Car Detection final report

qifengc2@uci.deu/ID:57004415

May 1, 2022

Contents

1 Datasets **2**

2 Introduction to DNN models **2**
2.1 NormalNet 2
2.2 ResNet 3
2.3 MobileNetV1 3
2.4 QfcNet 4
2.4.1 Reason 4
2.4.2 Realization 4

3 Results **5**

4 Conclusion **6**
4.1 Conclusion of the effct of factors 6
4.2 Conclusion of the residual block and depthwise separable convolution 6
4.3 Conclusion of the comparison of the four models 6

1 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.

2 Introduction to DNN models

2.1 NormalNet

NormalNet model contains convolutional layers and fully connected layers.
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 fully connected layer is one layer, the size of the kernel is 8x8, and the output is 1000.
The reason I make NormalNet by myself rather than just using Alexnet is that I want to turn the DNN theory that I just learned into practice. The process of making DNN can give me an better understanding of every function unit in

DNN, and encourage me to learn the background knowledge. This will help me in my future project.

2.2 ResNet

The resnet model contains residual block and fully connected layers.
The composition of the residual block is a shortcut and a regular network layer are superimposed and passed through the activation function.
The shortcut contains a convolutional layer and a batch normalization. The regular network layer consists of two parts: the first concludes the onvolutional layer, batch normalization and activation function. The second is the convolutional layer and batch normalization.
I choose this residual block it's because it can make the regular layer output size and the shortcut layer output size matched even they are not equal to the input size.

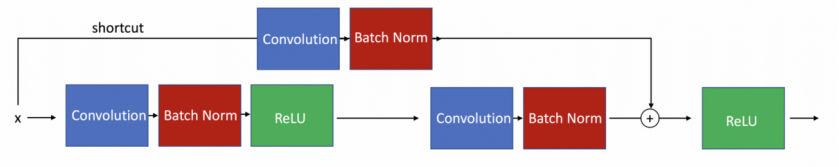

Figure 2: residual block

2.3 MobileNetV1

The model contains depthwise separable convolution layer, adaptive average pool and fully connected layers.
Depthwise Separable Convolution concludes Depthwise Convolution and 1x1 Convolution. They both have convolutional layer, batch normalization and activation function.
The size of the Depthwise Convolution kernel is 3x3, the default stride comes from input, and the padding is 2. The size of the1x1 Convolution is 1x1, the default stride is 1, and the padding is 0.
The Adaptive Average Pool, the size of the kernel is 1x1, and the output is 1024.

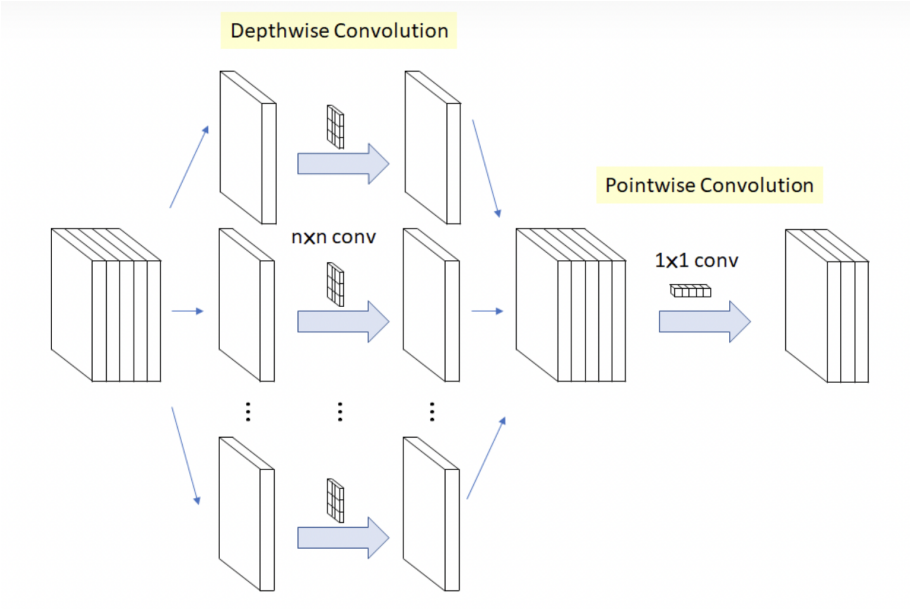

Figure 3: depthwise separabel convolution

2.4 QfcNet

2.4.1 Reason

When the depth of the neural network increases, we can use residual block to solve the problem of network degration. We can also use the depthwise separable convolution to decrease the parameter size.
However, if we use these two technologies seperately, for instance, we may face the problem of large parameter size when we only use a lot of residual blocks. Or network degration may occur if we only use many depthwise separable convolutions.
So in the last project, I make my own DNN: QfcNet, by usining the residual block and depthwise seperable convolution. I try to combine the adventage of these two technologies together.

2.4.2 Realization

The DNN I designed consists many residual blocks composed of depthwise separable convolutions.
In each residual block, I replace the original convolution by the depthwise seperable convolution. In other word, I insert the residual blocks in the MobileNetV1. At the same time, I use two different kinds residual blocks. One is add convolution layer and batch norm in the shortcut. This will help us fix the problem that the output size of depthwise separable convolution is not equal to the input size.
We don't need to add another convolution after depthwise separable convolution to change the output but just change the size of X by the shortcut.
This parallel structure can reduce network depth.

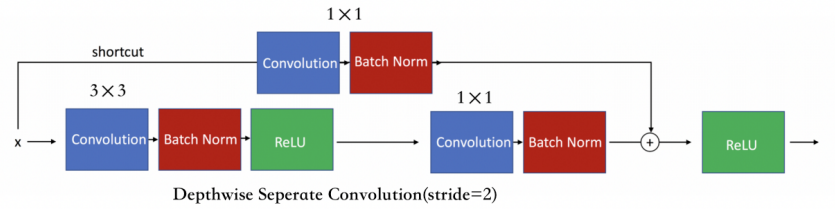

Figure 4: residual block 1

The other is add nothing in the shortcut. When the output size of depthwise separable convolution is equal to the input size, we can just pass input through the shortcut. This can reduce parameter size while adding the residual block.

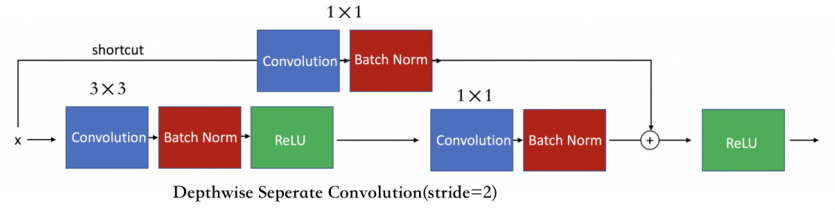

Figure 5: residual block 2

3 Results

In the previous project, in order to find the effect of the factors(LR, Activation Function, Epoch, etc) and the effect of residual block and depthwise separable convolution, I take the approach of controlling variables: change one paramter and fix other parameters.
And I measure Inference Time, Training Time, and Accuracy to evaluate those factors.
For my data set, the most suitable Activation Function is ReLU, the best Optimizer is Adam and a relatively good Learning Rate is 3e-4.

Model Name	LR	Activation Function	Epoch	Optimizer
NormalNet	3e-4	Relu	100	Adam
ResNet	3e-4	Relu	100	Adam
MobileNetV1	3e-4	Relu	100	Adam
QfcNet	3e-4	Relu	100	Adam

Table 1: Parameters for training

So in the last project, in order to compare the difference four models, I use those

parameters in the four models and get the results:

Model Name	inference Time	Training Time	accuray	Model Size
NormalNet	204.84 s	8247.03 s	96.18%	117.62 MB
ResNet	761.37 s	5194.21 s	97.86%	74.91 MB
MobileNetV1	198.25 s	5865.50 s	97.69%	33.65 MB
QfcNet	124.80 s	2777.39 s	98.94%	53.80 MB

Table 2: Evaluation parameters for the model

4 Conclusion

4.1 Conclusion of the effct of factors

1. Different activation functions have different degree of influence on training time and inference time.
2. 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.
3. Each model corresponds to a unique learning rate, and a smaller or larger learning rate will greatly reduce the accuracy. Increasing epochs may make the accuracy larger, however, this increasing rate will gradually decrease with the increase of epoch, and even reduce the accuracy. We should choose the most appropriate learning rate carefully.

4.2 Conclusion of the residual block and depthwise separable convolution

1. In a DNN, residual block can solves the problem of network degradation.
2. Depthwise separable convolution can reduce the size of network parameters effectively.

4.3 Conclusion of the comparison of the four models

1. QfcNet obviously have a short inference time and training time.
2. Compared with ResNet of the same depth, QfcNet's parameter size is smaller.
3. Compared with MobileNetV1 of the same depth, QfcNet significantly improves the accuracy, Although the introduction of residual blocks increases the parameter size, the cost is acceptable.