Car Classification using Neural Network

Group 17
Feifan Xu Yifan Ruan
Ke Han Yincheng Mao
{fex014, yir021, keh101, y7mao}@ucsd.edu

Introduction

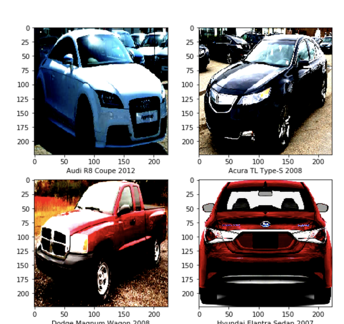
Car classification and detection are crucial to multiple fields including traffic management by government and vehicle business by companies. To enhance the accuracy and efficiency of classification, neural network is a promising method. In our project, we will implement several models to classify a dataset of car images and compare their performance.

Dataset

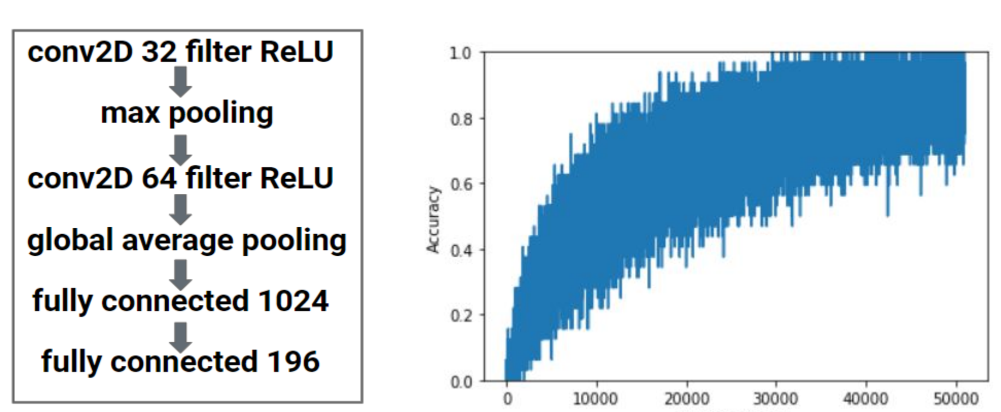
We used the Stanford Car Dataset which has 16,185 imaged into 196 classes. The Cars dataset contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of Make, Model, Year, ex. 2012 Tesla Model S or 2012 BMW M3 coupe.

Features Dataset

In this project, we crop images to get better feature maps. The examples shown right are resized as (224, 224) and normalized in [-1, 1].



Model 1: Baseline



Hard to convergence, test accuracy is 24.86%

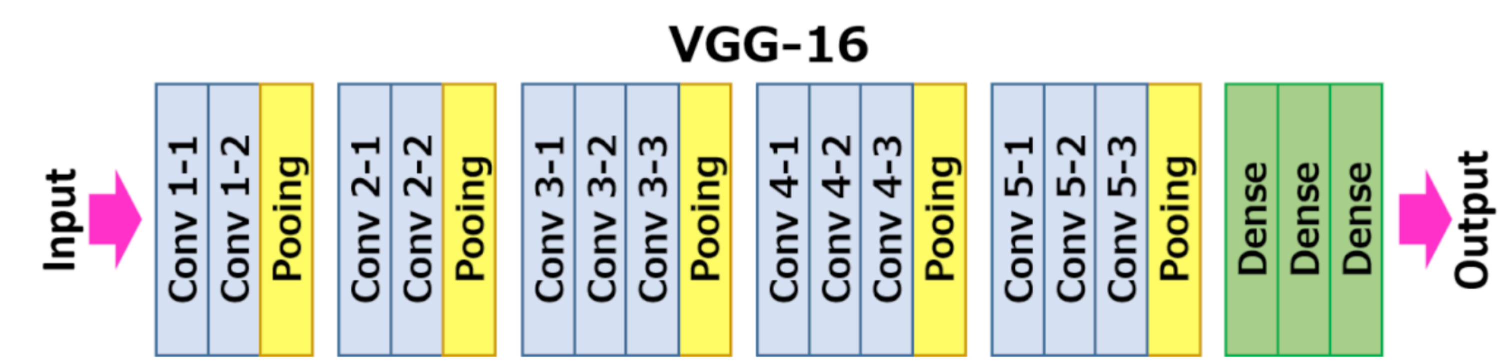
Model 2: ResNet

Methods

ResNet is short for Residual Network. We have been using ResNet18, ResNet50 and ResNet152 in this project.

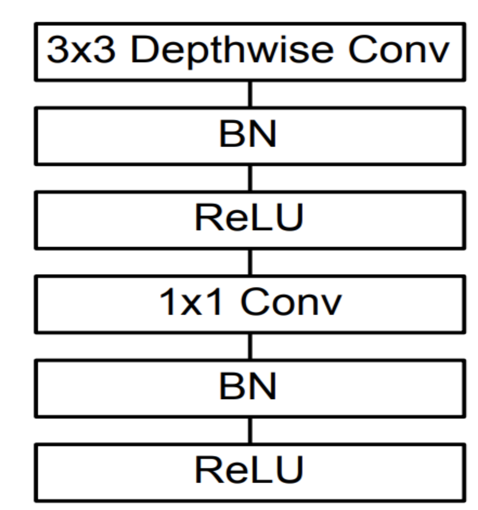
Model 3: VGG

A deep convolutional network for object recognition. The model achieves 92.7% top-5 test accuracy in ImageNet.

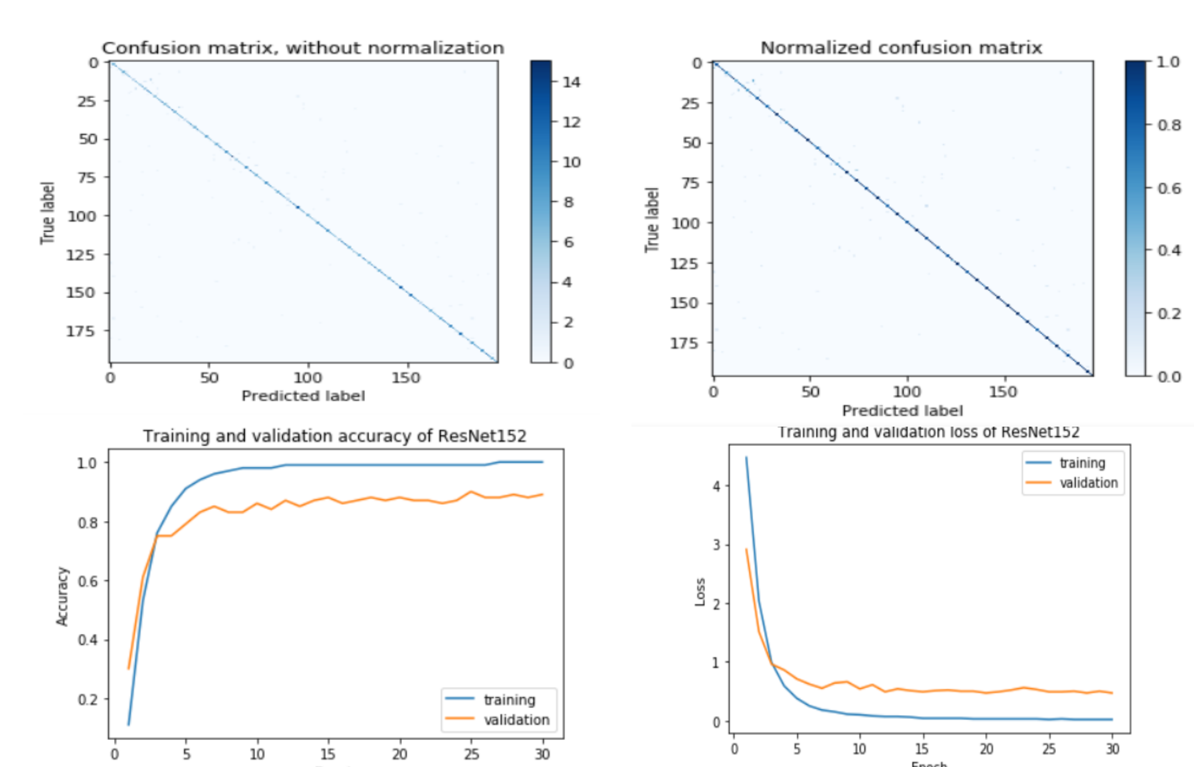


Model 4: MobileNet

The trick of MobileNet is to split standard convolution layer into two layers: depthwise convolution layer and pointwise convolution layer. It reduce the computation compared with standard convolution.

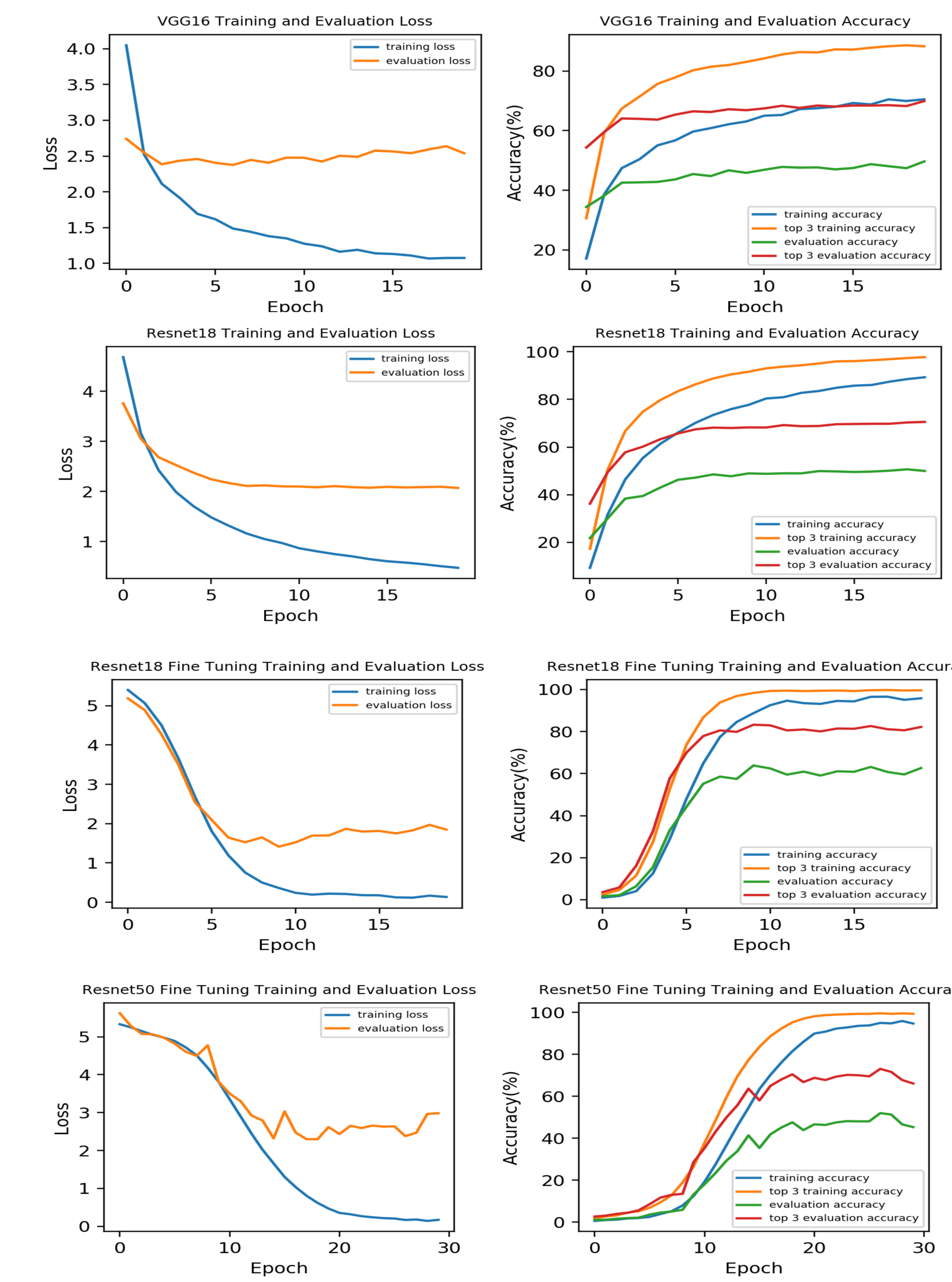


Result of ResNet152



Discussion: compared to vgg19, ResNet152 has an incredibly high number of layers (152 depth). With the increment of depth, neural networks are able to extract complicated features. However, degradation problem also appears with the increment and might result in a worse result. To solve this, ResNet performs a residual learning by adding new layers on a shallow network to finish the deep neural network and this structural trick helps to improve the performance.

Result of ResNet18 ResNet50 and vgg16



Discussion: This experiment is implemented using pytorch. From the results above, we can see that Resnet18 with fine tuning has the highest test accuracy (though still lower than Resnet152). This come from the reason that for deeper model such as Resnet50, the size of our data may not large enough for training. For VGG, due to its large number of parameters, it is also not easy to achieve a good result within only 20 epochs.

Future Work

- Try to use different Loss Function such as Focal Loss
- Try to use Resnext as feature extractor
- Include more image preprocessing, add dropouts

Reference:
[1] Krause, Jonathan, et al. "3d object representations for fine-grained categorization." Proceedings of the IEEE International Conference on Computer Vision Workshops. 2013.
[2] 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).
[3] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
[4] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).