Pneumonia Classification using ResNet-34

Abstract

This report presents a comparative study on pneumonia classification using the ResNet-34 deep learning architecture. Two training approaches are evaluated: training from scratch and fine-tuning a pre-trained ResNet-34 (ImageNet weights). Data preprocessing, augmentation techniques, training details, and results with extensive visualization are discussed. The study highlights the benefits of transfer learning for improved performance and convergence.

Contents

1 Dataset Preprocessing and Visualization

1.1 Data Splitting and Augmentation

Train/Validation:

- Severe class imbalance handled using Stratified k-fold (k=5) and WeightedRandom-Sampler.

Data Augmentation:

- Random rotation (between -10 and 10 degrees)
- Random horizontal flip
- Resizing and padding to 224x224
- Normalization using mean and standard deviation calculated on the training dataset

1.2 Visualization

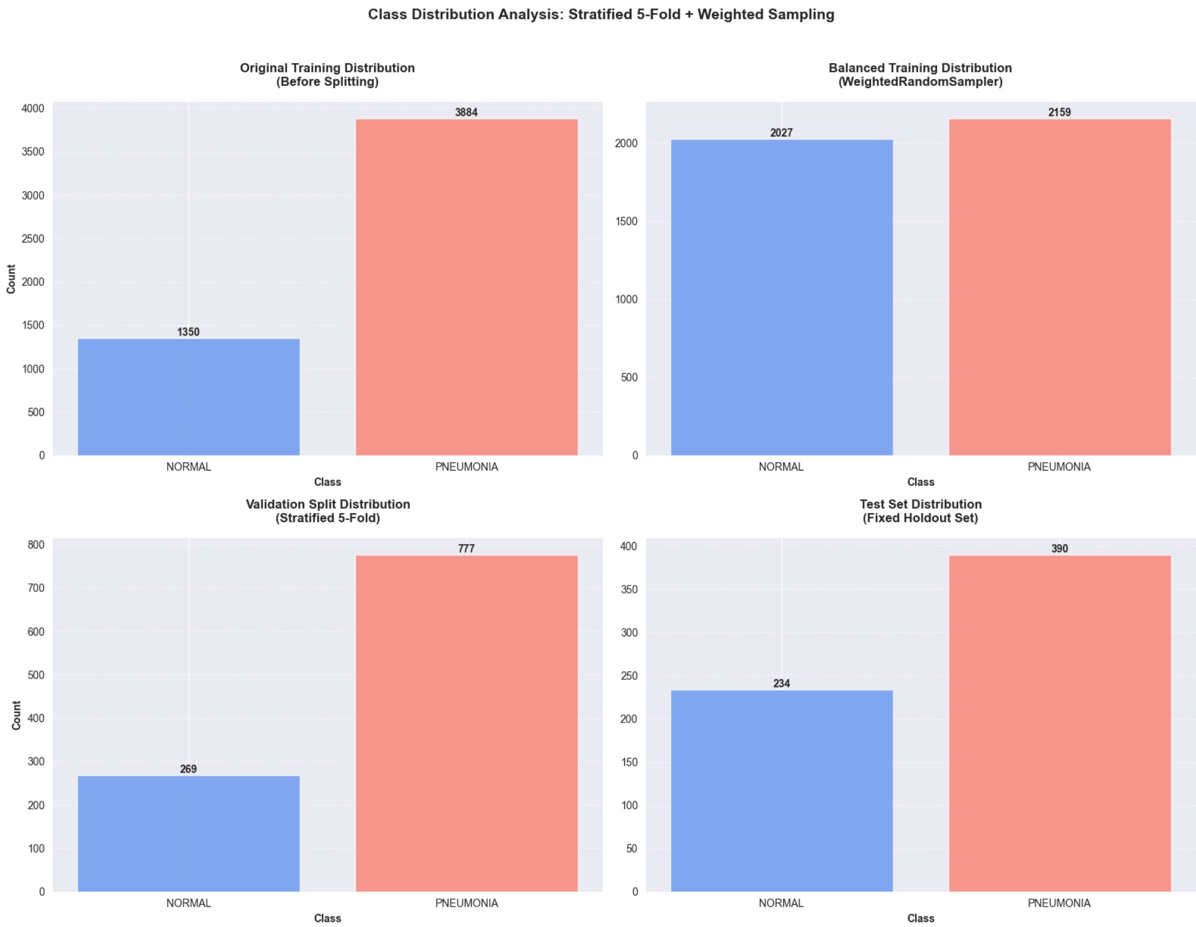

Figure 1: Class Distribution Analysis across Train, Validation and Test

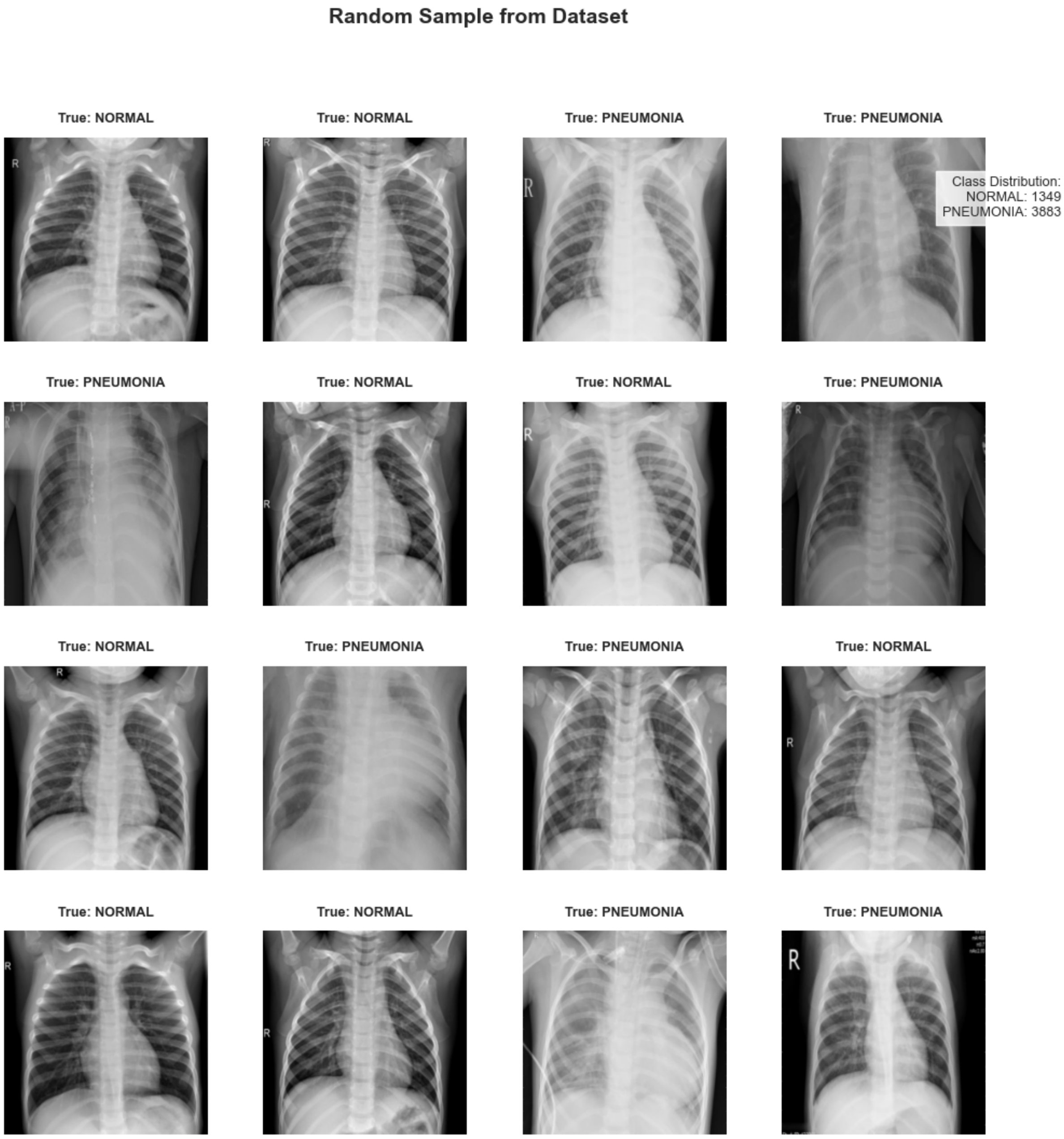

Figure 2: Random Image Visualization

2 Task 1.1 – Training ResNet-34 from Scratch

2.1 Implementation Details

Network Architecture:

- Model: ResNet-34
- Modification: Last fully connected layer adjusted for binary classification.
- Dropout: $p = 0.3$ added in the last fully connected layer.
- Weight Initialization: Testing Xavier Uniform initialization for randomness.

Hyperparameters:

- Batch Size: 32
- Image Size: 224x224
- Input Channels: 3
- Number of Classes: 2 (PNEUMONIA, NORMAL)
- Learning Rate: 0.001
- Weight Decay: 1×10^{-4}
- Epochs: 10

Optimizer and Loss Function:

- Optimizer: AdamW
- Loss Function: CrossEntropyLoss
- Scheduler: CosineAnnealingLR (with `tmax = EPOCHS` and `etamin = 0.0001`)

2.2 Results and Visualization

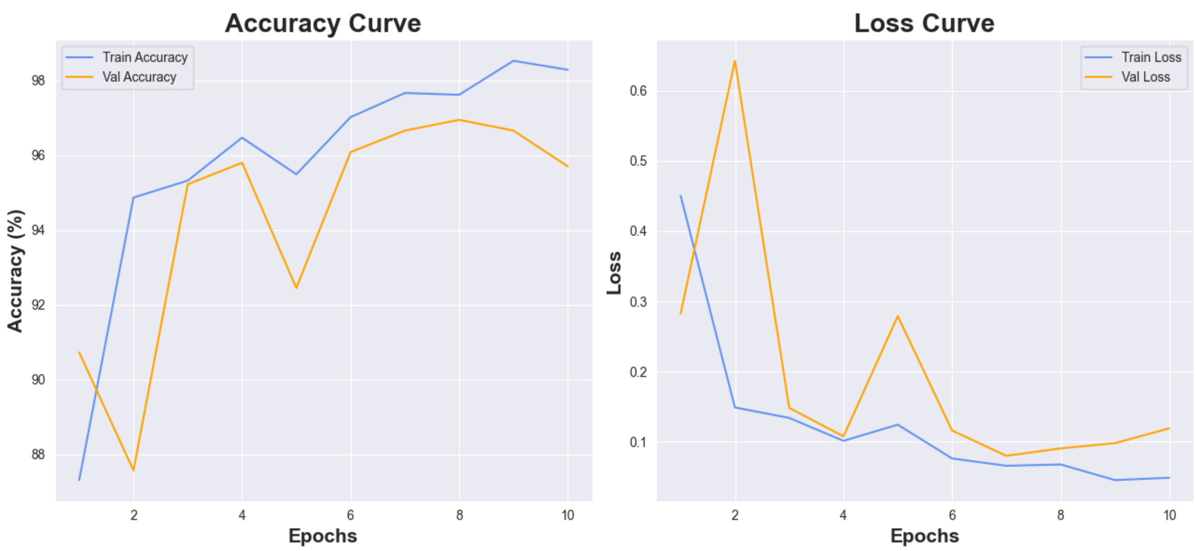

Figure 3: Training and Validation Curves (Task 1.1)

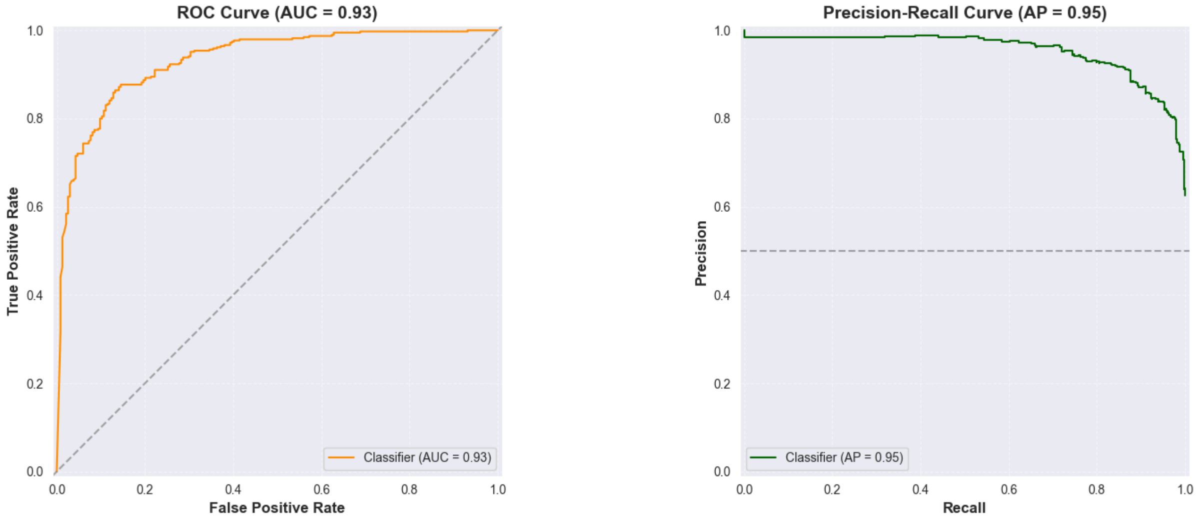

Figure 4: Recall and Precision Curves (Task 1.1)

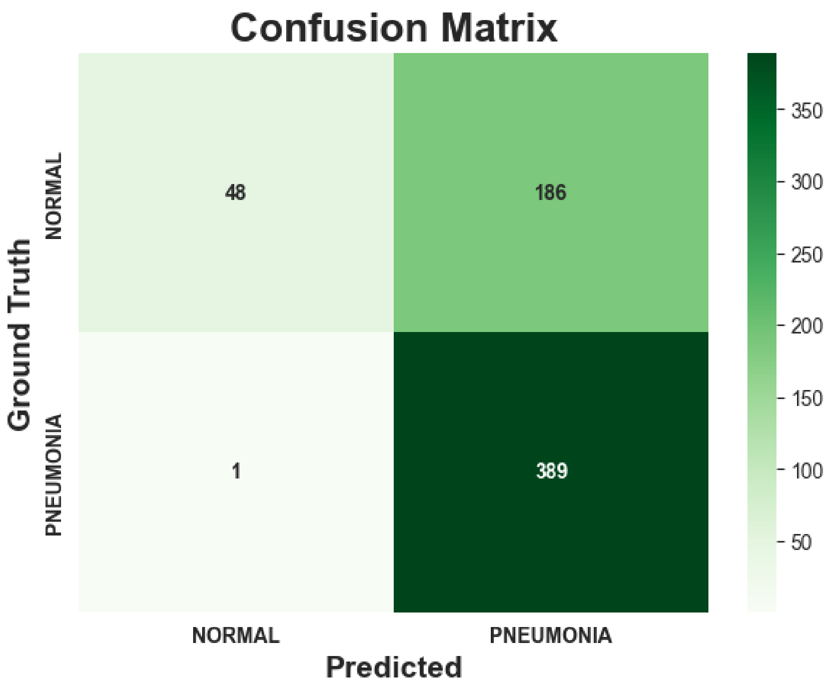

Figure 5: Confusion Matrix (Task 1.1)

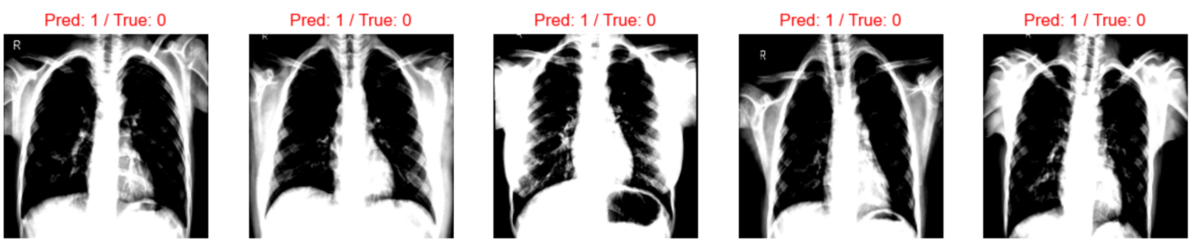

Figure 6: Predictions Overview (Task 1.1)

Note: A total of 28 images were misclassified in Task 1.1.

Figure 7: Examples of Misclassified Images (Task 1.1)

3 Task 1.2 – Fine-Tuning ResNet-34 Pre-Trained on ImageNet

3.1 Implementation Details

Network Architecture:

- Model: ResNet-34 pre-trained on ImageNet
- Model: Fine-Tuning enabled for all layers.
- Modification: Last fully connected layer adjusted for binary classification.
- Dropout: $p = 0.3$ added in the last fully connected layer.

Hyperparameters:

- Batch Size: 32
- Image Size: 224x224
- Input Channels: 3
- Number of Classes: 2 (PNEUMONIA, NORAML)
- Learning Rate: 0.001
- Weight Decay: 1×10^{-4}
- Epochs: 10

Optimizer and Loss Function:

- Optimizer: AdamW
- Loss Function: CrossEntropyLoss
- Scheduler: CosineAnnealingLR (with `tmax = EPOCHS` and `etamin = 0.0001`)

3.2 Results and Visualization

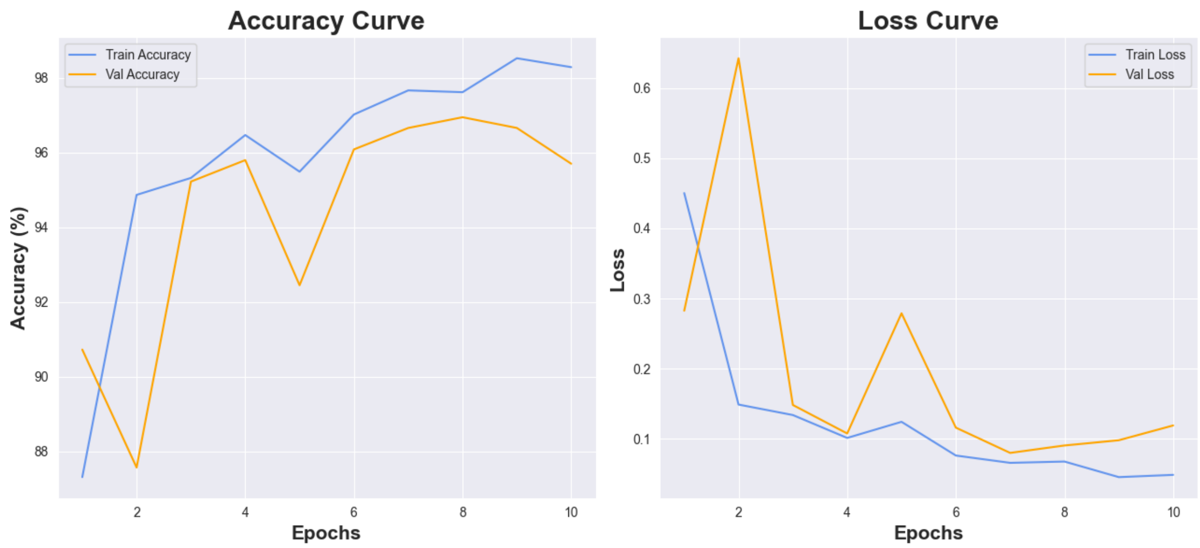

Figure 8: Training and Validation Curves (Task 1.2)

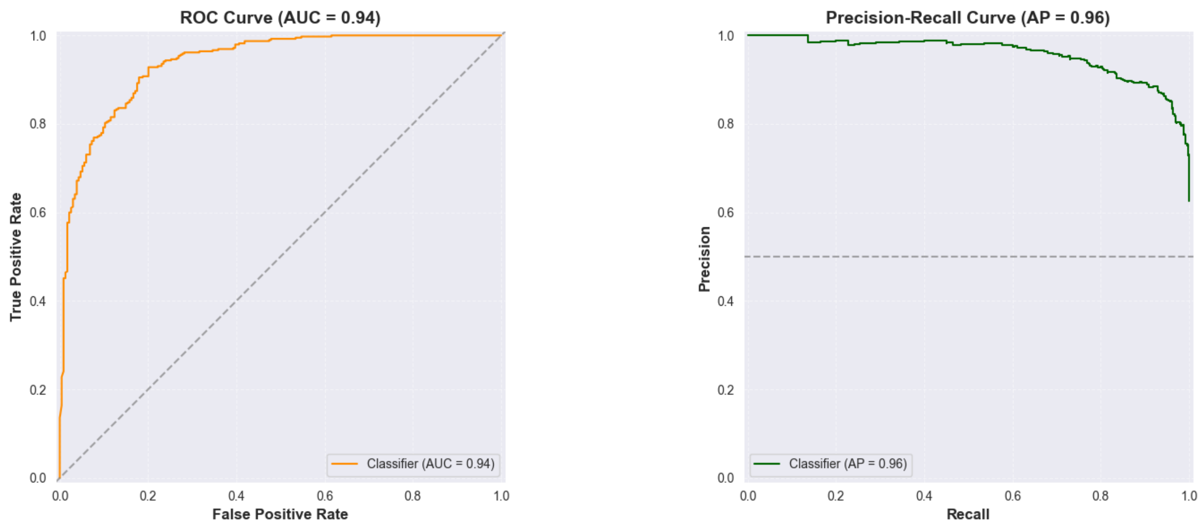

Figure 9: Recall and Precision Curves (Task 1.2)

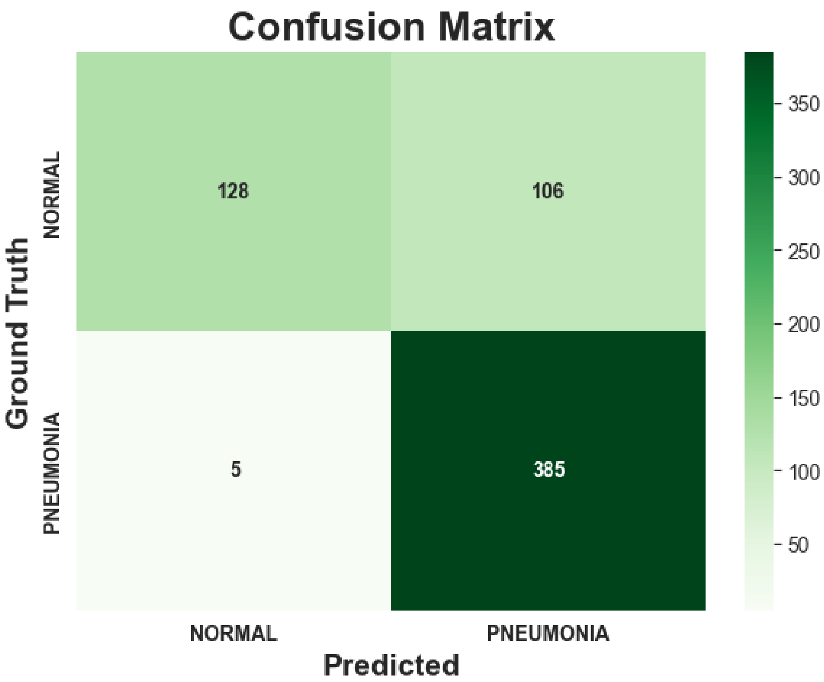

Figure 10: Confusion Matrix (Task 1.2)

Figure 11: Predictions Overview (Task 1.2)

Note: A total of 15 images were misclassified in Task 1.2.

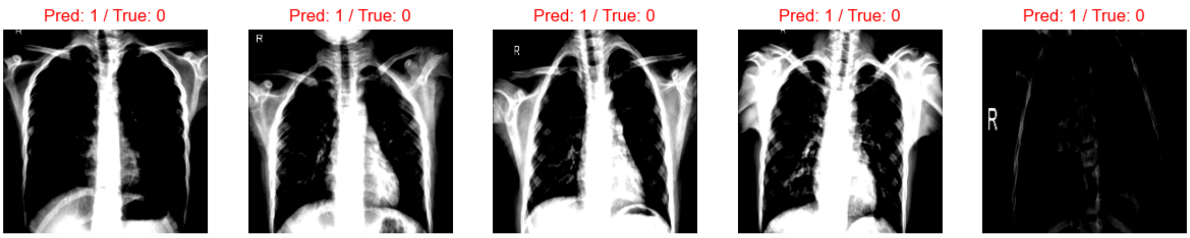

Figure 12: Examples of Misclassified Images (Task 1.2)

4 Comparisons and Conclusion

The two experiments were conducted using similar model configurations, differing only in the initialization of the model architecture. In the first experiment, the model was trained from scratch (with random initialization) whereas in the second experiment, the model was initialized with pretrained ImageNet weights.

Key observations include:

- **Test Accuracy:** The pretrained model achieved a test accuracy of 82%, compared to 70% for the non-pretrained model.
- **Class-wise Performance:**
  - The pretrained model reported a precision of 0.96 and a recall of 0.55 for the *NORMAL* class, and a precision of 0.78 with an excellent recall of 0.99 for the *PNEUMONIA* class.
  - The non-pretrained model showed a high precision of 0.98 for the *NORMAL* class but a very low recall of 0.21, while the *PNEUMONIA* class had a precision of 0.68 and a perfect recall of 1.00. This indicates that although both models are able to detect pneumonia effectively, the non-pretrained model misclassifies a considerably higher number of normal cases.
- **Validation Metrics:** Both models reach comparable training and validation accuracies; however, the best validation loss and accuracy indicate that the pretrained model converged to a slightly better optimum.

Classification Reports:

Task 1: Model with Random Weights

Class	Precision	Recall	F1-Score	Support
NORMAL	0.98	0.21	0.34	234
PNEUMONIA	0.68	1.00	0.81	390
Accuracy			0.70	624
Macro Avg	0.83	0.60	0.57	624
Weighted Avg	0.79	0.70	0.63	624

Table 1: Classification Report for Task 1

Task 2: Model with Pretrained Weights

Class	Precision	Recall	F1-Score	Support
NORMAL	0.96	0.55	0.70	234
PNEUMONIA	0.78	0.99	0.87	390
Accuracy			0.82	624
Macro Avg	0.87	0.77	0.79	624
Weighted Avg	0.85	0.82	0.81	624

Table 2: Classification Report for Task 2

Comparison Table:

Metric	ResNet (Pretrained on ImageNet)	ResNet (Non-pretrained)
Training Accuracy	98.85%	98.28%
Validation Accuracy	96.08%	95.70%
Test Accuracy	82%	70%
Best Validation Loss	0.0718	0.0800
Best Validation Accuracy	97.61%	95.79%

Table 3: Performance Comparison: Pretrained vs. Non-pretrained ResNet Architectures

Conclusion:

Fine-tuning a ResNet-34 model initialized with ImageNet pretrained weights leads to a significant improvement in test accuracy (82% vs. 70%) and a better balance in class compared to training from scratch. Despite both models achieving high training and validation accuracies, the pretrained model's improved convergence and enhanced ability to correctly classify normal cases shows the benefit of transfer learning for pneumonia classification. Upon pondering about on how to improve the performance for this task, I propose adding an attention mechanism like SE block and fine-tuning the hyperparams. Also, random spike were noticed in the Validation Accuracy and Loss which indicates that the model is not generalizing well and might be because of the nature of Data Processing or the Learning Rate and Scheduler which have to be examined further.