COVID-19 Detection using Chest X-ray Images

॥ त्वं ज्ञानमयो विज्ञानमयोऽसि ॥

CSL2020: Pattern Recognition and Machine Learning

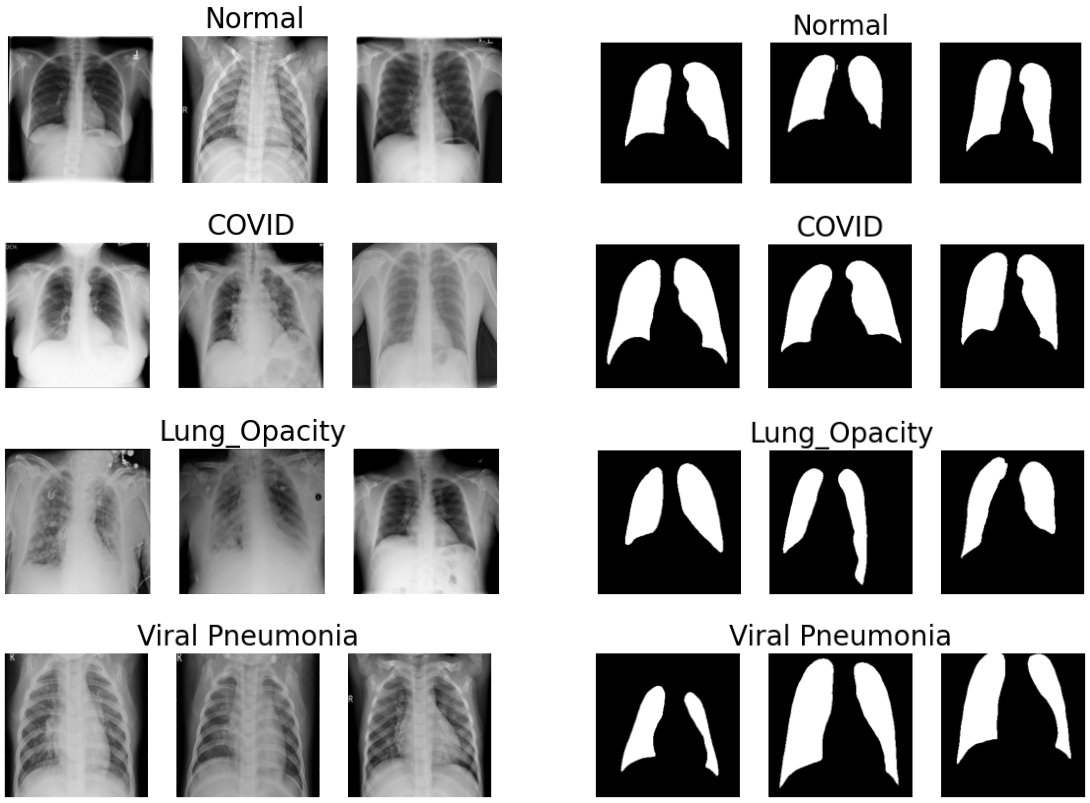
Full Name	Roll Number
Niharika Dadu	B21CS052
Amisha Kumari	B21CS007
Harshil Kaneria	B21CS033

1 | Abstract

The Coronavirus Disease 2019 (COVID-19) has brought a worldwide threat to the living society. One of the areas where machine learning can help is detecting COVID-19 cases from chest X-ray images. The task is a simple classification problem where given an input chest X-ray image, the machine learning-based model must detect whether the subject of study has been infected or not.

2 | Dataset

The COVID-19 Radiography Database contains a total of 21165 X-Ray images out of which 10192 samples represent chest x-rays of healthy individuals, 3616 represents an affliction with COVID-19 virus, 6012 images represent Lung opacity and 1345 images represent Viral pneumonia. Below, we present some examples from our data:



(a) Chest X-Rays without masks **(b)** Chest X-Rays with masks

Figure 2.1: Visualization of Chest X-rays

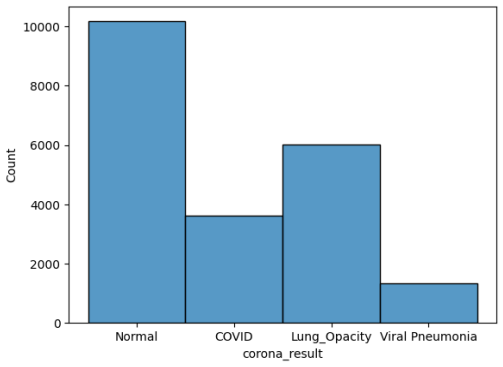
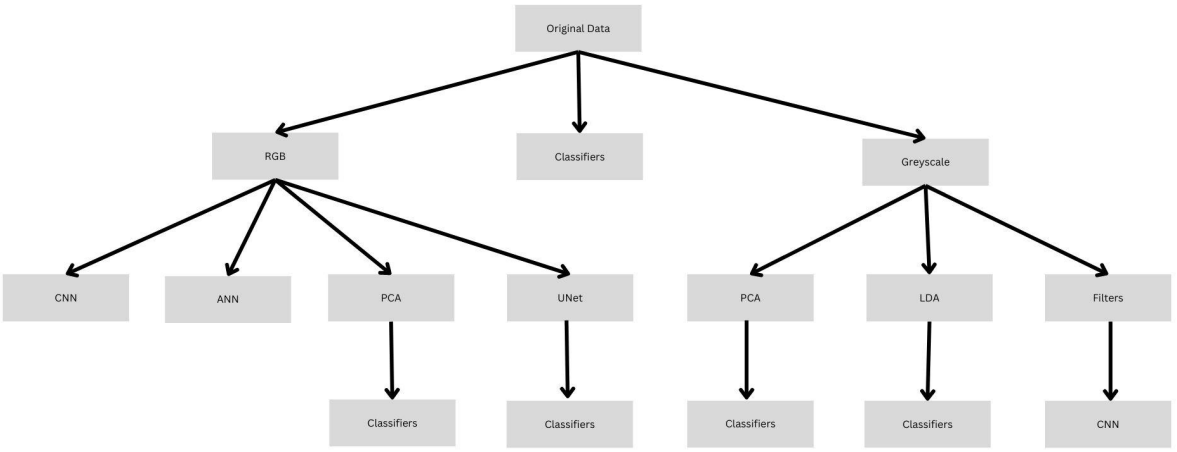
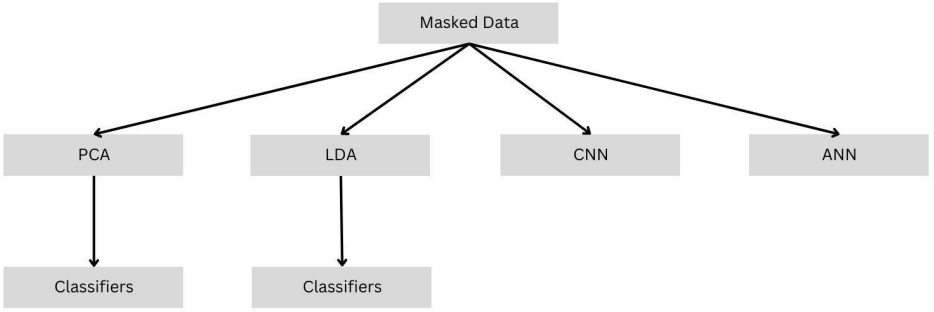


Figure 2.2: Bar plot of the dataset showing number of images in each class.

NOTE: We have provided a method to download the data from kaggle to the google drive directly to make it easier for the user to run the colab scripts on the downloaded data. This has been provided so that the user does not have to download and upload the dataset again and again. This can be seen in the Dataset Downloading and Preparation section of the colab file. This has been done using linux commands.

3 | Model Pipeline





4 | Preprocessing

We have used the popular Python library for image processing, *OpenCV (CV2)* and resize the images to a 50 x 50 resolution. This has been done as the system RAM of the colab was exceeding the original images . We have then performed one hot encoding on the data with images with class COVID as 1 and rest all as class 0 as our problem statement is covid detection. We have also scaled data using *StandardScaler()* from Scikit-learn. Subsequently multiple preprocessing techniques have been used for different classification techniques which are listed follow:

- Used various filters on images like
  - Histogram Equalization
- We have flattened the images and used a Pandas DataFrame for storing the same. In the case of RGB channels we got a numpy array with 7500 features and in the case of grayscale we got 2500 features. Additionally we also have a numpy array target which stores the class of the images.

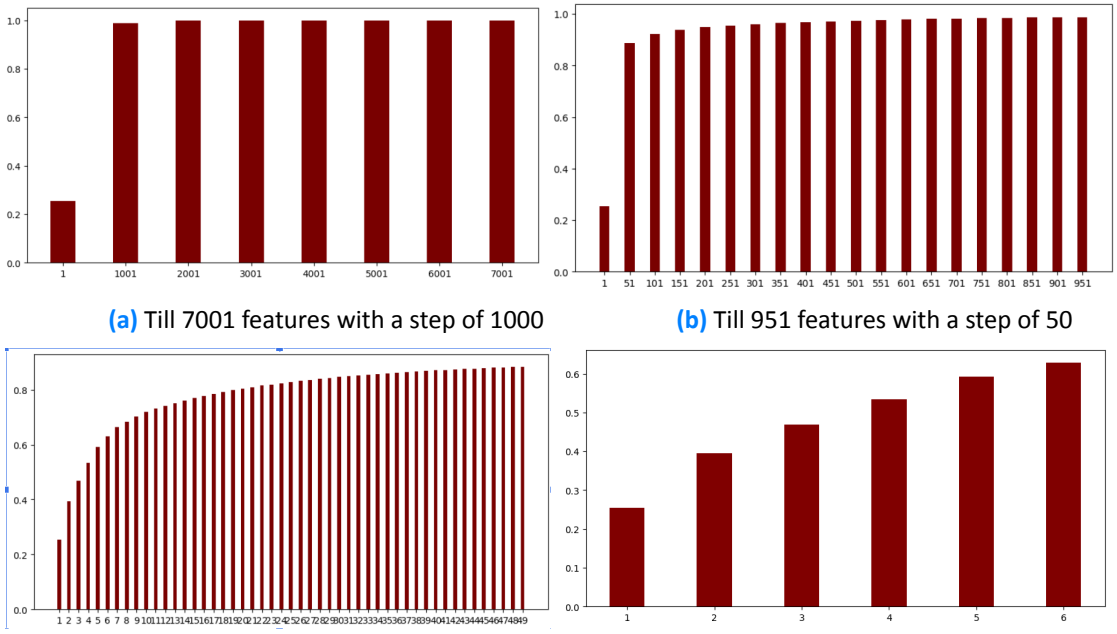
5 | Dimensionality Reduction

A dataset with 7500 features per sample is extremely dimensional data so we employ 3 techniques to reduce the dimensions to a lower space.

- Using Principal Component Analysis (PCA).
- Using Linear Discriminant Analysis.
- Converting RGB channels to Grayscale.

5.1 | Principal Component Analysis (PCA)

We have employed Principal Component Analysis (PCA) to reduce the feature space from 7500 dimensions. From figure 5.1(a), figure 5.1(b) and figure 5.1(c) we infer that the variance of the 30 features is relatively higher. From figure 5.1 (d), 6 features capture around 60% of the variance, hence have transformed our data to a 6 dimensional data using PCA.



(a) Till 7001 features with a step of 1000 **(b)** Till 951 features with a step of 50

(c) Till 49 features with a step of 1 **(d)** Till 6 features with a step of 1

Figure 5.1: Plots to show the optimum value of features using PCA.

5.2 | Linear Discriminant Analysis (LDA)

We have employed Linear Discriminant Analysis(LDA) to find a linear combination of features that maximally separates the classes in the data. The original dataset has 7500 features and only 2 classes. Therefore, the

maximum number of discriminant functions that can be obtained using LDA is min(7500, 2-1) = 1. This implies that LDA can only produce one discriminant function, which can be used to reduce the dimensionality of the dataset from 7500 features to a single feature that represents the most important information for discriminating between the two classes.

5.3 | RGB channels to Grayscale

We have used cv2 to change RGB image to grayscale because cv2 is a popular computer vision library with built-in functions for image processing. The cv2.cvtColor() function in cv2 allows us to convert an RGB image to grayscale quickly and easily. Converting RGB images to grayscale reduces the dimensionality of the data, improves image processing accuracy, and saves storage space.

6 | Splitting of Data

After the preprocessing, both the reduced and original data is split into train and validation data using 70-30 distribution. For this we have used *train_test_split()* provided by scikit-learn.

7 | Models

We have been introduced to the following classifiers:

- Decision Tree Classifier
- Random Forest Classifier
- K Nearest Neighbors
- Logistic Regression
- Gradient Boosting
- XGBoost Classifier
- AdaBoost Classifier
- LightGBM Classifier
- Multi Layer Perceptron (MLP) classifier
- Convolutional Neural Network

8 | Experiments and Results

8.1 | Original Data

We have trained the following classifiers on the original scaled dataset and report the testing accuracy, recall score and classification report in Table 8.1.1.

Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.88 **Recall score:** 0.66	precision recall f1-score support Class 0 0.93 0.93 0.93 4418 Class 1 0.64 0.66 0.65 863 accuracy 0.88 5281 macro avg 0.79 0.79 0.79 5281 weighted avg 0.89 0.88 0.89 5281
Random Forest Classifier	**Accuracy**: 0.92 **Recall score:** 0.56	precision recall f1-score support Class 0 0.93 0.99 0.96 4418 Class 1 0.91 0.60 0.72 863 accuracy 0.92 5281 macro avg 0.92 0.79 0.84 5281

		weighted avg 0.92 0.92 0.92 5281
K Nearest Neighbours Classifier	**Accuracy**: 0.90 **Recall score:** 0.47	precision recall f1-score support Class 0 0.90 0.98 0.94 4418 Class 1 0.82 0.47 0.60 863 accuracy 0.90 5281 macro avg 0.86 0.73 0.77 5281 weighted avg 0.89 0.90 0.88 5281
Logistic Regression	**Accuracy**: 0.85 **Recall score:** 0.47	precision recall f1-score support Class 0 0.90 0.93 0.91 4418 Class 1 0.56 0.47 0.51 863 accuracy 0.85 5281 macro avg 0.73 0.70 0.71 5281 weighted avg 0.84 0.85 0.85 5281
Gradient Boosting	**Accuracy**: 0.90 **Recall score:** 0.50	precision recall f1-score support Class 0 0.91 0.98 0.95 4418 Class 1 0.86 0.50 0.63 863 accuracy 0.91 5281 macro avg 0.89 0.74 0.79 5281 weighted avg 0.90 0.91 0.90 5281
XGBoost	**Accuracy**: 0.95 **Recall score:** 0.78	precision recall f1-score support Class 0 0.96 0.99 0.97 4418 Class 1 0.92 0.78 0.85 863 accuracy 0.95 5281 macro avg 0.94 0.88 0.91 5281 weighted avg 0.95 0.95 0.95 5281
AdaBoost Classifier	**Accuracy**: 0.89 **Recall score:** 0.50	precision recall f1-score support Class 0 0.91 0.96 0.93 4418 Class 1 0.72 0.51 0.60 863 accuracy 0.89 5281 macro avg 0.82 0.73 0.77 5281 weighted avg 0.88 0.89 0.88 5281
LightGBM Classifier	**Accuracy**: 0.92 **Recall score:** 0.61	precision recall f1-score support Class 0 0.93 0.98 0.96 4418 Class 1 0.88 0.61 0.72 863 accuracy 0.92 5281 macro avg 0.91 0.80 0.84 5281 weighted avg 0.92 0.92 0.92 5281

Table 8.1.1: Table to classification report, accuracy and recall score of the classifiers

8.2 | Principal Component Analysis

The Principal Component Analysis is a popular unsupervised learning technique for reducing the dimensionality of data. A dataset with 7500 features per sample is extremely dimensional data so we have PCA to reduce the dimensions to a lower space.

8.2.1 | PCA on the scaled data with RGB channels

We have trained the following classifiers on the PCA with scaled data and report the testing accuracy, recall score and the classification report in Table 8.2.1.1.

Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.93 **Recall score:** 0.17	precision recall f1-score support Class 0 0.97 0.96 0.97 5261 Class 1 0.14 0.18 0.16 174 accuracy 0.94 5435 macro avg 0.56 0.57 0.56 5435 weighted avg 0.95 0.94 0.94 5435
Random Forest Classifier	**Accuracy**: 0.96 **Recall score:** 0.08	precision recall f1-score support Class 0 0.97 1.00 0.98 5261 Class 1 0.74 0.08 0.15 174 accuracy 0.97 5435 macro avg 0.85 0.54 0.56 5435 weighted avg 0.96 0.97 0.96 5435
K Nearest Neighbours Classifier	**Accuracy**: 0.96 **Recall score:** 0.04	precision recall f1-score support Class 0 0.97 1.00 0.98 5261 Class 1 0.47 0.05 0.08 174 accuracy 0.97 5435 macro avg 0.72 0.52 0.53 5435 weighted avg 0.95 0.97 0.95 5435
Logistic Regression	**Accuracy**: 0.96 **Recall score:** 0.0	precision recall f1-score support Class 0 0.97 1.00 0.98 5261 Class 1 0.00 0.00 0.00 174 accuracy 0.97 5435 macro avg 0.48 0.50 0.49 5435 weighted avg 0.94 0.97 0.95 5435
Gradient Boosting	**Accuracy**: 0.96 **Recall score:** 0.04	precision recall f1-score support Class 0 0.97 1.00 0.98 5261 Class 1 0.47 0.05 0.08 174 accuracy 0.97 5435 macro avg 0.72 0.52 0.53 5435 weighted avg 0.95 0.97 0.95 5435

XGBoost	**Accuracy**: 0.96 **Recall score:** 0.10	precision recall f1-score support Class 0 0.97 1.00 0.98 5261 Class 1 0.45 0.10 0.17 174 accuracy 0.97 5435 macro avg 0.71 0.55 0.58 5435 weighted avg 0.95 0.97 0.96 5435
AdaBoost Classifier	**Accuracy**: 0.96 **Recall score: 0.03**	precision recall f1-score support Class 0 0.97 1.00 0.98 5261 Class 1 0.43 0.03 0.06 174 accuracy 0.97 5435 macro avg 0.70 0.52 0.52 5435 weighted avg 0.95 0.97 0.95 5435
LightGBM Classifier	**Accuracy**: 0.96 **Recall score:** 0.05	precision recall f1-score support Class 0 0.97 1.00 0.98 5261 Class 1 0.69 0.05 0.10 174 accuracy 0.97 5435 macro avg 0.83 0.53 0.54 5435 weighted avg 0.96 0.97 0.96 5435

Table 8.2.1.1: Table to classification report, accuracy and recall score of the classifiers

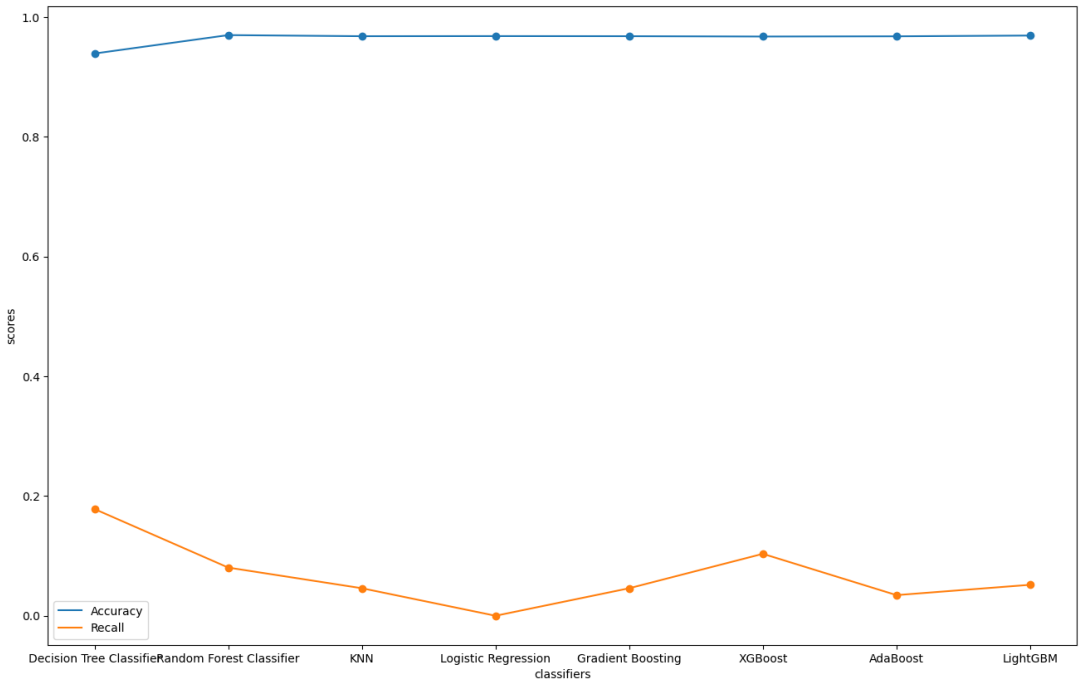


Figure 8.2.1.1: Plots to show accuracy and recall score of the classifiers

8.2.2 | PCA on the scaled Grayscale data

We have trained the following classifiers on the PCA with scaled data(grayscale channel) and report the testing accuracy, recall score and the classification report in Table 8.2.2.1.

Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.81 **Recall score:** 0.48	precision recall f1-score support Class 0 0.89 0.88 0.89 5261 Class 1 0.45 0.48 0.47 1076 accuracy 0.81 6337 macro avg 0.67 0.68 0.68 6337 weighted avg 0.82 0.81 0.82 6337
Random Forest Classifier	**Accuracy**: 0.87 **Recall score:** 0.39	precision recall f1-score support Class 0 0.89 0.97 0.93 5261 Class 1 0.75 0.39 0.51 1076 accuracy 0.87 6337 macro avg 0.82 0.68 0.72 6337 weighted avg 0.86 0.87 0.86 6337
K Nearest Neighbours Classifier	**Accuracy**: 0.86 **Recall score:** 0.32	precision recall f1-score support Class 0 0.87 0.97 0.92 5261 Class 1 0.68 0.32 0.43 1076 accuracy 0.86 6337 macro avg 0.78 0.64 0.68 6337 weighted avg 0.84 0.86 0.84 6337
Logistic Regression	**Accuracy**: 0.83 **Recall score:** 0.51	precision recall f1-score support Class 0 0.84 0.99 0.91 5261 Class 1 0.59 0.05 0.09 1076 accuracy 0.83 6337 macro avg 0.71 0.52 0.50 6337 weighted avg 0.79 0.83 0.77 6337
Gradient Boosting	**Accuracy**: 0.85 **Recall score:** 0.18	precision recall f1-score support Class 0 0.85 0.99 0.92 5261 Class 1 0.74 0.18 0.28 1076 accuracy 0.85 6337 macro avg 0.80 0.58 0.60 6337 weighted avg 0.84 0.85 0.81 6337
XGBoost	**Accuracy**: 0.86 **Recall score:** 0.43	precision recall f1-score support Class 0 0.89 0.95 0.92 5261 Class 1 0.63 0.43 0.51 1076 accuracy 0.86 6337 macro avg 0.76 0.69 0.71 6337 weighted avg 0.84 0.86 0.85 6337

AdaBoost Classifier	**Accuracy**: 0.84 **Recall score: 0.23**	precision recall f1-score support Class 0 0.86 0.97 0.91 5261 Class 1 0.58 0.23 0.33 1076 accuracy 0.84 6337 macro avg 0.72 0.60 0.62 6337 weighted avg 0.81 0.84 0.81 6337
LightGBM Classifier	**Accuracy**: 0.85 **Recall score:** 0.22	precision recall f1-score support Class 0 0.86 0.98 0.92 5261 Class 1 0.72 0.22 0.34 1076 accuracy 0.85 6337 macro avg 0.79 0.60 0.63 6337 weighted avg 0.84 0.85 0.82 6337

Table 8.2.2.1: Table to classification report, accuracy and recall score of the classifiers

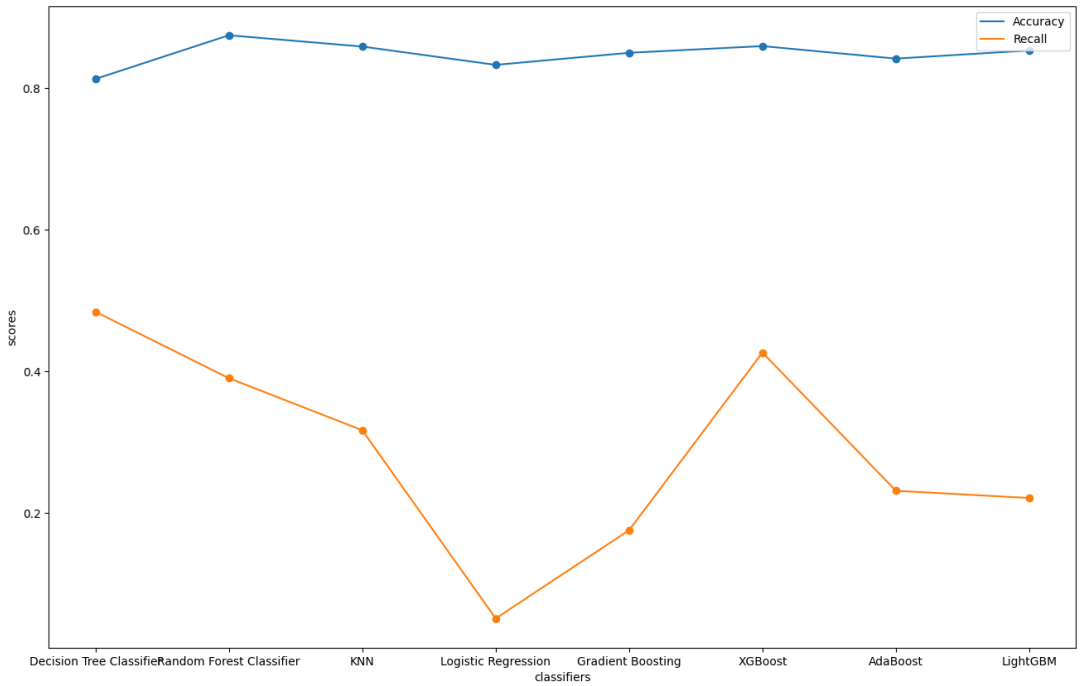


Figure 8.2.2.1: Plots to show accuracy and recall score of the classifiers

8.3 | CNN on data with RGB channels

We have implemented Convolution neural networks on the scaled data with RGB channels. Firstly we have used Adam as an optimiser and the Figure 8.3.1 reports the accuracy and the loss of the same.

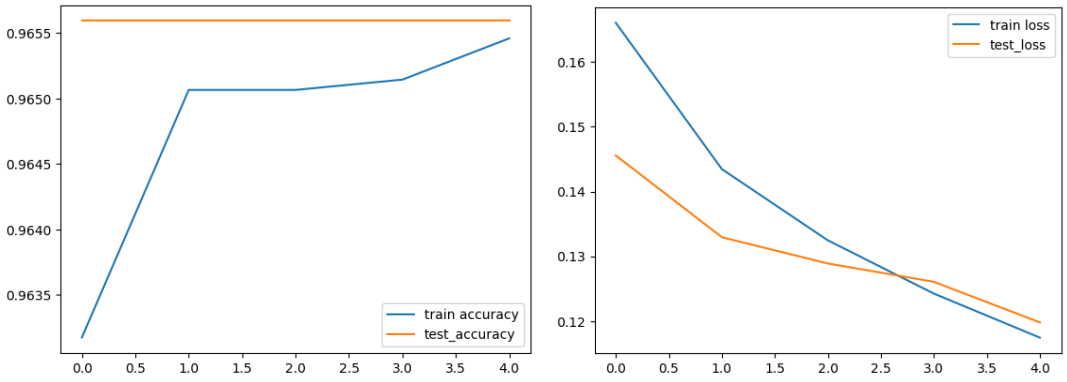


Figure 8.3.1: Plots to show accuracy and loss on CNN using Adam Optimiser

Then, we have used SGD as an optimiser and the Figure 8.3.1 reports the accuracy and the loss of the same.

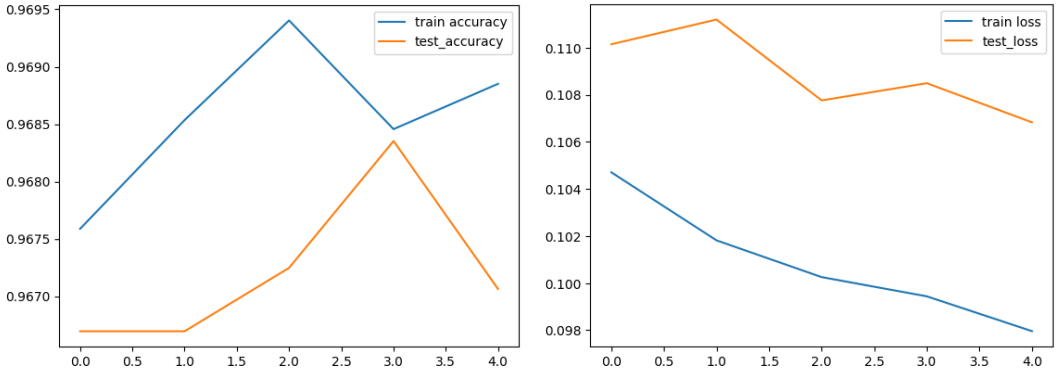


Figure 8.3.2: Plots to show accuracy and loss on CNN using SGD optimiser

8.4 | Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is one of the commonly used dimensionality reduction techniques in machine learning to solve more than two-class classification problems.

8.4.1 | LDA on scaled Grayscale data

We have trained the following classifiers on LDA with scaled data (grayscale channel) and report the testing accuracy, recall score and the classification report in Table 8.4.1.1.

Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.85 **Recall score:** 0.58	precision recall f1-score support Class 0 0.91 0.91 0.91 5258 Class 1 0.57 0.58 0.58 1079

		accuracy 0.85 6337 macro avg 0.74 0.74 0.74 6337 weighted avg 0.86 0.85 0.86 6337
Random Forest Classifier	**Accuracy**: 0.86 **Recall score:** 0.57	precision recall f1-score support Class 0 0.91 0.92 0.92 5258 Class 1 0.59 0.57 0.58 1079 accuracy 0.86 6337 macro avg 0.75 0.75 0.75 6337 weighted avg 0.86 0.86 0.86 6337
K Nearest Neighbours Classifier	**Accuracy**: 0.89 **Recall score:** 0.50	precision recall f1-score support Class 0 0.90 0.97 0.93 5258 Class 1 0.75 0.50 0.60 1079 accuracy 0.89 6337 macro avg 0.83 0.73 0.77 6337 weighted avg 0.88 0.89 0.88 6337
Logistic Regression	**Accuracy**: 0.90 **Recall score:** 0.57	precision recall f1-score support Class 0 0.92 0.96 0.94 5258 Class 1 0.76 0.57 0.65 1079 accuracy 0.90 6337 macro avg 0.84 0.77 0.80 6337 weighted avg 0.89 0.90 0.89 6337
Gradient Boosting	**Accuracy**: 0.90 **Recall score:** 0.58	precision recall f1-score support Class 0 0.92 0.96 0.94 5258 Class 1 0.75 0.59 0.66 1079 accuracy 0.90 6337 macro avg 0.83 0.77 0.80 6337 weighted avg 0.89 0.90 0.89 6337
XGBoost	**Accuracy**: 0.89 **Recall score:** 0.57	precision recall f1-score support Class 0 0.92 0.96 0.94 5258 Class 1 0.74 0.57 0.65 1079 accuracy 0.89 6337 macro avg 0.83 0.77 0.79 6337 weighted avg 0.89 0.89 0.89 6337
AdaBoost Classifier	**Accuracy**: 0.90 **Recall score: 0.55**	precision recall f1-score support Class 0 0.91 0.97 0.94 5258 Class 1 0.78 0.55 0.65 1079 accuracy 0.90 6337 macro avg 0.85 0.76 0.79 6337 weighted avg 0.89 0.90 0.89 6337
LightGBM Classifier	**Accuracy**: 0.89 **Recall score:** 0.62	precision recall f1-score support Class 0 0.92 0.95 0.94 5258 Class 1 0.72 0.62 0.67 1079

		accuracy 0.89 6337 macro avg 0.82 0.78 0.80 6337 weighted avg 0.89 0.89 0.89 6337

Table 8.4.1.1: Table to classification report, accuracy and recall score of the classifiers

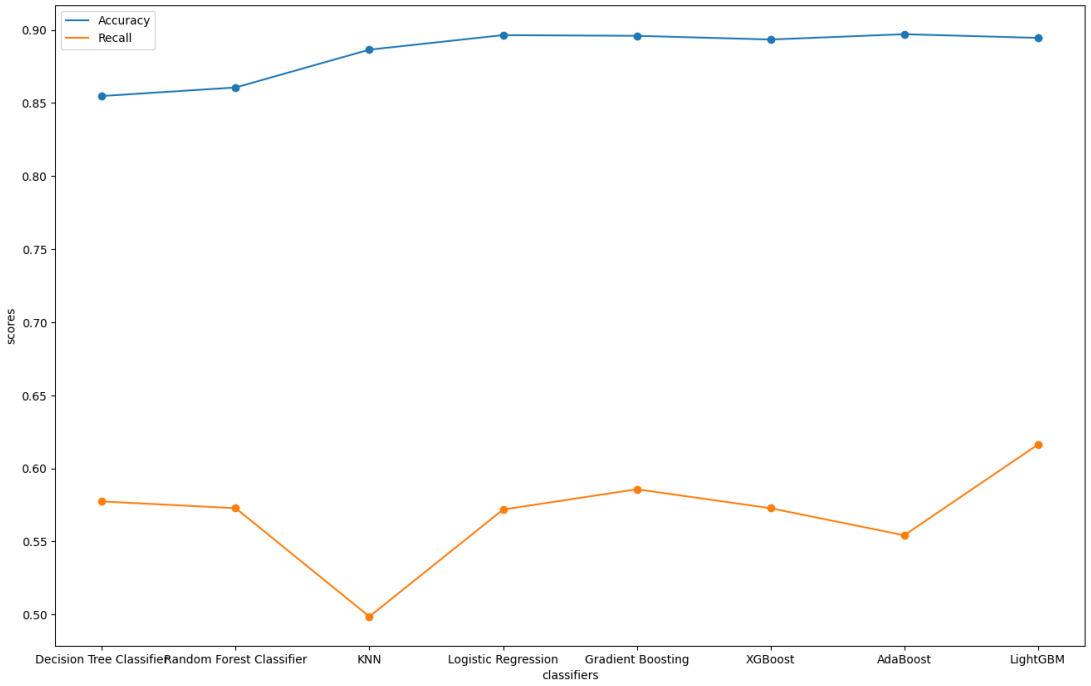


Figure 8.4.1.1: Plots to show accuracy and recall score of the classifiers

8.4.2 | LDA on masked data

We have trained the following classifiers on LDA with scaled masked data and report the testing accuracy, recall score and the classification report in Table 8.4.2.1.

Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.79 **Recall score:** 0.40	precision recall f1-score support Class 0 0.87 0.88 0.88 5217 Class 1 0.42 0.40 0.41 1121 accuracy 0.79 6338 macro avg 0.64 0.64 0.64 6338 weighted avg 0.79 0.79 0.79 6338
Random Forest Classifier	**Accuracy**: 0.80 **Recall score:** 0.40	precision recall f1-score support Class 0 0.87 0.89 0.88 5217 Class 1 0.43 0.40 0.41 1121

		accuracy 0.80 6338 macro avg 0.65 0.64 0.64 6338 weighted avg 0.79 0.80 0.80 6338
K Nearest Neighbours Classifier	**Accuracy**: 0.85 **Recall score:** 0.29	precision recall f1-score support Class 0 0.86 0.97 0.91 5217 Class 1 0.66 0.29 0.40 1121 accuracy 0.85 6338 macro avg 0.76 0.63 0.66 6338 weighted avg 0.83 0.85 0.82 6338
Logistic Regression	**Accuracy**: 0.86 **Recall score:** 0.32	precision recall f1-score support Class 0 0.87 0.97 0.92 5217 Class 1 0.71 0.32 0.44 1121 accuracy 0.86 6338 macro avg 0.79 0.65 0.68 6338 weighted avg 0.84 0.86 0.83 6338
Gradient Boosting	**Accuracy**: 0.86 **Recall score:** 0.29	precision recall f1-score support Class 0 0.87 0.98 0.92 5217 Class 1 0.73 0.30 0.42 1121 accuracy 0.86 6338 macro avg 0.80 0.64 0.67 6338 weighted avg 0.84 0.86 0.83 6338
XGBoost	**Accuracy**: 0.85 **Recall score:** 0.33	precision recall f1-score support Class 0 0.87 0.96 0.91 5217 Class 1 0.66 0.33 0.44 1121 accuracy 0.85 6338 macro avg 0.77 0.65 0.68 6338 weighted avg 0.83 0.85 0.83 6338
AdaBoost Classifier	**Accuracy**: 0.85 **Recall score:** 0.27	precision recall f1-score support Class 0 0.86 0.98 0.92 5217 Class 1 0.74 0.27 0.40 1121 accuracy 0.85 6338 macro avg 0.80 0.63 0.66 6338 weighted avg 0.84 0.85 0.83 6338
LightGBM Classifier	**Accuracy**: 0.86 **Recall score:** 0.32	precision recall f1-score support Class 0 0.87 0.97 0.92 5217 Class 1 0.72 0.32 0.44 1121 accuracy 0.86 6338 macro avg 0.79 0.65 0.68 6338 weighted avg 0.84 0.86 0.83 6338

Table 8.4.2.1: Table to classification report, accuracy and recall score of the classifiers

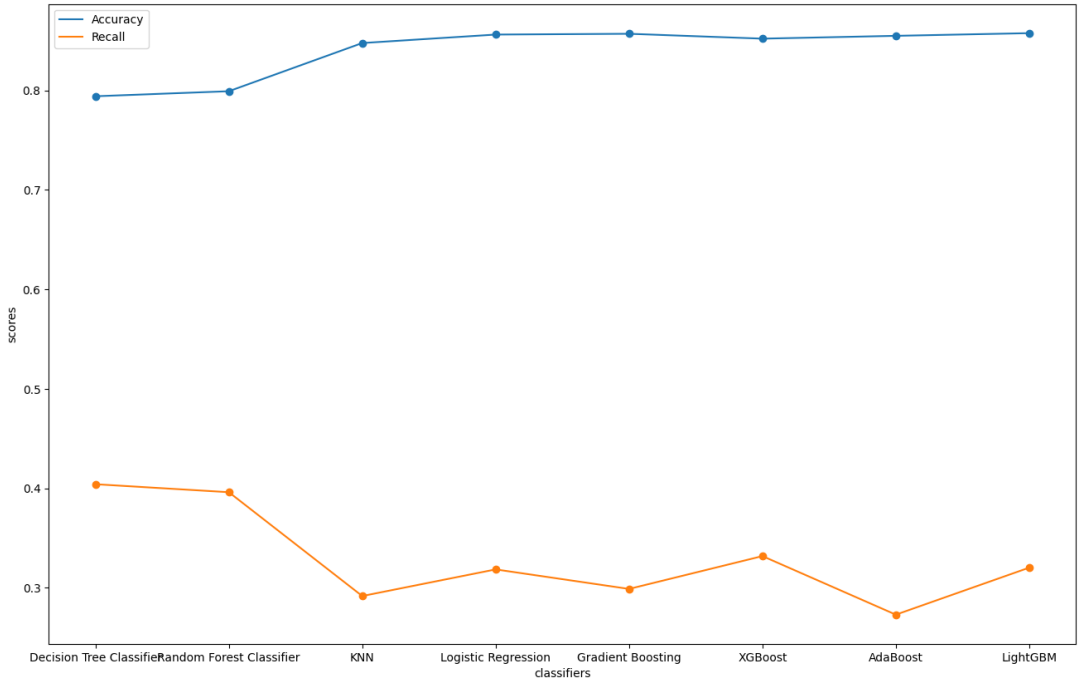


Figure 8.4.2.1: Plots to show accuracy and recall score of the classifiers

8.5 | Masked images

We have trained the following classifiers on the PCA with scaled masked data and reported the testing accuracy, recall score and the classification report in Table 8.5.1.

Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.75 **Recall score:** 0.30	precision recall f1-score support Class 0 0.85 0.85 0.85 5225 Class 1 0.31 0.31 0.31 1113 accuracy 0.76 6338 macro avg 0.58 0.58 0.58 6338 weighted avg 0.76 0.76 0.76 6338
Random Forest Classifier	**Accuracy**: 0.83 **Recall score:** 0.15	precision recall f1-score support Class 0 0.85 0.98 0.91 5225 Class 1 0.68 0.16 0.26 1113 accuracy 0.84 6338 macro avg 0.76 0.57 0.58 6338 weighted avg 0.82 0.84 0.80 6338
K Nearest Neighbours Classifier	**Accuracy**: 0.82 **Recall score:** 0.10	precision recall f1-score support Class 0 0.84 0.98 0.90 5225 Class 1 0.50 0.11 0.17 1113 accuracy 0.82 6338 macro avg 0.67 0.54 0.54 6338 weighted avg 0.78 0.82 0.77 6338

Logistic Regression	**Accuracy**: 0.82 **Recall score:** 0.03	precision recall f1-score support Class 0 0.83 1.00 0.90 5225 Class 1 0.64 0.03 0.06 1113 accuracy 0.83 6338 macro avg 0.74 0.52 0.48 6338 weighted avg 0.80 0.83 0.76 6338
Gradient Boosting	**Accuracy**: 0.83 **Recall score:** 0.06	precision recall f1-score support Class 0 0.83 1.00 0.91 5225 Class 1 0.76 0.06 0.12 1113 accuracy 0.83 6338 macro avg 0.80 0.53 0.51 6338 weighted avg 0.82 0.83 0.77 6338
XGBoost	**Accuracy**: 0.82 **Recall score:** 0.19	precision recall f1-score support Class 0 0.85 0.96 0.90 5225 Class 1 0.54 0.19 0.28 1113 accuracy 0.83 6338 macro avg 0.69 0.58 0.59 6338 weighted avg 0.79 0.83 0.79 6338
AdaBoost Classifier	**Accuracy**: 0.82 **Recall score:** 0.07	precision recall f1-score support Class 0 0.83 0.99 0.91 5225 Class 1 0.62 0.07 0.13 1113 accuracy 0.83 6338 macro avg 0.73 0.53 0.52 6338 weighted avg 0.80 0.83 0.77 6338
LightGBM Classifier	**Accuracy**: 0.83 **Recall score:** 0.07	precision recall f1-score support Class 0 0.83 0.99 0.91 5225 Class 1 0.73 0.07 0.13 1113 accuracy 0.83 6338 macro avg 0.78 0.53 0.52 6338 weighted avg 0.82 0.83 0.77 6338

Table 8.5.1: Table to classification report, accuracy and recall score of the classifiers

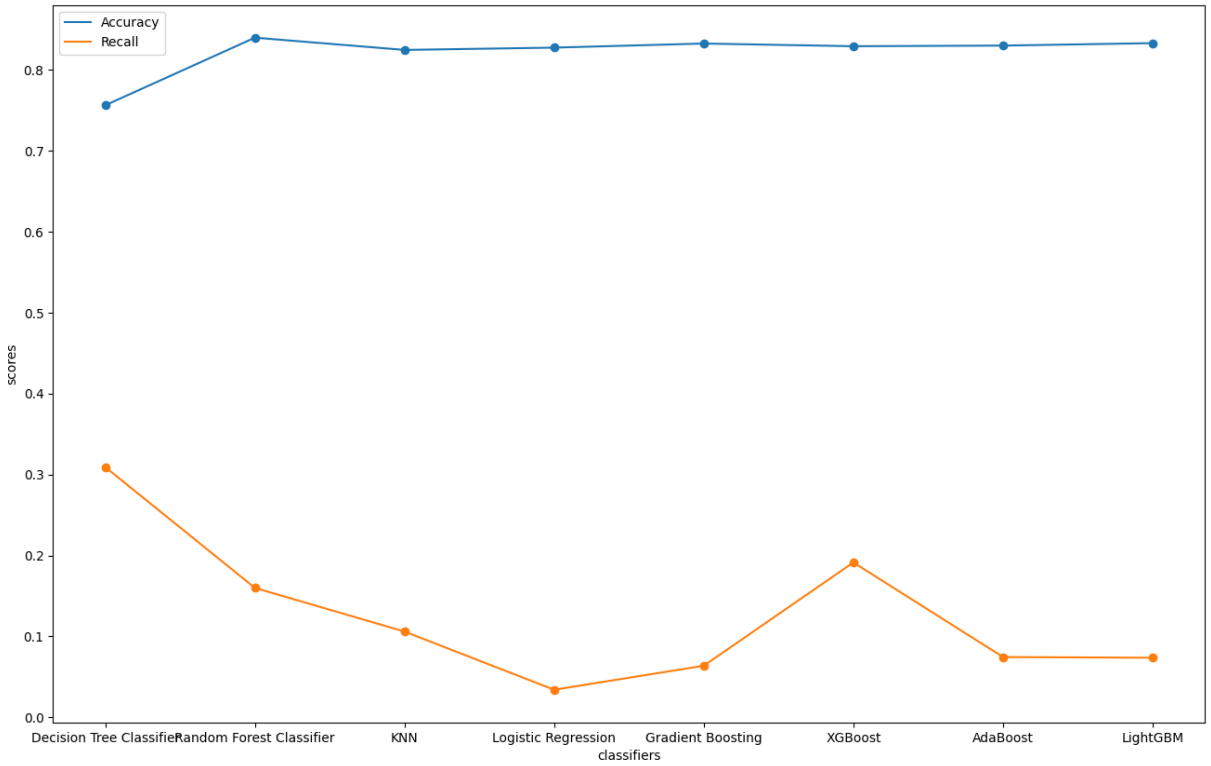


Figure 8.5.1: Plots to show accuracy and recall score of the classifiers

8.6 | CNN on masked images

We have implemented Convolution neural networks on the masked data. Firstly we have used Adam as an optimiser and the Figure 8.3.1 reports the accuracy and the loss of the same.

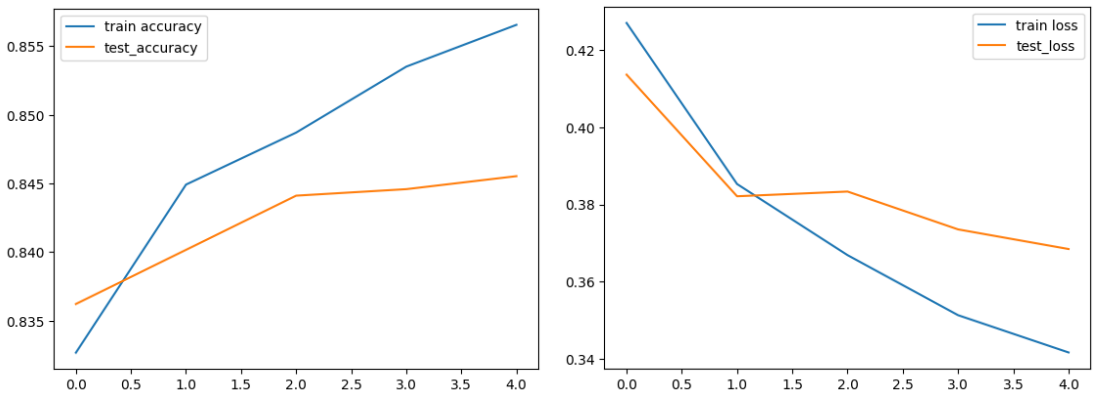


Figure 8.6.1: Plots to show accuracy and loss on CNN using Adam Optimiser

Then, we have used SGDas an optimiser and the Figure 8.3.1 reports the accuracy and the loss of the same.

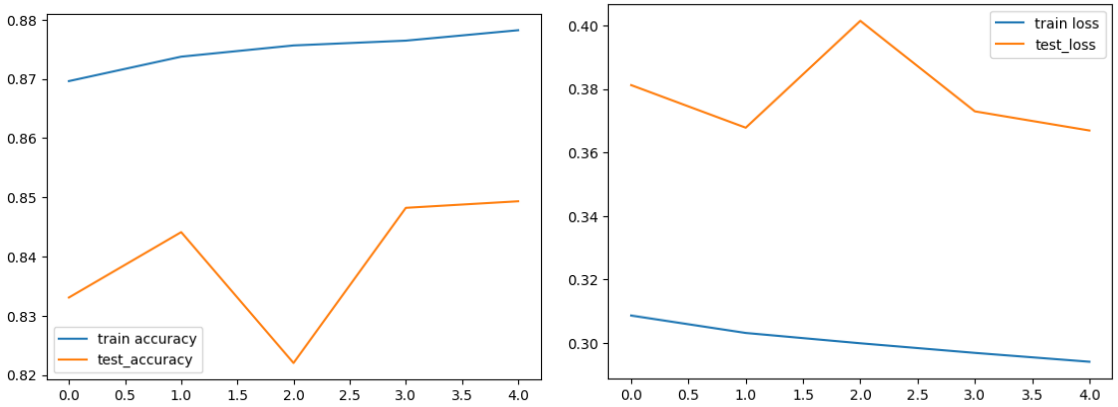


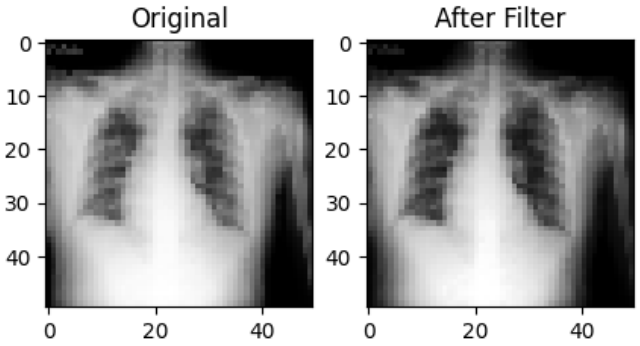
Figure 8.3.1: Plots to show accuracy and loss on CNN using SGD Optimiser

8.7 | Images with filters

We have trained the following classifiers on the data with filters applied on them and reported the testing accuracy, recall score and the classification report in Table 8.7.

8.7.1 |Histogram Equalization

Histogram Equalization is a computer image processing technique used to improve contrast in images.
We have trained the following classifiers on the data with Histogram Equalization filter applied on them and reported the testing accuracy, recall score and the classification report in Table 8.7.1.1.



Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.75 **Recall score:** 0.003	precision recall f1-score support Class 0 0.82 0.92 0.86 5256 Class 1 0.01 0.00 0.01 1081 accuracy 0.76 6337 macro avg 0.41 0.46 0.43 6337 weighted avg 0.68 0.76 0.72 6337
Random Forest Classifier	**Accuracy**: 0.70 **Recall score:** 0.0	precision recall f1-score support Class 0 0.81 0.85 0.83 5256 Class 1 0.00 0.00 0.00 1081

		accuracy 0.71 6337 macro avg 0.40 0.43 0.41 6337 weighted avg 0.67 0.71 0.69 6337
K Nearest Neighbours Classifier	**Accuracy**: 0.82 **Recall score:** 0.0009	precision recall f1-score support Class 0 0.83 0.99 0.90 5256 Class 1 0.03 0.00 0.00 1081 accuracy 0.82 6337 macro avg 0.43 0.50 0.45 6337 weighted avg 0.69 0.82 0.75 6337
Logistic Regression	**Accuracy**: 0.70 **Recall score:** 0.02	precision recall f1-score support Class 0 0.81 0.85 0.83 5256 Class 1 0.03 0.02 0.02 1081 accuracy 0.71 6337 macro avg 0.42 0.43 0.43 6337 weighted avg 0.67 0.71 0.69 6337
Gradient Boosting	**Accuracy**: 0.82 **Recall score:** 0.0	precision recall f1-score support Class 0 0.83 1.00 0.91 5256 Class 1 0.00 0.00 0.00 1081 accuracy 0.83 6337 macro avg 0.41 0.50 0.45 6337 weighted avg 0.69 0.83 0.75 6337
XGBoost	**Accuracy**: 0.69 **Recall score:** 0.005	precision recall f1-score support Class 0 0.80 0.84 0.82 5256 Class 1 0.01 0.01 0.01 1081 accuracy 0.70 6337 macro avg 0.41 0.42 0.41 6337 weighted avg 0.67 0.70 0.68 6337
AdaBoost Classifier	**Accuracy**: 0.81 **Recall score:** 0.05	precision recall f1-score support Class 0 0.83 0.97 0.89 5256 Class 1 0.25 0.06 0.09 1081 accuracy 0.81 6337 macro avg 0.54 0.51 0.49 6337 weighted avg 0.73 0.81 0.76 6337
LightGBM Classifier	**Accuracy**: 0.82 **Recall score:** 0.0	precision recall f1-score support Class 0 0.83 1.00 0.91 5256 Class 1 0.00 0.00 0.00 1081 accuracy 0.83 6337 macro avg 0.41 0.50 0.45 6337 weighted avg 0.69 0.83 0.75 6337

Table 8.7.1.1: Table to classification report, accuracy and recall score of the classifiers

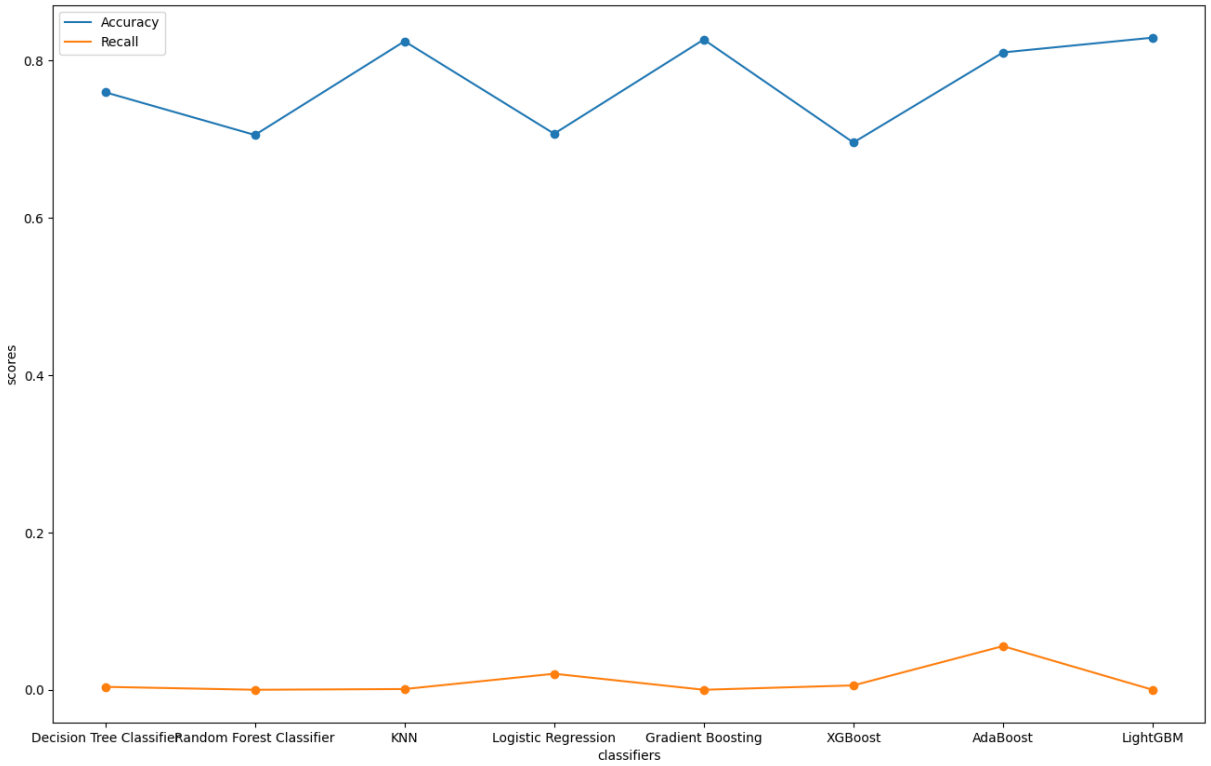


Figure 8.7.1.1: Plots to show accuracy and recall score of the classifiers

8.8 | CNN on images with filters

8.8.1 |Histogram Equalization

We have implemented Convolution neural networks on the filtered data. Firstly we have used Adam as an optimiser and the Figure 8.8.1.1 reports the accuracy and the loss of the same.

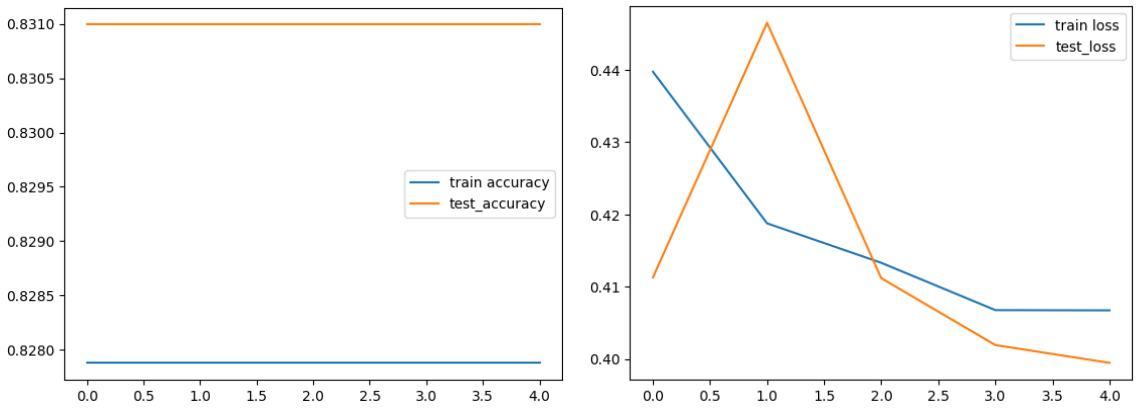


Figure 8.8.1.1: Plots to show accuracy and loss on CNN using Adam Optimiser

Then, we have used SGDas an optimiser and the Figure 8.8.1.2 reports the accuracy and the loss of the same.

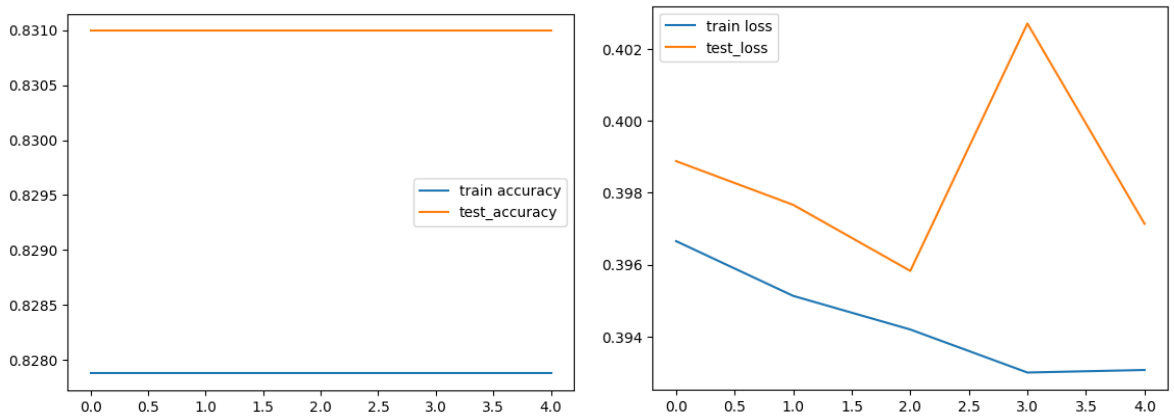


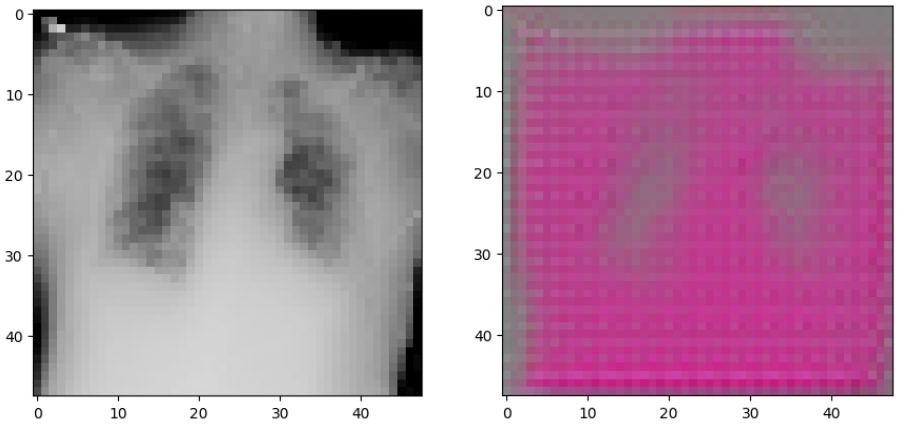
Figure 8.8.1.2: Plots to show accuracy and loss on CNN using SGD Optimiser

8.9 | UNET

UNet is a deep learning architecture for image segmentation tasks, consisting of an encoder and a decoder path with skip connections between corresponding layers.

We have trained the following classifiers on UNet segmented data and reported the testing accuracy, recall score and the classification report in Table 8.9.1.

The below images show the original image and segmented image by UNet.



Classifiers	Accuracy and Recall Score	Classification Report
Decision Tree Classifiers	**Accuracy**: 0.86 **Recall score:** 0.599	precision recall f1-score support Class 0 0.92 0.92 0.92 5243 Class 1 0.61 0.60 0.61 1092 accuracy 0.87 6335 macro avg 0.77 0.76 0.76 6335 weighted avg 0.86 0.87 0.87 6335
Random Forest Classifier	**Accuracy**: 0.91 **Recall score:** 0.54	precision recall f1-score support Class 0 0.81 0.85 0.83 5256 Class 1 0.00 0.00 0.00 1081

		accuracy 0.71 6337 macro avg 0.40 0.43 0.41 6337 weighted avg 0.67 0.71 0.69 6337
Logistic Regression	**Accuracy**: 0.86 **Recall score:** 0.34	precision recall f1-score support Class 0 0.88 0.97 0.92 5243 Class 1 0.73 0.34 0.47 1092 accuracy 0.86 6335 macro avg 0.80 0.66 0.70 6335 weighted avg 0.85 0.86 0.84 6335
Gradient Boosting	**Accuracy**: 0.89 **Recall score:** 0.43	precision recall f1-score support Class 0 0.89 0.99 0.94 5243 Class 1 0.88 0.43 0.58 1092 accuracy 0.89 6335 macro avg 0.88 0.71 0.76 6335 weighted avg 0.89 0.89 0.88 6335
XGBoost	**Accuracy**: 0.94 **Recall score:** 0.73	precision recall f1-score support Class 0 0.95 0.99 0.97 5243 Class 1 0.92 0.73 0.81 1092 accuracy 0.94 6335 macro avg 0.93 0.86 0.89 6335 weighted avg 0.94 0.94 0.94 6335
AdaBoost Classifier	**Accuracy**: 0.88 **Recall score:** 0.526	precision recall f1-score support Class 0 0.91 0.96 0.93 5243 Class 1 0.72 0.53 0.61 1092 accuracy 0.88 6335 macro avg 0.81 0.74 0.77 6335 weighted avg 0.87 0.88 0.88 6335
LightGBM Classifier	**Accuracy**: 0.91 **Recall score:** 0.56	precision recall f1-score support Class 0 0.92 0.99 0.95 5243 Class 1 0.89 0.56 0.69 1092 accuracy 0.91 6335 macro avg 0.90 0.77 0.82 6335 weighted avg 0.91 0.91 0.90 6335

Table 8.9.1.: Table to classification report, accuracy and recall score of the classifiers

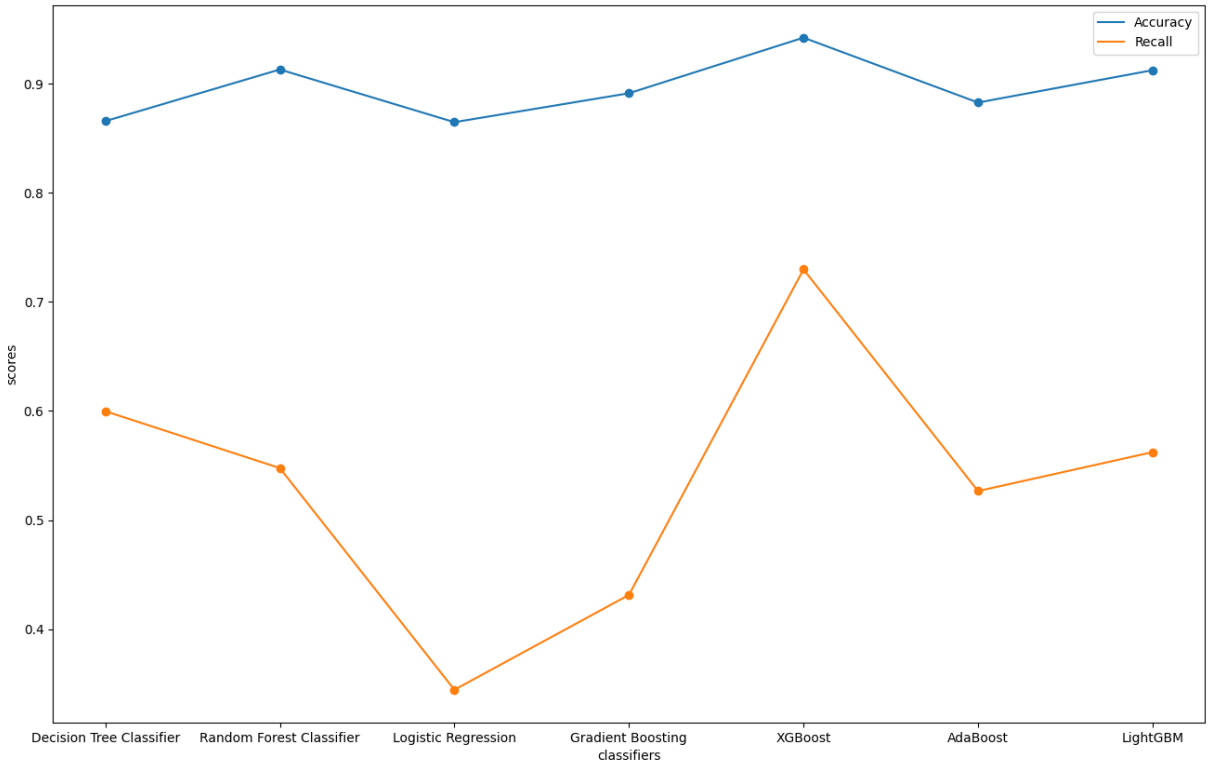


Figure 8.9.1: Plots to show accuracy and recall score of the classifiers

8.10 | Multi-Layer Perceptron (Artificial Neural Network)

An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain.

8.10.1 | ANN on scaled data

We have implemented an Artificial Neural Network on the original dataset. Firstly we have used Adam as an optimizer and the Figure 8.10.1.1. shows reports the accuracy and loss of the same.

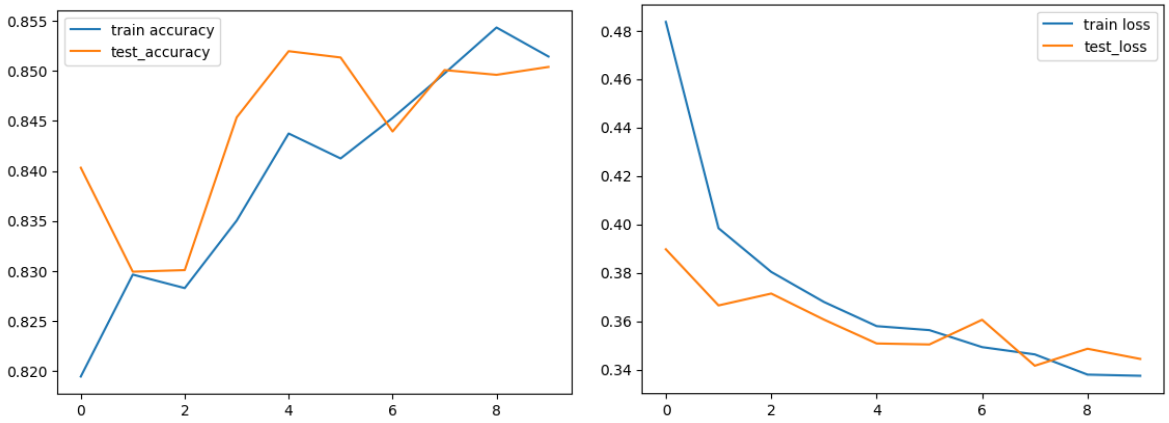


Figure 8.10.1.1: Plots to show accuracy and loss on ANN using Adam Optimiser

Then, we have used SGDas an optimiser and the Figure 8.10.1.2 reports the accuracy and the loss of the same.

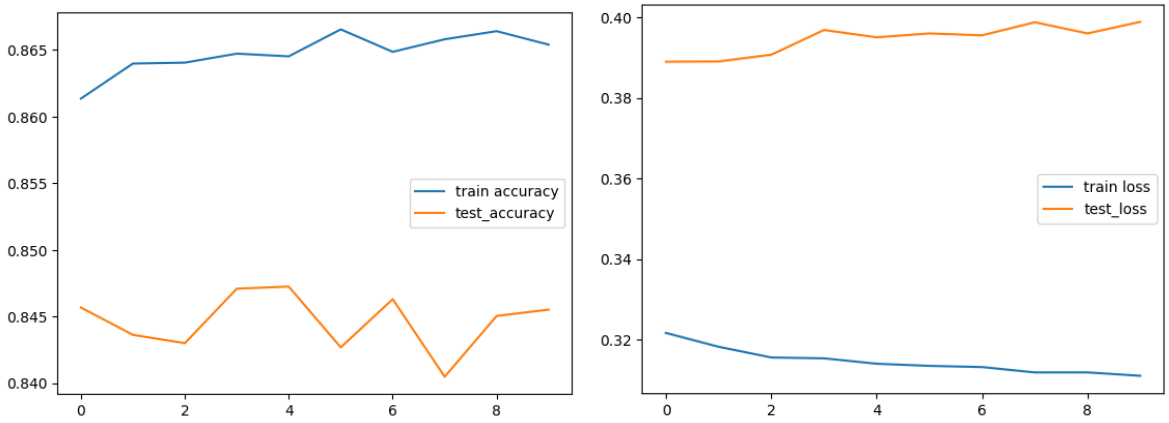


Figure 8.10.1.2: Plots to show accuracy and loss on ANN using SGD Optimiser

8.10.2 | ANN on masked data

We have implemented an Artificial Neural Network on the scaled masked dataset. Firstly we have used Adam as an optimizer and the Figure 8.12.2 shows reports the accuracy and loss of the same.

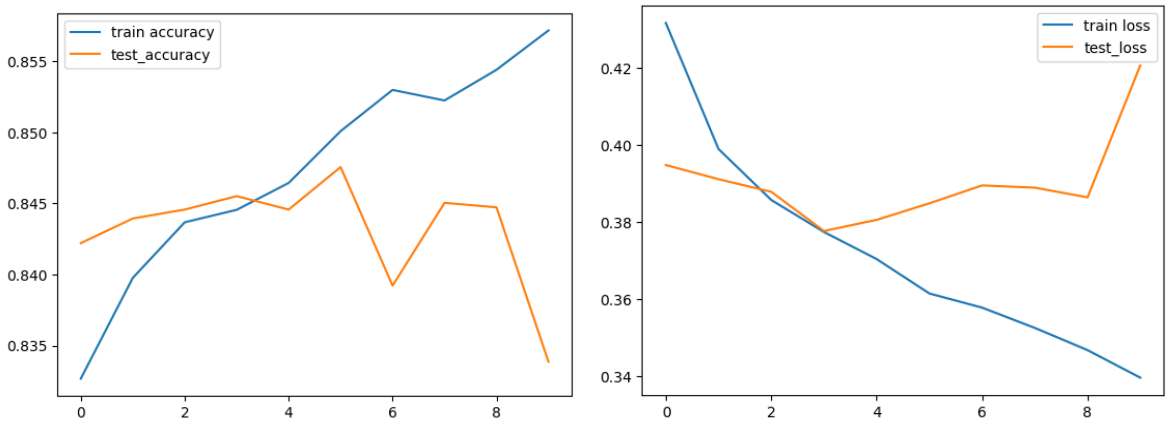


Figure 8.10.2.1: Plots to show accuracy and loss on ANN using Adam Optimiser

Then, we have used SGDas an optimiser and the Figure 8.10.1.2 reports the accuracy and the loss of the same.

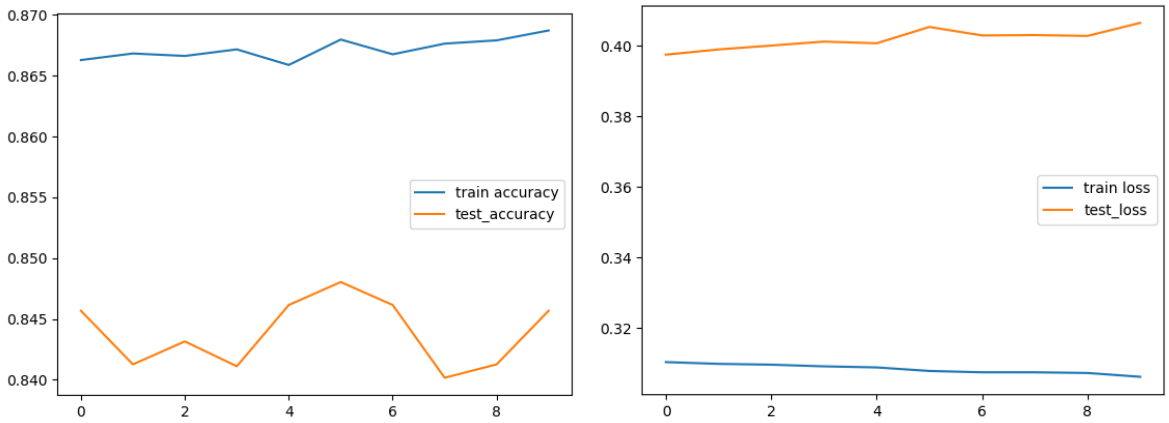


Figure 8.10.2.2: Plots to show accuracy and loss on ANN using SGD Optimiser

9 | Conclusion

Through our project, we analysed different approaches to best distinguish between the two classes in the data provided. Concluding our experiments, we can say that the convolutional neural networks with SGD as optimiser on the scaled data with rgb channels have the best test accuracy (96.71%) amongst all the classifiers. The convolutional neural networks with Adam as optimiser on the scaled data with rgb channels have the test accuracy of 96.55% which is very close to SGD optimiser. XGBoost classifier also performed well on original scaled data and gave an accuracy of 95% and recall score of 78%, however XGBoost is computationally expensive and time consuming.

10 | Contributions

NIHARIKA DADU (B21CS052)

- Downloaded Dataset Kaggle.
- Performed Data Exploration
- Designed Model Pipeline
- Performed preprocessing of the data
- PCA Data Model's written and trained.
- Masked Data Model's written and trained.
- Filtered Data Model's written and trained.
- Trained a CNN on our custom dataset
- Reported results for the above models
- Prepared the report for the project in Latex.

AMISHA KUMARI (B21CS007)

- Data Exploration
- Model Pipeline
- Converted RGB channel to grayscale.
- Trained classifiers on original data.
- Trained classifiers on PCA transformed data (grayscale).
- Implemented LDA on original data and trained classifiers (grayscale).
- Implemented LDA on masked data and trained classifiers.
- Reported results for the above models.
- Prepared the report of the project in Latex.

HARSHIL KANERIA (B21CS033)

- Downloading dataset using Kaggle API.
- Data Exploration.
- UNet Implementation and Tuning.
- Classifiers on segmented data from UNet.
- ANN on original data and Tuning.
- ANN on masked data and Tuning.
- Reported results for the above models.
- Prepared the report of the project in Latex.

11 | References

[1] Omprakash Patel, Yogendra P. S. Maravi and Sanjeev Sharma . *A COMPARATIVE STUDY OF HISTOGRAM EQUALIZATION BASED IMAGE ENHANCEMENT TECHNIQUES FOR BRIGHTNESS PRESERVATION AND CONTRAST ENHANCEMENT*, 2013.

[2] Jiuxiang Gua, Zhenhua Wangb, Jason Kuenb, Lianyang Mab, Amir Shahroudyb, Bing Shuaib, Ting Liub, Xingxing Wangb, Li Wangb, Gang Wangb, Jianfei Caic, Tsuhan Chenc . Recent Advances in Convolutional Neural Networks ,2017.

[3] Tawsifur Rahman, Dr. Muhammad Chowdhury, Amith Khandakar . COVID-19 Radiography Database ,2022.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox . https://arxiv.org/pdf/1505.04597v1.pdf, 2015.

Documentations we referred :

[1] sklearn.decomposition.PCA.
[2] sklearn.discriminant_analysis.LinearDiscriminantAnalysis.
[3] sklearn.ensemble.RandomForestClassifier
[4] sklearn.neighbors.KNeighborsClassifier
[5] sklearn.linear_model.LogisticRegression
[6] sklearn.ensemble.GradientBoostingClassifier
[7] sklearn.preprocessing.StandardScaler
[8] sklearn.ensemble.AdaBoostClassifier
[9] lightgbm.LGBMClassifier
[10] sklearn.tree.DecisionTreeClassifier.