CIFAR - 10: Colored Image Classification

Chang Li Qihan Zhao
Department of Electrical and Computing Engineering
University of Washington

Abstract

The CIFAR-10 is an established computer-vision dataset used for colored image classification. In our experiment, we employ three methods for the classification [1]. One is the traditional machine learning method named as support vector machine (SVM) associated with the Histogram of Oriented Gradients (HOG), which proves to outperform in feature extraction [2]. The other two are classical convolutional neural network (CNN), including VggNet [3] and ResNet [4]. We implement the CNN architecture based on the paper[2][3], reproduce the promising results, which obviously show that deep learning approaches outperform the machine learning one and the network will achieve higher accuracy with deeper layer and better feature extraction.

1. Introduction

The CIFAR-10 is a subset of 80 million tiny images dataset and consists of sixty thousand 32 x 32 color images belonging to one of ten classes, where each class owns six thousand images [1]. It is an established computer-vision dataset used for colored image classification. Our project employs the CIFAR-10 dataset and three classic methods associated with several experiments for the colored image classification. Our focus is on two fields: the comparison between performance from the machine learning algorithm and the deep neural network, the behaviors of the extremely deep neural network. In the first experiment, the HOG [2], a built-in library in python named as skimages.features, captures the very characteristic gradient structure of the local shape and extracts the feature set as the input. Then, the linear SVM, based on the built-in library sklearn in python, as the baseline classifier accepts the preprocessed inputs and make prediction on the test set, consist of ten thousand colored images. In the second experiment, we implement the VggNet architect, based on the paper [3] and the framework from TensorFlow, with a variety of the depth and the combination of convolution layers associated with pooling layers, named as Vgg9, Vgg11 and Vgg15. It is obvious that the network performs outstandingly with deeper layer but longer time to converge. In the third experiment, the other classic network type of CNN performs in both the plain layer, simpler stack layer, and the residual net, with the shortcut connections. It proves, to some extent, that the performance of the network has the positive relationship with the depth, however, going deeper does not necessarily boost the classification performance.

2. Dataset

The CIFAR-10 is an established computer-vision dataset used for colored image classification. It is a subset of 80 million tiny images and consists of sixty thousand 32 x 32 colored images belonging to one of ten classes, with six thousand images per class. The labels in the dataset are airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Fifty thousand images as training data feeds into the SVM classifier and the remaining ten thousand images as test data to verify the performance. For the deep learning algorithms, VggNet and ResNet, the fifty thousand training data is split into forty-five thousand training data and five thousand validation data. The same as the SVM classifier, ten thousand images sever as the test data to verify the performance.

3. Methodology and Experiments

3.1 Method1: HOG + SVM

The linear SVM serves as the baseline classifier for the machine learning algorithm, which is named as sklearn.svm package in python. The HOG algorithm focuses on feature extraction, which is named as skimages.features in python. The basic idea for HOG is to form the feature representation by diving the image into small cells and accumulating one dimensional histogram of each gradient directions over the pixels of each small cell. The paper [2] applies the combination of those accumulated histograms as the presentation or input to the classifier. As mentioned before, fifty thousand images serve as the training data for feature extraction and classifier train. And ten thousand images are used to evaluate the performance. After feature processing, classifier training and parameter tuning, the performance reaches around 50.83% in our test set. The poor performance intrigues us to replace the machine learning algorithm with the neural network. We believe that neural network will learn better features and achieve higher accuracy than the SVM classifier.

3.2 Method2: VggNet

Deep CNNs, such as VggNet and ResNet, are expected to be the dominant approach for feature extraction through the combination of several jointly layers of data processing and non-linear operators. We implement the classic VggNet to show how powerful the CNN model in image classification, compared to the machine learning model, and how good the performance could achieve. The framework is based on TensorFlow and we implement the combination of convolutional layers and operators. There are three types of architecture that we conduct experiment on. The first type, named as Vgg9, consists of three uniformed layers and three fully-connected layers. The uniformed layer is composed of two convolutional layers and one max pooling. The second one, named as Vgg11, consists of four uniformed layers and three fully-connected layers. In our case, the convolutional layer is set up as 3 x 3 with 16 filters, 32 filters and 64 filters. The uniformed layer includes two convolutional layers and one max-pooling layer. The fully-connected layer is in 64, 64 and 10. The purpose of Vgg9 and Vgg11 is to verify whether the depth of connected blocks influences the performance. The third architecture, name as Vgg15, is composed of four uniformed-two layers and three fully-connected layers. The uniformed-two

layer is the combination of three convolution layers, which set up as 3 x 3 with 16 filters, 32 filters and 64 filters, and one max-pooling layer. The purpose for Vgg11 and Vgg15 is to verify whether going deeper in layers will improve the classification performance. The accuracy for Vgg9, Vgg11 and Vgg15 is in slight difference 71.65%, 72.96% and 72.50%. The Results part will explain the comparison in detail.

3.3 Method3: ResNet

Both VggNet and ResNet are classical deep CNNS, which are both expected to achieve outstanding performance in our experiment. The purpose for VggNet and ResNet is to verify the general performance of deep CNNs extremely outperform the machine learning algorithms. We implement the ResNet based on the framework from TensorFlow and the paper [4]. There are three types of ResNet, which differs in the connections of blocks and the depth of layers.

The first type, named as Plain32, consists of fifteen uniformed layer, one average layer and one fully-connected layer. The second one, names as Resnet32, also consists of fifteen uniformed layers, the global average-pooling layer and one fully-connected layer. The uniformed layer is composed of two convolutional layers with hand-pick number of filters, which serves as down sampling. However, the way to connect uniformed layers for Resnet32 is slightly different from Plain32. In the Resnet32, we insert the shortcut connections [4] based on the Plain32. For example the Plain32 make direct connection among each uniformed layer and the fully-connected layer. But the Resnet32 make an extra connection between the input to each uniformed layer and sum it up to the output. In the other word, if the learnable representation from each uniformed layer does not have a positive or dominant effect in the result, which will be considered as useless and be abandoned. The purpose of Plain32 and Resnet32 is to verify the effect of the shortcut connection, where Resnet32 achieves 2% higher accuracy than Plain32. The third type, named as Resnet110, consists of thirty-six uniformed-two layers, the global average-pooling and the fully-connected layer. The uniformed-two layer is composed of three convolutional layers with hand-pick number of filters. The purpose for Resnet32 and Resnet110 is to verify whether going deeper will boost the performance. The Resnet110 achieves slightly better perform than Resnet32. The Results part will dive more into the data analysis and result comparison.

4. Results

4.1 Vgg9, Vgg11, Vgg15

network	experiment	Training time	Prediction accuracy
VGG	Vgg9	0:07:54	71.65%
	Vgg11	0:08:43	72.96%
	Vgg15	0:09:23	72.50%

Table1. Experiments for Vgg9, Vgg11 and Vgg15

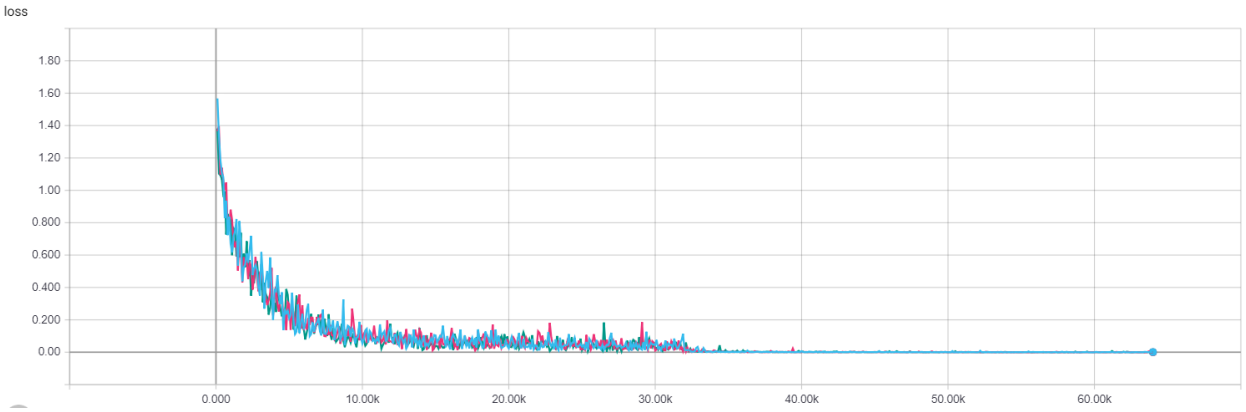

Figure 1. training loss (vgg9: green, vgg11:blue, vgg15: red)

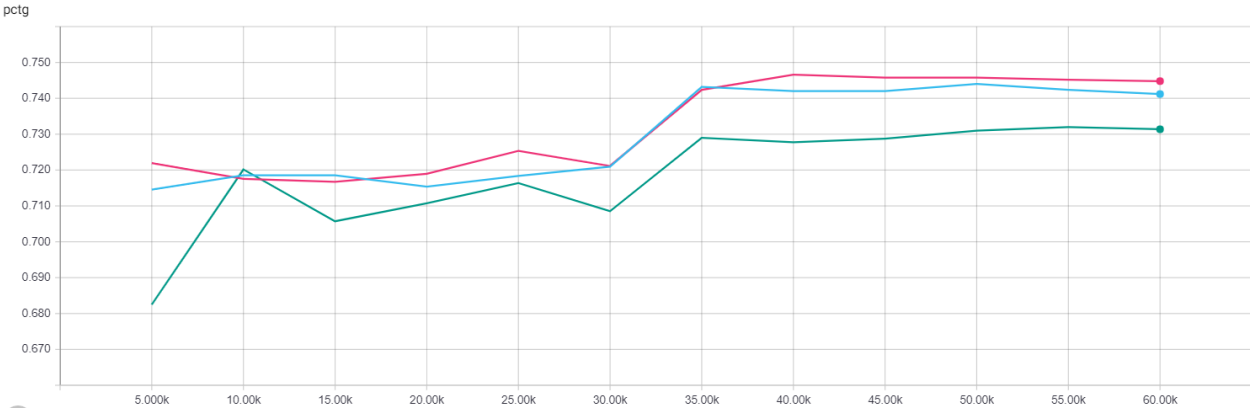

Figure 2. validation accuracy (vgg9: green, vgg11:blue, vgg15: red)

Vgg neural network works but does not performs better with deeper nets. Based on the Table1, it is obvious that the VggNet achieves the accuracy around 70%, where Vgg11 shows the best performance. In the other word, the depth does not necessarily boost the performance in our cases, which might due to that the CIFAR-10 dataset is a such simple classification.

4.2 Plain32, Resnet32, Resnet110

network	experiment	Training time	Prediction accuracy
Resnet	Plain32	0:33:12	81.56%
	Resnet32	0:34:22	83.12%
	Resnet110	2:47:30	83.94%

Table2. Experiments for Plain32, Resnet32 and Resnet110

Plain32 vs Resnet32

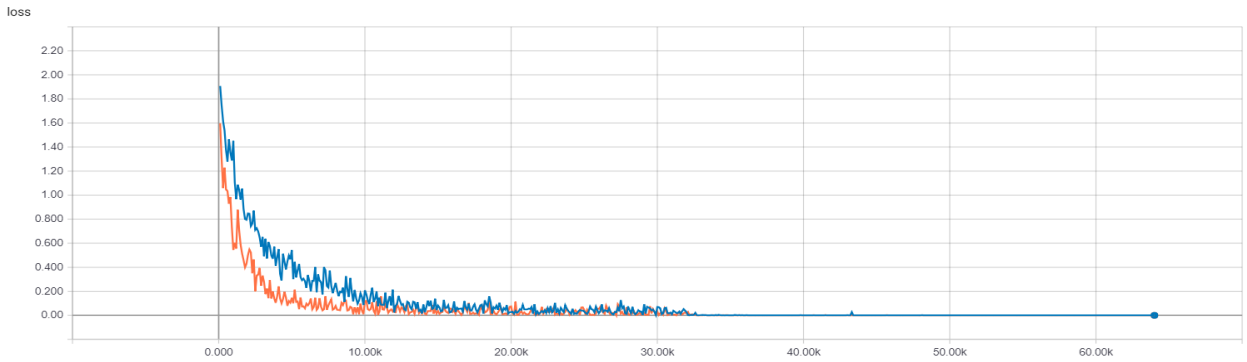

Figure 3. training loss of resnet32(orange) and plain32(blue)

Resnet32 performs better than Plain32, its plain counterpart, and it converges faster during training. The observation proves our assumption that the performance, under other same conditions, should be improved due to the insertion of the shortcut connect among layers. That is because the shortcut connection ensures that each uniformed-two layer provides learnable feature matrix, otherwise, the output will be overlooked.

Resnet32 vs Resnet110

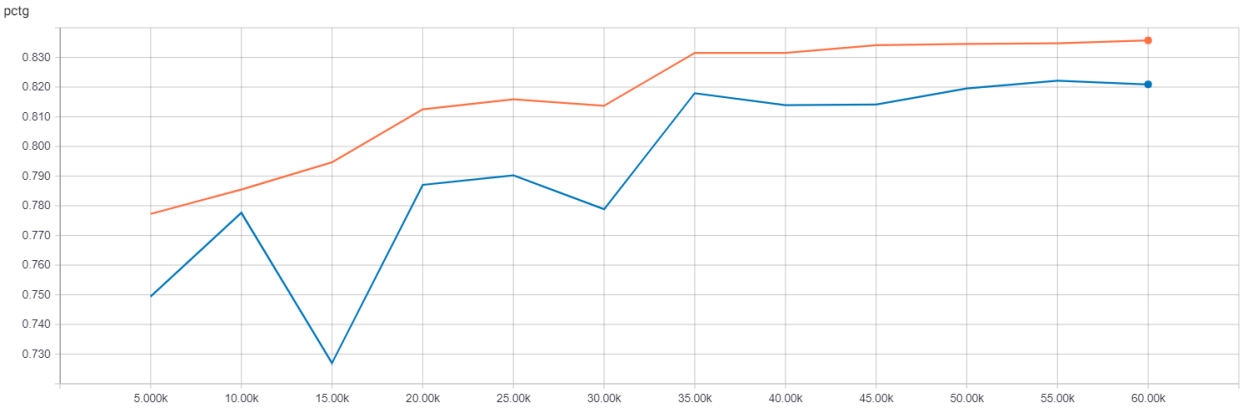

Figure 4. validation accuracy of resnet32(orange) and plain32(blue)

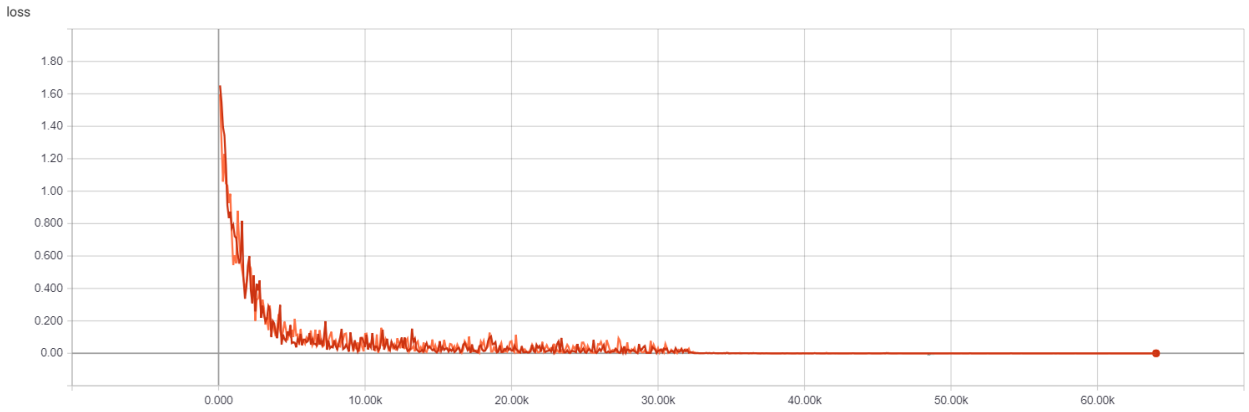

Figure 5. training loss of resnet32(orange) and resnet110(red)

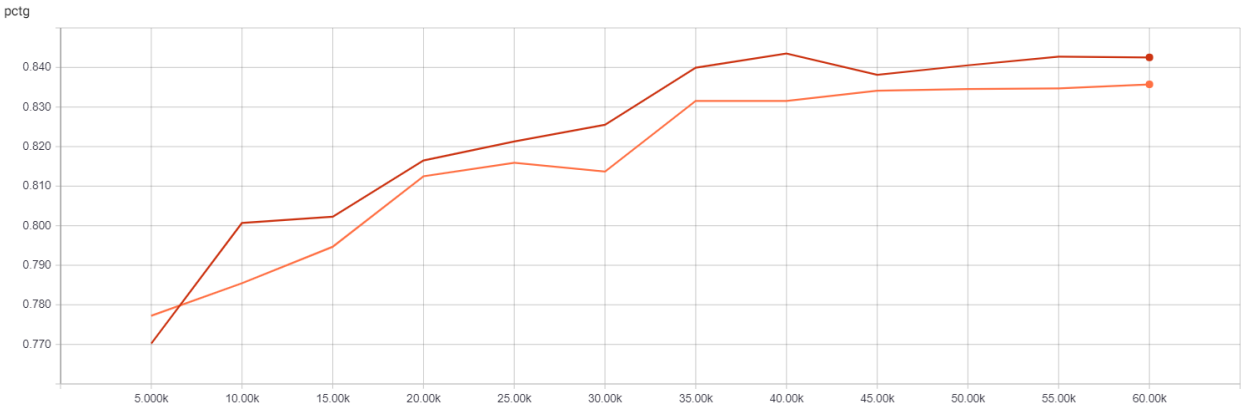

Figure 6. validation accuracy of resnet32(orange) and resnet110(red)

Our assumption is that, with the increasing of the depth of the neural network, the accuracy should be greatly improved. However, increasing the depth of ResNet from 32 to 110 does not boost the classification performance in a significant way but requires much more training time. The result is pretty like the experiment between Vgg11 and Vgg15.

4.3 SVM vs CNN

The best accuracy for the deep CNN could achieve 83%, however, the best one for SVM is around 50%. It shows that the feature matrix or parameters learnt from the machine learning algorithm is not such representative and can not be applied to those test cases. In the other word, the deep neural network employs complex convolutional layers, learns multi-dimensions feature matrix and applies to the general cases much better.

4.4 VggNet vs ResNet

Based on our experiment, the ResNet performs better than the VggNet, where the former one achieves 10% more accuracy than the later. It is hard to assume before the experiment, such as the ResNet should learn better parameters and achieve higher accuracy. That is because each deep

neural network has its own characteristics and advantages. Then, the relationship between the dataset and the learnable feature matrix will be dominate affect which neural network might perform better, which is the reason to conduct experiments on various deep CNNs.

5. Conclusion

Deep neural network methods perform better than traditional machine learning methods, which requires hand-pick feature extractions. Resnet performs better than VGG, proving the power of residual learning. For a given type of network, going deeper does not necessarily boost classification performance. Perhaps it is because Cifar-10 is a rather easy task as for classification. In the meantime, deeper networks require more parameters, operations, and training time. A balance should be sought between network performance and computational efficiency.

References

[1] Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2018). Do CIFAR-10 Classifiers Generalize to CIFAR-10?. *arXiv preprint arXiv:1806.00451*.

[2] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.

[3] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Appendix

How to run the code:

Step1:

Modify input and output file paths according to your local paths.

Step2:

Do data preprocessing work in utils.py. Run the script with ‘python utils.py’. After this, training set, validation set, and test set should be saved in npy form.

Step3:

Experiment with hog+svm classifier. Change all the file path in the script if needed, and then run the script with ‘python hog_svm.py’.

Step4:

Experiment with both deep learning methods. Six networks (from two methods) are implemented in network.py. Modify the string net_name in config.py to test on different networks. You can also choose a different GPU or tune the hyper-parameters in config.py. After this, run “python main.py”, a model will be trained and tested.