

Covid-19 detection using deep learning with X-ray

Submitted in partial fulfillment of the requirements for the reward of
Bachelor of Engineering degree in Computer Science and Engineering

By

M. PRAVEEN KUMAR(38110320)

M. SANJAY (38110310)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU.**

MARCH 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119
www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of
M. PRAVEEN KUMAR (38110320) , M. SANJAY (38110310)
who underwent the professional training in **“Covid-19 detection using deep
learning with X-ray”** under our supervision from Nov 2021 to March 2022.

Internal Guide

Mrs.M.s.Roobini,M.E., Ph.D.

Head of the Department

Dr.S.VIGNESHWARI.,M.E.,(Ph.D)

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I,M.PRAVEEN KUMAR(38110320), M. SANJAY(38110310)

hereby declare that the Professional Training Report on **“Covid-19 detection using deep learning with X-ray”** done by me under the guidance of **Mrs.M.s.Roobini,M.E., Ph.D.** at Sathyabama university is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in computer science and engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr .T. SASIKALA.,M.E.,Ph.D., Dean, School of Computing** and **Dr.S.VIGNESHWARI.,M.E.,(Ph.D) Head of the Department, Dept of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs.M.s.Roobini,M.E., Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-Teaching staff members of the Department of Computer science and Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Coronavirus disease 2019 also known as COVID-19 has become a pandemic. The disease is caused by a beta coronavirus called Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). The severity of the disease can be understood by the massive number of deaths and affected patients globally. If the diagnosis is fast-paced, the disease can be controlled in a better manner. Laboratory tests are available for diagnosis, but they are bounded by available testing kits and time. The use of radiological examinations that comprise Computed Tomography (CT) can be used for the diagnosis of the disease. Specifically, chest X-Ray images can be analysed to identify the presence of COVID-19 in a patient. In this paper, an automated method for the diagnosis of COVID-19 from the chest X-Ray images is proposed. The method presents an improved depth wise convolution neural network for analysing the chest X-Ray images. Wavelet decomposition is applied to integrate multiresolution analysis in the network. The frequency sub-bands obtained from the input images are fed in the network for identifying the disease. The network is designed to predict the class of the input image as normal, viral pneumonia, and COVID-19. A comparative study with the existing methods is also performed. The metrics like accuracy, sensitivity, and F1-measure are calculated for performance evaluation. The performance of the proposed method is better than the existing methodologies and thus can be used for the effective diagnosis of the disease.

TABLE

CHAPTER NO NO	TITLE	PAGE
1.	INTRODUCTION	1
	1.1 BACKGROUND AND BASICS	1
	1.2 EXISTING SYSTEM	1
	1.3 PROPOSED SYSTEM	2
	1.4 ADVANTAGES OF PROPOSED SYSTEM	2
2.	LITERATURE SURVEY	3
3.	METHODOLOGY	5
	3.1 SYSTEM REQUIRMENT SPECIFICATION	5
	3.1.1 HARDWARE REQUIREMENTS	5
	3.1.2 SOFTWARE REQUIREMENTS	5
	3.2 SOFTWARE ENVIRONMENT	5
	3.3 ARCHITECTURE	15
	3.4 MODULES	16
	3.5 DATAFLOW DIAGRAM	17
	3.6 UML DIAGRAM	20
	3.7 FEASIBILITY STUDY	23
	3.8 SYSTEM DESIGN AND TESTING PLAN	24
4.	RESULT AND DISCUSSION	29
5.	CONCLUSION	30

FIGURE NO	LIST OF FIGURES TITLE	PAGE NO
3.3.1	ARCHITETURE	15
3.5.1	DATAFLOW DIAGRAM 1	18
3.5.2	DATAFLOW DIAGRAM 2	18
3.5.3	DATAFLOW DIAGRAM 3	19
3.5.4	DATAFLOW DIAGRAM 4	19
3.6.1	USECASE DIAGRAM	21
3.6.2	SEQUENCE DIAGRAM	22
3.6.3	ACTIVITY DIAGRAM	23

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND AND BASICS

A pandemic is an outbreak of a disease globally affecting many populations. The world has witnessed many pandemics in the 20th century. The latest coronavirus disease known as COVID-19 has appeared and spread extremely fast. COVID-19 infection are observed to have serious pneumonia with abnormal observations on chest Computed Tomography (CT) examination. The unavailability of medicine for this disease requires efficient diagnosis methods for controlling the disease. A deep learning-based model is applied on CT images for detection of COVID-19. Some researchers have also developed public datasets comprising of chest X-Ray images of COVID-19 patients. A method named COVID-Net is developed and applied on these public datasets for diagnosis of COVID-19. The use of deep learning for diagnosis from the chest X-Ray images provides good results. Deep learning models are being widely used for medical image processing. The detection of pneumonia is done using convolution neural networks. In this paper, an automated method for the diagnosis of COVID-19 from a deep network is proposed. The proposed network utilizes the feature generated by multiresolution analysis. The combination of wavelet transforms along with the deep network brings multiple advantages. The wavelet decomposition is fed into the network. The network used is not the traditional Convolutional Neural Network (CNN). A depthwise separable network is utilized in this work.

1.2 EXISTING SYSTEM

The person administering the test will twirl a long stick with a soft brush on the end, similar to a pipe cleaner, up your nose for a few seconds. There will be a collection of secretions collected by the soft bristles for study. To get a good specimen, the swab must go back a long way, so cells and fluids must be extracted from the whole passageway that links the base of the nose to the back of the neck. However, since the body isn't used to having an object in that place, it produces a slew of strange sensations. For one thing, it activates the lachrymal reflex, which means it will bring tears to your eyes if it's done correctly. I wouldn't go so far as to say it hurt, but it is uncomfortable. Since the swab will also touch the back of the throat, it may also trigger a gag reflex.

1.3 PROPOSED SYSTEM

We have proposed an automatic prediction of covid-19 using a deep convolutional neural networks and Chest X-ray images. The proposed models will have end-to- end structure without manual feature extraction and selection methods. Chest Xray is the best tool for the detection of covid-19. The method presents an improved depth wise convolution neural network for analysing the chest X-Ray images. Wavelet decomposition is applied to integrate multiresolution analysis in the network. The frequency sub-bands obtained from the input images are fed in the network for identifying the disease. The network is designed to predict the class of the input image as normal, viral pneumonia, and COVID-19. A comparative study with the existing methods is also performed. The metrics like accuracy, sensitivity, and F1-measure are calculated for performance evaluation. The performance of the proposed method is better than the existing methodologies and thus can be used for the effective diagnosis of the disease.

1.4 ADVANTAGES OF PROPOSED SYSTEM

1. Pneumonia Detection
2. Lung Disease Detection

CHAPTER 2

LITERATURE SURVEY

- Narinder Singh Punj et al.[2]The flare-up of intense respiratory condition, known as COVID-2019 (SARS-CoV-2), has compromised the whole world. The world is giving a valiant effort to battle the spread of this dangerous infection in framework, finance, news markets, insurance, medical care and numerous different apparatuses. Craftsmanship scientists have zeroed in on their insight into numerical advancement to break down the condition of the scourge, utilizing information from the nation over.
- Palash Ghosh et al.[3]The sickness has spread to 212 nations and areas and has tainted (affirmed) multiple million individuals. In India, the infection was first given an account of January 30, 2020 to an understudy getting back from Wuhan in Kerala. Across India (as of May 3, 2020), the quantity of individuals tainted is north of 37,000. Most investigations and diaries center around the quantity of tainted individuals the nation over.
- Sina Ardabili et al[4].In the normal instances of foreseeing the worldwide COVID-19 plague, the epidemiological and measurable epidemiological model has been important to the specialists and notable in the media. Because of the great degree of vulnerability and absence of significant data, the standard models are not satisfactory over the long haul.
- Fotio Petropoulos, et al[5].Nonetheless, prediction requires adequate data. Simultaneously, what's to come isn't so unsurprising as it used to be. What's more, the speculation depends on the believability of the news media and the anticipated changes. Mental factors additionally assume a significant part in how individuals see and react with the impacts of the sickness and what it can mean for them.

- Arti. M. K, K. Bhatnagar et al[6].Nonetheless, prediction requires adequate data. Simultaneously, it is obscure right now what he will do subsequent to leaving the post. Moreover, the theory depends on the believability of the news media and the anticipated changes. Mental factors likewise assume a significant part in how individuals see and react to the danger of illness, and they dread that it could influence them.
- H.Shekhar et al[7] .The information from March 15, 2020 to April 30, 2020 is utilized to approve the model, and the interior speed is steady. In some Indian nations, for example, Maharashtra, Gujarat and Delhi, individuals are tainted every day. The underlying disease is supposed to be on the ascent and is relied upon to keep on developing. Unexpected blasts are characterized by contrasts in the way of life of these three nations.
- Yan, L., Zhang et al [8].The abrupt ascent in COVID-19 is putting a strain on worldwide wellbeing administrations. At this stage, it is critical to analyze the illness rapidly, precisely and early. To help independent direction and arranging in wellbeing programs, the review utilized blood tests from 485 tainted patients in Wuhan, China, to identify indications of death from the sickness.
- L. Li, Z. Yang et al [9].As indicated by the authority data transmission framework, this report inspects how the 2019 Coronavirus (COVID-19) is communicated. The mistake between the model and the information bend is tiny. At that point, he anticipated the plague and comprehended it prior to settling on a choice and dissecting the nations that aided settle on the choice.

CHAPTER 3

METHODOLOGY

3.1 System Requirement Specification

3.1.1 HARDWARE REQUIREMENTS:

System - Windows7/10
Speed - 2.4GHZ
Hard disk - 40GB
Monitor - 15VGA Color
Ram - 4GB

3.1.2 SOFTWARE REQUIREMENTS:

Operating system - Windows 7
Coding Language - PYTHON

3.2 SOFTWARE ENVIRONMENT

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++,

Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

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.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python
Python2.4.3(#1,Nov112010,13:34:43)
[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

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 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	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.

5	DELETE 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>

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

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="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**.

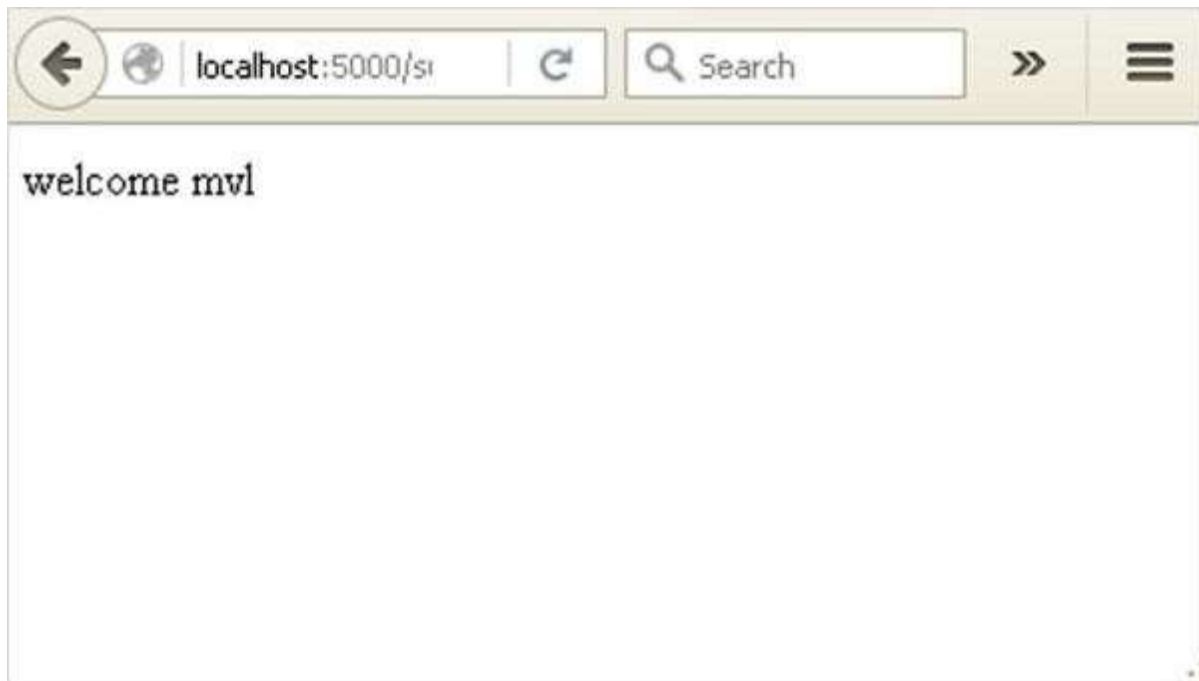
A screenshot of a web browser window. The address bar shows the file path 'file:///C:/login.ht'. The main content area displays a simple login form with the label 'Enter Name:', a text input field containing the text 'mvl', and a 'submit' button below it.

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.



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.

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have python installed on your computer, then you can download it for free from the following website: <https://www.python.org/>

Python Quickstart

Python is an interpreted programming language, this means that as a developer you write Python (.py) files in a text editor and then put those files into the python interpreter to be executed.

The way to run a python file is like this on the command line:

```
C:\Users\Your Name>python helloworld.py
```

Where "helloworld.py" is the name of your python file.

Let's write our first Python file, called helloworld.py, which can be done in any text editor.

```
helloworld.py
```

```
print("Hello, World!")
```

Simple as that. Save your file. Open your command line, navigate to the directory where you saved your file, and run:

```
C:\Users\Your Name>python helloworld.py
```

The output should read:

```
Hello, World!
```

Congratulations, you have written and executed your first Python program.

The Python Command Line

To test a short amount of code in python sometimes it is quickest and easiest not to write the code in a file. This is made possible because Python can be run as a command line itself.

Type the following on the Windows, Mac or Linux command line:

```
C:\Users\Your Name>python
```

From there you can write any python, including our hello world example from earlier in the tutorial:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
```

Which will write "Hello, World!" in the command line:

```
C:\Users\Your Name>python
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit
(Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
```

Whenever you are done in the python command line, you can simply type the following to quit the python command line interface:

```
exit()
```

Execute Python Syntax

As we learned in the previous page, Python syntax can be executed by writing directly in the Command Line:

```
>>> print("Hello, World!")
Hello, World!
```

Or by creating a python file on the server, using the .py file extension, and running it in the Command Line:

```
C:\Users\Your Name>python myfile.py
```

Python Indentations

Where in other programming languages the indentation in code is for readability only, in Python the indentation is very important.

Python uses indentation to indicate a block of code.

Example

```
if 5 > 2:
    print("Five is greater than two!")
```

Python will give you an error if you skip the indentation:

Example

```
if 5 > 2:
print("Five is greater than two!")
```

Comments

Python has commenting capability for the purpose of in-code documentation.

Comments start with a #, and Python will render the rest of the line as a comment:

Example

Comments in Python:

```
#This is a comment.
print("Hello, World!")
```

Docstrings

Python also has extended documentation capability, called docstrings.

Docstrings can be one line, or multiline.

Python uses triple quotes at the beginning and end of the docstring:

Example

Docstrings are also comments:

```
"""This is a
multiline docstring."""
print("Hello, World!")
```

3.4 MODULES

- Data collection Module
- Data Pre-Processing Module
- Training the CNN Module
- Prediction Module

MODULE DESCRIPTION

Data collection Module

This experiment leveraging the data from Covid Chest XRay Dataset and Pneumonia dataset by Praveen from GitHub and Kaggle respectively. Using these two datasets, we have created a customized dataset as normal, viral pneumonia, and COVID-19. The COVID-19 dataset comprises of the chest X-Ray images. These images are used for the detection. The images are of different sizes, thus they are resized.

Data Pre-Processing Module

The input images are normalized prior to any further processing. Normalized images are enhanced images with no errors due to lightening conditions. This segment mentioned the technique of preparing the raw data to make these suitable for building and training machine learning models. We used 3 repositories for data set preparation. 2 sets from GitHub and another from kaggle. 2 directories make for COVID19 and normal data set. Data set image will be classified to normal and COVID-19.

Training the CNN Module

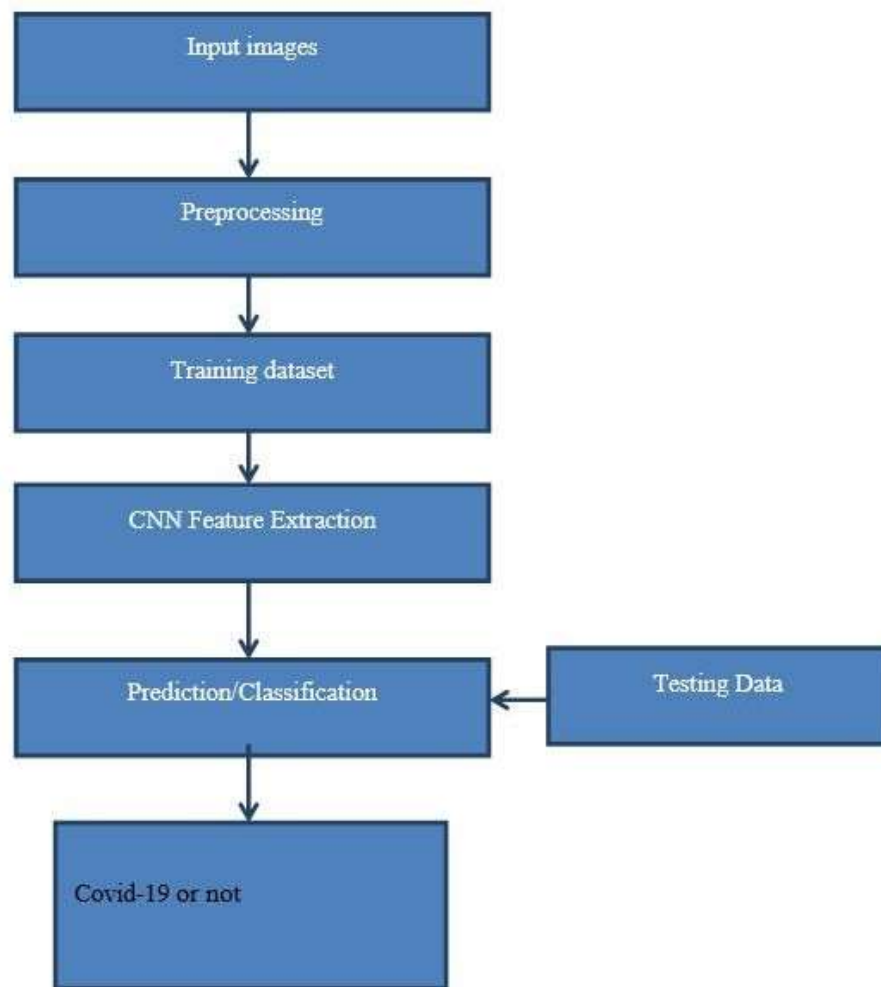
Before we thought of training the CNN, we will load the data and target. Before training the model, we will set up data generators to read images from source folders instead of labeling the image one by one. Basically, Image DataGenerator will label the images based on the directory the image is contained. It points to the sub-directory of the data. Grayscale normalization should be done in order for CNN to converge faster by using rescale parameter.

Prediction Module

Finally, any given input chest X-Ray image is classified into one of three classes, i.e., normal, COVID-19, or viral pneumonia.

3.5 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



UML DIAGRAMS

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.

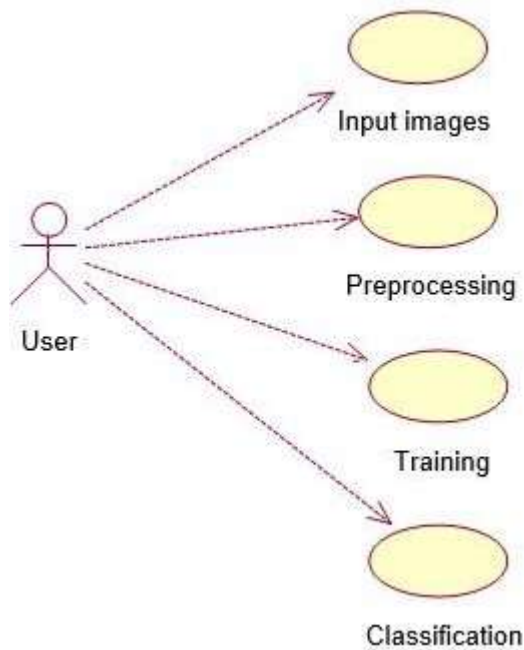
GOALS:

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 extendibility 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.

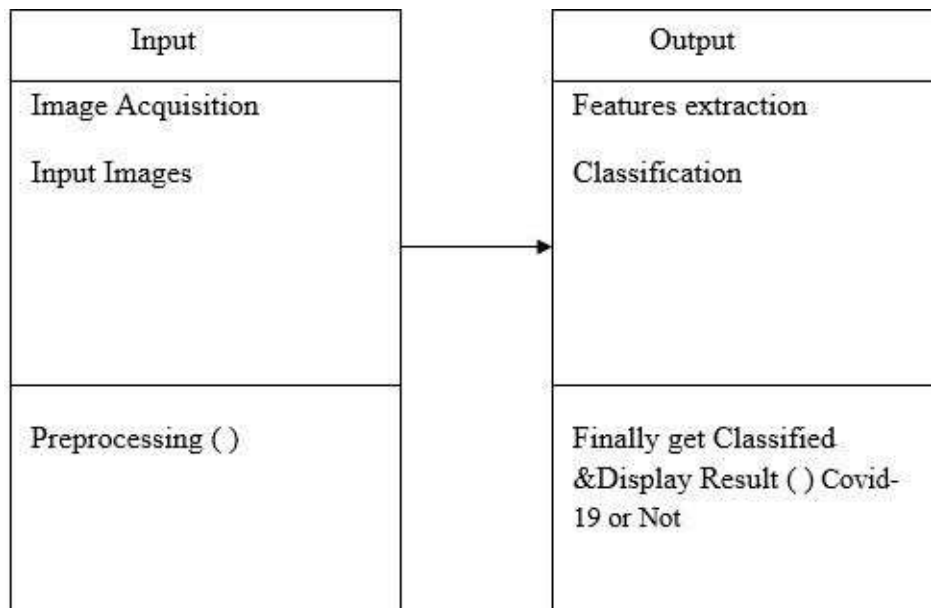
USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



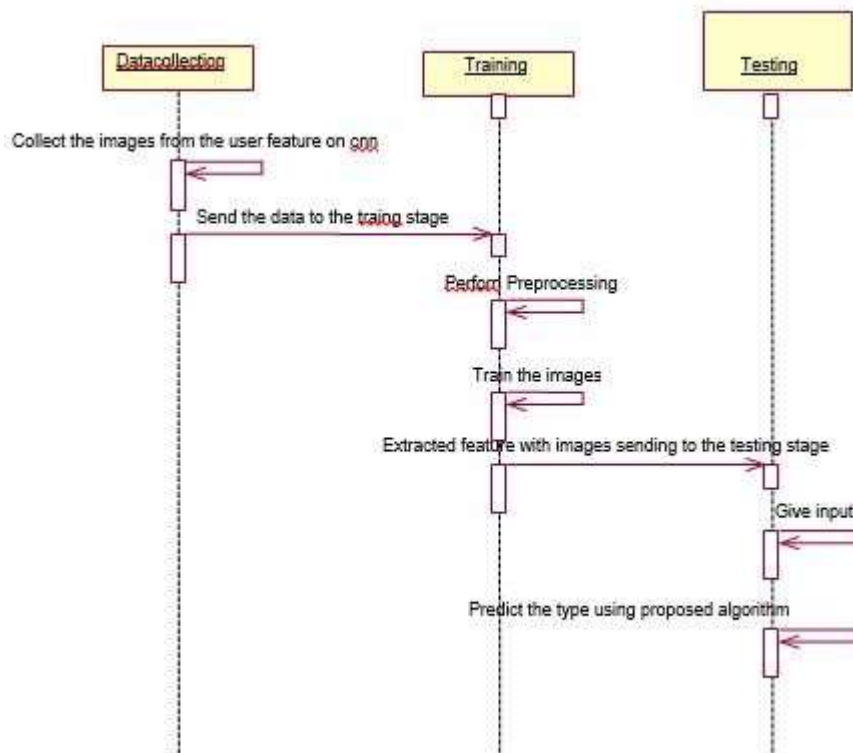
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



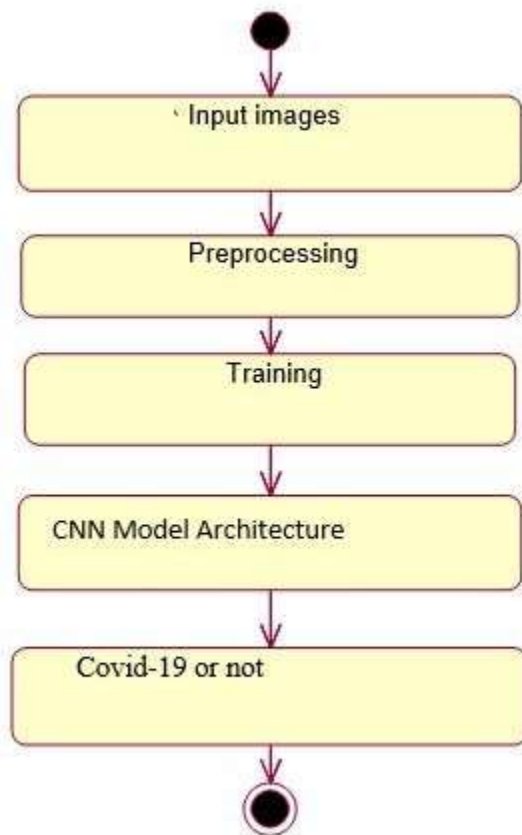
SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ ECONOMICAL FEASIBILITY
- ◆ TECHNICAL FEASIBILITY
- ◆ SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be

justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, 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.

6.2 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.

6.3 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.

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

CONCLUSION

The conclusion of the paper is that CNN model Architecture which is used in our proposed system to predict the pancreatic cancer performs well. The system classifies the image to normal and Covid-19 one. The accuracy of model is 100%. This system provided a proof of concept that CNN can accurately distinguish pancreatic cancer on CT images. The CNN model holds promise as a computer-aided diagnostic tool to assist radiologists and clinicians in diagnosing covid-19.

SOURCE CODE

```
#!/usr/bin/env python

import os

import sys


from flask import Flask, request, jsonify, send_file, render_template

from io import BytesIO

from PIL import Image, ImageOps

import base64

import urllib


import numpy as np

import scipy.misc

from tensorflow.keras.preprocessing import image

from tensorflow.keras.models import load_model

import os

import tensorflow as tf

import numpy as np

from tensorflow import keras

#from skimage import io
```

```
from tensorflow.keras.preprocessing import image
```

```
# Flask utils
```

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

```
from werkzeug.utils import secure_filename
```

```
from gevent.pywsgi import WSGIServer
```

```
from tensorflow.keras.models import load_model
```

```
import efficientnet.keras as efn
```

```
import efficientnet.tfkeras
```

```
from tensorflow.keras.models import load_model
```

```
app = Flask(__name__)
```

```
# Load your trained model
```

```
@app.route("/")
```

```
@app.route("/first")
```

```
def first():
```

```
    return render_template('first.html')
```

```
@app.route("/login")
```

```
def login():
```

```
    return render_template('login.html')
```

```
@app.route("/chart")
```

```
def chart():
```

```
    return render_template('chart.html')
```

```
@app.route("/performance")
```

```
def performance():
```

```
    return render_template('performance.html')
```

```
@app.route("/index",methods=['GET'])
```

```
def index():
```

```
    return render_template('index.html')
```

```

@app.route("/upload", methods=['POST'])

def upload_file():

    print("Hello")

    try:

        img = request.files['imagefile'].read()

        img = Image.open(BytesIO(img)).convert('RGB')

        img = ImageOps.fit(img, (224, 224), Image.ANTIALIAS)

    except:

        error_msg = "Please choose an image file!"

        return render_template('index.html', **locals())

# Call Function to predict

args = {'input' : img}

out_pred, out_prob = predict(args)

out_prob = out_prob * 100

print(out_pred, out_prob)

danger = "danger"

```

```

if out_pred=="You Are Safe, But Do keep precaution":

    danger = "success"

print(danger)

img_io = BytesIO()

img.save(img_io, 'PNG')


png_output = base64.b64encode(img_io.getvalue())

processed_file = urllib.parse.quote(png_output)


return render_template('result.html',**locals())

def predict(args):

    img = np.array(args['input']) / 255.0

    img = np.expand_dims(img, axis = 0)


    model = 'save.h5'

    # Load weights into the new model

    model = load_model(model)


    pred = model.predict(img)

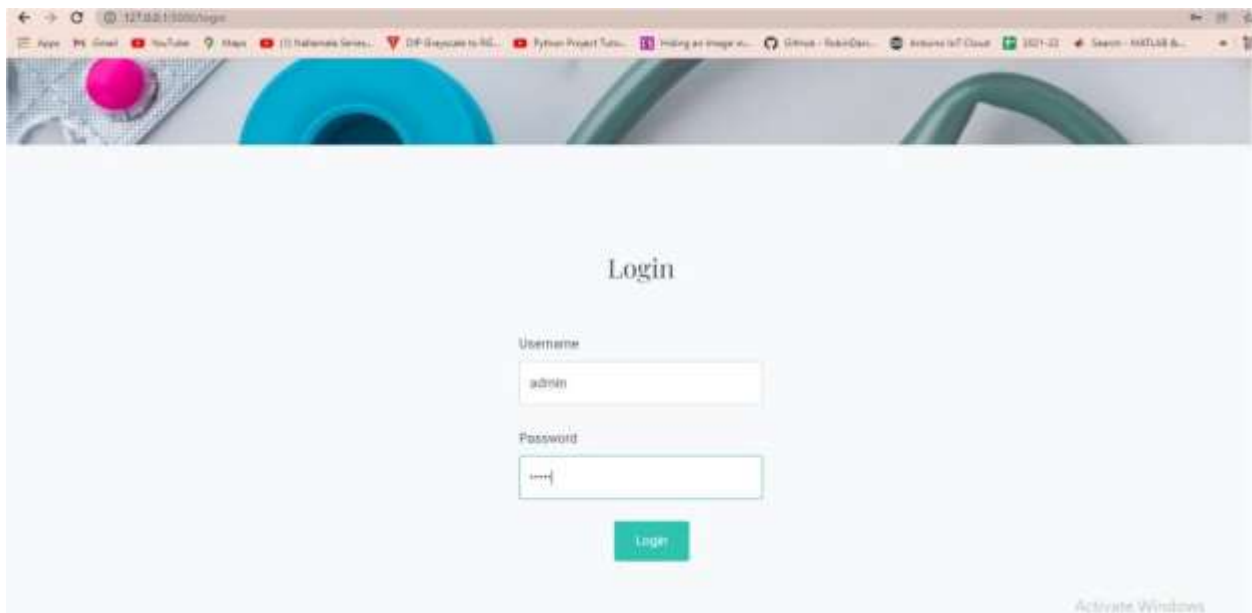
```

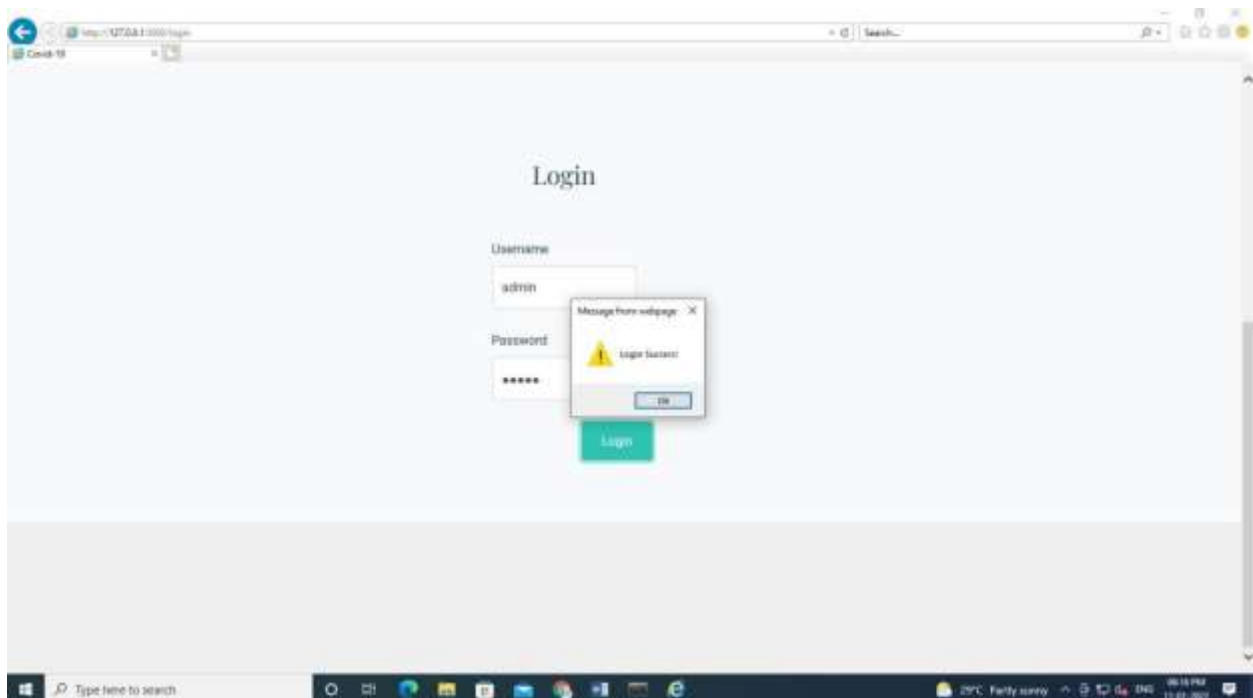
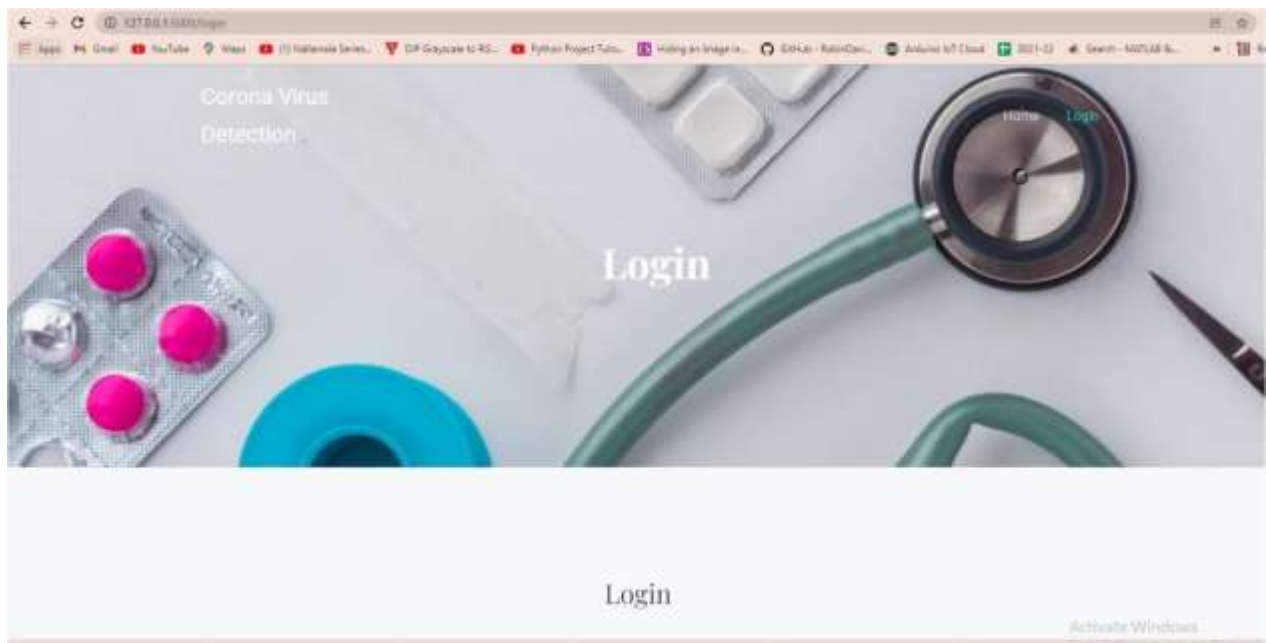
```
if np.argmax(pred, axis=1)[0] == 0:  
    out_pred = "Normal"  
elif np.argmax(pred, axis=1)[0] == 1:  
    out_pred = "CORONA-VIRUS AFFECTED"
```

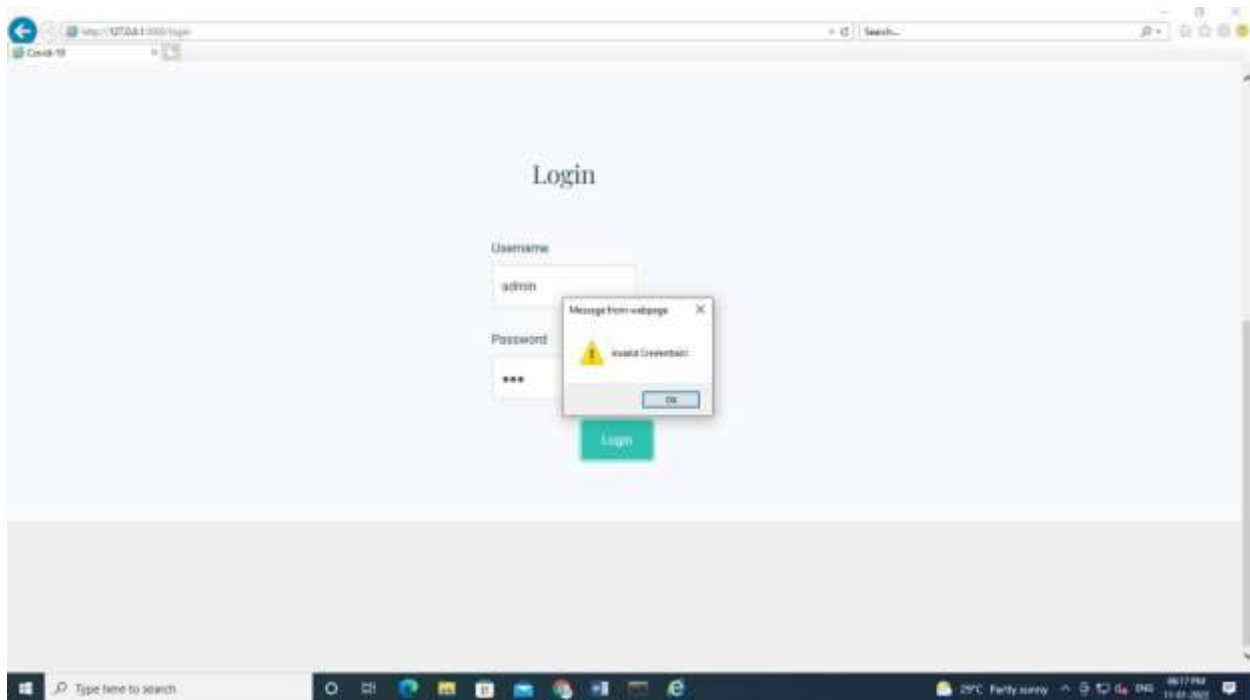
```
return out_pred, float(np.max(pred))
```

```
if __name__ == '__main__':  
    app.run(debug=True)
```

SCREEN SHOTS:









REFERENCES

- [1] Features, Evaluation and Treatment Coronavirus (COVID-19) . Available: <https://www.ncbi.nlm.nih.gov/books/NBK554776/>, May 18 (2020).
- [2] N. S Pun, S. K. Sonbhadra, S. Agarwal, “COVID-19 Epidemic Analysis using Machine Learning and Deep Learning Algorithms”, medRxiv, Available: June 1 (2020).
- [3] P. Ghosh, R. Ghosh, B. Chakraborty, “COVID-19 in India: State-wise Analysis and Prediction”, medRxiv, Available: doi: <https://doi.org/10.1101/2020.04.24.20077792>, May 19 (2020).
- [4] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. VarkonyiKoczy, U.Reuter, T. Rabczuk, P. M. Atkinson, “COVID-19 Outbreak Prediction with

Machine Learning”, Available at SSRN: <https://ssrn.com/abstract=3580188> or <http://dx.doi.org/10.2139/ssrn.3580188>, April 19 (2020).

[5] F. Petropoulos, S. Makridakis, “Forecasting the novel coronavirus COVID-19”, Available: <https://doi.org/10.1371/journal.pone.0231236>, Published: March 31 (2020)

[6] Arti. M. K, K. Bhatnagar, “Modeling and Predictions for COVID 19 Spread in India”, Available:, Published: April (2020).

[7] H. Shekhar, “Prediction of Spreads of COVID-19 in India from Current Trend”, medRxiv, Available: doi:, May 06, (2020).

[8] Yan, L., Zhang, H., Goncalves, J. et al. “An interpretable mortality prediction model for COVID-19 patients”. *Nature Machine Intelligence* 2, pp. 283–288, Published: 14 May (2020).

[9] L. Li, Z. Yang, Z. Dang, C. Meng, J. Huang, H. Meng, D. Wang, G. Chen, J. Zhang, H. Peng, Y. Shao, “Propagation analysis and prediction of the COVID-19”, *Infectious Disease Modelling* vol. 5, Available:

[10] S. Zhao, H. Chen, “Modeling the epidemic dynamics and control of COVID-19 outbreak in China”. *Quantitative Biology*, vol. 8, Issue March (2020).

[11] J. Xie, Z. Tong, X. Guan. et al. “Critical care crisis and some recommendations during the COVID-19 epidemic in China”. *Intensive Care Med*, vol.6 Issue 6, Available:, June (2020).

