NYCU Introduction to Machine Learning, final_project_report

110612117 張仲瑜

Part. 1, Environment Settings

1. Python version: 3.11.5 in anaconda
2. Framework: Pytorch with FastAi as packages
3. Hardware: i7-13700 + RTX3060 12G
4. The model downloaded from weight.txt must be place at 110612117_final\training\model\model.pkl

Part. 2, Implementation

Since I’ ve tried multiple models to get better result, the architecture of them will be introduced down below.

Data Preprocessing:

Scale the width and height of each training data to make width = 256 pixels. Crop the width an height of each training data to 224 * 224 pixel when loading them into a batch.

Training Strategy Overview:

A. Pretrained the model with imageNet1K_V1 (Used in image classification)
B. Use batch Normalization to stable the training process, increase the convergence speed and lower the dependency on the initial weights
C. Set weight decayed to 0.3 to punish higher weight on the purpose of preventing overfitting.
D. Use Early Stopping (abort when validation loss didn’ t get lower 5 epochs continuously) to prevent overfitting.
E. Only last few layers are trained in starting epochs (freeze epochs) to prevent the pre-trained weights are lost in the beginning.

1. VGG19_bn

Architecture: 16 convolution layers and 3 fully connected layers are used

Con1_1~2: 64 3*3 kernels with ReLU as activation function

BatchNorm2d(64)

MaxPooling 2*2 with stride = 2

Con2_1~2: 128 3*3 kernels with ReLU

BatchNorm2d(128)

MaxPooling 2*2 with stride = 2

Con3_1~4: 256 3*3 kernels with ReLU

BatchNorm2d(256)

MaxPooling 2*2 with stride = 2

Con4_1~4: 512 3*3 kernels with ReLU

BatchNorm2d(512)

MaxPooling 2*2 with stride = 2

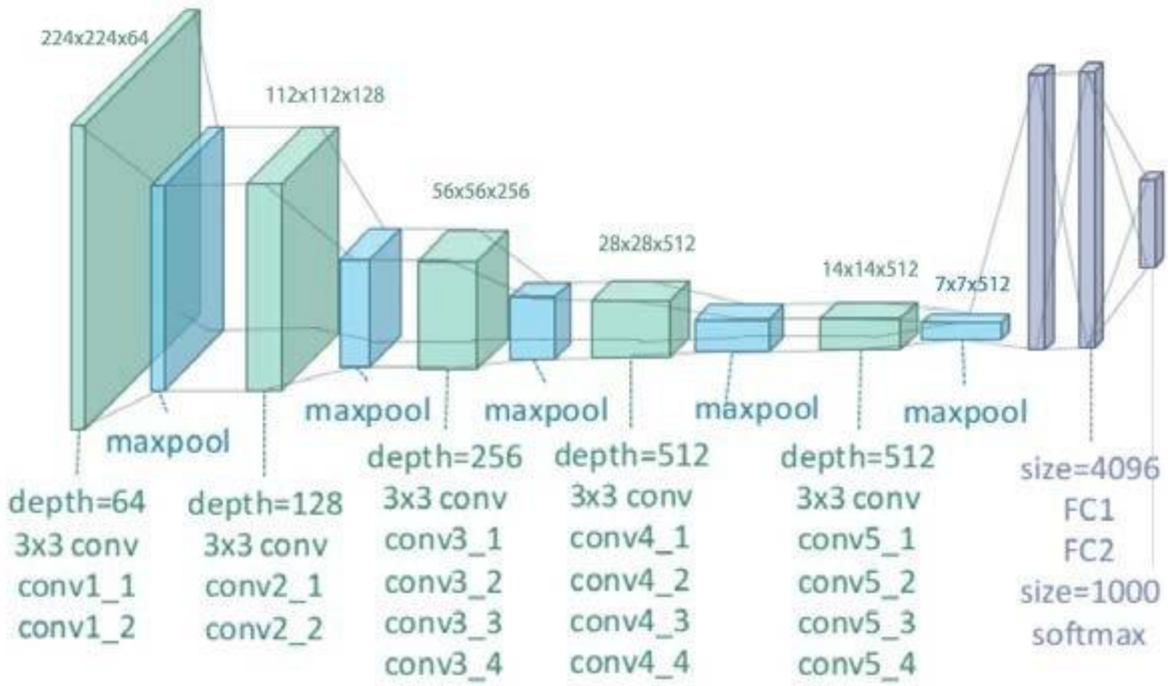
Con5_1~4: 512 3*3 kernels with ReLU

BatchNorm2d(512)

MaxPooling 2*2 with stride = 2

FC1~2: 4096 neurons with ReLU

FC2: 200 neurons to our classification task with softmax



VGG19 architecture from Zheng, Yufeng & Yang, Clifford & Merkulov, Aleksey. (2018). Breast cancer screening using convolutional neural network and follow-up digital mammography. 4. 10.1117/12.2304564.W

Hyperparameters:

Epoch = 10 freezing + 5 unfreezing

Learning rate = 0.005

Batch size = 64

Weighted decayed = 0.3

Early Stopped patience = 5

2. DenseNet121

Architecture: 4 dense blocks are stacked in total, and 6,12,24,24 dense layers are included respectively. Batch Normalization, ReLU and Convolution layer are stacked in dense layers.

A dense layer example:

DenseLayer(

(norm1): BatchNorm2d(64)

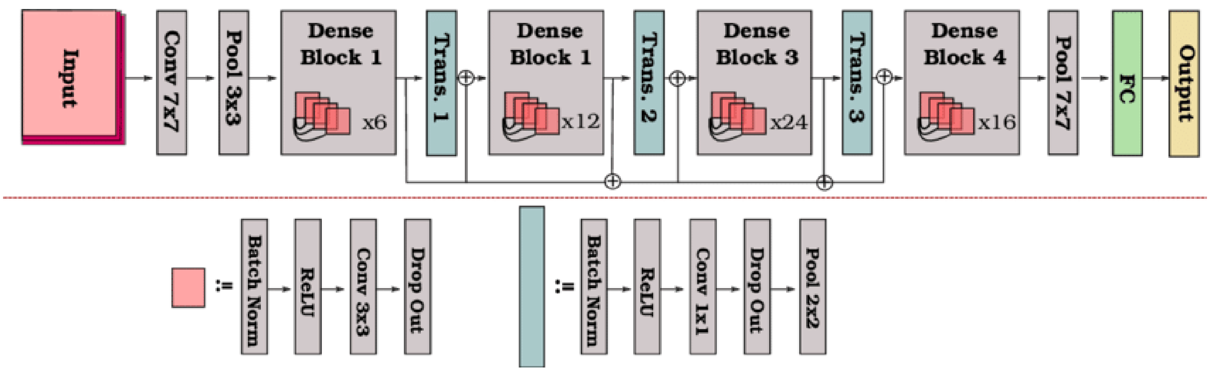
(relu1): ReLU()

(conv1): Conv2d(64, 128, kernel_size=(1, 1), stride=(1, 1), bias=False)

(norm2): BatchNorm2d(128)

(relu2): ReLU()

(conv2): Conv2d(128, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False))



DenseNet121 architecture from Radwan, Noha. (2019). Leveraging Sparse and Dense Features for Reliable State Estimation in Urban Environments. 10.6094/UNIFR/149856.

Hyperparameters:

Epoch = 10 freezing+ 5 unfreezing

Learning rate = dynamic as figure in experiment result

Batch size = 64

Weighted decayed = 0.3

Early Stopped patience = 5

3. ResNet50

Architecture:

Initialization convolution layer:

Conv 64 7 * 7 kernels

Batch normalization

ReLU activation

MaxPooling 3*3 with stride = 2

4 stages with 3,4,6,3 residual blocks

Residual blocks in Stage 1: 3 Conv layers with input = 64

Residual blocks in Stage2: 3 Conv layers with input = 256

Residual blocks in Stage3: 3 Conv layers with input = 512

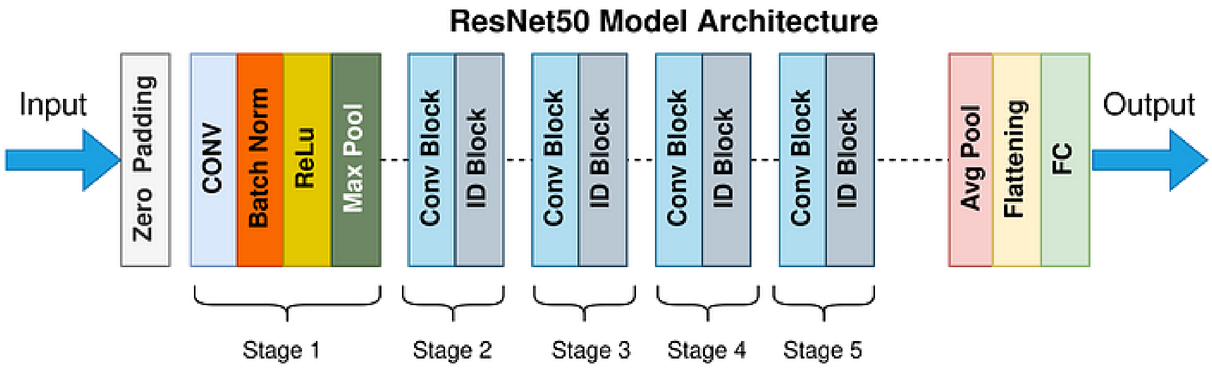
Residual blocks in Stage4: 3 Conv layers with input = 1024

Global Average Pooling

Apply global average pooling to the output of the last residual block reducing spatial dimensions to (1, 1).

Fully Connected Layer (Linear Layer):

Neurons = 200 in this case



resNet50 architecture from Gorlapraveen123

Hyperparameters:

Epoch = 10 unfreezing + 10 unfreezing

Learning rate = dynamic

Batch size = 64

Weighted decayed = 0.3

Early Stopped patience = 5

Part. 3, Experiment Result

1. VGG19_bn

   I. Evaluation matrix:

   Instead of showing the whole confusion matrix which is hard to show on the screen with 200 class classification, I list top 5 misclassified relationship between two classes that are easier read.

   Actual: 060.Glaucous_winged_Gull, Predicted: 066.Western_Gull

   Occurrences: 6, Confusion Value: 6

   Actual: 112.Great_Grey_Shrike, Predicted: 111.Loggerhead_Shrike,

   Occurrences: 6, Confusion Value: 6

   Actual: 143.Caspian_Tern, Predicted: 145.Elegant_Tern,

   Occurrences: 6, Confusion Value: 6

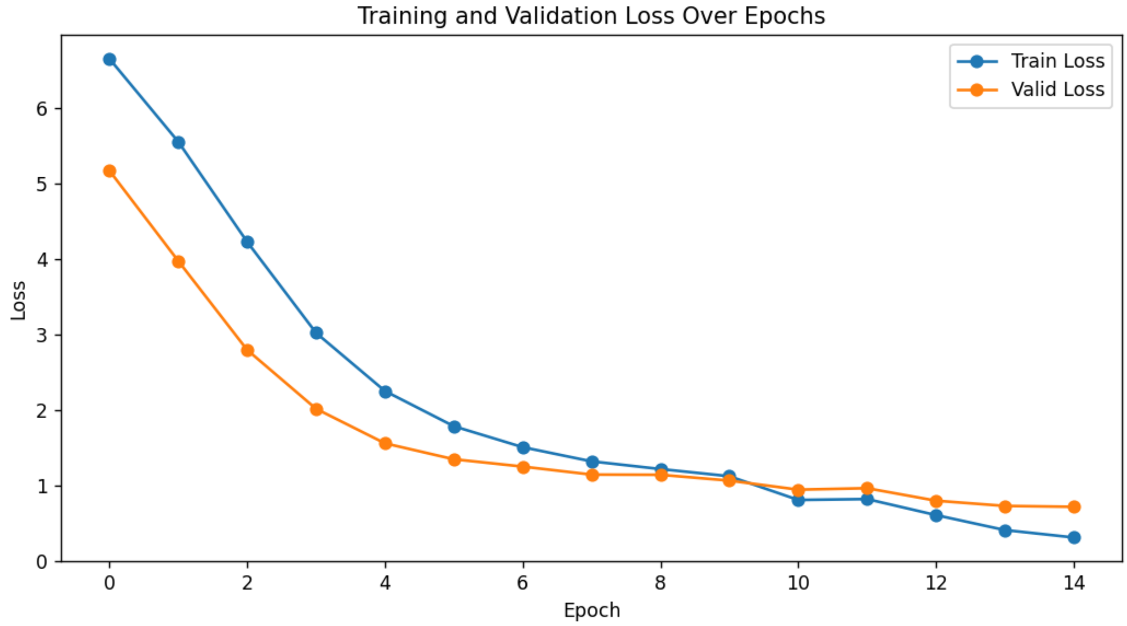
   Actual: 059.California_Gull, Predicted: 066.Western_Gull,

   Occurrences: 4, Confusion Value: 4

   Actual: 102.Western_Wood_Pewee, Predicted: 103.Sayornis,

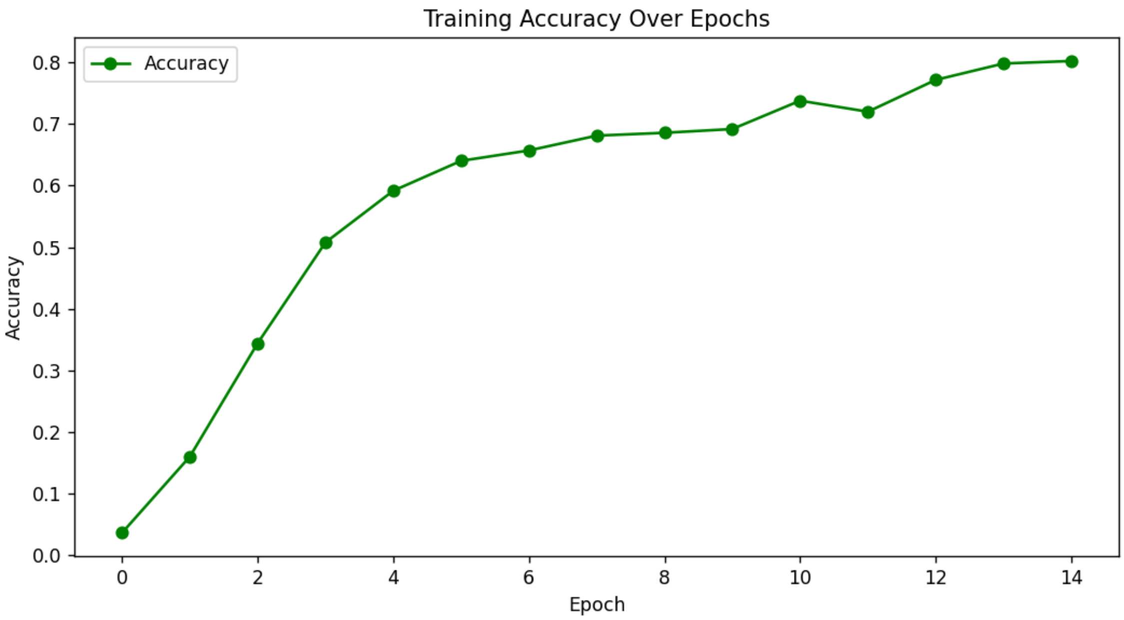
   Occurrences: 4, Confusion Value: 4

II. Learning curve and the learning rate



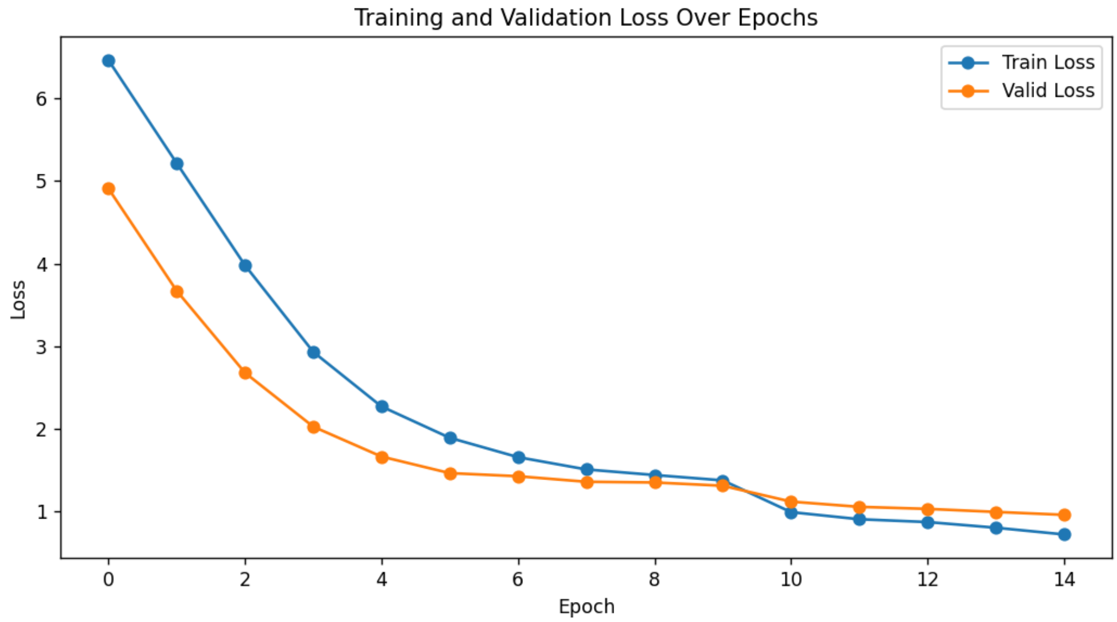
Training Loss at the end of the training: 0.3154

Valid Loss at the end of the training: 0.7215



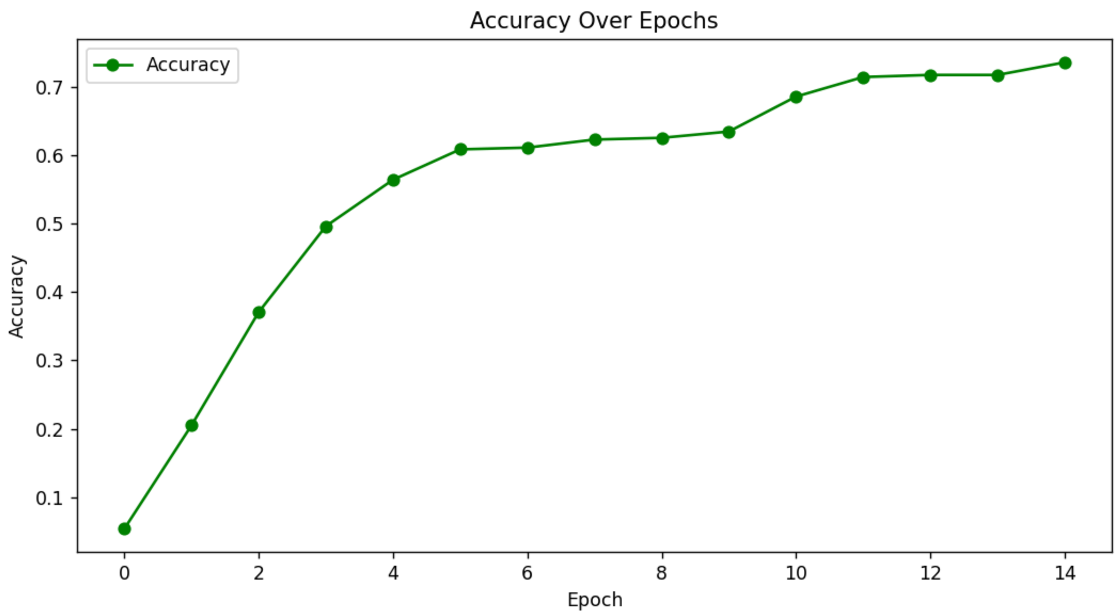
Accuracy at the end of the training: 0.8202

III. Ablation study: discard all the batch normalization part



Training Loss at the end of the training: 0.7233

Valid Loss at the end of the training: 0.9604



Accuracy at the end of the training: 0.7358

Observation: Batch Normalization had an effect on decreasing the loss and increasing the accuracy in the same number of epochs.

2. DenseNet121

I. Evaluation matrix (Top 5 misclassified):

Actual: 060.Glaucous_winged_Gull, Predicted: 066.Western_Gull, Occurrences: 6, Confusion Value: 6

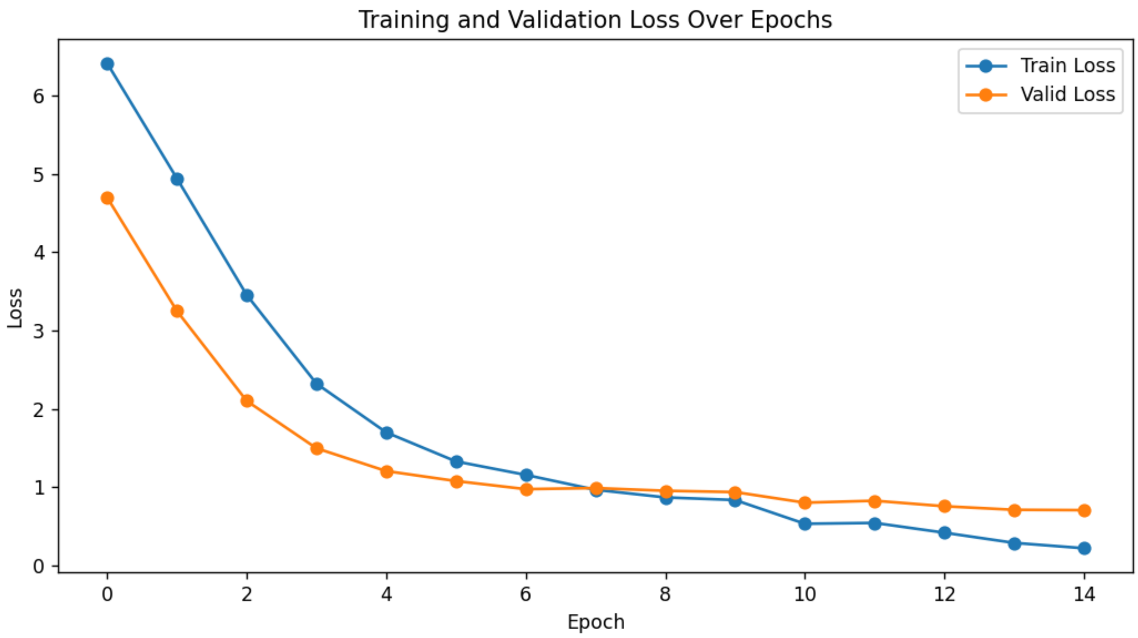
Actual: 112.Great_Grey_Shrike, Predicted: 111.Loggerhead_Shrike, Occurrences: 6, Confusion Value: 6

Actual: 143.Caspian_Tern, Predicted: 145.Elegant_Tern, Occurrences: 6, Confusion Value: 6

Actual: 059.California_Gull, Predicted: 066.Western_Gull, Occurrences: 4, Confusion Value: 4

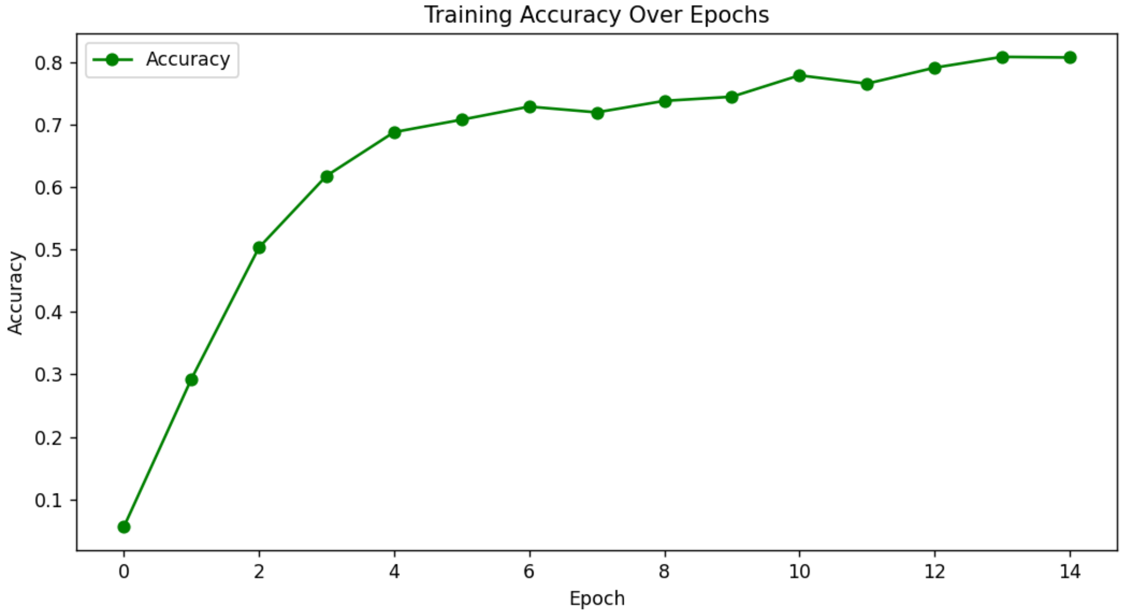
Actual: 102.Western_Wood_Pewee, Predicted: 103.Sayornis, Occurrences: 4, Confusion Value: 4

II. Learning curve and the learning rate



Training Loss at the end of the training: 0.2198

Valid Loss at the end of the training: 0.7064



Accuracy at the end of the training: 0.8079

3. ResNet50

I. Evaluation matrix (Top 5 misclassified):

Actual: 102.Western_Wood_Pewee, Predicted: 103.Sayornis,

Occurrences: 6, Confusion Value: 6

Actual: 031.Black_billed_Cuckoo, Predicted: 033.Yellow_billed_Cuckoo,

Occurrences: 5, Confusion Value: 5

Actual: 179.Tennessee_Warbler,

Predicted: 173.Orange_crowned_Warbler,

Occurrences: 5, Confusion Value: 5

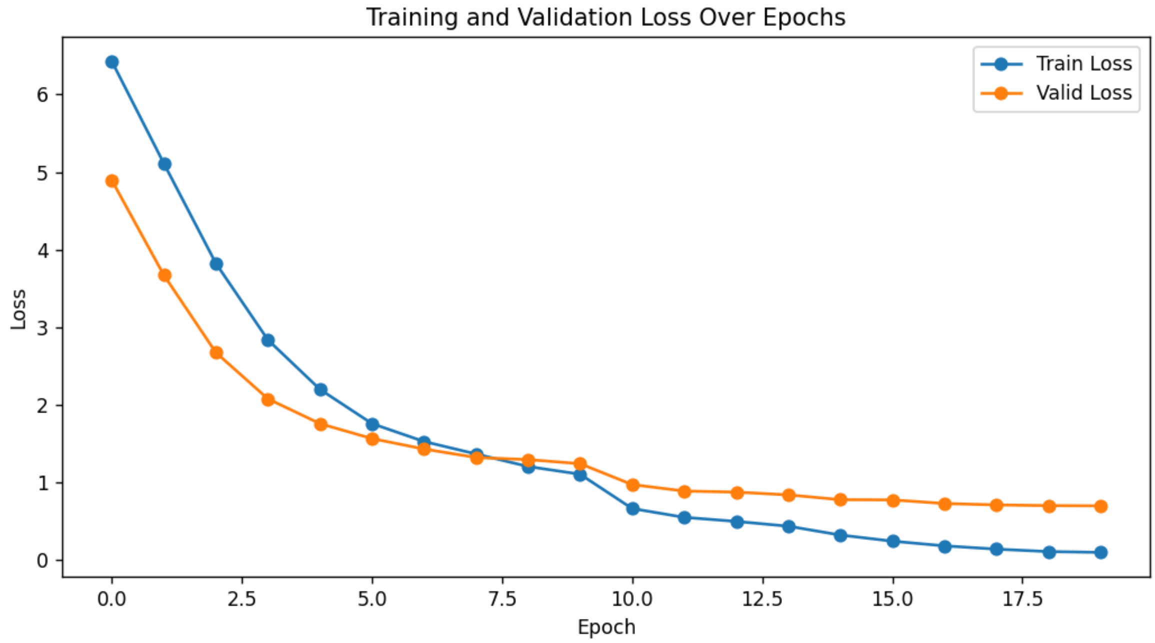
Actual: 022.Chuck_will_Widow, Predicted: 105.Whip_poor_Will,

Occurrences: 4, Confusion Value: 4

Actual: 060.Glaucous_winged_Gull, Predicted: 059.California_Gull,

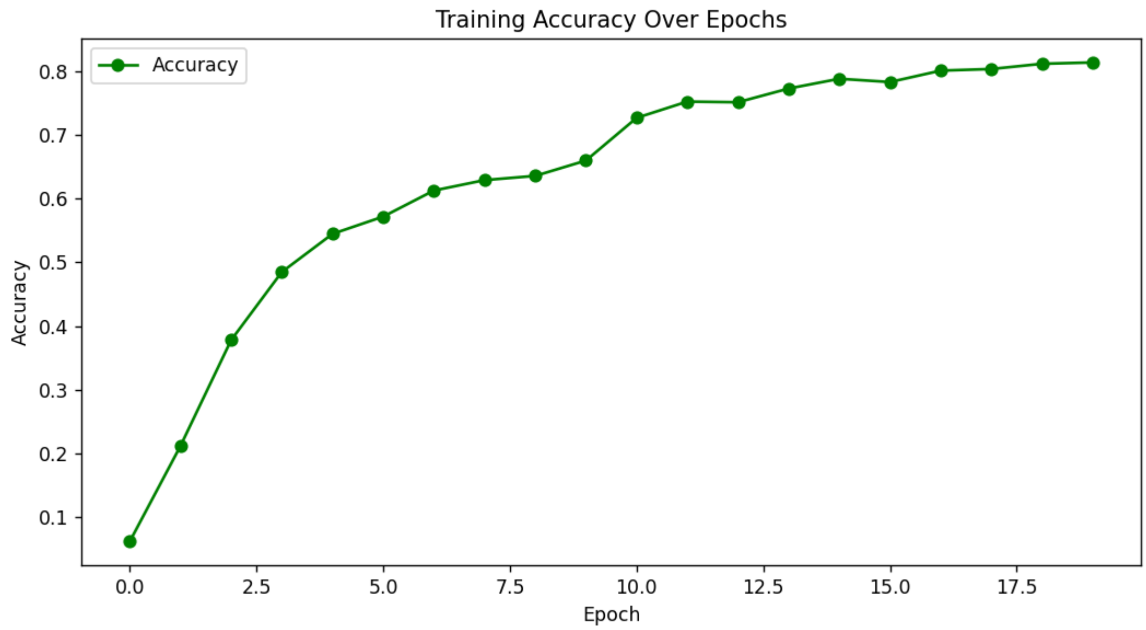
Occurrences: 4, Confusion Value: 4

II. Learning curve and the learning rate



Training Loss at the end of the training: 0.1010

Valid Loss at the end of the training: 0.7000



Accuracy at the end of the training: 0.8135

III. Ablation study: with the dropout layer

Training Loss at the end of the training: 0.3414

Valid Loss at the end of the training: 0.7776

Accuracy at the end of the training: 0.7833

Observation: Dropout did increase the performance of the model.

Part 4. Conclusion

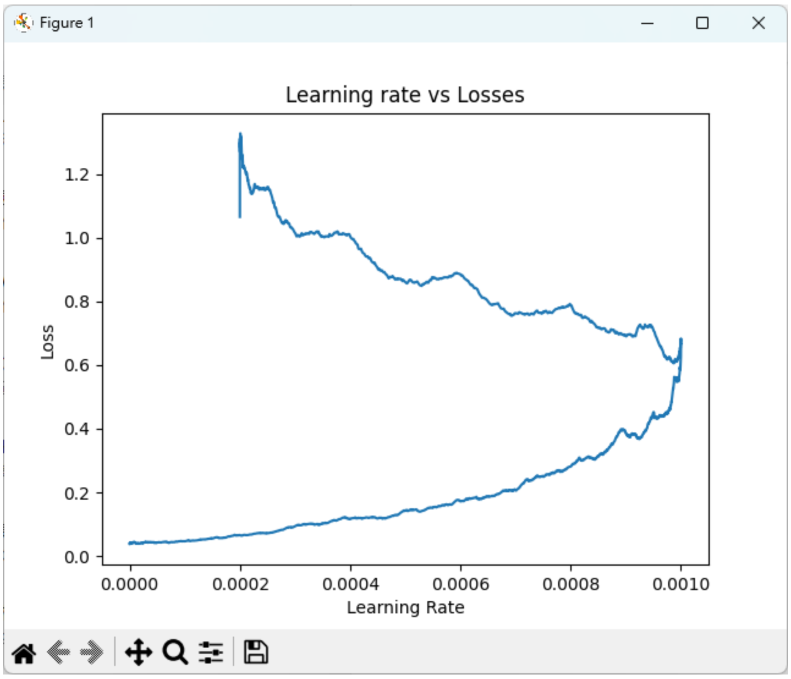
1. In most cases the validation loss is larger than the training loss in the last few epochs, which mean overfitting did happened. I tried several methods to decrease its effect, it turns out that batch normalization has great effect on it.
2. ResNet50 did a great job within limited epochs and time comparing to the other model.
3. When using a model pretrained on ImageNet1K, freezing most of the layers at first has a great performance since it prevented the important characteristic from losing at the very beginning.

Part 5. Thoughts after the project

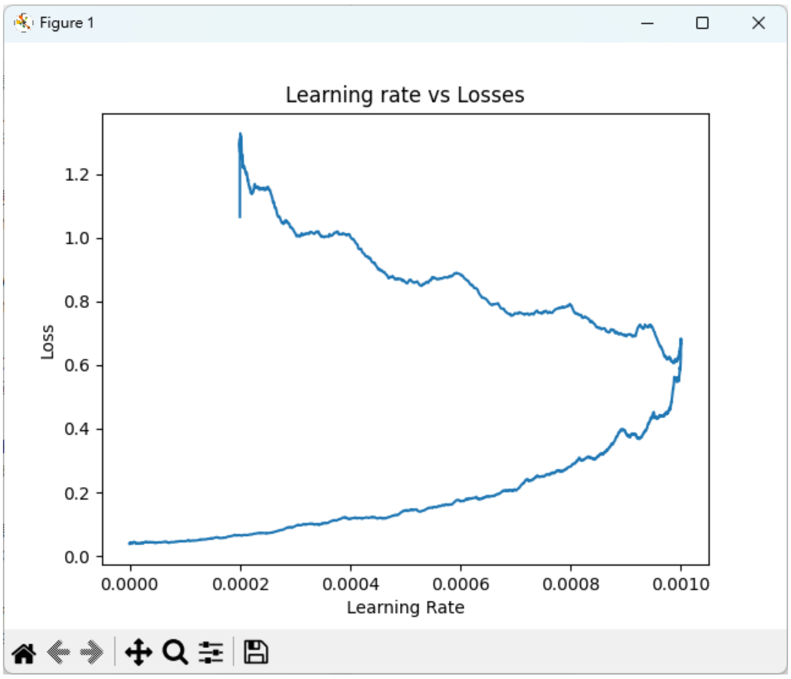
It was both entertaining and tiring to learn the concept of the models while get out hand dirty at the same time. It is a pity that I can not implement Vit and Swin-Vit due to the lack of time to boost the performance further, though. It is still helpful for us to learn how to adjust the hyperparameter and the model structure during the project.

Part6. Other figures

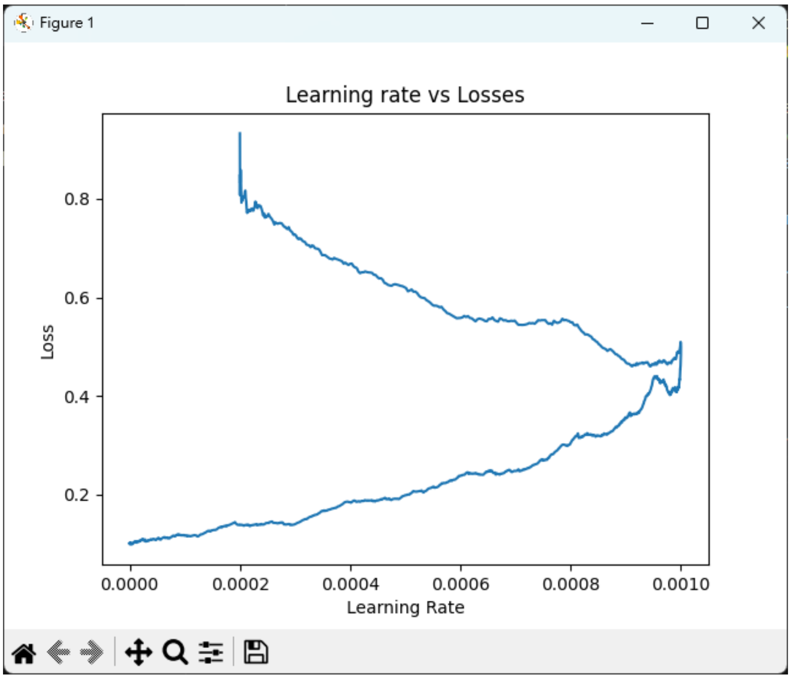
A. dynamic learning rate



VGG19_bn



DenseNet



ResNet50

Observation:

We can find that the slope the learning rate changes from small to big when the losses are larger, and goes the other way once the losses stop decreasing to avoid divergence.

B. Challenge Game (vs resNet50)

1. Given 3 pictures of 102.Western_Wood_Pewee, then select them in the following 4 pictures.

Answer: RT, LB, the others are 103.Sayornis, these two types of birds have been misclassified by ResNet50 6 times!