Exploring the viability of Convolutional Neural Networks (CNNs) on a multi-label classification task to simultaneously detect Pulmonary Edema and Pleural Effusion

Shiv Sakthivel[1]

[1]University of California, San Diego (ssakthiv@ucsd.edu)

ABSTRACT

This project focuses on training a Convolutional Neural Network (CNN) for a supervised classification task, specifically for predicting the presence of pulmonary edema and pleural effusion in chest radiographs. The project is a series of experiments to formulate a pipeline based on deep learning best practices, to achieve the best performing model for this multi-label classification task. The first experiment involved determining the appropriate application of transfer learning to chest radiograph image data. It was determined that the models should be trained through a serial unfreezing of the top layers in the ResNet152 architecture, at intervals of unfreezing one block of layers every 10 epochs with early stopping. The second experiment involved testing different formulations of the problem statement to achieve the best performing model. Separated binary label classifiers, a multi-label classifier and a multi-class classifier were all trained and evaluated using label prediction accuracy and the AUC (Area Under the Curve) of the ROC (Receiver Operating Characteristic) curve which is a measure of the model's discriminability. The multi-class classifier had the best overall performance, achieving an accuracy of 80% in the prediction of pulmonary edema and 83% in the prediction of pleural effusion, with the corresponding ROC AUC scores for both labels being 0.88 and 0.90 respectively.

INTRODUCTION

Radiographs are an imperative source of information for physicians in determining certain ailments their patients may have. With the huge advancements made in the field of deep learning, images can be processed in novel ways to produce insights efficiently, as opposed to time-intensive manual approaches even by professionals in that particular domain. This project deals with the application of convolutional

neural networks (CNNs), which are deep-learning architectures composed of a combination of convolutional layers (which consist of a set of filters to activate features of the input images), pooling layers (to reduce the dimensions within the network) and fully connected linear layers to ultimately produce the model's prediction. CNNs are most commonly used in image classification, a supervised machine-learning approach where the model is fed a fully labeled training dataset and predetermined classification categories, and it is able to learn features from the given data to accurately predict the class of unseen and unlabeled test data.

In the case of chest radiograph classification, there are significant time costs associated with the effort that radiologists need to apply to manually annotate radiographs for supervised training. Various high performance NLP tools like CheXpert and NegBio have been developed to automate this process. These algorithms can detect and infer medical terms directly from electronic health records and reports, associated with the radiographs. As a result, the necessary training data can be collated from the patient records without any additional manual effort. Expanding on the NLP tool, the CheXpert[2] initiative at Stanford significantly guided the approaches used in this project. The CheXpert task aimed to develop and evaluate deep learning algorithms for chest X-ray interpretation. To achieve this, a large-scale dataset of chest X-rays, along with associated radiology reports, was created and annotated. A Convolutional Neural Network (CNN) was trained on this dataset to classify the presence of 14 different thoracic diseases. The focus of this task was on leveraging the ability of deep learning algorithms to learn complex patterns in medical imaging data and apply them to make accurate predictions. The work behind training the CheXpert deep learning model has contributed to advancing the field of medical imaging and the development of more accurate and efficient algorithms for interpreting chest X-rays.

In this project, the MIMIC-CXR database of chest radiographs and patient reports will be used to train a CNN classifier, using transfer learning techniques. The MIMIC-CXR database[3,4] consists of 377,000 radiographs, stored in a DICOM format with resolutions of up to 4K. As a result, the size of this database

is 4 TB, and it is currently hosted in a Google Cloud bucket, where credentialed users can access the data. For this project, the classifier will predict two labels, associated with Pulmonary Edema and Pleural Effusion. In its current state, the database metadata contains information associated with 14 labels, automatically produced by running both the CheXpert and NegBio algorithms on the patient reports. Considering the performance of the model and computational resources available, the inputs to the classifier will be downsampled to standard 512*512 images, and a subset of 34,000 images which have labels associated with both conditions will be used for the training process.

OBJECTIVES

The first objective is to determine the balance point for unfreezing layers in the ResNet152 architecture during the model training phase, between faster convergence and overfitting. This will allow the development of a consistent training process with constrained hyperparameters to achieve a good model performance while allowing a comparison across models. The second objective is to train and compare the performances of a set of single binary label classifiers, a multi-label classifier and multi-class classifier, on the metrics of label prediction accuracy, ROC AUC score and overall training time. This will provide a basis for making a conclusion on how best to approach the task of multi-label classification for chest radiograph image data.

DATA

The MIMIC-CXR database is a collection of chest radiographs (CXRs) that has been curated from the Medical Information Mart for Intensive Care (MIMIC) database, which contains clinical information from over 60,000 critical care patients. The CXRs in the MIMIC-CXR database were acquired from various intensive care units (ICUs) using different radiographic equipment and techniques, resulting in a diverse range of imaging quality and patient positioning. The database contains over 377,000 de-identified CXRs, making it one of the largest publicly available datasets of chest radiographs.

The MIMIC-CXR database has been extensively annotated by expert radiologists to provide a rich set of labels and metadata. These annotations include radiographic findings, such as the presence or absence of lung opacities, pleural effusions, and cardiomegaly, as well as clinical data, such as patient demographics, hospital admission and discharge dates, and International Classification of Diseases (ICD) codes. The database also includes time-series data, allowing for the tracking of changes in radiographic features over time.

The labels used in this investigation correspond to Pulmonary Edema and Pleural Effusion. Pulmonary edema and pleural effusion are both medical conditions that can affect the lungs and breathing. Pulmonary edema is a condition in which excess fluid accumulates in the lungs, making it difficult to breathe. The excess fluid can be caused by several factors, such as heart failure, kidney failure, lung infections, or exposure to high altitudes. Pleural effusion is a condition in which excess fluid accumulates in the pleural space, which is the space between the two layers of tissue that surround the lungs. This can also make it difficult to breathe, as the excess fluid can put pressure on the lungs. While both conditions deal with the buildup of fluid, they present in different areas of a chest radiograph. Therefore, it would be worthwhile to train a CNN model to predict these labels simultaneously.

METHODS

I. EXPERIMENT I: DETERMINING A MODEL TRAINING PIPELINE

Transfer learning is a machine learning technique that involves using a pre-trained neural network to solve a new task. Instead of training a neural network from scratch, transfer learning involves taking an existing neural network that has already been trained on a large dataset and adapting it for a new task. The pre-trained neural network is typically a deep neural network that has learned to recognize complex features in images. To adapt the pre-trained network for a new task, the final layers of the network are typically replaced with new layers that are specific to the new task. These new layers are then trained on a smaller dataset that is specific to the new task. This

approach allows for the transfer of knowledge from the pre-trained network, which has learned to recognize general features, to the new task, which may require more specialized features. The following image shows how transfer learning is used for feature extraction:

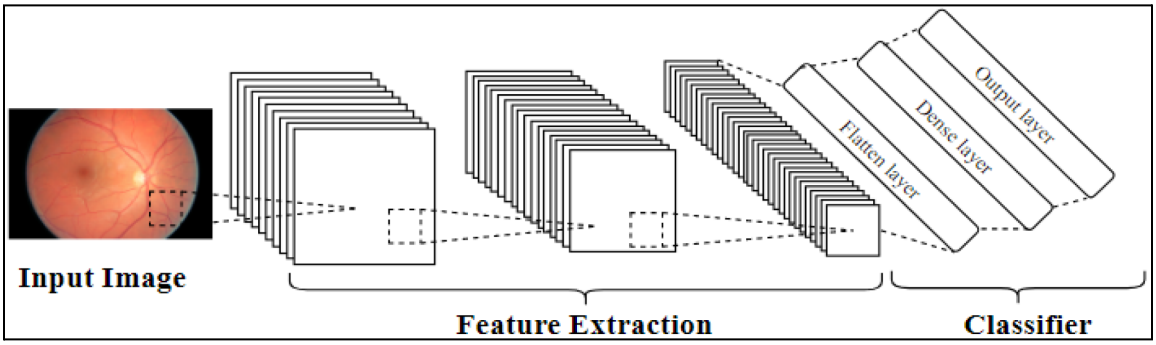


Figure 1: General use of CNN architectures in Transfer Learning[5]

The decision of when to unfreeze layers in a pre-trained neural network depends on several factors, such as the size and similarity of the new dataset to the original dataset, the complexity of the new task, and the performance of the model on the new task. In general, if the new dataset is small and similar to the original dataset, it may be beneficial to keep most or all of the layers in the pre-trained network frozen and only train the new layers that are specific to the new task. This approach helps to prevent overfitting on the small dataset and makes use of the pre-trained network's ability to recognize general features.

On the other hand, if the new dataset is large and dissimilar to the original dataset, it may be beneficial to unfreeze some of the layers in the pre-trained network and fine-tune them on the new task. In this instance, the ImageNet dataset consists of over 14 million images, which have been annotated with object labels and bounding boxes. The chest radiograph image data used in this project is dissimilar enough to the ImageNet dataset to warrant an investigation into whether layers with pre-trained weights should be unfrozen in the training process.

This experiment was conducted partly during Quarter 1, and extended for its specific application for this quarter's project. During Quarter 1, I trained a CNN regression model on similar chest radiograph image data with continuous BNPP serum biomarker labels from a dataset compiled at UCSD Health[6]. Initially, I left all 566 layers in the ResNet152 architecture frozen but quickly realized that due to the huge imbalance in number of trainable to untrainable parameters, the model simply wasn't learning. Therefore, I decided to explore unfreezing certain layers iteratively in the following schedule, and observed the following:

Epochs	**Unfrozen Layers**	**MAE on Test Set**	**Test Set Accuracy**	**Test Set ROC AUC**
0 - 35	-	0.6608	65.5%	0.66
35 - 50	Conv 5 Block 3	0.5818	72.2%	0.76
50 - 70	Conv 5 (All Blocks)	**0.5398**	**75.5%**	**0.81**
70 - 80	Conv 4 (Last 3 Blocks) + Conv 5 (All blocks)	0.5472	74.6%	0.80

Table 1: Iterative unfreezing of convolution layer blocks

The major observation from this experiment was that unfreezing further layers in the architecture allows the model to better fit to the new dataset, up until a certain point where it would begin to overfit as seen when Convolution 4 was also unfrozen for learning. Therefore, applying this understanding on where the ResNet152 architecture would begin to overfit, the following model training and serial layer unfreezing pipeline was developed as shown in Figure 2:

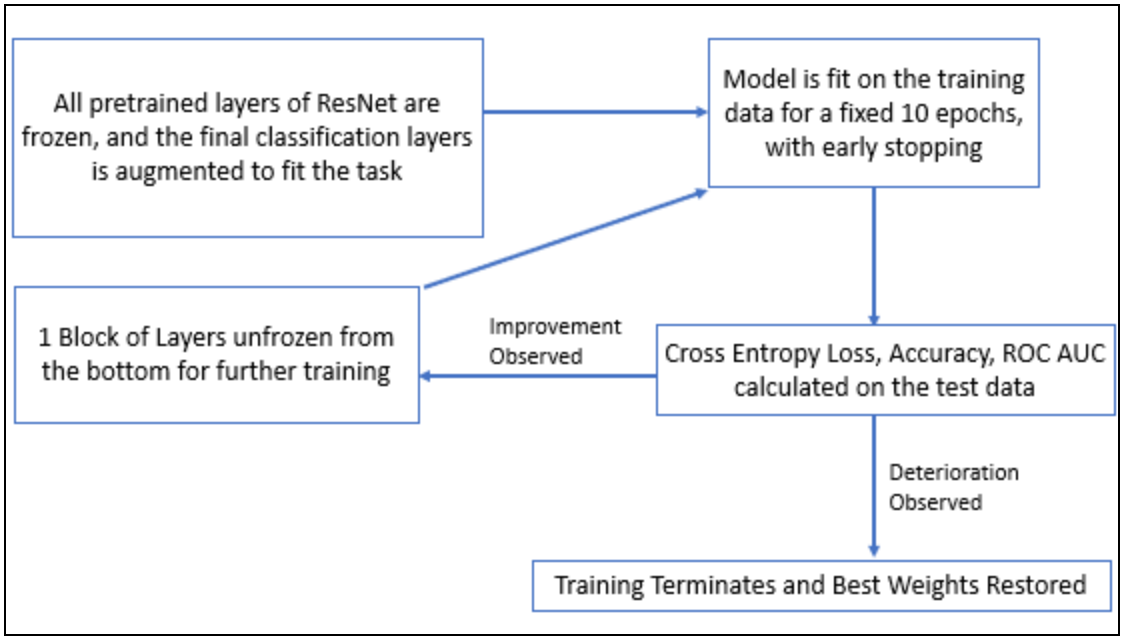


Figure 2: General Model Training Pipeline - Both the top and bottom layers of the architecture are augmented for the correct input image format and output label format, depending on the task.

The Model Training Pipeline shown in the figure was adapted from the Quarter 1 experiment with a few changes. Since the task at hand for this project is a classification task with binary labels as opposed to regression with a continuous target, the ResNet152 architecture tended to fit more quickly to the training data, despite the larger dataset being used for this project. Therefore, each setting of the model was trained for a shorter duration of 10 epochs with early stopping, and only one block of layers were unfrozen between settings as opposed to multiple blocks of layers as shown in Table 2. The training baseline of 10 epochs was determined after observing when early stopping took effect on a binary classifier for pulmonary edema using the MIMIC-CXR dataset with all frozen layers. Since this setting instantiates the least number of trainable parameters, it would also be the least likely to overfit. As hypothesized here, every further setting on the models for comparison converged sooner than 10 epochs, and the models are explored further in the following section.

II. EXPERIMENT II: DETERMINING THE BEST MODEL FOR THE MULTI-LABEL CLASSIFICATION TASK

Three approaches to implementing this multi-label classification task will be explored in this section, each with their own advantages and disadvantages. The following figure is a representation of the respective model architectures:

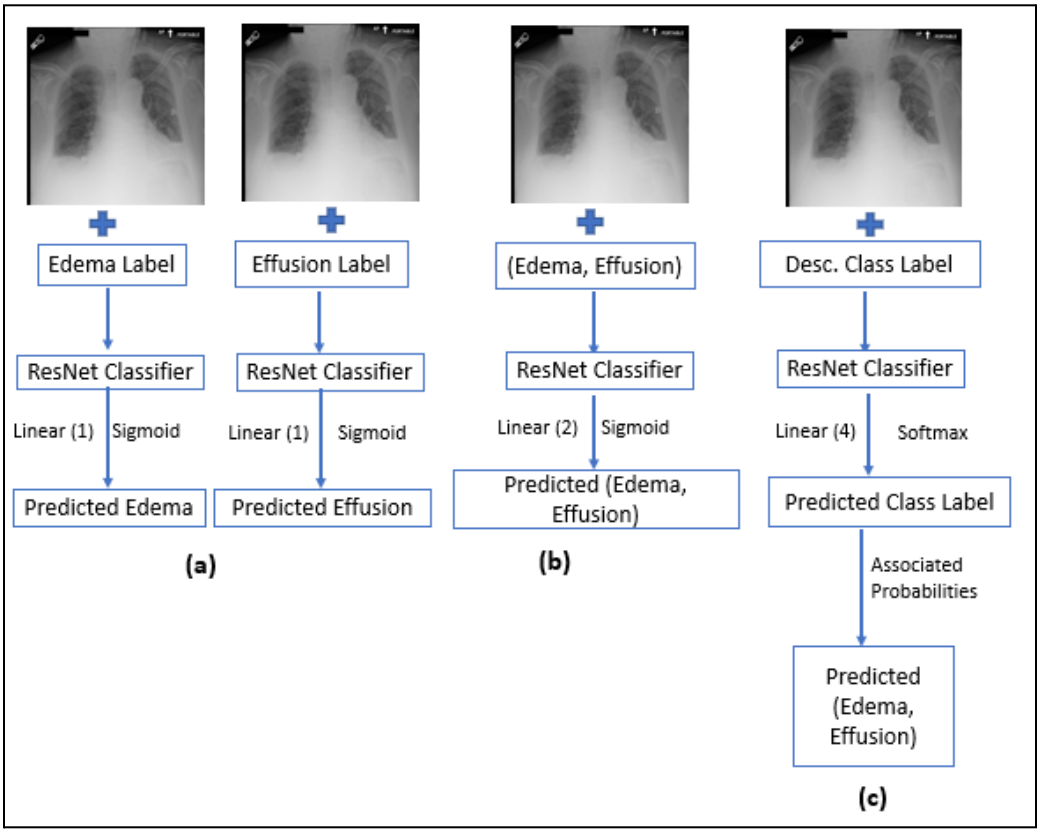


Figure 3: Model Training Methods

Model **(a)** describes a set of binary label classifiers. In this method, two separate ResNet152 architectures are trained, one using the Pulmonary Edema label and the other, the Pleural Effusion label. The final layer is augmented to a fully connected linear layer, with one output activated by a sigmoid function. The advantage of this approach is that the architecture can focus on fitting to a single target variable, rather than multiple variables. Therefore, the model expected to perform well on prediction accuracy. However, the time involved in training multiple classifiers is a major disadvantage compared to models (b) and (c).

Model **(b)** describes a multi-label classifier. In this method, one ResNet152 architecture is trained with both labels as the input. The final layer is augmented to a fully connected linear layer, with two outputs activated by a sigmoid function, corresponding to the pulmonary edema and pleural effusion labels. The scaleup in training time is the major advantage offered by this approach, however there is an expected tradeoff in prediction accuracy due to the same architecture, now having to fit to two labels.

Model **(c)** describes a multi-class classifier. The use of a multi-class classifier for a multi-label classification task is counterintuitive. However, when considering that pulmonary edema and pleural effusion are conditions that often present together in patients, the dataset of ~34,000 is skewed towards cases with the labels (0, 0) or (1, 1). More specifically, here is a table showing the value counts of radiographs for the different combinations of labels:

(Pulmonary Edema, Pleural Effusion)	**Number of Radiographs with these labels**
(0, 0)	11548
(0, 1)	5709
(1, 0)	2310
(1, 1)	14136

Table 2: Composition of the Training Dataset

As seen in Table 2, about ~26,000 of the training images account for cases with neither conditions present or both conditions present. Therefore, a multi-label classifier is likely to fit in such a manner, where the prediction of one label dictates the prediction of the other label, leading to misclassifications in cases where only one of the conditions are present in the radiograph. Therefore, with an ample number of training cases in each of the categories as shown in Table 2, it would be worthwhile to investigate a training method where the loss calculated during training

is weighted by a proportion inverse to its label representation in the dataset. Therefore, this class imbalance is addressed by reframing the multi-label classification task as a multi-class classification task where the array of labels are collapsed as: {(0, 0): 0, (0, 1): 1, (1, 0): 2, (1, 1): 3}, into 4 classes. Then, using the value counts from Table 2, class weights are calculated and accounted for in the model training process. For this classifier, the final layer is augmented to a fully connected linear layer, with four outputs activated by a softmax function, with each output corresponding to prediction confidence probability of each class. Once the model provides its prediction on the test data, the four outputs are concatenated using those probabilities in the following manner:

Prediction for Presence of Pulmonary Edema: **(0 * P(0)) + (0 * P(1)) + (1 * P(2)) + (1 * P(3))**

Prediction for Presence of Pulmonary Edema: **(0 * P(0)) + (1 * P(1)) + (0 * P(2)) + (1 * P(3))**

This is done instead of just extracting the maximum argument from the prediction and the label class it is associated with, in order to get a more exact representation of what the model has learned. By accounting for the class imbalance, it is expected that this model would perform better than its counterpart (b), and has the same computational resource and time efficiency advantages from (b) as well. Models (a) and (b) are converged on binary cross-entropy loss, while model (c) is converged on weighted categorical cross-entropy loss. Both these loss functions are the standard used in classification algorithms.

MODEL TRAINING SETTINGS

I. **Loss Function:** In the case of the single label and multi label classifiers with a sigmoid activated output layer, the model is fitted using binary cross entropy as the loss function. Binary cross entropy measures the difference between the predicted output probabilities and the true labels, and penalizes the model for incorrect predictions. It is defined as the negative sum of the logarithm of the predicted probabilities for the correct class, and is commonly used in conjunction

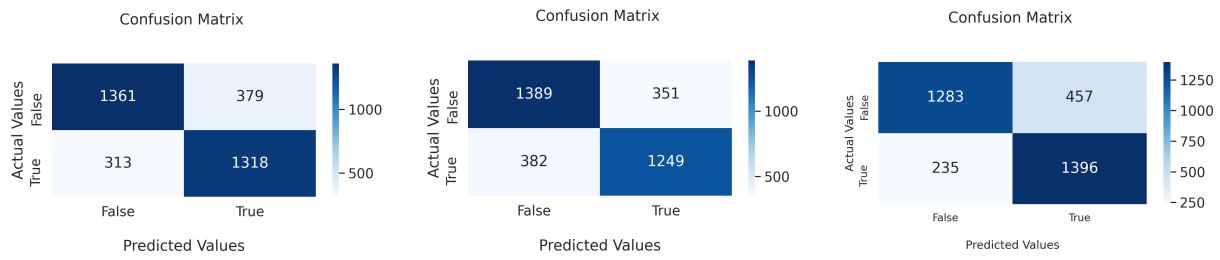
with a sigmoid activation function in the output layer of the neural network. The multi-class classifier with a softmax activated output layer is fitted using class weighted categorical cross entropy. Categorical Cross Entropy measures the dissimilarity between the predicted probability distribution and the true probability distribution, and penalizes the model for incorrect predictions. It is defined as the negative sum of the logarithm of the predicted probability for the true class, where the predicted probabilities are normalized by a softmax function to ensure that they sum up to one. Class weighted categorical cross entropy is a variation of the categorical cross entropy loss function that is used in multi-class classification problems where the classes are imbalanced. Class weighted categorical cross entropy addresses this issue by assigning a weight to each class proportional to its inverse frequency in the training set. This means that the loss contribution of rare classes is increased during training, while the loss contribution of frequent classes is decreased. This would allow the model to generalize better.

II. **Optimizer:** All the models are trained using the Adam optimizer, which is a popular optimization algorithm used in deep learning that adapts the learning rate based on the gradient of the loss function with respect to the model parameters. Adam maintains a moving average of the gradient and the squared gradient, and uses these estimates to update the learning rate for each parameter in a way that scales the learning rate differently for each parameter. This means that Adam can handle noisy or sparse gradients and converge faster than other optimization algorithms. Adam is a popular choice for many deep learning models and has been shown to be effective in achieving high accuracy with minimal tuning of the hyperparameters.

III. **Early Stopping:** For every training setting, early stopping is instituted with a tolerance of rising validation loss on three epochs. Early stopping is a technique used to prevent overfitting in machine learning models. Early stopping works by monitoring the performance of the model on a validation set during training and stopping the training process when the validation error starts to increase or stops decreasing.

RESULTS

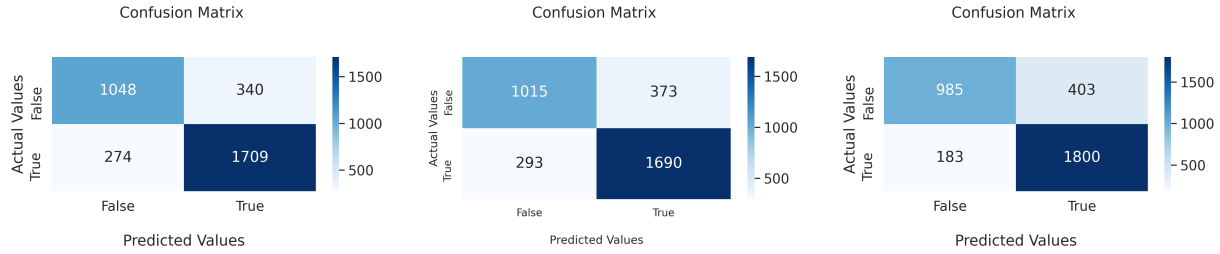
The following figures and table show the results of the best model obtained for the methods (a), (b) and (c). The first set of figures are the confusion matrices of the model predictions for the Pulmonary Edema Label on an unseen test set (n = 3371). A confusion matrix visualizes the number of true positives, false positives, false negatives and true negatives. The raw outputs of each model were converted to binary predictions at the threshold 0.5:



(a) **(b)** **(c)**

Figure 4: Confusion Matrices on the Test Set Edema Predictions

This next set of figures are the confusion matrices of the model predictions for the Pleural Effusion Label on the unseen test set:



(a) **(b)** **(c)**

Figure 5: Confusion Matrices on the Test Set Effusion Predictions

The following table describes the prediction accuracy of each of the models on the two target labels, along with a scatter plot to visualize their relative performances:

	Pulmonary Edema Label	Pleural Effusion Label
Single Binary Label Classifier	**0.797**	0.818
Multi-Label Classifier	0.783	0.802
Multi-Class Classifier	0.795	**0.826**

Table 3: Model Accuracy

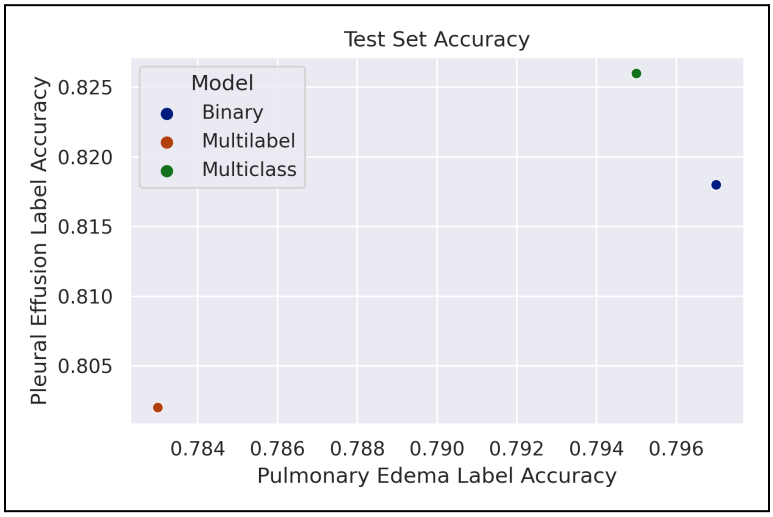


Figure 6: Model Accuracy Comparison Chart

The next set of figures show the ROC curve of the models on the unseen test along with annotations with their respective AUCs, for each target label:

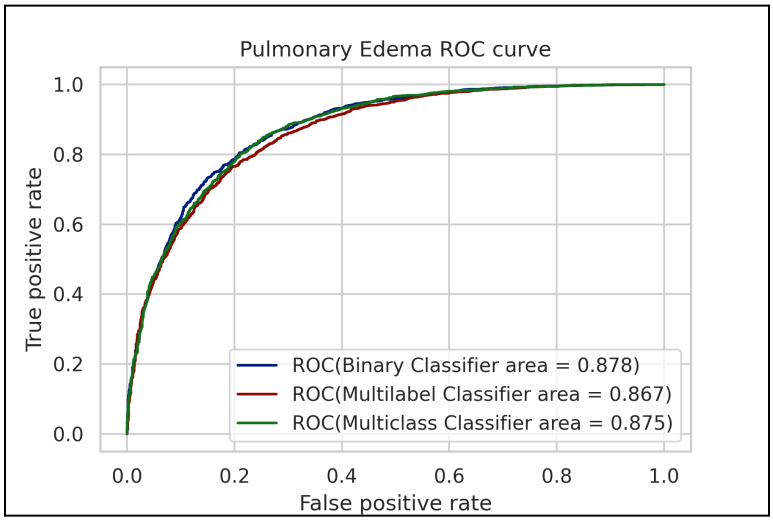


Figure 7: Model Comparison on Pulmonary Edema ROC Curves

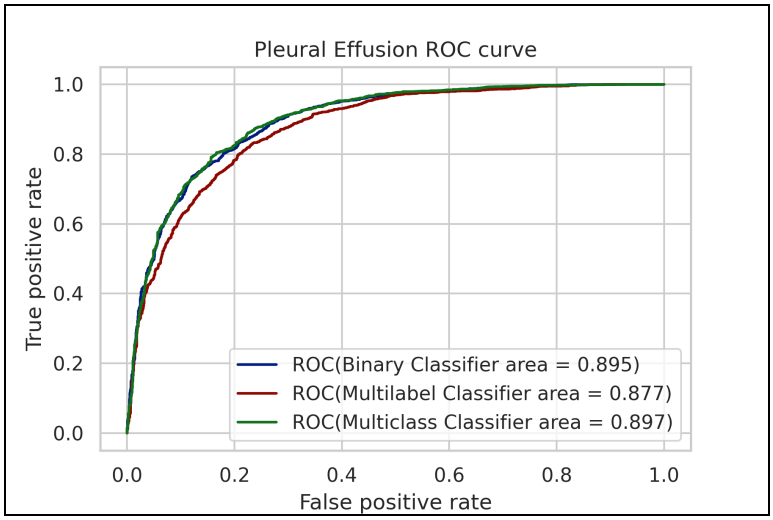


Figure 8: Model Comparison on Pleural Effusion ROC Curves

Finally, the following visualization is a scatterplot of the overall number of epochs against training time per epoch. For the binary label classifier, the number of epochs for both models is added together for this statistic, and the per epoch training time is included in the average. An efficient model would be observed towards the bottom left.

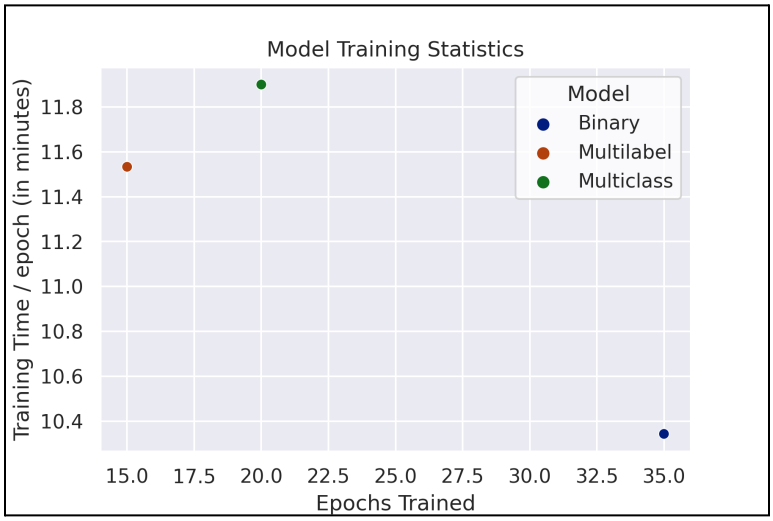


Figure 9: Model Comparison on Training Statistics

DISCUSSION

In this project, I investigated the effectiveness of progressively unfreezing blocks of layers in the large ResNet152 architecture for a binary-label classification task involving pulmonary edema and pleural effusion, and found that this technique led to a speed-up in model convergence before overfitting occurred, which improved the overall efficiency of the model. I also compared the performance of multiclass and multilabel classifiers for this task, and found that the multiclass classifier was a more efficient method when considering training time, with little to no tradeoff in overall prediction accuracy or model discriminability. Framing the multi-label classification task as a multi-class classification task was essential in accounting for the imbalance in training cases where either condition was observed in isolation. As seen in Figures 6, 7 and 8, the multi-class model outperformed the multi-label classifier in terms of discriminability and accuracy. However, it needs to be noted that extending this investigation to multi-label tasks with significantly more than 2 labels may not be feasible due to the difficulty in finding sufficient training data for every combination of labels. These findings provide valuable insights into the optimization of deep learning models for medical image classification tasks, specifically for binary-label classification tasks involving pulmonary edema and pleural effusion. The results suggest that progressively unfreezing blocks of layers and using multiclass classifiers can lead to more efficient and accurate models for this type of task.

ACKNOWLEDGEMENTS

- Dr Albert Hsiao, for his mentorship and close guidance in the various aspects and overall direction of this project.
- Peers in my DSC 180 section, for their valuable insights and collaboration across both quarters.
- Mr Suraj Rampure, and the DSC 180 Instructional Staff for their support throughout this process.

REFERENCES

[2] *Irvin et. al, CheXpert: A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison, arXiv:1901.07031 [cs.CV]*

[3] *Johnson, A., Pollard, T., Mark, R., Berkowitz, S., & Horng, S. (2019). MIMIC-CXR Database (version 2.0.0). PhysioNet. https://doi.org/10.13026/C2JT1Q.*

[4] *Johnson, A.E.W., Pollard, T.J., Berkowitz, S.J. et al. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. Sci Data 6, 317 (2019). https://doi.org/10.1038/s41597-019-0322-0*

[5] *Kandel, I.; Castelli, M. Transfer Learning with Convolutional Neural Networks for Diabetic Retinopathy Image Classification. A Review. Appl. Sci. 2020, 10, 2021. https://doi.org/10.3390/app10062021*

[6] *J. Huynh et al., "Deep Learning Radiographic Assessment of Pulmonary Edema: Optimizing Clinical Performance, Training With Serum Biomarkers," in IEEE Access, vol. 10, pp. 48577-48588, 2022, doi: 10.1109/ACCESS.2022.3172706.*