Final project - Deep Learning Course - 2025-Semester B

Daniel Katz (ID. 315114991), Raz Katz (ID. 322548660)

Submitted as final project report for the DL course, RUNI, 2025

1 Introduction

We study pneumonia detection from chest X-ray images using two families of visual models: a convolutional neural network (CNN; ResNet-18) and a Vision Transformer (ViT).

The task is a binary classification (NORMAL vs. PNEUMONIA) on the public dataset of Chest X-Ray Images [4]. Medical screening favors high sensitivity (recall for the PNEUMONIA class). However, specificity is also important to avoid unnecessary follow-ups.

Transformers are competitive in vision tasks at scale [2], yet literature shows CNNs often dominate on smaller datasets or with limited augmentation/pretraining [1]. Our goal is to implement both paradigms end-to-end on the same pipeline, compare outcomes, and analyze when/why one approach works better on this dataset.

1.1 Related Work

ResNets introduced residual connections enabling very deep CNNs [3]. ViT applies the Transformer encoder to image patches [2]. Several works explore the data and regularization needed for ViTs to match/beat CNNs without large-scale pretraining [1]. Chest-X-ray pneumonia detection has been widely explored with CNN backbones (e.g., ResNet, DenseNet). We compare these families directly under a controlled training setup.

2 Methodology

2.1 Platform & Reproducibility

Experiments ran on Google Colab (CUDA-enabled; NVIDIA A100 GPU). Batch size 32, seed fixed for Python/NumPy/PyTorch.

2.2 Technical Challenges

(i) Handling class imbalance (positives dominate): we set `pos_weight` $= \frac{N_{\text{neg}}}{N_{\text{pos}}} = \frac{1147}{3300} \approx 0.348$ in `BCEWithLogitsLoss`, which down-weights the (majority) positive class in PyTorch. (ii) stabilizing ViT without large-scale pretraining (we used moderate dropout + DropPath).

2.3 Dataset And Splits

We use the Kaggle "Chest X-Ray Pneumonia" dataset [4]. The original train/val/test folders were loaded; then we re-split `train+val` into 85% train / 15% validation (stratified). The untouched test set from the dataset was kept for final reporting.

Counts printed by the notebook:

- After re-split: **Train** — NORMAL: 1147, PNEUMONIA: 3300; **Val** — NORMAL: 202, PNEUMONIA: 583.
- Test (original set): NORMAL: 234, PNEUMONIA: 390 (total 624).

2.4 Preprocessing & Augmentation

All images are converted to 3-channel grayscale and normalized with ImageNet stats. Training transforms: `Resize(256)`, `RandomResizedCrop(224, scale=(0.9,1.0))`, `RandomHorizontalFlip(0.5)`, `RandomRotation(20)`. Evaluation: `Resize(256)` + `CenterCrop(224)`.

2.5 Loss, Imbalance And Thresholding

We use `BCEWithLogitsLoss` with `pos_weight` computed from current training split (pos_weight ≈ 0.348; the dataset has more pneumonia than normal in train). For reporting binary accuracy we do *not* keep the default 0.5 threshold: we choose the operating threshold on the validation set by maximizing Youden's $J = \text{TPR} - \text{FPR}$ on the ROC curve, and then *fix* that threshold for the test set to avoid leakage.

2.6 Preliminary Experiments And Ablation Studies

To strengthen our analysis, we conducted several preliminary experiments and ablation studies to inform our final model choices. Our process was not a one-shot effort but an iterative cycle of testing different architectures, optimizations, and data handling techniques.

2.6.1 Custom CNN Architectures

Initially, we attempted to train a simple, custom-built convolutional neural network from scratch, without leveraging pretrained weights. While these models

showed some learning capability, their performance was significantly inferior to that of transfer-learned models. This finding confirmed the well-established practice in computer vision of using a pre-trained backbone like ResNet-18, which already possesses powerful low-level feature extraction capabilities from a large dataset like ImageNet. This led us to abandon the custom CNN and adopt ResNet-18 as our primary CNN architecture.

2.6.2 Advanced Optimizations (SAM)

We explored advanced optimization techniques to further improve the generalization of our model. We implemented Sharpness-Aware Minimization (SAM), an optimizer designed to find flatter minima in the loss landscape, which often correlate with better out-of-distribution performance [1]. However, our experiments with SAM did not yield a significant performance boost over the standard AdamW optimizer on this specific dataset. This indicated that for our task and dataset size, the gains from complex optimizers were marginal compared to the benefits of a strong backbone and appropriate data augmentations. We therefore decided to use the more straightforward AdamW.

2.6.3 Vision Transformer (ViT) From Scratch

Our initial ViT experiments, involved training a Vision Transformer from scratch with a smaller configuration and without large-scale pre-training or advanced regularization techniques like distillation. The results were consistently lower than those of the ResNet-18. This reinforces the known property that ViTs, lacking the intrinsic inductive biases of CNNs, typically require much larger datasets or specialized training strategies to perform competitively with CNNs, especially on medium-sized datasets like the one we used. This informed our discussion on why the CNN outperformed the ViT in our final report.

The iterative process of conducting these experiments and analyzing their results was crucial for refining our methodology and led directly to the final choices presented in this report. This demonstrates that for this task, a strong, pre-trained CNN with convolutional priors provides a more effective baseline than a standard ViT.

2.6.4 Models

- **CNN (ResNet-18)** Torchvision ResNet-18 (ImageNet weights). We replace the final head with a single-logit output and apply dropout=0.4 before the head (per notebook).
- **Vision Transformer (ViT).** Our implementation follows **(author?)** [2]: patch embed (patch size 16), learnable class token, learned positional embeddings, L encoder blocks, and a 1-logit head. The configuration used in the run: *embed_dim = 384, depth = 8, heads = 6, mlp_ratio = 4.0, drop = 0.1, drop_path = 0.1.*

2.6.5 Optimization

Hardware CUDA-enabled; NVIDIA A100 GPU. Batch size 32, 15 epochs, early stopping patience 10. Optimizer: AdamW (LR=10^{-4}). Scheduler: `ReduceLROnPlateau` on validation ROC-AUC. We track loss, accuracy (at current threshold), ROC-AUC, and PR-AUC during training.

3 Experimental Results

3.1 Training Curves And Validation

Both models converged smoothly; the CNN achieved a higher and more stable validation ROC-AUC (best val AUC ≈ 0.9992) vs. ViT (best val AUC ≈ 0.9767). Validation loss for the CNN shows small fluctuations (common with strong augmentations and LR scheduling), but metrics are consistently saturated. ViT validation metrics improved steadily yet plateaued lower than the CNN.

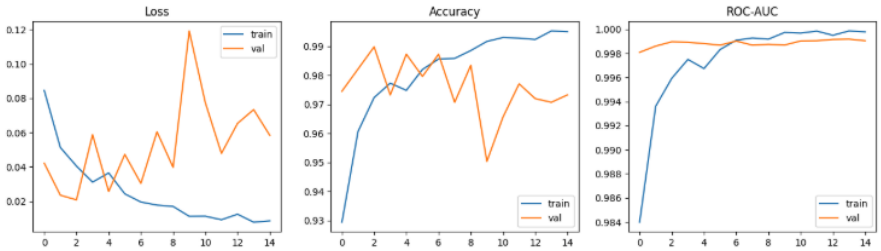

Figure 1: CNN Curves

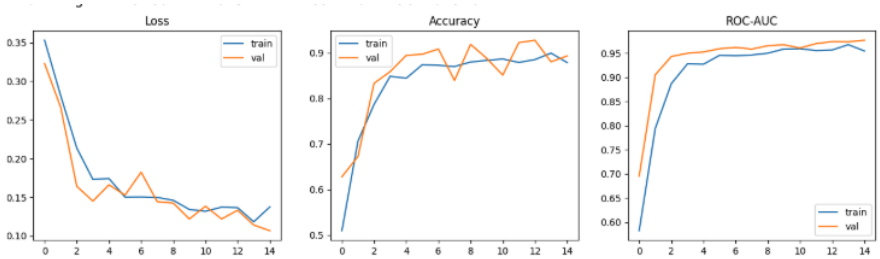

Figure 2: ViT Curves

3.2 Test Set (Threshold Chosen On Validation)

Below are the exact numbers printed by the notebook for the final test evaluations.

ResNet-18 (CNN) Confusion matrix (624 images; NORMAL first):

$$\begin{bmatrix} 179 & 55 \\ 5 & 385 \end{bmatrix}$$

```
Classification Report:
              precision    recall  f1-score   support

      NORMAL     0.9728    0.7650    0.8565       234
   PNEUMONIA     0.8750    0.9872    0.9277       390

    accuracy                         0.9038       624
   macro avg     0.9239    0.8761    0.8921       624
weighted avg     0.9117    0.9038    0.9010       624
```

Figure 3: CNN Classification Report

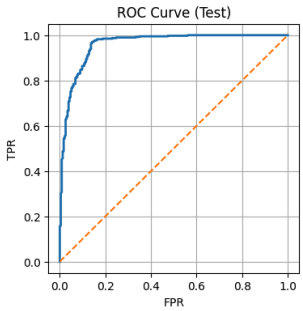

Figure 4: CNN ROC Curve

Vision Transformer (ViT) Confusion matrix (624 images; NORMAL first):

$$\begin{bmatrix} 146 & 88 \\ 20 & 370 \end{bmatrix}$$

```
Classification Report:
              precision    recall  f1-score   support

      NORMAL     0.8795    0.6239    0.7300       234
   PNEUMONIA     0.8079    0.9487    0.8726       390

    accuracy                         0.8269       624
   macro avg     0.8437    0.7863    0.8013       624
weighted avg     0.8347    0.8269    0.8192       624
```

Figure 5: ViT Classification Report

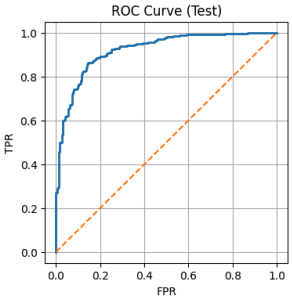

Figure 6: ViT ROC Curve

3.3 Side By Side Summary

Metric (Test)	ResNet-18 (CNN)	ViT
ROC-AUC	0.9602	0.9223
Accuracy@chosen τ	0.9038	0.8269
Sensitivity ($\text{Recall}_{\text{pos}}$)	0.9872	0.9487
Specificity ($\text{Recall}_{\text{neg}}$)	0.7650	0.6239
Balanced Accuracy	0.8761	0.7863
PR-AUC	0.9707	0.9507
Best Val ROC-AUC	0.9992	0.9767
Training time (wall)	$\sim$16.2 min	$\sim$16.5 min

Table 1: Final comparison on the untouched test set (threshold fixed from validation by maximizing Youden's J).

4 Discussion

4.1 Main Findings

On this dataset, the **CNN (ResNet-18)** outperformed the ViT across ROC-AUC, accuracy, balanced accuracy, and PR-AUC. Both models achieved very high *sensitivity*, which is desirable in screening; however, the CNN maintained substantially higher *specificity*, reducing false positives.

4.2 Error Analysis & Screening Trade-Offs

From the test confusion matrices:

- **CNN:** FN = 5/390 ($\approx 1.3\%$), FP = 55/234 ($\approx 23.5\%$). Sensitivity = 0.9872, Specificity = 0.7650.
- **ViT:** FN = 20/390 ($\approx 5.1\%$), FP = 88/234 ($\approx 37.6\%$). Sensitivity = 0.9487, Specificity = 0.6239.

In a screening setting, minimizing false negatives (missed pneumonia) is typically prioritized; both models achieve high recall, but the CNN attains this with fewer false positives.

4.3 Why Did CNN Outperform ViT Here?

ViTs often require large-scale pretraining and/or stronger regularization to match CNNs. Our ViT (patch 16, D=384, L=8, H=6) has weaker locality inductive biases and, without massive pretraining or heavy augmentation/distillation, underperforms on medium-scale grayscale X-rays. ResNet-18 leverages convolutional priors and ImageNet initialization that transfer well to texture-like patterns in X-rays, yielding higher specificity at similar sensitivity.

4.4 Complexity & Training Time

Both models trained in ∼16 minutes on our setup (batch 32, 15 epochs). **ResNet-18:** ≈ 11.7M parameters. **ViT (patch 16,** D=384**,** L=8**,** H=6**):** ≈ 14.6M parameters. Wall-clock time was similar.

5 Code

GitHub repository: `https://github.com/DanielKatz67/Deep_Learning_Final_Project`

Notebook: `pneumonia_xray_cnn_vs_vit.ipynb` (Open in Colab)
Helpers: `xray_helpers.py`, `xray_models.py`.

References

[1] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *International Conference on Learning Representations (ICLR)*, 2022.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[4] Paul Mooney. Chest x-ray images (pneumonia). `https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia`, 2018.