LAB1 Experiment Report

Pneumonia Classification from Chest X-ray Images

Student ID: 314553018

Name: 王致雅

Date: October 2025

1. Introduction

Pneumonia is a potentially fatal lung infection diagnosable through chest X-rays. Since manual interpretation requires expertise and varies between clinicians, automated diagnostic systems serve as valuable clinical support tools.

This lab develops a deep learning binary classifier for pneumonia detection from X-rays. Four deep learning models were implemented —ResNet18, ResNet50, DenseNet121, and Vision Transformer (ViT-Base)—using transfer learning on the Kaggle Chest X-Ray Images (Pneumonia) dataset. To improve generalization and reproducibility, I applied data augmentation, early stopping, and random seed control.

Enhanced data augmentation significantly boosted performance: ResNet18's accuracy increased by ~4% (87.18% to 91.51%) and validation loss decreased from 0.4035 to 0.1926. These improvements indicate better generalization and more stable data representations.

Most models exceeded 90% accuracy. ResNet50 performed best (91.67% accuracy, 0.9364 F1-score), closely followed by ResNet18 and ViT-Base. DenseNet121 converged fastest but achieved only 87.82% accuracy, suggesting poorer generalization. Results confirm both convolutional and transformer architectures effectively classify medical images when properly augmented.

This experiment provided hands-on experience with transfer learning, fine-tuning, and augmentation techniques on medical data, along with quantitative model evaluation across different architectures.

2. Experiment Setups

2.1 Model Details

In this lab, I implemented and evaluated four different deep learning architectures for pneumonia classification from chest X-ray images: **ResNet18**, **ResNet50**, **DenseNet121**, and **Vision Transformer (ViT-Base Patch16 224)**. All models were initialized with ImageNet pretrained weights. The original classifier heads were replaced with a fully connected layer outputting 2 classes (Normal and Pneumonia).

Table1. Model Structure

Model	Characteristics	Modification
ResNet18	Residual network with 18 layers	The final fully connected layer was replaced with a linear layer outputting 2 classes (Normal and Pneumonia)
ResNet50	Residual network with 50 layers	
DenseNet121	Dense connectivity between layers	The classifier layer was replaced for binary classification
ViT-Base	The input image is split into 16×16 patches, and a Transformer encoder with self-attention is applied.	Fine-tuned for binary classification using timm library

2.2 Training Setup

Most of the training parameters follow the default values provided in the assignment sample code. Additional settings, such as the random seed and Early Stopping patience, were added to ensure reproducibility and prevent overfitting.

Table2. Training Strategy

Parameter / Feature	**Value**	**Description / Purpose**
Early Stopping	Patience = 6 (based on epoch / 5)	Stops training automatically if validation loss does not improve for 6 consecutive epochs. Helps prevent overfitting and reduces unnecessary training time.
Random Seed	39	Applied to both NumPy and PyTorch to make data loading, weight initialization, and training deterministic. Ensures consistent and comparable results across runs.
Model Selection	Lowest validation loss	The best checkpoint for each architecture is selected based on validation performance, ensuring optimal generalization on unseen data.

2.3 Dataloader & Data Augmentation

A custom dataloader was implemented to efficiently load and preprocess chest X-ray images for both training and evaluation. For training, the assignment sample code provided two augmentations: Random Rotation and Resize. To improve model robustness and generalization, I **added four additional augmentation** techniques:

Table3. Data Augmentation Strategy

Enhancement	Parameters	Purpose
Random Resized Crop	scale=0.8 ~ 1.0	Introduces scale variation, simulates X-rays taken at different distances, and helps the model learn features at multiple scales.
Random Horizontal Flip	p=0.3	Exploits lung symmetry, increases data diversity while maintaining anatomical plausibility.

Color Jitter	brightness=0.05,
	contrast=0.05
		Simulates variations in X-ray exposure and imaging equipment.

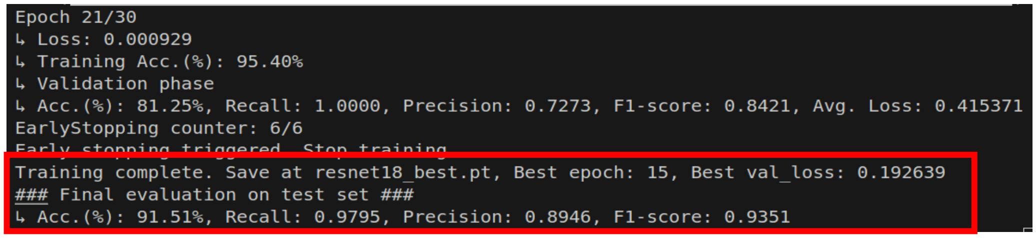
Random Erasing	p=0.1	Simulates occlusions or artifacts, forcing the model to rely on global context.

Normalization using ImageNet statistics was applied to all training images to standardize inputs for pretrained models.

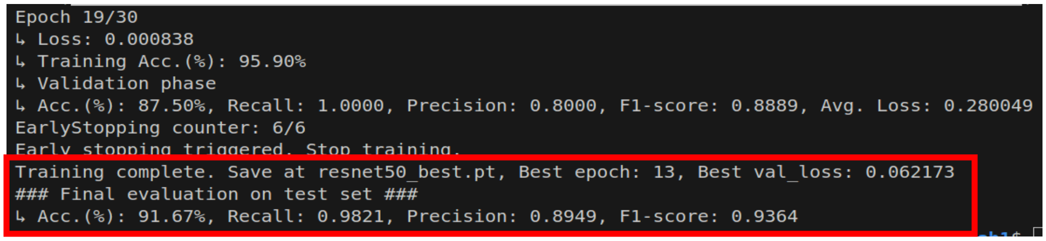
For validation and testing, only resizing and normalization were applied to ensure consistent evaluation. These preprocessing and augmentation steps ensure consistent input size, compatibility with pretrained models, and improved model generalization, helping to prevent overfitting, which is especially important when working with limited medical imaging datasets.

3. Experiment Result

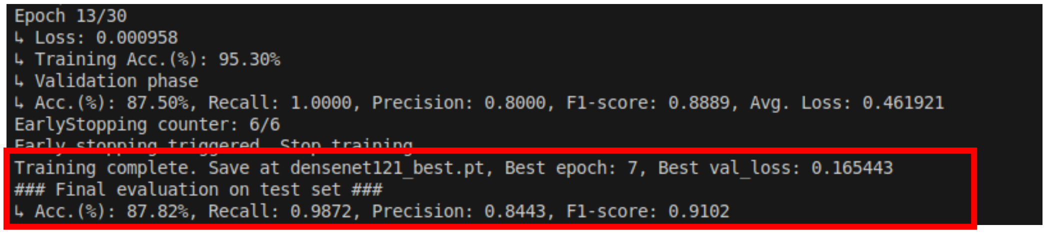
3.1 Highest testing accuracy and F1-score



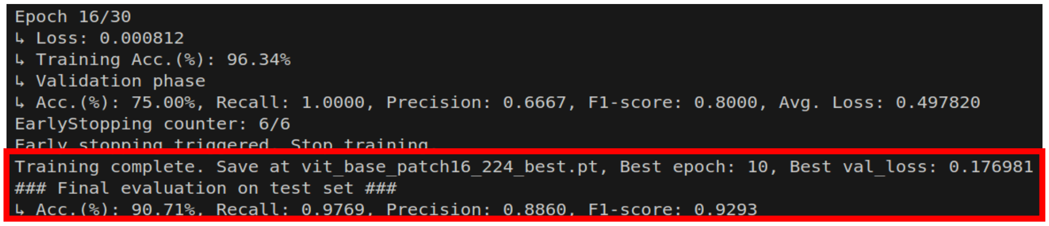
```
Epoch 21/30
↳ Loss: 0.000929
↳ Training Acc.(%): 95.40%
↳ Validation phase
↳ Acc.(%): 81.25%, Recall: 1.0000, Precision: 0.7273, F1-score: 0.8421, Avg. Loss: 0.415371
EarlyStopping counter: 6/6
Early stopping triggered. Stop training.
Training complete. Save at resnet18_best.pt, Best epoch: 15, Best val_loss: 0.192639
### Final evaluation on test set ###
↳ Acc.(%): 91.51%, Recall: 0.9795, Precision: 0.8946, F1-score: 0.9351
```



```
Epoch 19/30
↳ Loss: 0.000838
↳ Training Acc.(%): 95.90%
↳ Validation phase
↳ Acc.(%): 87.50%, Recall: 1.0000, Precision: 0.8000, F1-score: 0.8889, Avg. Loss: 0.280049
EarlyStopping counter: 6/6
Early stopping triggered. Stop training.
Training complete. Save at resnet50_best.pt, Best epoch: 13, Best val_loss: 0.062173
### Final evaluation on test set ###
↳ Acc.(%): 91.67%, Recall: 0.9821, Precision: 0.8949, F1-score: 0.9364
```



```
Epoch 13/30
↳ Loss: 0.000958
↳ Training Acc.(%): 95.30%
↳ Validation phase
↳ Acc.(%): 87.50%, Recall: 1.0000, Precision: 0.8000, F1-score: 0.8889, Avg. Loss: 0.461921
EarlyStopping counter: 6/6
Early stopping triggered. Stop training.
Training complete. Save at densenet121_best.pt, Best epoch: 7, Best val_loss: 0.165443
### Final evaluation on test set ###
↳ Acc.(%): 87.82%, Recall: 0.9872, Precision: 0.8443, F1-score: 0.9102
```



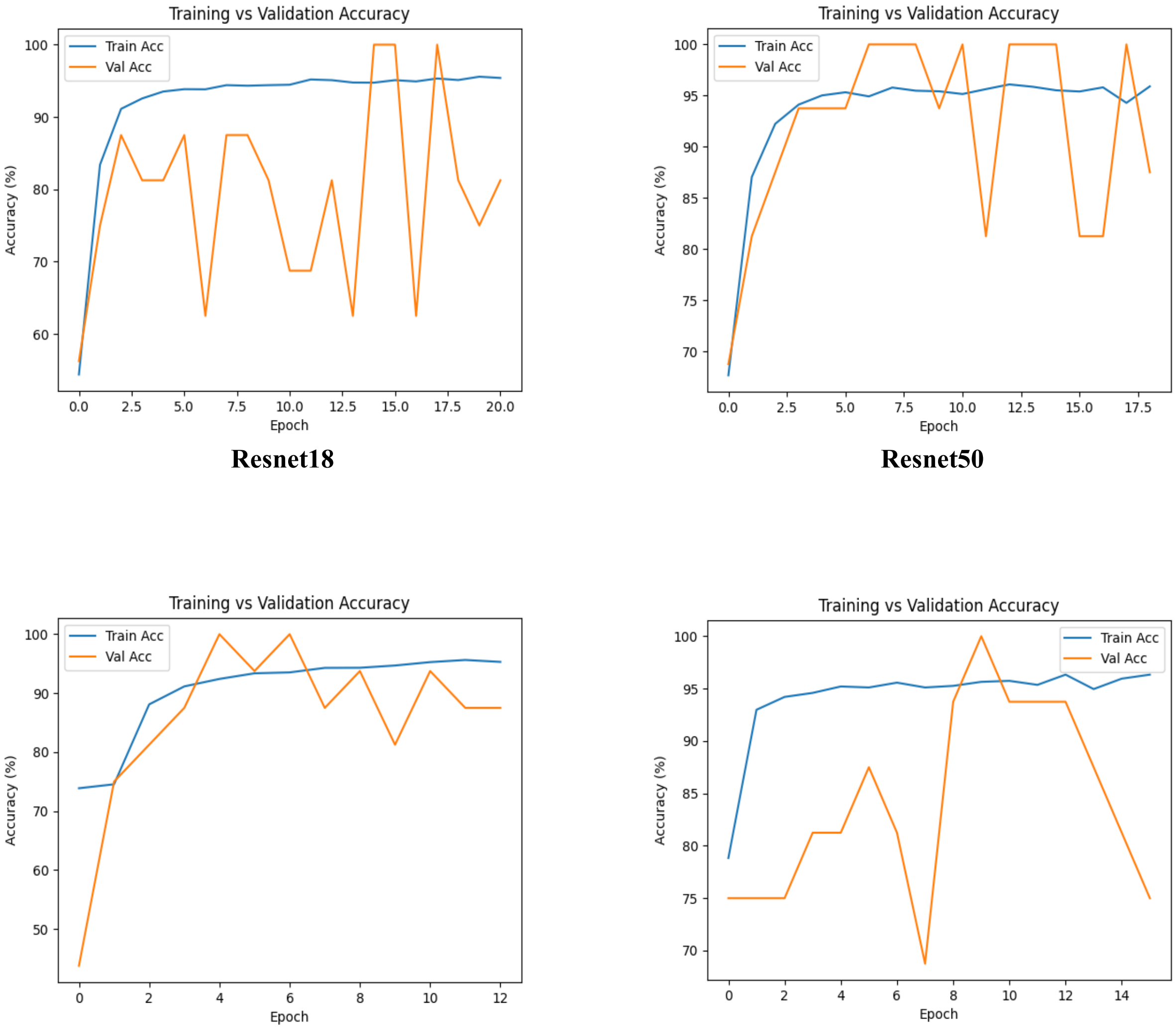
```
Epoch 16/30
↳ Loss: 0.000812
↳ Training Acc.(%): 96.34%
↳ Validation phase
↳ Acc.(%): 75.00%, Recall: 1.0000, Precision: 0.6667, F1-score: 0.8000, Avg. Loss: 0.497820
EarlyStopping counter: 6/6
Early stopping triggered. Stop training.
Training complete. Save at vit_base_patch16_224_best.pt, Best epoch: 10, Best val_loss: 0.176981
### Final evaluation on test set ###
↳ Acc.(%): 90.71%, Recall: 0.9769, Precision: 0.8860, F1-score: 0.9293
```

Table4. Model Performance Summary

Model	Best epoch	Accuracy	F1-Score	Precision	Recall
Resnet18	15	91.51%	0.9351	0.8946	0.9795
Resnet50	13	**91.67%**	**0.9364**	**0.8949**	0.9821
Densenet121	7	87.82%	0.9102	0.8443	**0.9872**
ViT-Base	10	90.71%	0.9293	0.8860	0.9769

3.2 Comparison figure

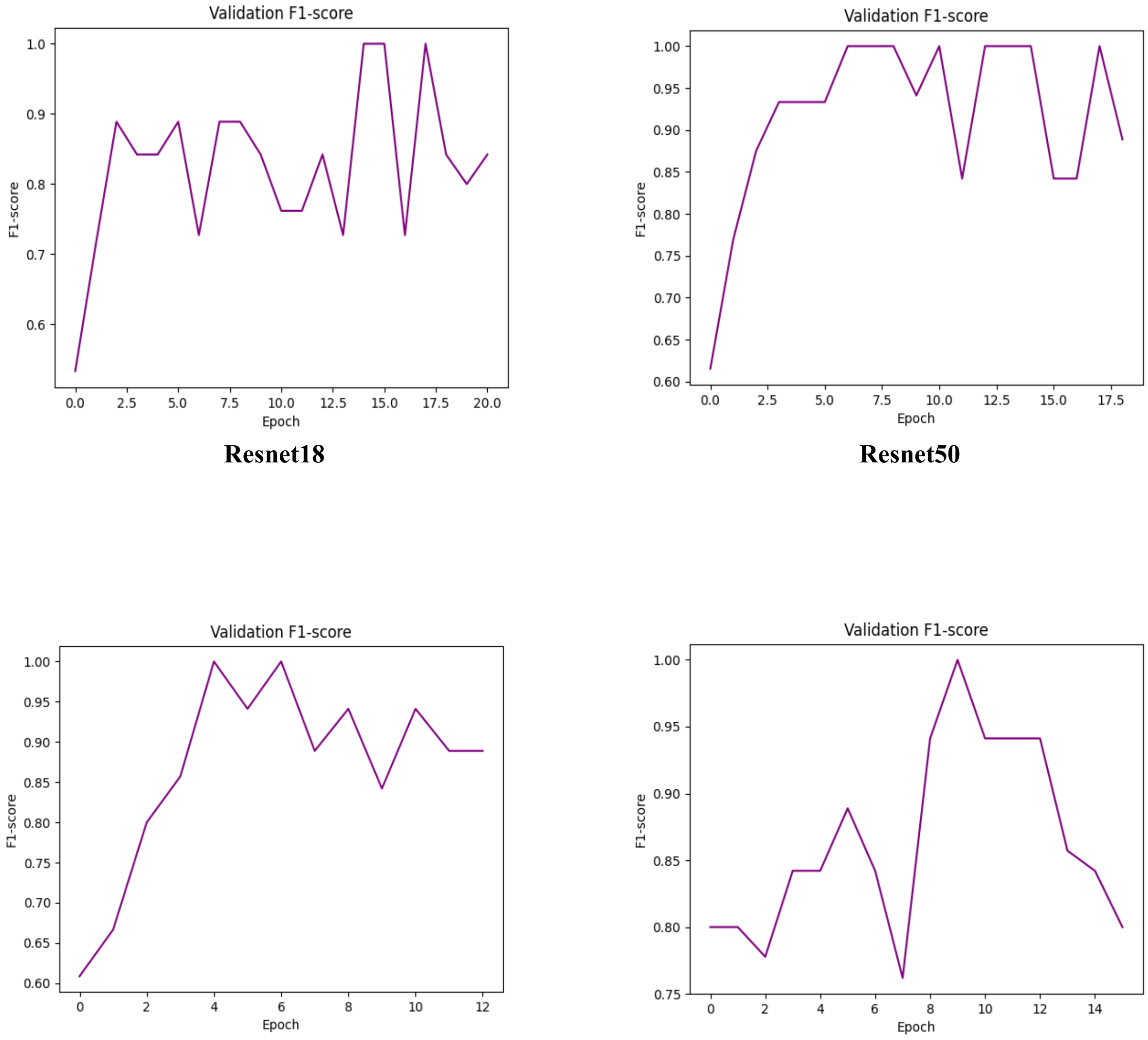
Figure1. Training and testing accuracy curve



Resnet18 **Resnet50**

Densenet121 **ViT-Base**

Figure2. Testing F1-score curve



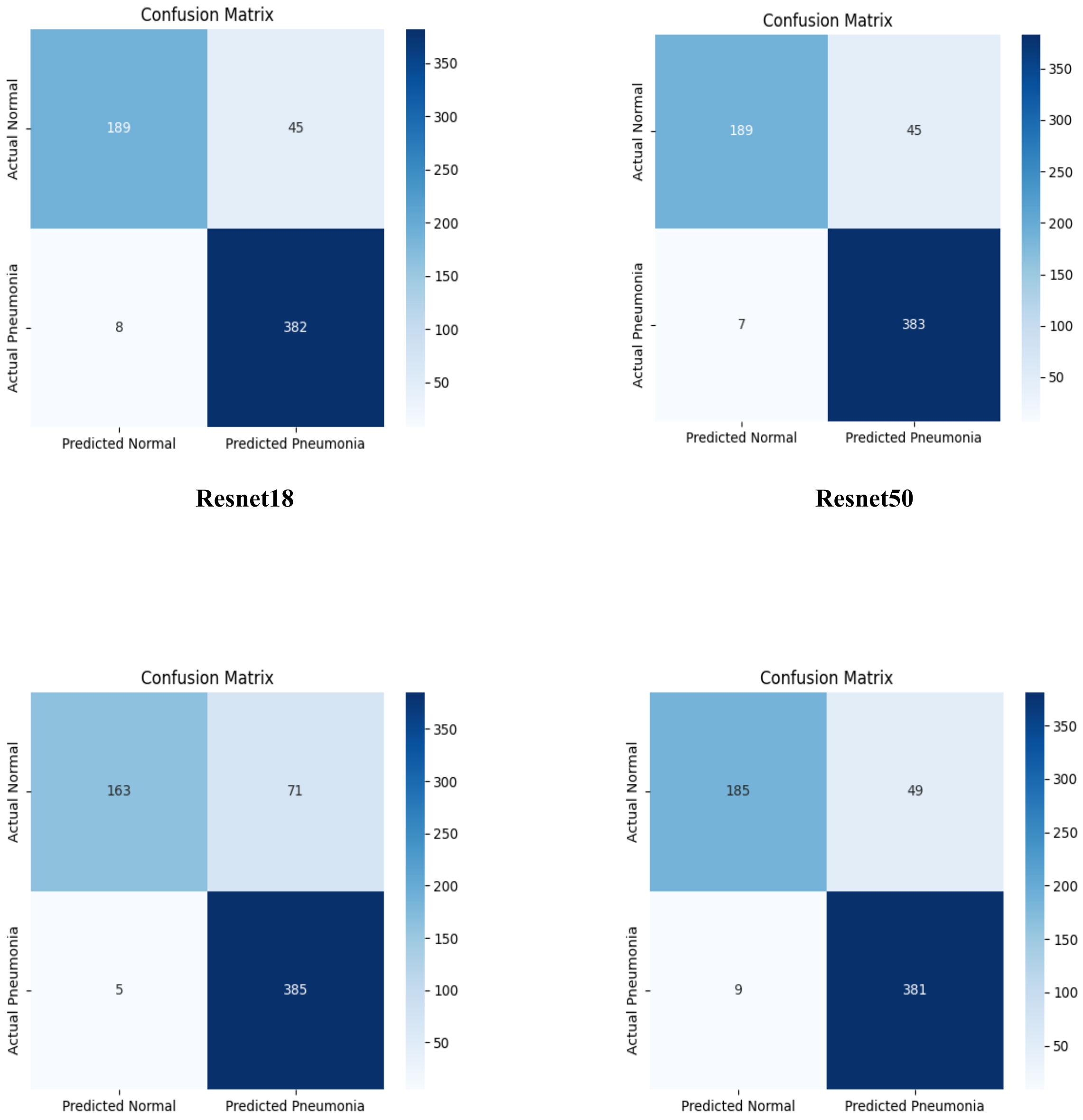
Resnet18

Resnet50

Densenet121

ViT-Base

Figure3. Highest testing accuracy heatmap



Resnet18

Resnet50

Densenet121

ViT-Base

4. Discussion

4.1 Model Selection Method

Originally, the best model in each training run was selected based on **validation accuracy**. However, it was observed that validation accuracy tended to fluctuate significantly, and in some cases reached 100% while training was still unstable. Such fluctuations made it difficult to reliably determine the optimal model.

In contrast, **validation loss** provided a smoother and more stable metric during training, without extreme spikes. By combining validation loss with the **Early Stopping** mechanism (patience), it was possible to efficiently and reliably identify the best model checkpoint for each architecture.

Therefore, in this experiment, the best model for each architecture was selected based on **validation loss** rather than validation accuracy. This approach is more sensitive to prediction confidence and helps reduce the risk of overfitting.

4.2 Impact of Data Augmentation

Data augmentation played a crucial role in improving the robustness and generalization of deep learning models in this experiment. To enhance performance, four additional augmentations were applied: Random Resized Crop, Random Horizontal Flip, Color Jitter, and Random Erasing, along with normalization.

Comparison between models trained with and without these additional augmentations is summarized below (using ResNet18):

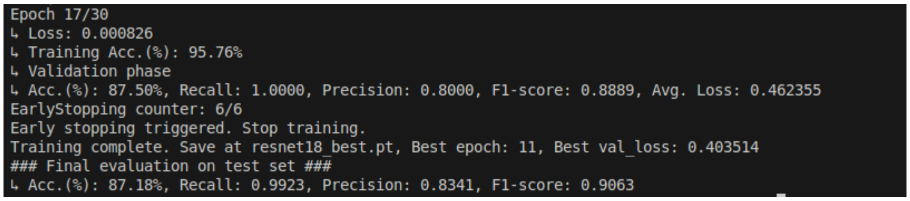
Table5. Comparison Before and After Data Augmentation with Resnet18

Training Setup	Without Additional Augmentation	With 4 Augmentations + Normalization
Best epoch	11	15
Best val loss	0.4035	0.1926
Test results	Accuracy = 87.18%, F1-score = 0.9063	Accuracy = 91.51%, F1-score = 0.9351

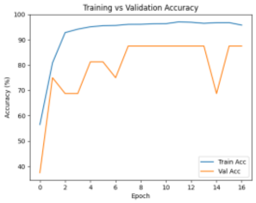
Without additional augmentations, validation accuracy lagged behind training accuracy, indicating overfitting. With the enhanced augmentations, the validation accuracy improved and the validation loss decreased significantly, showing more stable and confident predictions.

In conclusion, data augmentation substantially improves model generalization by exposing the network to variations in scale, orientation, exposure, and occlusion. The reduced gap between training and validation performance, along with the lower validation loss, indicates improved stability and reliability. A well-designed augmentation pipeline is therefore crucial for limited medical imaging datasets, as it enhances generalization and improves detection of critical cases.

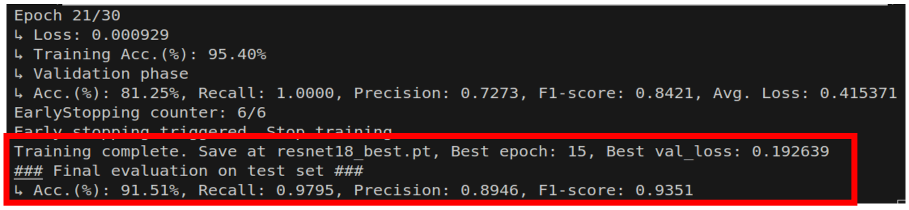
- Resnet18 without Additional Augmentation



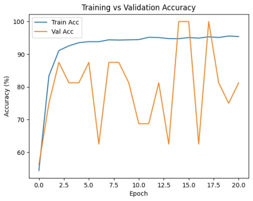
```
Epoch 17/30
↳ Loss: 0.000826
↳ Training Acc.(%): 95.76%
↳ Validation phase
↳ Acc.(%): 87.50%, Recall: 1.0000, Precision: 0.8000, F1-score: 0.8889, Avg. Loss: 0.462355
EarlyStopping counter: 6/6
Early stopping triggered. Stop training.
Training complete. Save at resnet18_best.pt, Best epoch: 11, Best val_loss: 0.403514
### Final evaluation on test set ###
↳ Acc.(%): 87.18%, Recall: 0.9923, Precision: 0.8341, F1-score: 0.9063
```



- Resnet18 with 4 Augmentations + Normalization



```
Epoch 21/30
↳ Loss: 0.000929
↳ Training Acc.(%): 95.40%
↳ Validation phase
↳ Acc.(%): 81.25%, Recall: 1.0000, Precision: 0.7273, F1-score: 0.8421, Avg. Loss: 0.415371
EarlyStopping counter: 6/6
Early stopping triggered. Stop training.
Training complete. Save at resnet18_best.pt, Best epoch: 15, Best val_loss: 0.192639
### Final evaluation on test set ###
↳ Acc.(%): 91.51%, Recall: 0.9795, Precision: 0.8946, F1-score: 0.9351
```



4.3 Model Comparison

As shown in Table 4 (Page 6), ResNet50 achieved the highest overall performance, demonstrating a balanced trade-off among accuracy, F1-score, precision, and recall. ResNet18 exhibited slightly lower performance, which may be attributed to its shallower network depth.

Interestingly, DenseNet121 reached its best model at the earliest epoch (epoch 7) according to the training and validation accuracy curves. However, it yielded the lowest overall effectiveness, indicating that rapid convergence does not necessarily translate into superior generalization.

Both ResNet50 and ResNet18 displayed gradual and stable improvements throughout training. ResNet50 converged marginally faster and achieved slightly better results in both training and validation metrics compared to ResNet18.

The ViT-Base model exhibited a distinctive training behavior: initial performance was comparatively lower, but it improved rapidly during mid-training, ultimately surpassing 90% accuracy. Moreover, ViT-Base required fewer training epochs than the ResNet architectures, highlighting the potential efficiency of Transformer-based models in medical image classification tasks.

These findings suggest that early convergence is not always indicative of final performance, and different network architectures can demonstrate distinct training dynamics even when trained on identical datasets with similar hyperparameter settings.

5. GitHub & Trained Weight Link

The complete implementation is available on GitHub:

https://github.com/jhihyawang/NYCU_114Autumn_AIMI/tree/main/Lab1

All trained model weights are uploaded to Google Drive for reproducibility and evaluation:

https://drive.google.com/drive/folders/1VGDWFlrFWiWzPz6YoIkRf70iwcZLnbyR

The file weight.zip contains the following checkpoints:

- resnet18_best.pt
- resnet50_best.pt
- densenet121_best.pt
- vit_base_patch16_224_best.pt

Each model was trained on the chest X-ray dataset following the training setup detailed above, and the weights were saved based on the lowest validation loss.