COVID-19 PREDICTION USING X-RAY

Submitted in partial fulfillment of the requirements of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

Gautam Sathe 231

Aniket Singh 240

Shivendra Singh 242

Kovid Thalia 244

Name of the Guide

Prof. Sanjivani Deokar

Department of Computer Engineering

Lokmanya Tilak College of Engineering

Koparkhairane, Navi Mumbai - 400 709

University of Mumbai

(AY 2021-22)

CERTIFICATE

This is to certify that the Mini Project entitled **"Covid-19 prediction using X-ray "** is a bonafide work of **Gautam Sateh (TEB231), Aniket Singh(TEB 240), Shivendra Singh (TEB242), Kovid Thalia (TEB244)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering" .**

(Prof. Sanjivani Deokar)
Project Guide

(Prof.Rajendra Gawali)
Head of Department

(Dr.Vivek Sunnapawar)
Principal

Mini Project Approval

This Mini Project titled "**Covid-19 prediction using X-ray"** by **Gautam Sathe(231), Aniket Singh(240), Shivendra Singh(242), Kovid Thalia(244)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Contents

Abstract **ii**

Acknowledgments **iii**

1 Introduction **1**

1.1 Introduction

1.2 Motivation

1.3 Problem Statement & Objectives

2 Literature Survey **10**

2.1 Survey of Existing System/**SRS**

2.2 Limitation Existing system or Research gap

3 Proposed System **13**

3.1 Introduction

3.2 Architecture/ Framework

3.3 Algorithm and Process Design

3.4 Experiment and Results for Validation and Verification

3.5 Conclusion and Future work.

References **32**

Abstract

Artificial intelligence (AI) techniques in general and convolutional neural networks (CNNs) in particular have attained successful results in medical image analysis and classification. A deep CNN architecture has been proposed in this paper for the diagnosis of COVID-19 based on the chest X-ray image classification. Due to the nonavailability of sufficient-size and good-quality chest X-ray image dataset, an effective and accurate CNN classification was a challenge. To deal with these complexities such as the availability of a very-small-sized and imbalanced dataset with image-quality issues, the dataset has been preprocessed in different phases using different techniques to achieve an effective training dataset for the proposed CNN model to attain its best performance. The preprocessing stages of the datasets performed in this study include dataset balancing, medical experts' image analysis, and data augmentation. The experimental results have shown the overall accuracy as high as 99.5% which demonstrates the good capability of the proposed CNN model in the current application domain. The CNN model has been tested in two scenarios. In the first scenario, the model has been tested using the 100 X-ray images of the original processed dataset which achieved an accuracy of 100%. In the second scenario, the model has been tested using an independent dataset of COVID-19 X-ray images. The performance in this test scenario was as high as 99.5%. To further prove that the proposed model outperforms other models, a comparative analysis has been done with some of the machine learning algorithms. The proposed model has outperformed all the models generally and specifically when the model testing was done using an independent testing set.

Acknowledgements

We remain immensely obliged to Prof. Sanjivani Deokar for providing us with the idea of this topic, and for her invaluable support in gathering resources for me either by way of information or computer, also her guidance and supervision which made this project successful. We would like to thank Mini Project Coordinators, Prof. R.D.Gawali Head, Computer Engineering Department , Dr. S.K. Shinde, Vice Principal, and Dr. Vivek Sunnapwar, Principal, LTCoE. We are also thankful to faculty and staff of the Computer Engineering Department and Lokmanya Tilak College of Engineering, Navi Mumbai for their invaluable support. We would like to say that it has indeed been a fulfilling experience working out this project topic.

1. Introduction

Coronavirus illness is a disease that comes from Severe Acute Respiratory Syndrome (SARS) and Middle East Respiratory Syndrome (MERS). A novel coronavirus, COVID-19, is the infection caused by SARS-CoV-2 (Zhang, 2020). In December 2019, the first COVID-19 cases were reported in Wuhan city, Hubei province, China (Xu et al., 2020). World Health Organization (WHO) declared COVID-19 a pandemic (Ducharme, 2020) on March 11 2021, up to July 13 of 2021 there are 188,404,506 reported cases around the world, which have caused 4,059,220 deaths (Worldometer, 2020).

These diseases cause respiratory problems that can be treated without specialized medicine or equipment. Still, underlying medical issues such as diabetes, cancer, cardiovascular and respiratory illnesses can make this sickness worse (World Health Organization, 2020). Reverse transcription Polymerase chain reaction (RT-PCR), gene sequencing for respiratory or blood samples are now the main methods for COVID-19 detection (Wang et al., 2020). Other studies show that COVID-19 has similar pathologies presented in pneumonic illness, leaving chest pathologies visible in medical images. Research shows RT-PCR correlation with Chest CT (Ai et al., 2020), while others study its correlation with X-ray chest images (Kanne et al., 2020). Typical opacities or attenuation are the most common finding in these images, with ground-glass opacity in around 57% of cases (Kong & Agarwal, 2020). Even though expert radiologists can identify the visual patterns found in these images, considering monetary resources at low-level medical institutions and the ongoing increase of cases, this diagnostic process is quite impractical. Recent research in Artificial Intelligence (AI), especially in Deep Learning approaches, shows how these techniques applied to medical images performed well.

There are only a few large open access datasets of COVID-19 X-ray images; most of the published studies use as a foundation the COVID-19 Image Data Collection (Cohen et al., 2020), which was constructed with images from COVID-19 reports or articles, in collaboration with a radiologist to confirm pathologies in the pictures taken.

Past approaches use different strategies to deal with small datasets such as transfer learning, data augmentation or combining different datasets, finding good results in papers as Civit-Masot et al. (2020) using a VGG16 with 86% accuracy; Ozturk et al. (2020) with a Dark Covid Net presents 87% accuracy classifying three classes in which is included Covid; Yoo et al. (2020) used a ResNet18 obtaining a 95% accuracy; Sethy et al. (2020) used a ResNet50 for a 95.33% accuracy, and Minaee et al. (2020) used Squeeze Net for a 95.45%

accuracy; Panwar et al. (2020) achieved 97.62% using a nCovnet; Apostolopoulos and Mpesiana (2020) improved the results using a VGG19-MobileNet with a 97.8% accuracy, and finally higher results are found in Jain et al. (2020) using a ResNet101 with 98.95% and Khan et al. (2020) with a 99% accuracy using CoroNet a model based on an Xception.

So we need a new approach which will focus on enhancing the preprocessing stage to obtain accurate and reliable results classifying COVID-19 from Chest X-ray images. The preprocessing step involves a network to filter the images based on the projection it is (lateral or frontal), some common operations such as normalization, standardization, and resizing to reduce data variability, which may hurt the performance of the classification models, and a segmentation model (U-Net) to extract the lung region which contains the relevant information, and discard the information of the surroundings that can produce misleading results (de Informática, 2020). Following the preprocessing stage comes the classification model (VGG16-19), using the transfer learning scheme that takes advantage of pre-trained weights from a much bigger dataset, such as ImageNet, and helps the training process of the network in performance and time to convergence. It is worth noting that the dataset used for this research is at least ten times bigger than the ones used in previous works. Finally, the visualization of heatmaps for different images provides helpful information about the regions of the images that contribute to the prediction of the network, which in ideal conditions should focus on the appearance of the lungs, backing the importance of lung segmentation in the preprocessing stage. After this section, the paper follows the next order: first, the Methodology applied for these approaches, followed by the experiments and results obtained, a discussion of the products, and lastly the conclusions.

1.1 Motivation:

- Coronavirus disease (COVID-19) a relatively new found virus has caused a pandemic all over the world. For the last 2 year we have suffered and witnessed the effects of the virus. As the testing of coronavirus happened manually in the initial stage, the ever-increasing number of COVID-19 cannot be handled efficiently. Also, the coronavirus is divided into 3 phases and it has different effects on lungs.
- To handle the above situation we have attempted to detect coronavirus using chest X-ray images and Chest CT scan images by using Artificial Intelligence technologies. AI helps to forecast the coronavirus cases for analyzing the virus structure and chest X-Ray and CT scan images helps to predict the stages of coronavirus.
- To fight against nCOVID-19 epidemic, the recent machine learning (ML) techniques can be embedded to develop an automatic computer-aided diagnosis (CAD) system. In this direction, many clinical

and radiological studies have been reported, describing various radio-imaging findings and epidemiology of nCOVID-19

- Further, many deep-learning models like deep convolutional network, recursive network, transfer learning models, etc. have been implemented to automatically analyze the radiological disease characteristics used a convolutional neural network (CNN) to assign a class label to different superpixels extracted from the lungs parenchyma and localize tuberculosis-infected regions in CXR images with average dice index of 0.67.

1.3 Problem Statement And Objective

Problem Statement

Most detection methods of coronavirus disease 2019 (COVID-19) use classic image classification models, which have problems of low recognition accuracy and inaccurate capture of modal features when detecting chest X-rays of COVID-19. This study proposes a COVID-19 detection method based on image modal feature fusion. This method first performs small-sample enhancement processing on chest X-rays, such as rotation, translation, and random transformation. Five classic pre-training models are used when extracting modal features. A global average pooling layer reduces training parameters and prevents overfitting. The model is trained and fine-tuned, the machine learning evaluation standard is used to evaluate the model, and the receiver operating characteristic (ROC) curve is drawn. Experiments show that compared with the classic model, the classification method in this study can more effectively detect COVID-19 image modal information, and it achieves the expected effect of accurately detecting cases.

Objectives:

- To handle the situation we have attempted to detect coronavirus or pneumonia using chest X-ray images and Chest CT scan images by using Artificial Intelligence technologies.
- To predict the coronavirus or pneumonia using the data collected from the testing.
- To create a system which would help in detection irrespective of the changing nature of the virus
- Create a Training Model to check the probability/chances of X-ray report being positive and in that process to develop a Convolutional Neural Network (CNN) Model.
- To save lives!

2. Literature Survey

2.1 Survey of Existing System/SRS

Paper 1 – COVID-19 Diagnosis using X-Ray Images and Deep learning

Year – 2021

Authors – Shivani Sharma; Shamik Tiwari

Methodology –In this paper, 3 models were developed. Namely CNN, VGG16 and VGG19 for multi classification of 3 classes (Covid-19, Pneumonia and Normal). Accuracy of the VGG16 and VGG19 was better (i.e 97%) than the CNN model(i.e 94%). But VGG16 and VGG19 use more computational power where CNN model took less computational power in comparison to VGG16 and VGG19.

Conclusion/Result – he proposed CNN model provides good accuracy on less number of layers and less number of epochs in comparison to pre-trained models.This CNN model is better than other models as it is less complex than others and achieving good accuracy on training, validation as well as on test dataset. CNN not only has less layers than the pre-built models, but also less number of epochs

Paper 2 – Automatic COVID-19 detection from X-ray images using ensemble learning with convolutional neural network

Year - 2021

Authors - Amit Kumar Das, Sayantani Ghosh, Samiruddin Thunder, Rohit Dutta, Sachin Agarwal, Amlan Chakrabarti

Methodology –In this paper, multiple CNN models—DenseNet201, Resnet50V2 and Inceptionv3, have been adopted in the proposed work. 538 images of COVID +ve patients and 468 images of COVID –ve patients have been divided into training, test and validation sets. The proposed approach

gave a classification accuracy of 91.62% which is higher than the state-of-the-art CNN models.

Conclusion/Result –The proposed model has achieved a classification accuracy of 91.62%. It yields a sensitivity of around 95% for COVID +ve cases i.e., out of 100 COVID +ve patients, more than 95 can be correctly diagnosed by the proposed model

Paper 3 - Pneumonia detection in chest X-ray images using an ensemble of deep learning models

Year - 2021

Author - Rohit Kundu, Ritacheta Das, Zong Woo Geem, Gi-Tae Han, Ram Sarkar

Methodology –In this study, they designed an ensemble of three convolutional neural network models: GoogLeNet, ResNet-18, and DenseNet-121. The proposed approach was evaluated on two publicly available pneumonia X-ray datasets using a five-fold cross-validation scheme. The proposed method achieved accuracy rates of 98.81% and 86.85% and sensitivity rates of 98.80% and 87.02% on the Kermany and RSNA datasets, respectively.

Conclusion/Result – In some instances this ensemble framework failed to produce correct predict

Paper 4 - COVID-19 detection in X-ray images using convolutional neural networks

Year - 2021

Authors - Daniel Arias-Garzón, Jesús AlejandroAlzate-Grisales, Simon Orozco-Arias, Harold Brayan Arteaga-Arteaga, Mario Alejandro Bravo-Ortiz, Alejandro Mora-Rubio, Jose Manuel Saborit-Torres, Joaquim Ángel Montell Serrano, Maria de la Iglesia Vayá, Oscar Cardona-Morales, Reinel Tabares-Soto

Methodology - This papers' approach uses existing deep learning models (VGG19 and U-Net) to process the images and classify them as positive or negative for COVID-19.

The proposed system involves a preprocessing stage with lung segmentation, removing the surroundings which does 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 and interpretation via heat maps visualization.

The best models achieved a detection accuracy of COVID-19 around 97%.

Conclusion/Result - This approach shows how existing models can be helpful for multiple tasks, especially if it is considered that the changed U-Net models do not have better performance. Image noise can generate bias in the models.

2.2 Limitations of Existing System

- Manual Checking Of X-ray by doctors.
- Sometimes errors in detecting the probability of X-ray diseases.
- Manual Accuracy is comparably low.
- It is a time Consuming process because it needs to go to the doctor after collecting an x-ray report.
- High Cost.

3. Proposed System

3.1 Introduction

In our proposed system or project we have developed CNN model to predict the x-ray report to identify the disease positive and negative which is an input data from the user. So after going through all the layers we have achieved a great accuracy in our project which is on average 96% approx so we can say that our model after deployment will give us most probably much more accurate result.

Adding on one more additional and important feature to the outcome we are using cross entropy loss in our model for inputs from users, which will give the probability and chances of the x-ray report being positive and negative. This feature will literally tell the user how affected the chest of that person is, to avoid future consequences.

As after developing our model we have deployed it using the flask framework, we have created a user friendly interface. So that all kinds of users can use our web application easily and much more efficiently. This System will let us avoid a long queue for Doctors Offline Diagnosis, Which will save us lots of time and money.

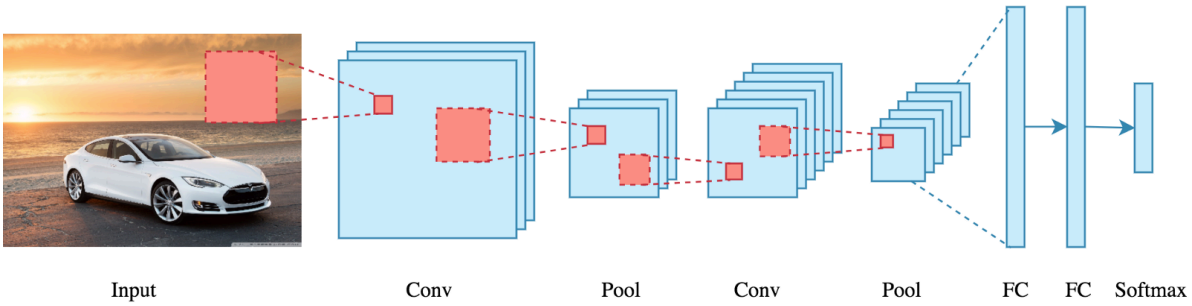
Also it will resolve one of the important problems i.e when the doctor's internship period is going on, since they are not having experience they can’t accurately tell us the result of the report. So this kind of system will help both those doctors to check their diagnosis skill and other patients to avoid getting wrongly diagnosed.

3.2 Architecture and Framework

General Framework

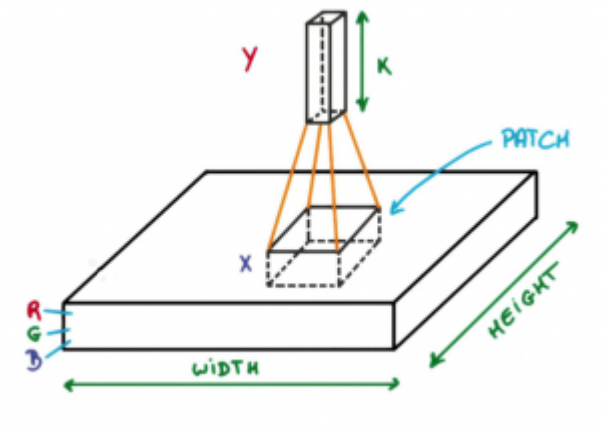
CNN (Convolutional Neural Network)

In deep learning, a **convolutional neural network** (**CNN/ConvNet**) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics **convolution** is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.



Now let's talk about a bit of mathematics that is involved in the whole convolution process.

- Convolution layers consist of a set of learnable filters (a patch in the above image). Every filter has small width and height and the same depth as that of input volume (3 if the input layer is image input).
- For example, if we have to run convolution on an image with dimension 34x34x3. The possible size of filters can be axax3, where 'a' can be 3, 5, 7, etc but small as compared to image dimension.
- During forward pass, we slide each filter across the whole input volume step by step where each step is called stride (which can have value 2 or 3 or even 4 for high dimensional images) and compute the dot product between the weights of filters and patch from input volume.
- As we slide our filters we'll get a 2-D output for each filter and we'll stack them together and as a result, we'll get output volume having a depth equal to the number of filters. The network will learn all the filters.



Layers used to build ConvNets

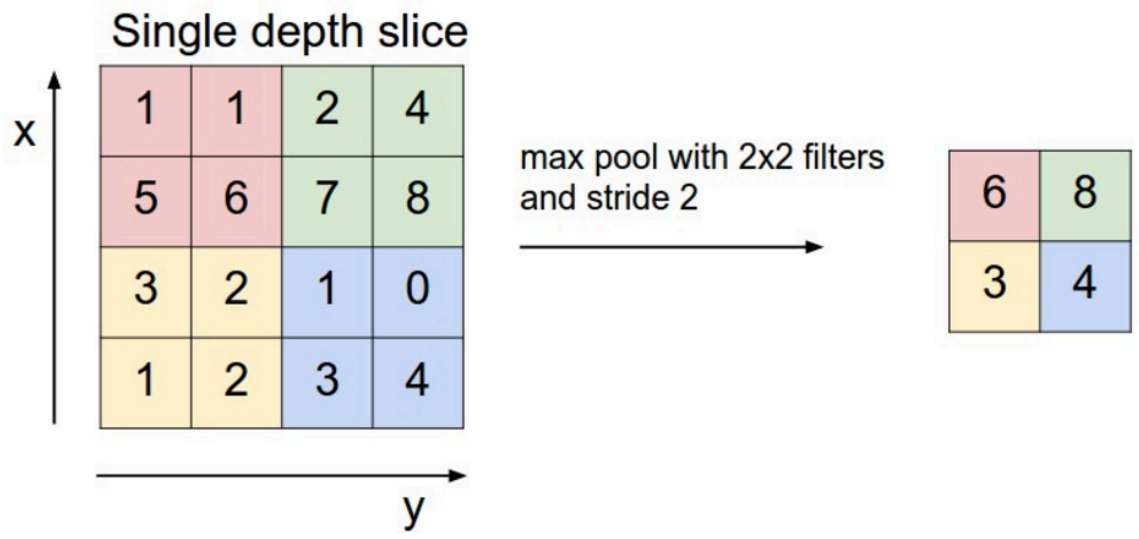
A covnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

Types of layers:

Let's take an example by running a covnets on an image of dimension 32 x 32

x 3.

1. **Input Layer:** This layer holds the raw input of the image with width 32, height 32, and depth 3.
2. **Convolution Layer:** This layer computes the output volume by computing the dot product between all filters and image patches. Suppose we use a total of 12 filters for this layer we'll get output volume of dimension 32 x 32 x 12.
3. **Activation Function Layer:** This layer will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Sigmoid: 1/(1+e^-x), Tanh, Leaky RELU, etc. The volume remains unchanged hence output volume will have dimension 32 x 32 x 12.
4. **Pool Layer:** This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast, reduces memory and also prevents overfitting. Two common types of pooling layers are **max pooling** and **average pooling**. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 16x16x12.



Layers used to build ConvNets

A covnets is a sequence of layers, and every layer transforms one volume to another through a differentiable function.

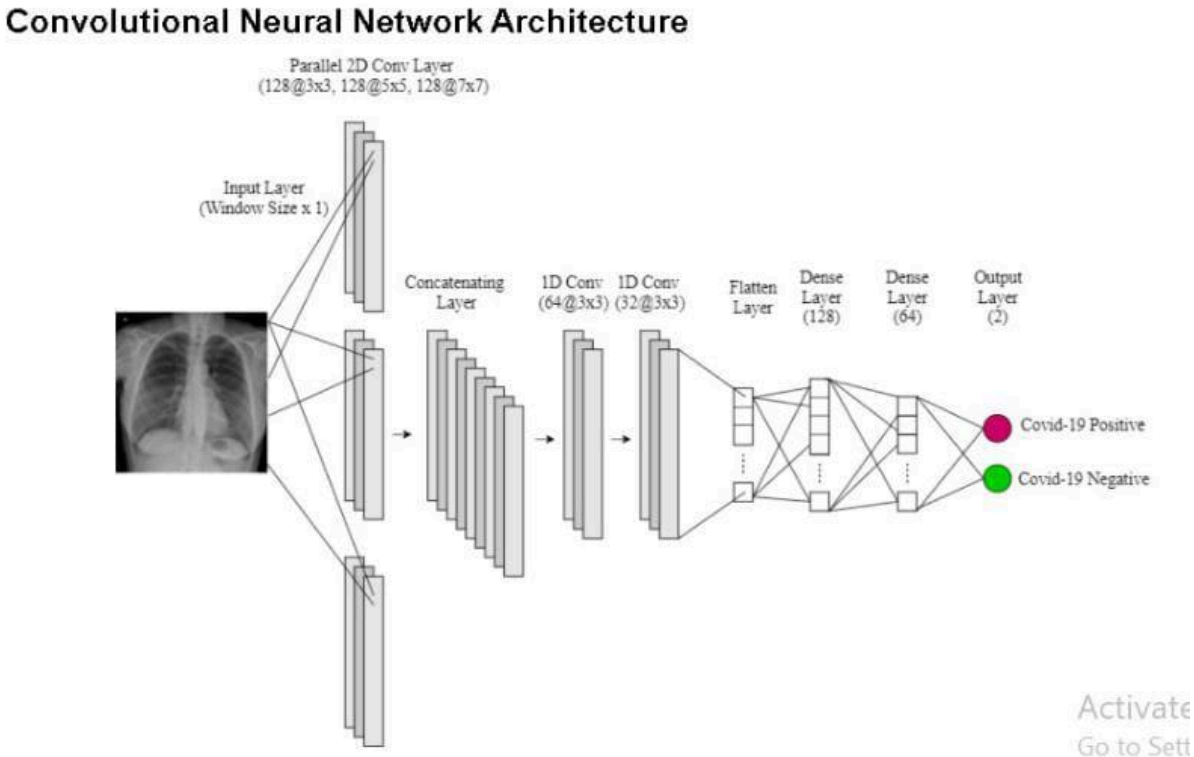
Types of layers:

Let's take an example by running a covnets on an image of dimension 32 x 32 x 3.

Input Layer: This layer holds the raw input of the image with width 32, height 32, and depth 3.

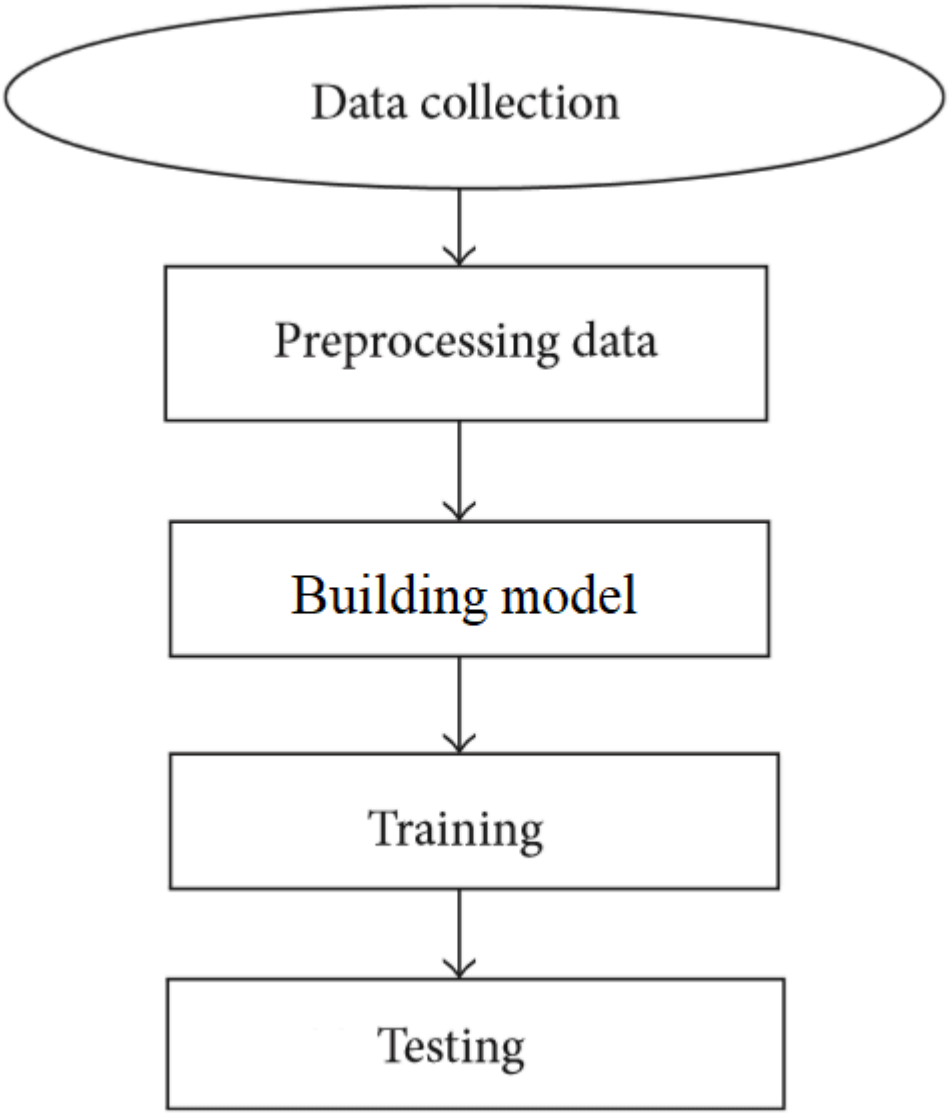
Convolution Layer: This layer computes the output volume by computing the dot product between all filters and image patches.

Proposed Framework



In our system, first we take images as an input from the user. Then, we extract certain features from the images and arrange them in parallel manner with 128 kernels (3 5 7). After which we concatenate all the parallel layers, then again segregating two different sets of parallel layers out of which one is 64 kernels and the second one is 32 kernels. Next, we extract the flattened layers from the previous layer again segregating in two different dense layers of 128 kernels and 64 kernels. Once all the previous features are extracted, we come out with two different neurons i.e. Covid-19 positive and COvid-19 negative.

3.3 Algorithm and Process Design



Implementation:

1) Get the Dataset

To create a machine learning model, the first thing we require is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the **dataset**.

Dataset may be of different formats for different purposes, such as, if we want to create a machine learning model for business purpose, then the dataset will be different with the dataset required for a liver patient. So each dataset is different from another dataset. To use the dataset in our code, we usually put it into a CSV **file**. However, sometimes, we may also need to use an HTML or xlsx file.

2) Importing Libraries

In order to perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs.

3) Importing the Datasets

Now we need to import the datasets which we have collected for our machine learning project. But before importing a dataset, we need to set the current directory as a working directory.

Extracting dependent and independent variables:

In machine learning, it is important to distinguish the matrix of features (independent variables) and dependent variables from the dataset.To extract an independent variable, we will use the **iloc[ ]** method of the Pandas library. It is used to extract the required rows and columns from the dataset.

Extracting dependent variable:

To extract dependent variables, again, we will use Pandas .iloc[] method.

4) Handling Missing data:

The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Ways to handle missing data:

There are mainly two ways to handle missing data, which are:

By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc. Here, we will use this approach.

To handle missing values, we will use **Scikit-learn** library in our code, which contains various libraries for building machine learning models

5) Encoding Categorical data:

Categorical data is data which has some categories such as, in our dataset; there are two categorical variables, **Country**, and **Purchased**.

Since the machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers.

6) Splitting the Dataset into the Training set and Test set

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model.

Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

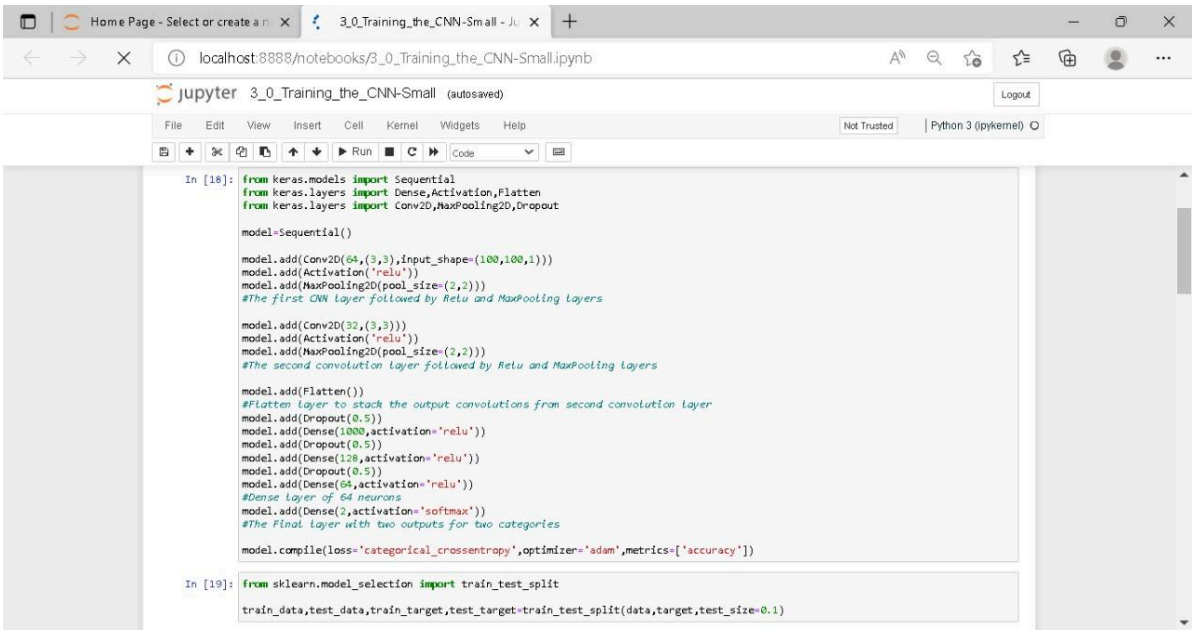
Training Set: A subset of dataset to train the machine learning model, and we already know the output.

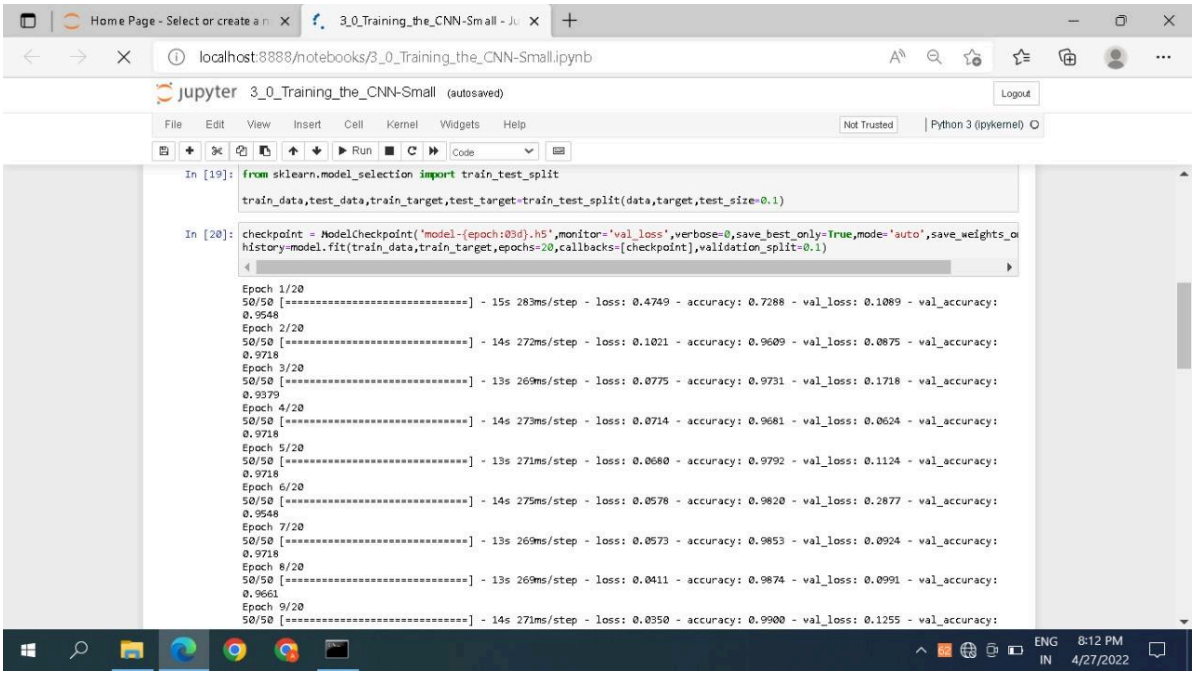
Test set: A subset of dataset to test the machine learning model, and by using the test set, the model predicts the output.

7) Feature Scaling

Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no variable dominates the other variable.

3.4 Experiment and Results for Validation and Verification





Home Page - Select or create a n | 3_0_Training_the_CNN-Small - Ju

localhost:8888/notebooks/3_0_Training_the_CNN-Small.ipynb

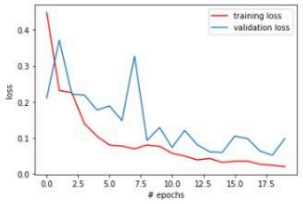
jupyter 3_0_Training_the_CNN-Small (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run Code

In [11]:
```
from matplotlib import pyplot as plt

plt.plot(history.history['loss'],'r',label='training loss')
plt.plot(history.history['val_loss'],label='validation loss')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```



In [12]:
```
plt.plot(history.history['accuracy'],'r',label='training accuracy')
plt.plot(history.history['val_accuracy'],label='validation accuracy')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

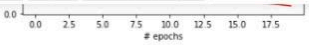
Home Page - Select or create a n | 3_0_Training_the_CNN-Small - Ju

localhost:8888/notebooks/3_0_Training_the_CNN-Small.ipynb

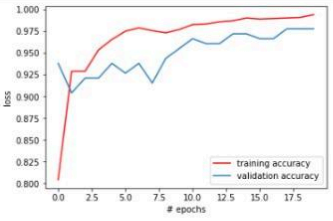
jupyter 3_0_Training_the_CNN-Small (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted | Python 3 (ipykernel)

Run Code



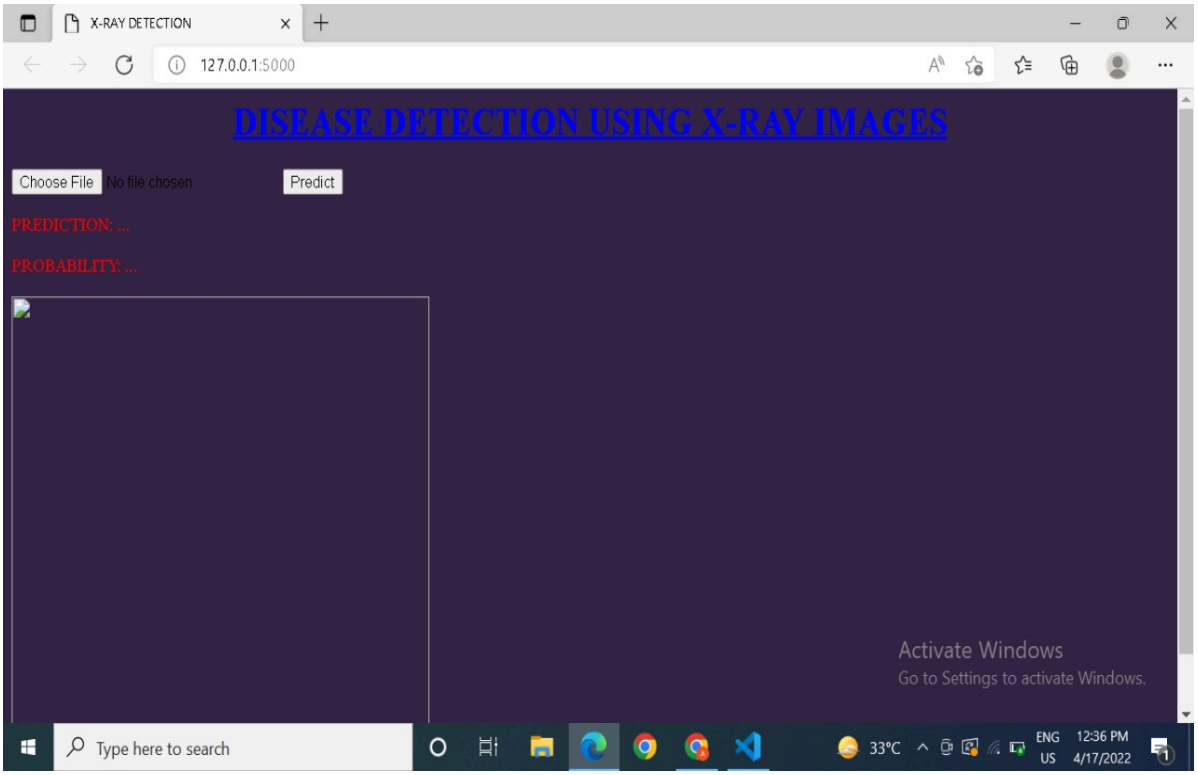
In [12]:
```
plt.plot(history.history['accuracy'],'r',label='training accuracy')
plt.plot(history.history['val_accuracy'],label='validation accuracy')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

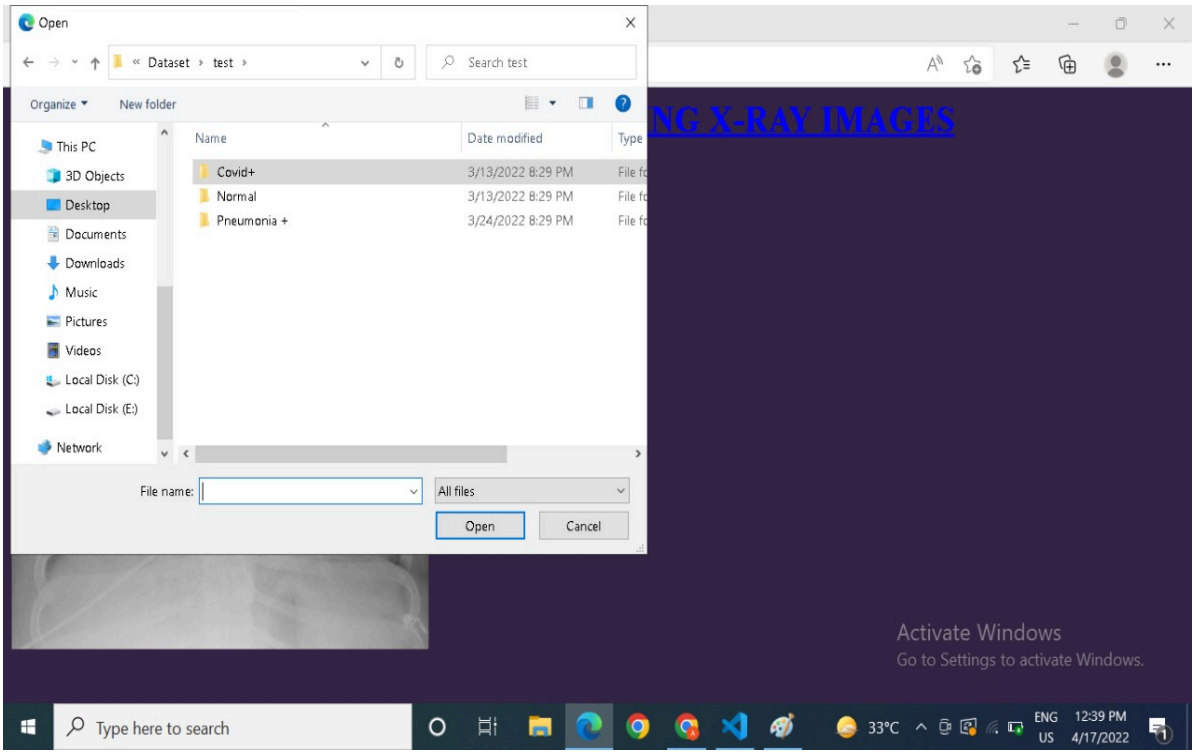


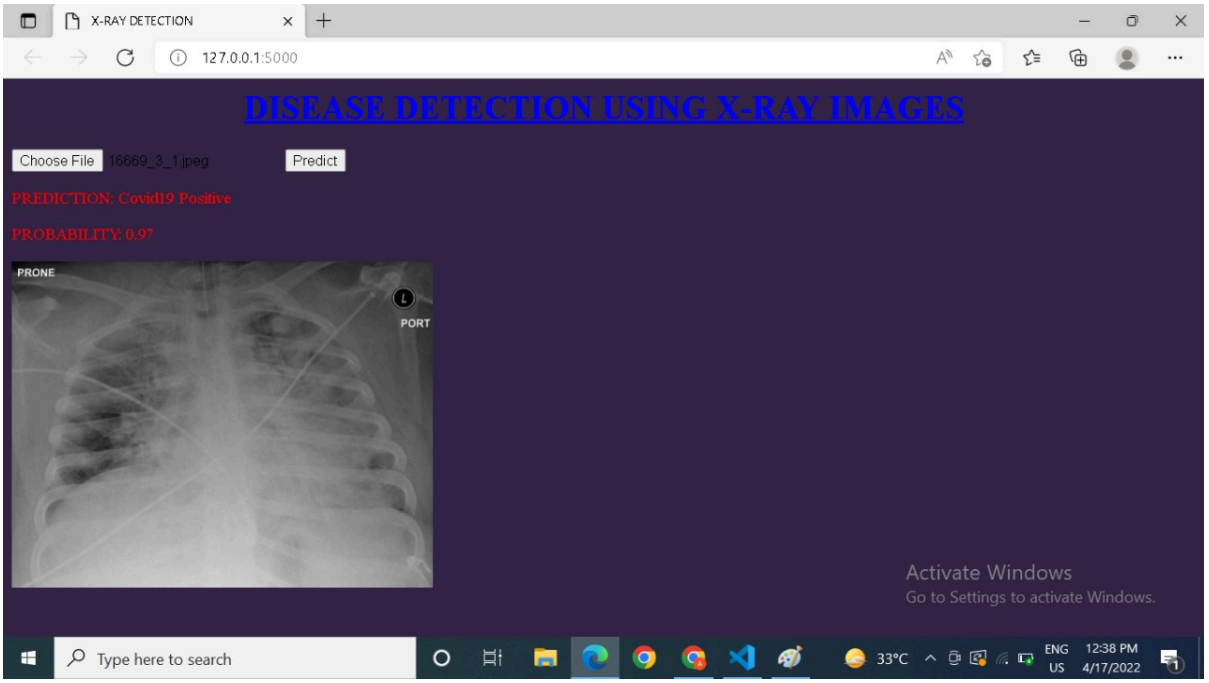
In [13]:
```
print(model.evaluate(test_data,test_target))
```

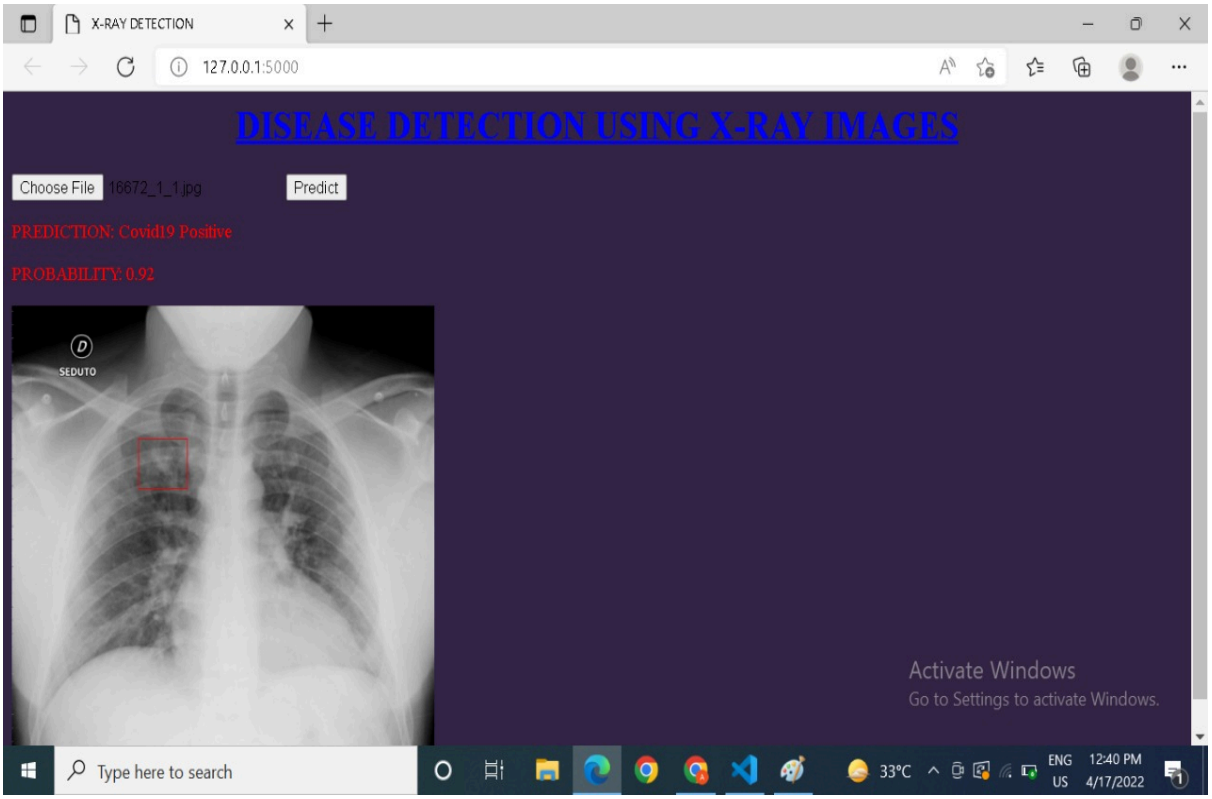
```
7/7 [==============================] - 5s 746ms/step - loss: 0.0319 - accuracy: 0.9898
[0.03191700205206871, 0.9897959232330322]
```

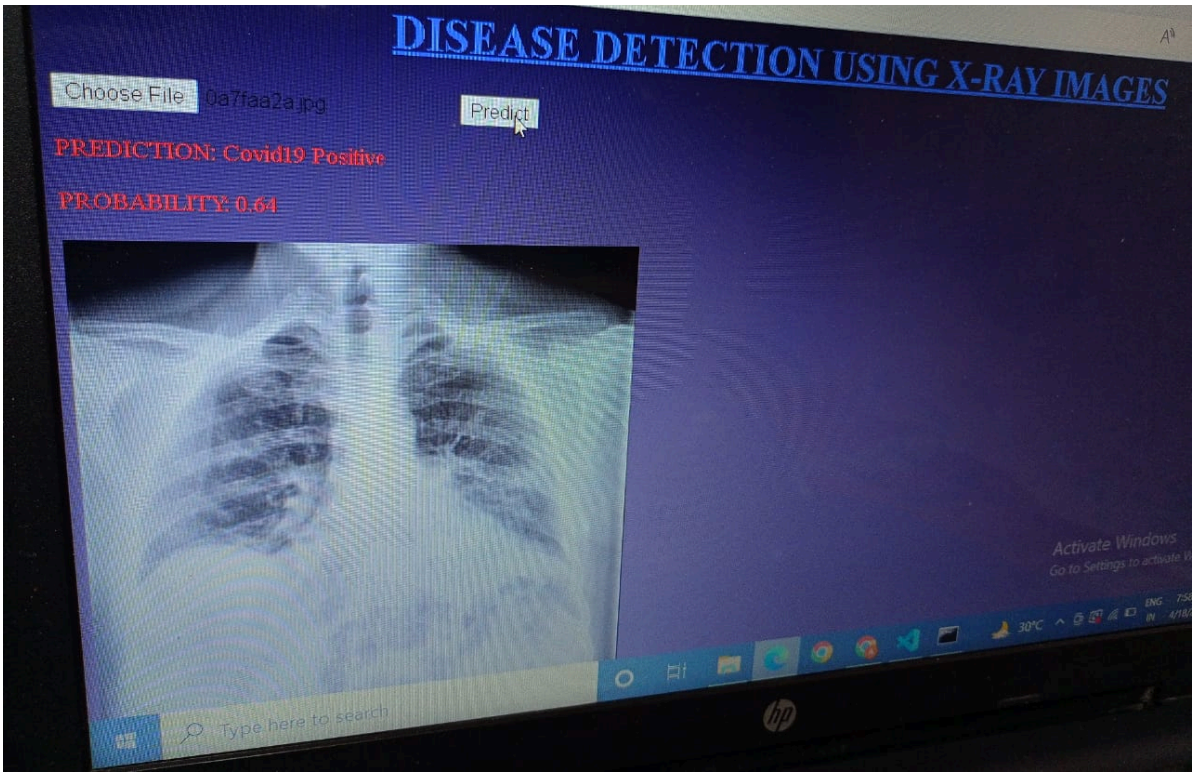
In [ ]:


X-RAY DETECTION
127.0.0.1:5000
DISEASE DETECTION USING X-RAY IMAGES
Choose File
No file chosen
Predict
PREDICTION: ...
PROBABILITY: ...
Activate Windows
Go to Settings to activate Windows.
Type here to search
33°C
ENG
US
12:36 PM
4/17/2022


Open
Dataset › test ›
Search test
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (E:)
Network
Name
Date modified
Type
Covid+
3/13/2022 8:29 PM
Normal
3/13/2022 8:29 PM
Pneumonia +
3/24/2022 8:29 PM
File name:
All files
Open
Cancel
NG X-RAY IMAGES
Activate Windows
Go to Settings to activate Windows.
Type here to search
33°C
ENG
US
12:39 PM
4/17/2022


X-RAY DETECTION
127.0.0.1:5000
DISEASE DETECTION USING X-RAY IMAGES
Choose File
16669_3_1.jpeg
Predict
PREDICTION: Covid19 Positive
PROBABILITY: 0.97
PRONE
PORT
Activate Windows
Go to Settings to activate Windows.
Type here to search
33°C
ENG
US
12:38 PM
4/17/2022


X-RAY DETECTION
127.0.0.1:5000
DISEASE DETECTION USING X-RAY IMAGES
Choose File
16672_1_1.jpg
Predict
PREDICTION: Covid19 Positive
PROBABILITY: 0.92
SEDUTO
Activate Windows
Go to Settings to activate Windows.
Type here to search
33°C
ENG
US
12:40 PM
4/17/2022



3.5 Conclusion and Future Work

The developed CNN model will provide good accuracy on less number of layers and less number of epochs in comparison to pre-trained models.

This CNN model will be better than other models as it is less complex than others and achieves good accuracy on training, validation as well as on test dataset.

The friendly front-end UI using the website will help user to easily diagnose the X-ray image and predict if the user is diagnosed with Covid-19 or Pneumonia.

Future Work:

In the system which we have been developed at present, there can be certain improvisation features which may be added on in it. Like using much bigger data as compared to now with increased number of kernels and much more in depth feature extraction of the data, this will definitely increase the accuracy of the model and reduce the validation loss in future. Hence the model which will be generated will be way better than now.

Also adding some more attributes will definitely classify the data with respect to different attributes, which will open up more information for all the users. So after using more attributes the model which will be developed is going to be much more informetic.

Using Grad Cam to properly visualize x-ray and differentiate features on part of diseases, this will make the x-ray report much more clear to user in terms of opacity by providing colorful output of the input image with flashing/focusing on the particular part of the x-ray images (that defines certain disease or problem of the x-ray).

Searching and using some other deep learning algorithms which will be more appropriate in terms of increasing the quality and accuracy of the model for deployment.

Connecting the system with user email and nearby hospitals for better health treatment and getting notified digitally, this will help hospitals and government to avoid the spreading of disease (as we have seen in COVID-19 pandemic) and taking precautionary steps for recovery of that patient.

Finally, increasing the quality of User Interface in terms of making it more user friendly.

References

1. https://www.sciencedirect.com/science/article/pii/S2666827021000694
2. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0256630
3. https://ieeexplore.ieee.org/document/9395851
4. https://link.springer.com/article/10.1007/s10044-021-00970-4
5. https://www.geeksforgeeks.org/introduction-convolution-neural-network/
6. https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
7. COVID-19: Origin, Detection and Impact Analysis Using Artificial Intel (routledge.com)
8. https://www.deeplearningbook.org/contents/convnets.html
9. Make Your Own Neural Network by Tariq Rashid (goodreads.com)
10. Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics | SpringerLink