COVID Detection using X-ray Images

Divyam Patel (B20EE082) and Jaimin Sanjay Gajjar (B20AI014)
Github Repo: PRML-Project

Abstract

The Coronavirus Disease 2019 (COVID-19) has brought a worldwide threat to the living society. The whole world is putting incredible efforts to fight against the spread of this deadly disease. During the recent global urgency, scientists, clinicians, and healthcare experts around the globe keep on searching for a new technology to support in tackling the COVID-19 pandemic. Therefore, we have tried to build a machine learning model that can predict the COVID-19 cases from chest X-ray images.

Index Terms

Classification, Random Forest, Multilayer Perceptron, Light Gradient Boosting, Convolutional Neural Network, VGG16, ResNet50, Principal Component Analysis, Linear Discriminant Analysis.

I. Introduction

WE understood that in COVID times, people had to wait in long queues to know their COVID results even after getting their X-Rays. So there was a urgent need to create a pipeline that will predict if they are affected with COVID-19 or not so that they can seek further guidance from doctor immediately.

In this project we have made a pipeline and a fast and easy to use interface that will predict the chance of having COVID-19 for a person just by providing a Chest X-Ray image. This project aims at classifying COVID-19 patients on basis of their Chest X-Ray image.

II. Data Preparation And Preprocessing

The dataset contained 5,500 Normal chest X-Ray images and 4,044 COVID affected chest X-Ray images. To make the dataset balanced, we used randomly selected 3,500 images for COVID and Non-COVID images. Then we divided those 3,500 images for each class into train, test, validation dataset with 5:1:1 ratio.

The images were read using opencv and converted to grayscale. Then the grayscale images were resized into (128, 128) to remove any chance of irregularities in data representation and then we flattened the image to get a 16,384 length feature vector for each data point. We have also used Principal Component Analysis to decrease the length of feature vector and with covering 99% variance, we got 1480 length feature vector for PCA.

We also used Linear Discriminant Analysis which is a supervised method to get a linear transformation which can increase the separability between the 2 classes. We have also generated heatmap of X-Ray images which gave us a proper idea to move in the right direction.

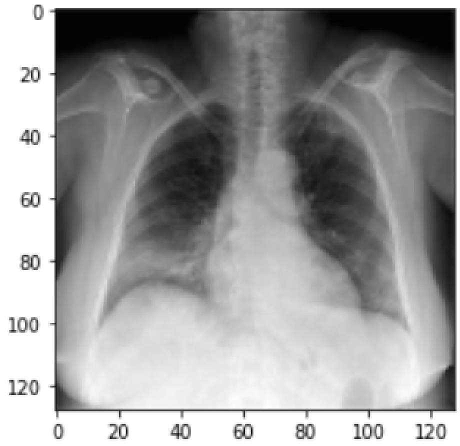


Fig. 1: Chest X-Ray Image

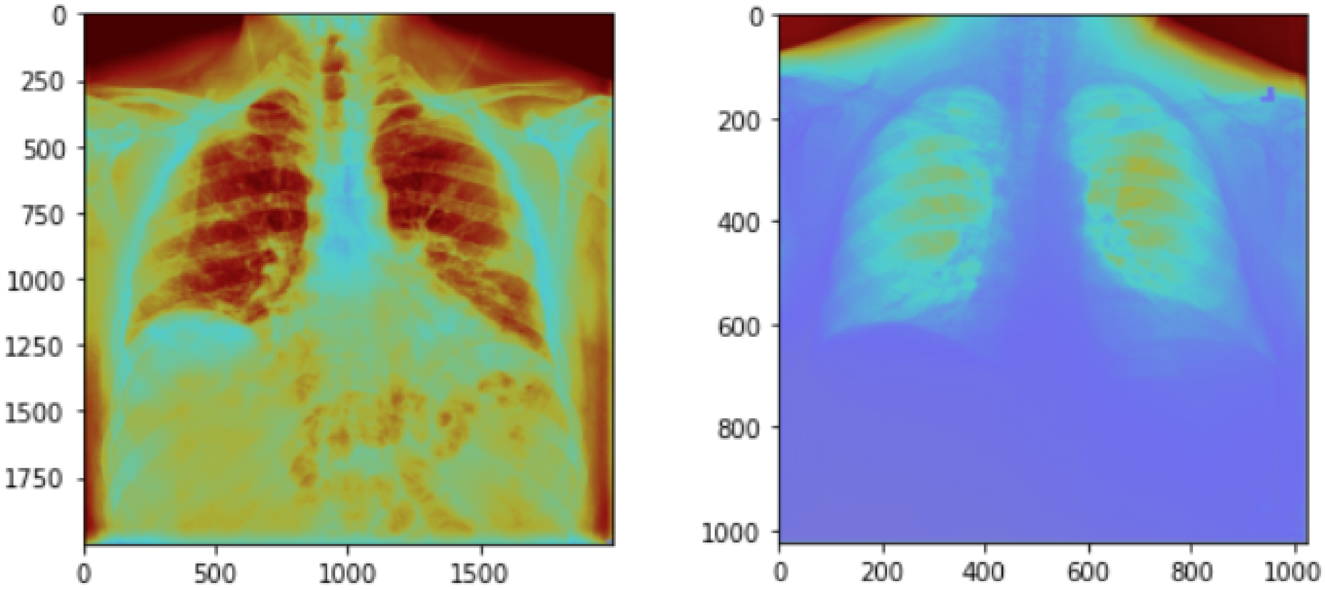


Fig. 2: Heatmap of Images

III. Machine Learning Models

A. Random Forest

This classifier contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Gave an accuracy of 86.2% which was expected as the COVID may affect person at different parts of lungs and the place may be difficult to generalize for a Random Forest Classifer.

B. Bayes Classification - Gaussian

Gaussian Naive Bayes was used. Gaussian (or Normal distribution) need only the mean and the standard deviation from the training data to estimate. Accuracy of 69.1%.This was ofcourse expected for Naive Bayes because this is extremely simple classifier and The assumption that all features are independent is not usually the case in real life so it makes naive bayes algorithm less accurate than complicated algorithms.

C. Light Gradient Boosting - LGBM

Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking and classification.Since it is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise. So when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms. This gave us the accuracy of 88% which was the highest amongst machine learning models.

D. XGBoost

XGBoost(Extreme Gradient Boosting), is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting. Implemented XGBoost without PCA/ LDA and also with PCA, LDA. Gave the accuracy of 86.2%

E. Logistic Regression

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1. This gave accuracy of 78% and it was expected as The reason is that the target label has no linear correlation with the features. In such cases, logistic regression (or linear regression for regression problems) can’t predict targets with good accuracy (even on the training data).

F. SVC

SVC was implented to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. SVC gave an accuracy of 86.7% as SVM works by mapping data to a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable.

G. KNN

In KNeighbours Classifier, a new data point is classified based on similarity in the specific group of neighboring data points. KNN model gave the accuracy of 85.3% which is quite high for KNN as it is difficult for a high dimentional data to predict with high accuracy in KNN because if curse of dimentionality.

H. MLP

Multi-layer Perceptron classifier optimizes the log-loss function using LBFGS or stochastic gradient descent. It is is a feedforward artificial neural network that generates a set of outputs from a set of inputs and uses backpropogation for training the network. Applying MLP Classifier gave an accuracy of 81.5% with PCA.

The models implemented were evaluated using techniques like - Precision, recall, f1 score and accuracy scores.

TABLE I: Evaluation of Models

Model	**Accuracy**	**F1 Score**	**Recall Score**
Random Forest	0.86	0.86	0.88
Bayes Classification	0.69	0.67	0.64
LGBM	0.88	0.86	0.90
XGBoost	0.87	0.87	0.89
Logistic Regression	0.78	0.79	0.78
SVC	0.86	0.88	0.93
KNN	0.85	0.85	0.88
MLP	0.65	0.49	0.34

TABLE II: Evaluation of Models with PCA and LDA

Model	**PCA**	**LDA**
Random Forest	0.74	0.75
Bayes Classification	0.61	0.75
LGBM	0.834	0.763
XGBoost	0.83	0.74
Logistic Regression	0.788	0.759
SVC	0.873	0.751
KNN	0.85	0.75
MLP	0.81	0.76

IV. Deep Learning Models

A. CNN Scratch 1

This is our best model among all others. In preprocessing we have used shear_range and zoom_range for augmentation. 4 convolutional layers and 2 Dense layers hidden with total 771,041 parameters. All the convolutional layers are followed by MaxPooling of size 2 with ReLU activationn with 32-32-64-64 nodes. The Dense layers comprises of 128-64 nodes. This achieved a validation accuracy of 92% and testing accuracy of 93% which proves that the model was able to generalize properly.

B. CNN Scratch 2

This network consists of 4 hidden layers with 933,378 parameters in total. Each convolutional layer is followed by a Maxpooling od size 3 and a dropout layer with ratio of 0.25. Each layer has a activation of ReLU except the last one with has Softmax activation. categorical_crossentropy was used for loss calculation and Adam optimizer was used. The validation accuracy was 91% but the testing accuracy was obtained as 90% which proves that the model was able to generalize the data provided.

C. ResNet50

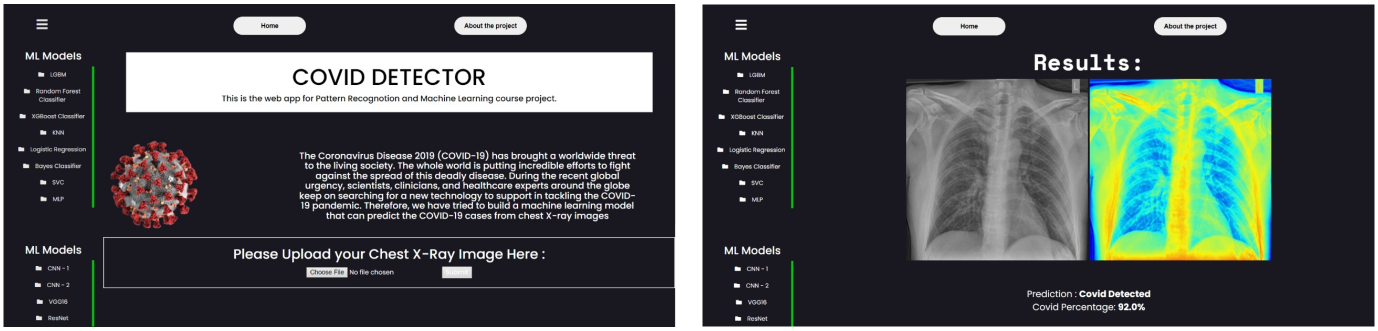
In ResNet, during preprocessing augmentation, horizontal flipping was not used.A convolution with a kernel size of 7x7 and 64 different kernels all with a stride of size 2 followed by a MaxPooling of size 2. Then, a 1x1,64 kernel following this a 3x3,64 kernel and at last a 1x1,256 kernel, all 3 repeated 3 times. Then 1x1, 128 after that a kernel of 3x3,128 and at last a kernel of 1x1,512 this step was repeated 4 time so giving us 12 layers in this step.Then, 1x1,256 and two more kernels with 3x3,256 and 1x1,1024 and this is repeated 6 time giving us a total of 18 layers. Then 1x1,512 kernel with two more of 3x3,512 and 1x1,2048 and this was repeated 3 times giving us a total of 9 layers.Then a connected layer containing 1000 nodes and at the end a softmax function so this gives us 1 layer. This gave us validation accuracy of 90.62% but testing accuracy of 87% which is quite good.

D. VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. This gave us validation accuracy 87% of and testing accuracy of 85% which was expected due to its Vanishing gradient problem.

V. Deployment

We stored all the models that we trained in this project and made a easy to use Web App. As the dependencies aggregatedly crossed 500MB limit of all the major free hosting services, we were not able to deploy the project universally. So we have run the app locally and have attached a few screenshots below.



COVID DETECTOR

This is the web app for Pattern Recognotion and Machine Learning course project.

The Coronavirus Disease 2019 (COVID-19) has brought a worldwide threat to the living society. The whole world is putting incredible efforts to fight against the spread of this deadly disease. During the recent global urgency, scientists, clinicians, and healthcare experts around the globe keep on searching for a new technology to support in tackling the COVID-19 pandemic. Therefore, we have tried to build a machine learning model that can predict the COVID-19 cases from chest X-ray images

Please Upload your Chest X-Ray Image Here :

No file chosen

((a)) Home Page

Results:

Prediction : **Covid Detected**
Covid Percentage: **92.0%**

((b)) Result Page

Fig. 3: Images of Web App

The web app gives an option to select the model for prediction and uses the appropriate selected model to make prediction. This returns original image, heatmap visualization, final COVID/ Non-COVID prediction with percentage of COVID Detected.

References

[1] https://www.kaggle.com/competitions/stat946winter2021/overview
[2] https://github.com/ieee8023/covid-chestxray-dataset
[3] https://towardsdatascience.com/covid-19-detector-flask-app-based-on-chest-x-rays-and-ct-scans-using-deep-learning-a0db89e1ed2a
[4] https://github.com/lindawangg/COVID-Net
[5] https://scikit-learn.org/stable/index.html