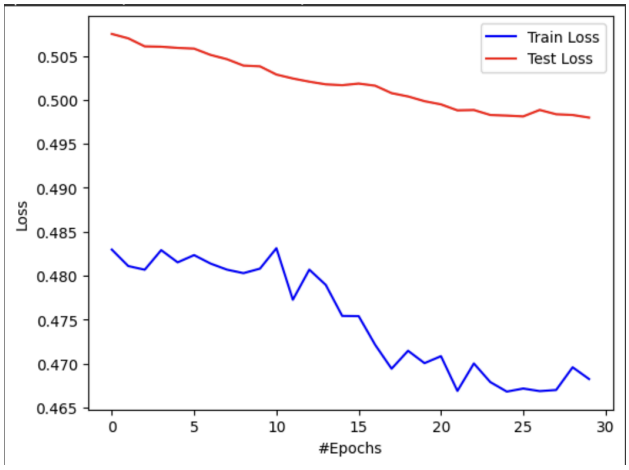
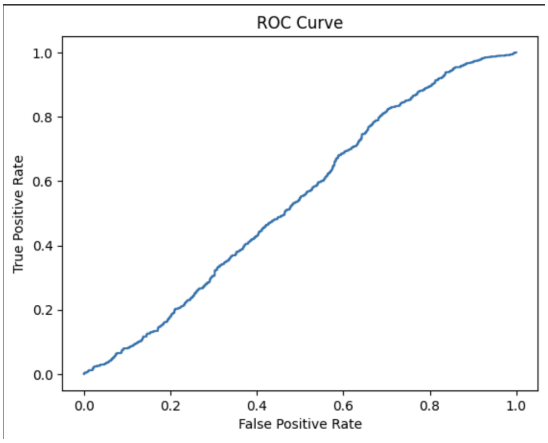
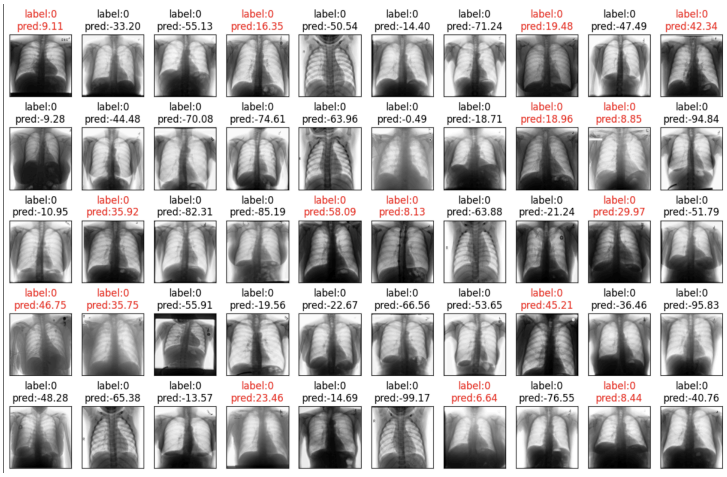
CSci 4521 Homework 4 Writeup
Binary Classification







In training our model to classify lung X-ray images into healthy and unhealthy categories, we adopted a binary classification approach to simplify the problem and ensure class balance. The data consisted of four classes: normal (healthy), COVID, pneumonia, and other infections (unhealthy). We first converted the multiclass labels into binary labels (healthy: 0 vs. unhealthy: 1) to focus on distinguishing between overall lung health conditions rather than specific diseases. This was done by treating all non-normal images as unhealthy. To avoid bias from class imbalance, we undersample the majority class to match the size of the minority class, resulting in a balanced dataset that we can then split into training and testing sets.

Our model, a linear classifier (linear regression), was trained over 30 epochs using stochastic gradient descent (SGD) with a very low learning rate of 0.00000001. This was initially selected to prevent large updates due to data scaling, but it likely contributed to underfitting. While the training and test loss decreased over epochs, the final performance metrics suggest that the model struggled to capture meaningful patterns. The accuracy was approximately 49.3% on the training set and 52.3% on the test set, which was barely above random chance. The Area Under the Curve (AUC) for the ROC curve was 0.54, which is just slightly better than random guessing, further confirming poor predictive power.

```
Loaded Data
Epoch [1/30], Train Loss: 0.4830, Test Loss: 0.5075
Epoch [6/30], Train Loss: 0.4823, Test Loss: 0.5058
Epoch [11/30], Train Loss: 0.4831, Test Loss: 0.5029
Epoch [16/30], Train Loss: 0.4754, Test Loss: 0.5018
Epoch [21/30], Train Loss: 0.4708, Test Loss: 0.4995
Epoch [26/30], Train Loss: 0.4672, Test Loss: 0.4981
```

```
Test Set: Accuracy: 281/537 (52.3%)

Train Set: Accuracy: 1061/2153 (49.3%)
```

The precision, recall, and F1 score metrics provide a deeper insight into the model's performance. Precision for the healthy class was 49.13%, and for the unhealthy class, it was 49.72%. This shows that the model was roughly equally accurate at predicting both classes but still left room for improvement in predicting true positive cases. Recall for the healthy class was 74.53%, indicating that the model was relatively good at identifying healthy lungs. However, recall for the unhealthy class was only 24.61%, highlighting a significant issue in the model's ability to identify unhealthy lungs. This suggests that the model struggles to detect unhealthy lung conditions effectively, particularly pneumonia, COVID-19, and other infections. The F1-score, which balances precision and recall, was 0.59 for the healthy class and 0.33 for the unhealthy class. The F1 disparity between healthy and unhealthy classes underlines the unreliability of predictions for unhealthy lungs, despite balanced input data. The count of samples in each class was approximately 1064 healthy images and 1089 unhealthy images, showing a fairly balanced dataset, yet the model's performance was still significantly skewed. The model achieved decent recall for the healthy class, but the substantial drop in recall for the unhealthy class indicates a need for improvement in classifying unhealthy lung conditions.

```
AUC: 0.5404074925606026
Precision [0.4913259  0.49721707]
Recall [0.74530075 0.24609734]
F1 [0.59223301 0.32923833]
Count [1064 1089]
```

Other Binary Classifiers

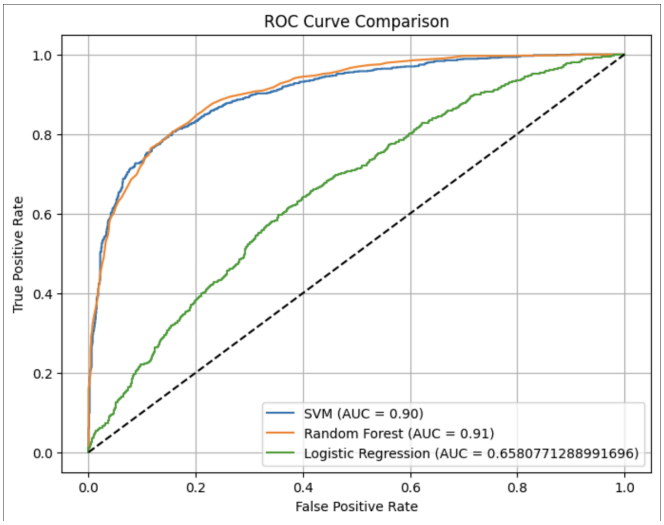
```
SVM Evaluation:
Accuracy: 0.8216
F1 Score: 0.8222
ROC AUC: 0.9024
```

```
Random Forest Evaluation:
Accuracy: 0.8221
F1 Score: 0.8181
ROC AUC: 0.9076
```

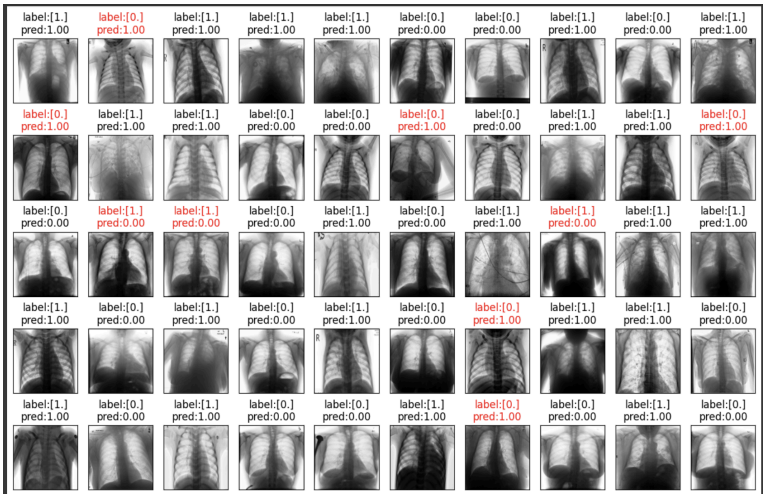
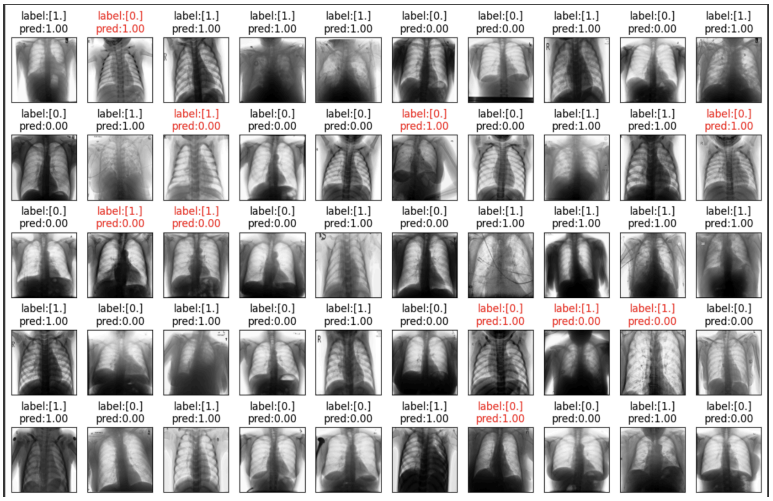
Our baseline model was logistic regression, trained using stochastic gradient descent with a very small learning rate of 0.00000001 over 30 epochs. Although both training and test losses decreased slightly, the model's performance was suboptimal. It achieved an accuracy of only 52.3%, an ROC AUC of 0.54, and F1 scores of 0.59 for healthy lungs and 0.33 for unhealthy lungs. The recall values further highlighted a critical imbalance in the model's predictive ability—it recalled 74.5% of healthy cases but only 24.6% of unhealthy cases. This indicates a strong bias toward predicting healthy lungs, leading to poor sensitivity in identifying diseases such as COVID-19 and pneumonia. The ROC curve confirmed this, with the logistic regression line hugging the diagonal, which reflects near-random performance.

To improve upon this, we implemented two more sophisticated models: Support Vector Machine (SVM) and Random Forest (RF), both using the Scikit-learn library. SVM's margin-based optimization and ability to handle high-dimensional, nonlinear boundaries likely contribute to its superior performance in separating subtle differences, and Random Forest benefits from ensemble averaging, making it robust to mislabeled examples and noise. These

models performed significantly better across all metrics. SVM achieved an accuracy of 82.16%, an F1 score of 0.8222, and an ROC AUC of 0.9024. Random Forest closely followed with an accuracy of 82.21%, an F1 score of 0.8181, and an ROC AUC of 0.9075. Both models' ROC curves stayed well above the diagonal and close to the top-left corner of the plot, indicating excellent sensitivity and specificity in classifying healthy versus unhealthy lungs. Compared to logistic regression, SVM and Random Forest demonstrated a dramatic improvement in both recall and balanced classification of the two categories. The ROC curve graph provided further reinforces the superiority of the Scikit-learn models—Support Vector Machine (SVM) and Random Forest—over logistic regression in the binary classification of lung health. The nearly overlapping curves of SVM and Random Forest show that both models perform consistently well, though the SVM curve appears to edge slightly higher, aligning with its marginally better metrics (accuracy and AUC). On the other hand, the logistic regression curve's proximity to the diagonal clearly indicates its ineffectiveness in identifying unhealthy lungs, as seen in its recall and F1 scores. Thus, this visual comparison strongly supports the conclusion that SVM is the most effective model for this task, followed closely by Random Forest, while logistic regression underperforms in every respect (for additional information on predictions, see table with plotted predictions of SVM and Random Forest).



Overall, while logistic regression provided a foundational starting point, it failed to properly identify unhealthy lungs, rendering it unsuitable for use in this context. SVM slightly outperformed Random Forest between the two improved models, particularly in ROC AUC and accuracy. Due to its ability to handle high-dimensional data and capture complex, nonlinear patterns, SVM emerges as the most effective approach for this binary classification task and is recommended for real-world implementation.

SVM:	
Random Forest:	

Specific Diagnosis Classification

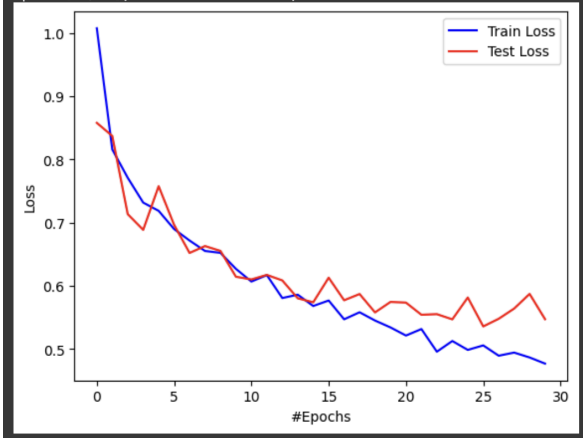
The neural network model we designed, named Classifier_4Layers, is a fully connected feedforward neural network built for a multi-class image classification task. The model processes 128x128 grayscale images, resulting in an input size of 16,384 features per sample after flattening. It consists of six linear layers with progressively decreasing dimensionality: 128, 64, 32, 16, 8, and finally, the number of output classes (4 in this case). The activation functions used are primarily ReLU, with a few layers using the absolute value (torch.abs) as a non-standard alternative.

The model was trained using a batch size of 32 and a small learning rate of 0.00001 for 30 epochs, using the Adam optimizer and CrossEntropyLoss as the loss function. These choices are generally appropriate for stable and gradual learning, particularly for deep networks. The dataset was split 80/20 for training and testing, and one-hot encoding was applied to the labels to match the output format.

The training process was successful, as indicated by the steady decrease in both training and testing loss over time. The plotted loss curves show that while the training loss consistently declines, the testing loss also follows a similar trend with some minor fluctuations, indicating good generalization and minimal overfitting. The final evaluation reported a test accuracy of 82.0% (3531 out of 4304 samples correctly classified) and a training accuracy of 78.3% (842 out of 1076 samples). Interestingly, the model performed slightly better on the test set than on the training set, which can occasionally occur due to variance in data distribution or the presence of regularization-like effects from the network structure.

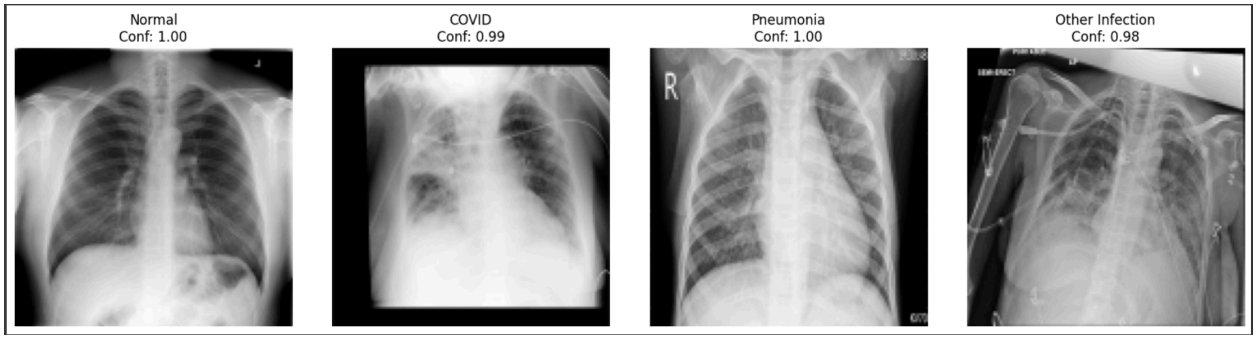
```
Test Set: Accuracy: 3531/4304 (82.0%)

Train Set: Accuracy: 842/1076 (78.3%)
```

```
Loaded Data
Epoch [1/30], Train Loss: 1.0074, Test Loss: 0.8578
Epoch [6/30], Train Loss: 0.6899, Test Loss: 0.6966
Epoch [11/30], Train Loss: 0.6070, Test Loss: 0.6103
Epoch [16/30], Train Loss: 0.5770, Test Loss: 0.6130
Epoch [21/30], Train Loss: 0.5216, Test Loss: 0.5737
Epoch [26/30], Train Loss: 0.5060, Test Loss: 0.5360
```



Overall, the model demonstrates solid predictive performance and effective learning. Future improvements could include experimenting with more conventional activation functions, adding dropout or batch normalization to improve generalization, and possibly increasing the learning rate to speed up convergence.

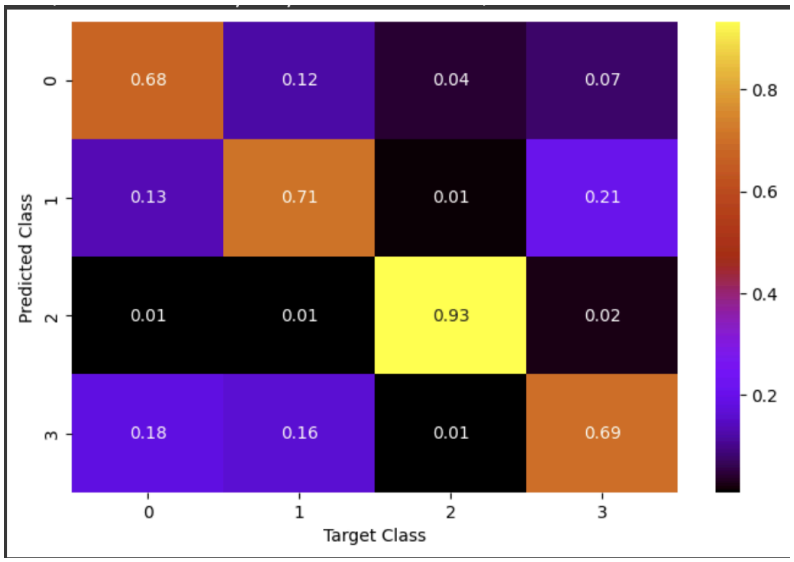
Visual Analysis



The model's most confident predictions for each diagnostic class—Normal, COVID, Pneumonia, and Other Infection—are illustrated in the images above. For the Normal class, the image is an excellent canonical example, displaying clear lung fields, no visible abnormalities, and well-defined anatomical structures such as the heart, diaphragm, and costophrenic angles. The model's perfect confidence score of 1.00 is well justified in this case. Similarly, the COVID prediction, with a confidence of 0.99, also represents a strong canonical example. The image shows diffuse bilateral opacities and ground-glass patterns that are typical of COVID-19-related lung infections, especially in the lower lobes, making it an appropriate reference point.

The image selected for Pneumonia, with a confidence of 1.00, is another strong example. It shows localized opacification in the right lung, characteristic of lobar pneumonia, while the rest of the lung field appears relatively unaffected. This clear presentation makes it a textbook case that aligns well with the model's high confidence. However, the image associated with Other Infection, despite the model's 0.98 confidence score, is not an ideal canonical sample. It includes medical devices such as wires or tubes and appears to be taken at an angle, making it harder to identify distinct infectious features. The lack of clear radiographic signs of infection, coupled with the presence of external hardware, introduces ambiguity that could potentially confuse both model training and human interpretation. This suggests a need to refine or better curate training examples for the "Other Infection" category to improve both diagnostic clarity and model performance.

The normalized confusion matrix highlights both the strengths and limitations of the model. One of the most frequent misclassifications occurs when "Other Infection" cases are predicted as "Normal" (18%). This suggests that the visual indicators of certain infections may be subtle or resemble normal anatomical structures, leading the model to interpret these X-rays as benign. Another pattern is the confusion between "COVID" and "Other Infection", with 21% of COVID cases being labeled as Other Infection and 16% of Other Infection cases being labeled as COVID. This misclassification likely stems from overlapping radiographic features such as opacities or consolidations, which are common to both conditions. The model appears to struggle in drawing a clear decision boundary between these two classes, possibly due to feature overlap or data imbalance in the training set. On the positive side, the model performs exceptionally well in identifying "Pneumonia" cases, achieving 93% accuracy, and shows reasonably good performance for "Normal" cases with 68% accuracy. To address the observed ambiguities, especially between COVID and Other Infections, further model tuning, incorporation of more labeled data, or integration of attention mechanisms to highlight infection-prone regions could enhance classification performance.

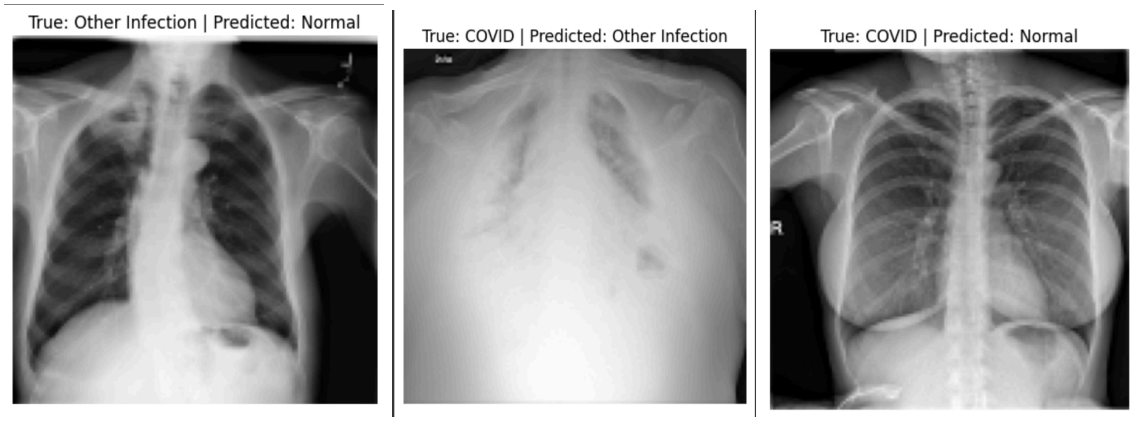


The provided radiographs visually represent the three most common misclassifications made by the model, offering insight into the challenges it faces in differentiating between similar conditions. The first image shows a case where the ground truth is "Other Infection" but was predicted as "Normal". This misclassification may be due to the subtlety of the infection's visual

cues; the X-ray appears largely unremarkable with clear lung fields and no obvious opacities, making it understandable why the model might interpret it as a normal case.

The second image presents a case where a "COVID" infection was predicted as "Other Infection". The radiograph is somewhat hazy with opacities that lack clear definition, which could resemble the presentation of other lower respiratory infections. This supports the observation from the confusion matrix that COVID and Other Infections share overlapping features, making it difficult for the model to distinguish between them reliably.

The third image shows another "COVID" case, but this time it was misclassified as "Normal". This image shows relatively clear lung fields with only very faint signs of possible abnormalities. If the COVID manifestation is mild or in early stages, it may not display strong radiographic signals, leading to this kind of error. Overall, these examples emphasize the limitations of the model in detecting subtle or overlapping radiographic features and highlight the importance of improving feature sensitivity, potentially with more balanced training data or model enhancements like attention-based mechanisms.



Note: All metrics computed are based on one run. They may change when re-run.