

# CHAPTER 1

## INTRODUCTION

### 1.1 AIM:

The project aims at developing a low cost, efficient and practical way that helps the users to identify the covid by uploading the x-ray and CT-scan images and thereby also predicts the disease if there is any. Disease identification has a major advantage at knowing covid, to identify the covid patient for save our country. It is very important to detect or classify images whether the image belongs to COVID positive or COVID negative. A few checkup like RT-PCR are available in the market, however, they are expensive and also take too time for detecting and giving results.

### 1.2 OBJECTIVE AND SCOPE OF THE PROJECT

The main goal of "*Artificial intelligence and covid-19: Deep learning approaches for diagnosis and treatment*" is to identify different patient that who is covid positive or who is covid negative from a real time report of X-ray and CT-Scan in a more effective and remarkable manner. This approaches can be useful in eliminating the disadvantages of the existing system like insufficient , number of available RT-PCR test kit , test costs and waiting time of test result. It is simple , accurate and fast Artificial Intelligence model that provide timely assistance to patients. There is no need to wait long hours for the radiologists to screen the images. The spread of diseases can be significantly reduced because of this system predict the result on time. The system can be used worldwide in hospitals for predict and identify the covid positive patient.

### 1.3 DEFINITION OF PROBLEM

To develop a novel model for automatic COVID-19 detection using raw images of chest x-ray and CT using ML techniques. Identifying covid positive patient is very important way for making safe to our country and the people of our country . It gives result in just a second for that we easily find out the person and start his treatment on time.

## 1.4 PROJECT DISCRPTION

The sudden spike in the number of patients with COVID-19 , a new respiratory virus, has put unprecedented load over healthcare systems across the world. In many countries, the healthcare systems have already been overwhelmed. There are limited kits for diagnosis, limited hospital beds for admission of such patients, limited personal protective equipment (PPE) for healthcare personnel and limited ventilators. It is thus important to differentiate which patients with severe acute respiratory illness (SARI) could have COVID-19 infection in order to efficiently utilize the limited resources. In this work we propose the use of chest X-Ray and CT-Scan to detect COVID-19 infection in the patients . Using our tool one can classify a given X-Ray in one of the two classes: normal, and covid Positive.

### **The use of X-Ray has several advantages over conventional diagnostic tests:**

1. X-Ray imaging is much more widespread and cost effective than the conventional diagnostic tests.
2. Transfer of digital X-Ray images does not require any transportation from point of acquisition to the point of analysis, thus making the diagnostic process extremely quick.
3. Unlike CT Scans, portable X-Ray machines also enable testing within an isolation ward itself, hence reducing the requirement of additional Personal Protective Equipment (PPE), an extremely scarce and valuable resource in this scenario. It also reduces the risk of hospital acquired infection for the patients.

The main contribution of this work is in proposing a novel deep neural network based model for highly accurate detection of COVID-19 infection from the chest X-Ray images of the patients. Radiographs in the current setting are in most cases interpreted by non-radiologists. Further, given the novelty of the virus, many of the radiologists themselves may not be familiar with all the nuances of the infection, and may be lacking in the adequate expertise to make highly accurate diagnosis. Therefore this automated tool can serve as a guide for those in the forefront of this analysis. We would like to re-emphasize that we are not proposing the use of the proposed model as alternative to the conventional diagnostic tests for COVID19 infection, but as a triage tool to determine the suitability of a patient to undergo the test for COVID-19 infection.

To help accelerate the research in this area, we are releasing our training code and trained models publicly for open access at [https://github.com/ arpanmangal/CovidAID](https://github.com/arpanmangal/CovidAID). However, we note that both the model and this report merely captures our current understanding of this rapidly evolving problem, that too on very limited data currently available. We will keep updating the model and this report as we get newer understanding and better results. This Covid-19 detection detects Covid-19, Pneumonia and Normal from Xrays & CT-scans. Its accuracy rate is 98% on training and 93% on testing

**METHODOLOGY:-** Here we take a 256 x 256 input image and multiply it with a different feature detector with a stride of 1 and also with rectilinear activation function to get the best feature from the input image .Here we take 3 x 3 feature detector and multiply with huge matrix of 256 x 256 input image to reduce the size of the matrix of the image. After that we apply Max pooling, so we take a box of two by two pixels and place it in the top left hand corner and we find the maximum value in that box and then we reduced only that value. Then we move the box to the right with a stride of two. Here we do Max pooling to reduce the size of the 32 different feature maps and as a result we get 32 pooled feature map. To get best accuracy in our train and test set we have added 3 more convolution layer - First one with 3 x 3 matrix of 32 feature detector. Second one is with 3 x 3 matrix of 64 feature detector and third one is 3 x 3 matrix of 128 feature detector and I have added max pooling with each of these convolution layer. Now we take each and every pooled feature map and flatten it into a column so basically you just take the number row by row from pooled feature map and put them into one long column to get a one huge vector of inputs for an artificial neural network .

Now we make two fully connected layer with output\_dim(Dimension of dense embedding ) of 128 and also with rectilinear activation function we use categorical cross entropy to calculate the loss so the error is calculated in the output layer with softmax activation function and with adam optimizer is back propagated through the network again and again to adjust the network and to optimize the performance. After that we train the dataset of Normal, and Covid-19 with a target size of 256 x 256 batch size of 10 and also the test set of Normal, and Covid-19 with a target size of 256 x 256 batch size of 2 and with 50 epoch .and we get 98% accuracy on train set and 93% accuracy on test set. Then we take a infected x-ray image of a lung and test it in which category it was either it is covid-19 infected lung, effected or normal.

If the result is either covid-19 we take the image and do some image processing on that image to get more details. The image processing we used in our model are Binary, Binary Inverted, Zero, Zero Inverted, contours and truncated.

**Way to detect COVID-19 :**

1. Ground-Glass Opacities
2. Crazy Paving Appearance
3. Effected Bronchioles
4. Air Space Consolidation
5. Texture and Sharpness

**The main contribution of my work are the following:**

1. We propose a CNN approach that learn image features capturing tissue heterogeneity, which can effectively COVID-19 infection with limited training data.
2. To the best of my knowledge, this is the first work to analyze deep features by integrating separate CT lung images (I.e. GGO and PE) in DTL models.
3. We present a comprehensive analysis of CNN model for COVID-19 predictions, involving several datasets of different modalities and six deep CNN architectures. Our results demonstrate the potential of the proposed approach for differentiating between COVID-19 and normal patients.

## **CHAPTER 2**

### **LITERATURE SURVEY**

#### **2.1 EXISTING SYSTEM**

1. we used RT-PCR test for finding the covid positive but it is expensive, as it requires trained professionals, RNA extraction machines, and a laboratory.
2. It takes minimum four hours to get the RT-PCR test results and gauge the extent of infection in an individual. The test can be completed in four to eight hours, however, the results are available in one day due to time taken in collection and in the transportation of samples to the labs. If the testing lab is far from the place where the sample of the person has been collected, the test result may be available after 48 hours or more, in some cases.
3. On the other hand, the results of the rapid antigen test can come out in about half an hour but the accuracy is less than that of an RT PCR test.
4. The Indian Council of Medical Research had earlier quoted Rs.4,500 as the cost of PCR-Test for private laboratories in India, same as the pathology lab. At present, the price of an RT-PCR test to detect COVID-19 is somewhere between Rs. 2,200 and Rs.3,000 in India
5. Because of time consuming and cost, most of people didn't do their RT-PCR test.

#### **DISADVANTAGE**

1. Time consuming
2. Costly compare to proposed system
3. Requires trained professionals, RNA extraction machines, and a laboratory.
4. Although screening, diagnosis, and progress assessment of COVID-19 have been effectively performed through reliance on radiological examinations, including CT and digital photography (DR), there has been not much prior experience that could come to help radiologists and technologists to deal with COVID-19 patients.

## **2.2 PROPOSED SYSTEM**

AI approaches can be useful in eliminating disadvantages such as insufficient number of available RT-PCR test kits, test costs, and waiting time of test results.

It is simple, accurate, and fast AI models may be helpful to overcome this problem and provide timely assistance to patients.

The proposed model has an end-to-end architecture without using any feature extraction methods, and it requires raw chest X-ray images to return the diagnosis.

The models can be readily used in healthcare centers. There is no need to wait long hours for the radiologists to screen the images. Thus, healthcare workers and patient relatives can focus on isolation of suspicious cases so that treatment can begin.

the spread of the disease can be significantly reduced. The patients can seek a second opinion if they are diagnosed as positive by our system. Hence, waiting time can be significantly reduced, and it will alleviate clinician workload.

The main objective of the proposed structure is to improve the accuracy and speed of recognition and classification of the issues caused by the virus by utilizing DL-based methods.

We proposed an LSTM equipped model, which is the classification of the best treatment method. LSTM networks seem to be good options for classification, process, and prediction according to time series data because lags of unknown duration may take place between major events in a time series.

### **ADVANTAGE**

- Take less time compare to existing system
- Less costly compare to existing system
- More Accurate upto 96%
- User Friendly

## **CHAPTER 3**

### **METHODOLOGY AND DESIGN**

Our proposed system could be divided into five main steps:

- ✓ PreProcessing
- ✓ Segmentation
- ✓ Feature extraction
- ✓ Training
- ✓ Testing and detection

#### **3.1 PREPROCESSING**

Preprocessing is required to clean image data for model input. Adjusting existing training data to generalize to other situations allows the model to learn from a wider array of situations.

##### **Algorithm:**

- ✓ Read the picture files (stored in data folder).
- ✓ Decode the JPEG content to RGB grids of pixels with channels.
- ✓ Convert these into floating-point tensors for input to neural nets.
- ✓ Rescale the pixel values (between 0 and 255) to the [0, 1] interval (as training neural networks with this range gets efficient).

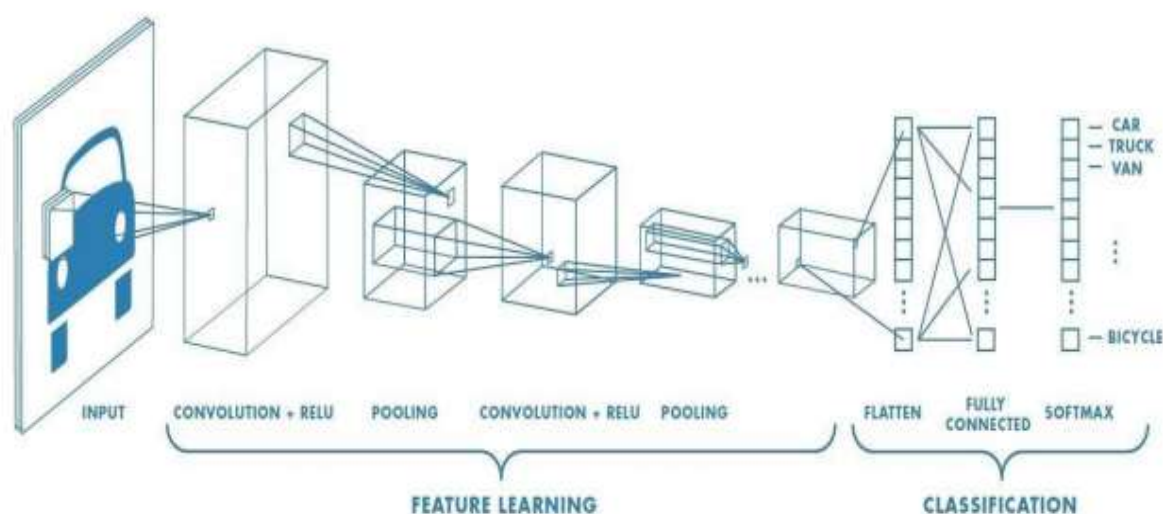
#### **3.2 SEGMENTATION :**

After the preprocessing step, an image of medicine is decomposed into sub-images of individual medicines. Preprocessed input image is segmented into isolated medicine name using a labeling process. This labeling provides information about number of medicine names in the image. Each individual digit is uniformly resized into 100 X 70 pixels for classification and recognition stage.

### 3.3 TRAINING AND TESTING :

After segmentation, CNN algorithm is trained separately with the training images.

#### 3.3.1 CONVOLUTIONAL NEURAL NETWORK



**Figure 3.3.1 A simple structure of CNN**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.



### 3.3.2. Convolution Layer — The Kernel

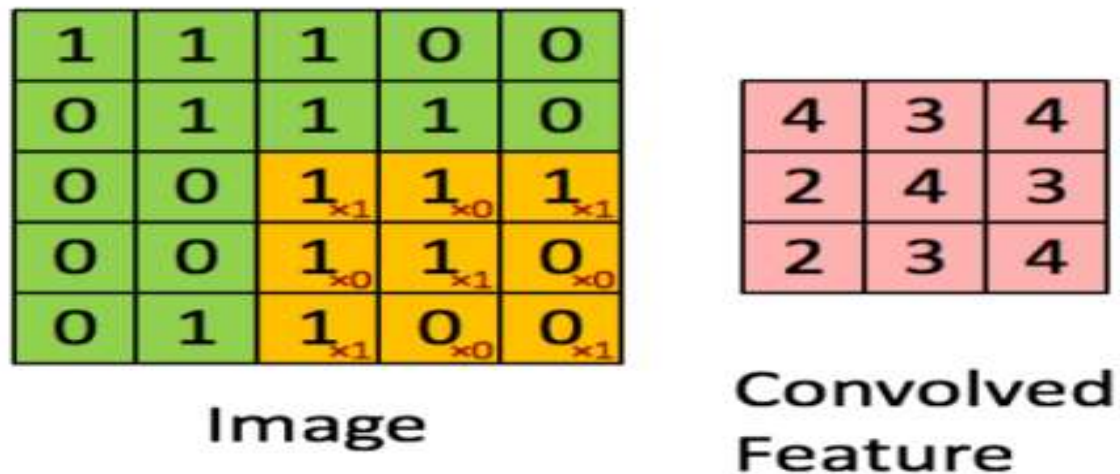


Figure 3.3.2 Convolved Layer-The Kernel

In the above demonstration, the green section resembles our input image. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel/Filter, K, represented in the color yellow. We have selected K as a 3x3x1 matrix.

The objective of the Convolution Operation is to extract the high-level features such as edges, from the input image. ConvNets need not be limited to only one Convolutional Layer. Conventionally, the first ConvLayer is responsible for capturing the Low-Level features such as edges, color, gradient orientation, etc. With added layers, the architecture adapts to the High-Level features as well, giving us a network which has the wholesome understanding of images in the dataset, similar to how we would.

### 3.3.3 POOLING

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model.

There are two types of Pooling: Max Pooling and Average Pooling. Max Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.

Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

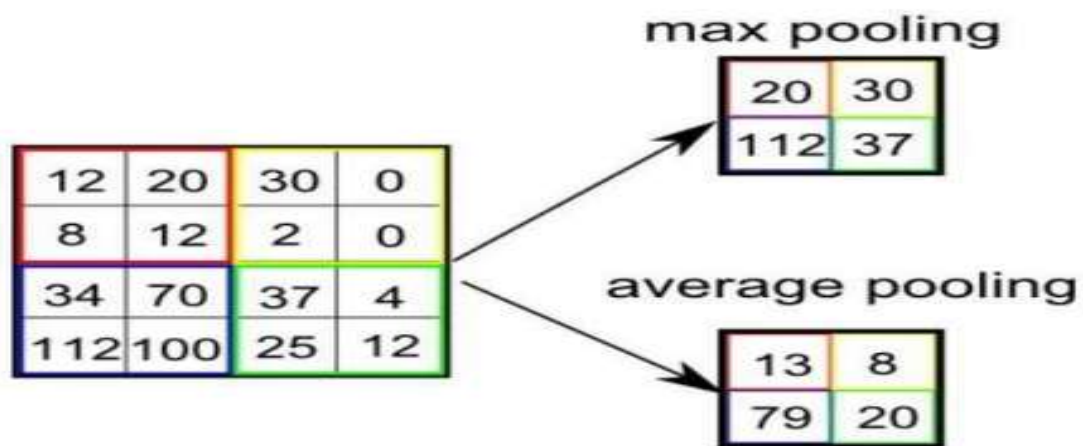


Figure 4.3.3: Types of pooling

The Convolutional Layer and the Pooling Layer, together form the i-th layer of a Convolutional Neural Network. Depending on the complexities in the images, the number of such layers may be increased for capturing low-levels details even further, but at the cost of more computational power. After going through the above process, we have successfully enabled the model to understand the features. Moving on, we are going to flatten the final output and feed it to a regular Neural Network for classification purposes.

### 3.4 ACTIVITY DIAGRAM

Activity diagrams can also be used to model a specific Actor's workflow within the entire system. Activity diagram can also be used independent of use cases for other purposes such as to model business process of a system, to model detailed logic of business rules etc. Activity diagram shows all potential sequence flows in an activity. The Activity Diagram is comprised of the following model elements.

#### Initial State

An initial state is a model element that explicitly shows the beginning of a workflow on an activity diagram. The initial state is drawn as a solid circle with an optional name or label.



#### Action

An Action is a model element that represents the performance of a task in workflow or operation. Action is drawn as a capsule shaped rounded rectangle with a name.



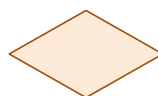
#### Transition

Transition element connects the various elements of the activity diagram.



#### Decision Point

A decision is a model element that typically has one incoming transition and two or more outgoing transitions based upon the outcome of guard conditions from the previous element.

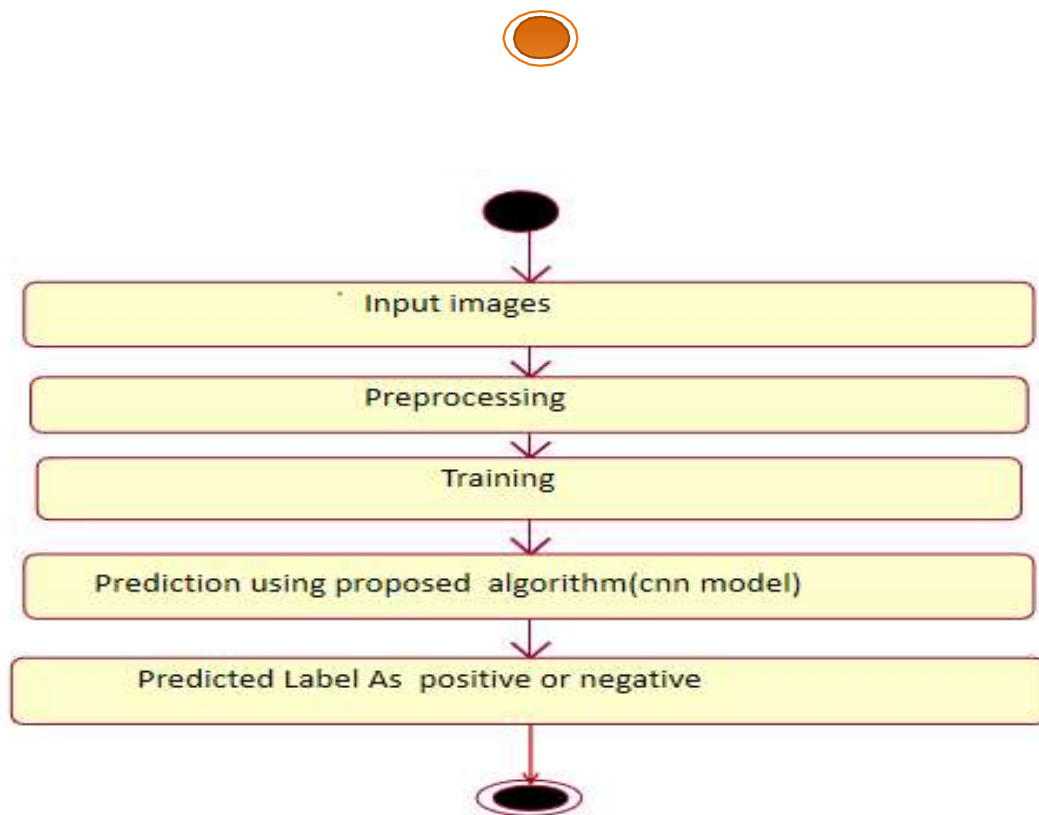


## Synchronization

Synchronization is a model element that allows modeling of simultaneous workflow.

## Final State

A final state is a model element that explicitly shows the end of a workflow.



**Fig 3.4.1** Project Activity Diagram

**Description:** Activity diagram is basically a flowchart to represent the flow from one activity to another activity.

### **3.5 UML DIAGRAM**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

#### **GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

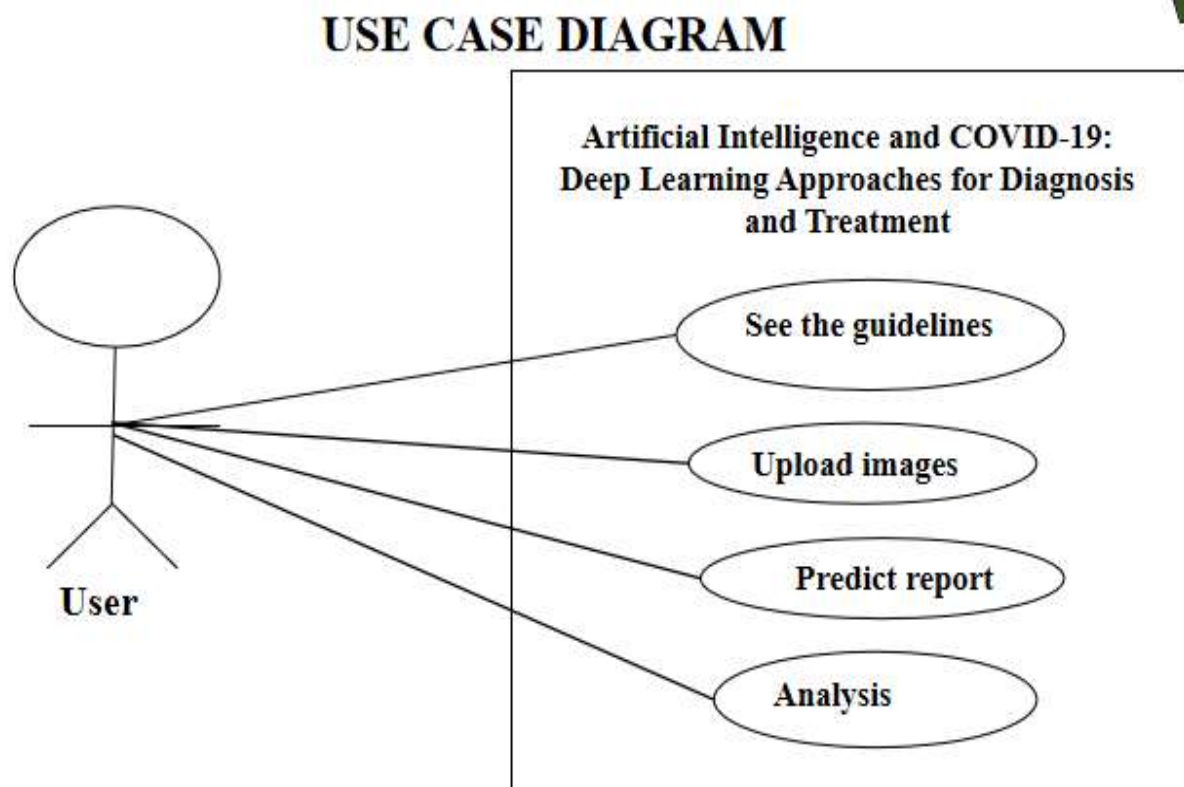
## USE-CASE DIAGRAM

The purpose of use case diagram is to capture the dynamic aspect of a system. Use case diagrams are used to gather the requirements of a system including internal and external influences. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

### Usage of use case diagrams are:

- Used to gather requirements of a system.
- Used to get an outside view of a system.
- Identify external and internal factors influencing the system.
- Show the interactions among the requirements and actors.

The use case diagrams consist of four objects: Actor, Use case, System and Package.



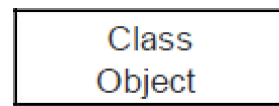
**Fig 3.5.1** Project Use-Case diagram

### 3.6 SEQUENCE DIAGRAM

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They are also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the progress of modeling a new system.

#### Notations:

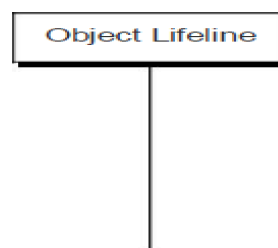
**Class Roles or Participants :-** Class roles describe the way an object will behave in context.



**Activation or Execution Occurrence :-** Activation boxes represent the time an object needs to complete a task. When an object is busy executing a process or waiting for a reply message, use a thin gray rectangle placed vertically on its lifetime.



**Lifelines: -** Lifelines are vertical dashed lines that indicate the objects presence over time.

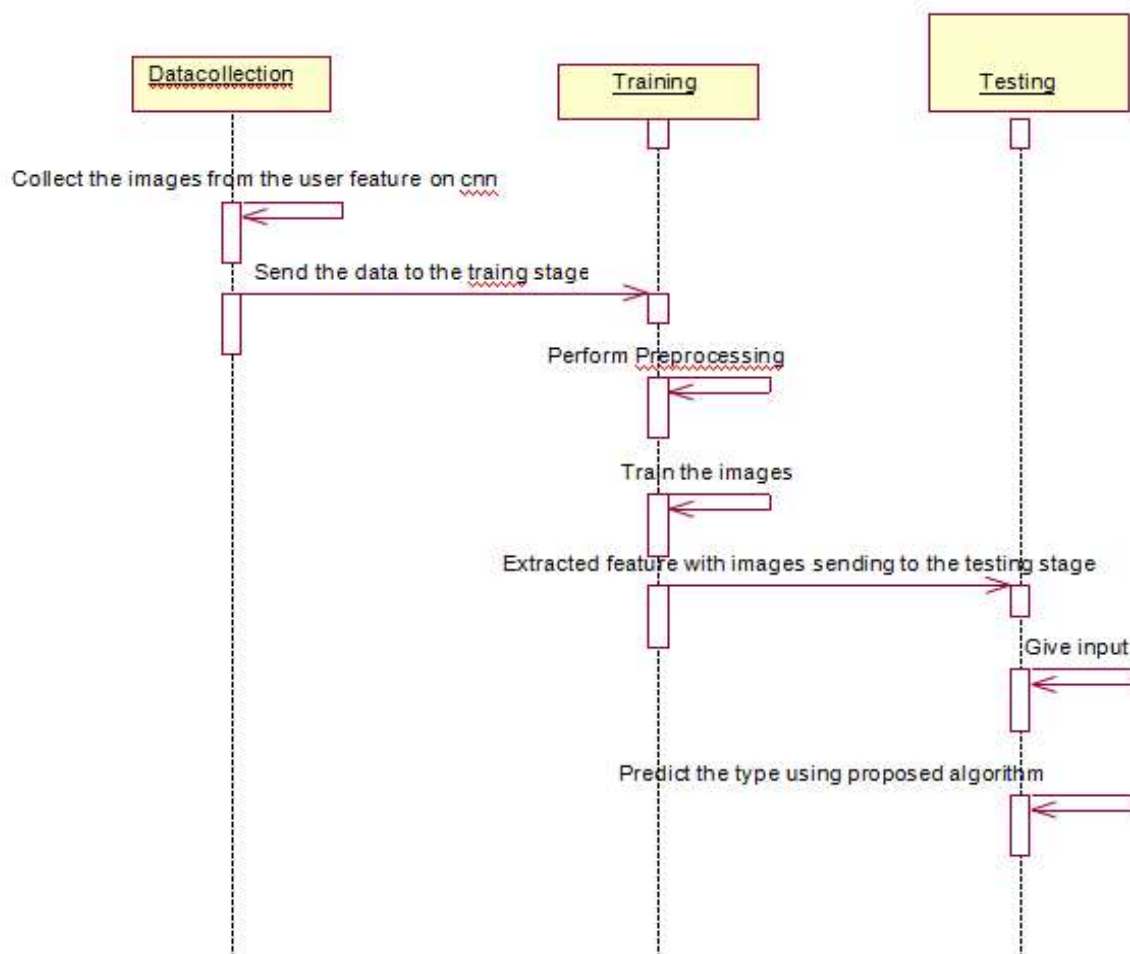


**Messages:** - Messages are arrows that represent communication between objects.

Usage of Sequence Diagram:

- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.

Plan and understand the detailed functionality of an existing or future scena.



**Figure 3.6.1:** Project Sequence diagram

### Description:

Sequence diagrams are sometimes called event diagrams or event scenarios. It shows how and in what order a group of objects work together.

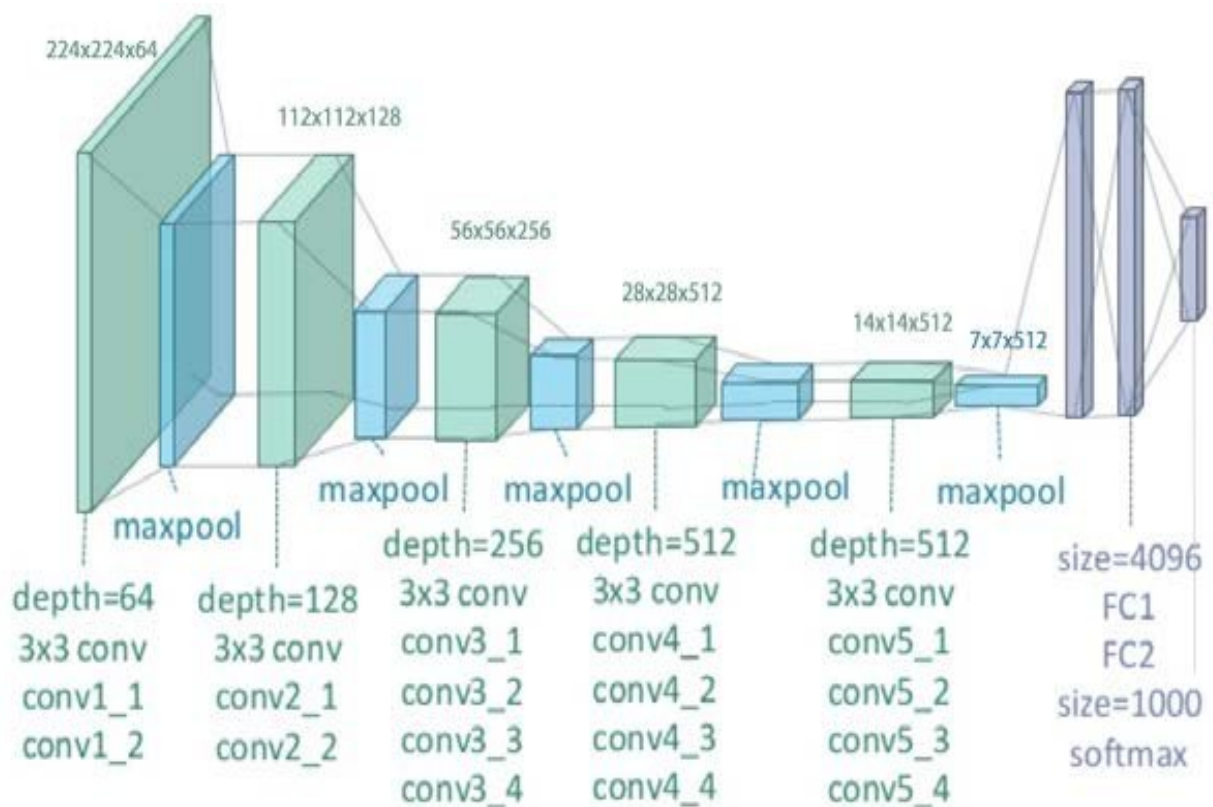


# CHAPTER 4

## IMPLEMENTATION

### 4.1 SYSTEM ARCHITECTURE

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.



**Fig 4.1.1** System Architecture

## **4.2 SOFTWARE IMPLEMENTATION:**

### **MODULES:**

- Dataset
- Importing the necessary libraries
- Retrieving the images
- Splitting the dataset
- Building the model
- Apply the model and plot the graphs for accuracy and loss
- Accuracy on test set
- Saving the Trained Model

### **MODULES DESCRIPTION:**

#### **Dataset:**

In the first module, we developed the system to get the input dataset for the training and testing purpose. We have taken the dataset from kaggle and some other out source.

Link: <https://www.kaggle.com/c/stat946winter2021/data?select=train>

Link: <https://www.kaggle.com/c/covidct/data>

The dataset consists of 4225 x-ray images.

The dataset consists of 672 ct-scan images.

#### **Importing the necessary libraries:**

We will be using Python language for this. First we will import the necessary libraries such as keras for building the main model, sklearn for splitting the training and test data, PIL for converting the images into array of numbers and other libraries such as pandas, numpy ,matplotlib and tensorflow.

#### **Retrieving the images:**

We will retrieve the images and their labels. Then resize the images to (224,224) as all images should have same size for recognition. Then convert the images into numpy array.

### **Splitting the dataset:**

Split the dataset into train and test. 80% train data and 20% test data.

### **Building the model:**

The concept of convolutional neural networks. They are very successful in image recognition. The key part to understand, which distinguishes CNN from traditional neural networks, is the convolution operation. Having an image at the input, CNN scans it many times to look for certain features. This scanning (convolution) can be set with 2 main parameters: stride and padding type. As we see on below picture, process of the first convolution gives us a set of new frames, shown here in the second column (layer). Each frame contains an information about one feature and its presence in scanned image.

Resulting frame will have larger values in places where a feature is strongly visible and lower values where there are no or little such features. Afterwards, the process is repeated for each of obtained frames for a chosen number of times. In this project I chose a classic VGG-16 and V3 model which contains only two convolution layers.

The latter layer we are convolving, the more high-level features are being searched. It works similarly to human perception. To give an example, below is a very descriptive picture with features which are searched on different CNN layers. As you can see, the application of this model is face recognition. You may ask how the model knows which features to seek. If you construct the CNN from the beginning, searched features are random. Then, during training process, weights between neurons are being adjusted and slowly CNN starts to find such features which enable to meet predefined goal, i.e. to recognize successfully images from the training set. Between described layers there are also pooling (sub-sampling) operations which reduce dimensions of resulted frames.

Furthermore, after each convolution we apply a non-linear function (called **ReLU**) to the resulted frame to introduce non-linearity to the model.

Eventually, there are also fully connected layers at the end of the network. The last set of frames obtained from convolution operations is flattened to get a one-dimensional vector of neurons. From this point we put a standard, fully-connected neural network. At the very end,

for classification problems, there is a softmax layer. It transforms results of the model to probabilities of a correct guess of each class

### **Apply the model and plot the graphs for accuracy and loss:**

For X-ray, We will compile the model and apply it using fit function. The batch size will be 100. Then we will plot the graphs for accuracy and loss. We got average accuracy of 65.7% and average validation accuracy of 70.01%.

For CT-Scan, We will compile the model and apply it using fit function. The batch size will be 100. Then we will plot the graphs for accuracy and loss. We got average accuracy of 95.6% and average validation accuracy of 70.3%.

### **Accuracy on test set:**

For X-ray, We got a accuracy of 97% on test set.

For CT-Scan, We got a accuracy of 97% on test set.

### **Saving the Trained Model:**

Once you're confident enough to take your trained and tested model into the production ready environment, the first step is to save it into a .h5 or .pkl file using a library like pickle. Make sure you have pickle installed in your environment. Next, let's import the module and dump the model into .h5 file

### **Home Page**

The user is redirected to the main page once login is done successfully. From this page, the user can perform the further operations that the user wishes to perform.

**A. Guidance to Use** - Here, I show the guidance for using this software that how it works

**B. X-Ray Report**- Here It takes x-ray report of chest as input and analysis that report using ANN, Generative Adversarial Networks (GANs), Extreme Learning Machine (ELM), and Long /Short Term Memory (LSTM). and produce the output as that given x-ray is covid positive or covid negative.

**C. CT-Scan** - Here It takes CT-Scan report of chest as input and analysis that report using ANN, Recurrent Neural Network (RNN), data mining and Artificial intelligence and produce the output as that given x-ray is covid positive or covid negative.

**D. Analysis-** Here we analysis the report of covid positive and covid negative and see the chart that how much persons are covid positive and covid negative.

## 4.3 SYSTEM REQUIREMENTS

### 4.3.1 Python:

Python is one of those rare languages which can claim to be both simple and powerful. You will find yourself pleasantly surprised to see how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in.



Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

Guido van Rossum, the creator of the Python language, named the language after the BBC show "Monty Python's Flying Circus". He doesn't particularly like snakes that kill animals for food by winding their long bodies around them and crushing them.

Used in Web Development, Game Development, Machine Learning and Artificial Intelligence, Data Science and Visualization, Desktop and business applications etc.

### Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly

- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

## Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

### Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

#### 4.3.2 NUMPY:

NumPy is the fundamental package for scientific computing with Python. NumPy is an acronym for “Numeric Python” or “Numerical Python”. It is an open source extension module for Python, which provides fast precompiled functions for mathematical and numerical routines. Furthermore, NumPy enriches the programming language Python with powerful data structures for efficient computation of multi-dimensional arrays and matrices. The implementation is even aiming at huge matrices and arrays. Besides that the module supplies a large library of high-level mathematical functions to operate on these matrices and arrays

#### 4.3.3 KERAS:

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

#### 4.3.4 TENSORFLOW:

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

#### 4.3.5 PILLOW:

It is a free and open-source additional library for the python programming language that adds support for opening, manipulating and saving many different image file formats like JPG, PNG, SVG etc.

#### 4.3.6 FLASK FRAMEWORK:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. HTTP protocol is the foundation of data communication in the world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

The following table summarizes different HTTP methods –

Sr.No	Methods & Description
1	<b>GET</b>  Sends data in unencrypted form to the server. Most common method.



2	<b>HEAD</b>  Same as GET, but without response body
3	<b>POST</b>  Used to send HTML form data to server. Data received by POST method is not cached by server.
4	<b>PUT</b>  Replaces all current representations of the target resource with the uploaded content.
5	<b>DELETE</b>  Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator. In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

**Save the following script as login.html**

```
<html>

<body>

<form action="http://localhost:5000/login" method="post">

<p>Enter Name:</p>

<p><input type="text" name="nm"/></p>

<p><input type="submit" value="submit"/></p>

</form>
```

```
</body>
```

```
</html>
```

Now enter the following script in Python shell.

```
from flask import Flask, redirect, url_for, request

app=Flask(__name__)

@app.route('/success/<name>')

def success(name):

    return 'welcome %s'% name

@app.route('/login', methods=['POST', 'GET'])

def login():

    if request.method == 'POST':

        user=request.form['nm']

        return redirect(url_for('success', name= user))

    else:

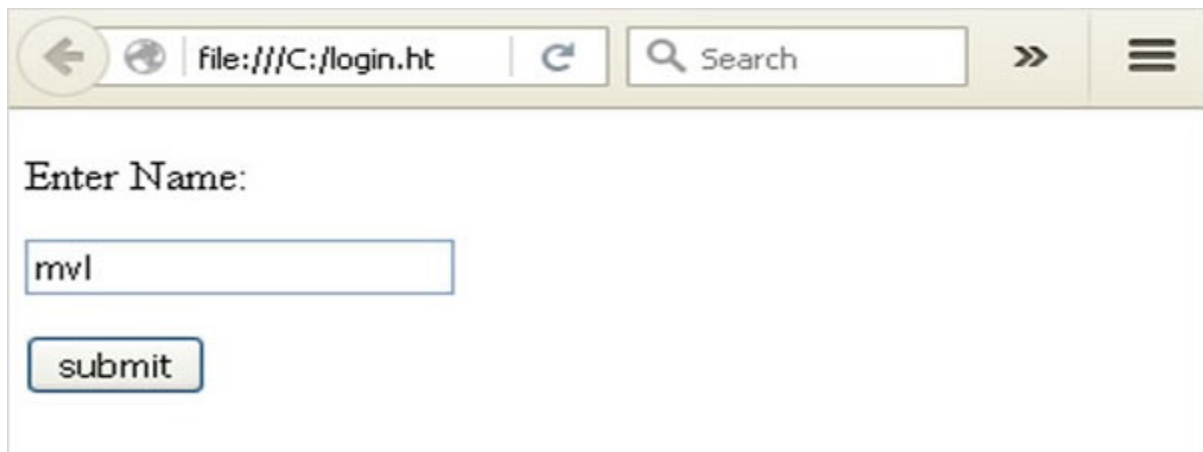
        user=request.args.get('nm')

        return redirect(url_for('success', name= user))

if __name__ == '__main__':

    app.run(debug = True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.



Enter Name:

Form data is POSTed to the URL in action clause of form tag.

**http://localhost/login** is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by –

```
user = request.form['nm']
```

It is passed to **‘/success’** URL as variable part. The browser displays a **welcome** message in the window.



welcome mvl

Change the method parameter to **‘GET’** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of 'nm' parameter is now obtained by –

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to 'nm' parameter is passed on to '/success' URL as before.

#### **4.3.7 WINDOWS 10:**

Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and broadly released for retail sale on July 29, 2015. Windows 10 receives new builds on an ongoing basis, which are available at no additional cost to users, in addition to additional test builds of Windows 10 which are available to Windows Insiders.

Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

One of Windows 10's most notable features is support for universal apps, an expansion of the Metro-style apps first introduced in Windows 8. Universal apps can be designed to run across multiple Microsoft product families with nearly identical code—including PCs, tablets, smart phones, embedded systems, Xbox One, Surface Hub and Mixed Reality. The Windows user interface was revised to handle transitions between a mouse-oriented interface and a touch screen-optimized interface based on available input devices— particularly on 2-in-1 PCs, both interfaces include an updated Start menu which incorporates elements of Windows 7's traditional Start menu with the tiles of Windows 8. Windows 10 also introduced the Microsoft Edge web browser, a virtual desktop system, a window and desktop management feature called Task View, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX 12.

Windows 10 received mostly positive reviews upon its original release in July 2015. Critics praised Microsoft's decision to provide a desktop-oriented interface in line with previous versions of Windows, contrasting the tablet-oriented approach of 8, although Windows 10's touch-oriented user interface mode was criticized for containing regressions upon the touch oriented interface of Windows 8. Critics also praised the improvements to Windows 10's

bundled software over Windows 8.1, Xbox Live integration, as well as the functionality and capabilities of the Cortana personal assistant and the replacement of Internet Explorer with Microsoft Edge. However, media outlets have been critical of changes to operating system behaviors, including mandatory update installation, privacy concerns over data collection performed by the OS for Microsoft and its partners and the adware-like tactics used to promote the operating system on its release.

Microsoft aimed to have Windows 10 installed on at least one billion devices in the two to three years following its release. Up to August 2016, Windows 10 usage was increasing, day by day. As of March 2019, the operating system is running on more than 800 million devices and has an estimated usage share of 32% on traditional PCs, making it the most popular version of Windows and the largest usage share of an OS overall and 15% across all platforms (PC, mobile, tablet, and console).

## CHAPTER 5

### TESTING

The aim of program testing is to identify all defects in a program. Testing a program involves providing the program with a set of test inputs (or test cases) and observing if the program behaves as expected. This phase is necessary in order to check the efficiency of the system developed. System testing is a critical element of the software quality assurance and represents the ultimate review of specification, design and coding. It is the process of exercising or evaluating a system by manual or automatic means to verify that it satisfies the specified requirements or to identify difference between expected and actual results.

Testing is the one step in the software engineering process that could be viewed as destructive rather than constructive. Testing requires that the developer discard preconceived notions of the correctness of the software just developed and overcome a conflict of interest that occurs when errors are uncovered.

If testing is conducted successfully, it uncovers errors in the software. As a secondary benefit, testing demonstrates that software functions appear to be working according to the specification. Testing provides a good indication of software reliability and some indication of software quality as a whole.

Testing cannot show the absence of defects, it can only show that software defects are present. As the developed software does not fulfill all the requirements of a user, so it is not possible to test with real time data. Still then we tried our best to test each individual module and also as an integrated modules (as a whole) with sufficient data that the user can have, fulfilling the objective of our Web Browser.

Testing performs a very critical role for quality assurance and ensuring there liability of the software. During testing, the program to be tested is executed with a set of test cases and output of the program for the test cases and output of the program for the test case is evaluated to determine if the program is performing as it is expected to. Hence

- Testing is the process of executing a program with the intention of finding errors.
- A successful test is one yet uncovers as yet undiscovered errors.

- A good test case is the one that has a high probability of finding as yet undiscovered error.

Testing is performed according to two different strategies:

- **Code Testing**-The code testing strategy examines the logic of program i.e. the analyst develops test cases that results in executing every instruction in the program. Basically during code testing every path through the program is tested.
- **Specification Testing**-To perform specification testing the analyst examines the specification starting what the program should do and how it should perform under various conditions. Then test cases are developed for each. In order to find which strategies to follow, levels of testing should be followed.

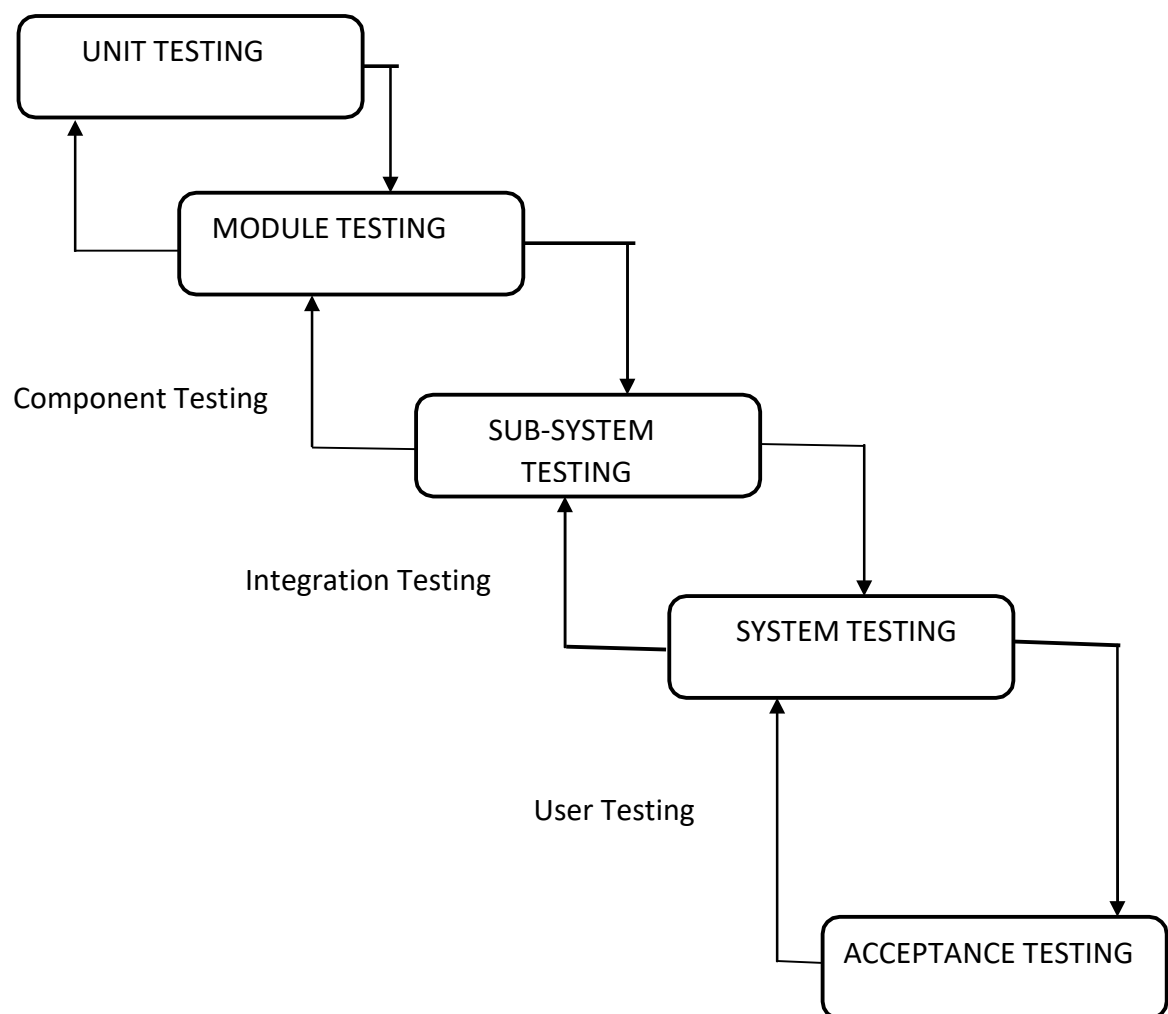
## 5.1 TESTING TECHNIQUES

### Levels of Testing:

The basic levels are unit testing, integration testing, system testing and acceptance testing. These different levels of testing attempt to detect different types of faults. The different levels of testing are as follows:

- **Unit Testing:** In this testing different modules are tested against specification produced during design of the modules. Unit testing is essential for verification of code produced during the coding phase and hence its main goal is to test internal logic modules.
- **Integration Testing:** In this testing tested modules are combined into subsystems which are then tested. The goal here is to see if the modules can be indicated properly and emphasis is being on testing interfaces between modules.
- **System testing:** In this testing the entire software system is tested. The reference document for this process is the requirements document and the goal is to see if the system meets its requirements. This is normally performing on realistic data of the client to demonstrate for the software is working satisfactorily. Testing here focus on external behavior of the system.

- **Black Box Testing:**Black Box Testing is testing the software without any knowledge of the inner working, structure and language of the module being tested. Black box tests, as most other kind of tests, must be written from a definite source document, such as specification and requirement documents. The test provides inputs and responds to outputs without considering how the software works.
- **White Box Testing:**White Box Testing is a testing in which software teacher has knowledge of the inner working structure and language of the software, or at least its purpose. It is used to test the code in deep that cannot be reached from a black box level.



**Figure 5.1.1:** Levels of Testing



## **5.2 USED TECHNIQUES**

### **Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software life cycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

### **Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

### **Test objectives**

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

### **Features to be tested**

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

### **Integration Testing:**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

### **Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

### **Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

# CHAPTER 6

## EVALUATION

### 6.1 FEATURES

The proposed system has many advantages over the existing system and practices. The developed application along with being different and reliable has the following advantages.

- AI approaches can be useful in eliminating disadvantages such as insufficient number of available RT-PCR test kits, test costs, and waiting time of test results.
- It is simple, accurate, and fast AI models may be helpful to overcome this problem and provide timely assistance to patients.
- The proposed model has an end-to-end architecture without using any feature extraction methods, and it requires raw chest X-ray images to return the diagnosis.
- The models can be readily used in healthcare centers. There is no need to wait long hours for the radiologists to screen the images. Thus, healthcare workers and patient relatives can focus on isolation of suspicious cases so that treatment can begin.
- The spread of the disease can be significantly reduced. The patients can seek a second opinion if they are diagnosed as positive by our system. Hence, waiting time can be significantly reduced, and it will alleviate clinician workload.

By using this application we can increase the number of recovered patient and save our country.

## 6.2 CRITICAL EVALUATION

The dataset is preprocessed such as Image reshaping, resizing and conversion to an array form. Similar processing is also done on the test image. A dataset consisting of about 13,500 images, out of which any image can be used as a test image for the software.

The train dataset is used to train the model (CNN) so that it can identify the test image and the disease it has CNN has different layers that are Dense, Dropout, Activation, Flatten, Convolution2D, and MaxPooling2D. After the model is trained successfully, the software can identify the disease if the plant species is contained in the dataset.

After successful training and preprocessing, comparison of the test image and trained model takes place to predict the disease. When we give a new input image first the module extracts the image features. Then it goes through the CNN model.

It then compares the features with already trained dataset. Then it goes through dense CNN and the image features are extracted separately. Then the module will predict whether the human is affected by COVID positive or not. It shows the output from the user friendly manner which are predetermined and trained.

## CHAPTER 7

### RESULTS

The web application, Smart Garden HTML pages, its activities and management operations have been checked and verified. There are multiple users for the Web Application

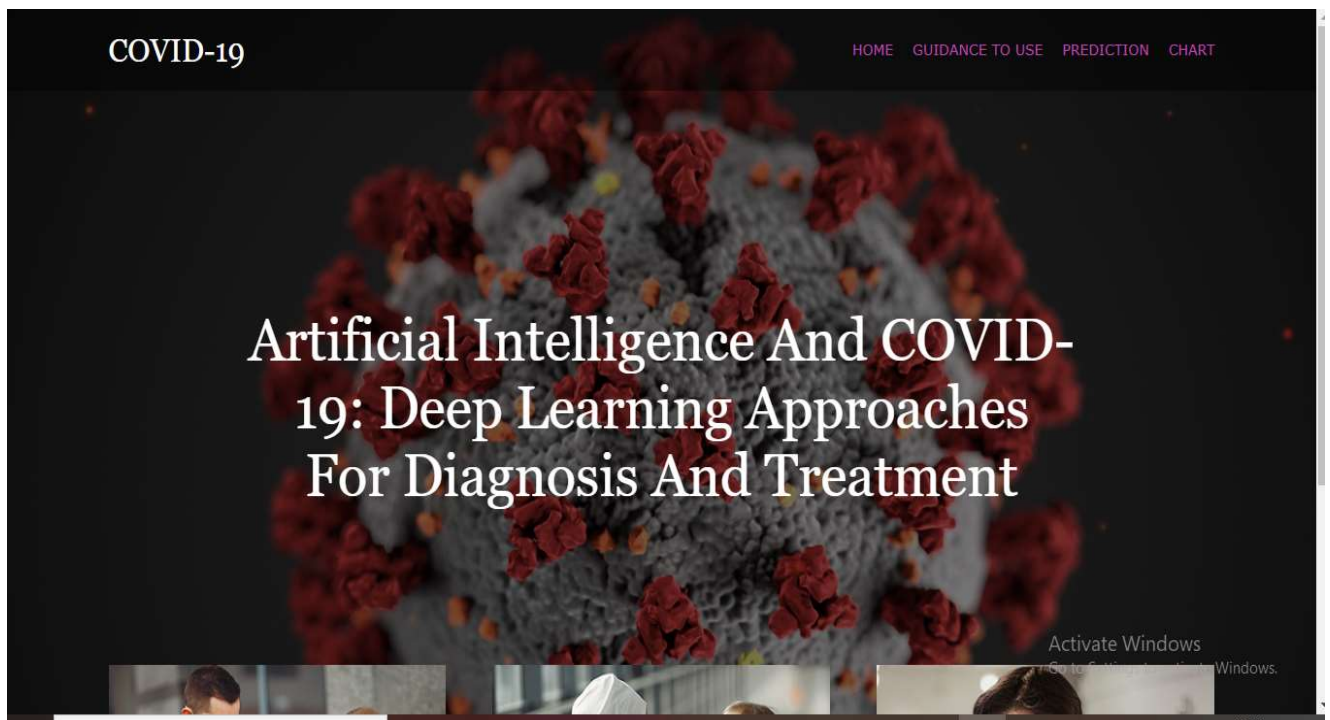


Fig 7.1 COVID-19 Detection Main Page

### GUIDENCE TO USE

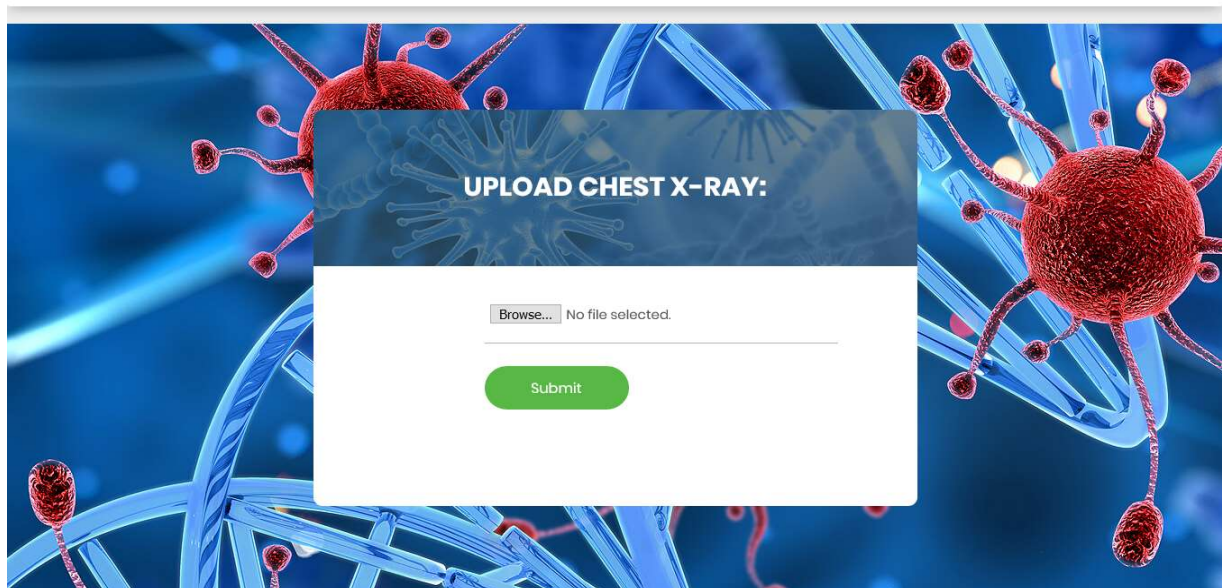
For using this software

- <1> upload the report of x-ray then click on predict button to see the result
- <2> upload ct-scan image after that click on predict button ,you will see the result
- <3> Click on chart to show the details of covid and non-covid person increment in months
- <4> Click on future to see the future scope of this software



Fig 7.2 guidance

## Test for COVID 19 using Chest X-Rays!



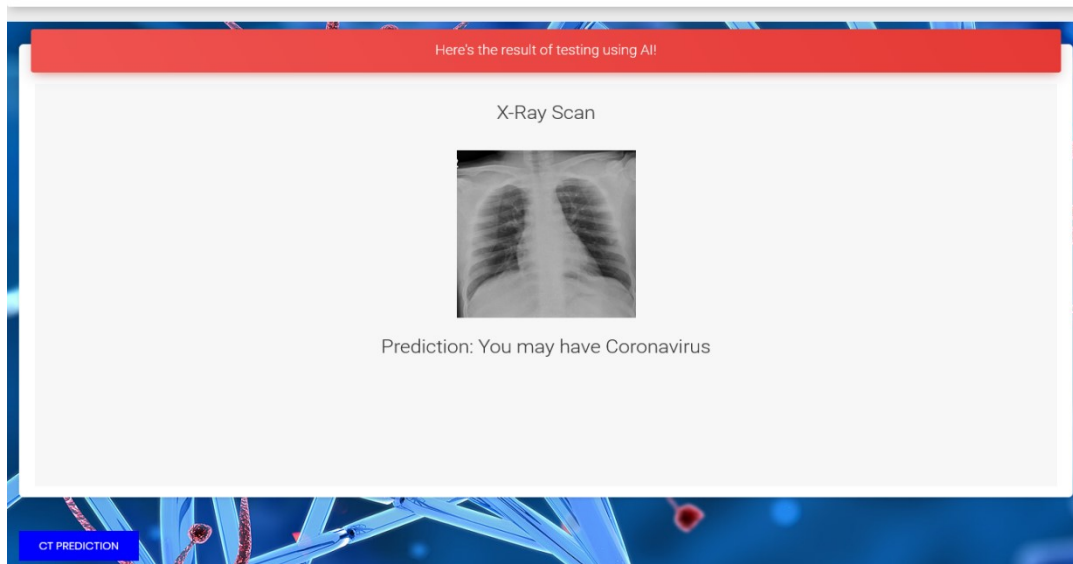
UPLOAD CHEST X-RAY:

Browse... No file selected.

Submit

**Fig 7.3 Uploading X-Ray Images**

## Test for COVID 19 using Chest X-Rays!



Here's the result of testing using AI!

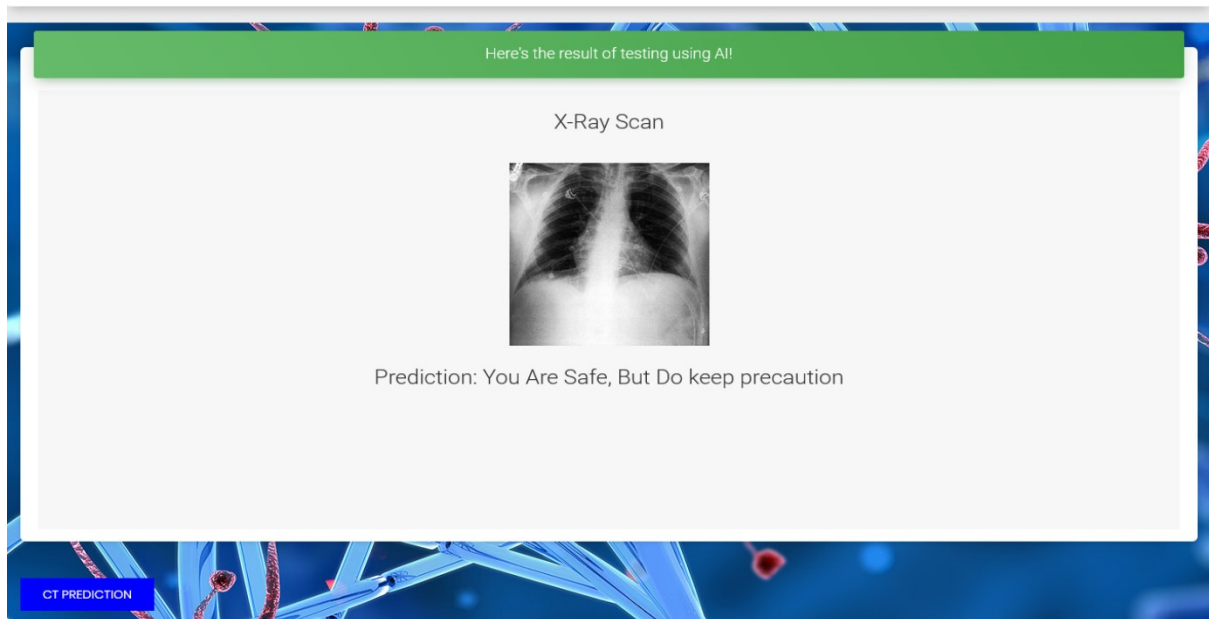
X-Ray Scan

Prediction: You may have Coronavirus

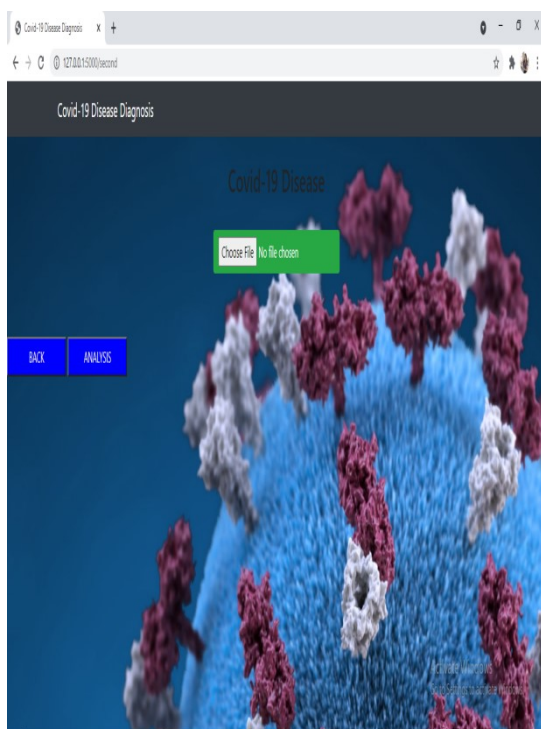
CT PREDICTION

**Fig 7.4 Chest x-ray covid positive report**

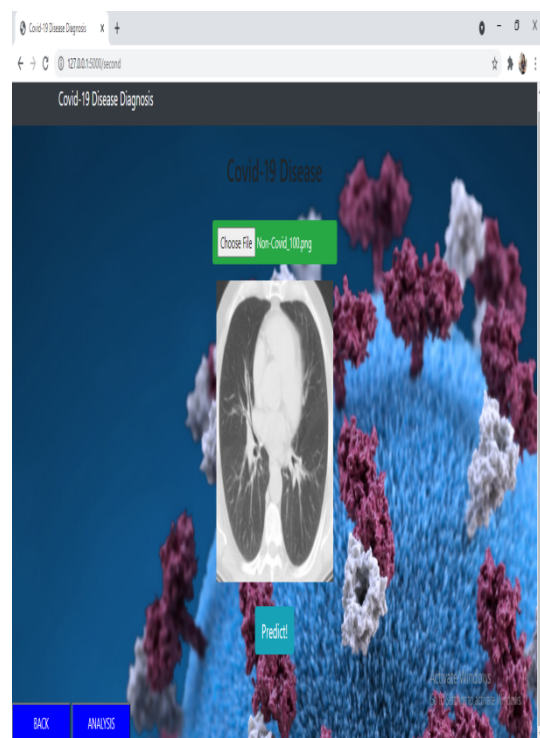
## Test for COVID 19 using Chest X-Rays!



**Fig 7.4.1 Chest x-ray covid negative report**

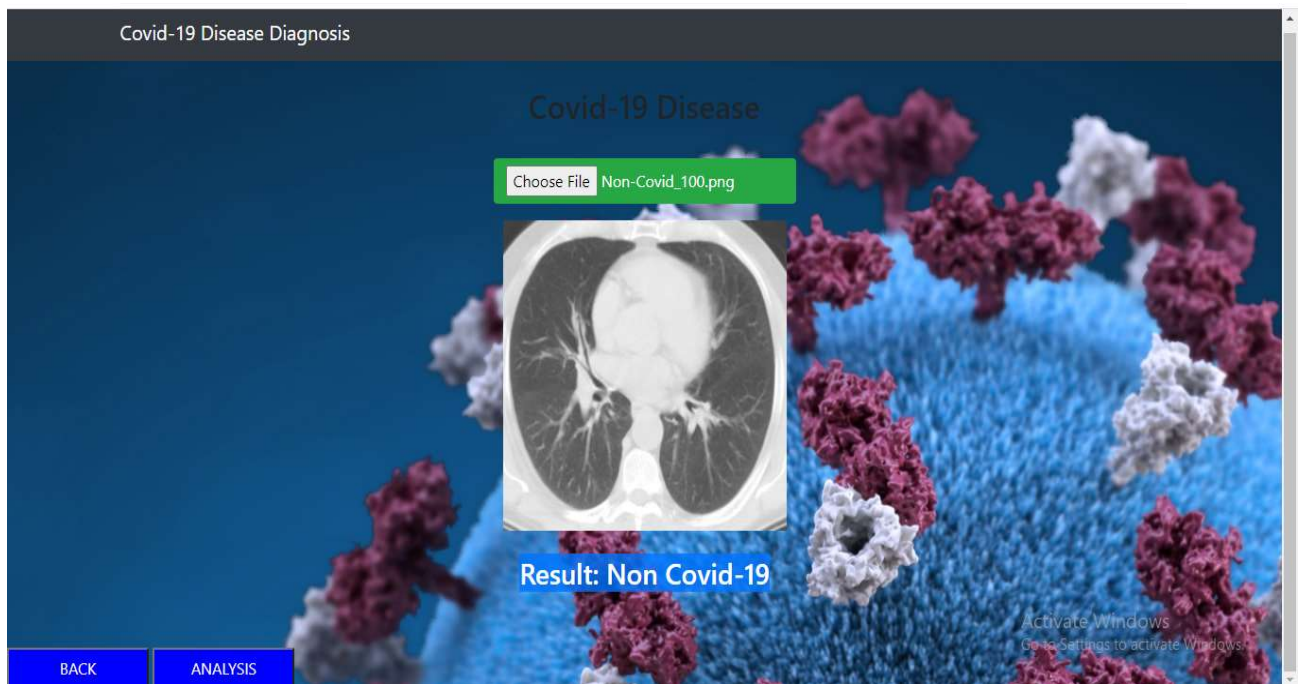


**Fig 7.5 Uploading CT-image**

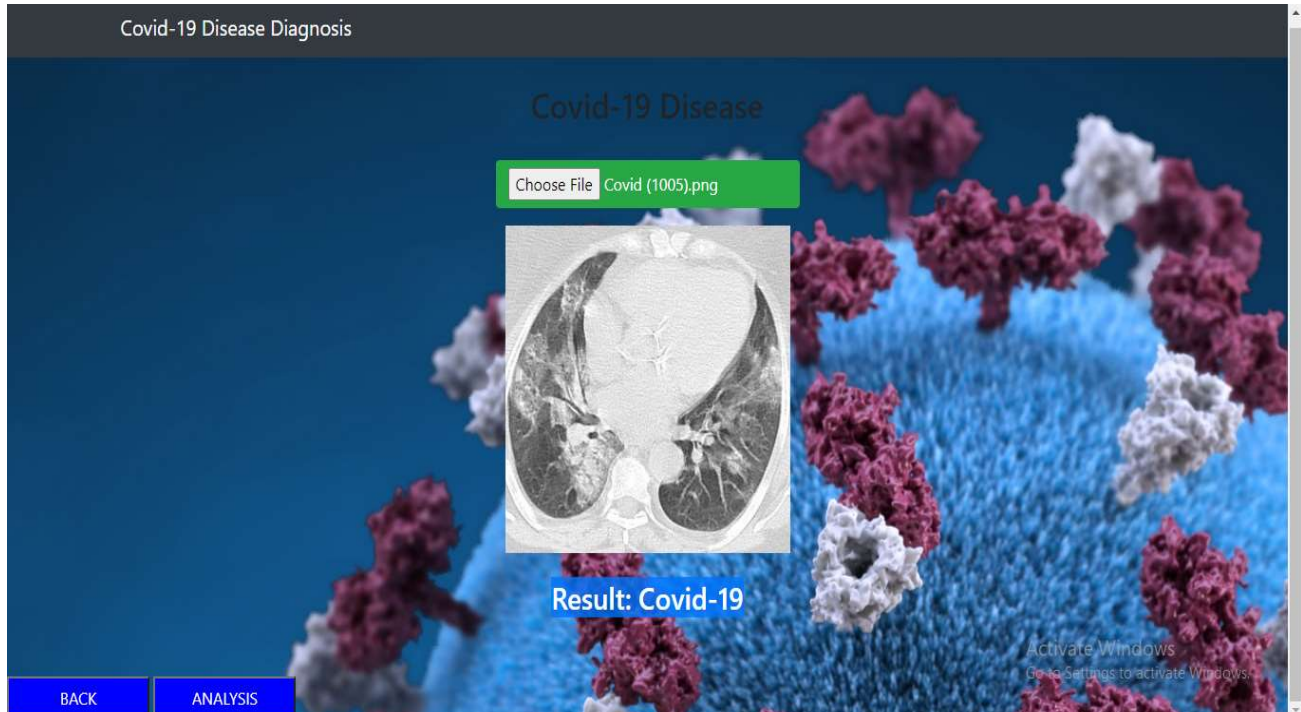


**Fig 7.5.1 Image uploaded successfully**

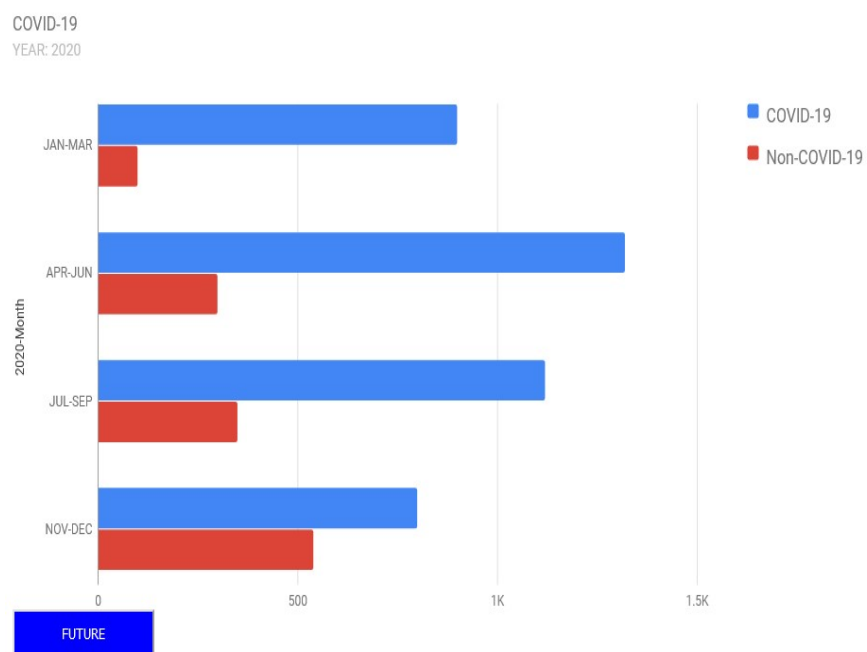




**Fig 7.6 CT-scan negative report**



**Fig 7.6.1 CT-Scan positive report**



**Fig 7.7 Analysis**



## **CHAPTER 8**

### **CONCLUSION AND FUTURE SCOPE**

#### **8.1 CONCLUSION**

The introduced conceptual structures and platforms in the research field of AI-based techniques, which are suitable for dealing with COVID-19 issues, have been studied in this paper. Different techniques have been developed, incorporating COVID-19's diagnostic systems, such as RNN, LSTM, GAN, and ELM. The geographical issues, high-risk people, and recognizing and radiology were the main problems with COVID-19 and have been studied and discussed in this work. Also, we showed a mechanism for selecting the appropriate models of estimation and prediction of desired parameters using a number of clinical and non-clinical datasets. Considering these platforms assists AI experts to analyze huge datasets and help physicians train machines, set algorithms or optimize the analyzed data for dealing with the virus with more speed and accuracy. We discussed that they are desirable because of their potential for creating a workspace while AI experts and physicians could work side by side. However, it should be noted while AI speeds up the methods to conquer COVID-19, real experiments should happen because a full understanding of advantages and limitations of AI-based methods for COVID-19 is yet to be achieved, and novel approaches have to be in place for problems of this level of complexity. Succeeding in the combat against COVID-19 toward its eventual demise is highly dependent on building an arsenal of platforms, methods, approaches, and tools that converge to achieve the sought goals and realize saving more lives.

## 8.2 FUTURE SCOPE

Right now we are using only image data (i.e., X-rays and CT) —so in future it should leverage multiple data sources not limited to just images, including patient vitals, population density etc. Image data by itself is typically not sufficient for these types of applications.

We will need more data at various stages, with the COVID-19 X-rays taken when the patients present several symptoms. More data gives more accuracy. Here I used only few data for this software testing. It gives now 97% accuracy but in future it will also give 99% accuracy if we used thousands of data because Adience dataset has a privileged to automatically check the images and finds the difference. So it gives more accurate output.

I will try to add some more section like MRI etc to detect COVID disease. Adding more section gives more privileged to get result on time. If we used only x-ray and CT and it gives too much headache to the doctors as well as radiologists because we know that in our country, there is very few hospitals and labs. So if we used so many images like PET, OMI or MRI then it will help us to find covid positive patients too earlier and save their life.

I will try to improve the Accuracy using different model like HSGO model (Hybrid-Social-Group-Optimization-algorithm), Inceptionv3, DenseNet121 etc. Here I used CNN model to find covid positive and negative person but CNN models take more time rather than HSGO.

I will try to show the probability and effected part of chest also because if we know the probability and the effected part of chest then we can easily start the treatment and easily resolved the issues.

## REFERENCES

- [1] **Mohammad (Behdad) Jamshidi et al.**: “Artificial Intelligence and COVID-19: Deep Learning Approaches for Diagnosis and Treatment”,IEEE Access 2020,DOI 2020.3001973
- [2] **N. Chen, M. Zhou, X. Dong, J. Qu, F. Gong,Y. Han,Y. Qiu, J.Wang,Y. Liu, Y. Wei, J. Xia, T. Yu, X. Zhang, and L. Zhang**, ``Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: A descriptive study," *Lancet*, vol. 395, no. 10223, pp. 507513, Feb. 2020.
- [3] **D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, and Y. Zhao**, ``Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirusinfected pneumonia in Wuhan, China," *Jama*, vol. 323, no. 11, pp. 10611069, 2020.
- [4] **K. Liu, Y.-Y. Fang, Y. Deng, W. Liu, M.-F. Wang, J.-P. Ma, W. Xiao, Y.-N. Wang, M.-H. Zhong, C.-H. Li, G.-C. Li, and H.-G. Liu**, ``Clinical characteristics of novel coronavirus cases in tertiary hospitals in Hubei province," *Chin. Med. J.*, vol. 133, no. 9, pp. 10251031, May 2020.
- [5] **T. Guo et al.**, ``Cardiovascular implications of fatal outcomes of patients with coronavirus disease 2019 (COVID-19)," *JAMA Cardiol.*, early access, Mar. 27, 2020, doi: 10.1001/jamacardio.2020.1017.
- [6] **P. Hamet and J. Tremblay**, ``Artificial intelligence in medicine," *Metabolism*, vol. 69, pp. S36S40, Apr. 2017.
- [7] **M. Jamshidi, A. Lalbakhsh, S. Lot, H. Siahkamari, B. Mohamadzade, and J. Jalilian**, ``A neuro-based approach to designing a Wilkinson power divider," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 30, no. 3, Mar. 2020, Art. no. e22091.

[8] **M. Jamshidi, A. Lalbakhsh, B. Mohamadzade, H. Siahkamari, and S. M. H. Mousavi**, "A novel neural-based approach for design of microstrip lters," *AEU-Int. J. Electron. Commun.*, vol. 110, Oct. 2019, Art. no. 152847.

[9] **M. B. Jamshidi, N. Alibeigi, A. Lalbakhsh, and S. Roshani**, "An ANFIS approach to modeling a small satellite power source of NASA," in *Proc. IEEE 16th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2019, pp. 459464.