Bilkent University

Department of Computer Science

CS 464 TERM PROJECT: DIAGNOSIS FROM CHEST X-RAYS

Project Group Number: 3

Section: 1

Final Report

Arda İynem 22002717
Hasan Ege Tunç 22003814
Enes Bektaş 22002401
Ömer Fırat Bekiroğlu 22002239
Sabri Eren Dağdelen 22001764

Instructor: Abdullah Ercüment Çiçek

Project Final
May 12, 2024

1.0 Introduction and Background Information....................3
2.0 Problem Description....................3
3.0 Methods....................4
3.1 Dataset (DS)....................4
3.1.1. DS-1....................5
3.1.2 DS-2....................6
3.1.3 DS-3....................7
3.1.4 Data Preprocessing & Augmentation....................7
3.2 Methods....................7
3.2.1 Traditional Methods....................7
3.2.1.1 Naive Bayes Classifier....................7
3.2.1.2 Random Forest Classifier....................7
3.2.1.3 AdaBoost Classifier....................8
3.2.1.4 KNN Classifier....................8
3.2.1.5 Decision Tree....................8
3.2.1.6 Support Vector Machine....................8
3.2.1.7 Logistic Regression....................8
3.2.2 Neural Network Methods....................8
3.2.2.1 Custom CNN Models....................8
3.2.2.2 MobileNet....................8
3.2.2.3 ResNet....................8
3.2.2.4 VGG19....................9
3.2.2.5 GoogLeNet (Inception)....................9
3.2.2.6 DenseNet121....................9
4.0 Results & Discussions....................10
4.1 Traditional Methods....................10
4.1.1 Naive Bayes Classifier....................10
4.1.2 Random Forest Classifier....................11
4.1.3 AdaBoost Classifier....................12
4.1.4 KNN Classifier....................13
4.1.5 Decision Tree....................14
4.1.6 Support Vector Machine....................15
4.1.7 Logistic Regression....................16
4.2 Neural Network Methods....................17
4.2.1 Custom CNN Models....................17
4.2.1.1 Custom CNN 1....................17
4.2.1.2 Custom CNN 2....................18
4.2.2 MobileNet....................19
4.2.3 ResNet....................20

4.2.3.1 ResNet with Pretrained ImageNet-18 Weights.......... 20
4.2.3.2 ResNet without Pre-trained Weights.......... 22
4.2.3.3 ResNet with Pre-trained ImageNet-18 Weights and Data Augmentation...... 23
4.2.4 VGG19.......... 24
4.2.5 GoogLeNet (Inception).......... 25
4.2.5.1 Inception without Pre-training.......... 26
4.2.5.2 Inception with Pre-Training.......... 27
4.2.6 DenseNet 121.......... 28
5.0 Conclusions.......... 29
6.0 Appendixes.......... 31
Appendix A - Contributions.......... 31
7.0 References.......... 31

1.0 Introduction and Background Information

Chest X-ray imaging stands as a cornerstone in medical diagnostics, playing a pivotal role in the detection and monitoring of a spectrum of thoracic pathologies [1]. Its widespread use underscores its importance as an initial diagnostic tool in medical practice. However, the efficacy of X-ray imaging encounters constraints inherent to the complexities of image interpretation and the nuanced nature of certain pathologies. These factors contribute to the potential for diagnostic errors, underscoring the need for a nuanced approach to clinical assessment.

To be explicit, despite its significance, chest X-ray imaging, particularly in the AP(anterior-posterior) or PA(posterior-anterior) views - AP view involves projecting X-rays from the front of the patient towards the back, while the PA view reverses this, projecting X-rays from the back to the front- may present limitations in certain clinical scenarios. The inherent two-dimensional nature of X-ray images can obscure subtle abnormalities and fail to provide sufficient detail for comprehensive diagnosis. It could even be the case that certain chest pathologies, such as nodules and masses, pleural thickening and effusion, or subtle pulmonary infiltrates, pose challenges in their differentiation solely through X-ray results, underscoring the necessity of complementary imaging techniques [2]. Consequently, supplementary imaging techniques, such as computed tomography (CT) or magnetic resonance imaging (MRI), may be warranted to complement X-ray findings and facilitate a more precise diagnosis.

Being aware of the chest X-ray's nature and equipped with the purpose of understanding to which extent the differentiation of chest diseases can be performed, we tried traditional techniques in machine learning used in solving classification problems (i.e. kNN, SVM, decision trees, logistic regression) and experimented custom CNN models and architectures. We also investigated the performance of models (i.e. VGG 19, ResNet, Inception V3, AlexNet) pre-trained on the ImageNet [3] dataset; thereby resulting in transfer learning. As a result, we concluded that pre-trained models are more useful and accurate compared to the traditional techniques. However, they are still deprived of providing a sufficient accuracy rate to be trusted, which can be linked to the constraints of X-ray imaging as well.

2.0 Problem Description

As an imaging technique, chest X-rays are not necessarily related to a single disease, meaning that one image can display indications of multiple diseases. For instance, infiltration can be paired with effusion, and they can even be accompanied by other diseases such as nodules and masses. Hence, we restricted our attention to classifying images that show either indications of a single disease or no indication. The choice of diseases classified is based on their frequency in the dataset or their distinguishability from each other, where the dominance of each criterion varies from time to time. To illustrate, while the masses and nodules are frequent in the dataset, only the images with “nodule” are chosen and those with “mass” are excluded. On the other hand, both atelectasis and effusion, two diseases that are likely to be confused, are included in the dataset since they are amongst the top frequent diseases in the dataset. Hence the main question we are trying to address is:

“To what extent do traditional learning methods and neural network architectures be effective in overcoming the limitations of X-ray imaging and distinguishing the various chest diseases?”

The question above leads to sub questions that need clarifying.

SQ1) While maintaining the diversity of classes, what could be the possible subsets of chest diseases in our dataset such that they can be distinguished?

SQ2) How should data be preprocessed to overcome the limitations of X-ray imaging?

SQ3) How can traditional learning methods and neural network architectures be compared regarding the accuracy they provide?

SQ4) To what extent can the traditional learning methods and neural network architectures distinguish the diseases that are possibly confused?

Our work was intended to answer the above questions under the umbrella of the main question.

3.0 Methods

3.1 Dataset (DS)

The DS used contains over 112,000 Chest X-ray images from more than 30,000 unique patients [4]. The DS comprises a collection of chest X-ray images obtained from the National Institutes of Health. Each image is multi-labeled with information regarding the presence of 14 common thoracic pathologies, including Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural thickening, Cardiomegaly, Nodule, Mass and Hernia or “No finding” [4]. In addition, the DS contains metadata for all images involving image index, finding labels, patient information, view position, original image size and original image, and finally pixel spacing. Progress report already presented a detailed analysis of DS but recalling some important ones could be beneficial:

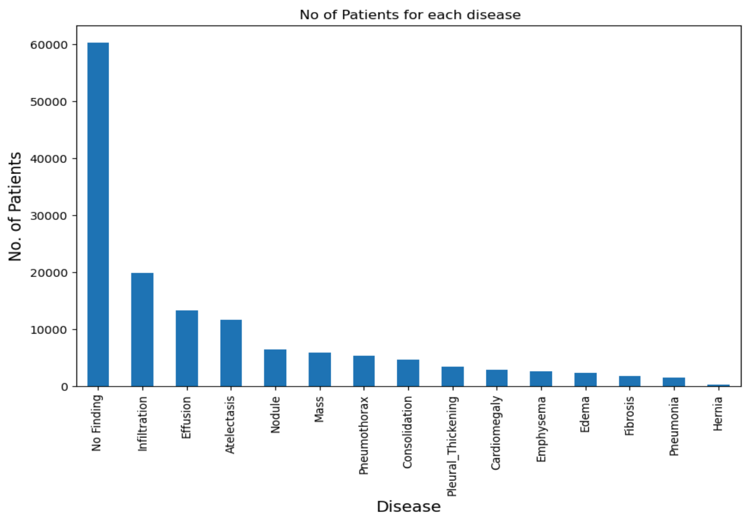


Fig. 1: Number of patients with respect to diseases (a patient can have more than a single disease)

Since the actual DS to be used is very large - around 45 GB, we decided on using a subset, which is small enough to be worked on. As mentioned, there are 14 different classes and many of the images are multi-labeled and we ended up choosing images that have a single label only. The most common findings combined with the intuition of disease separability from the chest X-rays are *atelectasis, effusion, nodule, pneumothorax, cardiomegaly, infiltration,* and *"no finding*". Therefore, these classes correspond to our answer to SQ1. So far, we have used 3 different subsampled DSs. These are as follows:

3.1.1. DS-1

We randomly subsampled 9000 images as the first data set **(DS-1)** which contains 3000 images of no finding and 1000 images per disease. Some of the models we trained in development used that data set. Having noticed that this subset is very biased (see Fig. 2), we shifted the dataset.

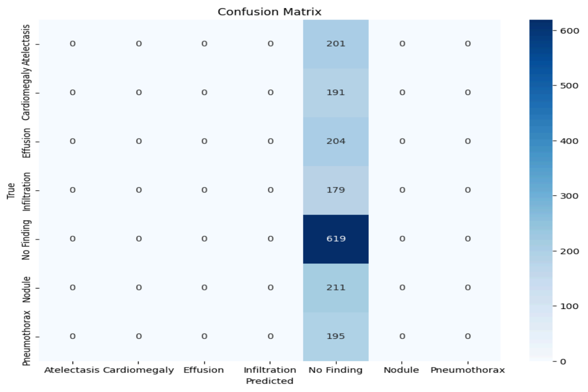


Fig. 2: Confusion matrix of Resnet architecture trained on DS-1 proving that DS-1 is completely biased.

3.1.2 DS-2

Having noticed that this subset is very biased, we even subsampled DS-1 to make it balanced, we ended up with 7000 images in total, 1000 images per finding, which constitutes the DS-2.

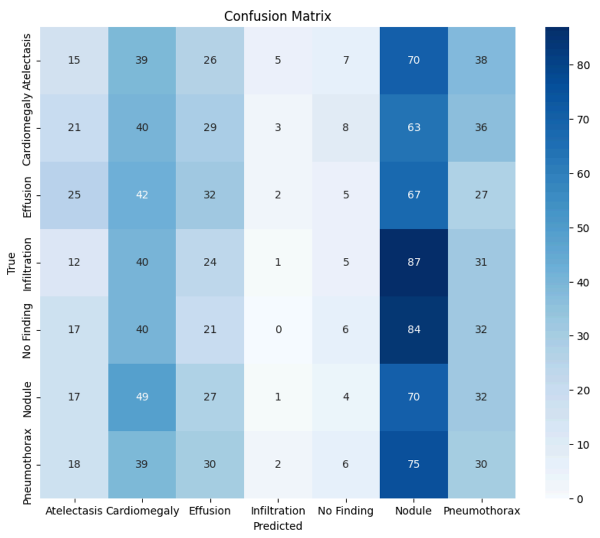


Fig. 3: Confusion matrix of LeNet-5 architecture trained on DS-2

As seen above, infiltration and "no finding" are rarely predicted, based on this argument we decided to remove infiltration instances from our DS-2, and further examined its structure. It became clear that original image sizes and view positions (AP and PA) differ very much (i.e. some images are nearly 3000 x 3000 in pixel length and width while some images are nearly 1000 x 1000); thereby, requiring handling. Furthermore, in data preprocessing, we are resizing the images for neural network architectures and it may happen that the information we lose highly depends on the image size. Thus, we need to ensure that image sizes are close to each other and view positions match. Our final analysis in the actual DS - the large one - yields the following numerical data for the number of images satisfying the above constraints:

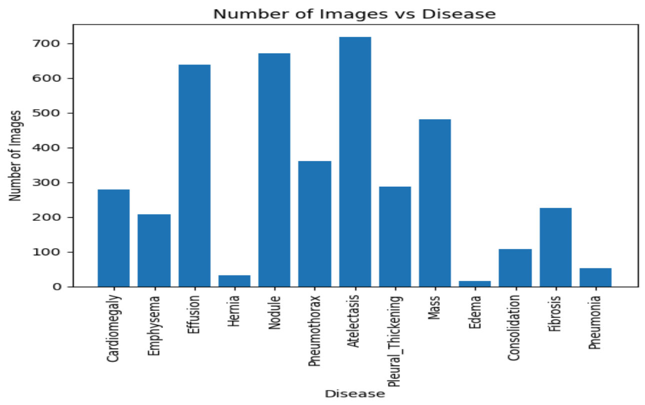


Fig. 4: Number of images in each category whose image sizes are around the mean and view position is PA.

Hence, we constructed the DS-3 that is used to obtain the results discussed throughout the report.

3.1.3 DS-3

The DS-3 contains 1000 images in total, 200 images per finding where findings are *atelectasis, effusion, nodule, pneumothorax,* and *“no finding*”. Note that these are the findings that appear the most in Fig. 3 and that are also used in DS-2. Hence the set F = {*atelectasis, effusion, nodule, pneumothorax, “no finding*”} is our final answer to the SQ1.

3.1.4 Data Preprocessing & Augmentation

Regarding the data augmentation, for the progress stage, we used ImageDataGenerator from tensorflow. keras.preprocessing.image including the operations rescale, rotation, width shift, height shift, shear, zoom and horizontal flip. However, these operations seem to further decrease our accuracy rate, and hence we avoided them in the final stage.

In addition, in the final stage, while training the ResNet model, we tried transformations of PyTorch including random horizontal flips, random rotations, color jittering, resizing, and converting to tensors. However, similar to what we have experienced from the progress stage, transforms did not impact our model positively. Therefore, as data preprocessing methods, we only preferred data standardization and resizing for computational efficiency. Indeed, this implies that our answer to SQ2 is that chest X-ray images carry sensitive information; thus, the fewer transformations are applied, the better accuracy rate is obtained. However, standardization techniques work.

3.2 Methods

3.2.1 Traditional Methods

3.2.1.1 Naive Bayes Classifier

Naive Bayes is a simple probabilistic classifier based on Bayes' theorem with conditional independence assumption between features.

3.2.1.2 Random Forest Classifier

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification.

3.2.1.3 AdaBoost Classifier

AdaBoost is an ensemble learning method that combines multiple weak classifiers to create a strong classifier. It focuses on improving classification accuracy by giving more weight to misclassified data points.

3.2.1.4 KNN Classifier

K-Nearest Neighbors is a simple, instance-based learning algorithm where the prediction is based on the majority class among the k-nearest neighbors of a data point in the feature space.

3.2.1.5 Decision Tree

Decision Trees recursively partition the feature space into regions, making decisions based on the value of features.

3.2.1.6 Support Vector Machine

SVM is a supervised learning algorithm used for classification and regression tasks. It finds the optimal hyperplane that best separates classes in the feature space.

3.2.1.7 Logistic Regression

Logistic Regression is a linear model used for binary classification tasks. It predicts the probability that a given instance belongs to a particular class using the logistic function.

3.2.2 Neural Network Methods

3.2.2.1 Custom CNN Models

We created two custom CNN models using convolution, Relu, max-pooling, fully connected, and softmax output layers.

3.2.2.2 MobileNet

It employs depth-wise separable convolutions, separating spatial and depthwise convolutions to reduce the number of parameters and computations. MobileNet strikes a balance between accuracy and computational efficiency, making it ideal for resource-constrained environments [5].

3.2.2.3 ResNet

Addressed the challenge of training very deep networks. ResNets introduce shortcut connections that bypass one or more layers, allowing the gradient to flow more easily during backpropagation. This architectural innovation facilitated the training of extremely deep networks, reaching hundreds of layers [5].

3.2.2.4 VGG19

VGG19 is a deep convolutional neural network architecture famous for its simplicity and effectiveness in image recognition tasks. It consists of 19 layers, mostly convolutional layers followed by max-pooling layers, ending with three fully connected layers for classification. We used this model as the base for our transfer learning method [5].

3.2.2.5 GoogLeNet (Inception)

Employs parallel convolutional operations with different kernel sizes. This architecture efficiently captures features at multiple scales, promoting better generalization [5].

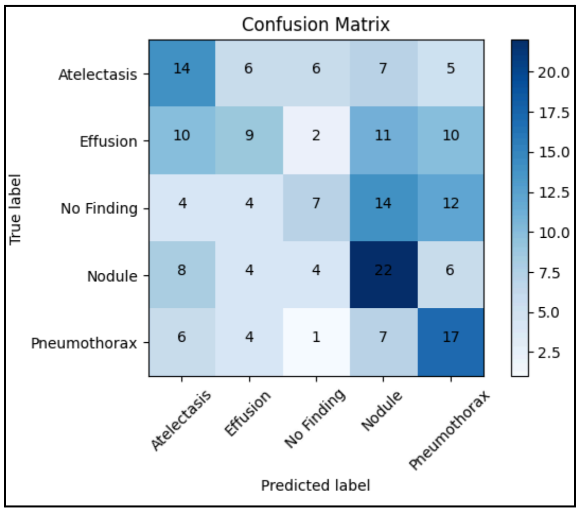
3.2.2.6 DenseNet121

DenseNet is a convolutional neural network where each layer is connected to all other layers that are deeper in the network, enabling maximum information flow along layers [6].

4.0 Results & Discussions

4.1 Traditional Methods

4.1.1 Naive Bayes Classifier



True label \ Predicted label	Atelectasis	Effusion	No Finding	Nodule	Pneumothorax
Atelectasis	14	6	6	7	5
Effusion	10	9	2	11	10
No Finding	4	4	7	14	12
Nodule	8	4	4	22	6
Pneumothorax	6	4	1	7	17

Fig. 5: Confusion Matrix of Naive Bayes

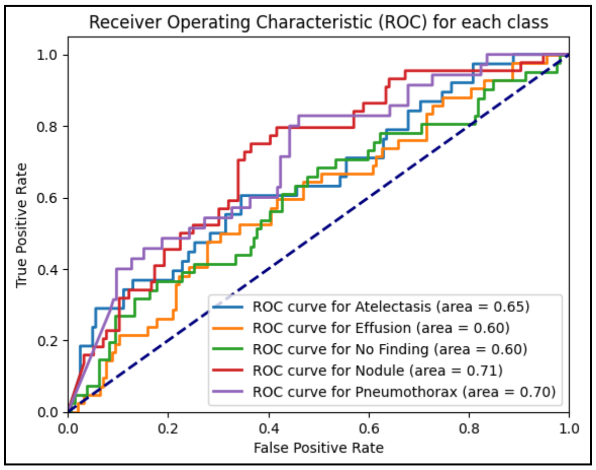


Fig. 6: ROC curve of Naive Bayes

```
Accuracy: 0.3450
Macro Precision: 0.3435
Macro Recall: 0.3478
Macro F1 Score: 0.3319
```

The Naive Bayes classifier demonstrated relatively modest performance on the dataset. Thus, we decided to use the grid search technique to refine parameters. This approach led to a slight enhancement in overall performance. As indicated by the ROC curves, the classifier more effectively identifies conditions like Nodule and Pneumothorax compared to other diseases. This differential performance could be influenced by the overlapping or similar symptoms among the various diseases, which complicates the model's ability to distinctly classify each condition.

4.1.2 Random Forest Classifier

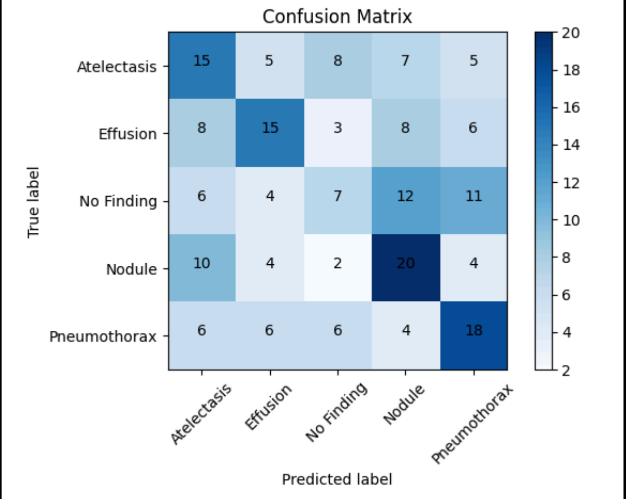


Fig. 7: Confusion Matrix of Random Forest

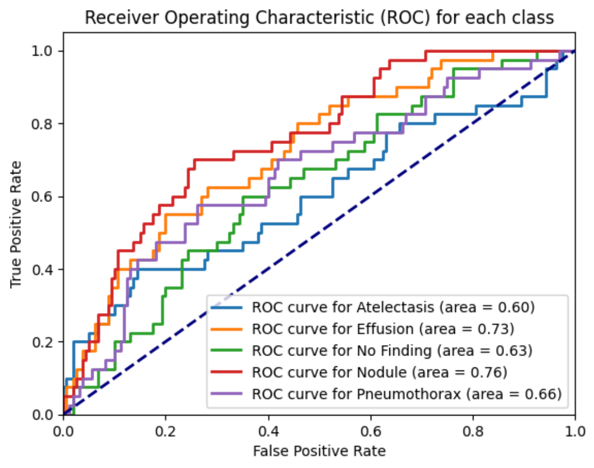


Fig. 8: ROC Curve for Random Forest

```
Best Model Parameters: {'max_depth': 10, 'min_samples_leaf':
1,'min_samples_split': 5, 'n_estimators': 150}

Accuracy: 0.3750
Macro Precision: 0.3690
Macro Recall: 0.3750
Macro F1 Score: 0.3677
```

The random forest classifier with default settings was no better than a random guess. Hence, we used GridSearchCV (5-fold cross-validation is done for each combination of hyperparameters) on hyperparameters 'max_depth', 'min_samples_leaf', 'min_samples_split', and 'n_estimators', which revealed the optimal hyperparameters given above. As a result, the accuracy of random forest increased to 0.375. This accuracy of the optimized random forest is as good as the custom CNN models we created. According to AUCROC values, Nodule and Effusion are the best-classified classes for Random Forest. Random forest gave the best results among traditional models.

4.1.3 AdaBoost Classifier

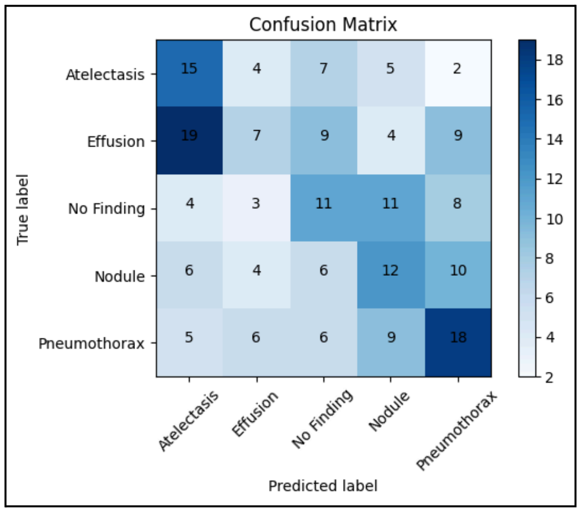


Fig. 9: Confusion Matrix of AdaBoost

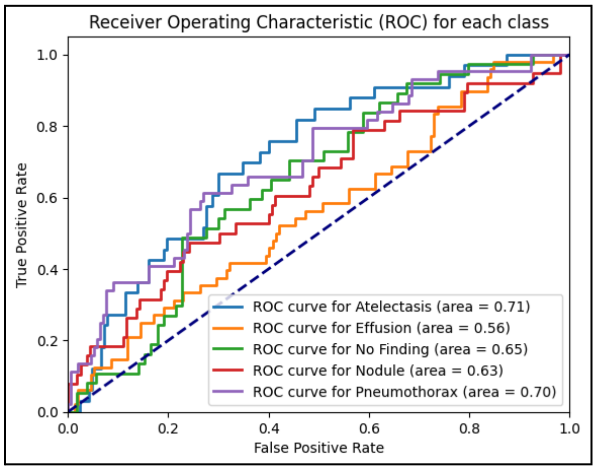


Fig. 10: ROC Curve for AdaBoost

```
Best Parameters: {'learning_rate': 0.1, 'n_estimators': 150}
Best Score: 0.31627289196774705

Accuracy: 0.3150
Macro Precision: 0.3111
Macro Recall: 0.3245
Macro F1 Score: 0.3098
```

AdaBoost, initially applied with default settings, yielded poor results, barely surpassing 20% accuracy (not better than a random guess considering we have 5 classes to predict). This required us to do an exhaustive optimization process using GridSearchCV (5-fold cross-validation is done for each combination of hyperparameters), revealing optimal parameters: a learning rate of 0.1 and 150 estimators (for the estimator of the AdaBoost, decision trees we created before with also GridSearchCV are used). According to grid search and AdaBoost ensemble technique, the best setting was to use 150 decision trees. Consequently, the ensemble method has shown its power and accuracy rose to 31.50%, indicating the improvement. Precision, recall, and F1 score metrics further illuminated the model's enhanced performance, while ROC curves highlighted the improved prediction capability compared to the decision tree where “Pneumothorax” was the best-performing class in terms of prediction. However, AdaBoost failed to discriminate between *“Atelectasis”* and *“Effusion”* as well as the decision tree; which is also a hard task to do in real life.

4.1.4 KNN Classifier

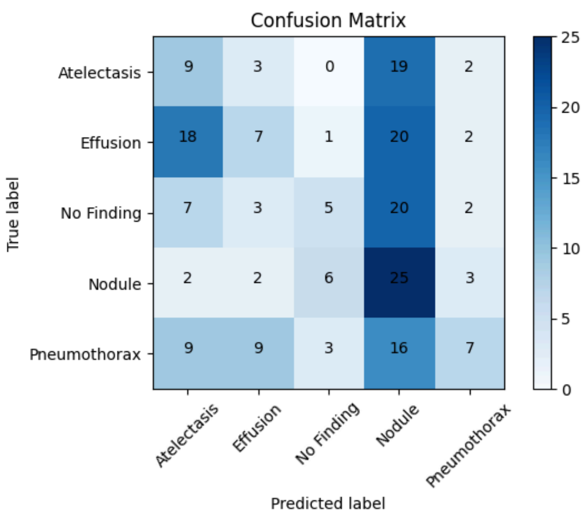


Fig. 11: Confusion Matrix of KNN with N=25

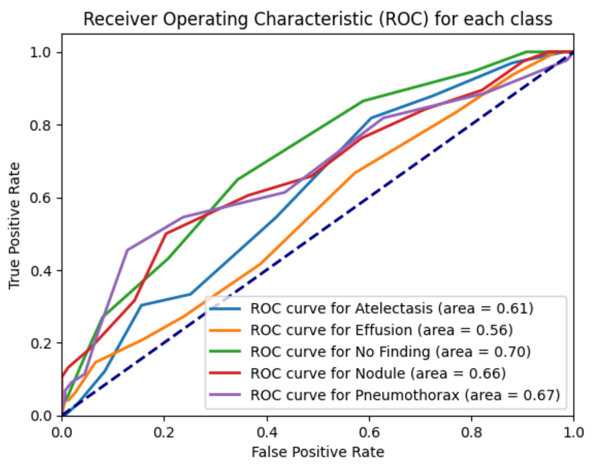


Fig. 12: ROC Curve for KNN with N=25

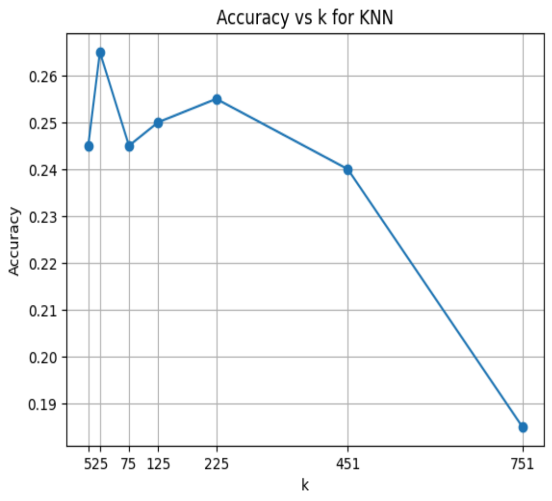
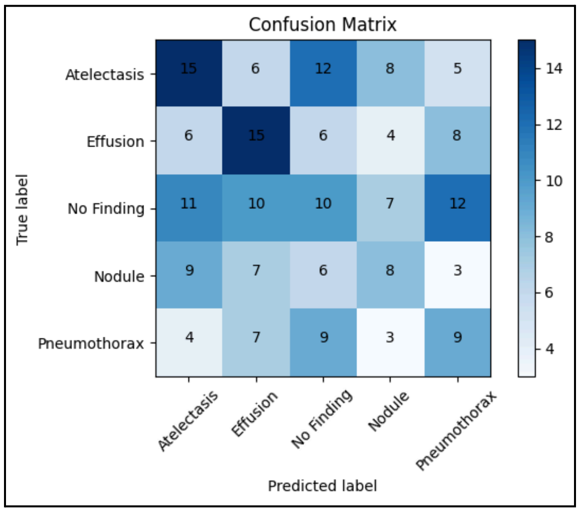


Fig. 13: Accuracy vs k

We trained K-NN using k values in the set {5,25,75,125,225,451,751} as seen from the plot, and k=25 seems to be the best choice. However, the K-NN algorithm appears to be a not good choice, because the model is very biased towards nodule class. The reason could be that images with nodules are distributed uniformly around the feature space.

```
Accuracy: 0.2650
Macro Precision: 0.3025
Macro Recall: 0.2741
Macro F1 Score: 0.2426
```

4.1.5 Decision Tree



	Atelectasis	Effusion	No Finding	Nodule	Pneumothorax
Atelectasis	15	6	12	8	5
Effusion	6	15	6	4	8
No Finding	11	10	10	7	12
Nodule	9	7	6	8	3
Pneumothorax	4	7	9	3	9

Fig. 14: Confusion Matrix of Decision Tree

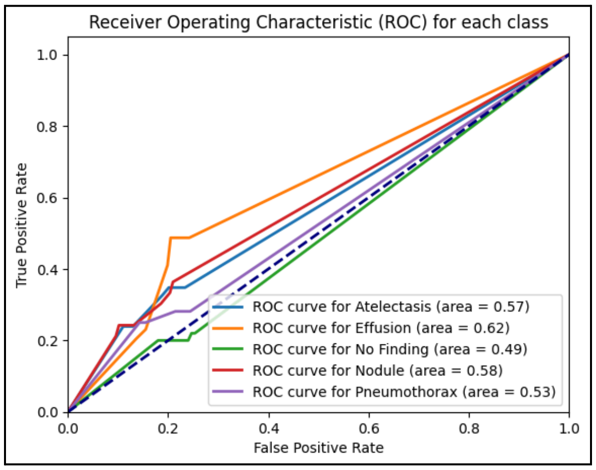


Fig. 15: ROC Curve for Decision Tree

```
Best Hyperparameters: {'criterion': 'gini', 'max_depth': None,
'max_features': 'sqrt', 'min_samples_leaf': 2,
'min_samples_split': 5}

Accuracy: 0.2850
Macro Precision: 0.2818
Macro Recall: 0.2869
Macro F1 Score: 0.2833
```

Initially, the decision tree classifier was experimented with no parameters, hence the default values were used as parameters. However, the results were so poor that (accuracy ≤ %20) an arbitrary guess would have almost the same accuracy. Therefore, besides data augmentation; fine-tuning for hyperparameters and a cross-validation for evaluation was required. For this purpose, *scikit-learn GridSearchCV* was utilized; which tries all hyperparameter combinations exhaustively and uses a training dataset within k-fold cross-validation for evaluating each hyperparameter combination. After the grid search and cross-validation combination, the best hyperparameters were found to be the ones in the above cell. Consequently, accuracy increased to %28.5 which is much better but not that good considering other methods' accuracy. Moreover, ROC curves depict that "*Nodule*" and *"Effusion"*, are the most truly predicted classes among all and the confusion matrix shows that *"Atelectasis"* and *"Effusion"* are rather well discriminated from each other, which is a hard task to do in real life. Nevertheless, the relatively low accuracy shows the need for stronger methods like ensemble learning.

4.1.6 Support Vector Machine

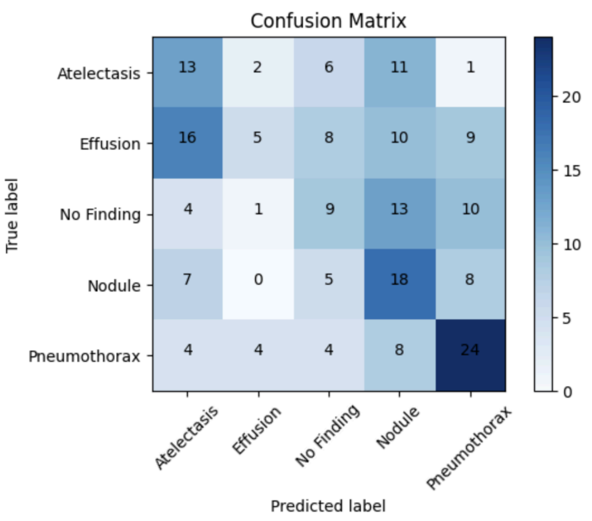


Fig. 16: Confusion Matrix of SVM

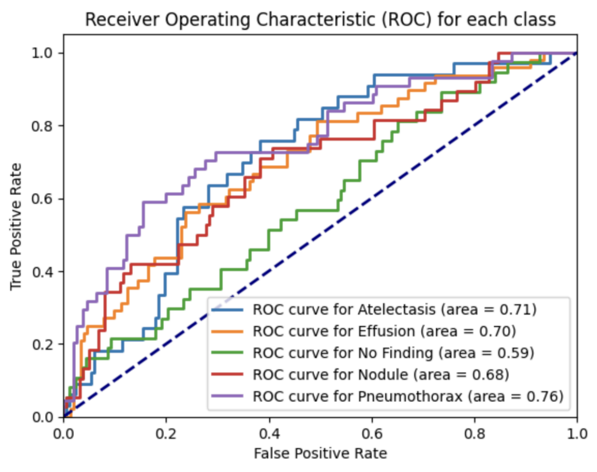


Fig. 17: ROC Curve for SVM

```
Accuracy: 0.345
Macro Precision: 0.3510
Macro Recall: 0.3521
Macro F1 Score: 0.3265
```

In the context of medical imaging, such as classifying diseases from chest X-rays, SVM can be particularly useful due to its ability to handle high-dimensional data and its effectiveness in cases where the number of dimensions exceeds the number of samples. However, the success of SVM heavily depends on the selection of the kernel function and its parameters, which can significantly affect the outcome of the model.

For instance, in a scenario where SVM was applied to classify diseases from chest X-rays and yielded an accuracy of 0.345, this result suggests challenges in the model's performance. Such a low accuracy could stem from various issues such as insufficient training data, poor choice of kernel, the presence of noisy or unrepresentative training samples, or the need for more sophisticated feature extraction techniques to capture relevant information from the X-rays. Addressing these issues could involve collecting more data, experimenting with different kernels (like linear, polynomial, or radial basis function), and enhancing pre-processing steps to improve model accuracy.

4.1.7 Logistic Regression

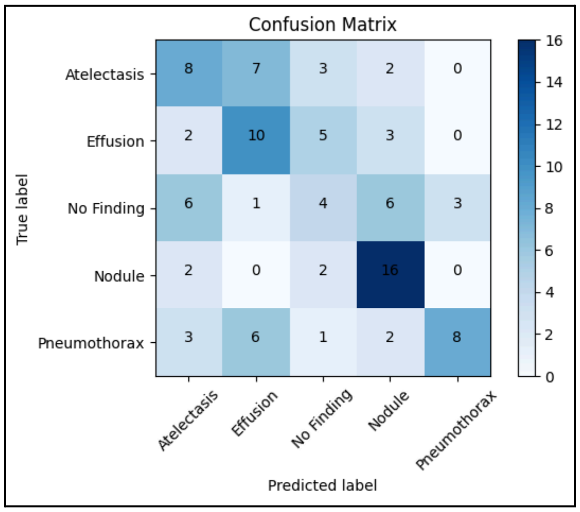


Fig. 18: Confusion Matrix of Logistic Regression

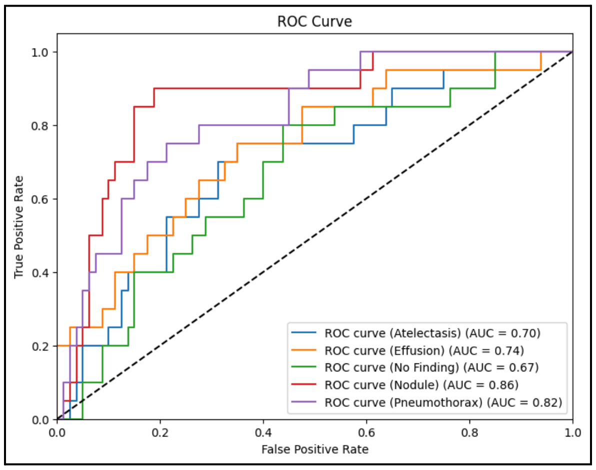


Fig. 19: ROC Curve for Logistic Regression

```
Best Hyperparameters: {'C': 1}

Accuracy: 0.3450
Macro Precision: 0.3419
Macro Recall: 0.3470
Macro F1 Score: 0.3364
```

Logistic regression, known for its simplicity and interpretability, often serves as a baseline model due to its ease of implementation. Despite its simplicity, it delivered impressive results, as evidenced by the achieved accuracy of 34.50% with optimal hyperparameters (C=1) which is searched by GridSearchCV (among 'C': [0.001, 0.01, 0.1, 1, 10, 100] and 5-fold cross-validation used for evaluation). Its straightforwardness makes it serve as a benchmark for more complex models and other traditional algorithms we implemented.

While the logistic regression may lack the sophistication of ensemble methods like AdaBoost or even the decision tree itself; the model's performance, as indicated by macro precision, recall, and F1 score metrics, suggests a reasonable level of predictive capability which is higher than both decision tree and AdaBoost.

The ROC curve shows that the classes best performed in prediction were “Nodule” and “Pneumothorax” and the confusion matrix (in heatmap format) is relatively better compared to decision trees and AdaBoost as the colors get darker through the diagonal.

4.2 Neural Network Methods

4.2.1 Custom CNN Models

4.2.1.1 Custom CNN 1

This model consists of three convolutional layers followed by ReLU activation functions and max-pooling layers, culminating in three fully connected layers with ReLU activation and a softmax output layer. As an optimizer, we used Adam since it performed better than SGD for this model. AUC-ROC values suggest that the Nodule class is the best-classified class for this classifier. The confusion matrix shows that this model is a bit biased towards the class Atelectasis.

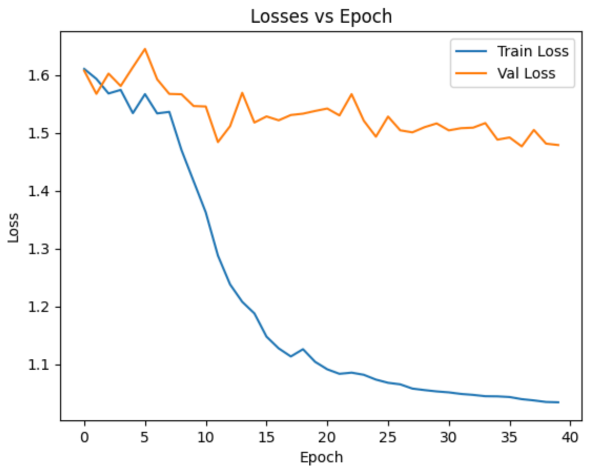


Fig. 20: Losses vs Epoch for Custom CNN1

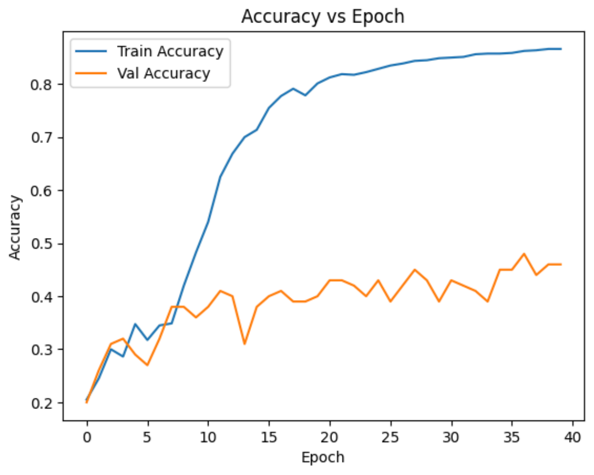


Fig. 21: Accuracy vs Epoch for Custom CNN1

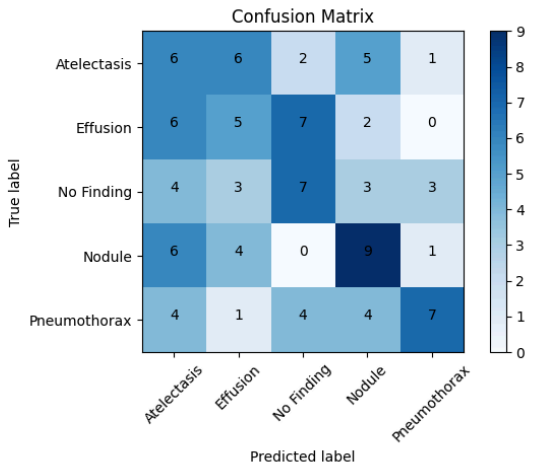


Fig. 22: Confusion Matrix for Custom CNN1

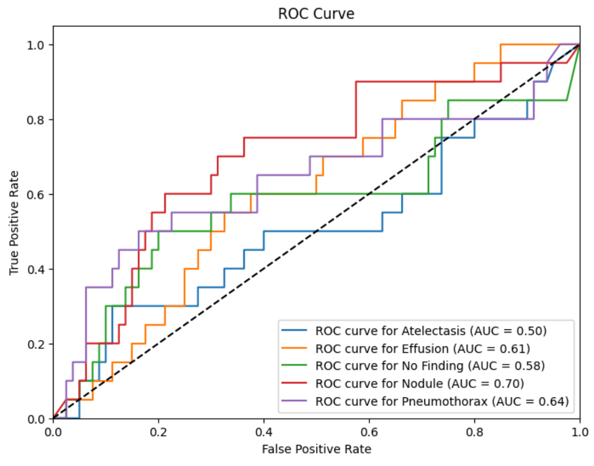


Fig. 23: ROC Curve for Custom CNN1

```
Mean Loss: 1.5203
Accuracy: 0.3400
Macro Precision: 0.3637
Macro Recall: 0.3400
Macro F1 Score: 0.3447
```

4.2.1.2 Custom CNN 2

This model consists of the same layers as the previous one. We changed the sizes of the convolutional layers and added one more fully connected layer to the end of the model. This model resulted in better accuracy compared to the previous one. The confusion matrix shows that this model is less biased than the previous one.

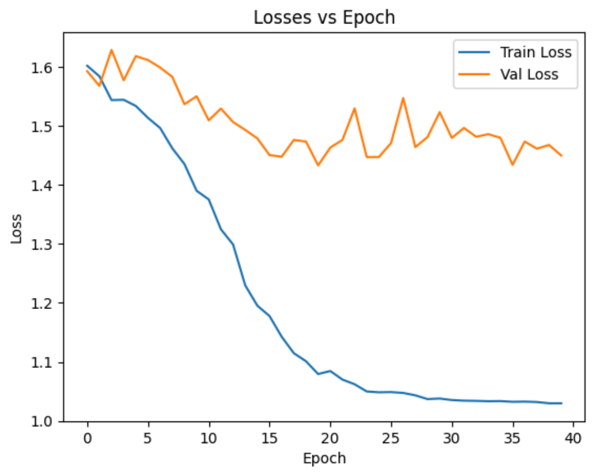


Fig. 24: Losses vs Epoch for Custom CNN2

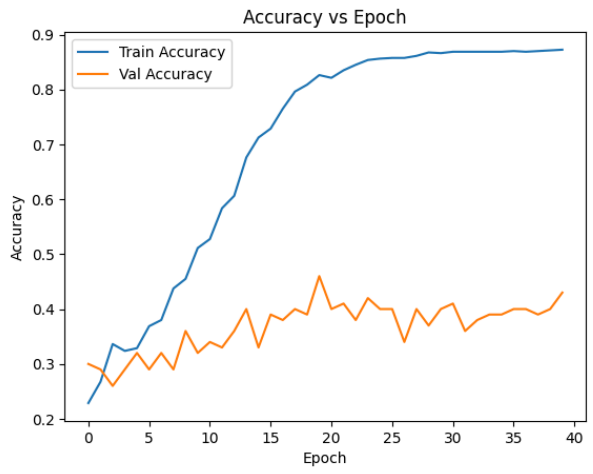
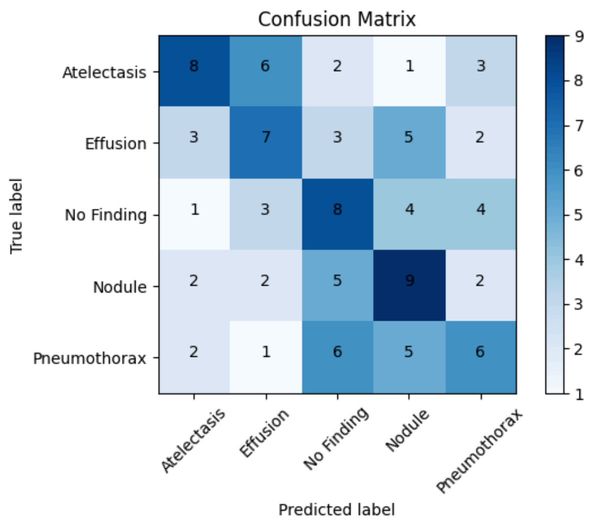


Fig. 25: Accuracy vs Epoch for Custom CNN2



True label \ Predicted label	Atelectasis	Effusion	No Finding	Nodule	Pneumothorax
Atelectasis	8	6	2	1	3
Effusion	3	7	3	5	2
No Finding	1	3	8	4	4
Nodule	2	2	5	9	2
Pneumothorax	2	1	6	5	6

Fig. 26: Confusion Matrix for Custom CNN2

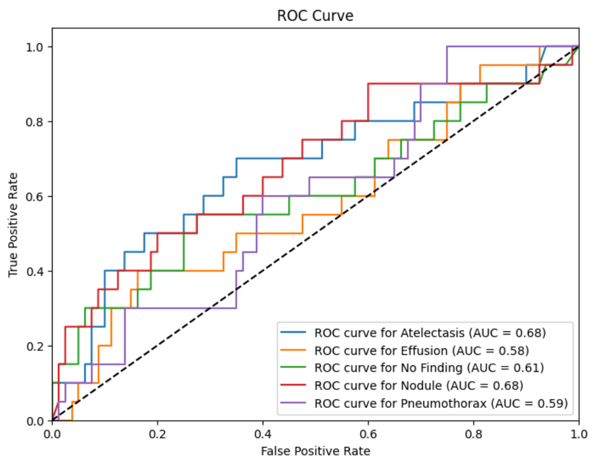


Fig. 27: ROC Curve for Custom CNN2

```
Mean Loss: 1.15111
Accuracy: 0.3800
Macro Precision: 0.3859
Macro Recall: 0.3800
Macro F1 Score: 0.3801
```

4.2.2 MobileNet

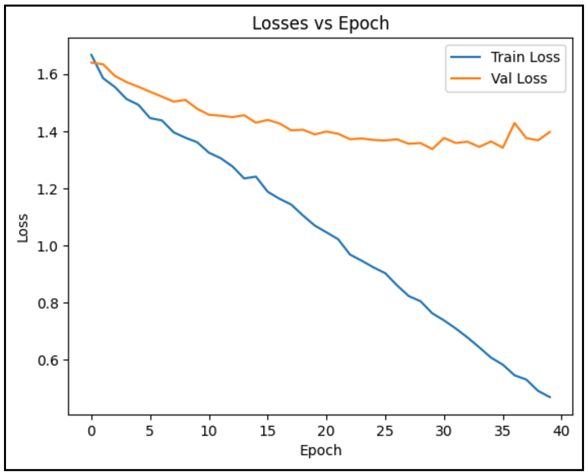


Fig. 28: Losses vs Epoch for MobileNet

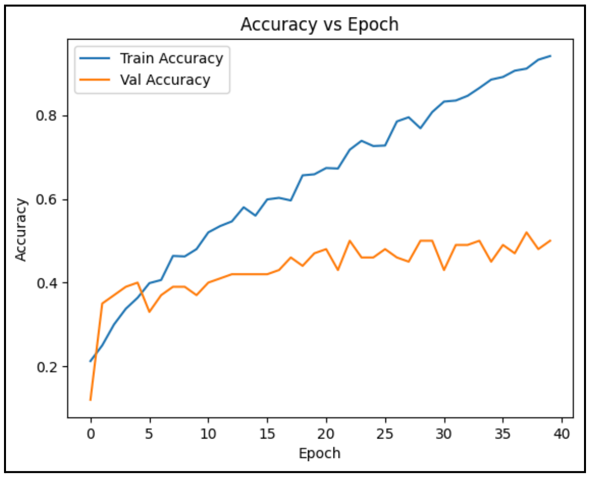


Fig. 29: Accuracy vs Epoch for MobileNet

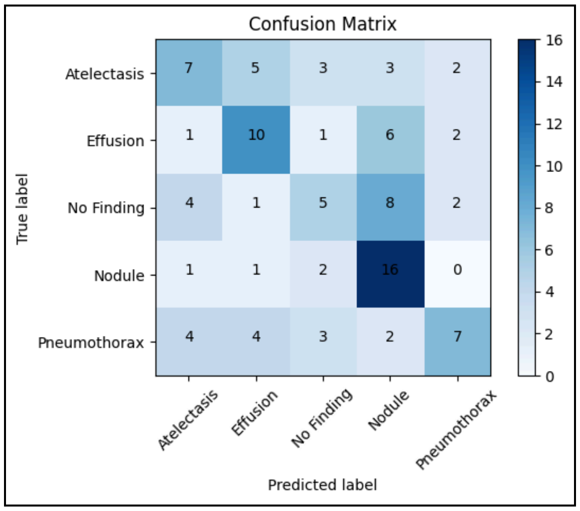


Fig. 30: Confusion Matrix for MobileNet

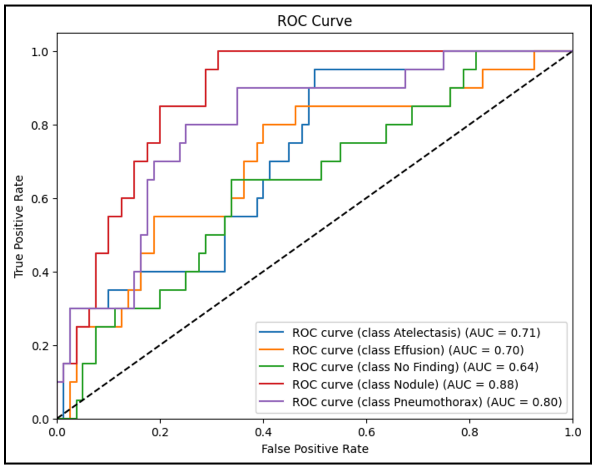


Fig. 31: ROC Curve for MobileNet

```
Mean Loss: 1.3715
Accuracy: 0.4500
Macro Precision: 0.4481
Macro Recall: 0.4500
Macro F1 Score: 0.4333
```

We observe a clear divergence between the training and validation loss, indicating that the model learns effectively from the training data. The accuracy graph confirms the model's struggle to perform consistently on unseen data. For better accuracy we found the learning rate using the Adam optimizer and tested the learning rate both on SGD and Adam optimizers which resulted in a final accuracy of 0.4500 underscores a moderate ability of the model to correctly classify diseases from X-rays. The model has strong performance in detecting 'Nodule' but poorer performance in other diseases.

4.2.3 ResNet

4.2.3.1 ResNet with Pretrained ImageNet-18 Weights

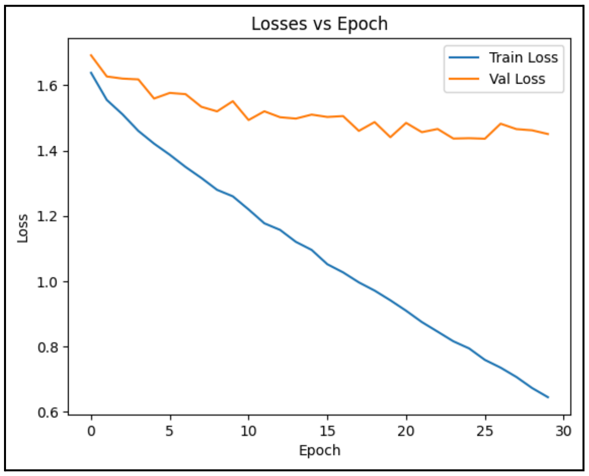


Fig. 32: Losses vs Epoch for ResNet

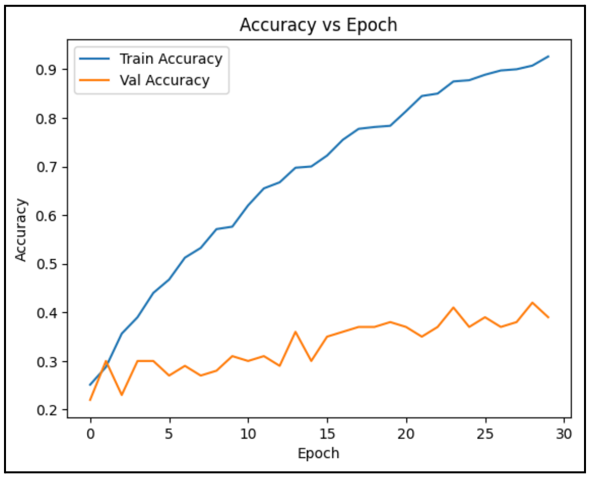


Fig. 33: Accuracy vs Epoch for ResNet

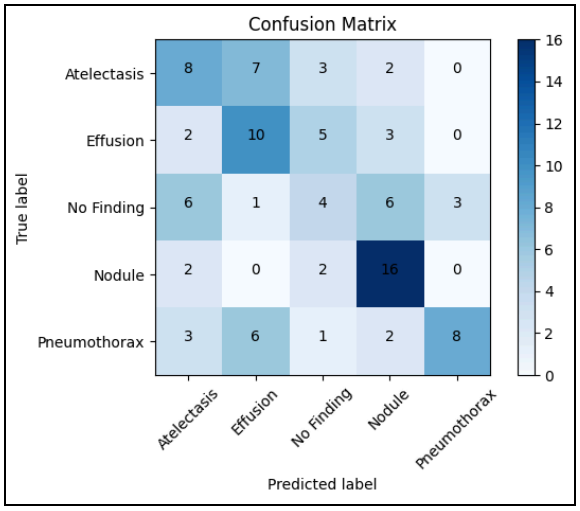


Fig. 34: Confusion Matrix for ResNet

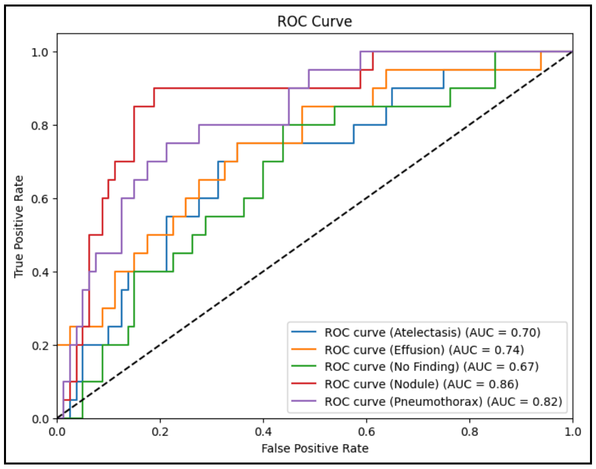


Fig. 35: ROC Curve for ResNet

```
Accuracy: 0.4600
Macro Precision: 0.4687
Macro Recall: 0.4600
Macro F1 Score: 0.4485
```

Initially, ResNet was experimented with ImageNet-18 pre-trained weights for observing the transfer learning method. For optimizers, both Adam and SGD optimizers were used for training separately. SGD optimizer with learning rate 1e-3 performed the best where Adam failed to obtain a better test accuracy upon multiple experiments. Compared to other CNN and transfer learning models, ResNet has performed impressively well. The ROC curve shows that the classes best performed in prediction were “Nodule” and “Pneumothorax” and the confusion matrix supports that finding as well as depicting that ResNet is rather good in terms of discriminating “Atelectasis” and “Effusion” diseases, which is a hard task to do in real life.

For experimenting the model without pretrained weights and data augmentation, the following experiments were also conducted yet the model with pre-trained weights and no data augmentation performed best for this task. Therefore we continued with the same strategy for other transfer learning methods too.

4.2.3.2 ResNet without Pre-trained Weights

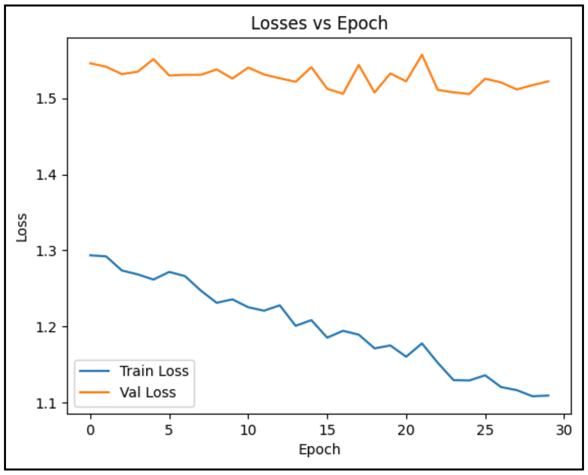


Fig. 36: Losses vs Epoch for ResNet

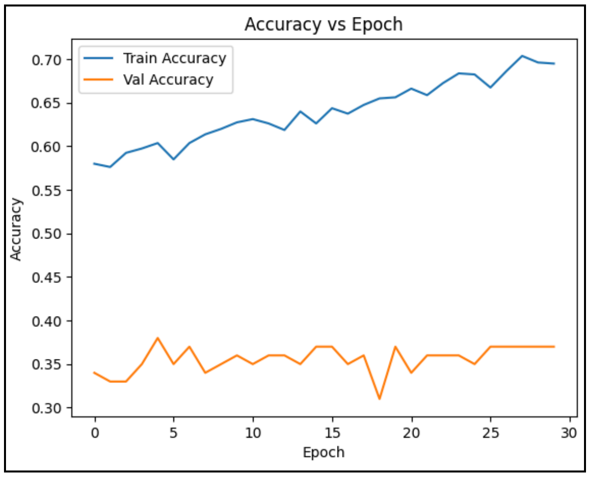


Fig. 37: Accuracy vs Epoch for ResNet

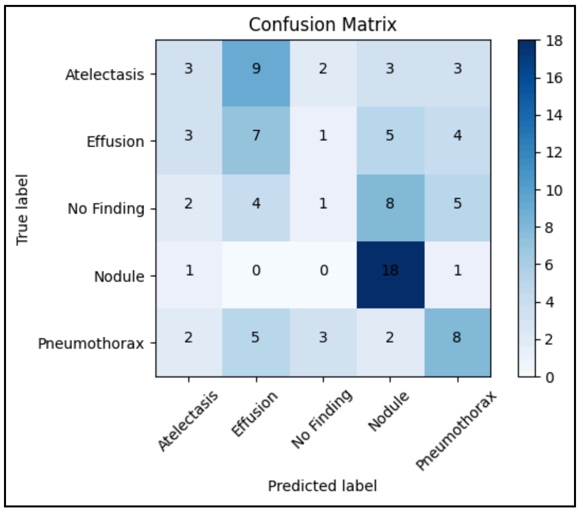


Fig. 38: Confusion Matrix for ResNet

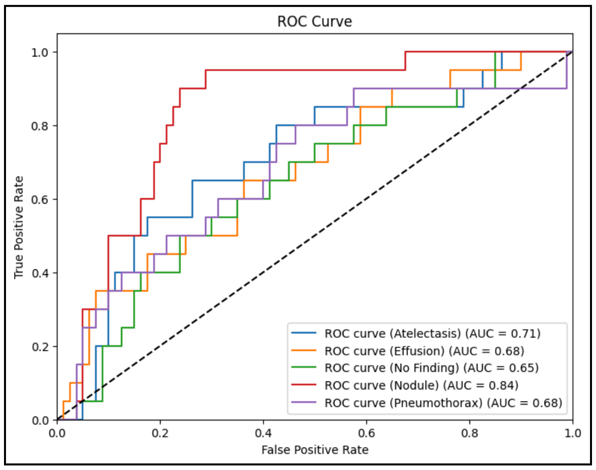


Fig. 39: ROC Curve for ResNet

```
Accuracy: 0.3700
Macro Precision: 0.3153
Macro Recall: 0.3700
Macro F1 Score: 0.3224
```

Secondly, ResNet was experimented without pre-trained weights for observing the models performance instead of transfer learning. For optimizers, both Adam and SGD optimizers were used for training separately. SGD optimizer with learning rate 1e-3 performed the best where Adam failed to obtain a better test accuracy upon multiple experiments. Compared to the ResNet with pre-trained weights, it has performed poorly.

The ROC curve shows that the classes best performed in prediction were “Nodule” and “Atelectasis” which is different then the pre-trained version. And the confusion matrix shows that ResNet without pre-trained weights is much worse in terms of discriminating “Atelectasis” and “Effusion” diseases compared to pre-trained version.

4.2.3.3 ResNet with Pre-trained ImageNet-18 Weights and Data Augmentation

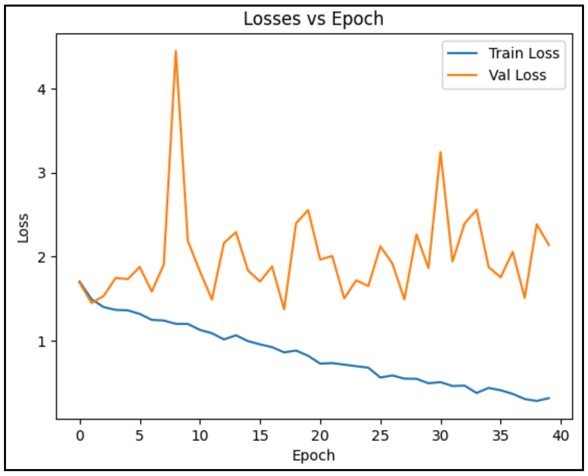


Fig. 40: Losses vs Epoch for ResNet

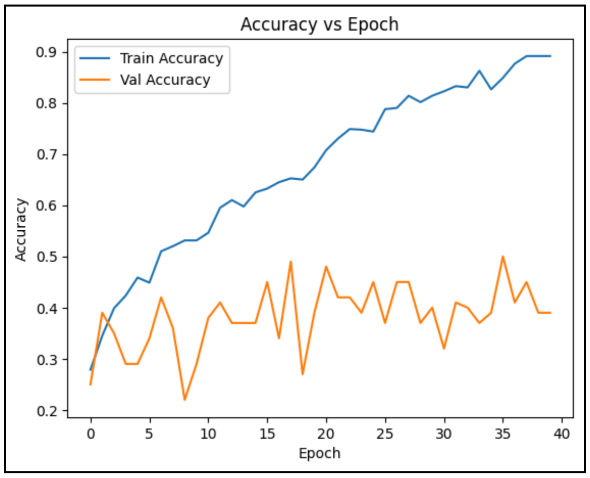


Fig. 41: Accuracy vs Epoch for ResNet

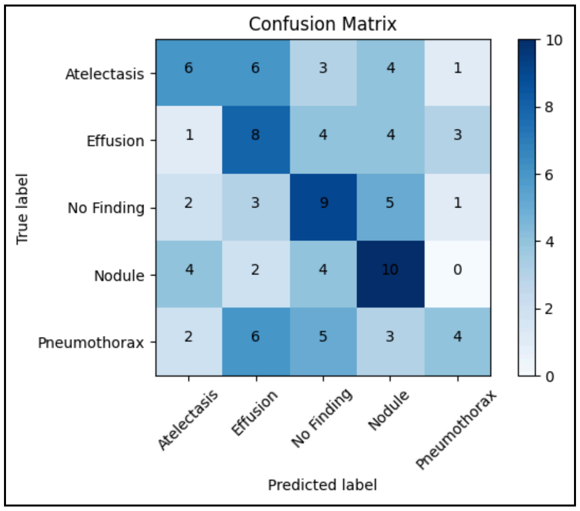


Fig. 42: Confusion Matrix for ResNet

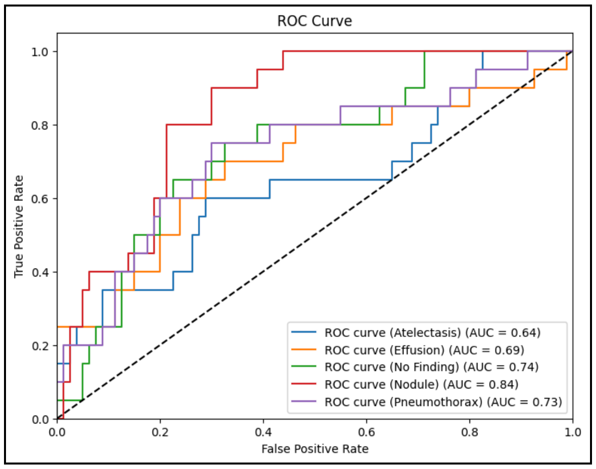


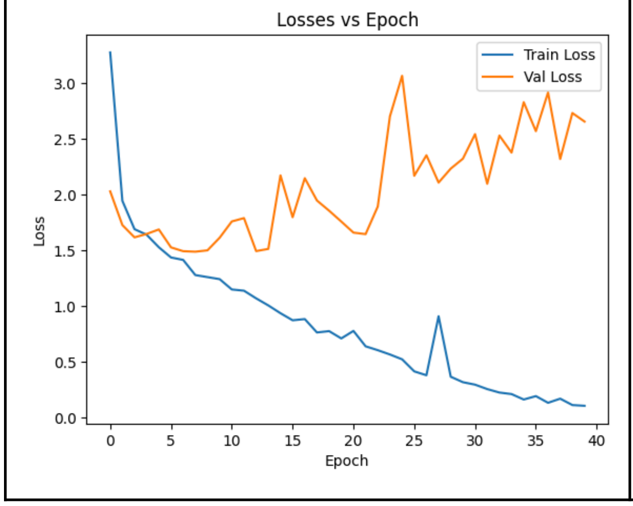
Fig. 43: ROC Curve for ResNet

```
Accuracy: 0.3700
Macro Precision: 0.3818
Macro Recall: 0.3700
Macro F1 Score: 0.3618
```

Finally, ResNet was experimented with pre-trained weights and data augmentation (RandomHorizontalFlip, RandomRotation(10), ColorJitter(brightness=0.1, contrast=0.1, saturation=0.1, hue=0.05)) for observing whether the model would benefit from augmented data by increased test accuracy. For optimizers, both Adam and SGD optimizers were used for training separately. SGD optimizer with learning rate 1e-3 performed the best where Adam failed to obtain a better test accuracy upon multiple experiments. Compared to the plain ResNet with pre-trained weights, it has performed poorly (nearly same performance without pretrained weights version). The ROC curve shows that the classes best performed in prediction were “Nodule” and “No Finding” which is different then the other versions. And the confusion matrix shows that ResNet with data augmentation is worse in terms of discriminating “Atelectasis” and “Effusion” diseases compared to pre-trained versions. In conclusion, ResNet has not benefited from data augmentation techniques as the accuracy has dropped. Moreover the computational load brought by the technique is not worth at all. Following these results, we decided to not utilize data augmentation with other transfer learning methods too and we avoided a lot of computation time by this strategy. Additionally, data augmentation might not be a good strategy for this specific task considering data consist of x-ray images which are very sensitive to small details and augmented data may not represent reality.

4.2.4 VGG19

For VGG19, we tried both Adam and SGD optimizers. SGD optimizer with learning rate 1e-3 performed the best. As AUC-ROC values show, Nodule is the best-classified class for VGG19. The confusion matrix suggests that VGG19 has a bias towards Effusion, resulting in lower performance for Effusion. Also, the model is struggling to differentiate Atelectasis and Effusion, which is a hard task also in real life.



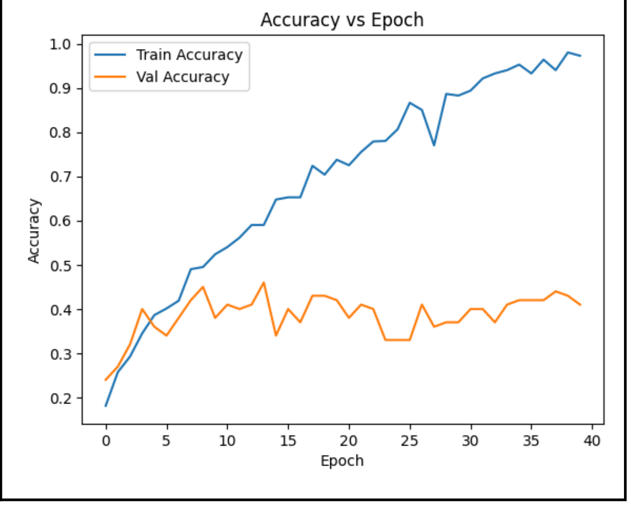
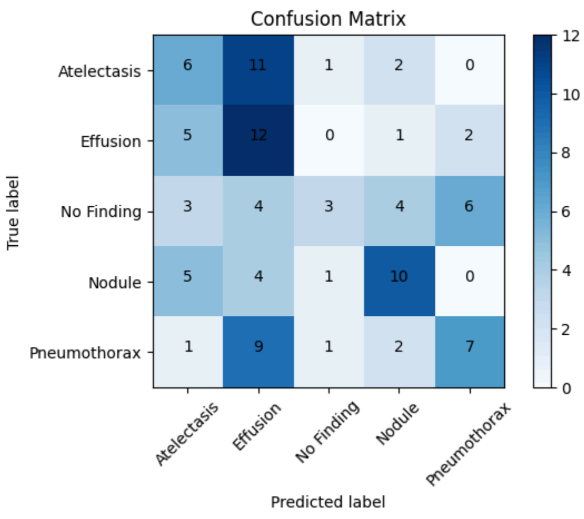
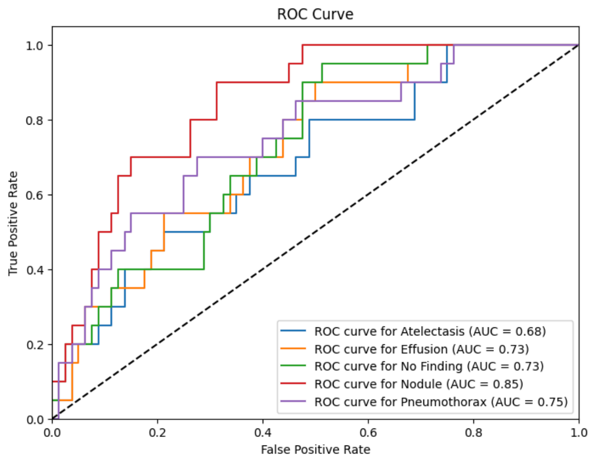


Fig. 44: Losses vs Epoch VGG19	Fig. 45: Accuracy vs Epoch for VGG19
 Fig. 46: Confusion Matrix for VGG19	 Fig. 47: ROC Curve for VGG19

```
Mean Loss: 1.3296
Accuracy: 0.3800
Macro Precision: 0.4186
Macro Recall: 0.3800
Macro F1 Score: 0.3687
```

4.2.5 GoogLeNet (Inception)

The training of the Inception model on our dataset is performed by first using the pretrained weights and training the model from scratch. Interestingly, the Inception model performed better in the latter case. In the former case, the model was very biased towards the nodule class with accuracy rate 0.26. However, initializing weights from scratch yielded improvement with accuracy rate 0.36. Nevertheless, the model became biased towards the “no finding” class this time. Finally, we can say that using only the neural network architecture instead of transfer learning for the Inception case is a better choice.

4.2.5.1 Inception without Pre-training

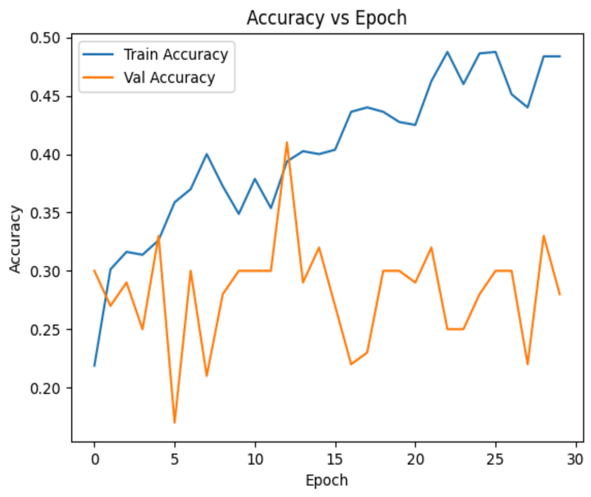


Fig. 48: Accuracy vs. Epoch for Inception

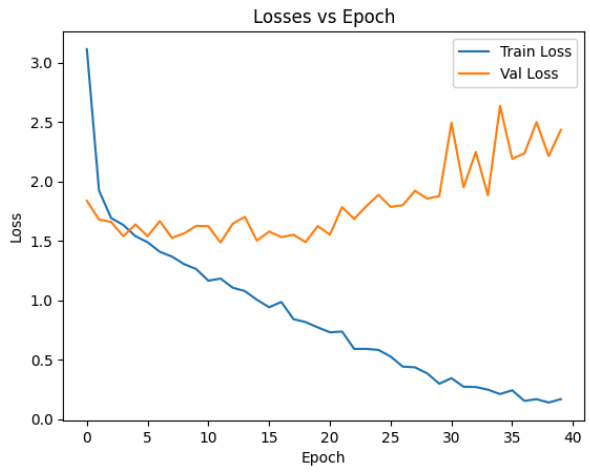
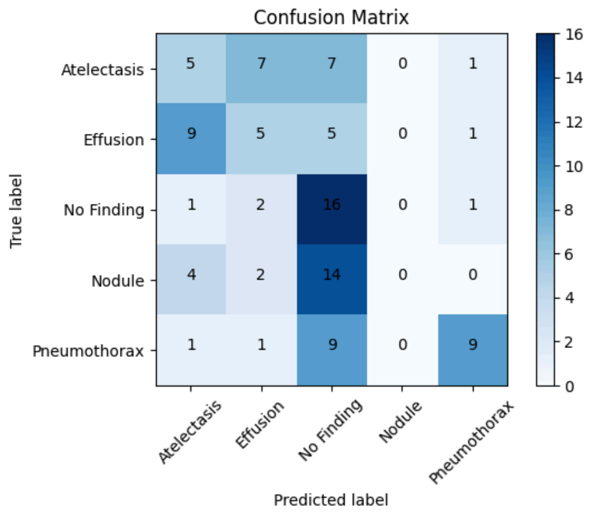


Fig. 49: Losses vs. Epoch for Inception



	Atelectasis	Effusion	No Finding	Nodule	Pneumothorax
Atelectasis	5	7	7	0	1
Effusion	9	5	5	0	1
No Finding	1	2	16	0	1
Nodule	4	2	14	0	0
Pneumothorax	1	1	9	0	9

Fig. 50: Confusion Matrix of Inception

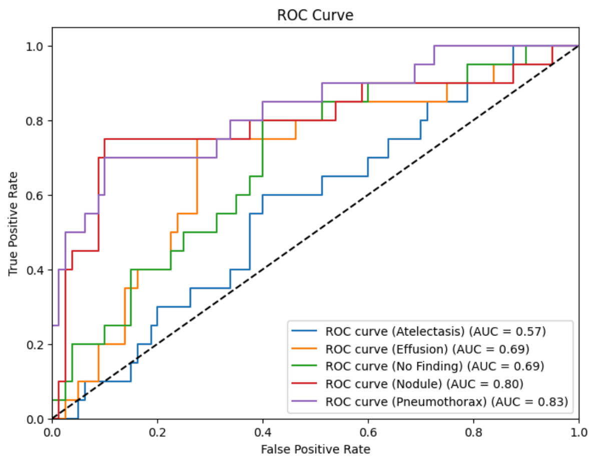


Fig. 51: ROC Curve of Inception

```
Mean Loss: 1.4938
Accuracy: 0.3500
Macro Precision: 0.3216
Macro Recall: 0.3500
Macro F1 Score: 0.3067
```

4.2.5.2 Inception with Pre-Training

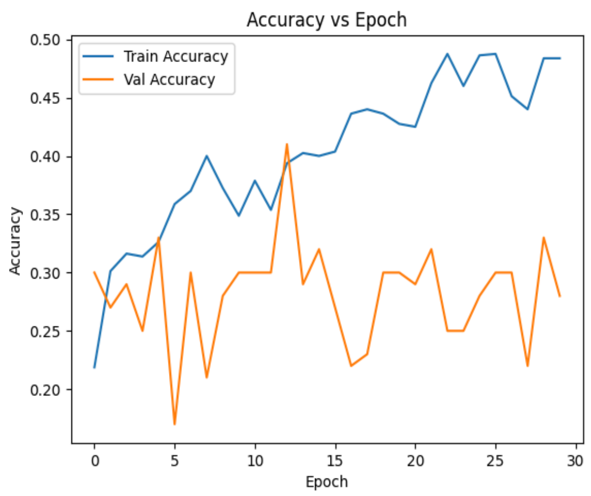


Fig. 52: Accuracy vs. Epoch for Inception

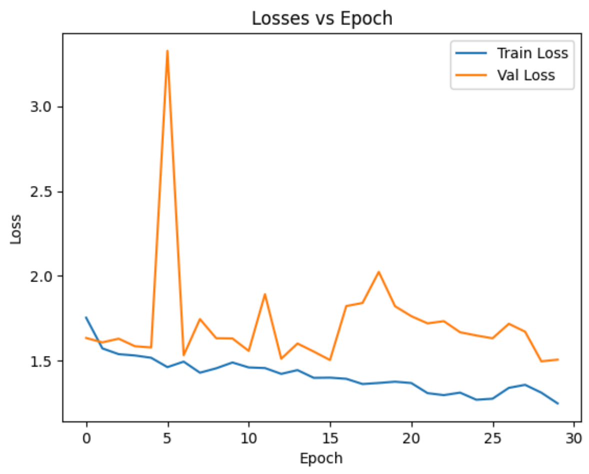
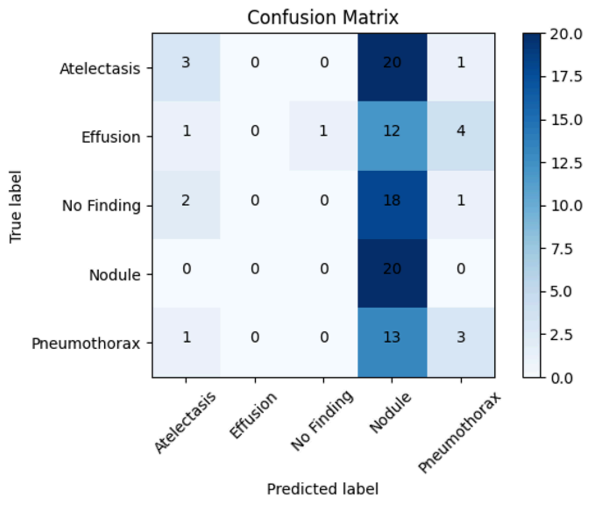


Fig. 53: Losses vs. Epoch for Inception



True label	Atelectasis	Effusion	No Finding	Nodule	Pneumothorax
Atelectasis	3	0	0	20	1
Effusion	1	0	1	12	4
No Finding	2	0	0	18	1
Nodule	0	0	0	20	0
Pneumothorax	1	0	0	13	3

Fig. 54: Confusion Matrix of Inception

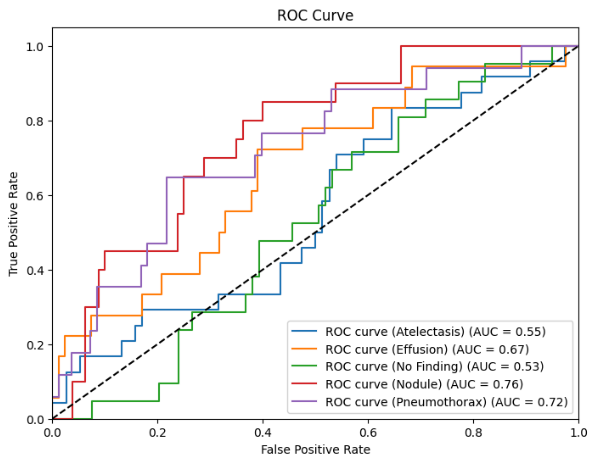


Fig. 55: ROC Curve of Inception

```
Mean Loss: 1.7313
Accuracy: 0.2600
Macro Precision: 0.2006
Macro Recall: 0.2603
Macro F1 Score: 0.1625
```

4.2.6 DenseNet 121

DenseNet-121 offers a compelling combination of parameter efficiency, feature reuse, gradient flow, and information flow, making it a popular choice for various computer vision tasks. For our case it seems to be a good choice by looking at the ROC-AUC curves for pneumothorax, nodule and effusion. The high value of the AUC reflects on the success of classifying these diseases, which can be seen from the confusion matrix as well.

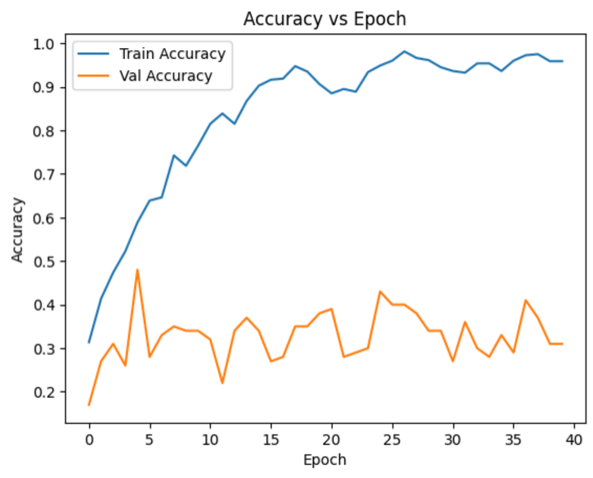


Fig. 56: Accuracy vs. Epoch for DenseNet

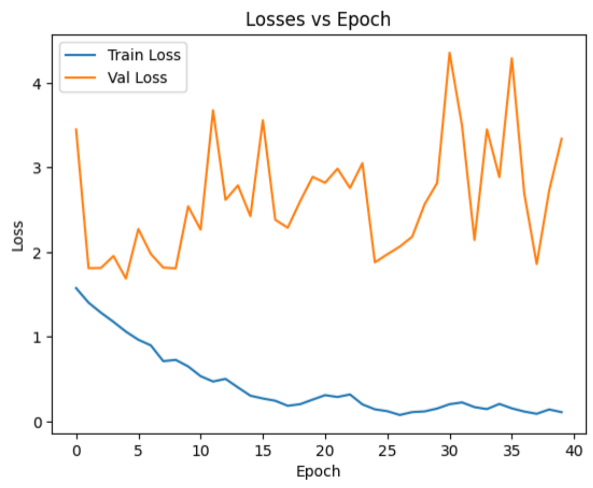


Fig. 57: Losses vs. Epoch for DenseNet 121

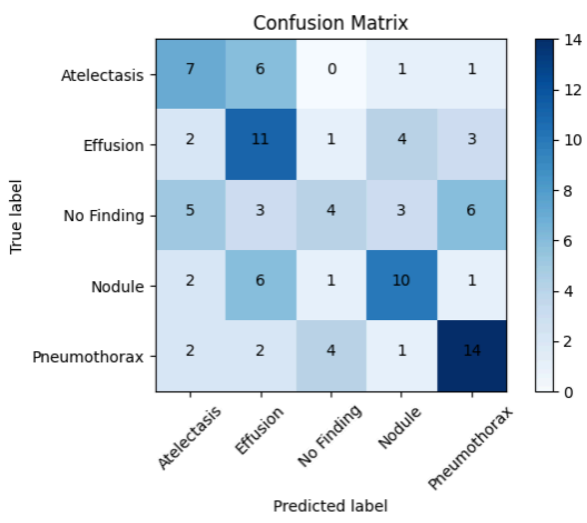


Fig. 58: Confusion Matrix for DenseNet 121

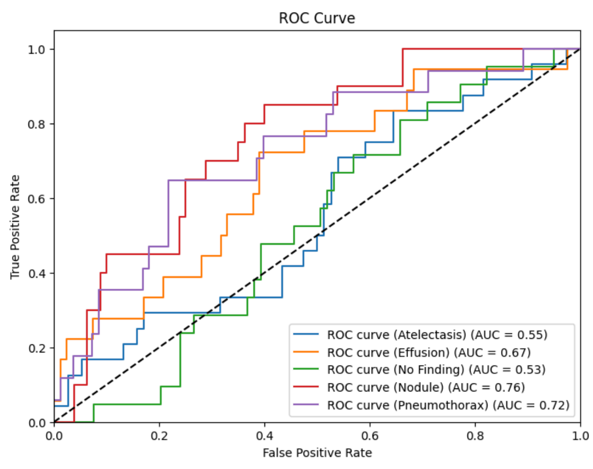


Fig. 59: ROC Curve for DenseNet121

```
Mean Loss: 1.2942
Accuracy: 0.4600
```

```
Macro Precision: 0.4536
Macro Recall: 0.4579
Macro F1 Score: 0.4455
```

5.0 Conclusions

Motivated by the aim of understanding the fact that to what extent traditional learning methods and neural network architectures can be effective in overcoming the limitations of X-ray imaging and distinguishing the various chest diseases, we tried various methods illustrated in Fig. 64.

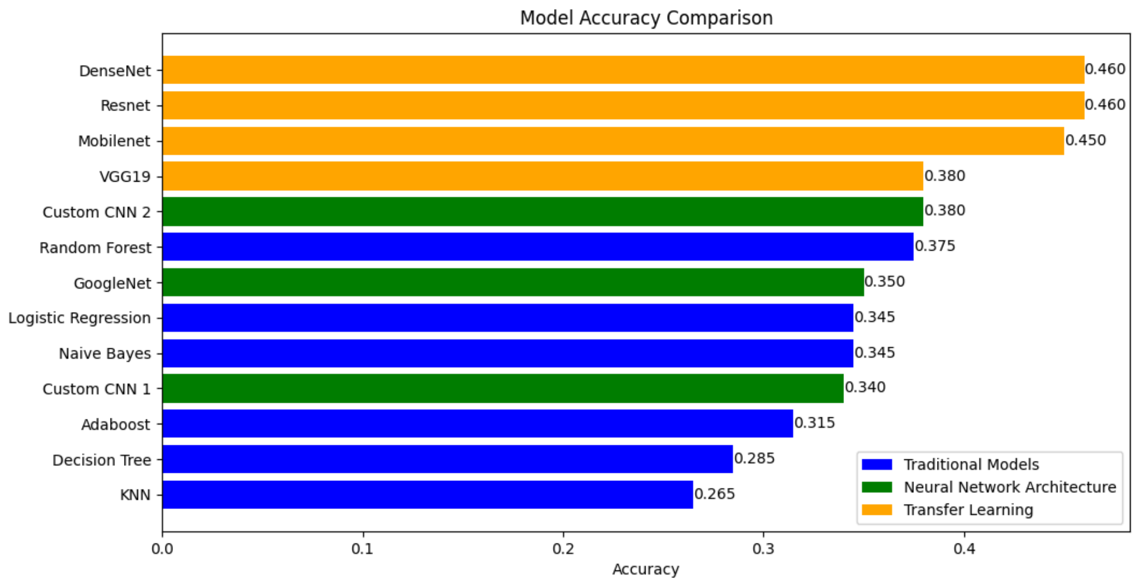


Fig 60. Model Accuracy Comparison

Recalling one of the questions, that is, how traditional learning methods and neural network architectures can be compared regarding accuracy, the above figure presents profound insight. It is evident that transfer learning outperformed neural network architectures trained from scratch and traditional methods. We can also claim that neural network architectures are more reasonable for our task compared to traditional models due to the ability of CNNs to automatically learn hierarchical features, exploit local spatial correlations, handle translation invariance, and benefit from regularization techniques that make them more suitable for image classification tasks.

In addition, one can wonder why GoogleNet is not labeled as a transfer learning method. The reason is that GoogleNet model without pre-training performed better, preventing it from becoming the transfer learning method for our case.

The findings of this project may indicate that commonly used transfer learning methods, NN architectures and traditional learning methods are insufficient in classifying X-ray images. Despite observing a hierarchy "Transfer Learning > Neural Learning Architecture > Traditional Methods" from Fig. 64, the results of Transfer learning are not optimal, with a maximum of 46% accuracy. This implies another fact in medicine: X-ray images may not be adequate for detecting a disease by itself. In medicine, doctors rely on other data of patients and further imaging techniques besides X-ray images for disease detection [2]. In the introduction, we referred to this situation before the project was implemented. The numerical results confirm the theoretical complexities behind classifying X-ray images.

Secondly, a more convincing argument for the results insufficiency could be that the dataset is not convenient for this project's purpose. Indeed, even though the dataset is very large, due to the reasons explained in the Dataset section, we could come up with a very small dataset. Furthermore, being suspicious about the correctness of the labels, we further conducted a live test with a 30-years-experienced chest diseases specialist, to whom we showed 10 pictures from our final dataset including 2 pictures from each class and she was able to identify very few of them, actually 1. This test clearly showed that we are right in our suspicions about the dataset. Moreover, such an incidence also validates the results we obtained.

6.0 Appendixes

Appendix A - Contributions

Arda İynem: I contributed to the initial implementations of all traditional methods, then I was responsible for the betterment of the Logistic Regression, Decision Tree, AdaBoost. I also implemented the ResNet with and without pretrained weights and data augmented version and contributed to the final report.

Hasan Ege Tunç: I contributed to the Final Report by writing the sections Introduction, Problem Description, Dataset section of Methods, and Conclusion, and implemented DenseNet, KNN, and GoogleNet models.

Enes Bektaş: I contributed to the implementation of custom CNN model 1 & 2, VGG19 and random forest. I also contributed to the preparation of the presentation, and sections 3.2, 4 and 5 of the final report.

Ömer Fırat Bekiroğlu: I contributed in proposal report, model development of progress report, preparation of presentation final report chapters 3, 4.1.1 & 4.2.2, and Implemented MobilNet mode and Naive Bayes classifier,

Sabri Eren Dağdelen: Implementation, training, testing and plotting of AlexNet and Support Network Machine. Contribution to final report and presentation documents.

7.0 References

[1] C. Crawford. “NIH Chest X-rays.” kaggle. https://www.kaggle.com/datasets/nih-chest-xrays/data/data (accessed May 12, 2024).

[2] Tunç, M. (March 10, 2024). Personal interview.

[3] ImageNet. https://www.image-net.org/(accessed May 12, 2024)

[4] C. Crawford. “NIH Chest X-rays/README.pdf” kaggle. https://www.kaggle.com/datasets/nih-chest-xrays/data/data (accessed May 12, 2024).

[5] Nabil, M. (2023, October 17). *Unveiling the diversity: A comprehensive guide to types of CNN Architectures*. Medium. https://medium.com/@navarai/unveiling-the-diversity-a-comprehensive-guide-to-types-of-cnn-architectures-9d70da0b4521 (accessed May 12, 2024).

[6] A. Sarkar . “Creating DenseNet 121 with TensorFlow” Medium. https://towardsdatascience.com/creating-densenet-121-with-tensorflow-edbc08a956d8 (accesed May 12, 2024)