

DIGITAL MEDICINE

CASE2:

COVID-19 PNEUMONIA
DETECTION

CT_02_IKEA

葉詠富 310554031

游智鈞 310551059

高承翰 310551106

CONTENTS

- **01 Introduction**
- **02 Image Preprocessing**
- **03 Method**
- **04 Result**



01 INTRODUCTION



TARGET AND DATASET

Target

Use chest X-ray image to determine the type of new coronary pneumonia.

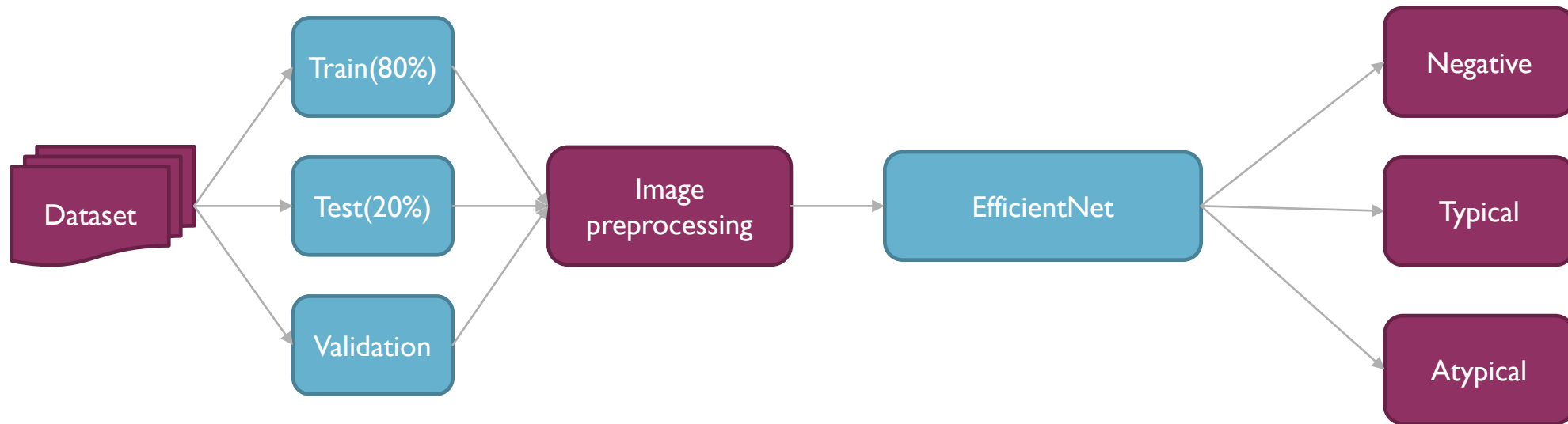
Train Dataset

Dicom files : X-ray images of 1200 patients
Excel files : labels of 1200 patients

Validation Dataset

Dicom files : X-ray images of 150 patients

DATA PIPELINE





02 IMAGE PREPROCESSING



IMAGE CUT

a. cut

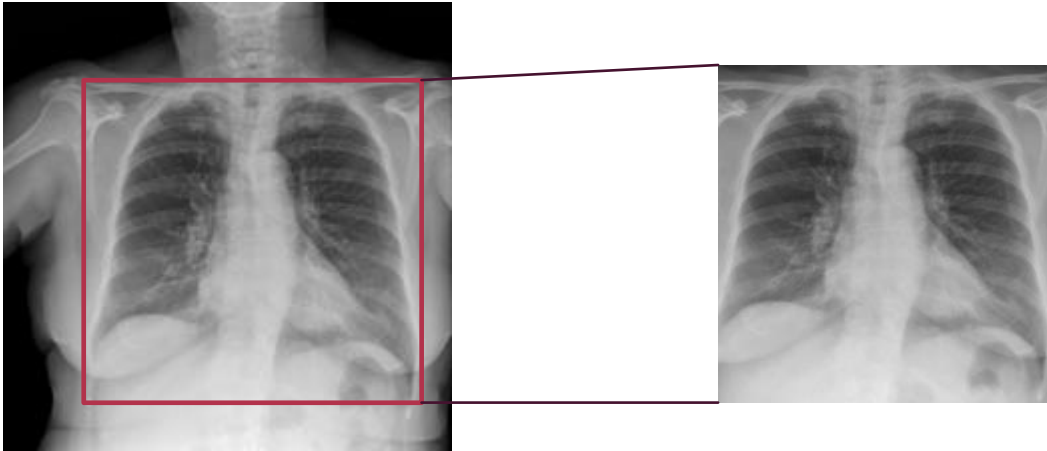


IMAGE AUGMENTATION

a. Rotation



b. Translation



c. Scaling



d. Flipping

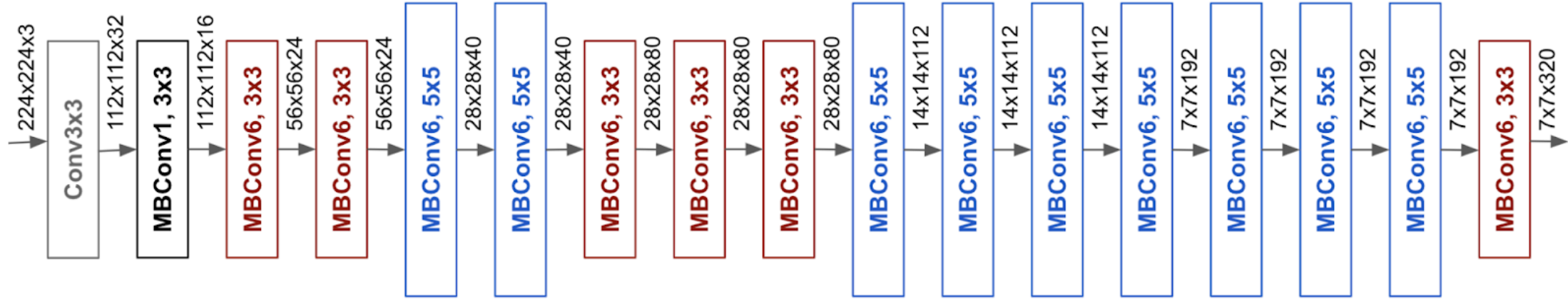




03 METHOD

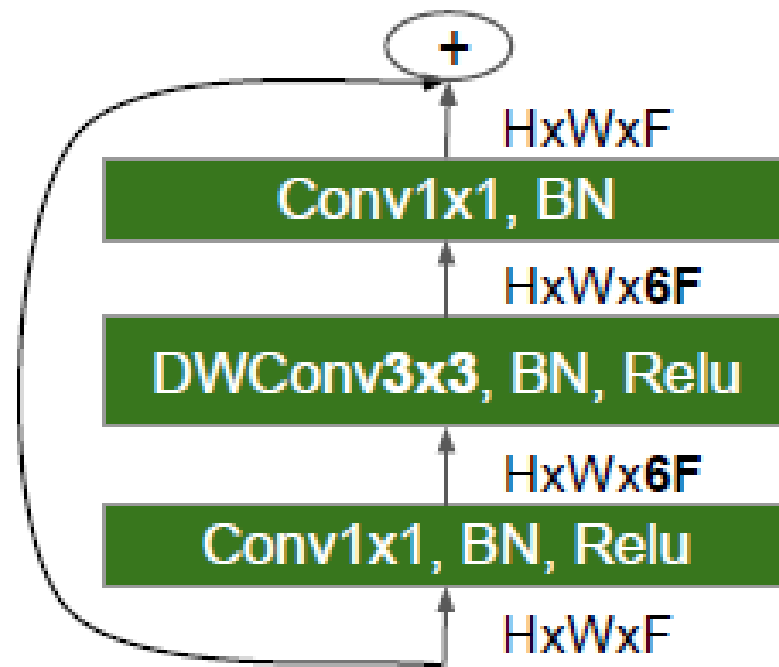


EFFICIENTNET

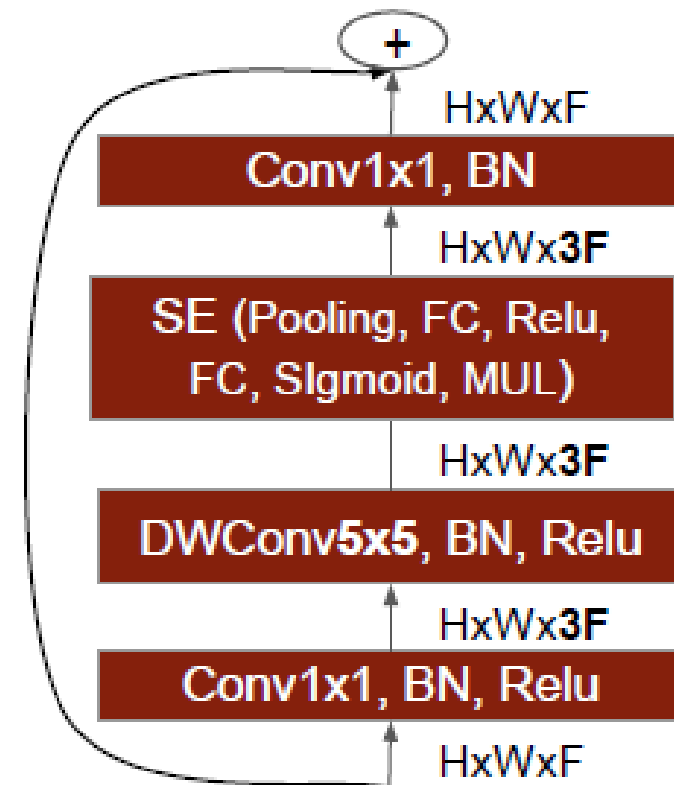


EFFICIENTNET

MBConv6 (k3x3)

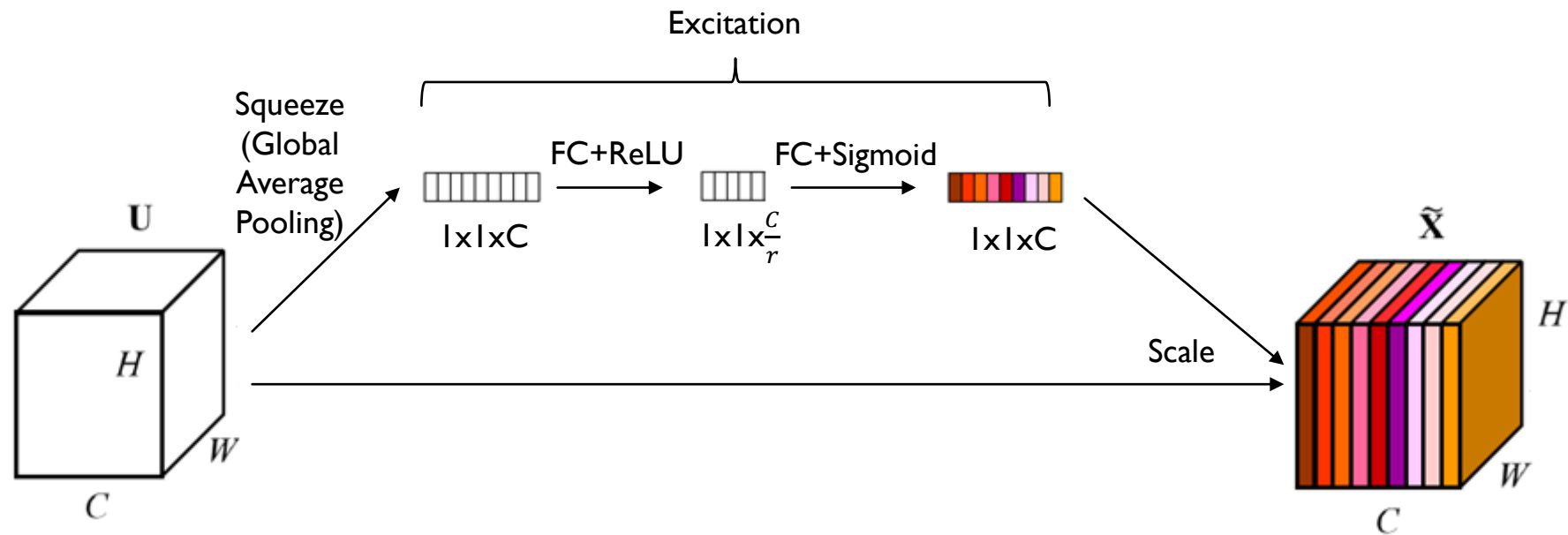


MBConv3 with SE (k5x5)



EFFICIENTNET

Squeeze-and-Excitation





04 RESULT



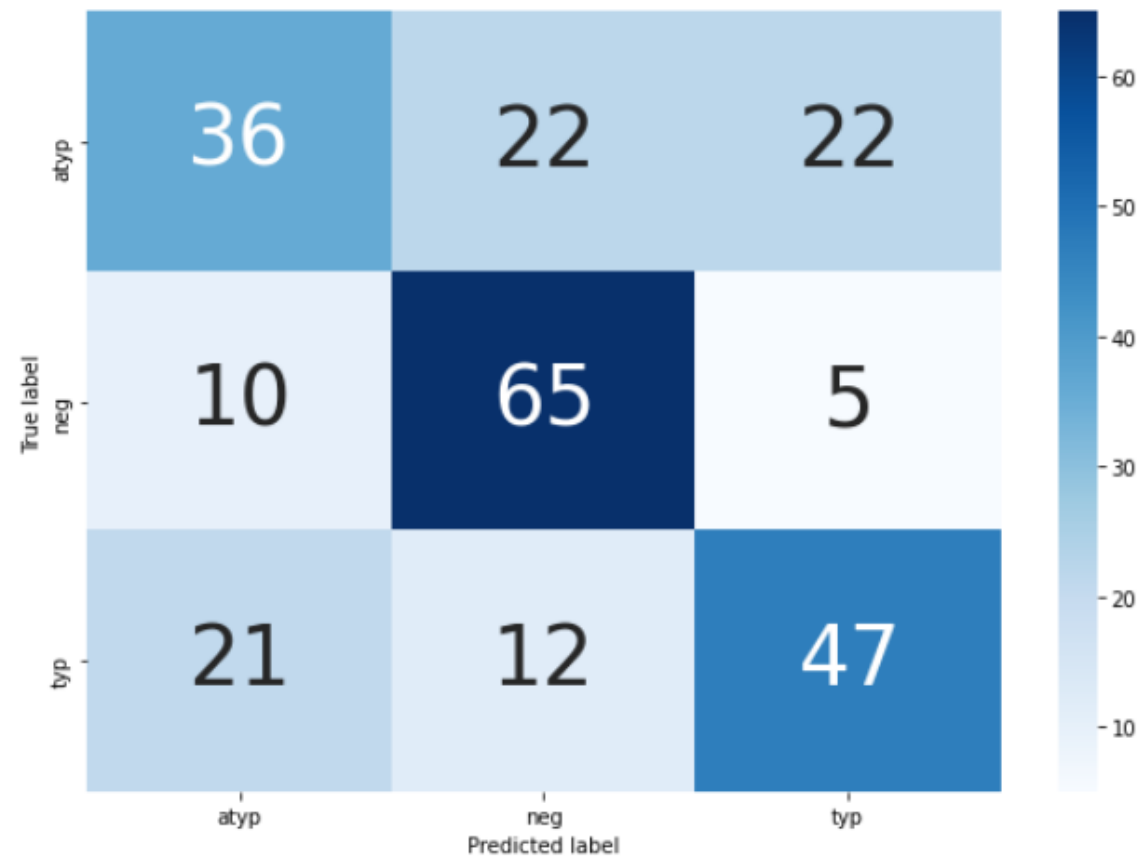
RESULT

a. Test dataset

FI-score	0.609

b. Validation

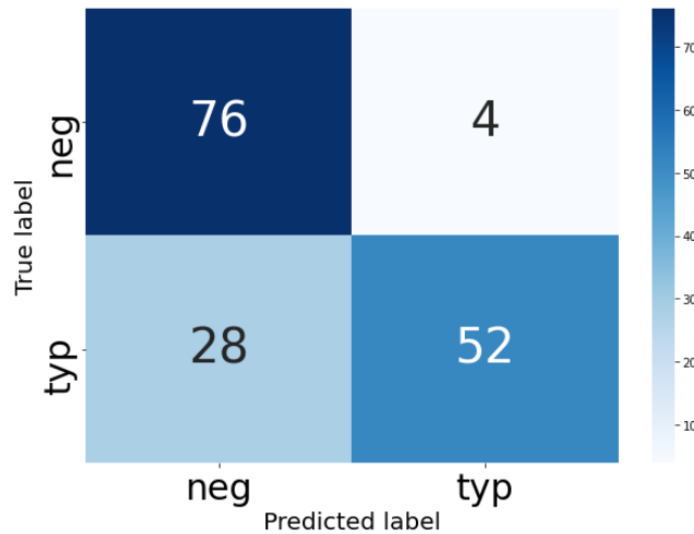
FI-score	0.6000



RESULT

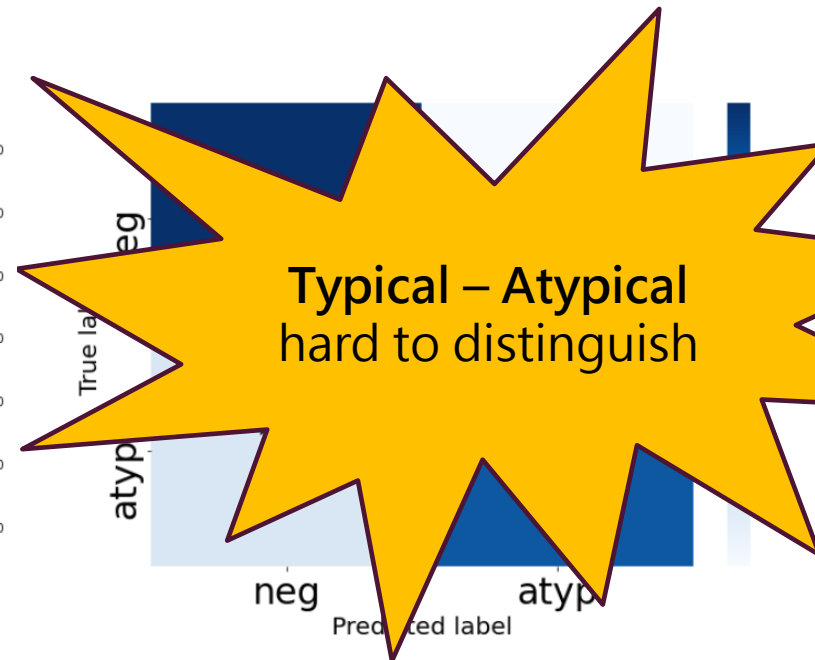
a. Negative - Typical

FI-score	0.80
----------	------



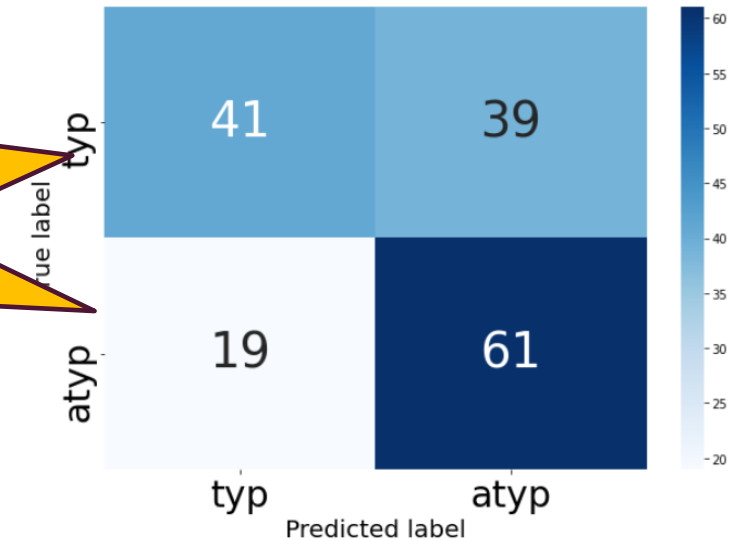
b. Negative - Atypical

FI-score	0.77
----------	------



c. Typical - Atypical

FI-score	0.63
----------	------



Typical – Atypical
hard to distinguish

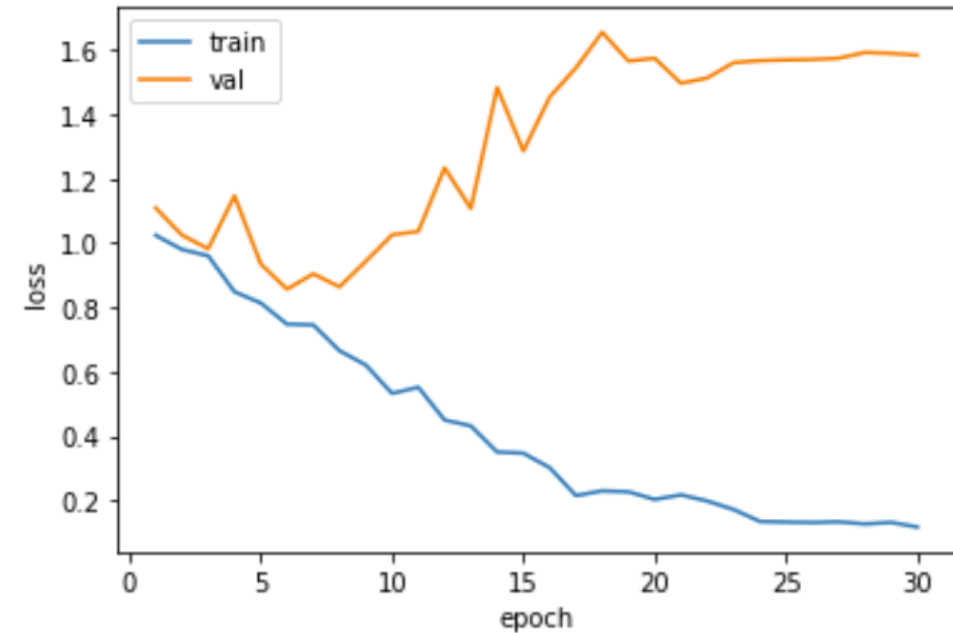
CONCLUSION

a. Problems

no	Problem
1	Overfitting?
2	Mislabeled?

b. Future Works

1. Learning how to distinguish these three types of images.
2. Combine different types of data except images.






GITHUB




GITHUB

Case presentation 2

https://github.com/CHKao777/DM_HW2

 CHKao777 Update README.md 2cf99f5 1 hour ago 16 commits

efficientnet_pytorch	Rename utilis (2).py to utilis.py	2 hours ago
README.md	Update README.md	1 hour ago
inference.py	Add files via upload	1 hour ago
requirements.txt	Update requirements.txt	2 hours ago
submission.py	Add files via upload	1 hour ago
train.ipynb	Add files via upload	2 hours ago

 README.md

Digital medical HW2

Environment setting

```
pip install -r requirements.txt
```

Retrain a model

1. Convert each .dcm file to .png image
2. Place data in the structure as shown below

```
data // for model training and testing
├── train
│   ├── Negative
│   │   ├── xxx.jpg
│   │   ├── xxx.jpg
│   │   └── ...
│   ├── Typical
│   │   ├── xxx.jpg
│   │   ├── xxx.jpg
│   │   └── ...
│   └── Atypical
│       ├── xxx.jpg
│       ├── xxx.jpg
│       └── ...
└── val
```



REFERENCE



REFERENCE

[1] CNN模型- ResNet、MobileNet、DenseNet、ShuffleNet、EfficientNet

(<https://cinnamonaitaiwan.medium.com/cnn%E6%A8%A1%E5%9E%8B-resnet-mobilenet-densenet-shufflenet-efficientnet-5eba5c8df7e4>)

[2] Pytorch Tutorial (<https://pytorch.org/tutorials/>)

[3] TorchVision Transforms: Image Preprocessing in PyTorch (<https://sparrow.dev/torchvision-transforms/>)

[4] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks (<https://arxiv.org/pdf/1905.11946.pdf>)



CONTRIBUTION OF GROUP MEMBERS



CONTRIBUTE

Name	Responsible	contact
葉詠富	<ol style="list-style-type: none">1. Image preprocessing2. Model training-CNN、Randomforest3. PPT revised and finalized	frankye100.c@nycu.edu.tw
游智鈞	<ol style="list-style-type: none">1. Image preprocessing2. Model training-Efficientnet3. Result analysis4. PPT draft	thomas91714@gmail.com
高承翰	<ol style="list-style-type: none">1. Image preprocessing2. Model training-Efficientnet3. Result analysis4. GITHUB5. Project speaker	climnehcc234@gmail.com



THANK YOU

