Project Report

Pneumonia Predicton Using X-Ray Images

Work Submitted by :
Shantanu Bhardwaj

Table of Contents

Introduction

1.1. Overview

Using Deep-Learning algorithm whether a human has pneumonia or not can be determined from an X-Ray image of the human lungs. The algorithm works in such a way that when an image is uploaded it predicts whether or not the given image is an X-Ray of lungs having pneumonia or normal.This is an example of binary classification.

Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs may fill with fluid or pus (purulent material), causing cough with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms, including bacteria, viruses and fungi, can cause pneumonia. Pneumonia can range in seriousness from mild to life-threatening. It is most serious for infants and young children, people older than age 65, and people with health problems or weakened immune systems. Through this project we have tried to obtain accurate and precise models for detecting whether a person has Pneumonia through his Chest-X Rays. We have leveraged the use of CNN (Convolutional Neural Networks) and ANN (Artificial Neural Networks) to obtain Deep learning Models with accuracy of ~91 % on test set and ~94 % on test set. The project Repository is published on github as an open source project.

1.2. Purpose

The purpose of this project is pnemonia prediction using x-ray images is to create a web application which users can upload an image of their X-ray and find out whether they have pneumonia or normal by developping an accurate model to predict Pneumonia through X-Ray Images. This is mainly to erase the Human error which cannot be controlled. With this model a user(patient) can upload his/her Chest X-ray images to the deployed application and test for themselves and to get medical attention if needed.

Literature Survey

2.1. Existing Problem

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.

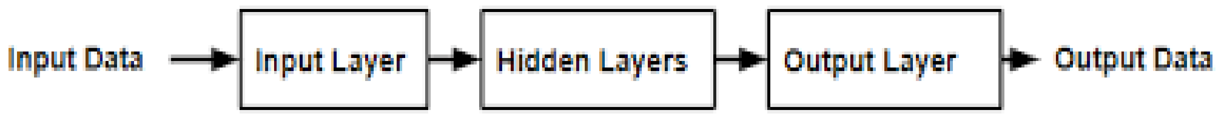
If a person thinks he is suffering from pneumonia then he should go to the hospital and there the doctors they will see the X-ray of the lungs to predict if there is pneumonia. However they cannot tell that the patient is suffering from pneumonia just by seeing the X-ray. Other tests must also be perform to determine with certainity that the patient has pneumonia.

2.2. Proposed Solution

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. A convolutional neural network model is built from scratch to extract features from a given chest X-ray image and classify it to determine if a person is infected with pneumonia. A web-interface is built where the user can upload the x - ray image and the result is shown on the UI (User Interface).

Theoretical Analysis

3.1. Block diagram



The Block Diagram as shown as in the following page shows the different parts of our system architecture. The Input Image is converted to a 96 X 96 X 3 matrix and is passed onto the Convolution layer. The Convolutional Layer Convolutes the input images based on several features. This is then passed to Max Pooling Layer. After which we take a Dropout Of 0.2 (i.e) take only 80% of the features. This is done to reduce Overfitting. This is then passed to Another Convolutional Layer followed by a Max Pooling Layer. The Features are then Flattened to Form a Single Dimension Array. Another Dropout of 0.2 is taken at this stage. The Flattened layer is then Passed to ANN(Artificial Neural Network) which contains two hidden layers(activation =relu) comprising 128 nodes each. This is then passed to Output Layer which has a sigmoid activation function for predicting results.

3.2. Hardware/Software Designing

For creating the CNN model we need:

keras 2.4.4 to import libraries for ImageDataGenerator for data preprocessing and sequential,dense,etc. for building the model.
Numpy and tensorflow 1.14. For creating app.py python code to run the web application we require to import os, global graph, flask and WSGIServer.

Experimental Investigations

First data preprocessing rescaling and transformation was performed on the images of the given dataset. Then libraries are imported and model was initialised. After initialising the model first convolution layer is created. 3x3 feature detecter is used. Then max pooling layer is added. 2x2 matrix is used. After that flatten layer is added and last hidden and output layers. After adding all the layers of the CNN model it is compiled and trained with 10 epochs. After training model was saved.

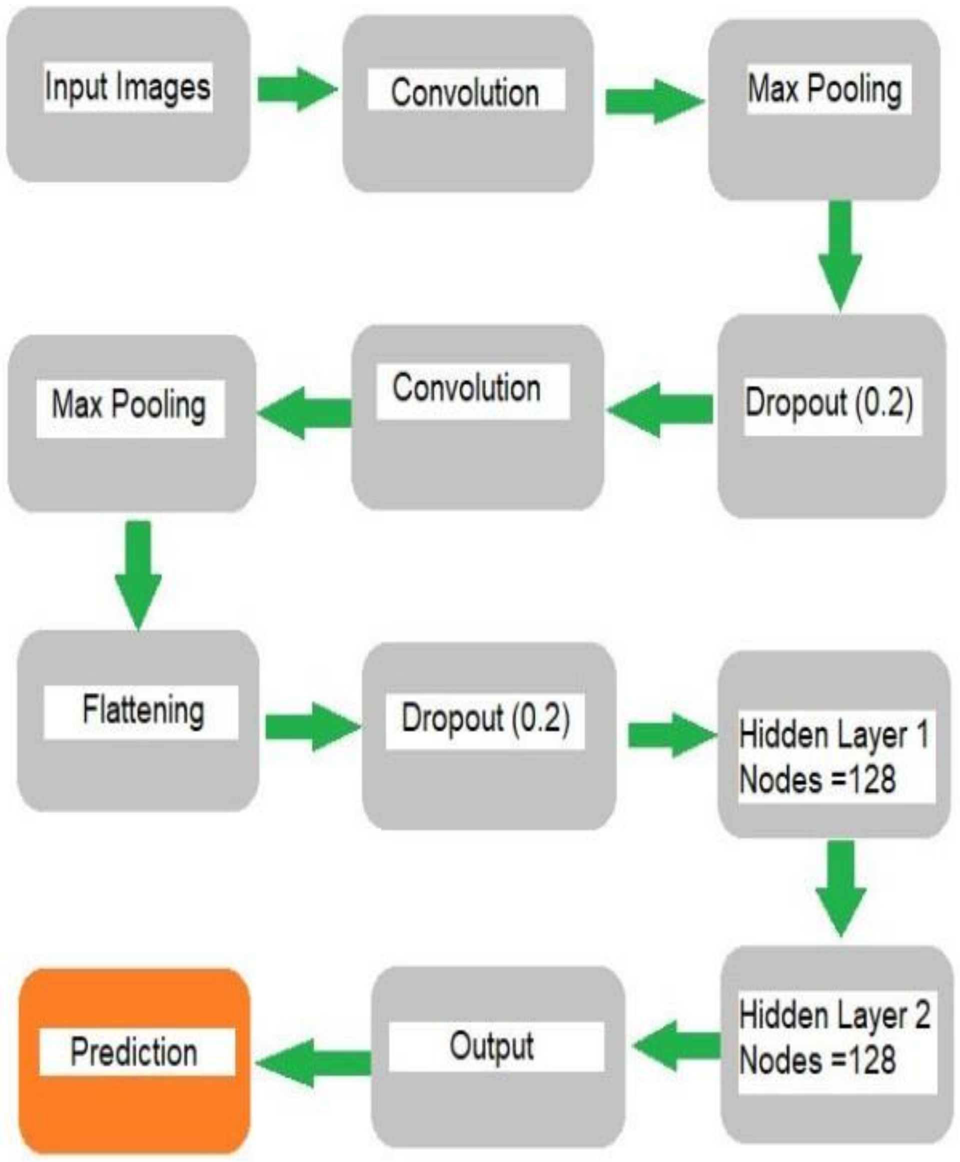
For prediction saved model is loaded, transformations takes place and the given image is predicted whether normal or pneumonia. X-ray image given here was normal and it predicted accurately as normal.

After saving cnn.h5 and deep learning algorithm file html file and corresponding styles.css and file.js are created. Css is for the layout of the web application and .js is neccesary for previewing the image uploaded in the web application.

Last python code is created app.py. This is to run the model in the web application. After importing the libraries model is loaded. Html file is rendered and uploaded image is saved to uploads folder. The image is resized, converted into array form then interpreted by the machine when running the application. Finally prediction is displayed.

Upon Conducting Experiments for tuning hyper parameters of our model to obtain better accuracy. We found that the best set of parameters were batch size
=32,Input Image size =96 x 96. The model was rescaled to 1./255 of its values to obtain a fast convergence. Two Dropout Layers were added when the model showed signs of overfitting. Dropout1 (0.2) was added after the Max Pooling 2d_1 and Dropout2(0.2) was added after flatten_1. The image data was augmented so that the machine can understand the features better. shear_range = 0.2, zoom_range rotation_range = 30, horizontal_flip = False,vertical_flip=False.

Flowchart


Input Images
Convolution
Max Pooling
Dropout (0.2)
Convolution
Max Pooling
Flattening
Dropout (0.2)
Hidden Layer 1 Nodes =128
Hidden Layer 2 Nodes =128
Output
Prediction

Result

To evaluate and validate the effectiveness of the proposed approach, we conducted the experiments 10 times each for three hours, respectively. Parameters and hyperparameters were heavily turned to increase the performance of the model. Different results were obtained, but this study reports only the most valid. As explained above, methods such as data augmentation, learning rate variation, and annealing were deployed to assist in fitting the small dataset into deep convolutional neural networks final results obtained are training loss 0.190, training accuracy 0.9467, validation loss: 0.0566, and validation accuracy of 0.9167.

When app.py is running in the terminal in the web application image can be uploaded and the model will predict whether it is pnemonia or normal. The X-Ray image will also be previewed

Advantages and disadvantages

The advantage of this cnn is it gives accurate prediction even when it is trained with less number of epochs. Generally accuarcy will be high only when more than 100 epochs are initialised.

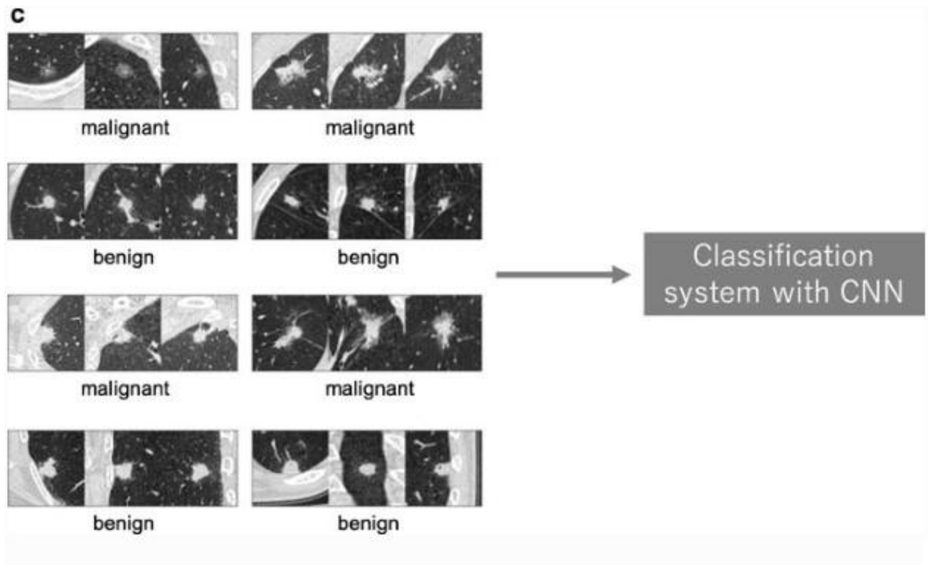
Although the results were overwhelming, there were still some limitations in our model which we believe are vital to keep in consideration. The first biggest limitation is that there is no history of the associated patient considered in our evaluation model. Secondly, only frontal chest X-rays were used but it has been shown that lateral view chest X-rays are also helpful in diagnosis. Thirdly, since the model exercises a lot of convolutional layers, the model needs very high computational power otherwise it'll eat up a lot of time in computations and spatial invarience is not possible.

Applications

Pneumonia prediction using x-ray images can be applied in medical field for identifying whether a human has pneumonia or normal.

CNN Application in Radiology:

This section introduces recent applications within radiology, which are divided into the following categories: classification, segmentation, detection, and others. Classification In medical image analysis, classification with deep learning usually utilizes target lesions depicted in medical images, and these lesions are classified into two or more classes. For example, deep learning is frequently used for the classification of lung nodules on computed tomography (CT) images as benign or malignant . As shown, it is necessary to prepare a large number of training data with corresponding labels for efficient classification using CNN. For lung nodule classification, CT images of lung nodules and their labels (i.e., benign or cancerous) are used as training data. show two examples of training data of lung nodule classification between benign lung nodule and primary lung cancer; Fig. 11b shows the training data where each datum includes an axial image and its label, and Fig. 11 c shows the training data where each datum includes three images (axial, coronal, and sagittal images of a lung nodule) and their labels. After training CNN, the target lesions of medical images can be specified in the deployment phase by medical doctors or computer-aided detection (CADe) systems .



Segmentation:

Segmentation of organs or anatomical structures is a fundamental image processing technique for medical image analysis, such as quantitative evaluation of clinical parameters (organ volume and shape) and computer-aided diagnosis (CAD) system. In the previous section, classification depends on the segmentation of lesions of interest. Segmentation can be performed manually by radiologists or dedicated personnel, a time-consuming process. However, one can also apply CNN to this task as well. Figure 12 a shows a representative example of segmentation of the uterus with a malignant tumor on MRI [24,44,45]. In most cases, a segmentation system directly receives an entire image and outputs its segmentation result. Training data for the segmentation system consist of the medical images containing the organ or structure of interest and the segmentation result; the latter is mainly obtained from previously performed manual segmentation. Figure 12 b shows a representative example of training data for the segmentation system of a uterus with a malignant tumor. In contrast to classification, because an entire image is inputted to the segmentation system, it is necessary for the system to capture the global spatial context of the entire image for efficient segmentation.

A schematic illustration of the system for segmenting a uterus with a malignant tumor and representative examples of its training data. a Segmentation system with CNN in the deployment phase. b Training data used in the training phase.

Detection:

A common task for radiologists is to detect abnormalities within medical images. Abnormalities can be rare and they must be detected among many normal cases. One previous study investigated the usefulness of 2D-CNN for detecting tuberculosis on chest radiographs. The study utilized two different types of 2D-CNN, AlexNet and GoogLeNet , to detect pulmonary tuberculosis on chest radiographs. To develop the detection system and evaluate its performance, the dataset of 1007 chest radiographs was used. According to the results, the best area under the curve of receiver operating characteristic curves for detecting pulmonary tuberculosis from healthy cases was 0.99, which was obtained by ensemble of the AlexNet and GoogLeNet 2D-CNNs. Nearly 40 million mammography examinations are performed in the USA every year. These examinations are mainly performed for screening programs aimed at detecting breast cancer at an early stage. A comparison between a CNN-based CADe system and a reference CADe system relying on hand-crafted imaging features was performed previously . Both systems were trained on a large dataset of around 45,000 images. The two systems shared the candidate detection system. The CNN-based CADe system classified the candidate based on its region of interest, and the reference CADe system classified it based on the hand-crafted imaging features obtained from the results of a traditional segmentation algorithm. The results show that the CNN-based CADe system outperformed the reference CADe system at low sensitivity and achieved comparable performance at high sensitivity.

Others:

Low-dose CT has been increasingly used in clinical situations. For example, low-dose CT was shown to be useful for lung cancer screening . Because noisy images of low-dose CT hindered the reliable evaluation of CT images, many techniques of image processing were used for denoising low-dose CT images. Two previous studies showed that low-dose and ultra-low-dose CT images could be effectively denoised using deep learning . Their systems divided the noisy CT image into image patches, denoised the image patches, then reconstructed a new CT image from the denoised image patches. Deep learning with encoder–decoder architecture was used for their systems to denoise image patches. Training data for the denoising systems consisted of pairs of image patches, which are obtained from standard-dose CT and low-dose CT. Figure 13. shows a representative example of the training data of the systems. Fig. 13

A schematic illustration of the system for denoising an ultra-low-dose CT (ULDCT) image of phantom and representative examples of its training data. a Denoising system with CNN in deployment phase. b Training data used in the training phase. SDCT, standard-dose CT.

Conclusion

In this project deep learning cnn model and web application to predict pneumonia from an x-ray image has been created. An x-ray can be uploaded and the algorithm will detect pneumonia.

Due to the importance of medical image classification and the particular challenge of the medical image-small dataset, this paper chose to study how to apply CNN-based classification to small chest X-ray dataset and evaluate their performance.

From the experiments, the following finding presented. CNN-based transfer learning is the best method of all three methods. The capsule network is better than the ORB and SVM classifier. Generally speaking, CNN based methods are better than traditional methods because they can learn and select features automatically and effectively; The best results come from the transfer learning of VGG16 with one retrained ConvLayer, which is slightly higher than the start-of-art result.

With the unfrozen ConvLayer, the specific feature can learn from the new dataset. Therefore, the specific feature is an essential factor to improve accuracy; The balance of a model's power of expression and overfitting is necessary. A too simple network usually cannot learn enough from the data, and therefore cannot get high accuracy. On the other hand, a very complex network is hard to train and tends to overfit quickly. As a result, accuracy is still low. Only a network model with proper size and other effective methods preventing overfit, such as proper dropout rate and proper data augmentation, can get the best results.

However, because of the limited time, future research needs to be done: In transfer learning, training a fine-tuned deep neural network with unfrozen ConvLayers tends to overfit. What can effective methods be done to stabilize the training process? Other more powerful CNN model, such as ResNetv2 and ensemble of multiple CNN models have not been evaluated, but they could improve the results; Visualization needs to be added to improve the understanding and explanation of the results of the CNN-based system, because those are essential for the adoption of a CNN-based system in real clinical applications.

Future scope

The web application can be used for identifying pneumonis in a patient. Currently it is a time consuming process to diagnos a patient with pneumonia but with this application it can identify in very less time and patients suffering from severe pneumonia can get treatment immediately.

Deep learning rose to its prominent position in computer vision when neural networks started outperforming other methods on several high-profile image analysis benchmarks. Most famously on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012. when a deep learning model (a convolutional neural network) halved the second best error rate on the image classification task. Enabling computers to recognize objects in natural images was until recently thought to be a very difficult task, but by now convolutional neural networks have surpassed even human performance on the ILSVRC, and reached a level where the ILSVRC classification task is essentially solved (i.e. with error rate close to the Bayes rate).

Deep learning techniques have become the de facto standard for a wide variety of computer vision problems. They are, however, not limited to image processing and analysis but are outperforming other approaches in areas like natural language processing, speech recognition and synthesis and in the analysis of unstructured, tabular-type data using entity embeddings. The sudden progress and wide scope of deep learning, and the resulting surge of attention and multi-billion-dollar investment, has led to a virtuous cycle of improvements and investments in the entire field of machine learning. It is now one of the hottest areas of study world-wide, and people with competence in machine learning are highly sought-after by both industry and academia.

Healthcare providers generate and capture enormous amounts of data containing extremely valuable signals and information, at a pace far surpassing what “traditional” methods of analysis can process. Machine learning therefore quickly enters the picture, as it is one of the best ways to integrate, analyse and make predictions based on large, heterogeneous data sets. Healthcare applications of deep learning range from one-dimensional bio signal analysis and the prediction of medical events,
e.g. seizures and cardiac arrests , to computer-aided detection and diagnosis supporting clinical decision making and survival analysis to drug discovery and as an

aid in therapy selection and pharmacogenomics , to increased operational efficiency , stratified care delivery , and analysis of electronic health records.

The use of machine learning in general and deep learning in particular within healthcare is still in its infancy, but there are several strong initiatives across academia, and multiple large companies are pursuing healthcare projects based on machine learning. Not only medical technology companies, but also for example Google Brain DeepMind Microsoft and IBM. There is also a plethora of small and medium-sized businesses in the field

Bibilography

[1] P. Rui, K. Kang, National Ambulatory Medical Care Survey: 2018 Emergency Department Summary Tables.
[2] E. Sayed, et. al, Computer-aided Diagnosis of Human Brain Tumor through MRI: A Survey and a new algorithm, Expert System with Applications (41): 2014.
[3] D.K. Das, M. Ghosh, M. Pal, A.K. Maiti, and C. Chakraborty, Machine learning approach for automated screening of malaria parasites using light microscopic images, Micron 45, 97106, 2013.
[4] M. Poostchi, K. Silamut, R. Maude, S. Jaeger, and G. Thoma, Image analysis and machine learning for detecting malaria, Translational Research, 2018.
[5] N.E. Ross, C.J. Pritchard, D.M. Rubin, and A.G. Duse, Automated image processing method for the diagnosis and classification of malaria on thin blood smears, Medical and Biological Engineering and Computing 44, 5, 427436, 2006.
[6] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 806813, 2014.
[7] Fafi, Kusworo and Catur, Breast Tumor Detection using Haar-Like Feature Method on Ultrasonography (USG) Imaging, International Journal of Innovative Research in Advanced Engineering, Volume V, 154-157. doi://10.26562/IJIRAE.2018.MYAE10082.
INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9,
ISSUE 04, APRIL 2020 ISSN 2277-8616 1337 IJSER©2020 www.ijstr.org
[8] Abiyev R. H., Maaitah M. K. S., Deep Convolutional Neural Networks for Chest Diseases Detection, Journal of Healthcare Engineering, 2018.
[9] A. Krizhevsky, I. Sutskever, & G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, Neural Information Processing Systems, 25. 10.1145/3065386, 2012.

Source code

Pneumonia.py(Deep learning Algorithm)

```
import numpy as np
import pandas as pd

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

project=Sequential()
project.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
project.add(MaxPooling2D(pool_size=(2,2)))
project.add(Flatten())
project.add(Dense(output_dim =128,init ='uniform' , activation='relu'))
project.add(Dense(output_dim=1, init='uniform',activation='sigmoid'))


from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2, zoom_range=0.2
,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)

x_train=train_datagen.flow_from_directory(r"C:\Users\lavke\Downloads\Dataset_Chest_XR
ay\train",target_size=(64,64), batch_size=32 ,class_mode='binary')
x_test=test_datagen.flow_from_directory(r"C:\Users\lavke\Downloads\Dataset_Chest_XRay\
test" ,target_size=(64,64),batch_size=32 , class_mode='binary')


print(x_train.class_indices)

project.compile(loss= 'binary_crossentropy', optimizer="adam" , metrics=["accuracy"])
project.fit_generator(x_train , steps_per_epoch =500,epochs =10,validation_data=x_test
,validation_steps= 100)
project.save("pneumonia.h5")


#Predicting output from our trained model
from keras.preprocessing import image
img=image.load_img(r"C:\Users\lavke\Downloads\chestimg.jpg" ,target_size=(64,64))
```

```
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)


pred =project.predict_classes(x)
pred
```

app.py(Flask Code)

```
from __future__ import division , print_function
import sys
import os
import glob
import numpy as np
from keras.preprocessing import image
from keras.applications.imagenet_utils import preprocess_input, decode_predictions

from keras.models import load_model
from keras import backend
from tensorflow.keras import backend


import tensorflow as tf
global graph
graph = tf.get_default_graph()
from skimage.transform import resize
from flask import Flask,request,render_template,redirect,url_for
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

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

@app.route('/',methods = ["GET"] )
def index():
    return render_template('base.html')

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']
        print("current path")
        basepath = os.path.dirname(__file__)#gives the current path of this 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)

        with graph.as_default():
            preds = model.predict_classes(x)


            print("prediction",preds)

        index = ['NORMAL' , 'PNEUMONIA']

        text = "The Prediction is....: " + str(index[preds[0][0]])

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

base.html

```
<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>PNEUMONIA</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: #0a1c4ed1;
            }
            </style>
</head>

<body>
    <nav class="navbar navbar-dark bg-dark">
        <div class="container">
```

```
            <a class="navbar-brand" href="#">PNEUMONIA PREDICTION USING CNN</a>
        </div>
    </nav>
    <div class="container">
        <div id="content" style="margin-top:2em">
                <div class="container">
                  <div class="row">
                        <div class="col-sm-6 bd" >
                          <h3>PNEUMONIA PREDICTION USING X-RAYS </h3>
                          <br>
                          <p>Pneumonia is a lung inflammation caused by a viral or bacterial
infection that can range from mild to severe cases. This inflammation makes the patient
unable to breathe enough oxygen to reach the bloodstream. It happens when an infection
makes the air sacs (alveoli) in the lungs fill with fluid or pus that might affect either one or
both lungs. If your doctor thinks you might have pneumonia, a chest X-ray will be performed
to find the infection in the patient's lungs and how far it’s spread.</p>

                        </div>
                        <div class="col-sm-6">
                                <div>
                                        <h4>Please upload your x-ray of chest</h4>
                        <form action = "http://localhost:5000/predict" id="upload-file"
method="post" enctype="multipart/form-data">
                                <label for="imageUpload" class="upload-label">
                                        Choose...
                                </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">Click on this to see what animal it is!</button>
                                </div>
                        </div>

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

                        <h3>
```

```
                                        <span id="result"> </span>
                              </h3>

                    </div>
                              </div>

                      </div>
                    </div>
                    </div>
    </div>
</body>

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

</html>
```

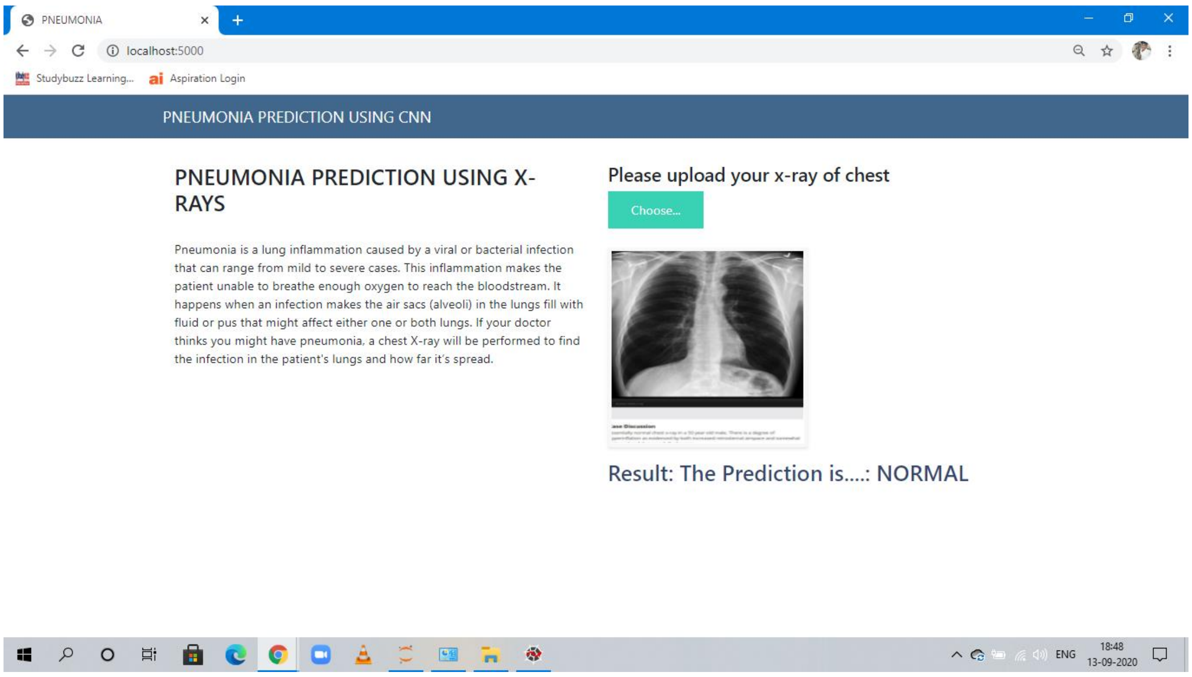
main.css

```
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}

.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

main.js

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image', 'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
    $("#imageUpload").change(function () {
        $('.image-section').show();
        $('#btn-predict').show();
        $('#result').text('');
        $('#result').hide();
        readURL(this);
    });

    // Predict
    $('#btn-predict').click(function () {
        var form_data = new FormData($('#upload-file')[0]);

        // Show loading animation
        $(this).hide();
        $('.loader').show();

        // Make prediction by calling api /predict
        $.ajax({
            type: 'POST',
            url: '/predict',
            data: form_data,
            contentType: false,
            cache: false,
            processData: false,
            async: true,
            success: function (data) {
                // Get and display the result
                $('.loader').hide();
                $('#result').fadeIn(600);
                $('#result').text(' Result:  ' + data);
                console.log('Success!');
            },
        });
    });

});
```

UI Output Screenshots



PNEUMONIA

localhost:5000

Studybuzz Learning... Aspiration Login

PNEUMONIA PREDICTION USING CNN

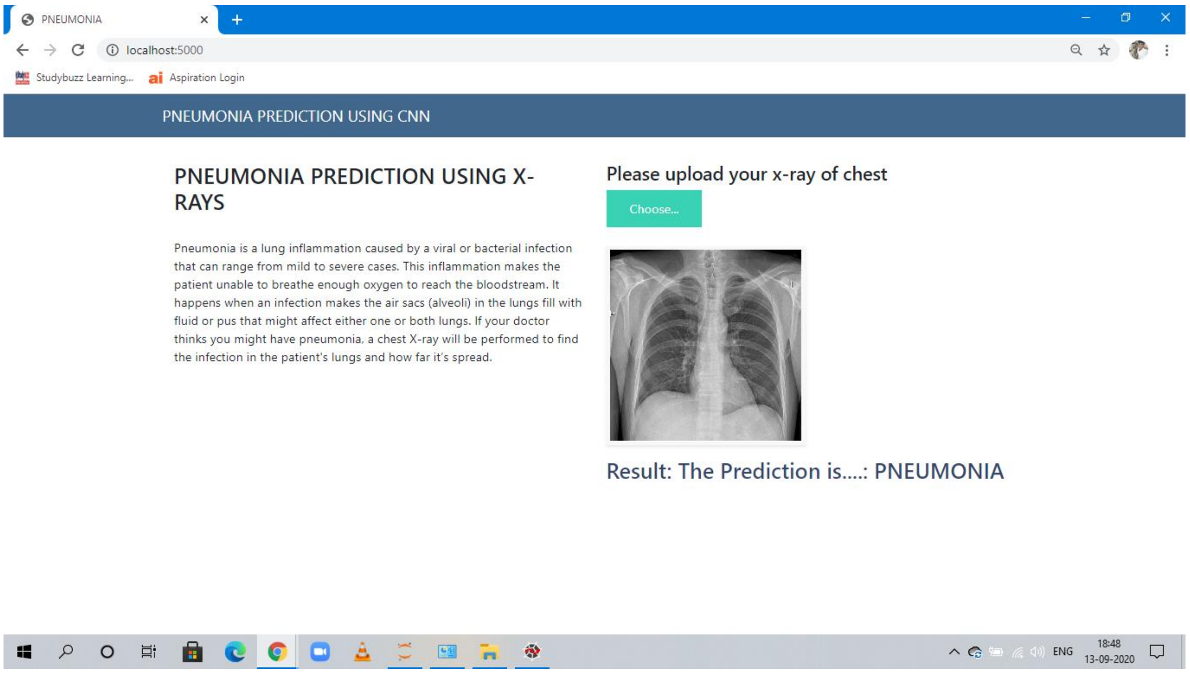
PNEUMONIA PREDICTION USING X-RAYS

Pneumonia is a lung inflammation caused by a viral or bacterial infection that can range from mild to severe cases. This inflammation makes the patient unable to breathe enough oxygen to reach the bloodstream. It happens when an infection makes the air sacs (alveoli) in the lungs fill with fluid or pus that might affect either one or both lungs. If your doctor thinks you might have pneumonia, a chest X-ray will be performed to find the infection in the patient's lungs and how far it's spread.

Please upload your x-ray of chest

Choose...

Result: The Prediction is....: NORMAL



PNEUMONIA

localhost:5000

Studybuzz Learning... Aspiration Login

PNEUMONIA PREDICTION USING CNN

PNEUMONIA PREDICTION USING X-RAYS

Pneumonia is a lung inflammation caused by a viral or bacterial infection that can range from mild to severe cases. This inflammation makes the patient unable to breathe enough oxygen to reach the bloodstream. It happens when an infection makes the air sacs (alveoli) in the lungs fill with fluid or pus that might affect either one or both lungs. If your doctor thinks you might have pneumonia, a chest X-ray will be performed to find the infection in the patient's lungs and how far it's spread.

Please upload your x-ray of chest

Choose...

Result: The Prediction is....: PNEUMONIA