Deep Learning for Pneumonia Detection in Chest X-rays

Authors: Lilach Pardess (204506802), Assaf Taubenfeld (302318456)

Introduction

This project's goal is to implement, evaluate, and compare two different neural network architectures, CNN and Vision Transformer (ViT), for classifying pediatric chest X-ray images as either 'Normal' or 'Pneumonia'.

Dataset

The dataset used for this project is the "Chest X-Ray Images (Pneumonia)" collection, available on Kaggle. The dataset contains 5,863 JPEG images, categorized into two classes: Pneumonia and Normal. The images are split into training, validation, and test sets. Initial analysis revealed significant challenges, including class imbalance and a very small validation set, which are addressed in the Methodology section.

Methodology

Implementation details

- **Framework:** The models were implemented using the **PyTorch** deep learning library.
- **Environment:** Experiments were conducted within **Google Colab notebooks** (.ipynb), and Python
- **Hardware:** Training was performed on a local machine equipped with an **NVIDIA GeForce RTX 3060 GPU**.

Implementation plan

1. Implement CNN and ViT networks to classify the images
2. Compare the following parameters and find the best for the dataset:
    a. Optimizer: SGD vs Adam
    b. Initial learning rate
    c. Batch size
    d. Data augmentation
3. Use the learned parameters as a base line for future experimentation with CNN and ViT architectures.
4. Test CNN architecture. We will test CNN with 2,3,4 convolution layers.
5. Test ViT architecture. We test different freeze options. As we fine tune the bigger model, we will test what will happen if we un-freeze the feature extraction layers for 1, 5, 20 epochs and compare the results.
6. Compare the best CNN and ViT results.

*Note: the comparison of optimizer, initial learning rate and batch size are not included in this report for the sake of simplicity (and as it is less interesting testing in our opinion).

CNN design rationale

A custom CNN was implemented from scratch to gain a granular understanding of the architecture's components. This approach allowed for direct experimentation with network depth and filter sizes. The experimental process was iterative: a baseline model was established, and individual parameters were modified one at a time. The best-performing configuration became the new baseline for subsequent experiments.

ViT design rationale

We will follow the provided paper and fine tune an existing ViT model to our use case. As the paper suggests the ViT performs well only after training on a large dataset, our relatively small dataset does not allow us to train from scratch.

We will use "google/vit-base-patch16-224" as the base model to fine tune.

Experimental results

Evaluation Metrics

Model performance was assessed using a standard set of classification metrics to provide a comprehensive evaluation:

- **Accuracy:** The ratio of correctly classified images.
- **Precision:** The model's ability to correctly identify true 'Normal' cases (True Negatives / (True Negatives + False Positives)).
- **Recall (Sensitivity):** The model's ability to correctly identify true 'Pneumonia' cases (True Positives / (True Positives + False Negatives)).
- **F1 Score:** The harmonic mean of Precision and Recall, serving as the **primary metric** for model selection due to its effectiveness in balancing the two, especially with class imbalance.
- **AUC-ROC:** The area under the Receiver Operating Characteristic curve, which measures the model's discriminative power across all classification thresholds.

Data Preprocessing

Initial data analysis revealed two key challenges that required preprocessing steps:

1. **Class Imbalance:** The training data exhibited a significant imbalance, with 'Pneumonia' cases outnumbering 'Normal' cases by a nearly 3:1 ratio (3,106 to 1,079). To mitigate bias towards the majority class, a **weighted loss function** was employed by setting the pos_weight parameter in PyTorch's BCEWithLogitsLoss. This increases the penalty for misclassifying the minority 'Normal' class.
2. **Insufficient Validation Set:** The original validation set contained only 16 images, which was inadequate for reliable model evaluation due to high metric variance and potential for poor class representation. To resolve this, the training data was re-partitioned using an **80-20 stratified split**, creating a more robust validation set.

The final data distribution after preprocessing was:

- **Training Set:** 4,185 images (3,106 Pneumonia, 1,079 Normal)
- **Validation Set:** 1,047 images (777 Pneumonia, 270 Normal)

Experiment 1: The impact of data augmentation (CNN only).

This experiment evaluated CNN and ViT models performance under three distinct data augmentation strategies: no augmentation, light augmentation, and aggressive augmentation.

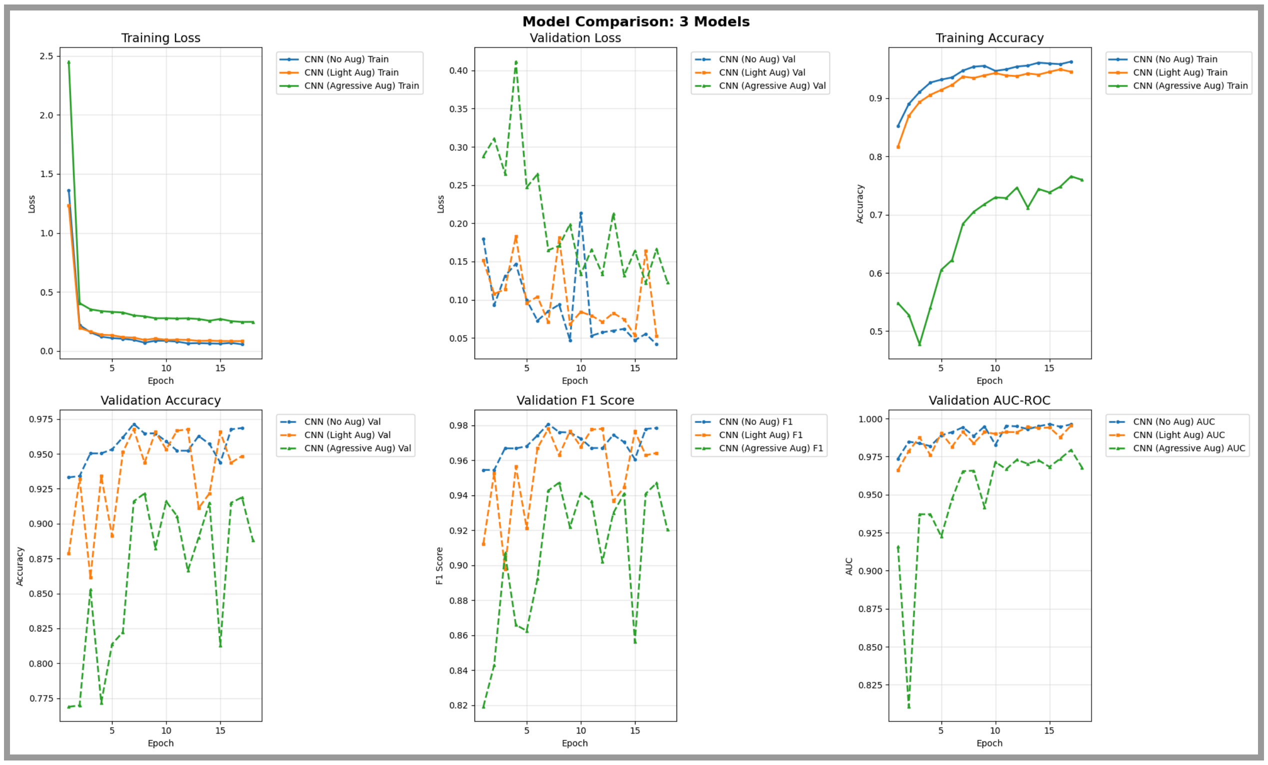
- **Light Augmentation:** Utilized transforms.RandomAffine to apply gentle, geometric transformations. Each image was randomly rotated (±10°), translated horizontally and vertically (±10%), and scaled (90%-110%). These changes were designed to be clinically plausible, simulating minor variations in patient positioning.

- **Aggressive Augmentation:** Employed transforms.RandAugment, which applies a sequence of diverse, randomly selected transformations (e.g., shearing, solarizing, extreme contrast adjustments) with a magnitude of 9.

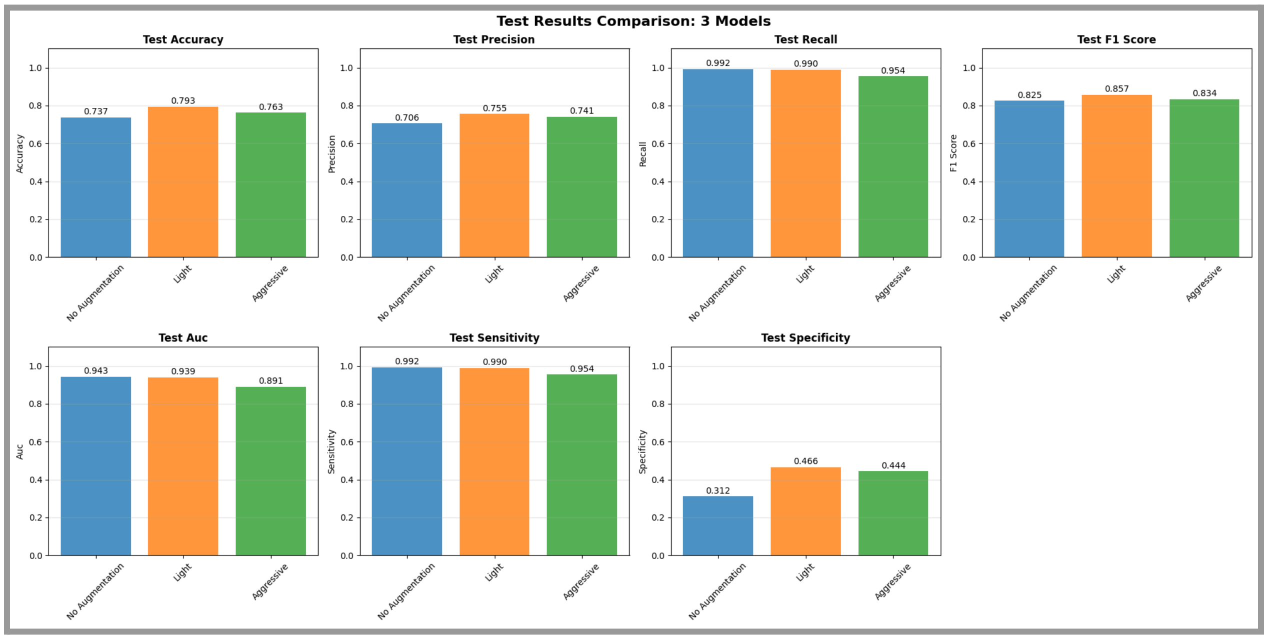
***Training time (for all 3 models, sequentially): 9 minutes 35 seconds**

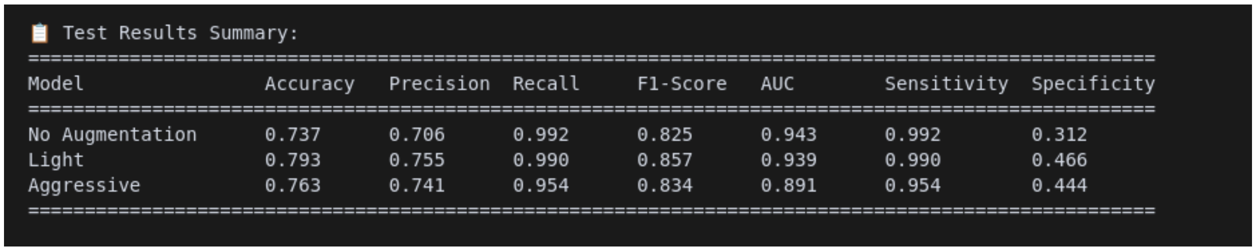
Results:

Training & Validation



Test





```
📋 Test Results Summary:
================================================================================================
Model                Accuracy   Precision  Recall     F1-Score   AUC        Sensitivity  Specificity
================================================================================================
No Augmentation      0.737      0.706      0.992      0.825      0.943      0.992        0.312
Light                0.793      0.755      0.990      0.857      0.939      0.990        0.466
Aggressive           0.763      0.741      0.954      0.834      0.891      0.954        0.444
================================================================================================
```

Experiment 2: Impact of Network Depth (CNN only)

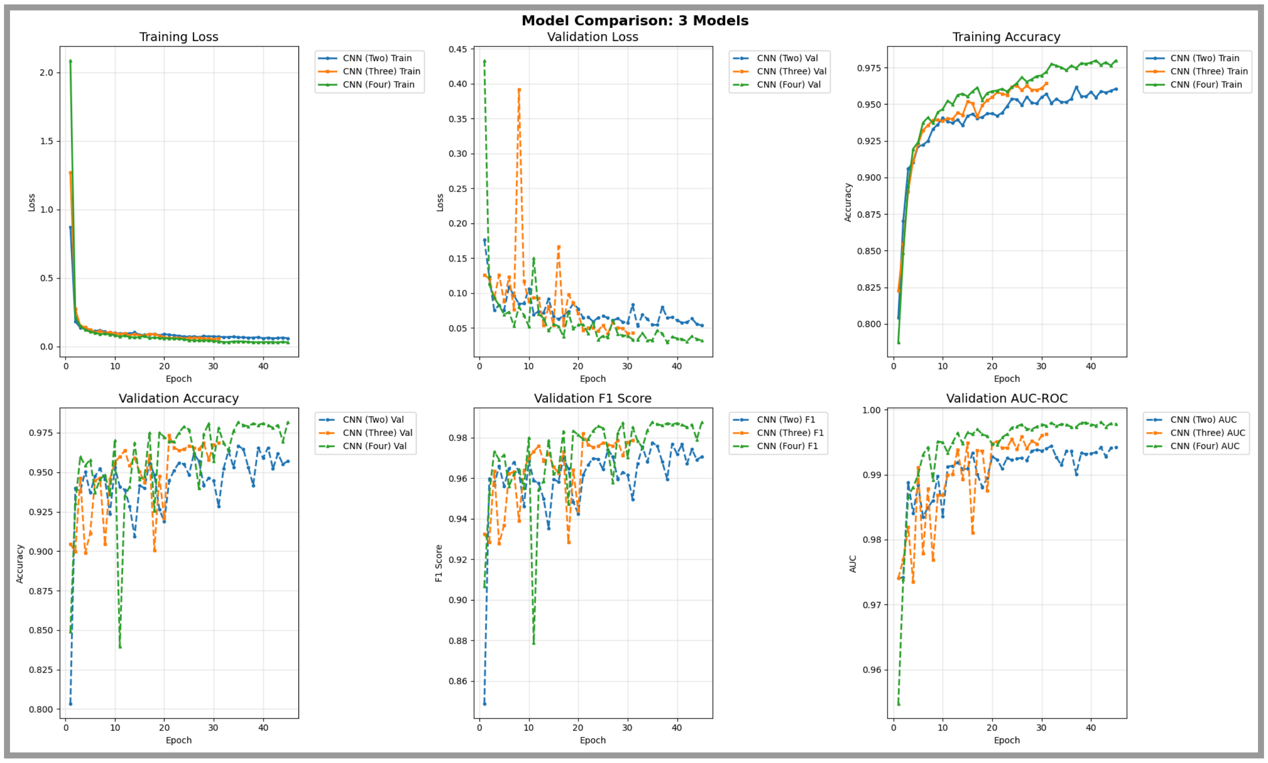
In this experiment, we used the best performing model from Experiment #1 (light augmentation) as the baseline configuration model. This experiment evaluated 3 network depths, testing with 2,3,4 convolution layers.

Based on the previous results, we wanted to test if the difference between the evaluation and test results can be (at least partially) explained by overfitting to the training examples.

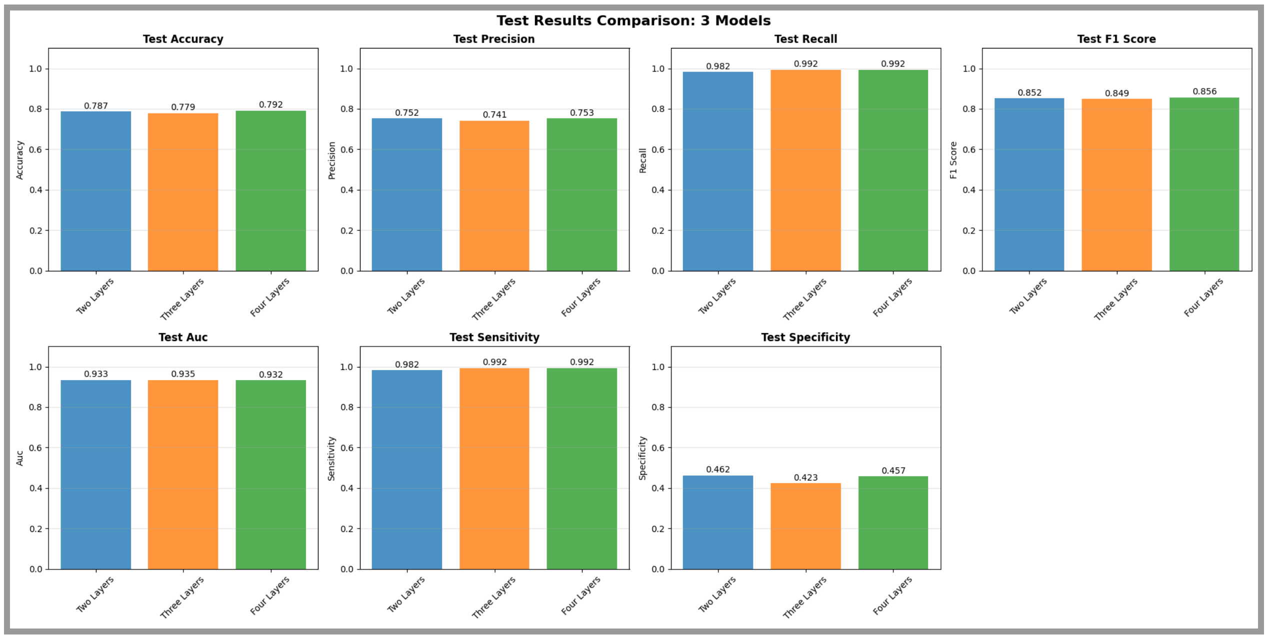
***Training time (for all 3 models, sequentially): 20 minutes 31 seconds**

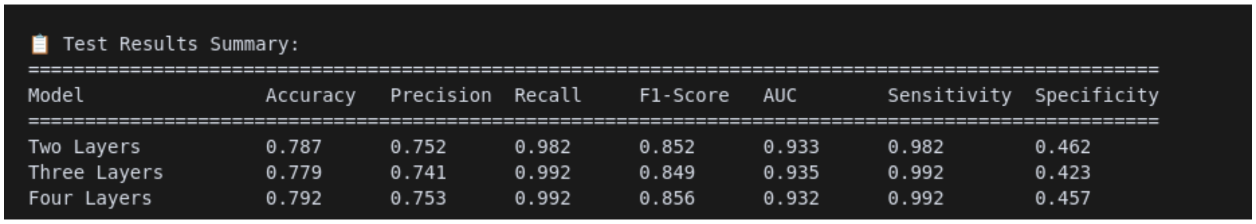
Results:

Training & Validation



Test





```
📋 Test Results Summary:
==========================================================================================
Model                Accuracy   Precision  Recall     F1-Score   AUC        Sensitivity  Specificity
==========================================================================================
Two Layers           0.787      0.752      0.982      0.852      0.933      0.982        0.462
Three Layers         0.779      0.741      0.992      0.849      0.935      0.992        0.423
Four Layers          0.792      0.753      0.992      0.856      0.932      0.992        0.457
```

Experiment 3: Impact of Freezing feature extraction layers (ViT only)

In this experiment, we evaluated the impact of the number of epochs where the feature extraction layers in the ViT model are unfroozen.

The classifier will have up to (depends on early stopping) **50 - N** epochs to learn (where N is based on the number of unfreeze epochs), and the feature extraction will be updated to additional **N** epochs.

We have tested 3 options for the number of unfreeze epochs: 1, 5, 20.

In this experiment we used the Light augmentation, from experiment #1 to all models.

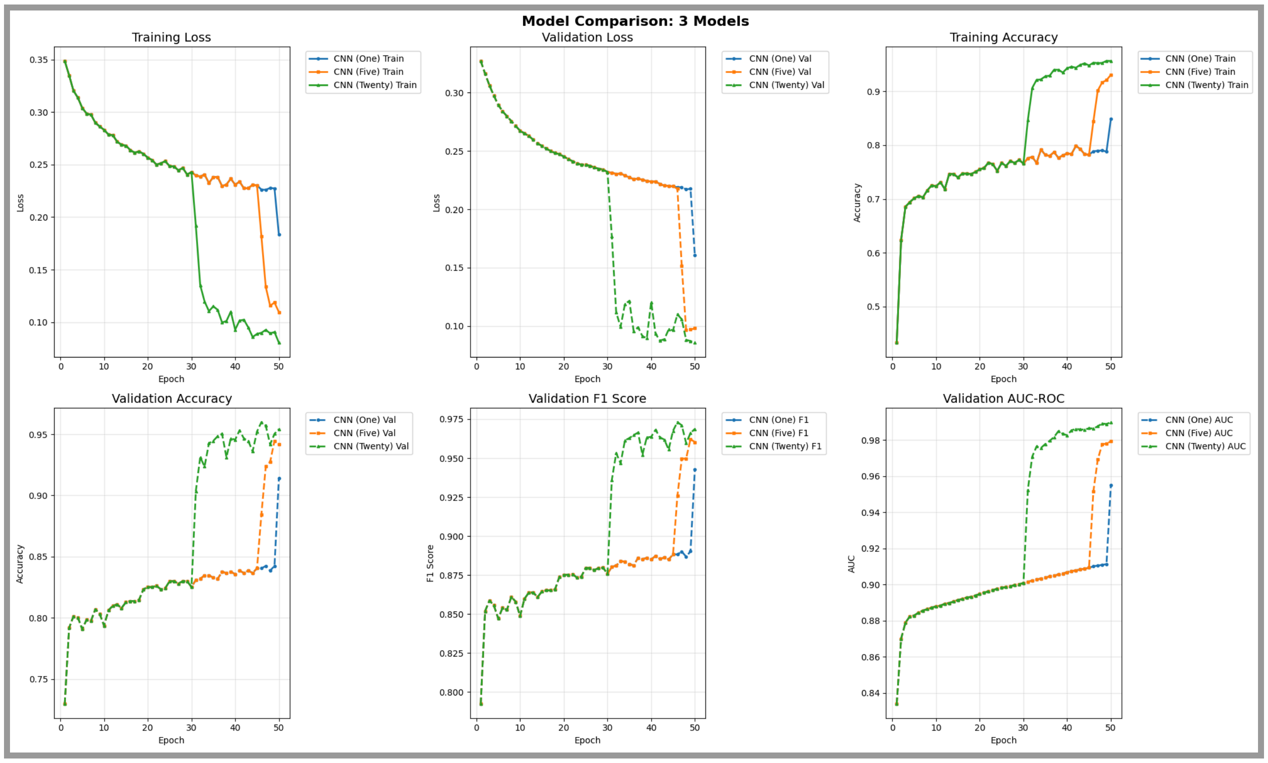
***Training times:**

1 + 5 unfreeze layers (sequentially): 58minutes 5 seconds

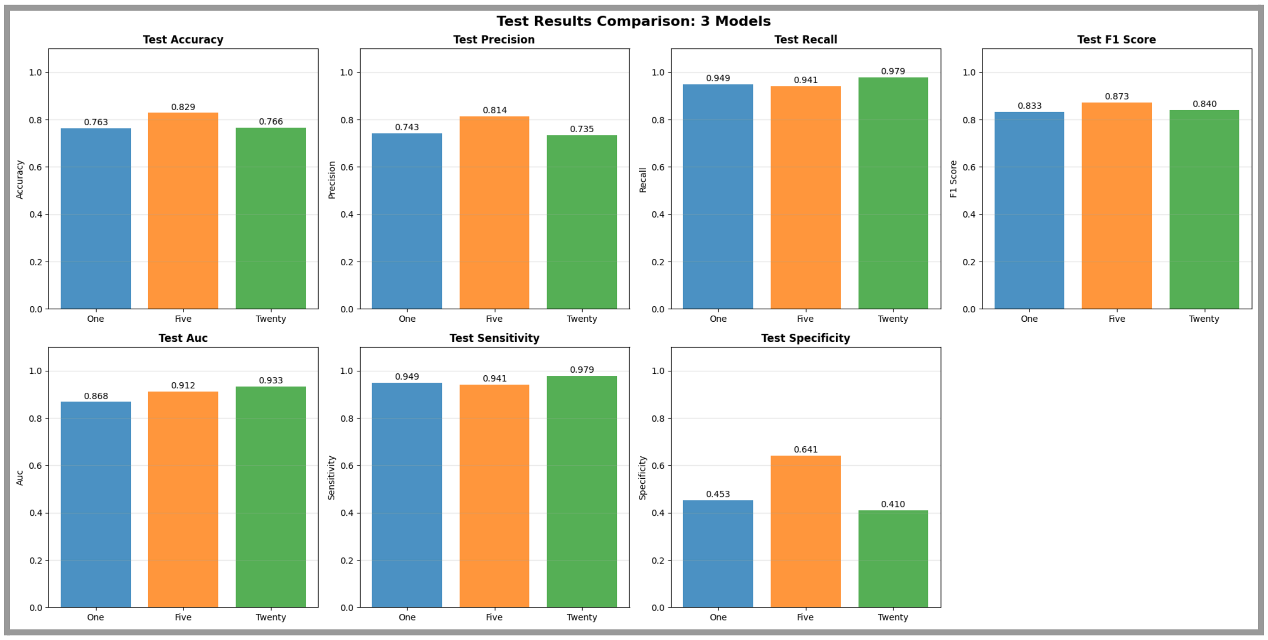
20 unfreeze layers:45 minutes, 30 seconds

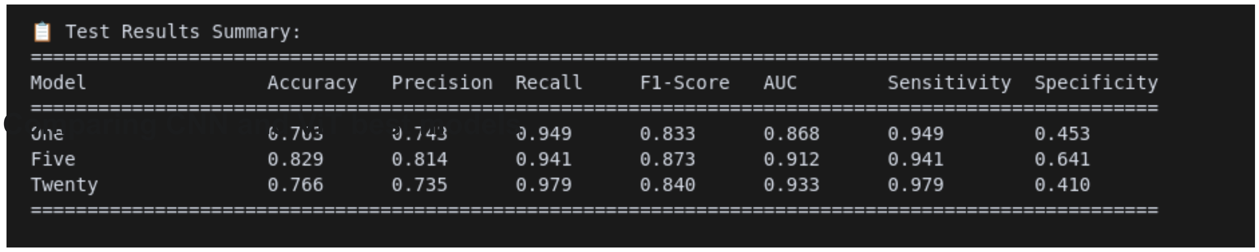
Results:

Training & Validation

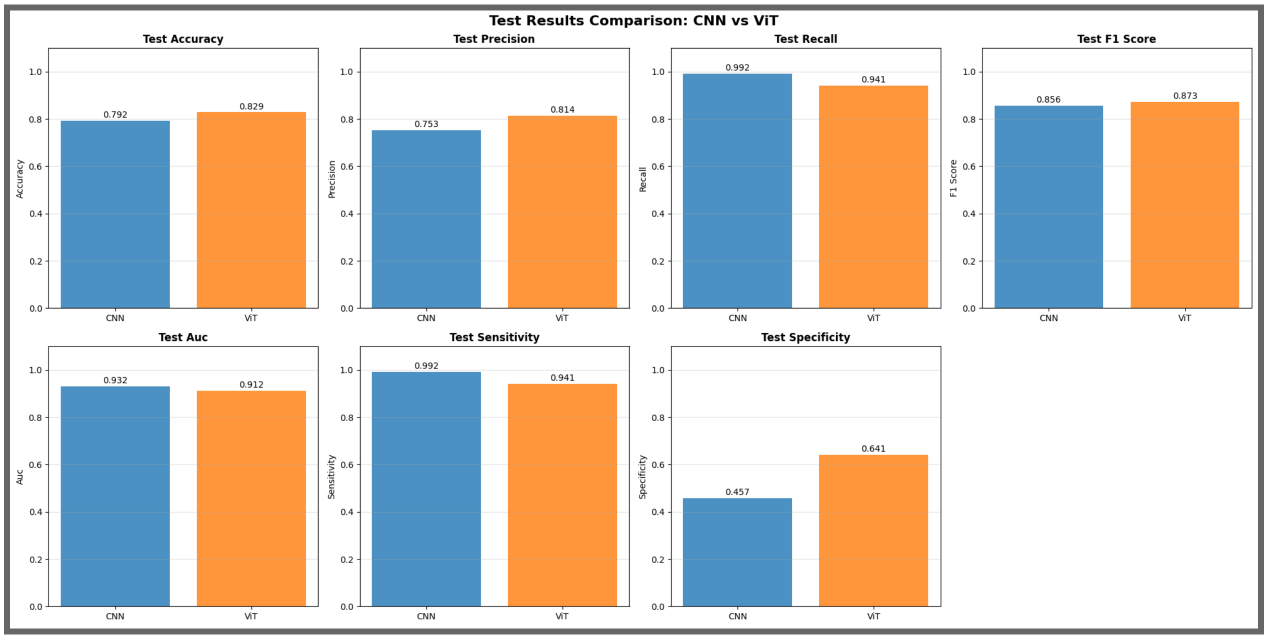


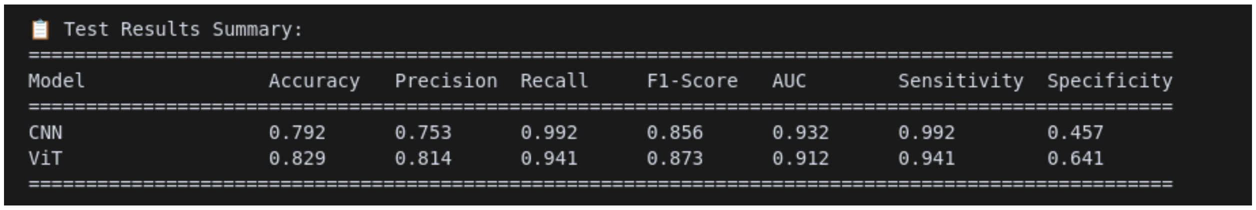
Test





```
📋 Test Results Summary:
================================================================================
Model                Accuracy   Precision  Recall     F1-Score   AUC        Sensitivity  Specificity
================================================================================
One                  0.763      0.743      0.949      0.833      0.868      0.949        0.453
Five                 0.829      0.814      0.941      0.873      0.912      0.941        0.641
Twenty               0.766      0.735      0.979      0.840      0.933      0.979        0.410
================================================================================
```





```
📋 Test Results Summary:
====================================================================================================
Model               Accuracy   Precision  Recall     F1-Score   AUC        Sensitivity  Specificity
====================================================================================================
CNN                 0.792      0.753      0.992      0.856      0.932      0.992        0.457
ViT                 0.829      0.814      0.941      0.873      0.912      0.941        0.641
====================================================================================================
```

Discussion

Data Augmentation

The comparison of different data augmentation showed interesting results. While the light augmentation showed the best results, the aggressive augmentation showed similar results to no augmentation at all.

Optional explanation for it is that while the light augmentation performed minor, realistic variation in how X-ray images are captured, and allowed a more robust dataset, the aggressive augmentation might be too far from real world scenarios and pollute the data.

It seems that this outcome underscores that effective data augmentation requires domain knowledge to ensure that generated samples remain within the distribution of real-world data.

Impact of CNN Model Depth

The performance of the 2, 3, and 4-layer CNNs was highly similar, indicating a point of diminishing returns with increased model depth.

The features necessary to distinguish between 'Normal' and 'Pneumonia' in the provided images appear to be simple enough that they can be effectively captured by a relatively shallow, 2-layer network.

Adding more layers did not provide significant new representational power for this specific task and slightly increased the risk of overfitting to the training set. This illustrates the classic trade-off between a model's capacity and its ability to generalize.

Fine-Tuning of Vision Transformers

The results from fine-tuning the ViT model highlight the balance required to adapt large, pre-trained models.

- **1 Epoch:** Unfreezing the model for only one epoch was insufficient for the powerful, pre-trained feature extraction layers to adapt to the specific nuances of chest X-ray images.
- **20 Epochs:** Fine-tuning for 20 epochs likely led to performance degradation due to overfitting or **"**catastrophic forgetting." In this scenario, the model starts to forget the general visual knowledge from its vast pre-training dataset as it over-specializes on the smaller, specific X-ray dataset, harming its ability to generalize.

The **5-epoch** fine-tuning period proved to be the optimal "sweet spot," striking the right balance between adapting to the new domain and retaining the valuable, generalized knowledge from its initial training.

Overfitting

Overfitting was identified as a primary factor limiting the model's performance, particularly manifesting as low specificity on the test set. We believe that the most significant cause was the **small and imbalanced dataset**.

A strong prevalence of 'Pneumonia' images over 'Normal' images created a biased learning environment. The model could achieve high training accuracy by simply specializing in the features of the majority 'Pneumonia' class. Due to the scarcity of 'Normal' examples, it failed to learn a robust decision boundary, leading to the misclassification of many healthy patient images during testing.

In future work, we will need to find ways to have a more balanced dataset and try better ways to overcome the Pneumonia:Normal imbalance.

A more robust data augmentation, and better class weights techniques are possible solutions for this problem.

Code: https://github.com/AssafTaubenfeld/X-ray-Classification.git