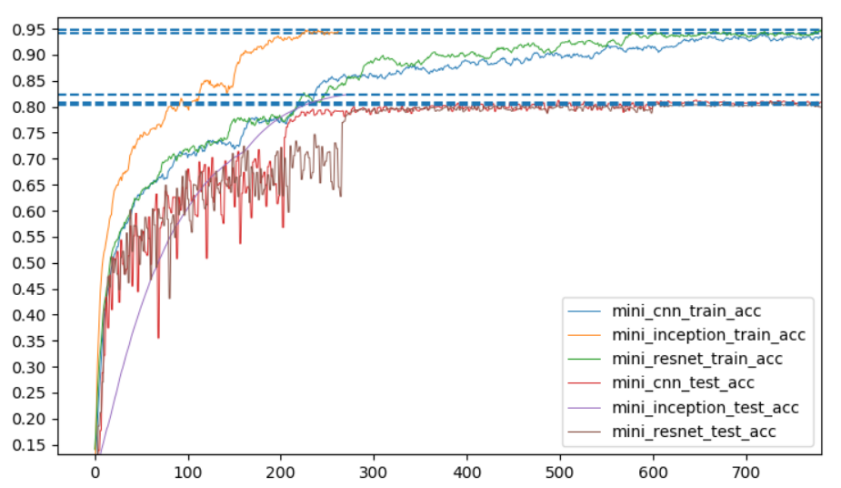


Figure 8: Compare the test accuracy between training set and testing set

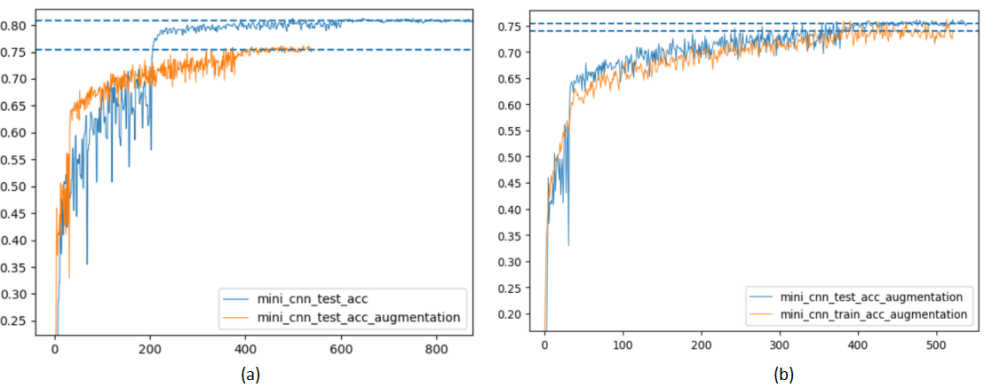


Figure 9: Left:(a)After using data augmentation, it is difficult for the model to achieve the same accuracy before data augmentation since the mapping distance between training data set and testing set become large. Right:(b) In the same time, the gap between training data set and testing data set become narrow.

It is easy to draw a conclusion that the more increasing the model size the better performance may be acquired. I experiment a deeper version of my mini model. In order to release the power of residual model, I adjust the network structure and change the number of layers of the residual network. And the parameters of these three models are still try to limit in a similar range: (1) deeper CNN model with 1191402 (2)deeper inception model with 1181034 (3)deeper residual network with 1278698. And the result is shown in the figure 10. The final result of deeper models goes a bit more 85% which is 5% higher than original models.

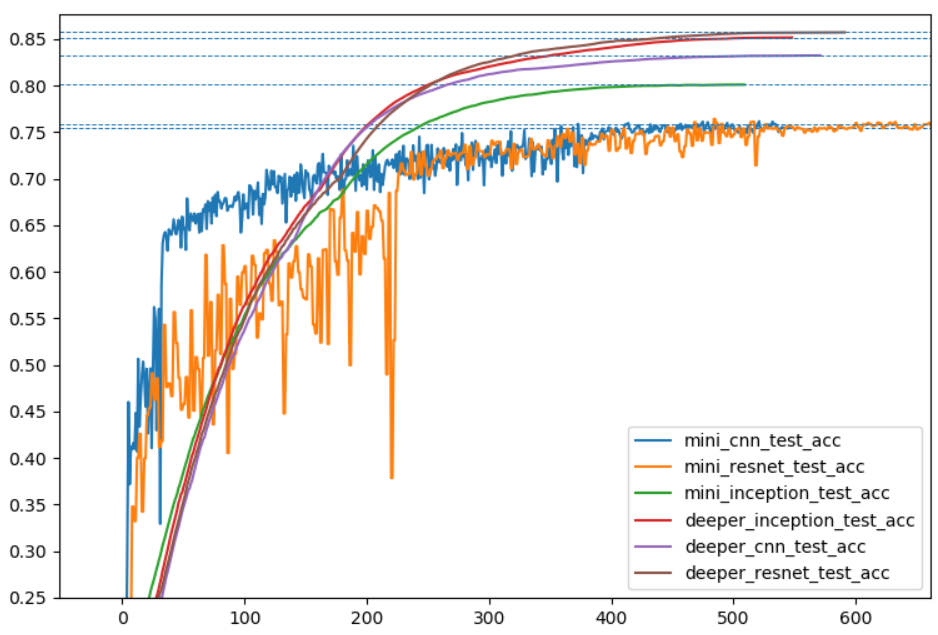


Figure 10: Compare the test accuracy between mini models and deeper models

### Conclusion

In this report we compare the performance of different models on the CIFAR-10 dataset. Firstly, we use SVM and Random Forest to do experiments. PCA as a pre-processing technique can greatly improve the performance of these two models. It reduces the unimportant features and also significantly decreases the computation time.

Then we provide several experiments and study them in the context of different network structure in the image classify task. Although the number of experiments is not enough, some knowledge is concluded until this moment. Inception block has a better information understanding ability than normal convolution operation and can gain a better performance. Residual network with deeper layers with more powerful nonlinear. And deeper and larger model has better performance than smaller model though we intend to minimize the model size.

As we can see from above experiments, convolution network performs much better than some simple models like SVM and Random Forest. And CNN still has great potential to improve its performance on image classification problems.