DR. B.R. AMBEDKAR NATIONAL INSTITUTE OF TECHNOLOGY,
JALANDHAR - 144011

MINOR PROJECT REPORT

ON

PREDICTION OF COVID-19 FROM CHEST X-RAY AND COVID-19 TRACKER

For the award of the

Degree of Bachelor of Technology
In Computer Science and Engineering

ACADEMIC YEAR: 2020-2021

SUBMITTED BY:
Rajan Kataria (18103076)
Rippendeep Kaur (18103078)
Sanyamdeep Singh (18103083)

ACKNOWLEDGEMENT

It is rightly said that, "***We can't accomplish all that we need to do without working together***". The final result of our project required a lot of guidance and help from many people and we, are extremely fortunate to have got this throughout the journey of our project.

Whatever we are today is only due to such supervision and assistance and we would not forget to thank them from the core of our heart.

We are extremely thankful to Professor A.L. Sangal, Head, Department of Computer Science & Engineering who has provided all his direct and indirect support.

We are thankful to the In-charge, Minor Project Pre-Final Year, for providing us mentor and all other support.

We are extremely thankful to our mentors, who believed in our idea and suggested new ideas wherever required.

We are extremely thankful to Director sir, Professor Lalit Kumar Awasthi for constant encouragement and guidance at all times.

We are thankful to all the faculties of Department of Computer Science & Engineering (CSE) who gave us their time and suggestions and hence helped us in successfully completing our project work.

Thank you.

Group Members:
Rajan Kataria (18103076)
Rippendeep Kaur (18103078)
Sanyamdeep Singh (18103083)

DECLARATION

We, the members of this group, hereby declare that our minor Project titled- “PREDICTION OF COVID-19 FROM CHEST X-RAY AND COVID-19 TRACKER" being submitted by us in the Department of Computer Science and Engineering is a project work carried by us under the noble supervision of our mentors and the project has not been copied from anywhere and has been made solely by us.

We will be solely responsible if some Plagiarism is found.

Thank You All.

Date: 25th May, 2021

Group Members:
Rajan Kataria (18102076)
Rippendeep Kaur (18103078)
Sanyamdeep Singh (18103083)

TABLE OF CONTENTS

1. INTRODUCTION- BUILDING UP THE FOUNDATION

The main aim of this project is to provide an online tool that enables people to predict the chances of being infected with COVID-19 infection. As the name specifies, it is the software to predict the disease from the image of the chest x-ray of a person using a Machine Learning based model. Not only this, it also keeps track of the deadly pandemic COVID-19's scenario in the various parts of the world.

1.1 Problem Statement and Its Necessity

No one is unaware of the current scenario of our country. India is going through a tough phase where the infection has badly affected the country. The second layer of COVID-19 has hit the country and now the health system of the country is on the verge of being collapsed. There is lack of beds, lack of oxygen cylinders in the hospital. The number of positive cases per day have crossed the barrier of 2 lakhs and there are many such that are left undetected.
Below are some reasons that lead us to come up with the idea of COVID-19 prediction and tracking software:

1.1.1 Lack of Accessibility to Sample Collection/Testing Centers:

There are many people who are unable to get themselves tested in the hospitals or other COVID-19 sample collection/testing centers. There can be several reasons behind this. One could be the limited number of sample collection centers. Not everyone has access to these centers and very few among these centers provide test facility free of cost. Many of the COVID-19 testing centers charge for testing the sample and not everyone can afford it. The free sample collection and detection in the government hospitals itself is bound to some constraints. There is a limit on the number of persons that can be present at the testing center at a given time due to social distancing and other preventive measures. Another bound is on the number of samples being collected.

1.1.2 Delay in the Test Report:

The RT-PCR test is considered as the most reliable test for the detection of the COVID-19 infection. But this test itself has some drawbacks. The report comes after 72 hours of sample collection and within these 72 hours the person is unsure whether he/she is infected or not. Though it is recommended for the person getting tested to isolate himself/ herself till the negative report comes. But; if someone is asymptomatic then he/she may meet other people thinking that the report will be negative and in such cases there are chances of the spread of the viral infection to other people. Also; the condition of patient may worsen in these 72 hours. The chances of recovery are highest in the early stage.

1.1.3 Unavailability of Doctor:

A patient can get himself detected in case he/she is unable to get an appointment with the doctor as per the current scenario of the country. In such a case he/she can perform a self-test using the image of his/her chest x-ray. This software is helpful in those cases when the person is unable to read his/her x-ray so can perform the detection test and share the report with the doctor so that the doctor could assure further and suggest the preventive measures and remedies to be taken.

1.1.4 Lack of Information/Statistical Data:

Not all the persons watch/listen to news daily as it can have a negative impact on their mental health in this tough time. So, not everyone is aware of the statistical data related to COVID-19 all around the world. The Covid-19 Tracker provides the figures of confirmed cases, recovered cases and the deaths that have happened in various countries of the world due to this pandemic.

This wraps up the Problem statement, need and aims and objectives of this project. The aim is to an economically feasible platform that could contribute to the Health System of the country and cater to the problems faced by the people and the Administration in getting themselves tested for COVID-19.

1.2 Feasibility - Technical and Non-Technical

The feasibility study of the project is the most crucial task before beginning to work on any project.
The Various Kinds of Feasibilities of this project can be summed up as follows:

1.2.1 TECHNICAL FEASIBILITY:

- Web Application Supporting Devices as mobiles, tablets, desktops, laptops.
- Internet connectivity is required for the system.
- Chest X-ray Image is required in jpg/png format.

1.2.2 NON-TECHNICAL FEASIBILITY:

SOCIAL: -

- No such software exists in India.
- It will make the COVID-19 detection easier and feasible as well as it would help in reducing the spread of virus as there as chances of spread even in the COVID-19 test centers. At the same time, it is providing the figures related to COVID-19 all over the world.
- It will support the digitization movement started by Honorable Prime Minister of India.

ECONOMICAL FEASIBILITY: -

- This project doesn't require any cost in development.
- This software is available freely to be used by any person making COVID-19 detection economically feasible for all.

SCOPE: -

- The COVID-19 detector and tracker is aimed in making COVID-19 early detection a simpler and easier process so that patients can isolate themselves and start the medications at the earliest.
- It is needed in every city, every state, everywhere. So, the scope of this project is worldwide.

2. PROPOSED SOLUTION: ICE BREAKING

"Knowing the obstacles makes you great, but making the way through them makes you the greatest."

2.1 Identifying Stakeholders

The stakeholders in our project are the users of the application. The users can be particular persons or any particular organization such as hospitals and other institutes using the application for the COVID-19 detection from chest x-ray of the patients/staff.

2.2 Detailed Solution

Below is presented our solution in a detailed manner, presenting the various functionalities offered to our stakeholders:

2.2.1 Each Individual Having his own Space - A Centralized Application:

The individuals who are using this application are required to first register themselves. Then only they would be able to login. While login, a person is required to provide the username and password.

2.2.2 Self-Assessment in One's Own hands:

The main functionality of this app is to make the users able to detect COVID-19 from the chest x-ray at home before the RT-PCR test so that he/ she can consult doctor at the earliest in case a high probability of being tested positive is shown by the ML model. This is very beneficial for a common person who does not know how to read chest x-ray. Getting a doctor's appointment these days is itself a challenging task.

2.2.3 World COVID stats, a click away:

The main aim of the tracker id to provide the country-wise COVID-19 data till present date to the users so that they are aware of the worldwide situation of COVID-19 and take preventive measures to stay safe and healthy at home and practice social distancing.

3. TECHNICAL ANALYSIS

3.1 Activity Diagram

- Activity diagram is an important behavioral diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.
- Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.
- The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

The basic purpose of activity diagrams is to capture the dynamic behaviour of the system. The activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques.

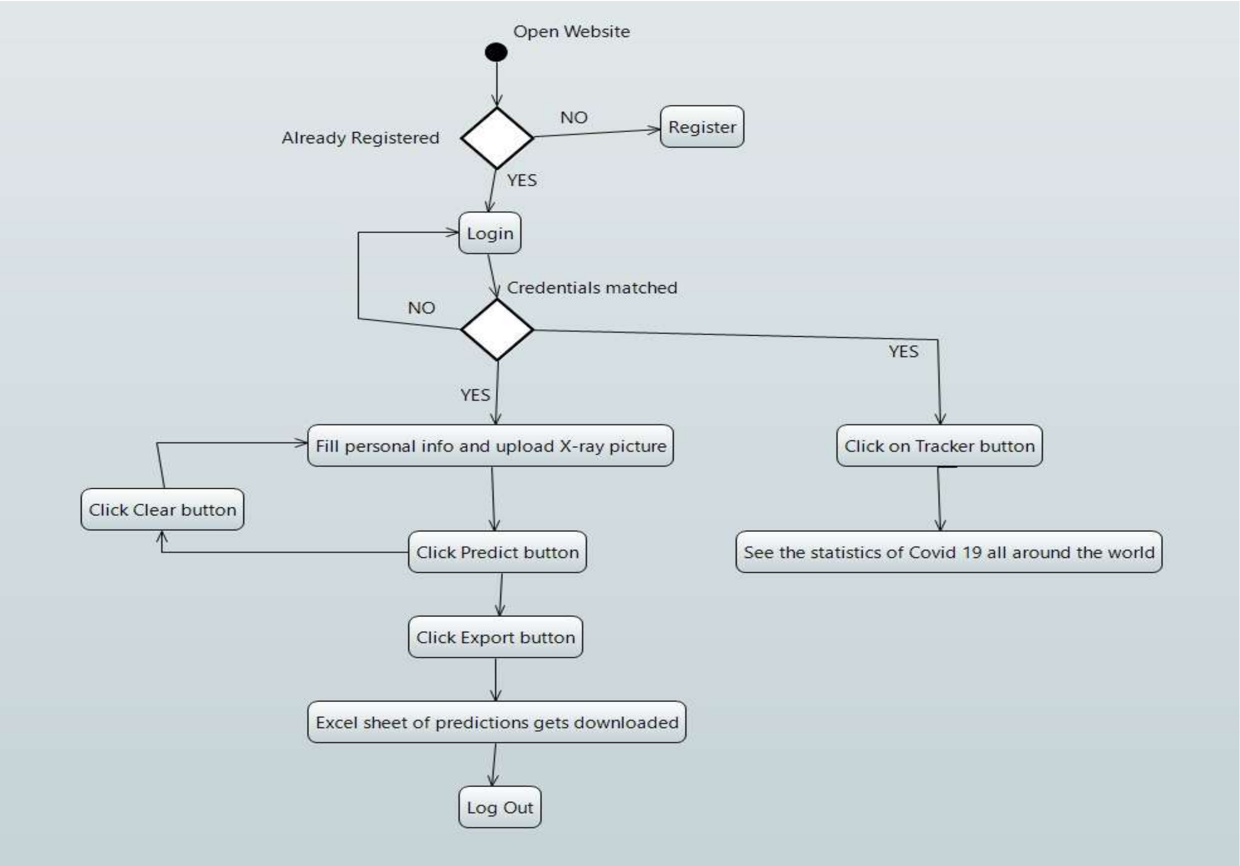


Fig: Activity Diagram

As Activity Diagram is graphical representation of workflow and in every next step there are choice, iteration and concurrency. After opening the website, the user can either be already registered, in this case he can enter his/her credentials and proceed to login. If the user is not already registered, the he/she has to first register and then only he/she can login. After logging in, the user will be taken to the main page where he/she will find options to either fill personal information (name and Aadhaar number), upload the chest x-ray image and predict whether there are chances of him/her of getting tested positive or negative. After clicking the predict button the user can see the result provided by the trained ML model. The user can further click on the export button to download the excel sheet of predictions. The user can clear the data entered by clicking on the clear button and provide new information for another predictions as many times as he wants and then export the list of all the predicted cases. The second option available to the user on the main page is to click on the tracker button and land on the COVID-19 tracker page showing the country wise data of total confirmed cases, recovered cases and deaths due to COVID-19.

3.2 Tech-stack Analysis

In order to achieve various solutions, we have used a variety of Tech Stacks. All these technologies have been chosen on the basis of the following few criteria:
1. Ease of Usage and Ease of Learning
2. Time Available to build
3. Efficiency
4. Security

On the basis of the above-mentioned criteria the following Technologies have been used:

3.2.1 phpMyAdmin

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc.) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.
USAGE: We have used phpMyAdmin for our Real-time Database and for the Storage of user data (login credentials/registered account details, prediction data etc.).

3.2.2 Machine Learning

Machine Learning is the heart of this project. The ML model is used to detect the chances of a person being tested positive and negative for COVID-19. The TensorFlow model uses a CNN. The model gets trained upon the given set of images of the chest x-ray (Of covid positive and negative patients). We have used Keras API to implement the neural network architecture and the Tensorflow is the actual backend that is used. 7 layered Convolutional Neural Network is trained on the dataset (90-10 train-test split) which gave an accuracy of 97% on the training data and 99% on the validation data.
The modules and libraries used in Python are given as follows:

- **keras**
  Keras is a neural networks library written in Python that is high-level in nature – which makes it extremely simple and intuitive to use. It works as a wrapper to low-

level libraries like TensorFlow or Theano high-level neural networks library, written in Python that works as a wrapper to TensorFlow or Theano.

- **OpenCV (CV2)**
  OpenCV-Python is a library of Python bindings designed to solve computer vision problems.OpenCV-Python makes use of Numpy, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.
- **OS**
  The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The os and os. path modules include many functions to interact with the file system.
- **numpy**
  NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy stands for Numerical Python.
- **pandas**
  pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

3.2.3 Flask

Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

USAGE: The ML model is integrated in the web application using the python flask.

3.2.4 Web Designing - HTML, CSS, JS

Our project is a web application, so all these technologies are the core of our project. This web application basically has three pages- the Login page, the COVID-19 Detection page and the Tracker. All the three pages are designed using HTML, CSS and JS. In the Tracker, the geoPlugin API is used which on the basis of IP address of the user returns the country code of the user and then using the various country codes the COVID-19 data for various countries is fetched using Fetch API. The Fetch API provides a JavaScript interface for accessing and manipulating parts of the HTTP pipeline, such as requests and responses. It also provides a global fetch()

method that provides an easy, logical way to fetch resources asynchronously across the network.

4. WORKING AND IMPLEMENTATION

4.1 Image Classification CNN Model

4.1.1 Introduction

Image Processing:

Image processing aims to transform an image into digital form and performs some process on it, to get an enhanced image or take some utilized information from it. It is a method that develops to convert the image into digital form and perform some operations to obtain specific models or to extract useful information from it.

It is used in many domains. Some of them are listed as follows:

- Image sharpening and restoration
- Pattern recognition.
- In developing games
- Human position tracking
- In Medical field (X-RAY, MRI, CT-SCAN)

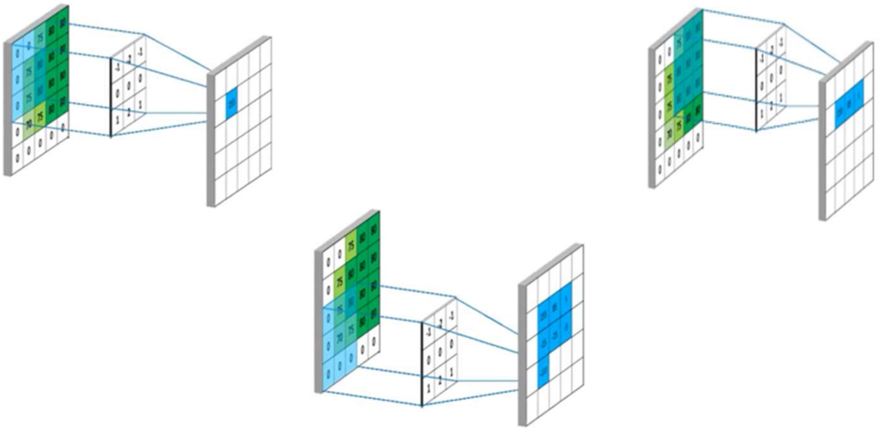


Fig: Image Processing

Convolutional Neural Networks

Convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data. CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vison that includes image and video recognition, along with recommender systems and natural language processing (NLP).

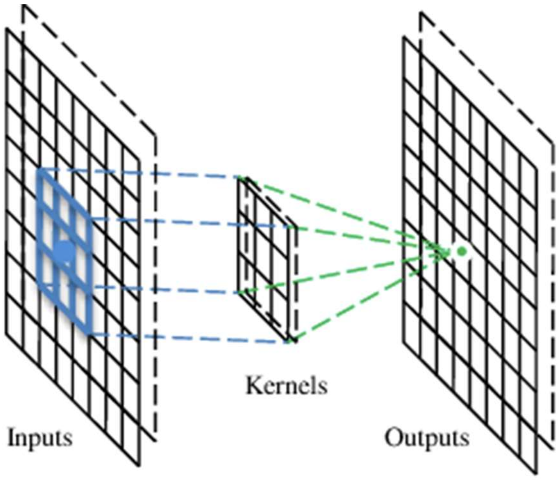


Fig: Convolutional Neural Network

4.1.2 Creating the Customized dataset

We have used two parent datasets from which we have extracted our main dataset comprising of Covid-19 Positive and Covid-19 Negative images in the separate folders.

The links to those two parent datasets are given as follows:

- CoVID chest X-ray dataset (GitHub): https://github.com/ieee8023/covid-chestxray-dataset
- Pnuemonia dataset by Praveen (Kaggle): https://www.kaggle.com/praveengovi/coronahack-chest-xraydataset

The dataset used in this model has been created by combining the images from the above mentioned datasets. The model will classify the images into 2 categories, i.e. COVID 19 Positive and COVID 19 Negative and will also tell the probability of each.

So, to train the model we have to use two types of chest X-RAY images; those which tested positive and the normal chest.

The collection of COVID 19 positive chest X-RAY images has been taken from the first mentioned dataset "COVID chest X-RAY dataset". This dataset is a collection of images that have been tested positive for various diseases (COVID, Pneumonia, SARS, etc.). The "finding" attribute of the dataset specifies the disease. From these we have chosen the images having ‘COVID’ in “the finding” attribute.

The normal chest X-RAY images have been taken from the second mentioned dataset which is a collection of Pneumonic and normal chest X-RAY images and from these images we have chosen those having label 'normal'.

The final dataset created is a collection of 1576 COVID 19 Negative images and 579 COVID 19 Positive images.

4.1.3 Preprocessing of the dataset

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format.

The following steps have been performed during the data preprocessing:

- **Conversion of images from BGR to grayscale format:**

  Color increases the complexity of the model. One may want to introduce an image processing tool using gray level images, as opposed to color, not because of the "format" of gray level images, but because the inherent complexity of gray level images is lower than that of color images. E.g., one can talk about brightness, contrast, edges, shape, contours, texture, perspective, shadows, and so on, without addressing color. After presenting a gray-level image model/method, in most cases it can be afterwards be extended to color images.

  Because it is a one-layer image from 0-255 whereas the RGB/BGR have three different layer image. So that is a reason we prefer grey scale image instead of RGB/BGR.

- **Resizing the images:**

  All the images were resized to 100x100 so that all the images can have a common size. Image resizing is necessary when you need to increase or decrease the total number of pixels, whereas remapping can occur when you are correcting for lens distortion or rotating an image.

- **Normalization:**

  In image processing, normalization is a process that changes the range of pixel intensity values. We have normalized the value of each pixel by dividing by 255 so that the pixel intensity lies between 0-1.

  The preprocessed dataset is shown as follows:

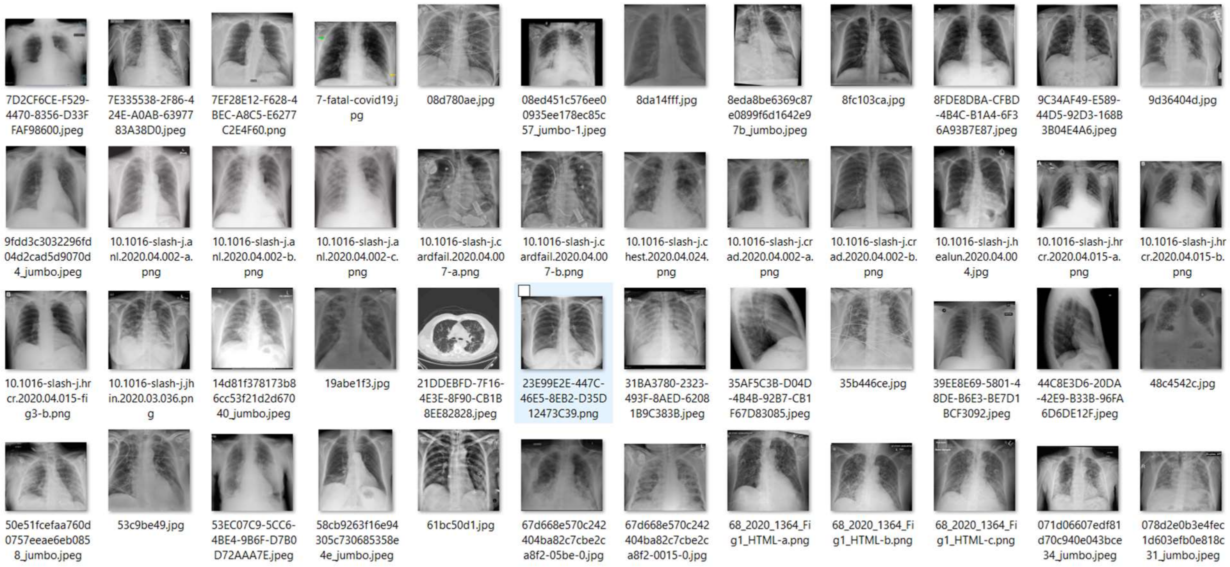

7D2CF6CE-F529-4470-8356-D33FFAF98600.jpeg
7E335538-2F86-424E-A0AB-6397783A38D0.jpeg
7EF28E12-F628-4BEC-A8C5-E6277C2E4F60.png
7-fatal-covid19.jpg
08d780ae.jpg
08ed451c576ee00935ee178ec85c57_jumbo-1.jpeg
8da14fff.jpg
8eda8be6369c87e0899f6d1642e97b_jumbo.jpeg
8fc103ca.jpg
8FDE8DBA-CFBD-4B4C-B1A4-6F36A93B7E87.jpeg
9C34AF49-E589-44D5-92D3-168B3B04E4A6.jpeg
9d36404d.jpg
9fdd3c3032296fd04d2cad5d9070d4_jumbo.jpeg
10.1016-slash-j.anl.2020.04.002-a.png
10.1016-slash-j.anl.2020.04.002-b.png
10.1016-slash-j.anl.2020.04.002-c.png
10.1016-slash-j.cardfail.2020.04.007-a.png
10.1016-slash-j.cardfail.2020.04.007-b.png
10.1016-slash-j.chest.2020.04.024.png
10.1016-slash-j.crad.2020.04.002-a.png
10.1016-slash-j.crad.2020.04.002-b.png
10.1016-slash-j.healun.2020.04.004.jpg
10.1016-slash-j.hrcr.2020.04.015-a.png
10.1016-slash-j.hrcr.2020.04.015-b.png
10.1016-slash-j.hrcr.2020.04.015-fig3-b.png
10.1016-slash-j.jhin.2020.03.036.png
14d81f378173b86cc53f21d2d67040_jumbo.jpeg
19abe1f3.jpg
21DDEBFD-7F16-4E3E-8F90-CB1B8EE82828.jpeg
23E99E2E-447C-46E5-8EB2-D35D12473C39.png
31BA3780-2323-493F-8AED-62081B9C383B.jpeg
35AF5C3B-D04D-4B4B-92B7-CB1F67D83085.jpeg
35b446ce.jpg
39EE8E69-5801-48DE-B6E3-BE7D1BCF3092.jpeg
44C8E3D6-20DA-42E9-B33B-96FA6D6DE12F.jpeg
48c4542c.jpg
50e51fcefaa760d0757eeae6eb0858_jumbo.jpeg
53c9be49.jpg
53EC07C9-5CC6-4BE4-9B6F-D7B0D72AAA7E.jpeg
58cb9263f16e94305c730685358e4e_jumbo.jpeg
61bc50d1.jpg
67d668e570c242404ba82c7cbe2ca8f2-05be-0.jpg
67d668e570c242404ba82c7cbe2ca8f2-0015-0.jpg
68_2020_1364_Fig1_HTML-a.png
68_2020_1364_Fig1_HTML-b.png
68_2020_1364_Fig1_HTML-c.png
071d06607edf81d70c940e043bce34_jumbo.jpeg
078d2e0b3e4fec1d603efb0e818c31_jumbo.jpeg

Fig: Covid 19 Positive Images

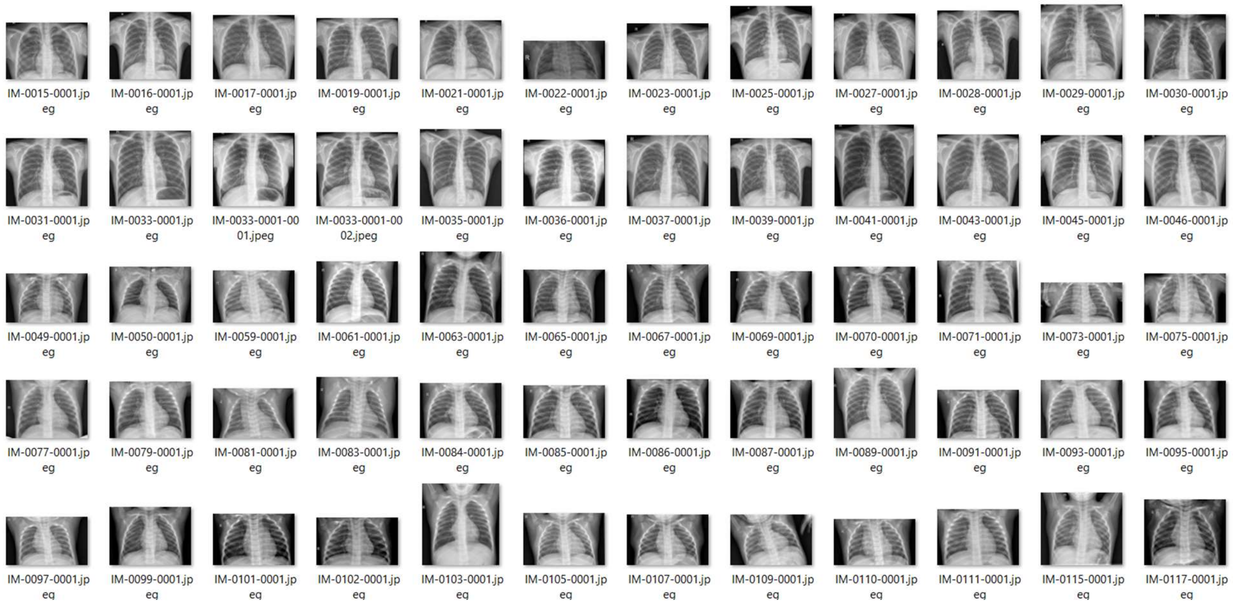


Fig: Covid 19 Negative Images

4.1.4 Training the Convolutional Neural Network

The 8-layer CNN model has been used to classify the images. Following is the summary of different layers of the model:

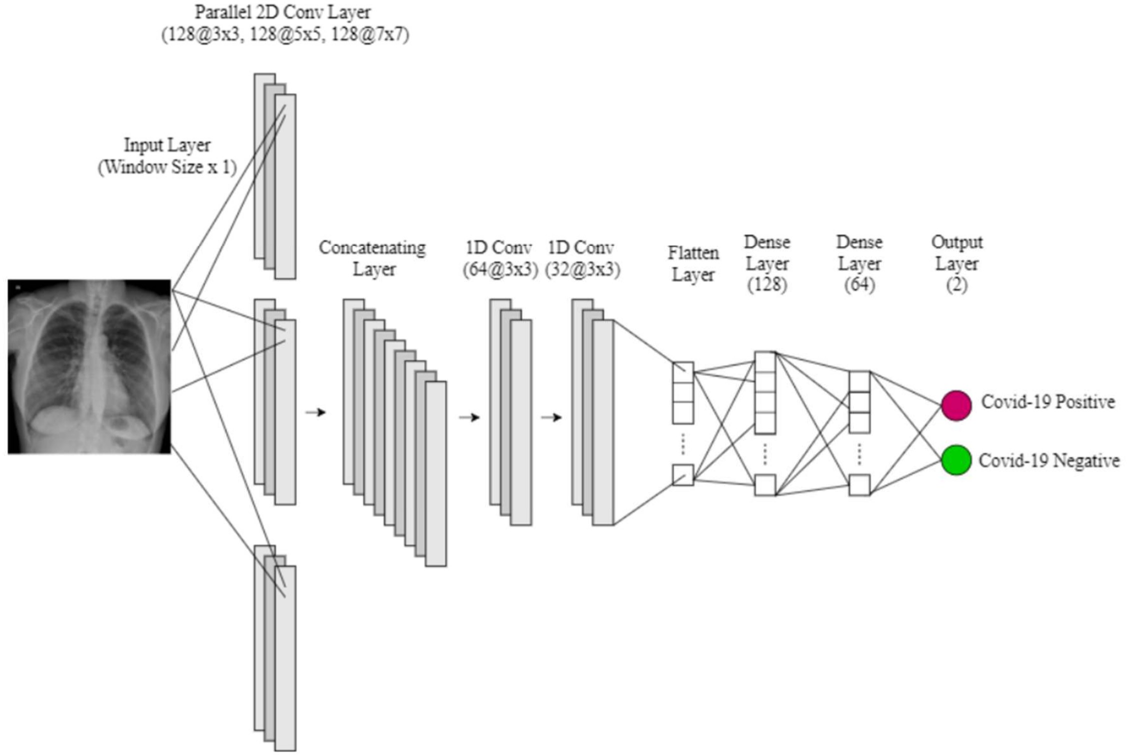
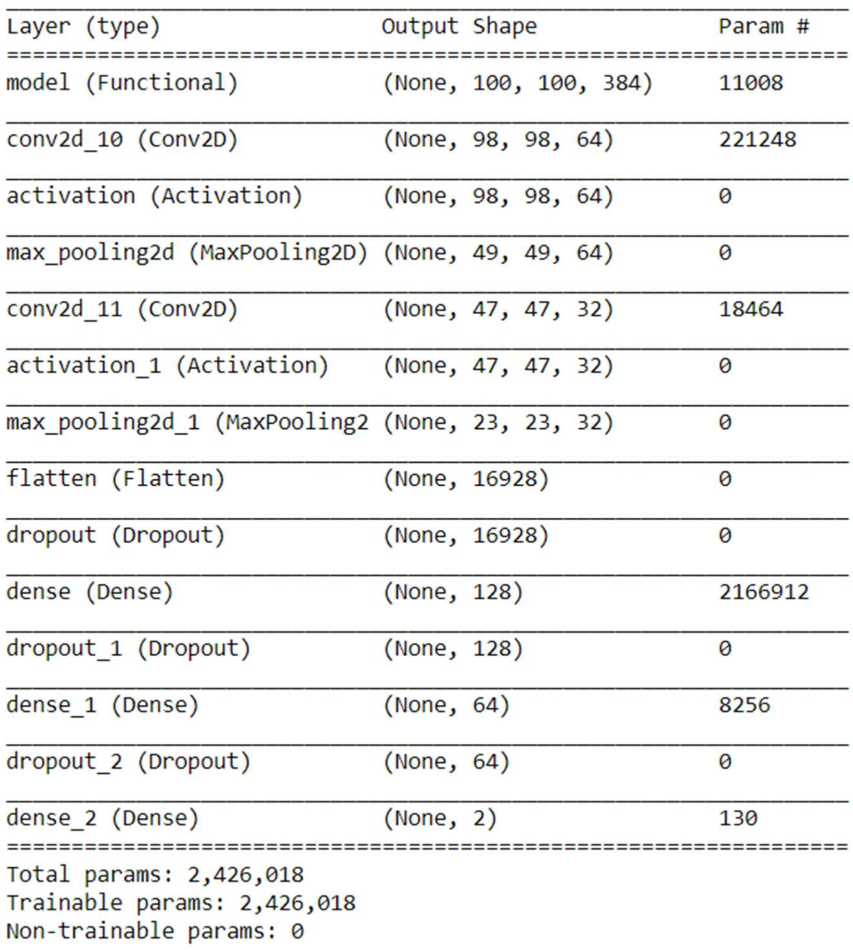


Fig: Layers in Convolutional Neural Network

- **Parallel 2D Convolutional Layers:**
  Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. This is where most of the user-specified parameters are in the network. The most important parameters are the number of kernels and the size of the kernels.
  The first layer consists of 3 parallel 2D Convolutional Layers having 128 kernels each and kernel size of 3,5,7 respectively. This means we have 3 parallel layers with size 128@3x3 ,128@5x5 and128@7x7 respectively.
- **Concatenating layer:**
  This is the layer that concatenates a list of inputs. It takes as input a list of tensors, all of the same shape except for the concatenation axis, and returns a single tensor that is the concatenation of all inputs. This layer concatenates the outputs coming from 3 parallel Convolutional layers.
- **General Convolutional layer1:**
  The third layer is a 2D convolutional layer having 64 kernels of size(3x3).
- **General Convolutional layer2:**
  The fourth layer is again a 2D convolutional layer with 32 kernels of size 3x3.
- **Flatten layer:**
  Between the convolutional layer and the fully connected layer, there is a 'Flatten' layer. Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier.
- **Dense layer1:**
  The next layer is a dense layer having 128 neurons. The dense layer is the regular deeply (fully) connected neural network layer. It is most common and frequently used layer. Dense layer does the below operation on the input and return the output.
  output = activation (dot (input, kernel) + bias).
- **Dense layer2:**
  The seventh layer is a dense layer having 64 neurons.
- **Output layer:**
  The final layer is the output layer having two nuerons that classify images either as COVID +ve or COVID -ve.

model.summary() function can be used to view the Neural Network architecture. Dropout layers have been added to minimize the overfitting of the model (SS). Activation criterion used for output layer is 'softmax'.

The output of model.summary() function is given as follows:



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
model (Functional)           (None, 100, 100, 384)     11008
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 98, 98, 64)        221248
_________________________________________________________________
activation (Activation)      (None, 98, 98, 64)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 49, 49, 64)        0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 47, 47, 32)        18464
_________________________________________________________________
activation_1 (Activation)    (None, 47, 47, 32)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 32)        0
_________________________________________________________________
flatten (Flatten)            (None, 16928)             0
_________________________________________________________________
dropout (Dropout)            (None, 16928)             0
_________________________________________________________________
dense (Dense)                (None, 128)               2166912
_________________________________________________________________
dropout_1 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 130
=================================================================
Total params: 2,426,018
Trainable params: 2,426,018
Non-trainable params: 0
```

Fig: Summary of CNN Model

We have used 'Categorical Crossentropy ' as the loss function and 'adam' optimizer. Dataset is then split into training and testing (Split ratio - 90:10). In training the model, 20 epoches have been used. Generally, more epoches means more accuracy but more epoches consume time as the computations involved are complex.

Model checkpoints have been created to save the model whenever a model has less validation loss (val_loss) than the previous model and hence more accuracy than the previous model. Not every model after an epoch is saved. Only the better models will be saved by monitoring the validation loss (or validation accuracy).

After each epoch, the loss, accuracy, validation loss and validation accuracy can be viewed.

4.1.5 Results and Accuracy

The Plots have been drawn for training loss/validation loss and training accuracy/validation accuracy after each epoche.

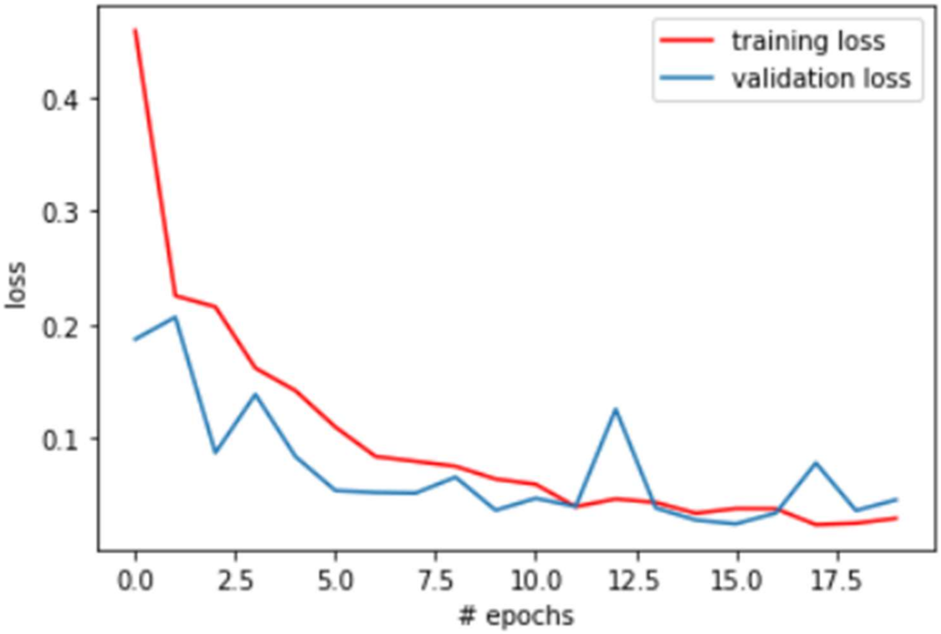


Fig: line Plot of Training Loss and Validation Loss v/s Epoches

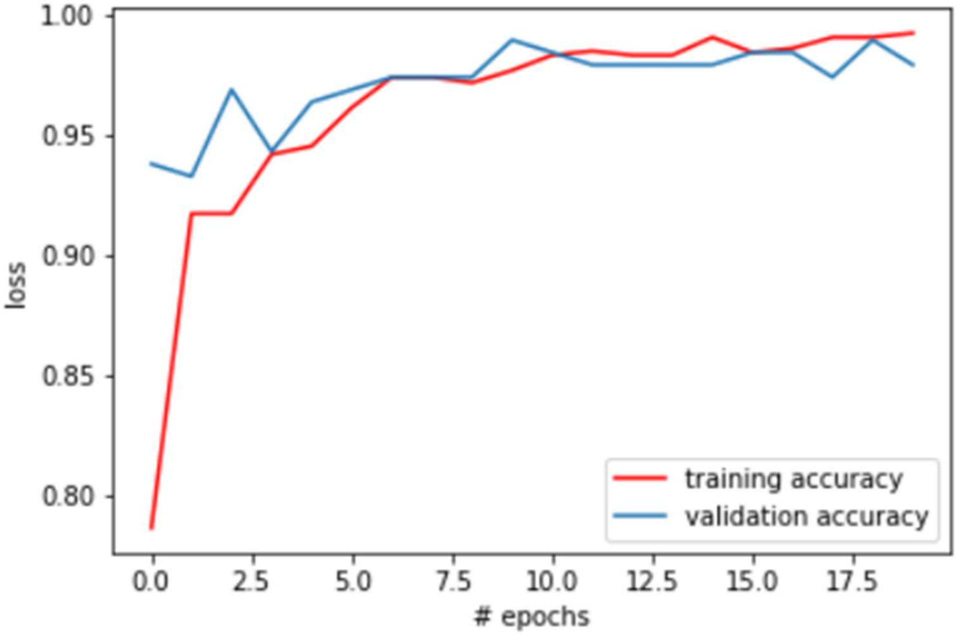
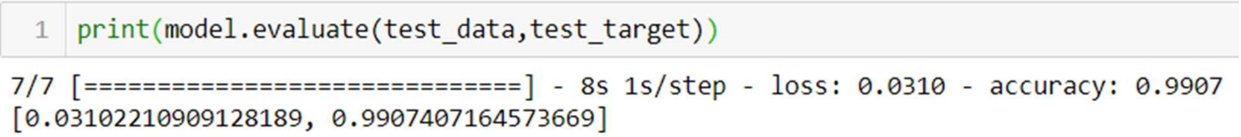


Fig: Line plot of Training Accuracy and Validation Accuracy v/s Epoches

The accuracy achieved by model after 20 epoches on testing set is 99.07% and the loss is 3.10%.



```
print(model.evaluate(test_data,test_target))
```

```
7/7 [==============================] - 8s 1s/step - loss: 0.0310 - accuracy: 0.9907
[0.03102210909128189, 0.9907407164573669]
```

Fig: Accuracy of the CNN Model

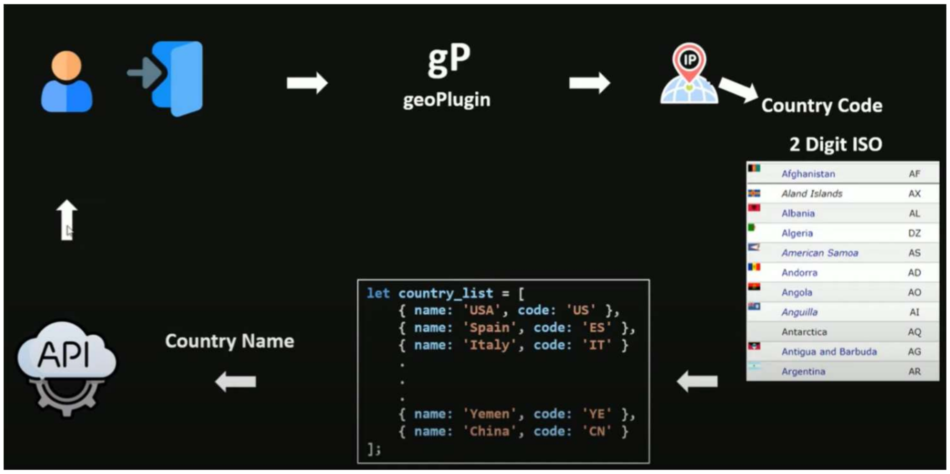
4.2 COVID-19 Statistics Tracker

When a user gets directed to the tracker page it sees the COVID -19 statistics. The user can see the stats of any country in the world by searching in the search box. It can see total cases, recovered cases and total deaths. By default, they will see the stats of their current location i.e. their country as it uses their IP address. It's an responsive layout where user sees the stats in form of charts.

We have used a plugin called geoPlugin which is a JavaScript geolocation web service (an API). It returns the country code (2 digit ISO) of the user by tracking its IP address. It provides us with many functions out of which we have used geoplugin_countryCode() to get the country code of the user. Then we have used an array country_list that returns the country of the user based on the country code passed.

Then we have used fetch API that takes an URL as an argument in the form of .json that returns a promise. The response that we get from API is a HTTP response from which we extract the JSON body content. After extracting it, the data we get is the JavaScript object or an array. Now we can do anything we want from it.

Then the country name is sent to the API to extract all the stats of the Covid-19 of that particular country and send back to the user. The API used is https://api.covid19api.com.



Afghanistan	AF
Aland Islands	AX
Albania	AL
Algeria	DZ
American Samoa	AS
Andorra	AD
Angola	AO
Anguilla	AI
Antarctica	AQ
Antigua and Barbuda	AG
Argentina	AR

```
let country_list = [
    { name: 'USA', code: 'US' },
    { name: 'Spain', code: 'ES' },
    { name: 'Italy', code: 'IT' }
    .
    .
    .
    { name: 'Yemen', code: 'YE' },
    { name: 'China', code: 'CN' }
];
```

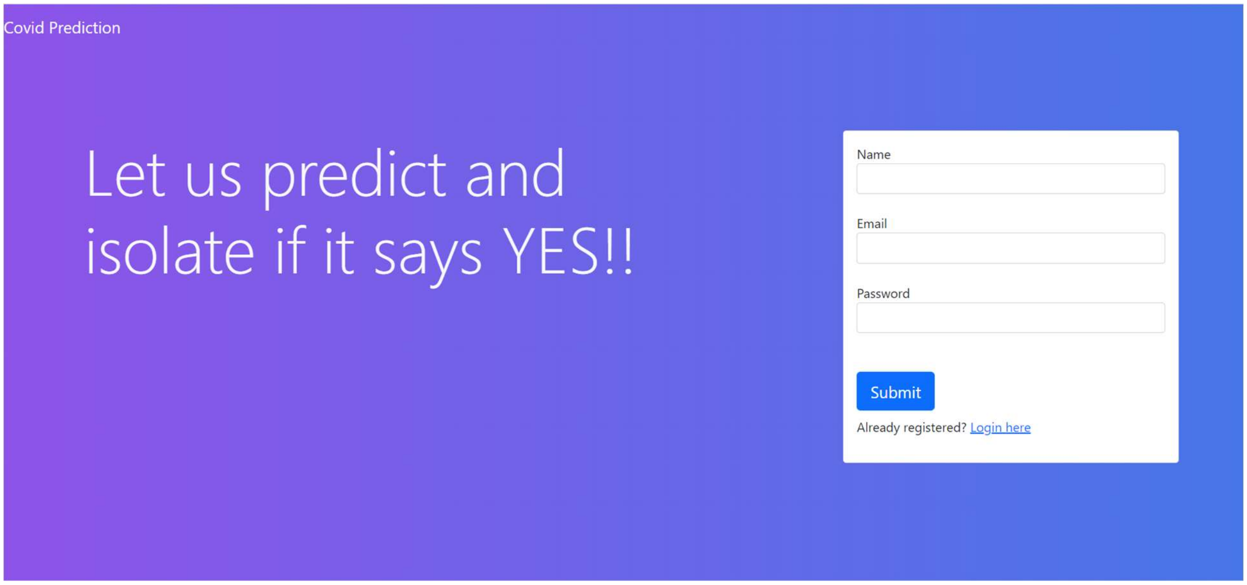
5. RESULT AND DISCUSSION

5.1 App Usage Instructions

Let us take the opportunity to provide the details about the basic functionalities of this app in this section.

5.1.1 Login/Register Page:

On opening the app, the users will first see the login page which will ask them to login by entering their username and password if already registered and if not already registered, then the users will first have to register themselves by providing basic details and creating a password for their account.



Covid Prediction

Let us predict and isolate if it says YES!!

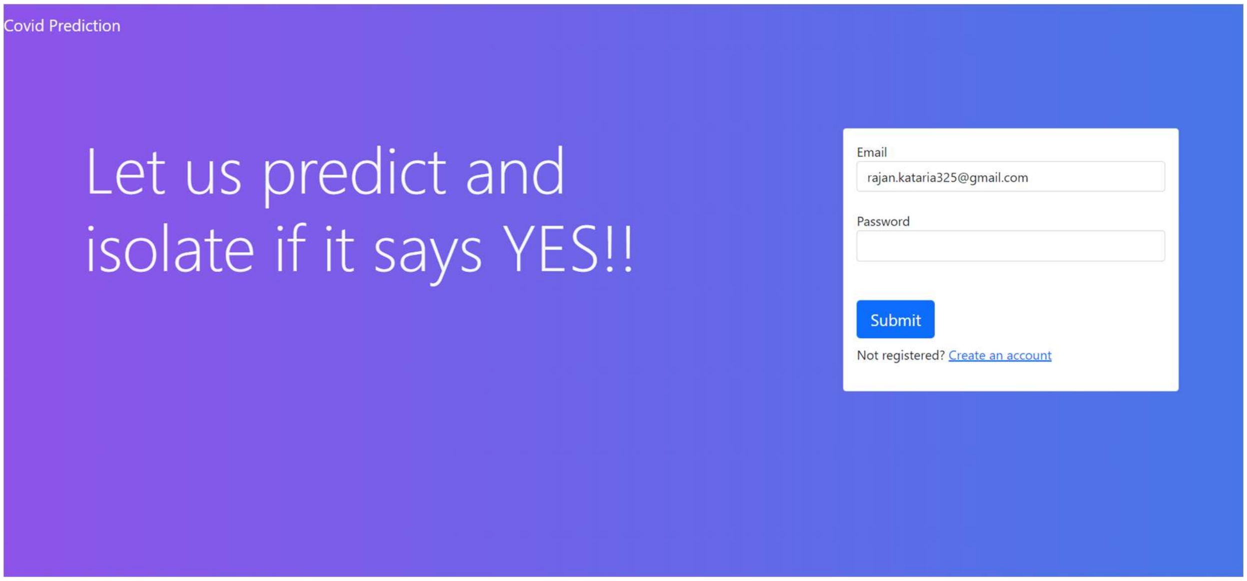
Name

Email

Password

Already registered? Login here

Fig: Register Page



Covid Prediction

Let us predict and isolate if it says YES!!

Email: rajan.kataria325@gmail.com

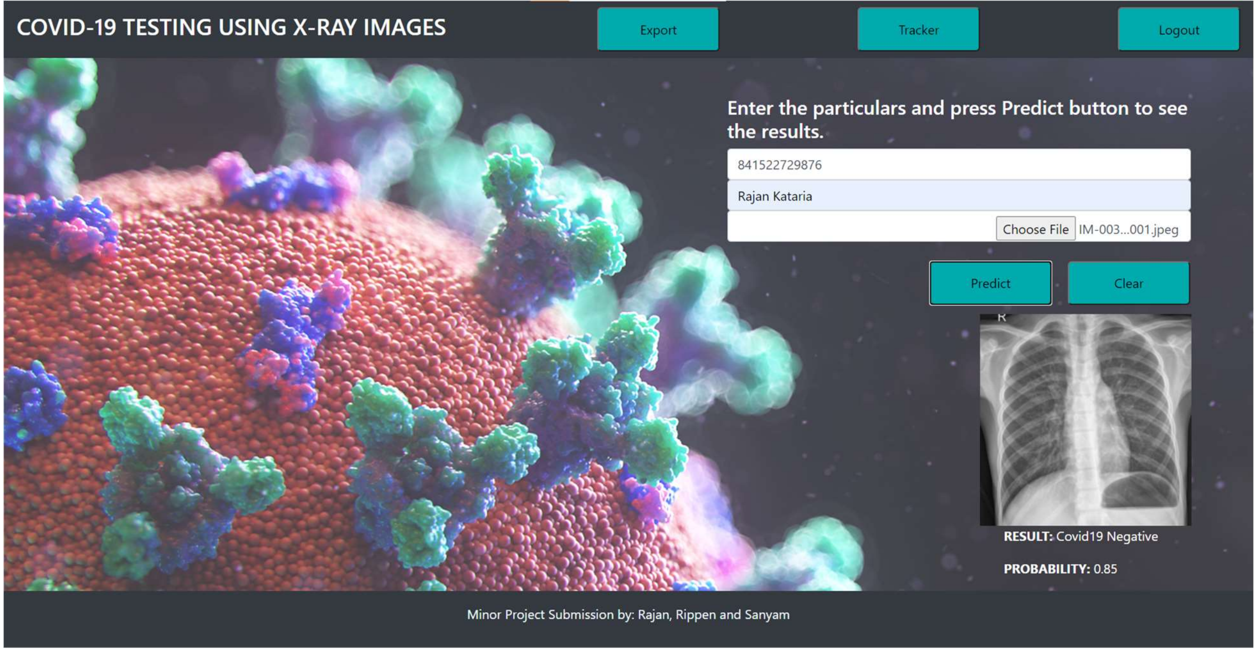
Password

Not registered? Create an account

Fig: Login page

5.1.2 COVID-19 Detection Page:

After successful login, the users will land on another page where they can provide the image of chest x-ray and some basic details (name of the patient and Aadhar number) and check the probability of being tested positive or negative for COVID-19. The users can check the predictions for as many chest X-RAYs as they want. Another functionality on this page is the **Export button.** On clicking this button, the users will be able to download the excel sheet containing the results of predictions made after the login. By clicking the **Tracker button** on third page, the users will be taken to the next page. Users can logout of the app by clicking on **Logout** button on this page.



Enter the particulars and press Predict button to see the results.

841522729876

Rajan Kataria

IM-003...001.jpeg

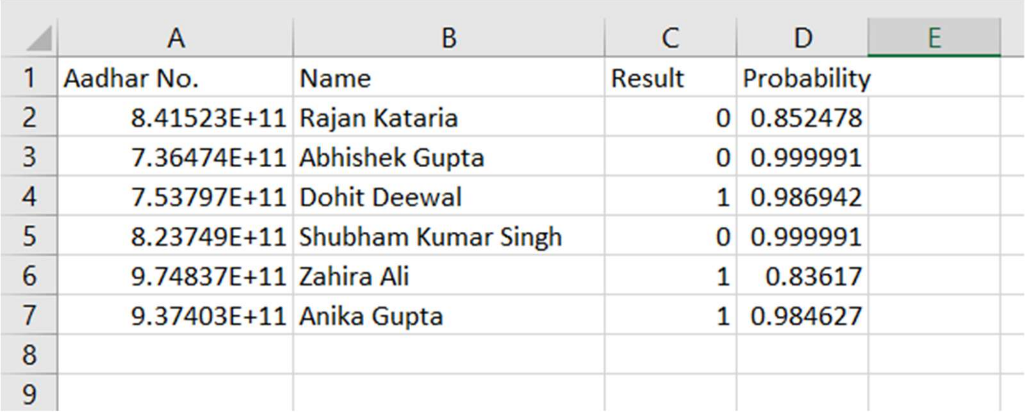
RESULT: Covid19 Negative

PROBABILITY: 0.85

Minor Project Submission by: Rajan, Rippen and Sanyam

Fig: Dashboard page

Spreadsheet created after pressing Export button can be viewed as predictions.csv.

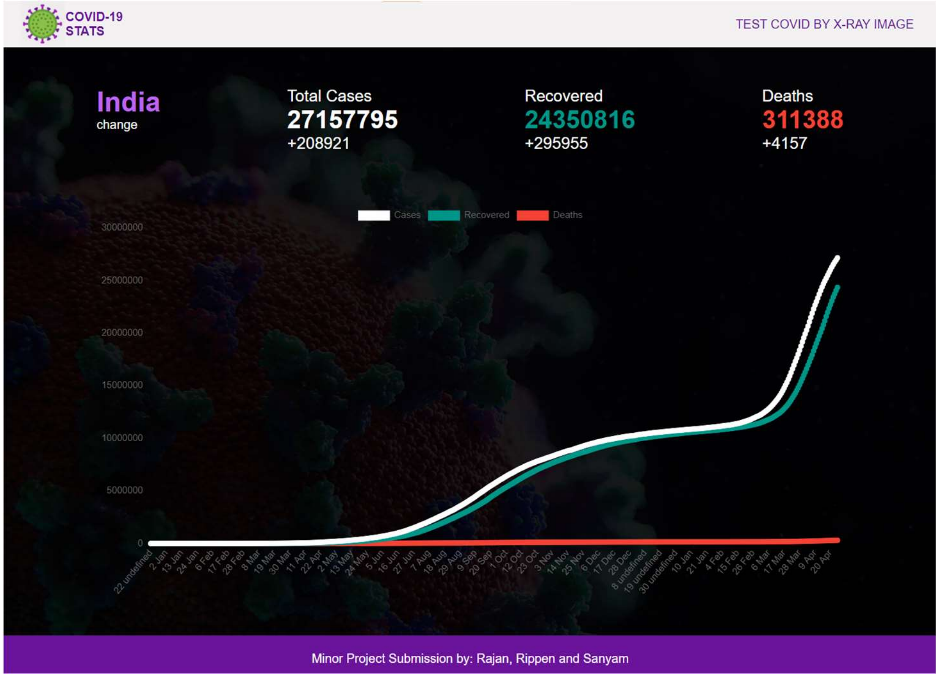


	A	B	C	D	E
1	Aadhar No.	Name	Result	Probability	
2	8.41523E+11	Rajan Kataria	0	0.852478	
3	7.36474E+11	Abhishek Gupta	0	0.999991	
4	7.53797E+11	Dohit Deewal	1	0.986942	
5	8.23749E+11	Shubham Kumar Singh	0	0.999991	
6	9.74837E+11	Zahira Ali	1	0.83617	
7	9.37403E+11	Anika Gupta	1	0.984627	
8					
9					

Fig: Screenshot of Excel Sheet created

5.1.3 COVID-19 Tracker Page:

In this page, the users will be able to see the COVID-19 data for various countries of the world. They have to select a particular country from the list and then they can view the current COVID statistics (total confirmed cases, recovered cases and total deaths till date) of that particular country. A graphical representation of this data is also provided.



India
change

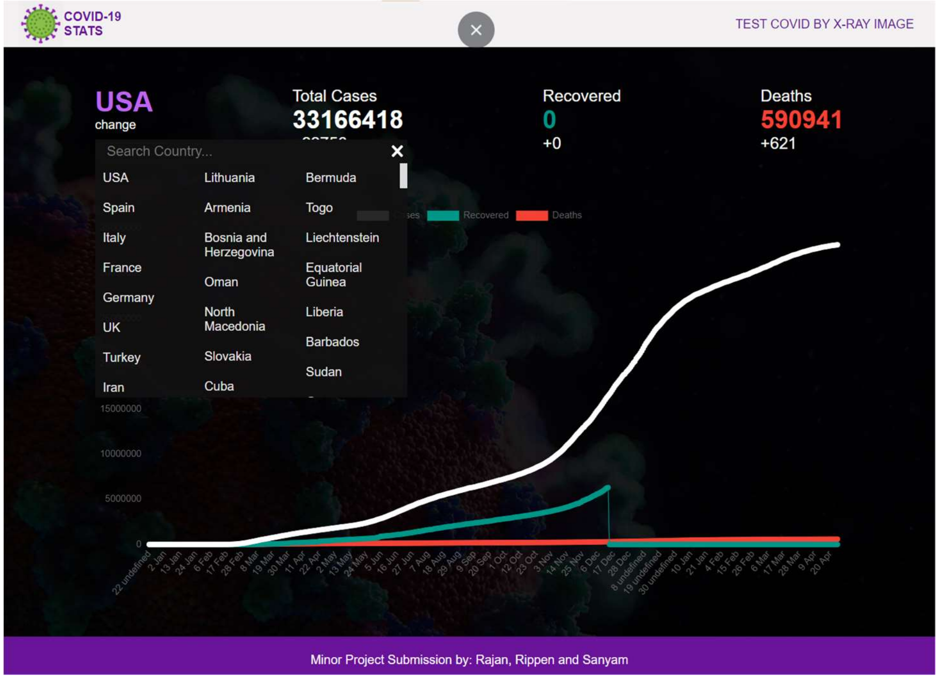
Total Cases
27157795
+208921

Recovered
24350816
+295955

Deaths
311388
+4157

Minor Project Submission by: Rajan, Rippen and Sanyam

Fig: Screenshot of COVID-19 Tracker



USA
change

Total Cases
33166418

Recovered
0
+0

Deaths
590941
+621

Minor Project Submission by: Rajan, Rippen and Sanyam

Fig: Screenshot of COVID-19 Tracker

5.2 Deployment and Testing Status

We are happy to announce that the Web Application is ready to be deployed at any cloud platform.
Still, the testing is going on. When the testing will be complete, we will be deploying it on to herokuapp.com.

6. CONCLUSION

Our Project- COVID-19 Detection and Tracker is complete and fully functional. We hope that the aims and objectives would be achieved and this project would contribute in improving the current scenario of our country by helping in the prevention of spread of the virus and also it would prove beneficial for the patients by providing them with a probability that how likely are they positive for COVID-19 infection so that they can isolate themselves and start the cure.

Conclusively, what we have achieved is: -

- To provide an easy to use, economically feasible solution that could help in early detection of COVID-19 at home as well as it provides the worldwide statistics of COVID-19.

This is in-line with the pathway to Digitization where everything can be done with a button-click. We have learned a lot both- technically as well as non-technically (including team work, time management etc.).

Thank You. Any kind of suggestions are most welcomed.

---------------END OF REPORT--------------