Pattern Recognition and Machine Learning

Indian Institute of Technology, Jodhpur

Major Course Project

Project-1

"COVID Detection using X-ray Images"

Dev Agarwal – B21CS023

Devang Shrivastava – B21CS024

Keshav Malani – B21CS038

Abstract

The COVID-19 pandemic has had a significant impact on society, prompting an urgent need to optimize the healthcare industry with tools that can effectively address the outbreak and its consequences on citizens. To this end, we employed machine learning to distinguish between chest X-ray images of individuals with COVID-19 and those who are healthy. Our project report details the implementation of different machine-learning approaches in this image classification task, using a provided chest X-ray dataset. Overall, our project aims to contribute to the ongoing efforts to combat the COVID-19 pandemic and improve public health

In our model, we proposed two strategies:

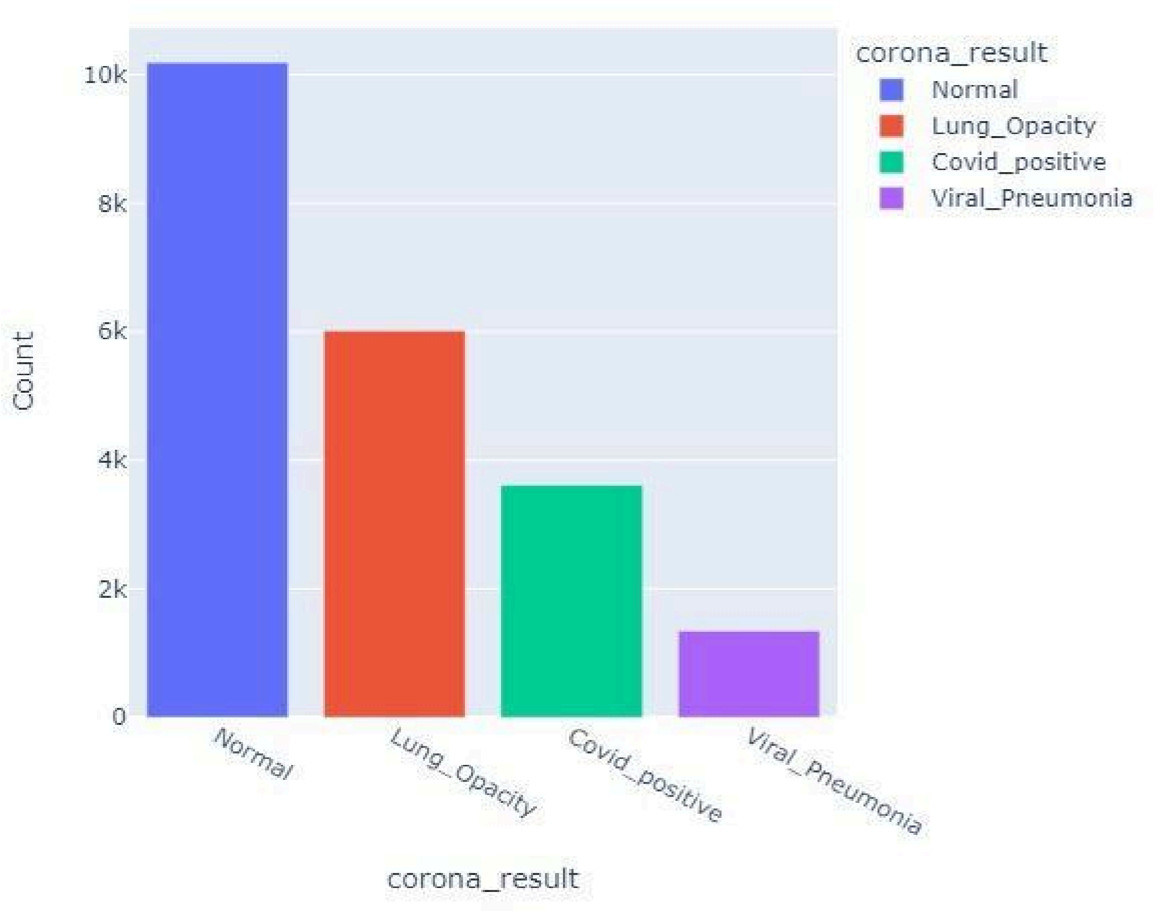
1. Either we have classified the image as covid / non-covid.
2. Or have classified them to covid, healthy, lung infection, or, pneumonia.

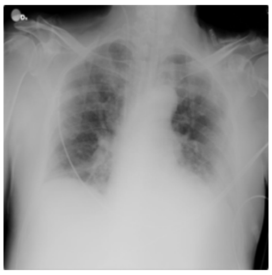
Dataset

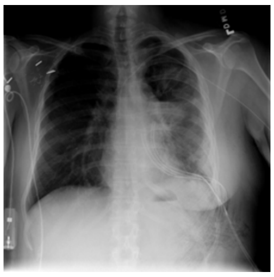
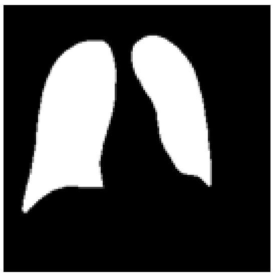
The provided dataset contains a total of 21165 X-Ray images out of which

- 3616 samples represent an affliction with the COVID-19 virus
- 10192 samples represent healthy individuals
- 6012 samples represent Lung Opacity (Non-COVID lung infection) and,
- 1345 samples represent Viral Pneumonia images

Also, the masked image of each dataset was provided. We used that image to transform the X-ray image, capturing only the lungs.

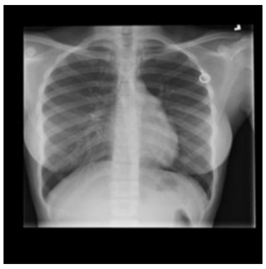


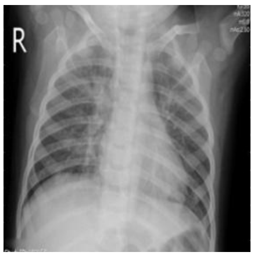


COVID

LUNG OPACITY





NORMAL

Pneumonia

Pre-processing

We have applied many machine-learning algorithms to our data and accordingly pre-processed the data. First, we transformed our dataset capturing only the lungs’ part with the help of masked data. For transformation, we only had to resize the images to the same pixels and applied bitwise AND. We resized the image to 75 x 75 resolution with a single channel, i.e., convert RGB to grayscale images using OpenCV. Subsequently, we flattened the images and used a Pandas Data Frame for storing the same, yielding a Data Frame having 5625 features and a target vector containing the class index.

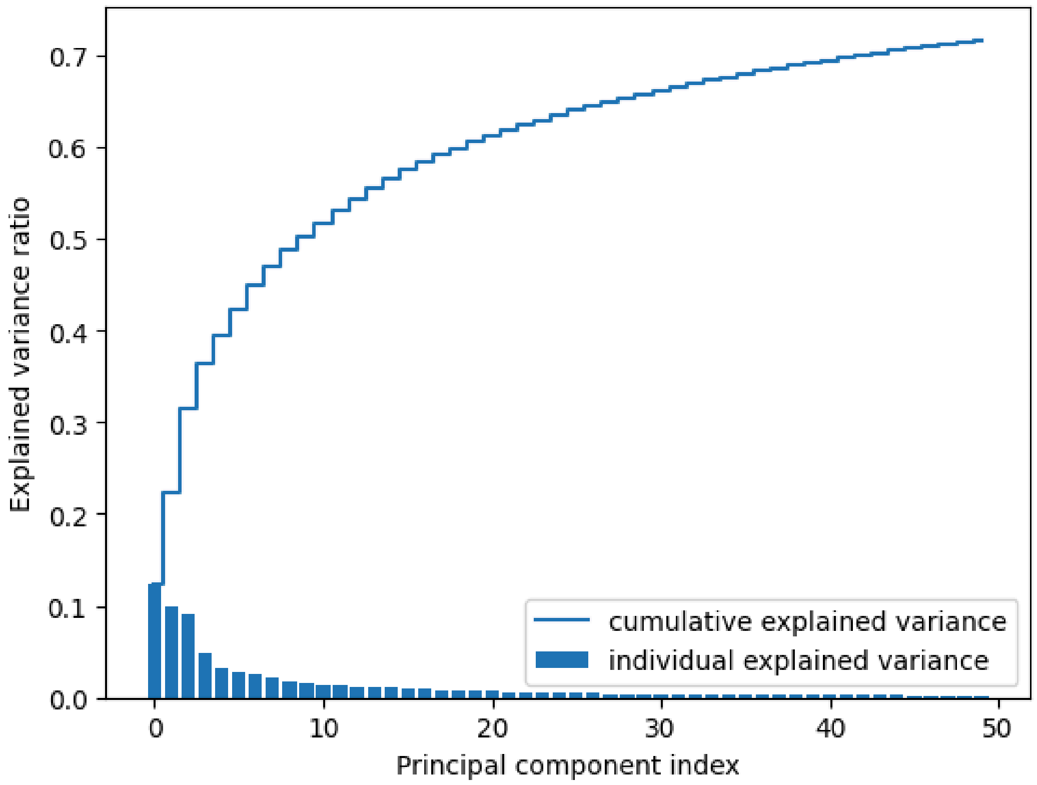
Alternatively, while applying CNN, we have transformed the images the same way as done above and resized the image to 80 x 80 resolution with 3 color channels. And then performed feature filtering and pooling 3 times, each time reducing the channels from 128 to 64 to 32 and then flattening it. Also, we have prepared a dataset that we haven’t transformed using masked images and directly resized it.

Dimensionality Reduction

PCA

We have applied various dimension reduction strategies to the dataset with 5625 features as it is too huge to compute.

The first method employs Principal Component Analysis (PCA) to reduce the dimensions to a lower space. we take the number of principal components as 50 because the variance of the first 50 components is comparatively higher than the rest capturing approximately 70 percent of the total variance.



Feature Filtering from CNN

The second method involves the reduced data set we get after applying CNN. In CNN we initially had 80 x 80 x 3 features, we reduced the number of features by consecutively applying 3 times convolution and max pooling layers. Then we flattened the output which gave us 4096 features.

We also applied LDA on these 4096 and observed the performance of different models.

Experiments and Strategies:

After the pre-processing, both the reduced and original data are split into train and validation data using a 70-20-10 distribution(train-val-test). Models In this course, we have been introduced to various models, some of which are:

Strategy – 1:

In this strategy, we have first applied PCA as described above and applied the following model:

1. Decision Tree Classifier,
2. Random Forest
3. Linear Support Vector Machine (SVM),
4. Radial Basis Function SVM,
5. Multi-Layer Perceptron (MLP) Classifier,
6. Ensemble Learning (eg. XG Boost, AdaBoost, LightGBM).

Hyperparameters were tuned for every model and the best hyperparameters were selected.

Strategy – 2:

In this strategy, we have applied CNN(Convolutional Neural Network) on the dataset. This is because of their ability to recognize visual patterns in image and classify them accurately based on the identified features.

Strategy – 3:

In this strategy, after we got the filtered features from CNN (refer to strategy-2), we chose those features and made a separate dataset, applied LDA to it with n_component = 3, and applied the following model:

1. Decision Tree Classifier,
2. Random Forest
3. Linear Support Vector Machine (SVM),
4. Radial Basis Function SVM,
5. Multi-Layer Perceptron (MLP) Classifier,
6. Ensemble Learning (eg. XG Boost, AdaBoost, LightGBM).

Hyperparameters were tuned for every model and the best hyperparameters were selected.

For all strategies, we have made models for classifying data into either the 4 classes given or into the binary that is into covid or non-covid.

Observations:

1. Dataset PCA reduced

   Classes: 0,1 i.e., covid or non-covid respectively

MODELS	**ACCURACY**
Decision Tree Classifier	59.69
Random Forest	70.95
Multi-Layer Perceptron (MLP)	67.39
XG Boost	71.95
AdaBoost	71.65
LightGBM	72.06

2. Dataset PCA reduced

   Classes: 0,1,2,3 i.e., covid, Lung Opacity, Healthy, Pneumonia respectively

MODELS	ACCURACY	RECALL
Decision Tree Classifier	82.36	100
Random Forest	84.65	100
XG Boost	84.43	96.04
AdaBoost	84.65	96.02
LightGBM	85.14	96.04
Multi-Layer Perceptron (MLP)	81.82	-

3. Dataset □ LDA reduced

   Classes: 0,1,2,3 i.e., covid, Lung Opacity, Healthy, Pneumonia respectively

MODELS	ACCURACY
Decision Tree Classifier	90.62
Random Forest	90.59
Linear Support Vector Machine	90.84
Radial Basis Function SVM	90.74
XG Boost	90.24
AdaBoost	89.62
LightGBM	90.66

4. Dataset □ LDA reduced

   Classes: 0,1 i.e., covid or non-covid respectively

MODELS	ACCURACY	RECALL
Decision Tree Classifier	96.40	98.98

Random Forest	96.36	98.55
Linear Support Vector Machine	90.84	99.77
Radial Basis Function SVM	90.74	96.08
XG Boost	96.29	98.18
AdaBoost	96.48	98.86
LightGBM	96.81	98.64

5. Dataset□ Resized data in RGB channel
   ->The basic operation of a CNN is convolution, which involves applying a set of filters to the image. Each filter is a small matrix of numbers that is convolved (slid) over the image, computing a dot product at each position. This produces a set of output maps that highlight different features of the image, such as edges, textures, and shapes.
   ->We have applied 3 such layers with 16,32 and 64 filters of size(3,3) respectively where each filter would highlight a different feature.
   ->The number of filters are increasing with the depth of the network as initially the model captures low level features such as edges and textures which do not require more filters but as the model goes deeper it has to recognize more complex and high level features using the output of the previous layer and hence more filters are required.
   ->Also a max-pooling layer of size(2,2) is added after each convolution layer to reduce the size of the output feature map.
   ->To prevent overfitting of the model we have also added dropout layers which randomly drops out(set to zero) a certain number of neurons during training.
   ->After doing this the output that we get is flattened and a series of fully connected dense layers are added to the model which perform the classification task using the same flattened output.
   ->Our initial dataset was of dimension 80*80*3 (=19200) which was reduced to 4096 by the convolution and max-pooling layers.

-> We extracted this data and further applied LDA on it and used it for different classifiers the details of which are discussed in strategy 3

-> We have used 'adam' optimizer and sparse categorical cross entropy for our model to tune our model.

->We applied 4 different CNN models and the results are as follows.

Dataset	**Classification type**	**Accuracy**
Resize data in RGB channel	Binary	97.0%
Resize data in RGB channel	Multiclass	89.1%
RGB data transformed using masked images	Binary	90%
RGB data transformed using masked images	Multiclass	81%

->We also tried to apply the CNN model on single channel grayscale converted from 3 channel RGB images but the results which we were getting were not satisfactory.