Project: COVID Detection using X-ray Images

Biswas Gupta (B22BB018)

Abstract

COVID- 19 global pandemic affects health care and lifestyle worldwide, and its early detection is critical to control cases' spreading and mortality. The actual leader diagnosis test is the Reverse transcription Polymerase chain reaction (RT-PCR), result times and cost of these tests are high, so other fast and accessible diagnostic tools are needed. This project's approach uses various machine learning models to process these images and classify them as positive or negative for COVID-19. The proposed system involves removing the surroundings which do not offer relevant information for the task and may produce biased results; after this initial stage comes the classification model trained under the transfer learning scheme; and finally, results analysis. The best models achieved a detection accuracy of COVID-19 around 85%.

1. Introduction

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 the COVID-19 cases using 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. In this project, we've analyzed the given chest x-ray image dataset using essential exploratory data analysis techniques and drawn predictions about whether the subject of study has been infected or not.

2. About the dataset

COVID Detection using X-ray Images dataset

- The COVID-19 dataset consists of Non-COVID and COVID cases of X-ray images.

- The associated dataset is augmented with different augmentation techniques to have about 17099 X-ray images.
- The dataset contains two main folders, one for the X-ray images, which includes two separate sub-folders of 5500 Non-COVID images and 4044 COVID images

3. Importing the Dataset

The dataset was downloaded and then uploaded to google drive from where it was imported using tensorflow.keras.utils.image dataset from directory() with batch size = 32, image width and height = 224, color mode as grayscale and using 80% for training and rest 20% for validation.

4. Data Preprocessing and Analysis

As the images come from several datasets with different image sizes and acquisition conditions, a preprocessing step is applied to reduce or remove effects on the performance of the models due to data variability.

- Preprocessings like normalisation of the images, conversion to gray scale, resizing to standard 224 x 224 size was handled during the import by keras library.

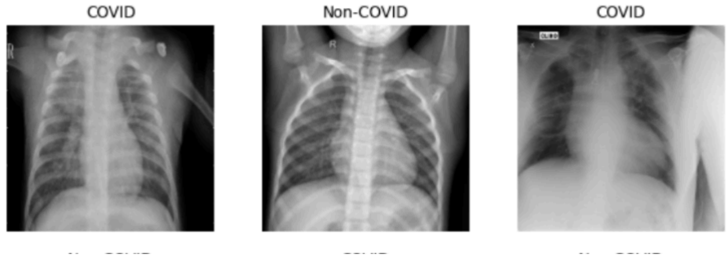


Figure 1: Images after preprocessing

- Performed dimensionality reduction using PCA, explained in brief in section 5.

5. Dimensionality reduction using PCA

PCA was done to :

- Reduce the time and storage space required.
- Remove multi-collinearity which improves the interpretation of the parameters of the machine learning model.

For deciding the number of Principle components, a scree plot was plotted and a threshold variance of 95% was set and image dimensions were reduced as shown.

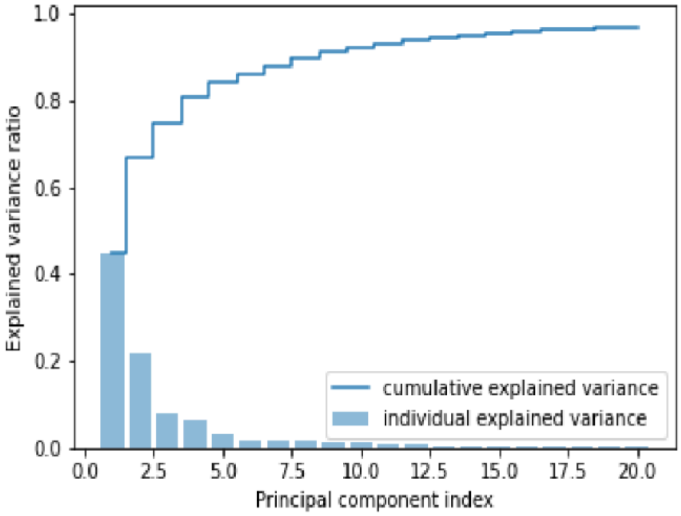


Figure 2: Scree Plot

20 n components were used as they covered 95% of the cumulative variance.

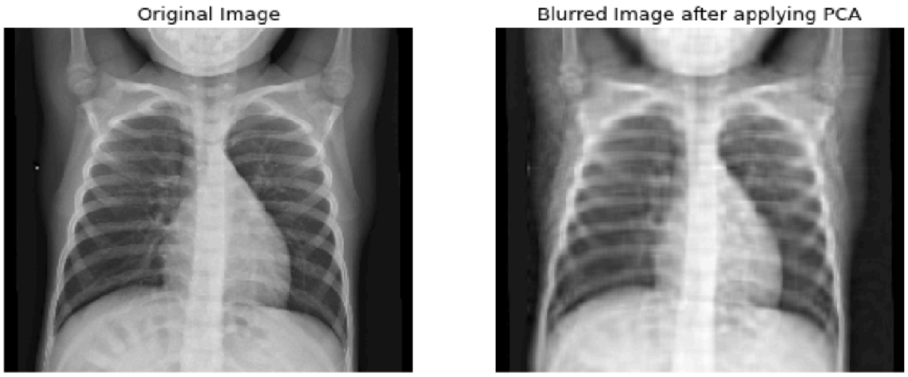


Figure 4: Dimensionality reduction using PCA

6. Application of machine learning models

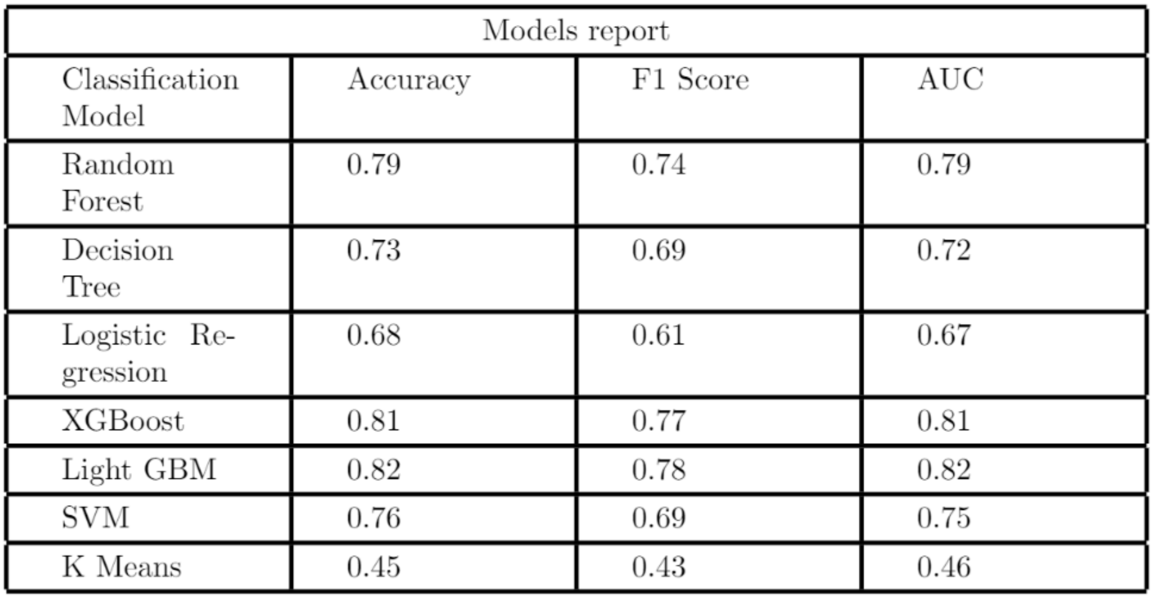
6.1 Creation of dataset

Dimensions of each image were reduced to 224x20 using PCA, which then was flattened and stored in a numpy array.

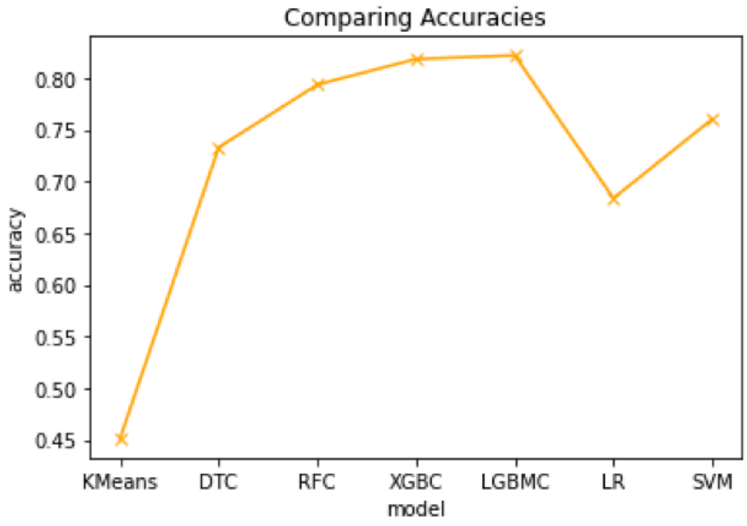
6.2 Train Test Split

Using train test split function from sklearn the given dataset was split into train and test dataset in 70:30 ratio.

6.3 Results on dataset



Models report			
Classification Model	Accuracy	F1 Score	AUC
Random Forest	0.79	0.74	0.79
Decision Tree	0.73	0.69	0.72
Logistic Regression	0.68	0.61	0.67
XGBoost	0.81	0.77	0.81
Light GBM	0.82	0.78	0.82
SVM	0.76	0.69	0.75
K Means	0.45	0.43	0.46


Comparing Accuracies
accuracy
KMeans
DTC
RFC
XGBC
LGBMC
LR
SVM
model