Individual Studies in Medical Image Analysis

Image Classification with Covid-19

Reporter : Cheng-You Gao

Advisor : Dr. Wei-Yang Lin

Name:	ID number:
Jia-Cheng, Syu	611410013
Yu-Ting, Liu	611410083
Wei-Chieh, Tseng	611410146
Cheng-You, Gao	611410166

Outline

I. INTRODUCTION
II. PROPOSED METHOD
III. IMPLEMENTATION
IV. EXPERIMENTAL RESULTS
V. DISCUSSION AND FUTURE RESEARCH
VI. CONCLUTION

Outline

I. INTRODUCTION
II. PROPOSED METHOD
III. IMPLEMENTATION
IV. EXPERIMENTAL RESULTS
V. DISCUSSION AND FUTURE RESEARCH
VI. CONCLUTION

I. INTRODUCTION

1. Dataset :

6,893 normal images and 7,593 COVID-19 images from Large COVID-19 CT scan slice dataset. Resizing the original image from 512x512 to 224x224.

2. Device :

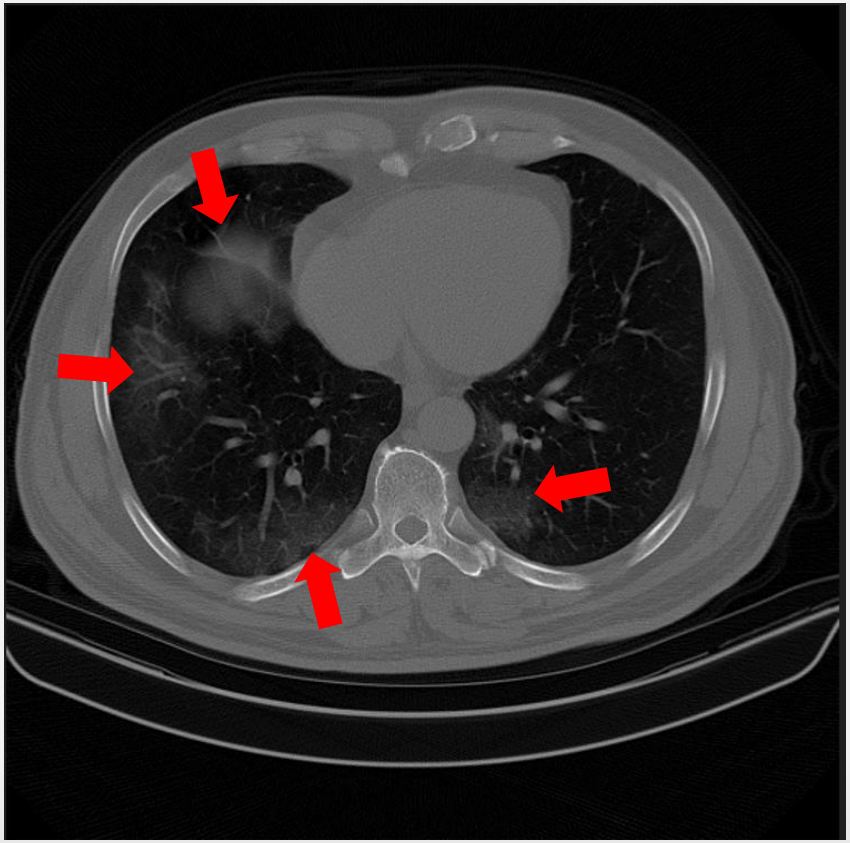
Software : Python, PyTorch

Hardware : RTX2080 Ti

https://www.kaggle.com/datasets/maedemaftouni/large-covid19-ct-slice-dataset

I. INTRODUCTION

Normal image



Covid-19 image

Normal image : 6_Rahimzadeh_normal2_patient205_SR_4_IM00022.png
Covid-19 image : 6_Rahimzadeh_137covid_patient63_SR_4_IM00025.png

Outline

I. INTRODUCTION
II. PROPOSED METHOD
III. IMPLEMENTATION
IV. EXPERIMENTAL RESULTS
V. DISCUSSION AND FUTURE RESEARCH
VI. CONCLUTION

II. PROPOSED METHOD

1.1 Model Architecture

ResNet

VGG

DenseNet

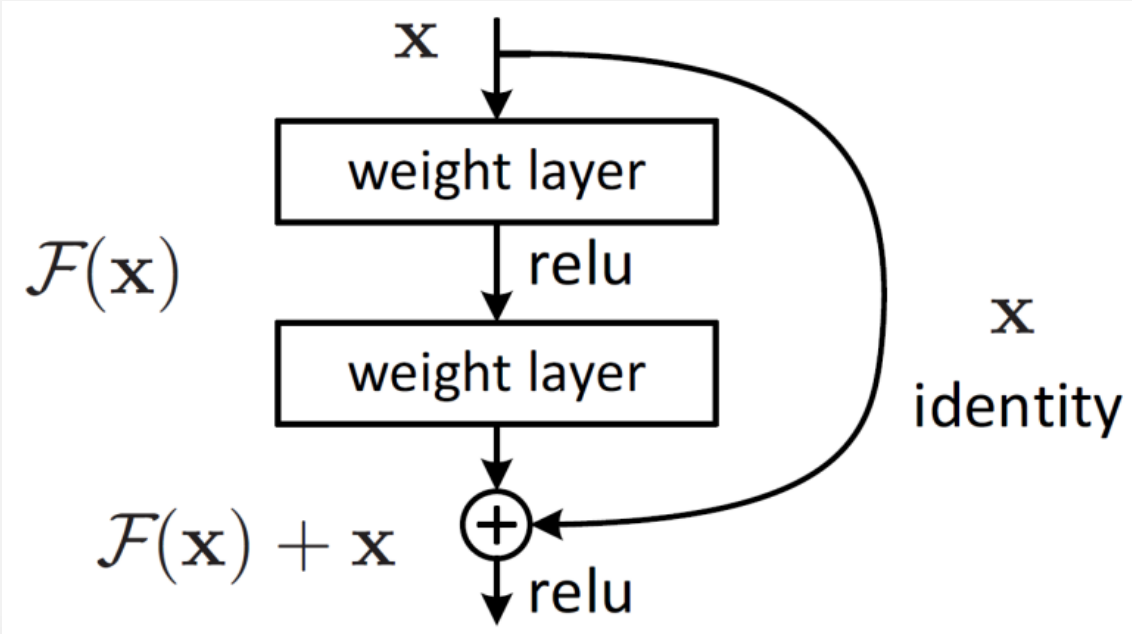
II. PROPOSED METHOD

1.1 Model Architecture

ResNet

VGG

DenseNet



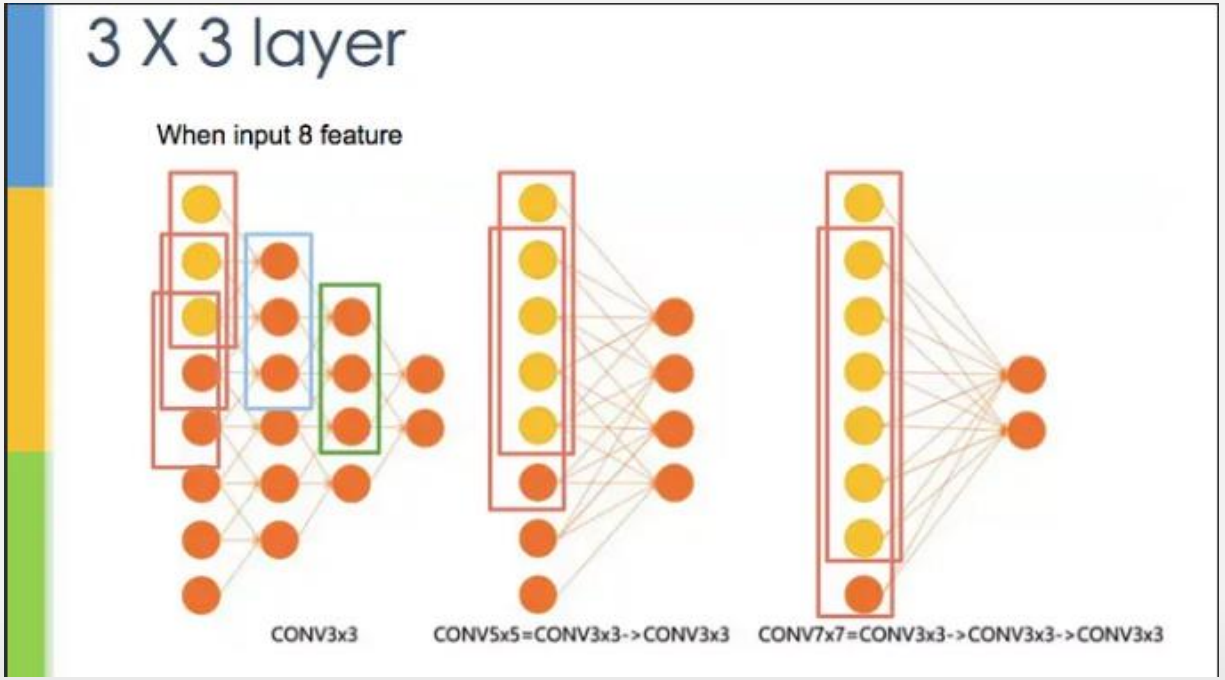
II. PROPOSED METHOD

1.1 Model Architecture

ResNet

VGG

DenseNet



https://danjtchen.medium.com/vgg-%E6%B7%B1%E5%BA%A6%E5%AD%B8%E7%BF%92-%E5%8E%9F%E7%90%86-d31d0aa13d88

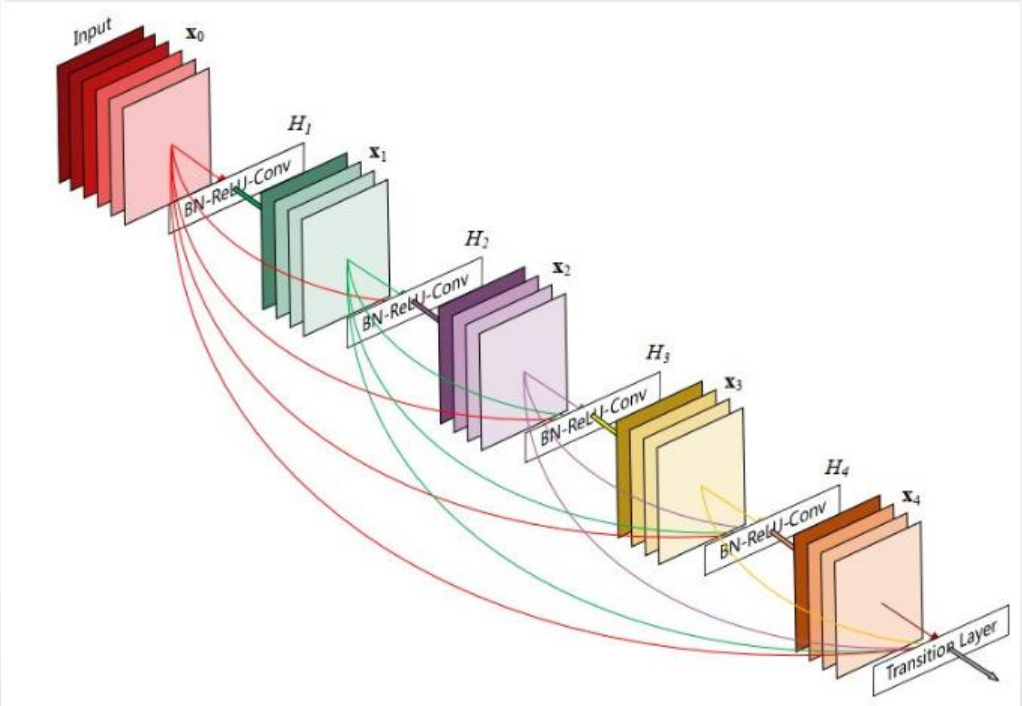
II. PROPOSED METHOD

1.1 Model Architecture

ResNet

VGG

DenseNet



https://ithelp.ithome.com.tw/articles/10265328

II. PROPOSED METHOD

1.2 Number of parameters

ResNet18 : 11 million

VGG11 : 133 million

DenseNet121 : 7 million

2. Hyperparameter

Leaning rate

Epoch

Batch size

Data splitting(training/validation/testing)

3. Preprocessing

Data augmentation

Outline

I. INTRODUCTION
II. PROPOSED METHOD
III. IMPLEMENTATION
IV. EXPERIMENTAL RESULTS
V. DISCUSSION AND FUTURE RESEARCH
VI. CONCLUTION

III. IMPLEMENTATION

1. Model Comparation

Use hyperparameters(learning rate, epoch) to compare which is suitable model.

ResNet18 vs. ResNet
ResNet18 vs. VGG
ResNet18 vs. DenseNet

III. IMPLEMENTATION

2. Hyperparameter setting

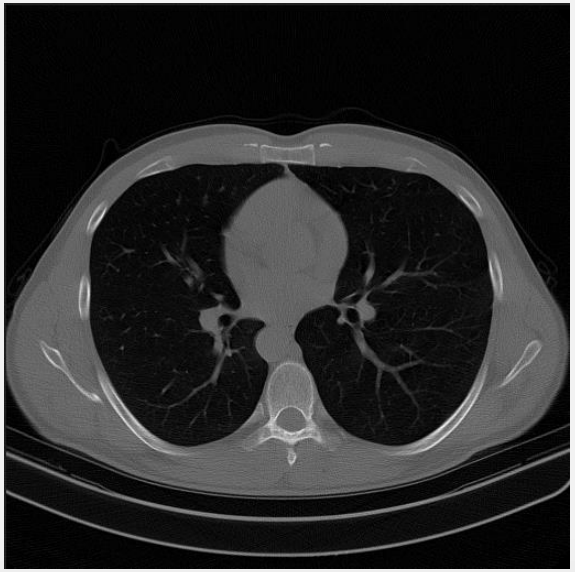
Hyperparameter	Value
Learning rate	0.001
Epoch	30
Batch size	9, 16
Data splitting	(6:2:2), (7:1:2)

III. IMPLEMENTATION

3. Preprocessing

We flip images to horizontal and vertical , increasing data variability.

Original image



Horizontal image

Vertical image

Outline

I. INTRODUCTION
II. PROPOSED METHOD
III. IMPLEMENTATION
IV. EXPERIMENTAL RESULTS
V. DISCUSSION AND FUTURE RESEARCH
VI. CONCLUTION

IV. EXPERIMENTAL RESULTS

1.1 First stage

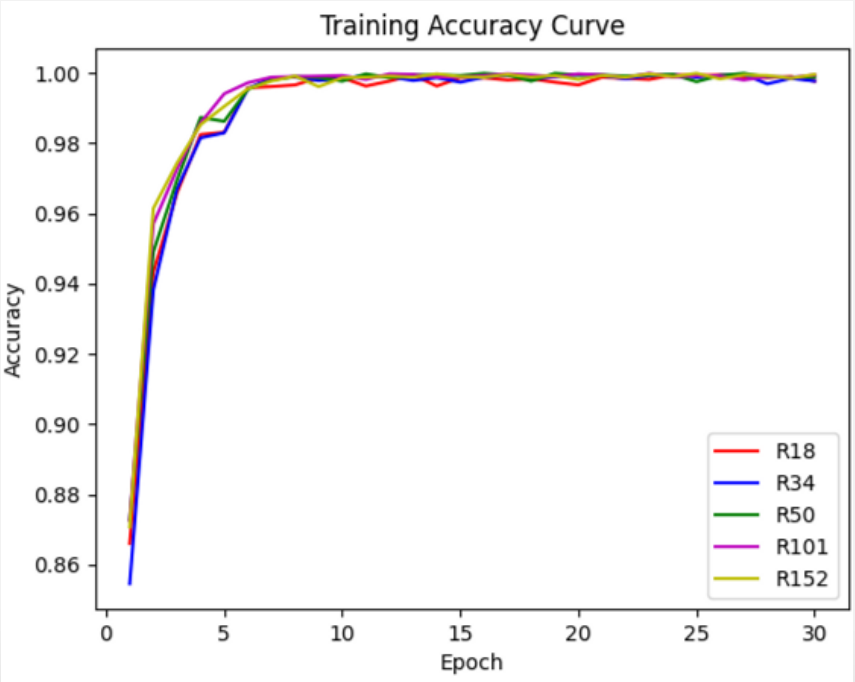
Compare ResNet with different layer

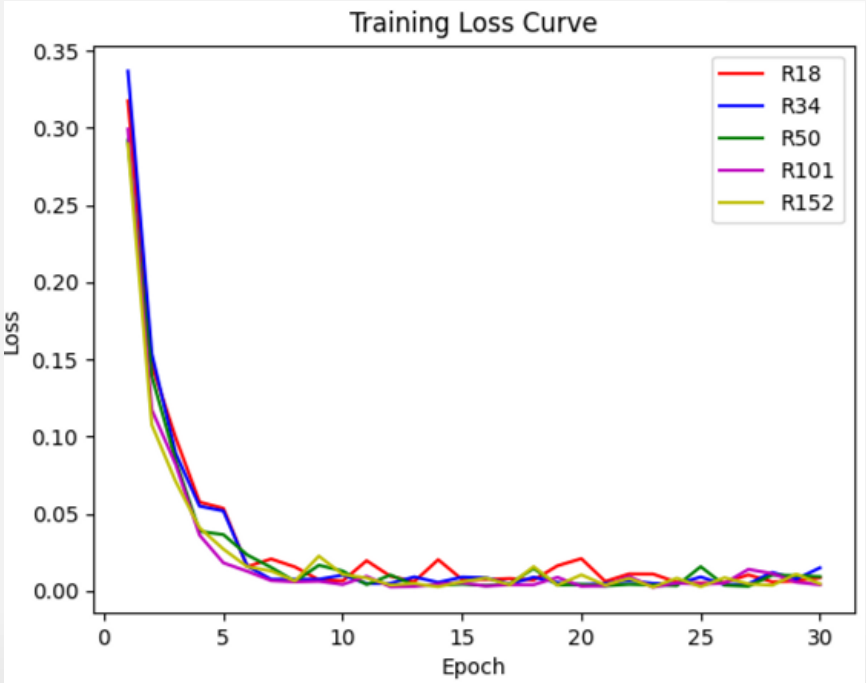
Model	Learning Rate	Epoch	Batch Size	Split	Opti.	Loss Func.	Test Acc.
ResNet18	0.01	5	9	6:2:2	SGD	Cross Entropy	79.97%
ResNet18	0.001	30	9	6:2:2	SGD	Cross Entropy	83.07%
ResNet34	0.001	30	9	6:2:2	SGD	Cross Entropy	83.97%
ResNet50	0.001	30	9	6:2:2	SGD	Cross Entropy	84.64%
ResNet101	0.001	30	9	6:2:2	SGD	Cross Entropy	86.91%
ResNet152	0.001	30	9	6:2:2	SGD	Cross Entropy	87.49%

IV. EXPERIMENTAL RESULTS

1.1 First stage

Compare ResNet with different layer





IV. EXPERIMENTAL RESULTS

1.2 First stage

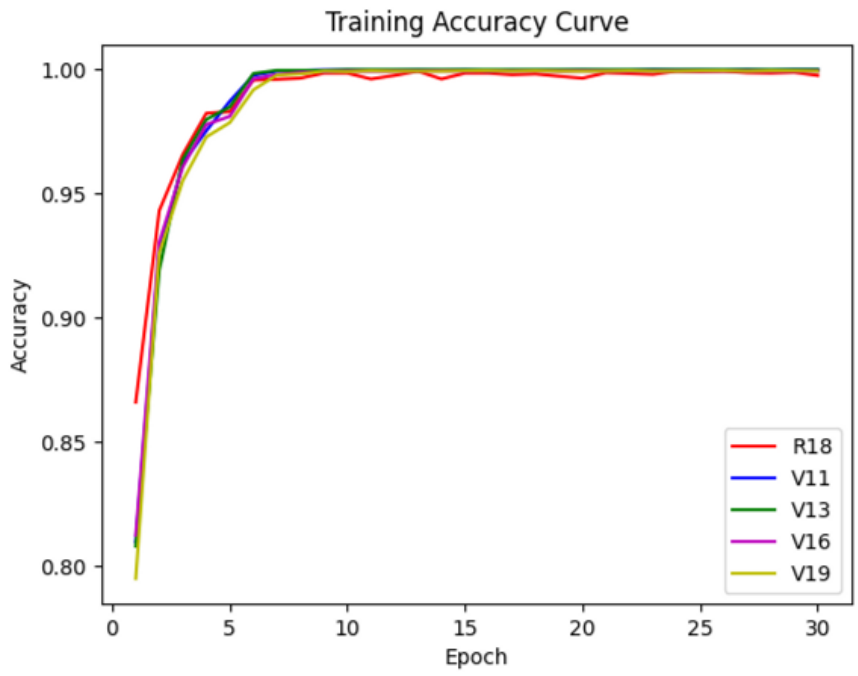
Compare VGG with different layer

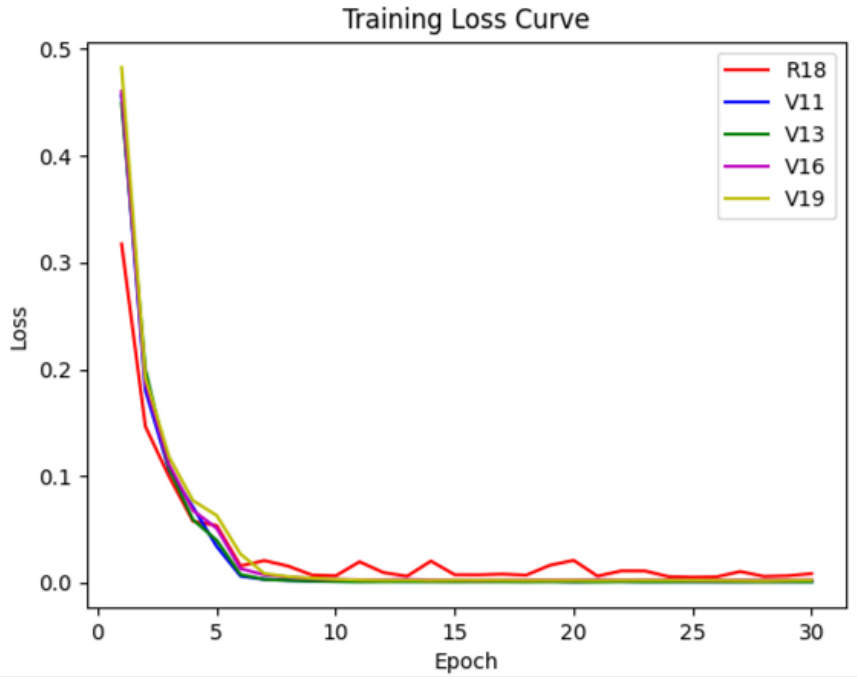
Model	Learning Rate	Epoch	Batch Size	Split	Opti.	Loss Func.	Test Acc.
VGG11	0.001	30	9	6:2:2	SGD	Cross Entropy	87.52%
VGG13	0.001	30	9	6:2:2	SGD	Cross Entropy	83.36%
VGG16	0.001	30	9	6:2:2	SGD	Cross Entropy	85.24%
VGG19	0.001	30	9	6:2:2	SGD	Cross Entropy	91.23%

IV. EXPERIMENTAL RESULTS

1.2 First stage

Compare VGG with different layer





IV. EXPERIMENTAL RESULTS

1.3 First stage

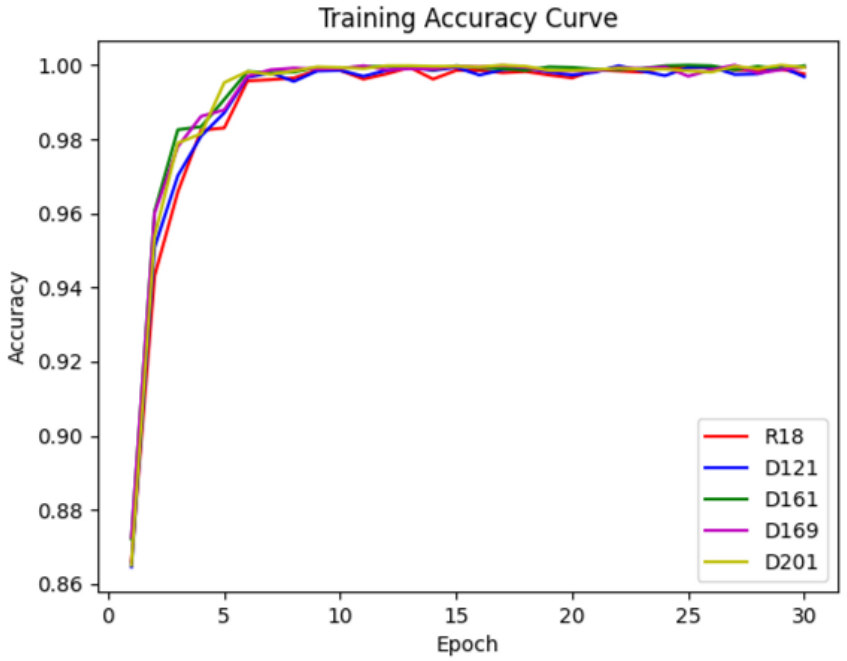
Compare DenseNet with different layer

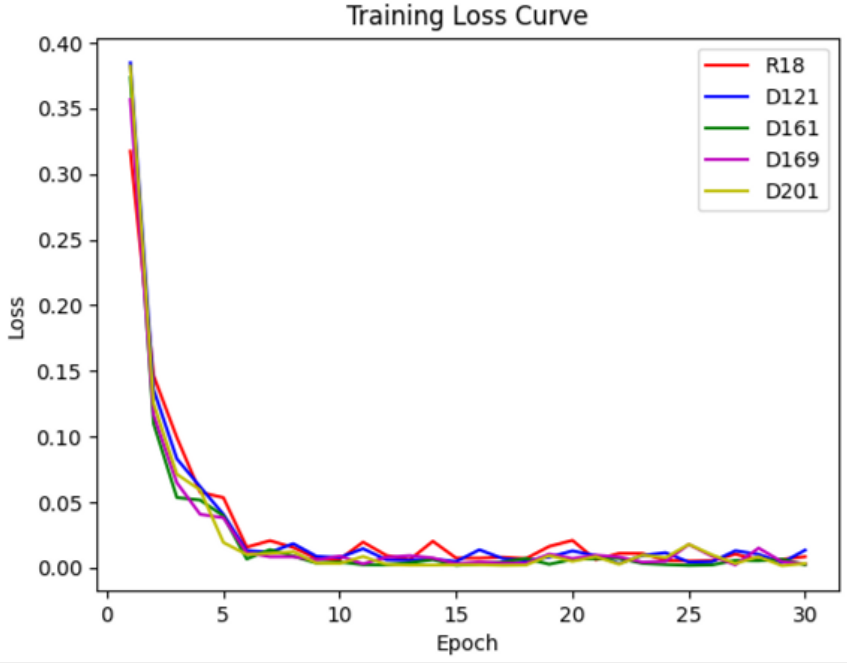
Model	Learning Rate	Epoch	Batch Size	Split	Opti.	Loss Func.	Test Acc.
DenseNet121	0.001	30	9	6:2:2	SGD	Cross Entropy	86.88%
DenseNet161	0.001	30	9	6:2:2	SGD	Cross Entropy	89.35%
DenseNet169	0.001	30	9	6:2:2	SGD	Cross Entropy	86.95%
DenseNet201	0.001	30	9	6:2:2	SGD	Cross Entropy	88.59%

IV. EXPERIMENTAL RESULTS

1.3 First stage

Compare DenseNet with different layer





IV. EXPERIMENTAL RESULTS

1.4 First stage

Compare different model architecture

Model	Learning Rate	Epoch	Batch Size	Split	Opti.	Loss Func.	Test Acc.
Res152	0.001	30	9	6:2:2	SGD	Cross Entropy	87.49%
VGG19	0.001	30	9	6:2:2	SGD	Cross Entropy	91.23%
DenseNet161	0.001	30	9	6:2:2	SGD	Cross Entropy	89.35%

IV. EXPERIMENTAL RESULTS

2. Second Stage

Hyperparameters setting of VGG19

Model	Learning Rate	Epoch	Batch Size	Split	Opti.	Loss Func.	Test Acc.
VGG19	0.001	30	9	6:2:2	SGD	Cross Entropy	91.31%
VGG19	0.001	30	16	6:2:2	SGD	Cross Entropy	89.82%
VGG19	0.001	30	9	7:1:2	SGD	Cross Entropy	89.12%
VGG19	0.001	30	16	7:1:2	SGD	Cross Entropy	91.63%

IV. EXPERIMENTAL RESULTS

3. Third stage

Data augmentation of data set

Model	Learning Rate	Epoch	Batch Size	Split	Opti.	Loss	Data Aug.	Test acc.
VGG19	0.001	30	16	7:1:2	SGD	Cross Entropy	-	91.63%
VGG19	0.001	30	16	7:1:2	SGD	Cross Entropy	Horizontal	89.03%
VGG19	0.001	30	16	7:1:2	SGD	Cross Entropy	Vertical	94.39%
VGG19	0.001	30	16	7:1:2	SGD	Cross Entropy	Horizontal +Vertical	92.47%

IV. EXPERIMENTAL RESULTS

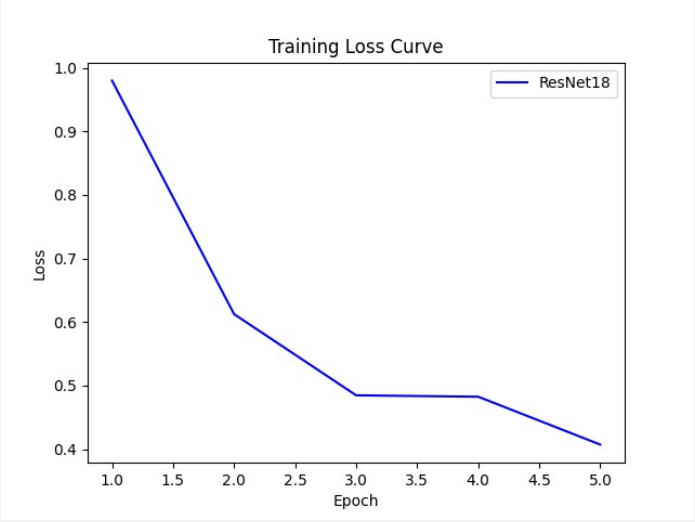
4.1 Default and Best

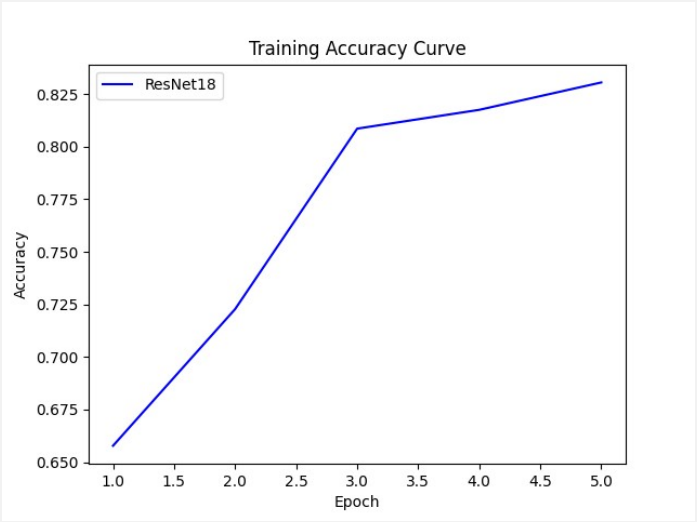
Model	Learning Rate	Epoch	Batch Size	Split	Opti.	Loss Func.	Data Aug.	Test Acc.
Default (ResNet18)	0.01	5	9	6:2:2	SGD	Cross Entropy	-	79.97%
Best (VGG19)	0.001	30	16	7:1:2	SGD	Cross Entropy	Vertical	94.39%

IV. EXPERIMENTAL RESULTS

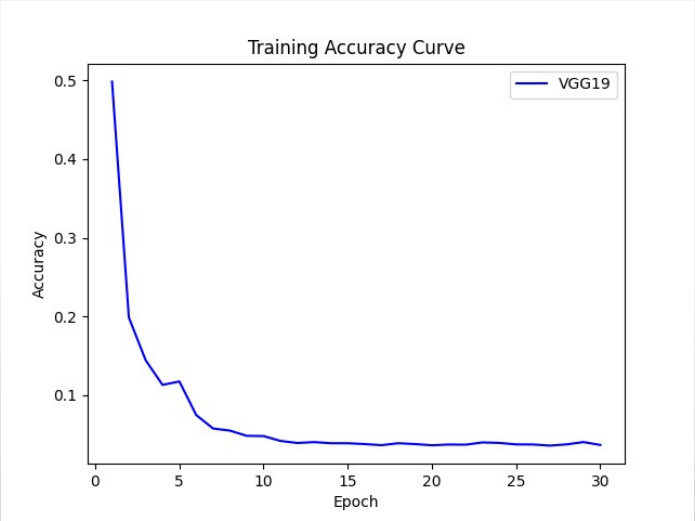
4.2 Training Accuracy and Loss Curve

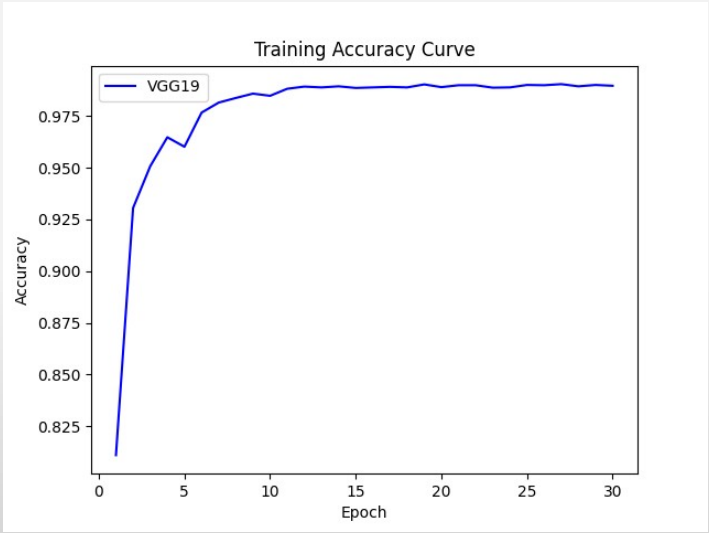
Default (ResNet18)





Best (VGG19)



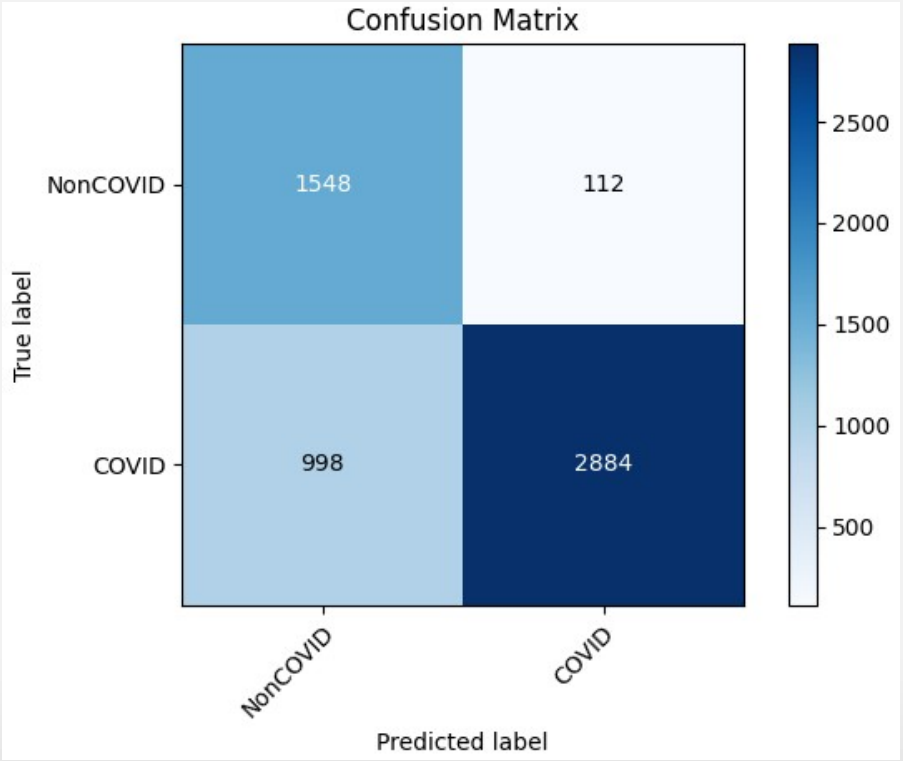


Loss

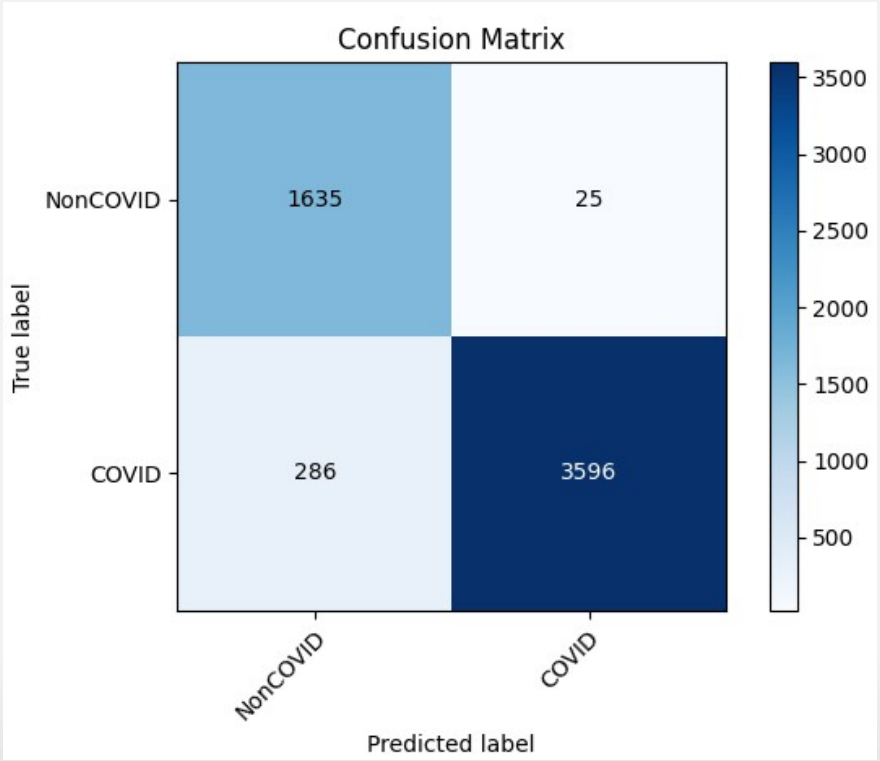
Accuracy

IV. EXPERIMENTAL RESULTS

4.3 Confusion Matrix



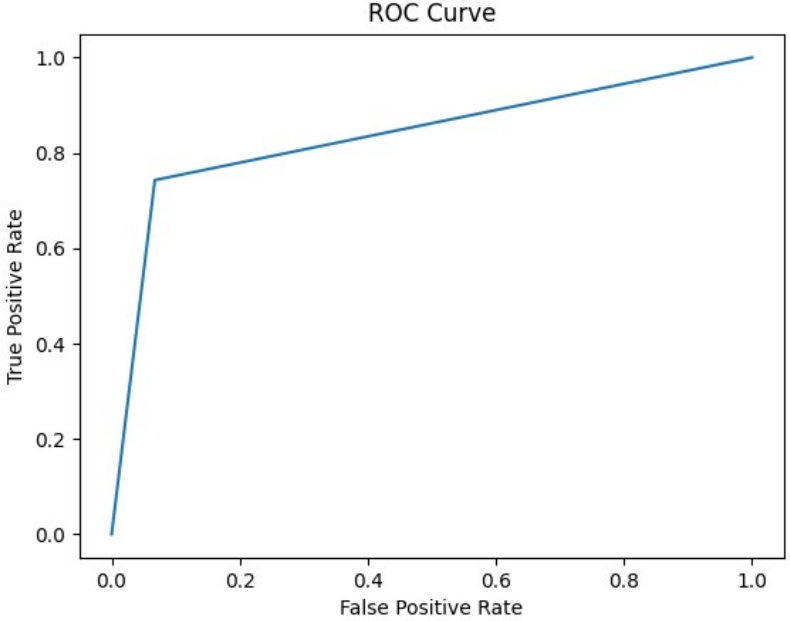
Default



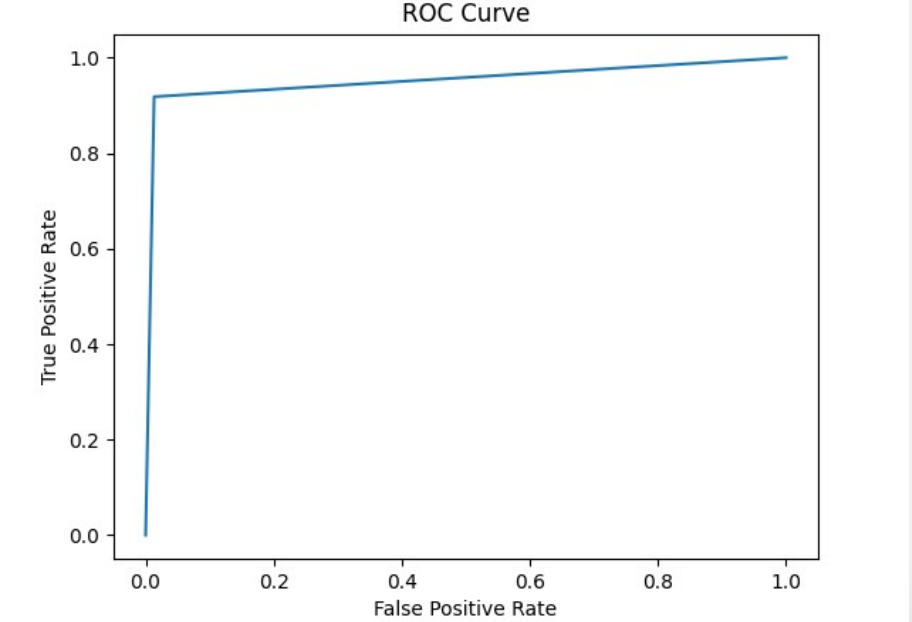
Best

IV. EXPERIMENTAL RESULTS

4.4 ROC Curve



Default



Best

IV. EXPERIMENTAL RESULTS

4.5 Evaluation Metrics

Model	Precision	Recall	F1 Score	AUC	Test Acc.
Default (ResNet18)	0.9626	0.7429	0.8386	0.8377	79.97%
Best (VGG19)	0.9931	0.9263	0.9585	0.9556	94.39%

Outline

I. INTRODUCTION
II. PROPOSED METHOD
III. IMPLEMENTATION
IV. EXPERIMENTAL RESULTS
V. DISCUSSION AND FUTURE RESEARCH
VI. CONCLUTION

V. DISCUSSION AND FUTURE RESEARCH

In the future, we can increase the dataset (color, angle, deformation), train with different models, and adjust various hyperparameters to achieve 100% accuracy for our model.

Outline

I. INTRODUCTION
II. PROPOSED METHOD
III. IMPLEMENTATION
IV. EXPERIMENTAL RESULTS
V. DISCUSSION AND FUTURE RESEARCH
VI. CONCLUTION

VI. CONCLUSION

We used three different models and found that VGG19 was the most suitable one.

Afterward, we adjusted the hyperparameters, setting the learning rate to 0.001, epoch to 30,data splitting to 7:1:2 and batch size to 16.

Then we performed image preprocessing, including vertical flipping in order to increase data augmentation.

Finally, our accuracy reached 94.39%.