

## Project Two Report

Siyuan Wu

Submission name on Kaggle: Siyuan(Leon) Wu

Best accuracy: 70.9%

### Part 1

#### Layer Structure Table

Layer No.	Layer Type	Kernel size	Input Output Dimension	Input Output Channels
1	Conv2d	3	32 32	3 64
2	Relu	-	32 32	-
3	Batchnorm	-	32 32	-
4	Conv2d	3	32 32	64 64
5	Relu	-	32 32	-
6	Batchnorm	-	32 32	-
7	Maxpool2d	2	32 16	-
8	Conv2d	3	16 16	64 128
9	Relu	-	16 16	-
10	Batchnorm	-	16 16	-
11	Conv2d	3	16 16	128 128
12	Relu	-	16 16	-
13	Batchnorm	-	16 16	-
14	Maxpool2d	2	16 8	-
15	Conv2d	3	8 8	128 256
16	Relu	-	8 8	-
17	Batchnorm	-	8 8	-
18	Conv2d	3	8 8	256 512
19	Relu	-	8 8	-
20	Batchnorm	-	8 8	-
21	Linear	-	32768 2048	-
22	Batchnorm1D	-	2048 2048	-
23	Relu	-	2048 2048	-
24	Linear	-	2048 2048	-
25	Relu	-	2048 2048	-
26	Linear	-	2048 100	-

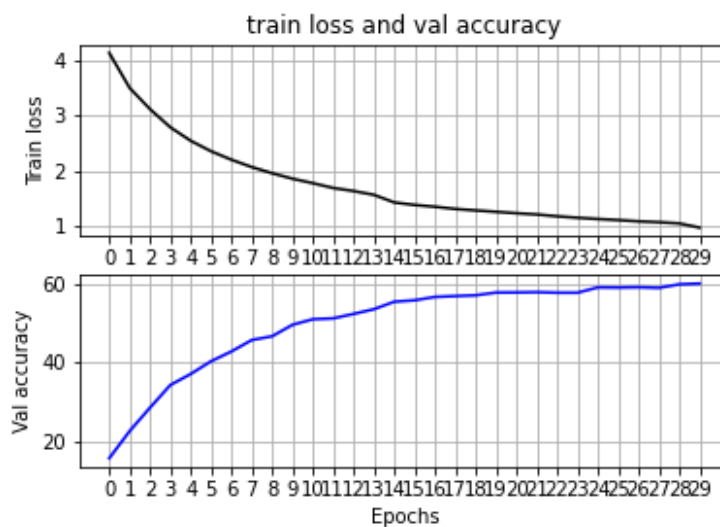
#### Structure Explain

Part of the structure is referenced from the VGG16 network model, but many modifications have been made to make the net tailored for picture size 32-by-32. First,

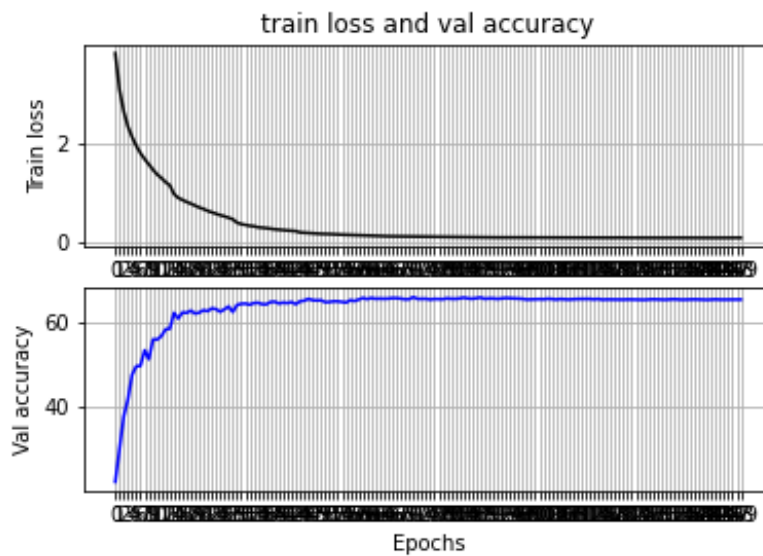
the number of weight layers is reduced to 9 from VGG16's 16 weight layers. This modification prevents the net from overfitting and reduces the training speed. Then, a 2d batch normalizing layer is added after the RELU layer of each convolution layer. Pooling layers are also reduced to 2 to avoid too much information loss. Moreover, the fully connected layer decrease from VGG's 4 to 3, and the 1d batch normalizing layer is also used after a fully connected layer.

### Training and validation accuracy plot

Stage One: During this stage, I was trying to optimize the structure and parameters of the network, therefore, the number of epochs is limited to 30.



Stage Two: After the structure and parameters of the network are optimized, I try to use the data that augmented in different ways to fortify the net, and for each method, there will be 100 epochs. There were 400 epochs in total.



## Ablation Study

Adding data normalization: accuracy increase from 23 % to 27%

Adding three more convolution layers and change all convolution layers' kernel size to 3: accuracy increase from 27% to 33%

Increase filter size for all convolution layer, and the maximum filter in convolution layers become 512: accuracy increase from 33% to 37%

Increase filter size for all convolution layer once again, and maximum filter increase to 1024. Batch normalizing layer added after each convolution layer: accuracy increase from 37% to 49%

One additional linear layer with 1d batch normalizing was added: accuracy increase from 49% to 57%

Adapting VGG16 structure, with two convolution layers followed by a pooling layer. 6 convolution layers, 3 pooling layers, and 3 linear layers in total is used: accuracy increase from 57% to 62%

Removing the last pooling layer: accuracy increase from 62% to 64%

Changing the last convolution layer's filter size to 512: accuracy increase from 64% to 66%

Training the net with different data augmentation methods. The net will be trained with all four kind of transformed data(random vertical flip, random horizon flip, random rotation, grayscale ) one by one for 100 epochs each: accuracy increase from 66% to 71%

## Part 2

### Model and Hyperparameters

For the second part, I used resnet50 as my pretrained net and my data transformation include random resize crop of 256, random horizon flip and normalization. The net trains with learning rate of 0.0005 and a batch size of 8. First the net is train only with the last layer. The training accuracy and test accuracy with 50 epochs are below.

#### Training accuracy for RESNET\_LAST\_ONLY = True

```
TRAINING Epoch 50/50 Loss 0.2405 Accuracy 0.6440  
Finished Training  
-----
```

#### Test accuracy for RESNET\_LAST\_ONLY = True

```
[29] test(model, criterion)  
  
👉 Test Loss: 0.2888 Test Accuracy 0.4207
```

The second net is a fully trained net. The training accuracy and test accuracy with 50 epochs are below.

#### Training accuracy for RESNET\_LAST\_ONLY = False

```
TRAINING Epoch 50/50 Loss 0.0573 Accuracy 0.9037  
Finished Training  
-----
```

#### Test accuracy for RESNET\_LAST\_ONLY = False

```
[25] test(model, criterion)  
  
👉 Test Loss: 0.1851 Test Accuracy 0.6423
```

## Reference

- [1] VGG16 - Convolutional Network for Classification and Detection. (2018, November 21). Retrieved October 11, 2020, from <https://neurohive.io/en/popular-networks/vgg16/>