

Image Classification

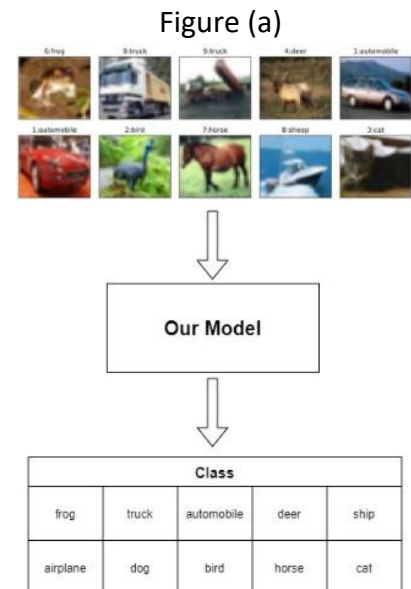
Github repo link

https://github.com/Tiffamy/Object_Recognition-AI_project-

1. Introduction:

The definition of image classification: Given images with a single label, the researcher needs to predict the class of a picture never seen before, and measure result's accuracy. The problem faces many challenges including lighting, noise, angle, scale changes, image distortion, and so on. Image classification is important on computer vision, it can apply to different fields such as self-driving, medical diagnosis, Image monitoring, etc. The wide application of image classification evokes our interest to build a model with high accuracy.

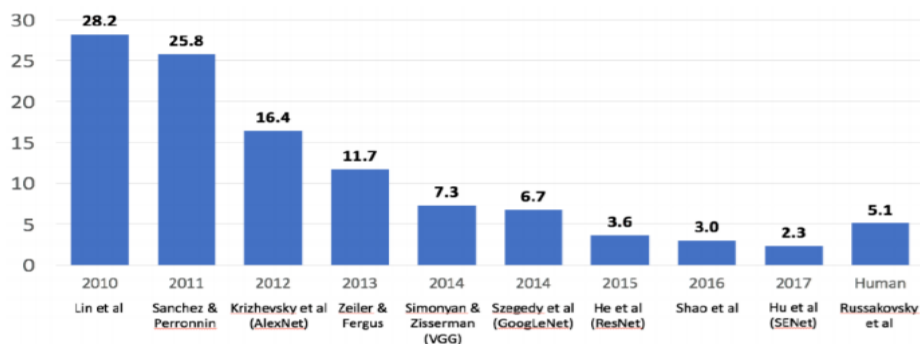
Convolutional Neural Network (CNN) has shown excellent performance in computer vision problems. The purpose of our experiment is to compare the performance of common models, and take the advantages of each models to build a new one and we hope this new model has some progression. We use the dataset: CIFAR-10 to train all the CNNs. CIFAR-10 consists of 60000 32x32 color images in 10 classes, which provides 50000 images as training data and 10000 images as testing data. The performance we concerned includes accuracy and wasting time. The figure(a) shows the concept of solving this problem.



The figure(b) shows the prediction error rate of different models in the ImageNet competition. In 2012, the appearance of Alexnet launched a new era of image classification. From 2015, the improvement of CNN has exceeded human's ability.

Figure (b)

Since 2012, **convolutional neural networks (CNN)** has become the most important tool for object recognition



2.Related Work:

Original CNNs often use 2 x 2 max-pooling to reduce the size of data. However, this method reduced the result of getting features because of disjoint and too fast size-reducing. Fractional Max-Pooling [3], proposed in 2015, randomly chooses 1 or 2 as kernel size one time, getting the size of one layer to be $\sqrt{2}$ on average. As a result, there will be twice as many layers of pooling to get more features.

In the recent paper, usually a deeper neural network outperforms than a fewer layers one in the same method, for example vgg19 has a better accuracy than vgg16. However, this is not always true. In [1] and [2], it's verified that adding more layers to a suitably deep model leads to higher training error. Therefore, a new method [4] "residual block" was proposed to tackle with degradation problem, it aims to optimize the residual mapping than to optimize the original, unreferenced mapping. In this way, their deep residual nets can easily enjoy accuracy gains from greatly increased depth, producing results substantially better than previous networks.

With the ongoing traction of mobile and embedded computing, the efficiency of an algorithm is getting more and more important. A special framework, inception layer [5], is proposed to focus on this problem instead of the pursuit of accuracy so that it could be put to real world use at a reasonable cost.

2-1 contribution:

The main contribution of our work is listed below:

- A. We make a comparison with all the current method we tested and the method we proposed on the performance.
- B. We integrated these models with different techniques to generate our new model which outperforms other models in some specific classes.
- C. We wrote a program to return the predicted class of the image provided by the user.

2-2 Improvement:

Compared to current models, our 3rd and 4th model reveal better performance which exceed 96%. In the specific class, 3rd model has much stronger ability to classify dog with cat, while 4th model specialize in picking up images of birds.

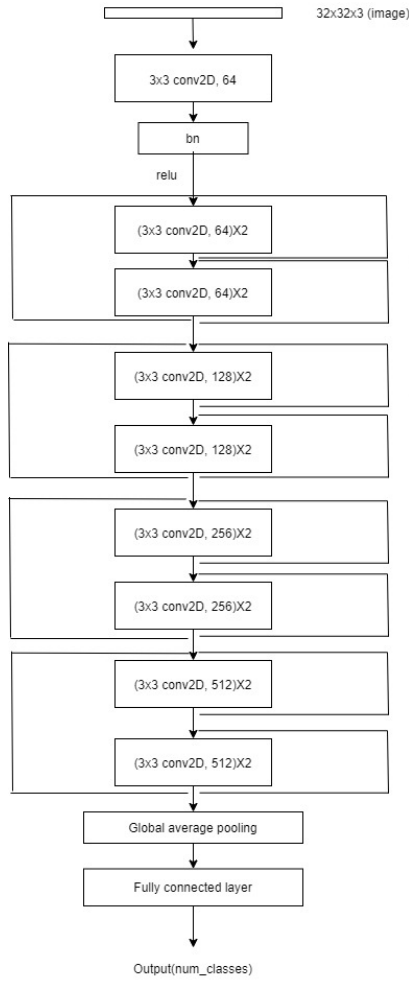


Figure (c)

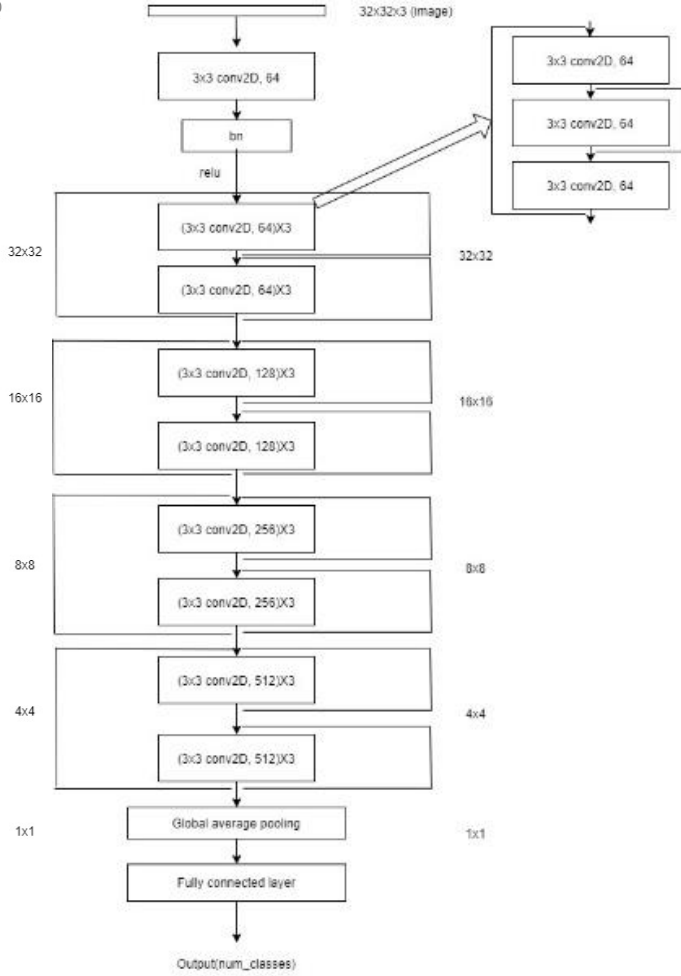


Figure (d)

3.Methodology:

We use resnet18 as original model, applying with different techniques to examine the changes in performance.

The first version of our model structure is defined in figure (c). We add shortcuts across two blocks to make good use of the advantage of residual blocks.

The second version of our model structure is defined in figure (d). Based on the structure of version 1, We replace each block “(3x3 conv2D, 64)X2” with “(3x3 conv2D, 64)X3 with a shortcut between 1st to 2nd layer and 2nd layer to 3rd layer”.

The third version of our model focus on the essential of data augmentation, and this model is based on the structure of our model version 1. Besides improving the structure of models, data augmentation also plays an important role in increasing the accuracy of image classification. Instead of manually designing the augmentation, we exploit “AutoAugment” [6], a search algorithm to find the best choices and orders of image processing function (e.g., translation, rotation, or color normalization). In our code, the technique of AutoAugment is used after data reading and before entering

the model. We also apply fractional max-pooling layer with our model to get more features.

The fourth version of our model is based on our model version 2. Similar to our model version 3, we apply the technique of “AutoAugment” after reading data, and also plus fractional max-pooling layer with our model.

4.Experiments:

We tested and compared the performance of 7 models released from 2014 to 2019, including vgg16, resnet18, resnext29, densenet121, googlenet, mobilenet, mobilenetV2. Based on these method, we integrate them with other technique such as fractional max-pooling, residual blocks, and inception layer in order to create a new model hopefully with better performance.

To test our method, we used Cifar-10 as our dataset. There are 50000 pieces of images used as training data and 10000 pieces of images used as testing data. They are all 32x32 pixels, which were divided into 10 classes such as airplanes, automobiles, trucks, bird, deer, and so on. The advantage of using Cifar-10 lies in these aspects:

- A. They are all true pictures instead of handcrafts.
- B. Each of them only has a main part which makes the model easy to train.
- C. There may be some partial occlusions on images which fits the common situation in life.
- D. The classes have large variance between classes, small variance within classes, which makes it easy to distinguish categories.

In the process of testing accuracy of all models, we found out an interesting fact that most models tend to confuse dog with cat. It can be verified by the confusion matrix at the figure (e). Nevertheless, the two categories can be better classified by our model in the comparison with the original type of resnet18, and the figure (f) is our 3rd model, showing the contrast.

Figure (e): resnet18

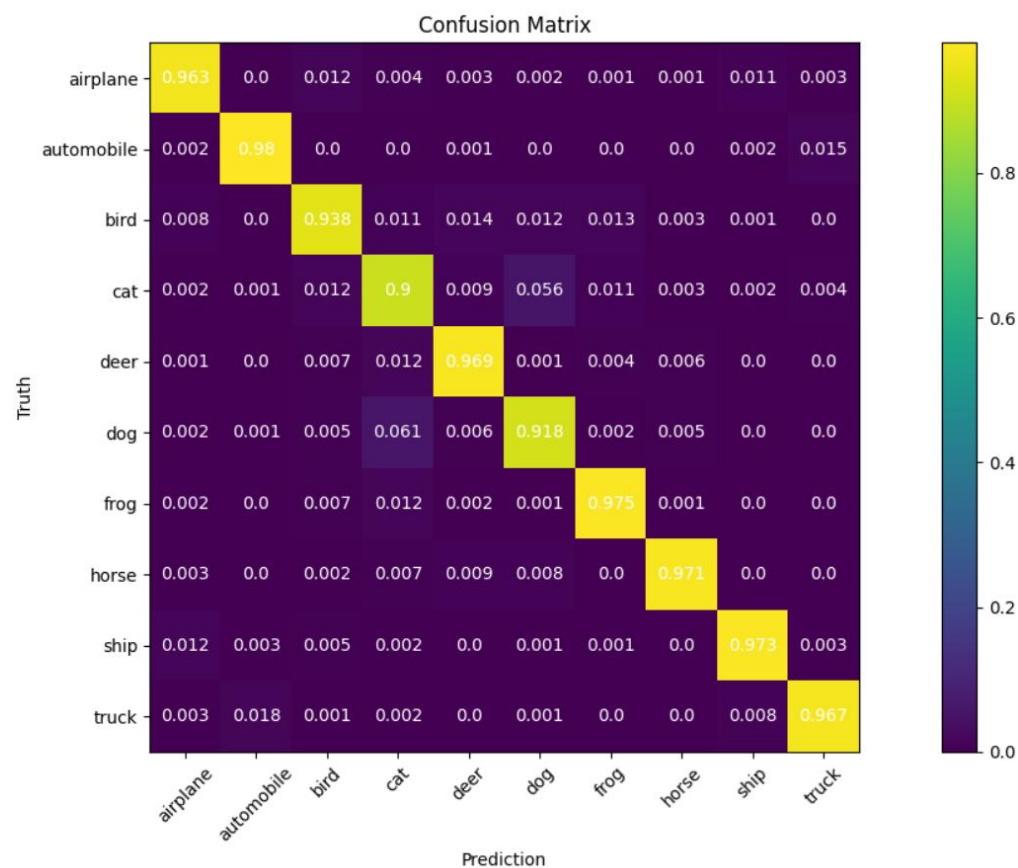
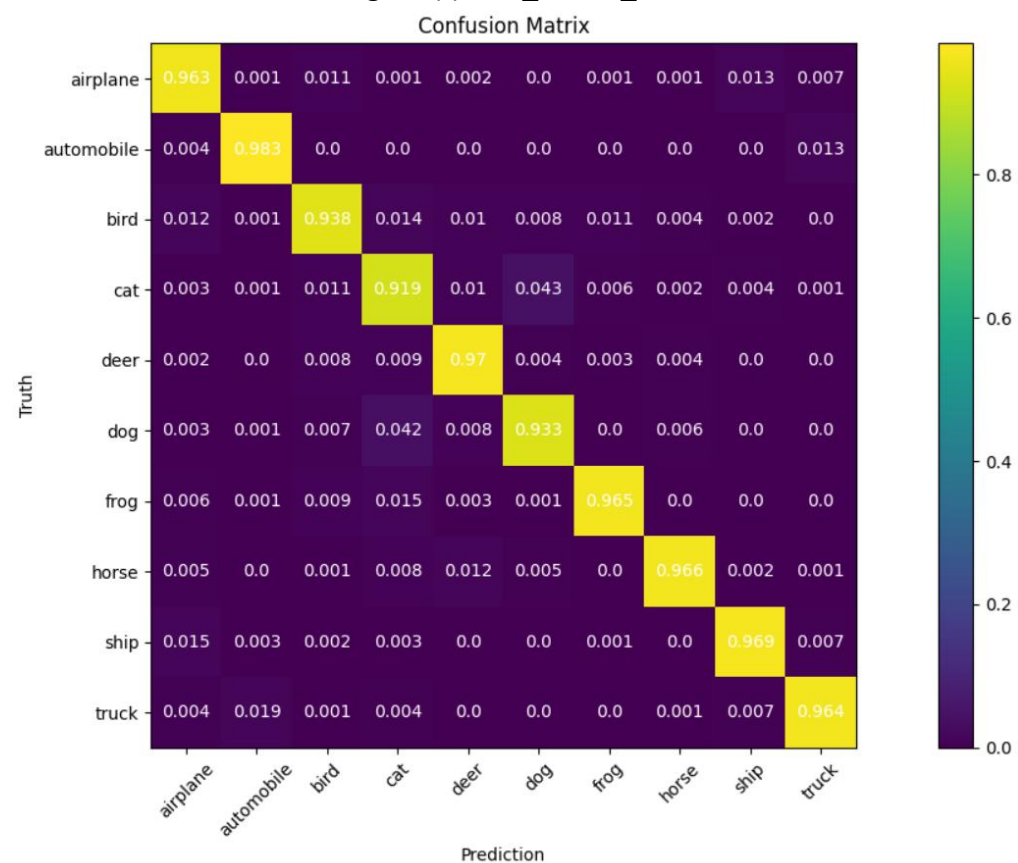
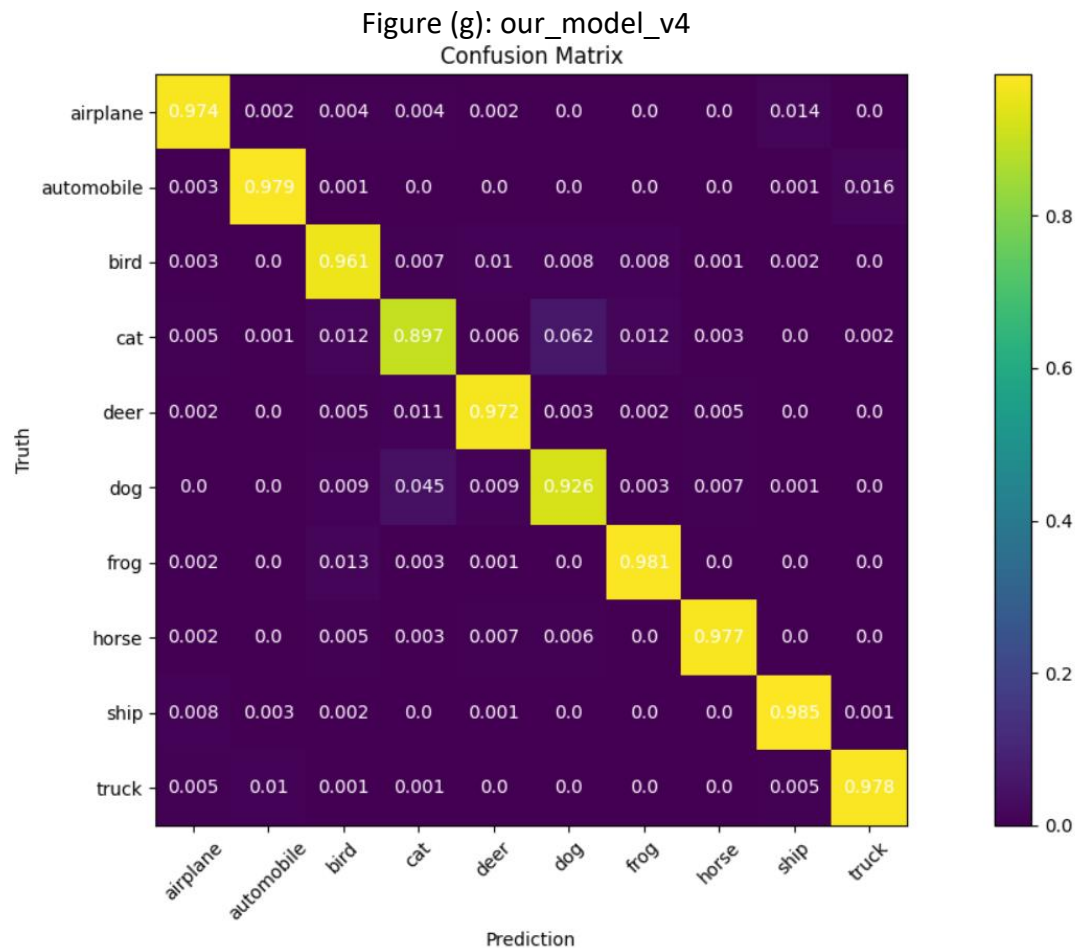


Figure (f): our_model_v3



In the process of pursuing the classification of dog and cat, we encounter a bottleneck that it seems to hit a limit. At the cost of accuracy of dog and cat, we give another try and get higher accuracy in other classes. For example, the accuracy of bird's classification in resnet18 and our_model_v3 are both 93.8%. In the contract, the accuracy of bird's classification in our_model_v4 is 96.1%. As shown in figure (g), it shows a significant growth.



To make it clearer, we fill in all the accuracy of all classes in all models in the table (a). Our 3rd and 4th model are the only two models that shows accuracy higher than 96%. In the phase of training, we found out that these two models' training accuracy grows slower than other models. As a consequence, we speculate that the existence of AutoAugment and the more complicate structure make the model hard to learn, which avoid the model itself walking into the dead end of overfitting.




Table (a)

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	total
mobilenet	91.7	96.5	88.5	79	89.6	87	94.2	92.2	95.1	95	90.88
mobilenetV2	94.1	96.5	90.1	83.1	92.6	88.8	95.5	95.2	95.4	95.3	92.66
resnet18	96.3	98	93.8	90	96.9	91.8	97.5	97.1	97.3	96.7	95.54
resnext	96.2	98.2	94.1	90.4	97.3	93	98	96.7	96.8	97.5	95.82
vgg16	94.5	93.7	85.4	83.2	91.9	88.5	92.8	94	95	95	91.4
densenet	95.9	98.2	94.8	90.4	95.7	93.6	97	97.5	97.8	97.7	95.86
googlenet	96.4	97.7	93.9	90.8	96.2	92.5	97.1	96.8	96.6	96.5	95.45
our_model	96	97.9	93.6	89.7	96.6	91.8	97.9	97.1	97	96.2	95.31
our_model_v2	95.9	98.2	94.8	90.4	95.7	93.6	97	97.5	97.8	97.7	95.7
our_model_v3	96.3	98.3	93.8	91.9	97	93.3	96.5	96.6	96.9	96.4	96.42
our_model_v4	97.4	97.9	96.1	89.7	97.2	92.6	98.1	97.7	98.5	97.8	96.49

Although we have built models with better performance, we are not satisfied with the theoretical value. Therefore, we wrote a program to classify the image provided by the user. Despite the fact that the dataset of Cifar-10 is all true images in real life, we still want to test all the models with pictures of toys, painting, cosplay, or other formats. In the Table (b), we tested all the models with the pictures we found, and recorded the corresponding prediction of each model.

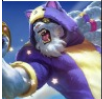
We are curious about a weird situation that our_model_v3 should be the most powerful model in classifying dog and cat of all the other ones. After meditation, we conclude that maybe our 3rd model isn't adapted to different kinds of pictures. Because all the data in Cifar-10 are true images in real life, there is a possibility that our 3rd model is trained based on real world images. As a result, it loses flexibility with objects that are not realistic, for example, logo of bird, cosplay of cat and cartoon of dog.

Table (b)

	our_model	our_model_v2	our_model_v3	our_model_v4	mobilenet	mobilenet_v2	vgg16	resnet18	resnext129	googlenet	densenet121	Answer
	bird	bird	bird	bird	airplane	airplane	airplane	bird	airplane	airplane	bird	bird
	cat	cat	dog	cat	horse	cat	cat	cat	horse	dog	frog	cat
	dog	dog	automobile/ dog	dog	bird	horse	dog	dog	cat	dog	cat	dog

We want to specially mention an example, the character in Table (c) is actually a lion. Specifically, lions belong to cat family, so we wish the prediction of models should be cat. However, among all models, only our 3rd model, which specializes in classification of dog and cat, guessed the right answer.

Table (c)

	our_model	our_model_v2	our_model_v3	our_model_v4	mobilenet	mobilenet_v2	vgg16	resnet18	resnext129	googlenet	densenet121	Answer
	bird	airplane	cat	cat	airplane	airplane	airplane	airplane	bird	automobile	airplane	cat

5.Conclusion:

Through this experience of final project, we understood a paper's structure and tried to follow the framework to sort out our thoughts. We also considered how to design an experiment and form a benchmark for the comparison of methods.

From the aspect of improving our ability, we got familiar with the usage of PyTorch, and we got accustomed to upload our code and explained the corresponding usage for the public. Also, we gradually had an overview of deep neural network at the same time.

However, as we realized more current methods to solve the problem of image classification, we found it difficult to come up with a new implement way to get a better performance. Nevertheless, we still believed that there is always a better way.

After gaining the above abilities, we had more confidence in facing "Computer Science and Engineering Projects (II)" in the next semester. From finding a problem, getting equip with knowledge in the corresponding field, to solving it and comparing the performance between our methods and others, we have learned the overall process of solving a problem.

6.Reference:

- [1] K. He and J. Sun. Convolutional neural networks at constrained time cost. In CVPR, 2015.
- [2] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. arXiv:1505.00387, 2015.
- [3] <https://arxiv.org/abs/1412.6071>
- [4] <https://arxiv.org/pdf/1512.03385v1.pdf>
- [5] <https://arxiv.org/pdf/1409.4842.pdf>
- [6] https://openaccess.thecvf.com/content_CVPR_2019/papers/Cubuk_AutoAugment_Learning_Augmentation_Strategies_From_Data_CVPR_2019_paper.pdf