PNEUMONIA DETECTION USING CNN

1.INTRODUCTION

Overview:

In general, a patient suffering from Pneumonia goes to the hospital to take an X-ray image waits for the doctor and then the doctor will check the X-ray then he decides whether the person has pneumonia or not. The results are not only concluded based on just seeing the X-ray images but furthermore, tests were conducted on the patient to verify the results of the doctor. The process is time-consuming and if the patient has severe pneumonia or not he has to wait several days to get the test results. But in recent developments of the artificial intelligence and the computational powers of the computers have increased it helps in predicting pneumonia by just passing the X-ray image as an input to our model.

Purpose:

The main objective of this project is to help the doctors to predict the pneumonia disease more accurately using a deep learning model. The objective is not only to help the doctors but also to the patients to verify whether they have pneumonia or not. By using this model we can precisely predict pneumonia.

2. Literature Survey

Existing Problem: Trained doctors need to be hired to identify whether a person has pneumonia or not by inspection of X-Ray images of chest. This is time and cost consuming.

Proposed Solution: We can create a user friendly software where the patient can uploag his X-Ray image of chest and software will tell whether he has Pneumonia or not.

3. Theoretical Analysis

Block Diagram:

Hardware/Software Designing:

Code for creating model:

Data Preprocessing

In [1]:

```
#importing the liabrary
from keras.preprocessing.image import ImageDataGenerator
```

```
Using TensorFlow backend.
```

In [3]:

```
#configuration
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip
=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [4]:

```
#Apply to dataset
x_train=train_datagen.flow_from_directory(r'E:\Project\dataset\train_set',target_size=(64,64),batch_s
ize=4,class_mode='binary')
x_test=test_datagen.flow_from_directory(r'E:\Project\dataset\test_set',target_size=(64,64),batch_siz
e=4,class_mode='binary')
```

```
Found 408 images belonging to 2 classes.
Found 96 images belonging to 2 classes.
```

In [5]:

```
print(x_train.class_indices)
```

```
{'NORMAL': 0, 'PNEUMONIA': 1}
```

Building the model

In [6]:

```
#import the liabraries
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
```

In [7]:

```
#initialize the model
model=Sequential()
```

In [8]:

```
#add convolution layers
model.add(Convolution2D(32,(3,3),input_shape = (64,64,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

```
WARNING:tensorflow:From
C:\Users\Lenovo\Anaconda3\lib\site-packages\tensorflow_core\python\ops\resource_variable_ops
.py:1630: calling BaseResourceVariable.__init__ (from tensorflow.python.ops.resource_variable_ops)
with constraint is deprecated and will be removed in a future version.
Instructions for updating:
If using Keras pass *_constraint arguments to layers.
WARNING:tensorflow:From
C:\Users\Lenovo\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:4070: The
name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.
```

In [9]:

```
#Adding Dense layers
model.add(Dense(output_dim=64,init='uniform',activation='relu'))
model.add(Dense(output_dim=1,activation='sigmoid',init='uniform'))
```

```
C:\Users\Lenovo\Anaconda3\lib\site-packages\ipykernel_launcher.py:2: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(activation="relu", units=64, kernel_initializer="uniform")`

C:\Users\Lenovo\Anaconda3\lib\site-packages\ipykernel_launcher.py:3: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(activation="sigmoid", units=1, kernel_initializer="uniform")`
 This is separate from the ipykernel package so we can avoid doing imports until
```

In [10]:

```
#Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam',metrics=['accuracy'])
```

```
WARNING:tensorflow:From
C:\Users\Lenovo\Anaconda3\lib\site-packages\tensorflow_core\python\ops\nn_impl.py:183: where
(from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

In [11]:

```
#Train the model
model.fit_generator(x_train,steps_per_epoch=102,
epochs=20,validation_data=x_test,validation_steps=24)
```

```
WARNING:tensorflow:From
C:\Users\Lenovo\Anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:422: The
name tf.global_variables is deprecated. Please use tf.compat.v1.global_variables instead.

Epoch 1/20
102/102 [==============================] - 31s 304ms/step - loss: 0.6660 - accuracy: 0.5637 -
val_loss: 0.6940 - val_accuracy: 0.4479
Epoch 2/20
102/102 [==============================] - 24s 234ms/step - loss: 0.4977 - accuracy: 0.7647 -
val_loss: 1.0676 - val_accuracy: 0.5625
Epoch 3/20
102/102 [==============================] - 24s 238ms/step - loss: 0.3538 - accuracy: 0.8578 -
val_loss: 0.8260 - val_accuracy: 0.6562
Epoch 4/20
102/102 [==============================] - 24s 234ms/step - loss: 0.3513 - accuracy: 0.8456 -
val_loss: 0.8991 - val_accuracy: 0.6250
Epoch 5/20
102/102 [==============================] - 24s 238ms/step - loss: 0.2829 - accuracy: 0.8897 -
val_loss: 1.4944 - val_accuracy: 0.5938
Epoch 6/20
102/102 [==============================] - 24s 236ms/step - loss: 0.3066 - accuracy: 0.8873 -
val_loss: 0.4375 - val_accuracy: 0.5208
Epoch 7/20
102/102 [==============================] - 24s 232ms/step - loss: 0.2361 - accuracy: 0.9044 -
val_loss: 0.5664 - val_accuracy: 0.6458
Epoch 8/20
102/102 [==============================] - 24s 236ms/step - loss: 0.2373 - accuracy: 0.8922 -
val_loss: 0.2722 - val_accuracy: 0.6771
Epoch 9/20
102/102 [==============================] - 24s 238ms/step - loss: 0.2446 - accuracy: 0.8922 -
val_loss: 1.8753 - val_accuracy: 0.5729
Epoch 10/20
102/102 [==============================] - 25s 241ms/step - loss: 0.2437 - accuracy: 0.8971 -
val_loss: 0.2695 - val_accuracy: 0.6875
Epoch 11/20
102/102 [==============================] - 25s 242ms/step - loss: 0.2439 - accuracy: 0.9044 -
val_loss: 1.2844 - val_accuracy: 0.6667
Epoch 12/20
```

```
102/102 [==============================] - 24s 235ms/step - loss: 0.1794 - accuracy: 0.9314 -
val_loss: 1.5520 - val_accuracy: 0.6979
Epoch 13/20
102/102 [==============================] - 24s 238ms/step - loss: 0.1758 - accuracy: 0.9289 -
val_loss: 0.3974 - val_accuracy: 0.7292
Epoch 14/20
102/102 [==============================] - 25s 242ms/step - loss: 0.1850 - accuracy: 0.9314 -
val_loss: 0.5262 - val_accuracy: 0.7188
Epoch 15/20
102/102 [==============================] - 24s 235ms/step - loss: 0.2142 - accuracy: 0.9265 -
val_loss: 0.2487 - val_accuracy: 0.6042
Epoch 16/20
102/102 [==============================] - 25s 240ms/step - loss: 0.1912 - accuracy: 0.9191 -
val_loss: 0.8659 - val_accuracy: 0.6771
Epoch 17/20
102/102 [==============================] - 24s 239ms/step - loss: 0.1923 - accuracy: 0.9167 -
val_loss: 1.5001 - val_accuracy: 0.5938
Epoch 18/20
102/102 [==============================] - 24s 236ms/step - loss: 0.1444 - accuracy: 0.9461 -
val_loss: 0.8107 - val_accuracy: 0.6042
Epoch 19/20
102/102 [==============================] - 24s 236ms/step - loss: 0.1889 - accuracy: 0.9191 -
val_loss: 1.0621 - val_accuracy: 0.7083
Epoch 20/20
102/102 [==============================] - 25s 246ms/step - loss: 0.2077 - accuracy: 0.9191 -
val_loss: 0.3915 - val_accuracy: 0.7083
```

Out[11]:

```
<keras.callbacks.callbacks.History at 0x209c4940388>
```

In [12]:

```
#save model
model.save('PNEUMONIA.h5')
```

Model Testing Code:

```
#import libraries
from keras.models import load_model
from keras.preprocessing import image

#load model
```

```
model=load_model("PNEUMONIA.h5")

img= image.load_img(r"E:\chest_xray\val\PNEUMONIA\img4.jpeg",
target_size=(64,64))

import numpy as np
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)

model.predict_classes(x)
```

Flask Files:

```
import numpy as np
import os
from keras.models import load_model
from keras.preprocessing import image
import tensorflow as tf
from flask import Flask , request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

app = Flask(__name__)
model = load_model("PNEUMONIA.h5")

@app.route('/')
def index():
    return render_template('base.html')

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']
```

```
        basepath = os.path.dirname(__file__)
        print("current path", basepath)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)

        img = image.load_img(filepath,target_size = (64,64))
        x = image.img_to_array(img)
        x = np.expand_dims(x,axis =0)
        preds = model.predict_classes(x)
        index = ['NORMAL','PNEUMONIA']
        text = "The health status is : " + (index[preds[0][0]])

    return text

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

HTML file:

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Pnuemonia Detection System</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
```

```
    <script
src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
        <style>
        .bg-dark {
                background-color: #42678c!important;
        }
        #result {
                color: "green";
        }
        </style>
        <style>
       .vertical1 {
          border-left: 3px solid grey;
          height: 650px;
          position:absolute;
          left: 50%;
       }
       .vertical2 {
          border-left: 3px solid grey;
          height: 650px;
          position:absolute;
          left: 10%;
       }
       .vertical3 {
          border-left: 3px solid grey;
          height: 650px;
          position:absolute;
          left: 90%;
       }
    </style>
</head>
```

```
<body style="background-color:rgb(255,255,125)">
   <nav class="navbar navbar-dark bg-light">
      <div><img src="{{ url_for('static', filename='hospital.png' ) }}" width="80"
height="60"></div>
      <div class="container">
         <a class="navbar-brand" href="#"><font size="10"
color="red">Pneumonia Detection using CNN</font></a>
      </div>
   </nav>
   <div class="vertical1"></div>
   <div class="vertical2"></div>
   <div class="vertical3"></div>
   <div class="container">
      <div id="content" style="margin-top:2em">
            <div class="container">
              <div class="row">

                  <div class="col-sm-6">
                        <div>
                              <h4>Kindly upload <b><font size="6"><u>Chest
X-Ray Image</u></font></b></h4>
                  <form action = "http://localhost:5000/" id="upload-file"
method="post" enctype="multipart/form-data">
                              <label for="imageUpload" class="upload-label">
                                    Select...
                              </label>
                              <input type="file" name="image" id="imageUpload"
accept=".png, .jpg, .jpeg">
                  </form>
```

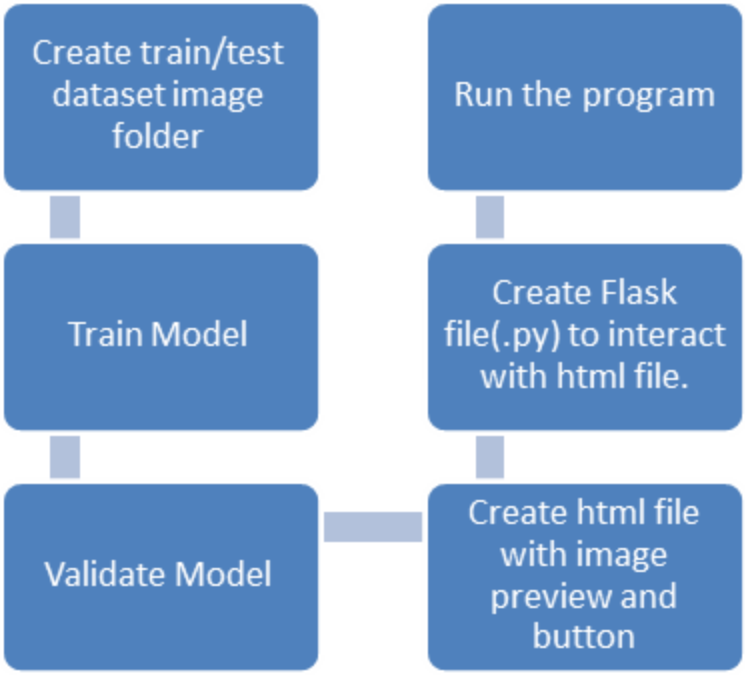
```
                    <div class="image-section" style="display:none;">
                        <div class="img-preview">
                            <div id="imagePreview">
                            </div>
                        </div>
                        <div>
                            <button type="button" class="btn btn-info btn-lg "
id="btn-predict">CHECK!</button>
                        </div>
                    </div>

                    <div class="loader" style="display:none;"></div>

                    <h3>
                        <span id="result"> </span>
                    </h3>
                </div>
                    </div>
                    <div class="col-sm-6 bd" >
                      <h3><font size="6"><u><i>Pneumonia
Detection</i></u></font></h3>
                      <br>
                      <p align="justify"><font size="4"><i>In general, a patient
suffering from Pneumonia goes to the hospital to take an X-ray image waits for the
doctor and then the doctor will check the X-ray then he decides whether the person
has pneumonia or not. The results are not only concluded based on just seeing the
X-ray images but furthermore, tests were conducted on the patient to verify the
results of the doctor. The process is time-consuming and if the patient has severe
pneumonia  or not he has to wait several days to get the test results. But in recent
developments of the artificial intelligence and the computational powers of the
computers have increased it helps in predicting pneumonia by just passing the
X-ray image as an input to our model.
```

```
                    </i></font></p>
          <img src="{{ url_for('static', filename='pneu.png' ) }}" width= "522"
height="174">
                          </div>
                    </div>
                  </div>
                  </div>
   </div>
</body>

<footer>
   <script src="{{ url_for('static', filename='js/main.js') }}"
type="text/javascript"></script>
</footer>

</html>
```

4. Experimental Investigations:

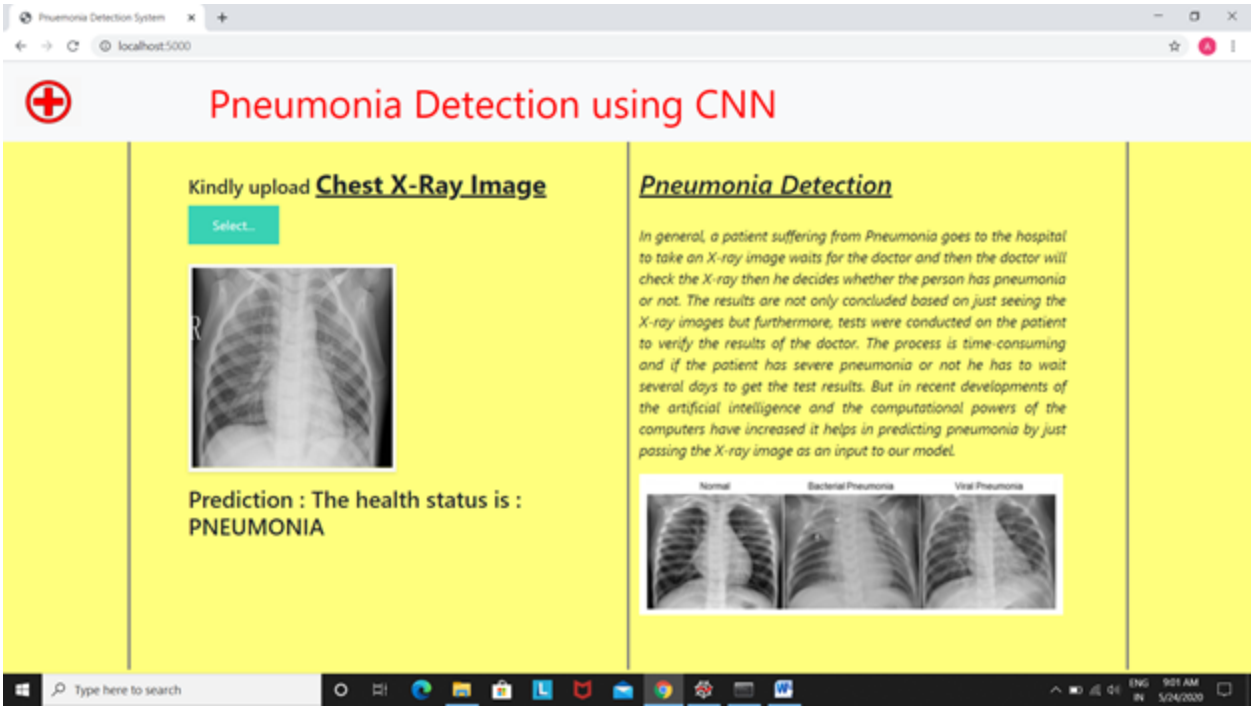
i) Chest X-Ray sample images are available on the internet.
ii) Using tensorflow 2.0 version simplifies code.

5. Flowchart:



6. Result:

See the output screenshot below:



7. Advantages and Disadvantages:

Advantages:
i)Medical services can be given to large number of people.
ii) Cost is less
Disadvantages:
i) Does not provide medical advice in case of Pneumonia positive.

8. Applications:

In public hospitals for Pneumonia verification. Public hospitals have high patient load, especially in India. So that load can be automated.

9. Conclusions:

This process raises the speed and efficiency of checking for Pneumonia.

10. Future Scope:

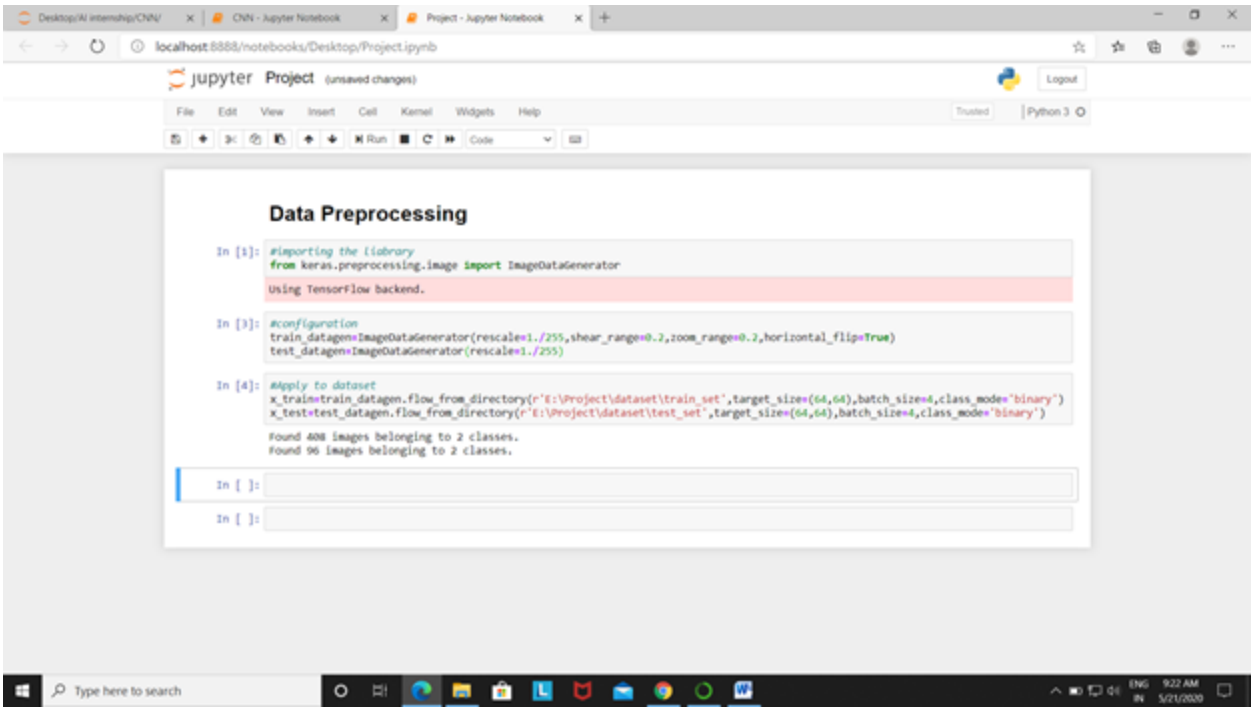
Solution can be extrapolated where diagnosis is to be done using charts, images etc. like ECG or so.

11. Bibilography:

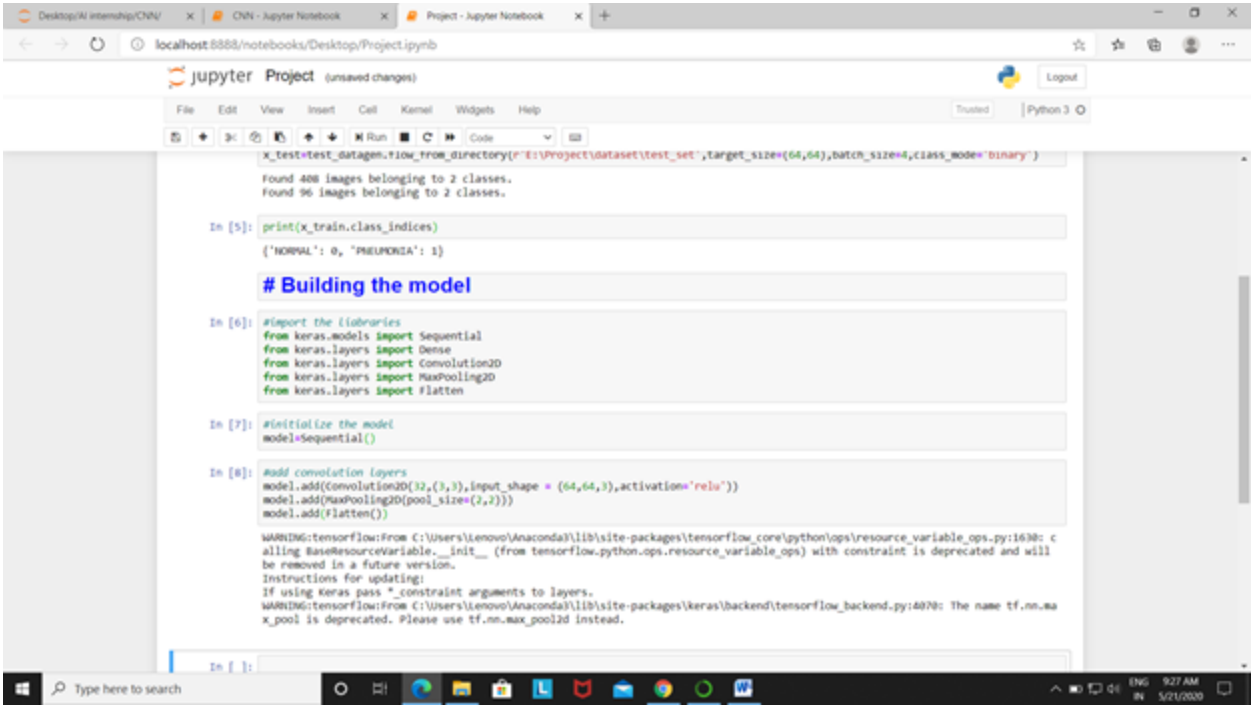
i) Image dataset: kaggle.com
ii)HMTL editing: https://www.w3schools.com/tags/tag_img.asp

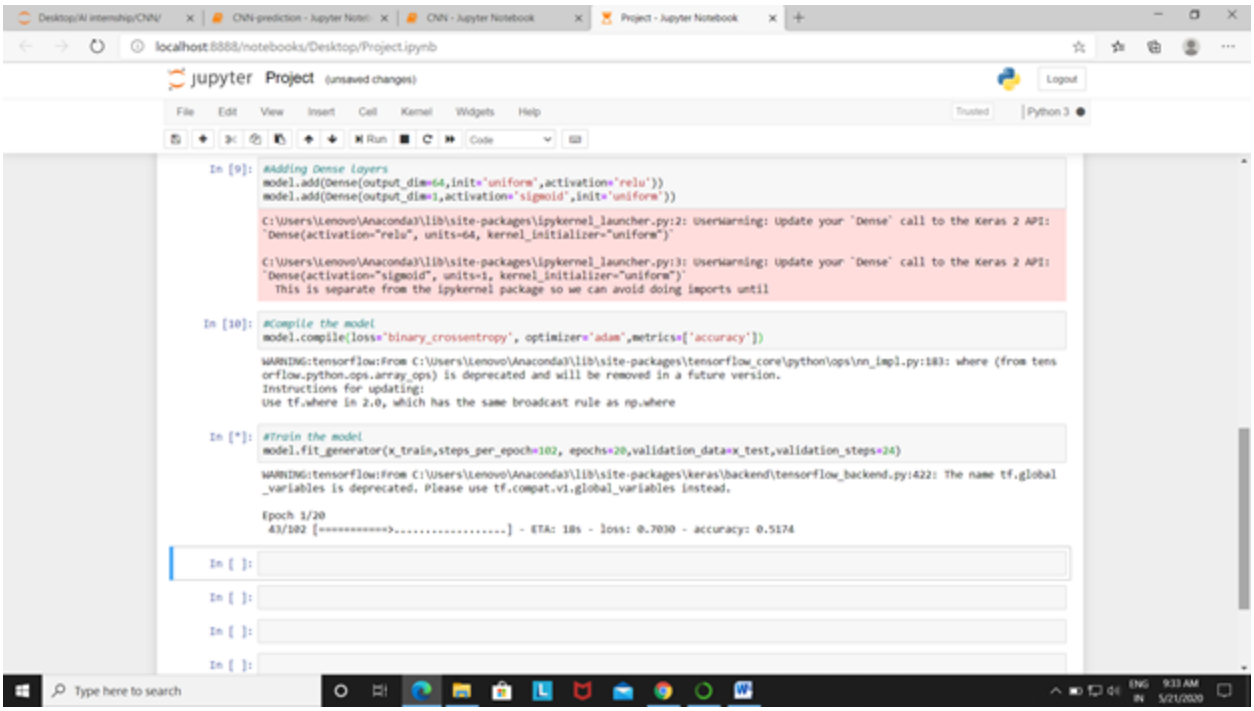
Please find screenshots of step by step program execution:

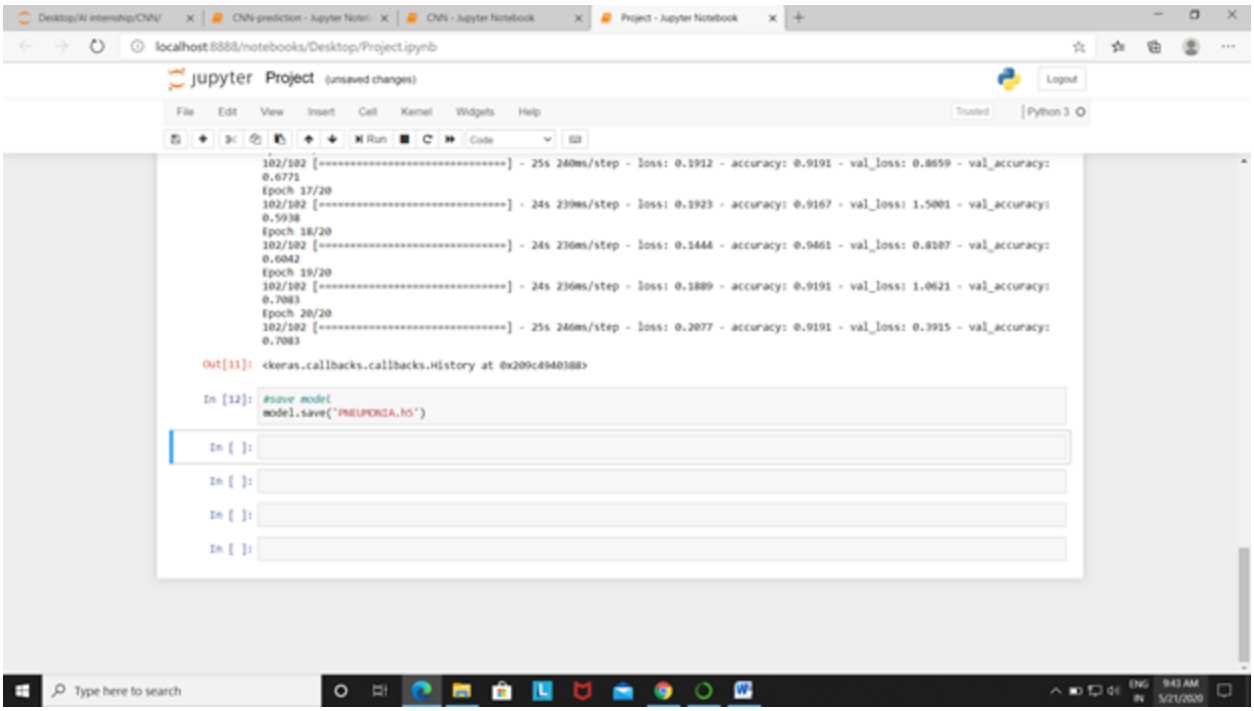
1)Data preprocessing:



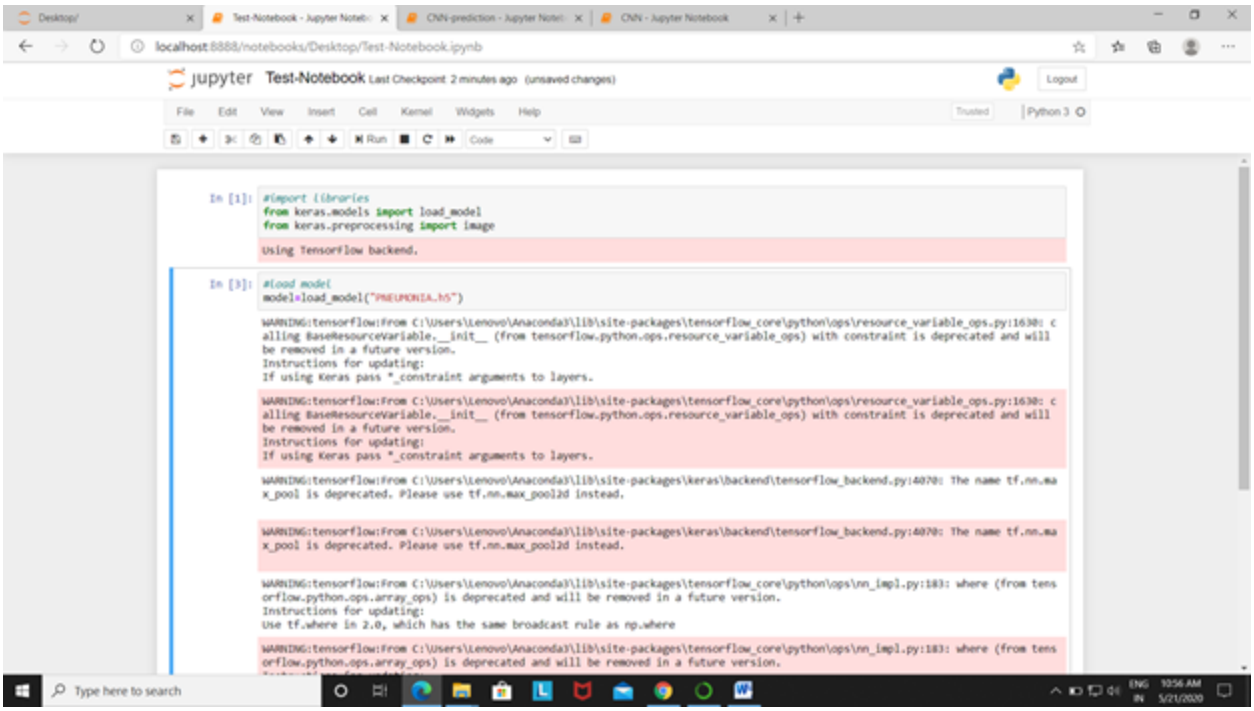
2) Model building:

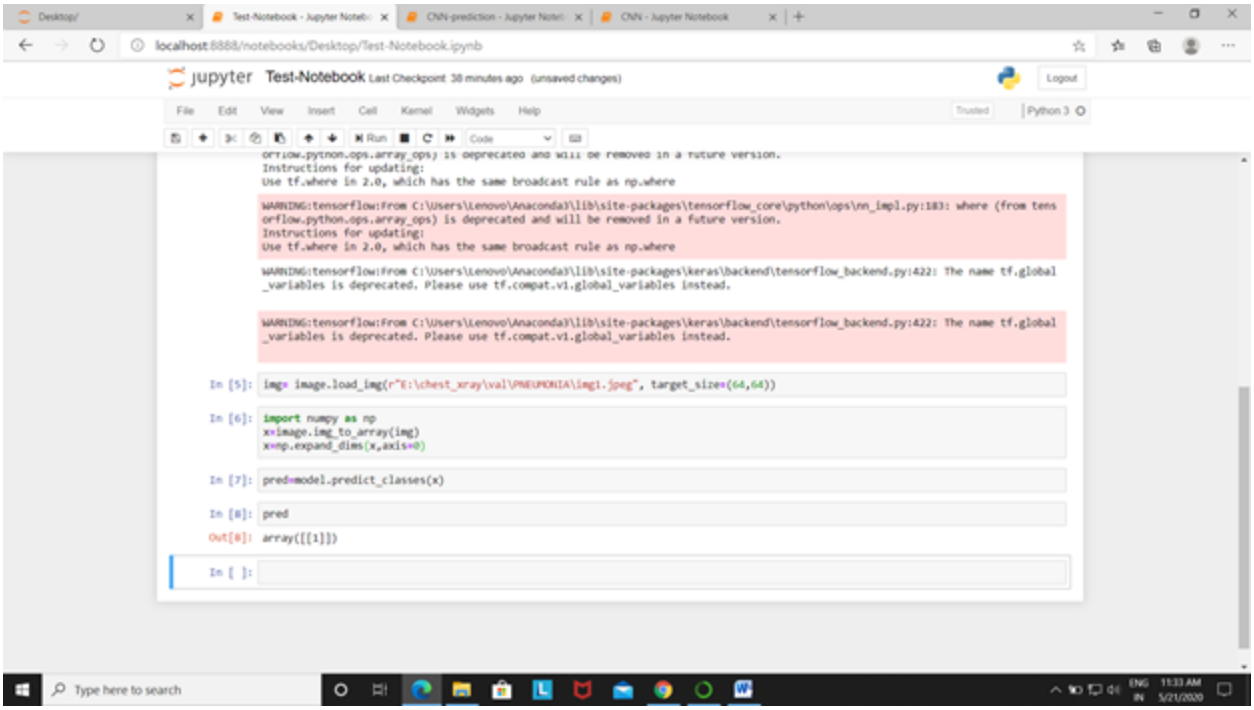




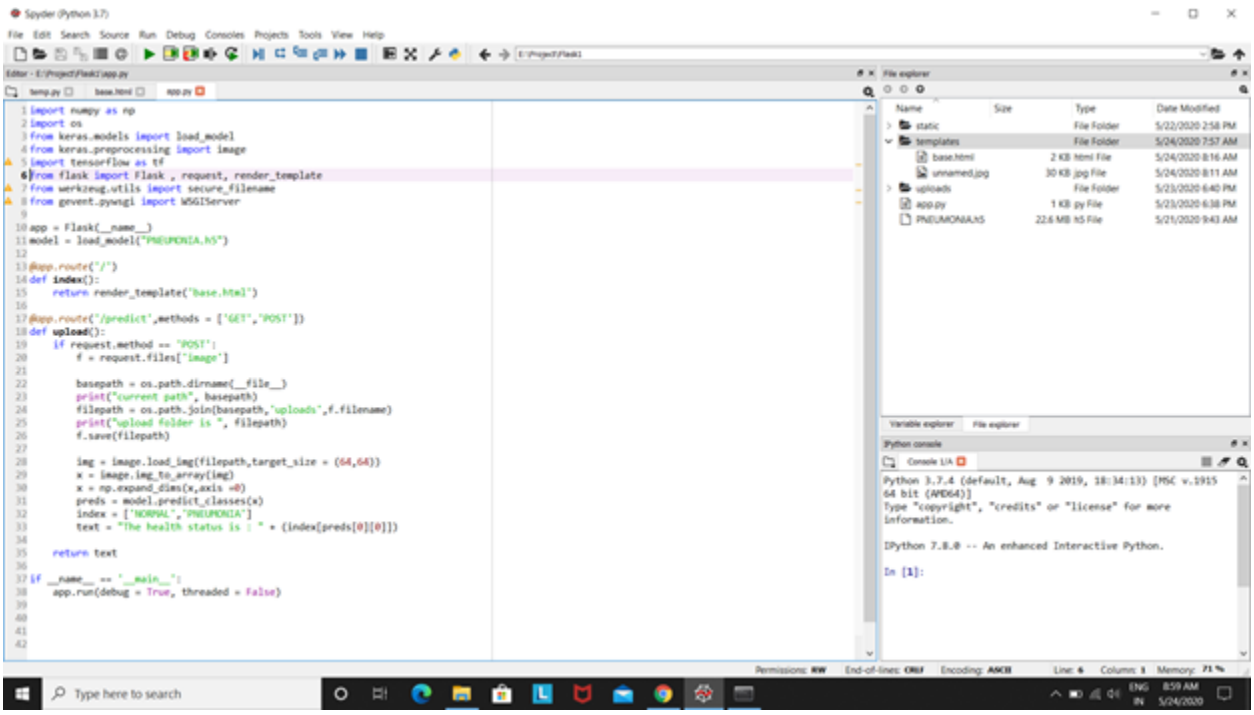


3)Validation of model:



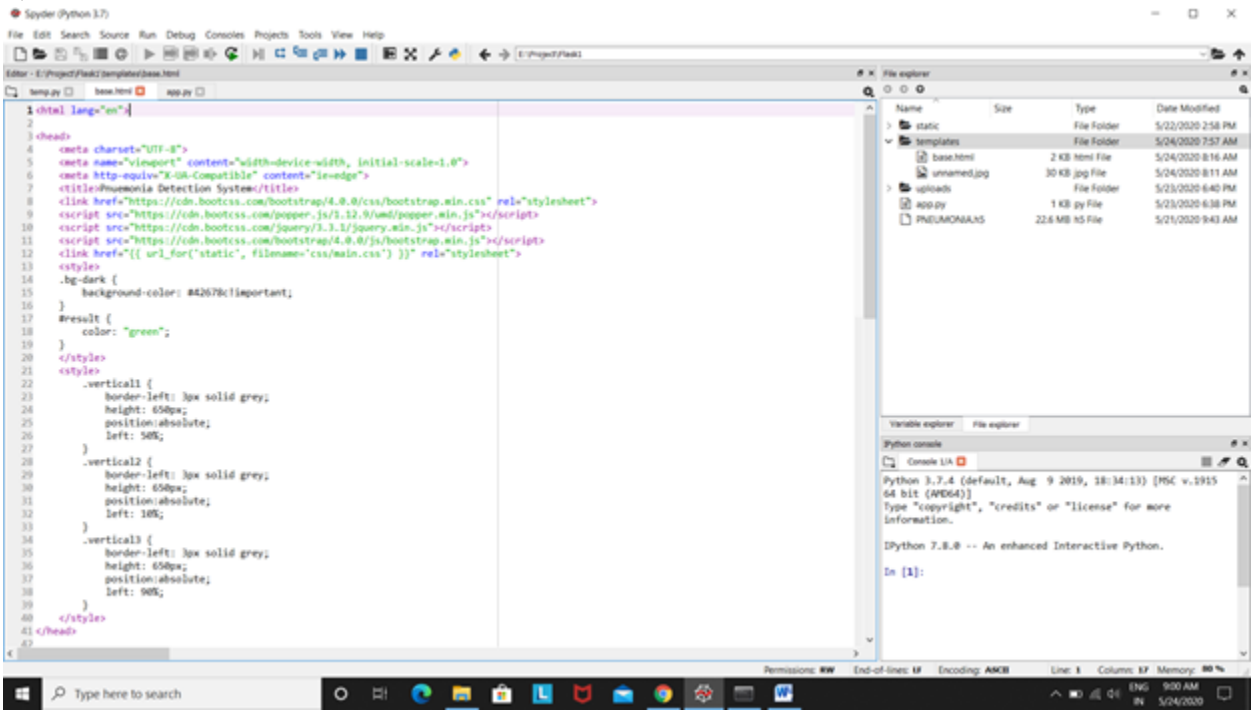


4)Flask files:



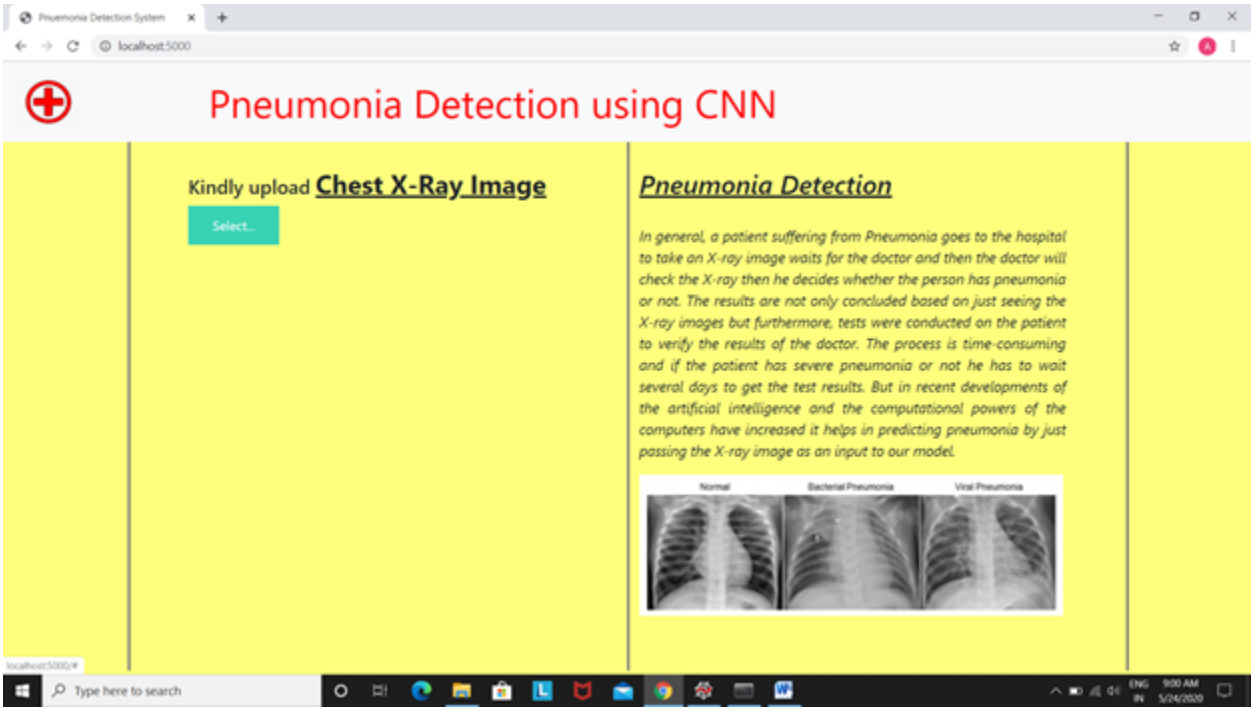
```
import numpy as np
import os
from keras.models import load_model
from keras.preprocessing import image
import tensorflow as tf
from flask import Flask , request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

app = Flask(__name__)
model = load_model("PNEUMONIA.h5")

@app.route('/')
def index():
    return render_template('base.html')

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']

        basepath = os.path.dirname(__file__)
        print("current path", basepath)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)

        img = image.load_img(filepath,target_size = (64,64))
        x = image.img_to_array(img)
        x = np.expand_dims(x,axis =0)
        preds = model.predict_classes(x)
        index = ['NORMAL','PNEUMONIA']
        text = "The health status is : " + (index[preds[0][0]])

    return text

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

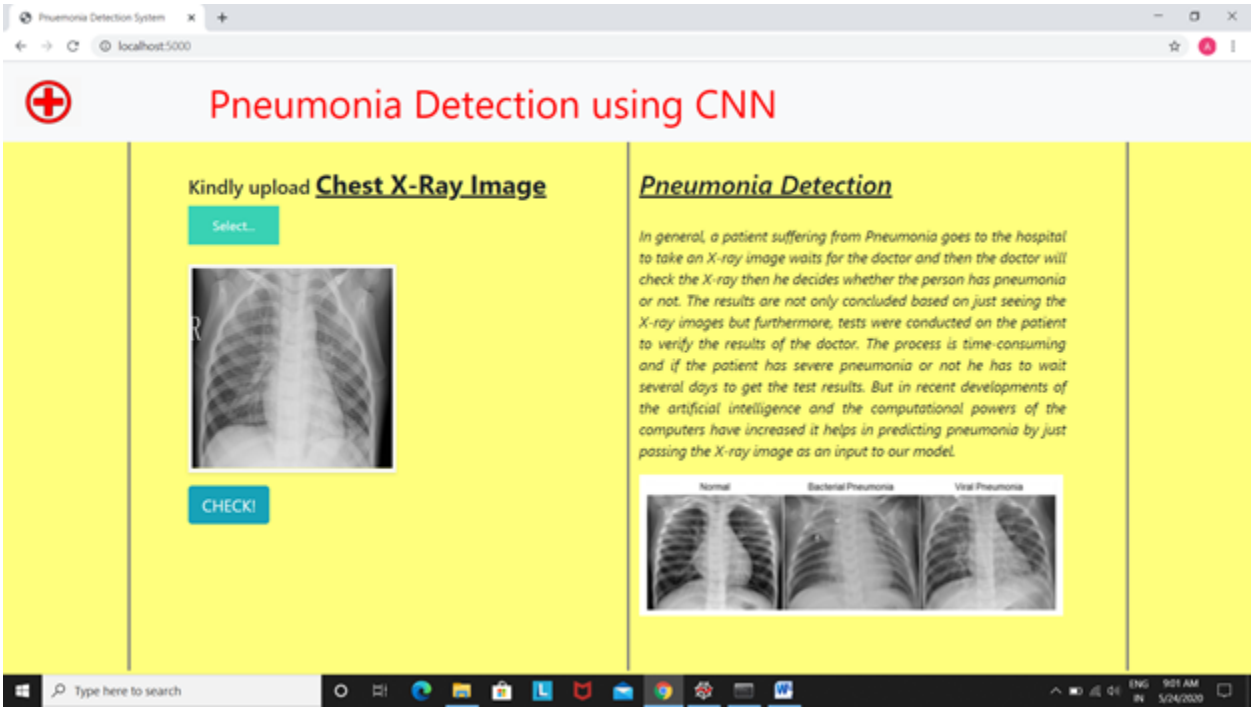
5)HTML file:



```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Pnuemonia Detection System</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
    <style>
    .bg-dark {
        background-color: #42678c!important;
    }
    #result {
        color: "green";
    }
    </style>
    <style>
        .vertical1 {
            border-left: 3px solid grey;
            height: 650px;
            position:absolute;
            left: 50%;
        }
        .vertical2 {
            border-left: 3px solid grey;
            height: 650px;
            position:absolute;
            left: 10%;
        }
        .vertical3 {
            border-left: 3px solid grey;
            height: 650px;
            position:absolute;
            left: 90%;
        }
    </style>
</head>
```

6)Execution:


Pneumonia Detection System
localhost:5000
Pneumonia Detection using CNN
Kindly upload Chest X-Ray Image
Select...
Pneumonia Detection
In general, a patient suffering from Pneumonia goes to the hospital to take an X-ray image waits for the doctor and then the doctor will check the X-ray then he decides whether the person has pneumonia or not. The results are not only concluded based on just seeing the X-ray images but furthermore, tests were conducted on the patient to verify the results of the doctor. The process is time-consuming and if the patient has severe pneumonia or not he has to wait several days to get the test results. But in recent developments of the artificial intelligence and the computational powers of the computers have increased it helps in predicting pneumonia by just passing the X-ray image as an input to our model.
Normal
Bacterial Pneumonia
Viral Pneumonia
Type here to search


Pneumonia Detection System
localhost:5000
Pneumonia Detection using CNN
Kindly upload Chest X-Ray Image
Select...
CHECK!
Pneumonia Detection
In general, a patient suffering from Pneumonia goes to the hospital to take an X-ray image waits for the doctor and then the doctor will check the X-ray then he decides whether the person has pneumonia or not. The results are not only concluded based on just seeing the X-ray images but furthermore, tests were conducted on the patient to verify the results of the doctor. The process is time-consuming and if the patient has severe pneumonia or not he has to wait several days to get the test results. But in recent developments of the artificial intelligence and the computational powers of the computers have increased it helps in predicting pneumonia by just passing the X-ray image as an input to our model.
Normal
Bacterial Pneumonia
Viral Pneumonia
Type here to search

Pneumonia Detection System

localhost:5000

Pneumonia Detection using CNN

Kindly upload **Chest X-Ray Image**

Select...

Pneumonia Detection

In general, a patient suffering from Pneumonia goes to the hospital to take an X-ray image waits for the doctor and then the doctor will check the X-ray then he decides whether the person has pneumonia or not. The results are not only concluded based on just seeing the X-ray images but furthermore, tests were conducted on the patient to verify the results of the doctor. The process is time-consuming and if the patient has severe pneumonia or not he has to wait several days to get the test results. But in recent developments of the artificial intelligence and the computational powers of the computers have increased it helps in predicting pneumonia by just passing the X-ray image as an input to our model.

Type here to search

ENG IN 9:01 AM 5/24/2020

Pneumonia Detection System

localhost:5000

Pneumonia Detection using CNN

Kindly upload **Chest X-Ray Image**

Select...

Prediction : The health status is : PNEUMONIA

Pneumonia Detection

In general, a patient suffering from Pneumonia goes to the hospital to take an X-ray image waits for the doctor and then the doctor will check the X-ray then he decides whether the person has pneumonia or not. The results are not only concluded based on just seeing the X-ray images but furthermore, tests were conducted on the patient to verify the results of the doctor. The process is time-consuming and if the patient has severe pneumonia or not he has to wait several days to get the test results. But in recent developments of the artificial intelligence and the computational powers of the computers have increased it helps in predicting pneumonia by just passing the X-ray image as an input to our model.

Type here to search

ENG IN 9:01 AM 5/24/2020