Team 40 ROB 535 Perception Report

Chia-Ching Tsai, Chia-Yin Lin, Wan-Yi Yu, Yi-Cheng Lai, Yu-Min Jian, Yueh-Lin Tsai

***Abstract*—The Convolutional Neural Network (CNN) has been widely used in the self-driving car's industry. It brings the possibility for cars to identify different objects on the road. In this work, we implemented a modified VGG11 architecture to conduct vehicles classification in images from a game engine, achieving up to 49% categorization accuracy.**

I. Introduction

IMAGE classification using CNN model could be separated into three main different parts, which are data labeling, separating training data and validation data, preprocessing images, and training.

II. Methodology

A. Data Labeling

Before putting images into CNN model, every image in the training data set should be given a label(in this task, labels are 0, 1, 2 for different types of vehicles). The team changes the labels into a 1 by 3 matrix, where [1,0,0] for 0, [0,1,0] for 1, [0,0,1] for 2. Output in matrix would be more suitable for the cross entropy function which is the loss function the team use.

B. Separating training and validation data

The team separated $\frac{1}{10}$ of training image set to be the validation data. It helps avoid overfitting for training model, especially non-linear model like CNN. If there is no validation set for checking the overfitting, the model might get 100 percent accuracy on the training data set but perform poorly on the testing data.

C. Image preprocessing

The team use randomflip() and randomhorizontal() functions to randomly rotate or flip the images in training data set. This will help introduce the variety of images into the data set by adding images that seem like taken in different scenes and will help increase the accuracy for testing.

D. VGG11 model

The reason for using this model for image classification is that there are fewer hidden layers and weight parameters inside the model. In addition, it is a small model suitable for simple image classification such as dog and cat classifying. Because of that, fewer resources are required during training and good results can still be obtained when the GPU resources are insufficient.

III. Results

The team get an accuracy of 49% for the testing data. The team reduces the number of fully connected layers at the back section to two and add ReLU function after the first fully connected layers. In addition, to speed up the training process the team resize the image into 256 by 256 pixels. This step discard a great deal of information in the image and causing a lower accuracy performance.

IV. Future Work

The result still have a large room for improvement. To further improve the accuracy of classification, the VGG11 model could be modified to VGG16 or VGG19 to increase hidden layers and weight parameters for training. The trade-off would be the cost of time for training. Or maybe, the team can try model with different architecture such as ResNet. There is a unique architecture in ResNet, called Residual Block, which includes a shortcut for input data to skip the step of convolution and activation function during training. The original input data will be combined with the output of those layers before being fed to the next block. It might help reserve the information before convolution and activation to enhance accuracy.

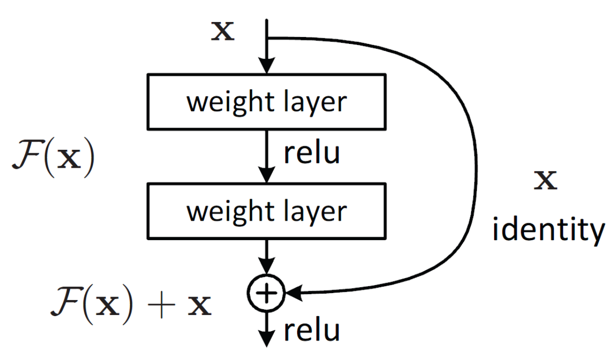

Fig. 1: Residual Block
[1]

References

[1] S. R. J. S. Kaiming He, Xiangyu Zhang, *Deep Residual Learning for Image Recognition*, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385