Car Detection

qifengc2@uci.deu/ID:57004415

May 1, 2022

Contents

1 Introduction

2021 can be said to be the first year of commercialization of autonomous driving. Governments, auto industry giants, and even many Internet companies have begun to focus on the field of autonomous driving.
Object Detection based on deep learning have been widely used in visual perception systems for Autonomous Driving. As an important part of Object Detection problem in autonomous driving, Cars Detection still has a very broad development prospect.
I used to work in an autonomous driving company and am very interested in object detection based on DNN, which is also one of the reasons why I choose this application.

2 Datasets

The dataset contains vehicle data and non-vehicle data. The size of all the images is 64 × 64. There are 8793 RGB images in vehicle data set and 8969 RGB images in non-vehicle data set.

(a) Image in vehicle set

(b) Image in vehicle set

(c) Image in non-vehicle set

(d) Image in non-vehicle set

Figure 1: Image In Dataset

The dataset itself does not split the data into training and test sets. In order to facilitate subsequent operations, we will divide both vehicle and non-vehicle data sets into two parts: training set and test set.
The test set accounts for 1/4 of the total data, and the training set accounts for 3/4 of the total data.

3 Model

3.1 Model Architectures

My model contains three convolutional layers and one fully connected layer.

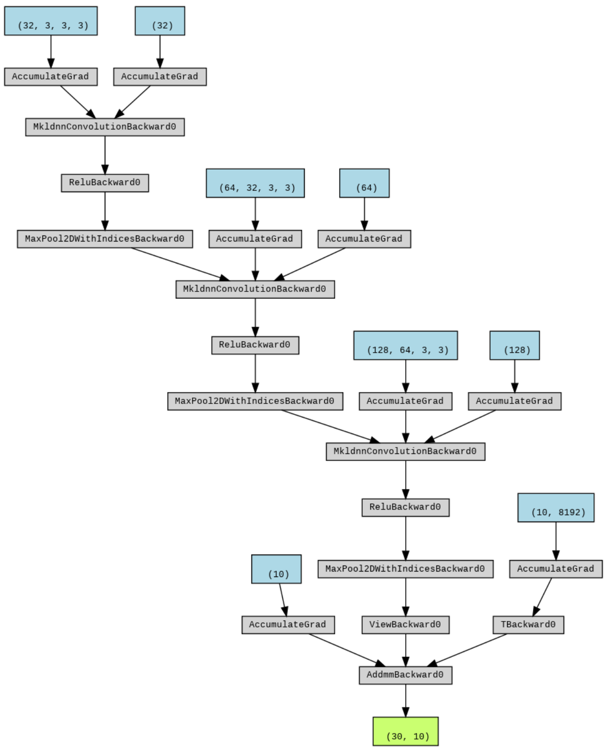

Figure 2: Model Architecture

3.2 Convolution Layer

The structure of each convolution layer is convolution function + Activation Function + pooling function. The size of the convolution kernel is 3x3, the default stride is 1, and the padding is 2.
The first convolutional layer has 3 input channels and 32 output channels, the second convolutional layer has 32 input channels and 64 output channels, and the third convolutional layer has 64 input channels and 128 output channels.

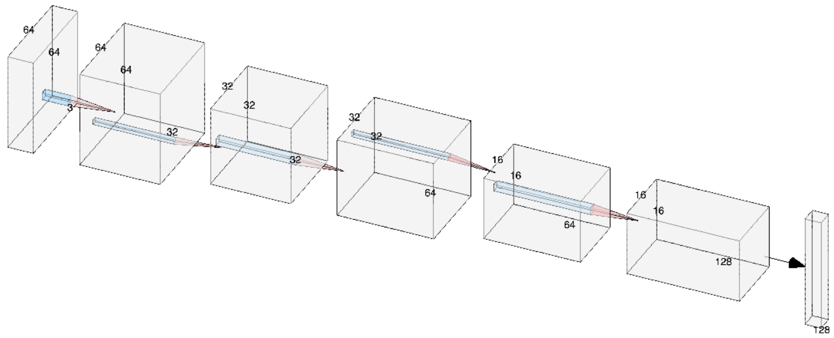

Figure 3: CNNs Model Architectures

3.3 Fully Connected Layer

The fully connected layer is one layer, the size of the kernel is 8x8, and the output is 10.

Layer Name	Channel Size	Kernel Size	Input Size	Output Size
Cov1	32	3×3	3×64×64	32×64×64
Pool1	32	2×2	32×64×64	32×32×32
Cov2	64	3×3	32×32×32	32×32×64
Pool2	64	2×2	32×32×64	16×16×64
Cov3	128	3×3	16×16×64	16×16×128
Pool3	128	2×2	16×16×128	8×8×128
Fc1	10	8×8	8×8×128	10

Table 1: The Table Of The Layers

4 Training the model

In order to study the influence of different factors on the accuracy of the results, I choose to control the variables to train the model.

4.1 Activation Functions effect

To study the influence of activation function on Inference Time, Training Time and accuracy, I will take the method of controlling Optimizer and Learning Rate and changing the Activation Function of the model for training.
I select Adam as the Optimizer and 3e-4 as the Learning Rate. I will use ReLU, Sigmoid, LeakyReLu, ReLU6, and GELU as activation function to train the model.

4.2 Different optimizer effect

To study the effect of the Optimizer and different epochs on the accuracy, I will take the method of controlling the Activation Function and Learning Rate, and changing the Optimizer and epochs of the model for training.
I will choose Adam as the Activation Functio and 3e-4 as the Learning Rate value. I will use SGD, Adam, AdamW, Adadelta, Adamax, ASGD as Optimizer and train the model for 10 epochs, 20 epochs, 50 epochs for each Optimizer.

4.3 Different learning rate effect

To study the effect of Learning Rate and different epochs on accuracy, I will take control of Activation Function and Optimizer, and change the Learning Rate and epochs of the model for training.
I will choose Adam as the Activation Function and Adam as the Optimizer. The Learning Rate values are 3e-12,3e-8, 3e-4, 3e-2 and 300 and the models will be trained for 10 epochs, 20 epochs, and 50 epochs for each value.

5 Results

5.1 Activation Functions effect

Activation Function	ReLU	Sigmoid	LeakyReLu	ReLU6	GELU
Inference Time	1	2.3%	1.9%	2.3%	1.5%
Training Time	1	1.2%	3.0%	2.8%	1.8%
Accuracy	97%	91%	96%	97%	97%

Table 2: Activation Functions effects on Trainng time and Accuracy

I measure the training time and inference time of ReLU and set ReLU's ratio as unit 1. By measuring the training time and inference time of other activation functions, comparing them with the training time and inference time of ReLU and calculating the percentage, the ratio of training and testing of other activation functions relative to ReLU is faster or slower.
By comparison I found that most of the activation functions have little effect on the training and test accuracy of my model, but the sigmoid function makes the model test accuracy have a certain reduction.
Different activation functions have different degrees of influence on training time and inference time, and ReLU has relatively good accuracy and relatively short training time and inference time.

5.2 Different optimizer effect

Optimizer	SGD	Adam	AdamW	Adadelta	Adamax	ASGD
10 epochs	92%	97%	96%	85%	97%	95%
20 epochs	93%	98%	96%	90%	97%	95%
50 epochs	93%	96%	93%	92%	97%	94%

Table 3: Different optimizer and how the accuracy changes

By observing the measurement results, it can be found that different optimizers have different effects on the accuracy. Models trained with Adam and Adamax perform better, while Adadelta is relatively poor.
For the optimizer with relatively poor training effect, the effect can be improved by increasing the epochs, but the improvement is obviously reduced as the epochs continue to increase. For an optimizer with relatively good training effect, increasing epochs may make the training effect worse.

5.3 Different learning rate effect

LR	3e-12	3e-8	3e-4	3e-2	300
10 epochs	0%	69%	97%	49%	0%
20 epochs	0%	67%	98%	50%	0%
50 epochs	0%	80%	98%	50%	0%

Table 4: Different learning rate and how the learning rate change the results

By observing the measurement results, it can be found that a relatively small or relatively large learning rate will reduce the accuracy to almost 0, that is, the trained model cannot be used.
At the same time, it can be found that if the learning rate is smaller or larger than a certain value, the accuracy will be greatly reduced, that is, the effect of the trained model will be worse.

5.4 Summary

According to the above three sets of tables, I came to the conclusion: Compared with Optimizer and Activation Function, the value of Learning Rate has a greater impact on the accuracy of model training results. This shows that the value of the learning rate should be more stringent , and a better learning rate should be obtained by continuous testing to prevent it from having a greater impact on the model.