PREDICTION OF COVID-19 USING CHEST X RAY IMAGES USING DEEP LEARNING NETWORKS AND MACHINING LEARNING CLASSIFIERS

Dr. Manish Narwaria[1], Aryan Garg[2] and Ayush Shukla[3]

[1]Assistant Professor, [2,3]B.Tech. Students, Indian Institute of Technology Jodhpur.

(Summer Challenge Competition 2020)

Abstract-

In the present situation, the coronavirus (COVID-19) pandemic is putting even the best healthcare systems across the world under tremendous pressure. The early detection of this virus will help in reducing the pressure of the healthcare systems. Chest X-rays has been playing a crucial role in the diagnosis of this virus. As COVID-19 is a type of influenza like Pneumonia, which directly affects the lungs of the patients. So, it is possible to diagnose this virus using imaging techniques. With rapid development in the field of Machine Learning (ML) and Deep learning, there have been intelligent systems that can classify patients between Covid, Pneumonia and Normal. In this paper we have presented the use of Convolutional Neural Networks for image feature extraction from the CXR images and using these image features to train various machine learning models to classify them as COVID, PNEUMONIA or NORMAL. Our model is achieving an accuracy of 96.37% on Random Forest classifier and 95.87% on K-Nearest Neighbor classifier. Image augmentation is also applied to make the dataset balanced and for achieving higher accuracies. Such methodologies can become an alternative way for coronavirus testing, as some countries are facing shortage of testing kits.

Introduction-

COVID-19 or Coronavirus disease has come out as a lethal SARS infection. Believed to have originated from Wuhan, China has rapidly spread over the whole world. Owing to its devastating nature, the World Health Organization (WHO) declared novel COVID-19 a global pandemic. Symptoms of the disease resemble those of the common cold flu or respiratory infection. Later studies showed that severely infected patients faced shortness of breath, pneumonia or even

multiple organ failures. As of now 14-Aug-2020, covid-19 had spread to six continents and according to WHO, 21,067,137 cases have already been confirmed positive to this novel virus where 753,479 people have already lost their lives. India is currently the third worst affected country worldwide having total confirmed cases to be 2,459,613 and death toll rises to 48,144. International travel and international travel history are believed to be a major cause of this global pandemic. Though scientists and researchers are working hard to have a vaccine or a drug to end the pandemic yet a proper cure is not available causing a great panic for human life. Covid-19 cases till 14-Aug-2020 are depicted in **Table 1.**

Country	Confirmed Cases	Recovered	Deaths	Tests per million population
USA	5,415,666	2,843,204	170,415	206,876
Brazil	3,229,621	2,356,640	105,564	63,290
India	2,459,613	1,750,636	48,144	19,431
Russia	907,758	716,396	15,384	216,513
South Africa	572,865	437,617	11,270	55,820
Peru	507,996	348,006	21,713	81,060
Mexico	505,751	341,507	55,293	8,847
Colombia	433,805	250,494	14,145	40,500
Chile	380,034	353,131	10,299	100,994
Spain	355,856	N/A	28,605	159,806

Table 1: Statistics of top 10 worst affected countries due to COVID-19 as on 14-Aug-2020.

Talking more about India, various crucial steps have been by the Prime Minister to control and break the growing chain of the virus. Whole nation had witnessed a complete lockdown of more than 78 days and a partial lockdown is still going on. Lower mortality rate and high recovery rate gives a sigh of relief but whether sufficient testing for the large population is being done or not is still under doubt.

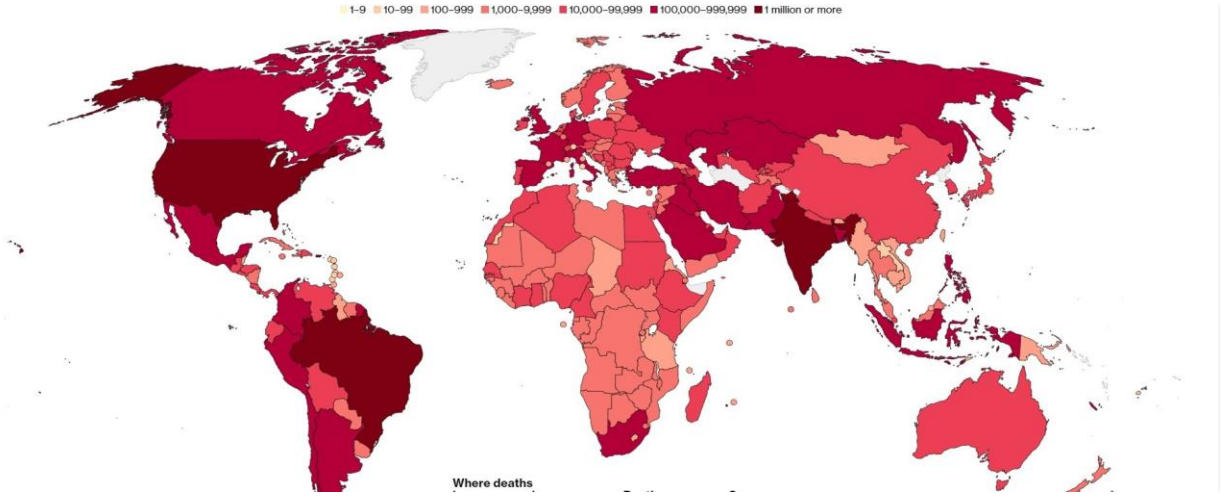


Figure 1. Countries with reported confirmed cases of COVID-19 as on August 14, 2020

Preparation of Dataset-

We have taken a publicly available dataset from Kaggle, which link is provided in the reference section. It contains 1583 images of normal patients, 576 images of Covid positive patients and about 4096 images of pneumonia infected patients (But only 1900 images are being taken, as more images requires more computational power). The images of Covid positive patients are increased to 1776 using image augmentation, to make the dataset balanced.

The dataset has been divided into 3 categories - training set, validation set and testing set, details are shown in below **Table 2**. These values are chosen randomly using the scikit learn train-test split function.

Classes	COVID-19	Normal	Pneumonia	Total
Training set	1500	1266	1500	4266
Test set	166	190	240	596
Validation set	110	127	160	397
Total	1776	1583	1900	5259

Table 2. Showing the distribution of CXR images for different classes.

Image augmentation is applied only on training set, while no augmentation is applied on validation & test sets. This is why the number of testing and validation images are less as compared to training images. Shown below in **Figure 2** are some examples of CXR images of different classes, taken from the dataset.

NORMAL **PNEUMONIA** **COVID-19**

Figure 2. Examples of CXR images from the dataset.

Pre-processing of the Dataset-

It could be seen that all the acquired CXR images are of variable shapes and sizes, which increases the difficulty in effective classification. In order to effectively perform the classification process, image pre-processing is performed.

So, we have taken low resolution colored images and converted all the images into a standard size of 224x224 pixels and then normalize each pixel value between 0 to 1, to ease the calculation process.

Architecture-

In our model we have used Convolutional Neural Network (CNN) for extracting deep image features from the input images. CNN is a deep-learning based algorithm that takes an image as an input and then extracts various image identities like edges, color, gradient, orientation by applying various filters or kernels. ConvNet is reducing the image size without losing important image information. They give the right prediction even if the training data is not huge. They are also less computationally expensive as compared to other algorithms. That are some reasons why we prefer ConvNets in our model.

Then after extracting the image features through CNN, these are fed into ML Classifiers for the final classification process. The detailed explanation of the overall architecture is shown in **Figure 3**. And the architecture of our proposed CNN algorithm is explained in **Figure 4**.

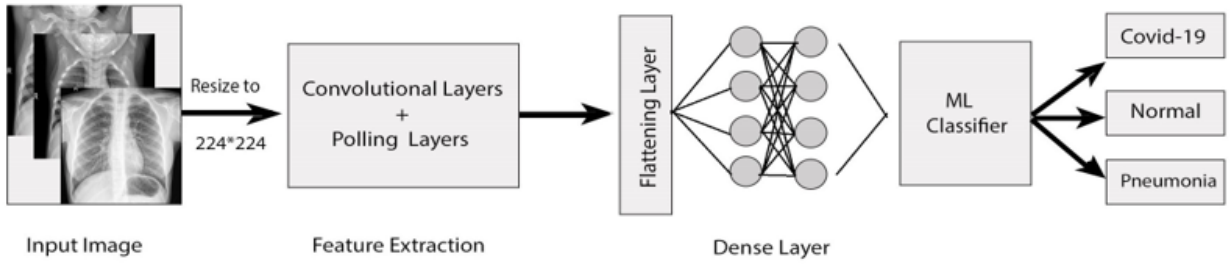


Figure 3. Showing the architecture of our proposed method of classification

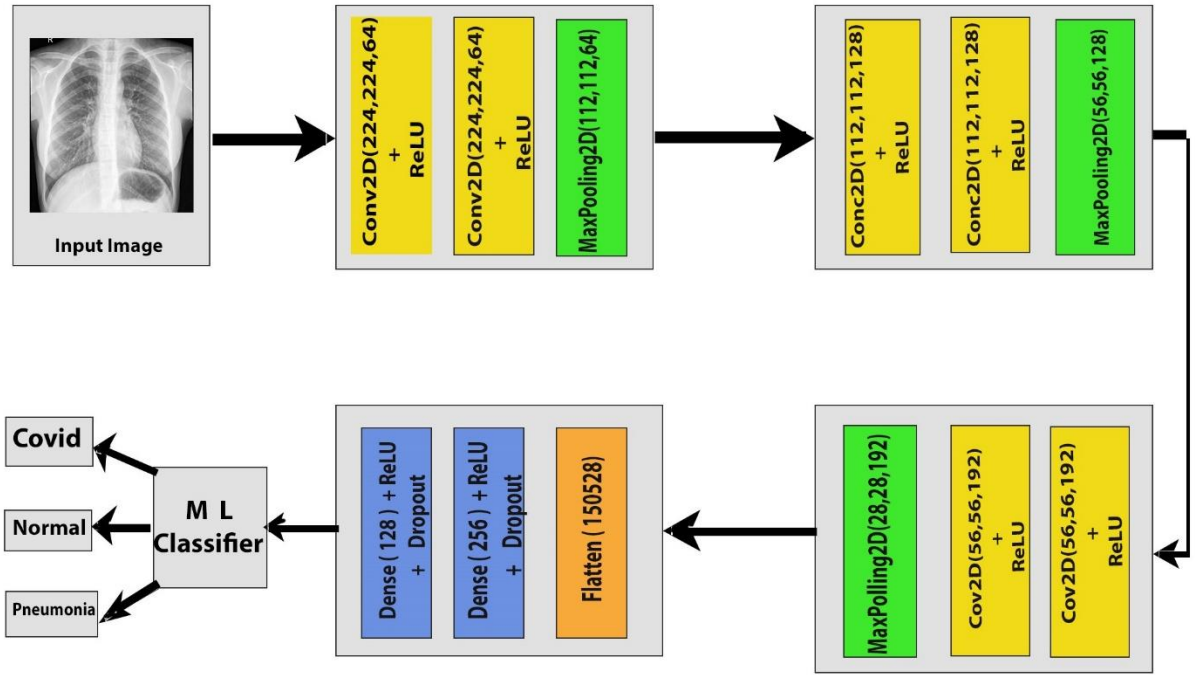


Figure 4. The CNN algorithm used in the proposed model for feature extraction.

1) Input Layer- Training process starts by converting the input image from the user to an Array of size 224 x 224 x 3, where 224 represents the pixel value of image height and width represents the (RGB) color channels.

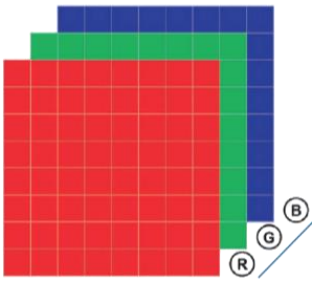


Figure 5. RGB Channels of input image.

2) Convolutional Layer- Convolutional layer is applied to extract image features such as edges, color, brightness, etc. This operation is carried out by applying different types of filters or kernels as shown below in **Figure 6,** that can automatically train themselves to find specific points in an image through several iterations. After this a feature map is created which again passes through different layers.

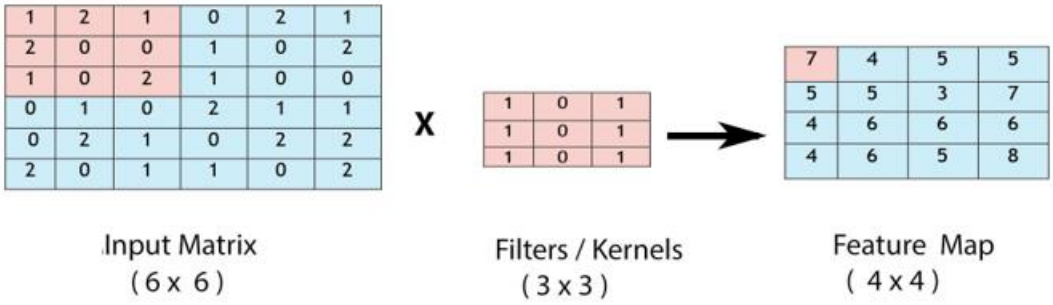


Figure 6. Functioning of Filters/Kernels in Convolutional Layer.

3) ReLU Activation Function- It stands for Rectified Linear Unit. The output of this layer is f(x) = max (0, x). This applied to give linearity to the feature map by converting all the negative values to 0. There are also some other activation functions like sigmoid, tanh etc. ReLU activation function has been applied on every convolutional layer as well as every dense layer in our architecture.

4) Max Pooling Layer- It extracts the maximum value pixel from a region of interest in a feature map, and creates a new map with these maximum values as shown below in **Figure 7**. It is used to decrease the complexity of the image as well its dimensions. Max Pooling discards the noisy

activations from the image without losing accuracy. There are also other pooling techniques like min pooling, average pooling. Max pooling layer has been applied 3 times in our architecture.

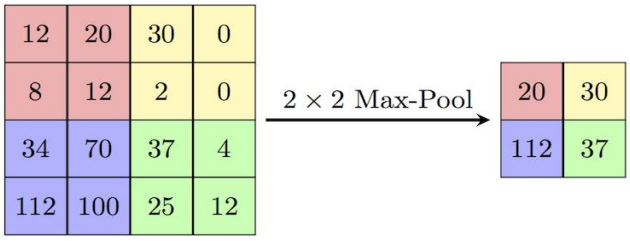


Figure 7. Max Pooling layer

5) Flattening Layer- It is applied to convert the 2D array from Convolutional layers into a single long continuous linear array. After that this linear array is fed to the fully connected layers for the final feature extraction.

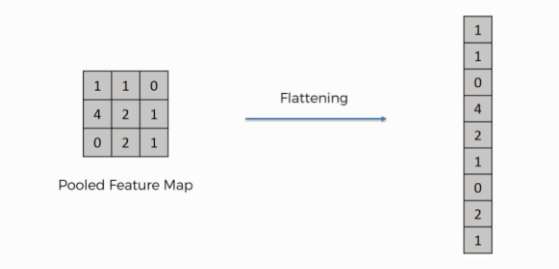


Figure 8. Flattening Layer

6) Dense Layer- It uses the output of the flattened layer to classify the images on the basis of their labels. Basically, the features given by convolutional layers are combined by the group of dense layers to create models. The final output of the dense layer has the same number of nodes as the number of output classes.

But in our case, only 2 dense layers are applied after the flattening layer as seen above in **Figure 4**. 1st layer contains 256 neurons and the second layer contains 128 neurons, and between these two layers a dropout layer is also applied to reduce overfitting in the model.

So, the 2nd dense layer outputs 128 different image features which are fed into various ML predictive classifiers for the final classification of the CXR images as COVID, NORMAL or PNEUMONIA.

Results & Discussions-

The results of final classification by the proposed methodology is shown below in **Table 3**. In this table accuracy score, precision value, recall value and F1-score of various potential machine learning classifiers are given, to evaluate their performance on the classification task. It can be seen from the table that the metrics corresponding to the Random Forest (RF) classifier and K-Nearest Neighbor (NN) classifier outperformed from the rest indicating that they have a better understanding of the image features which were input to them.

Method	XGB Classifier	Random Forest	Logistic Regression	K-Nearest Neighbors	Naive Bayes
Accuracy	95.568	96.374	95.770	95.871	95.669
Precision	95.570	96.377	95.767	95.874	95.670
Recall	95.568	96.374	95.770	95.871	95.669
F1-Score	95.566	96.376	95.768	95.872	95.670

Table3. Showing the performance metrics of various classifiers.

Confusion matrix is also estimated for the best two classifiers i.e. RF and K-NN, to evaluate the efficiency of the proposed framework, which gives a detailed understanding of the classification process as shown below in **Figure 9**. Corresponding graphs of the Receiver Operating Characteristics (ROC) curves for RF and K-NN are being shown below in **Figure 10**.

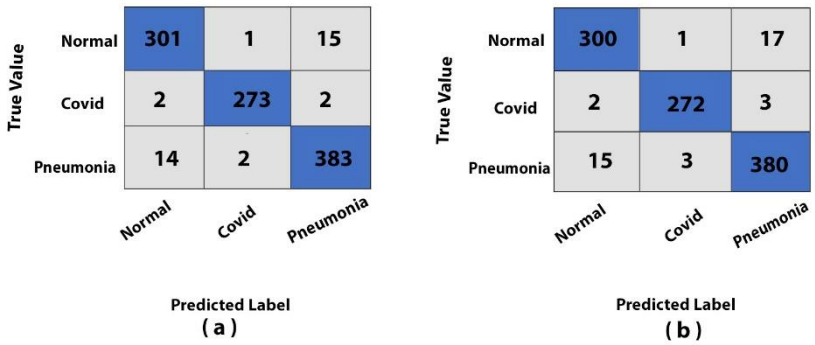
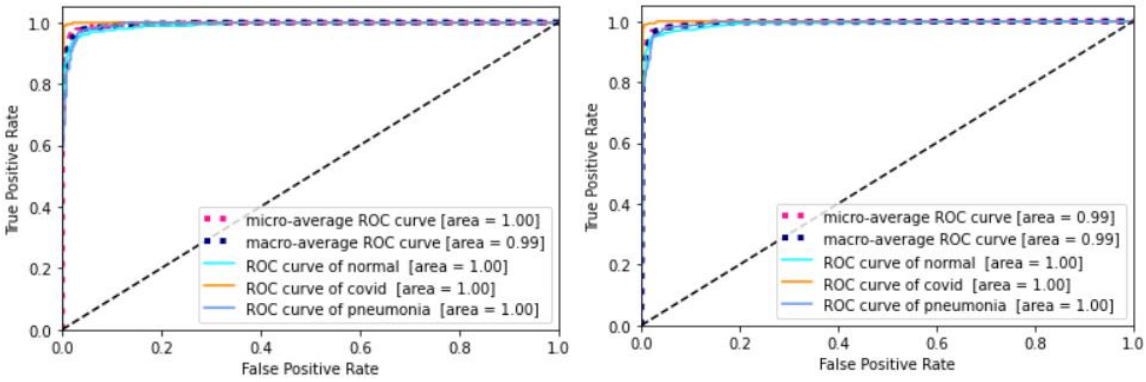


Figure 9. Confusion matrix of the top two accuracy of the model obtained by respective classifiers (a) RF and (b) K-NN with respect to Normal, Covid and Pneumonia CXR images.



(a) **(b)**

Figure 10. Showing the ROC-AUC Curves for (a) RF and (b) K-NN

CNN Model Performance- It has achieved an accuracy of 97% on the training set and 95.64% accuracy on the validation set, after 20 epochs of training. Optimizer function has been taken as ADAM with a learning rate of 1e-4, and the loss function was taken as categorical cross entropy. Further model has also achieved an accuracy of 96.13% on testing data, which was less than the accuracy achieved by Random Forest Classifier on the same test data. Training and testing performances are shown below in **Figure 11**.

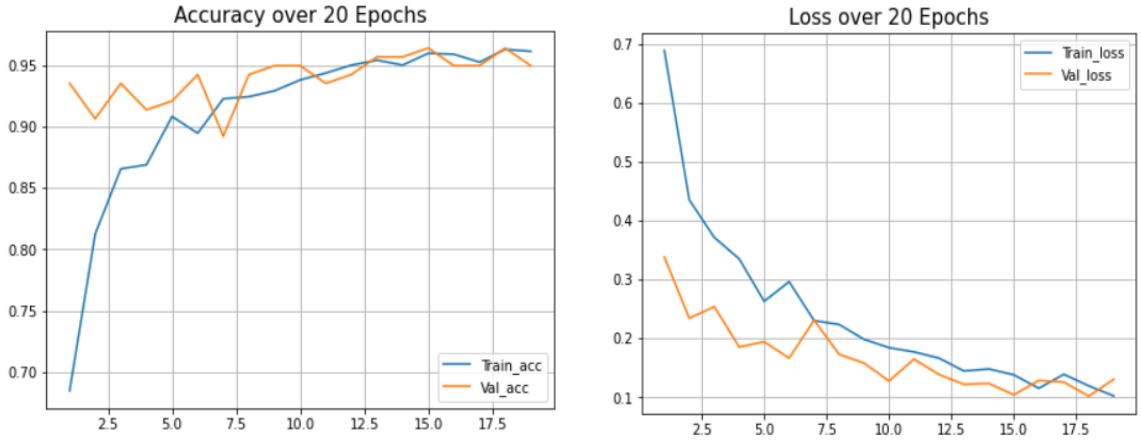


Fig. 11(a). Training & validation accuracy graph. **Fig. 11(b).** Training & validation loss graph.

We have also made a **User-friendly Web Based GUI** in which anyone can upload his/her chest X Ray image by clicking up on the Upload button. Then this CXR image is processed by the trained model and outputs the results.

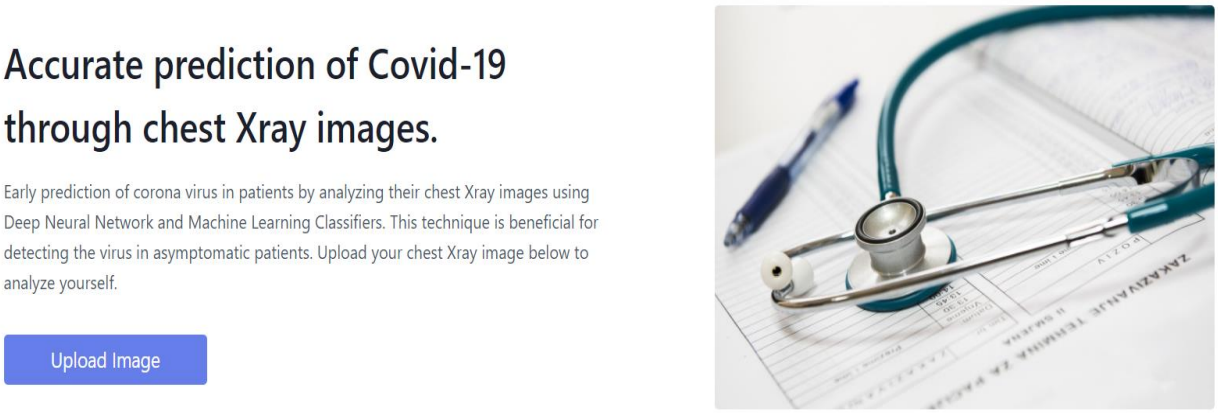


Fig 12. User Interface

This backend interface is made through the python programming language using the Django framework, while the frontend interface is made through HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). These are two of the core technologies for building Web pages. **HTML** provides the structure of the page, **CSS** the (visual and aural) layout, for a variety of devices. **Django** is an open-source python web framework used for rapid development, pragmatic, maintainable, clean design, and secure websites. The **GitHub link** to the **code of complete website** as well as **GitHub link** of the **static page of User Interface** has been provided in the **reference section.**

When **CXR images** of a **PNEUMONIA** infected patient, a **Covid-19** infected patient and a **Healthy** person are uploaded on the **GUI,** the output results of the **GUI** are shown below in **Fig. 13(a), Fig. 13(b)** and **Fig. 13(c)** respectively**.** These results are obtained using Random Forest Classifier.

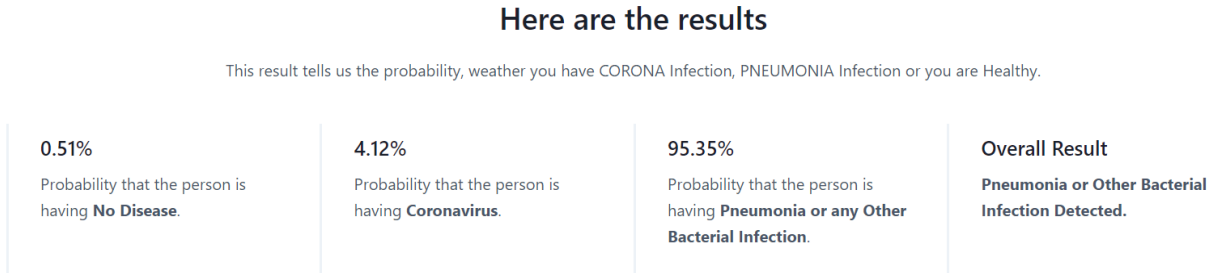


Fig. 13(a). Output (Result - Pneumonia, Accuracy - 95.35%)

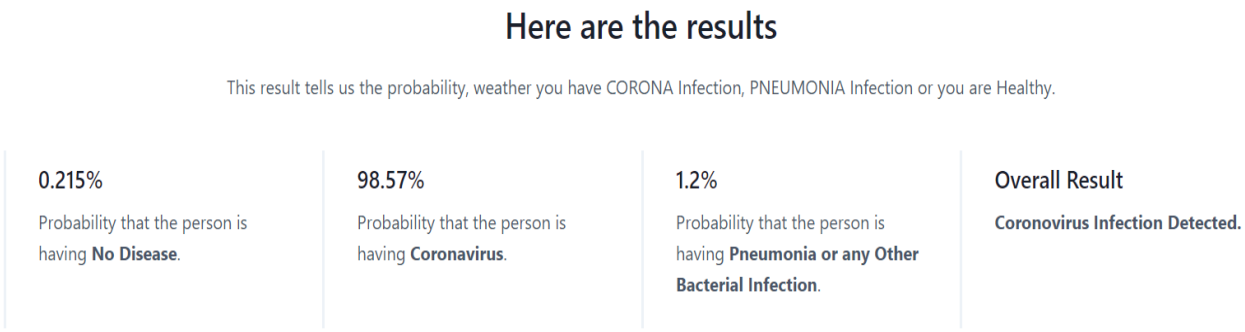


Fig. 13(b). Output (Result – Covid-19, Accuracy - 98.57%)

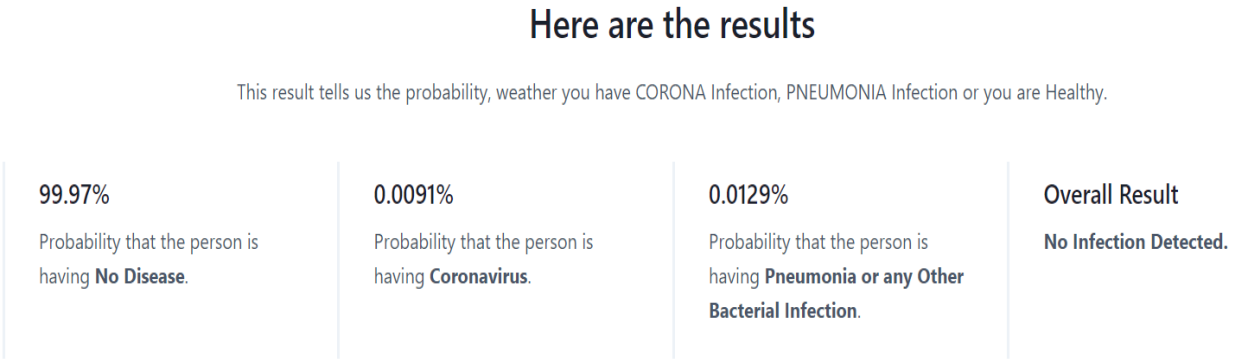


Fig. 13(c). Output (Result - Normal, Accuracy - 99.97%)

Experimental Setups-

We have implemented the proposed classification system for COVID-19 diagnosis using Python 3.8 programming language with a processor of IntelR Core i5-10210U CPU @ 2.30GHz × 8 and RAM of 8GB running on Windows 10 with NVIDIA Geforce MX 110 with 2GB Graphics.

Conclusion-

In this work, we have presented the use of Convolutional Networks and Machine Learning classifiers for the effective classification of COVID-19 through CXR Images. We also augmented the Covid-19 images because to reduce data unbalancing. After Image Feature extraction through CNN, machine learning algorithms are applied for final classification leading to the best result obtained by K-Nearest Neighbor with the Accuracy of 95.87% and 96.37% for Random Forest. Therefore, this approach of using X-ray images and Image Processing Techniques can be used as a massive, faster and cost-effective way of screening. Also, it brings down the time for testing

drastically. To make a clinically effective prediction of COVID-19, training with more massive datasets can lead to better accuracy.

References-

1) World meters: COVID-19 CORONAVIRUS PANDEMIC. Accessed on: August 14, 2020. https://www.worldometers.info/coronavirus/
2) GitHub link to the code of complete website GUI- https://github.com/aryan0141/Covid-19-Detection-Through-CXR-Images
3) GitHub link to the static page of User Interface- https://aryan0141.github.io/Covid_Xray_Detection/
4) Kaggle link to the Dataset- https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia?
5) Ian Goodfellow, Yoshua Bengio, Aaron Courville – Deep Learning (Convolutional Networks), 330 to 372 (2017).
6) Diederik P. Kingma, Jimmy Lei Ba – ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. Arxiv (2015).
7) Wikipedia – Random Forest Classifier. https://en.wikipedia.org/wiki/Random_forest#:~:text=Random%20forests%20or%20random%20decision,prediction%20(regression)%20of%20the%20individual
8) Wikipedia – K-Nearest Neighbor Classifier. https://en.wikipedia.org/wiki/Knearest_neighbors_algorithm#:~:text=In%20pattern%20recognition%2C%20the%20k,used%20for%20classification%20and%20regression.&text=The%20output%20depends%20on%20whether,output%20is%20a%20class%20membership.
9) Wikipedia – Receiver Operating Characteristics (ROC). https://en.wikipedia.org/wiki/Receiver_operating_characteristic
10) Wikipedia – Python Django Framework. https://en.wikipedia.org/wiki/Django_(web_framework)
11) Mozilla Organization – Cascading Style Sheets (CSS). https://developer.mozilla.org/en-US/docs/Web/CSS