Deep Learning For Detecting Pneumonia From X-Ray Images

1. INTRODUCTION

1.1 OVERVIEW

The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. Over 150 million people get infected with pneumonia on an annual basis especially children below 5 years. A patient suffering from Pneumonia takes an X-ray image to the doctor; with them he predicts pneumonia. The results are not just based on seeing the X-ray images, furthermore, tests will be conducted on the patient. The process was time-consuming, but in recent days artificial intelligence helps in predicting pneumonia bypassing the X-ray image. 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 precisely predict pneumonia.

1.2 PURPOSE

In this Project, we will detect Pneumonia using Deep Learning. We will create a model that will classify whether the patient is normal or suffering from pneumonia by looking at Chest X-ray images. The algorithm or the model which we will create should be extremely accurate because the lives of people are at stake.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

Pneumonia produces pericardial effusion, a disease wherein fluids fill the chest and create inhaling problems. It is a difficult step to recognize the presence of pneumonia quickly in order to receive treatment services and improve survival chances.

Chest X-rays are primarily used for the diagnosis of this disease. However, even for a trained radiologist, it is a challenging task to examine chest X-rays. There is a need to improve the diagnosis accuracy.

2.2 PROPOSED SOLUTION

Detecting pneumonia from X-ray images using deep learning is a valuable application in the field of medical image analysis. Further, DL models produced more effective performance than traditional techniques using chest X-ray images from pneumonia patients. Here is a

proposed solution for building a deep learning model to detect pneumonia from X-ray images.

Deep learning models can automate the process of pneumonia detection, reducing the need for manual interpretation of X-ray images. This can significantly speed up diagnosis and improve efficiency in healthcare settings.

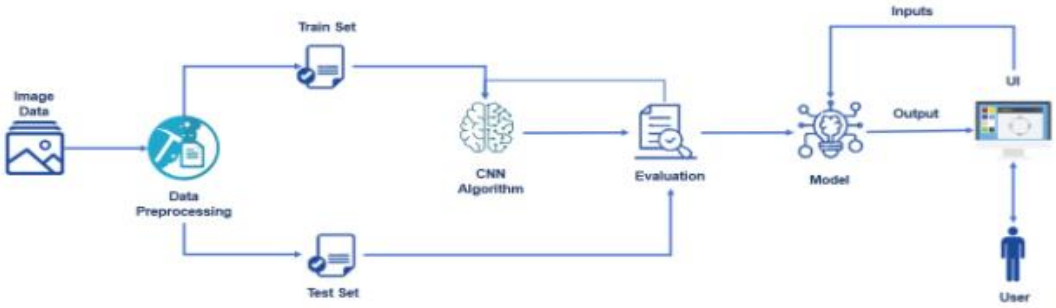
Deep learning models provide consistent results and do not suffer from fatigue or variations in human judgment. They can maintain a high level of accuracy even when processing a large number of images.

Deep learning models can process X-ray images rapidly, making them valuable for time-sensitive situations, such as identifying potential pneumonia cases in emergency rooms or during outbreaks.

Once deployed, deep learning models can be available 24/7, providing continuous support to healthcare professionals and allowing for the immediate analysis of X-ray images.
Over the long term, deep learning-based solutions can be cost-effective, as they reduce the need for extensive manual labor in image interpretation.

3. THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM



3.2 HARDWARE / SOFTWARE DESIGNING

HARDWARE REQUIREMENTS

Memory(RAM) : 4.00 GB
Processor : Intel(R)pentium(R) CPU N4200 @1.10GHz 1.10GHz
System type : 64-bit operating system,x64-based processor

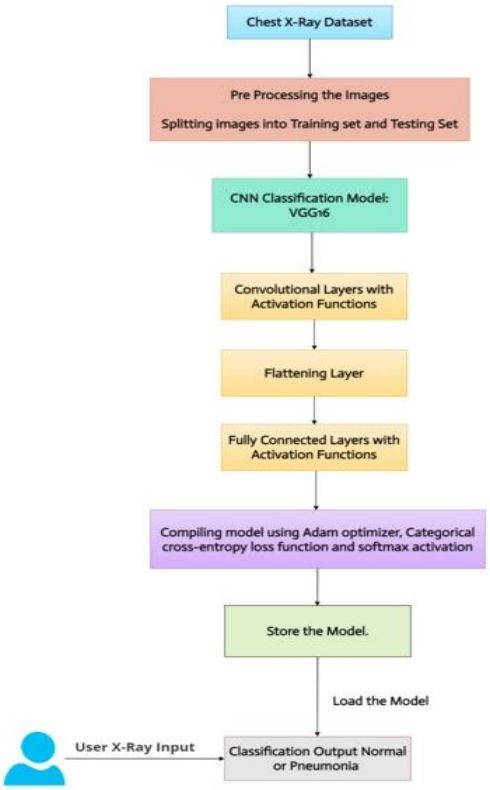
SOFTWARE REQUIREMENTS

Front End: Html, CSS, Bootstrap.
Back End: Flask, Python.

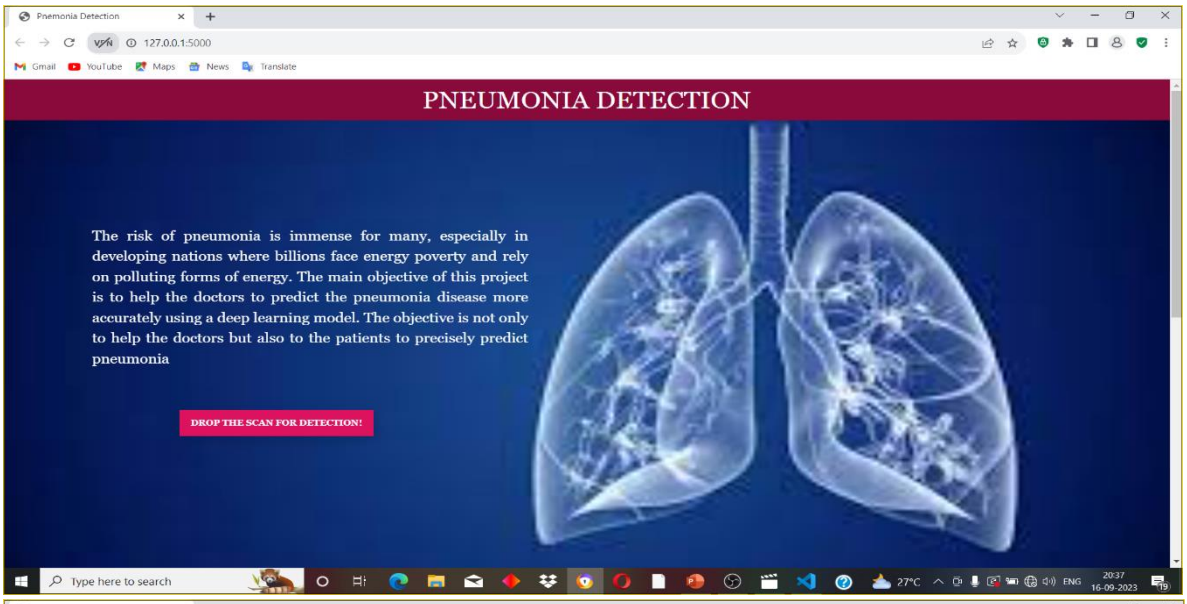
4. EXPERIMENTAL INVESTIGATIONS

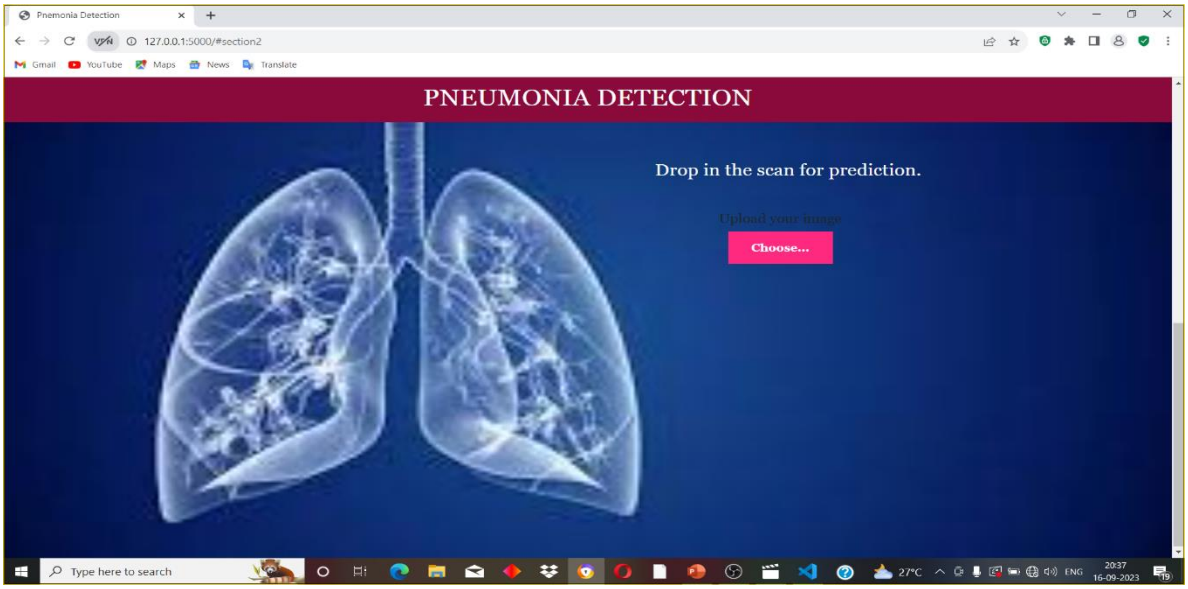
Working with pneumonia prediction using x-ray images experimental investigation
Play a crucial role in understanding the images, classifying the images, model building and fiting the model. Collect a comprehensive dataset of chest X-ray images that includes normal and pneumonia-positive cases.Ensure data quality, and address any class imbalance issues. Preprocess images by resizing, normalizing, and augmenting the dataset to increase diversity. Difficult to Choose a deep learning architecture that suits the scope of the investigation. Enhancing an existing model, may fine-tune pre-trained models like ResNet, Inception, or VGG with your dataset. Design the architecture of your deep learning model, considering the number of layers, activation functions, and regularization techniques. Split the dataset into training, validation, and test sets. Train the model using the training set with appropriate loss functions and optimizers. Monitor and fine-tune the model using the validation set to prevent overfitting. Analyzing the model's performance across different subsets of data. Ensure that data privacy and informed consent are respected. Document the experimental setup, including dataset details, model architecture, hyperparameters, and training progress.

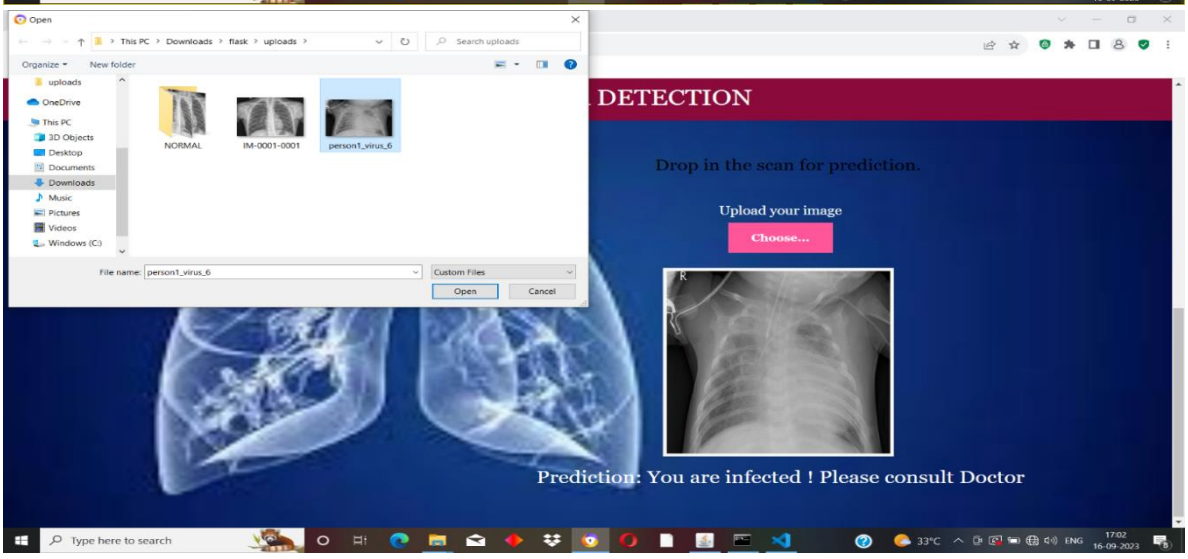
5. FLOWCHART

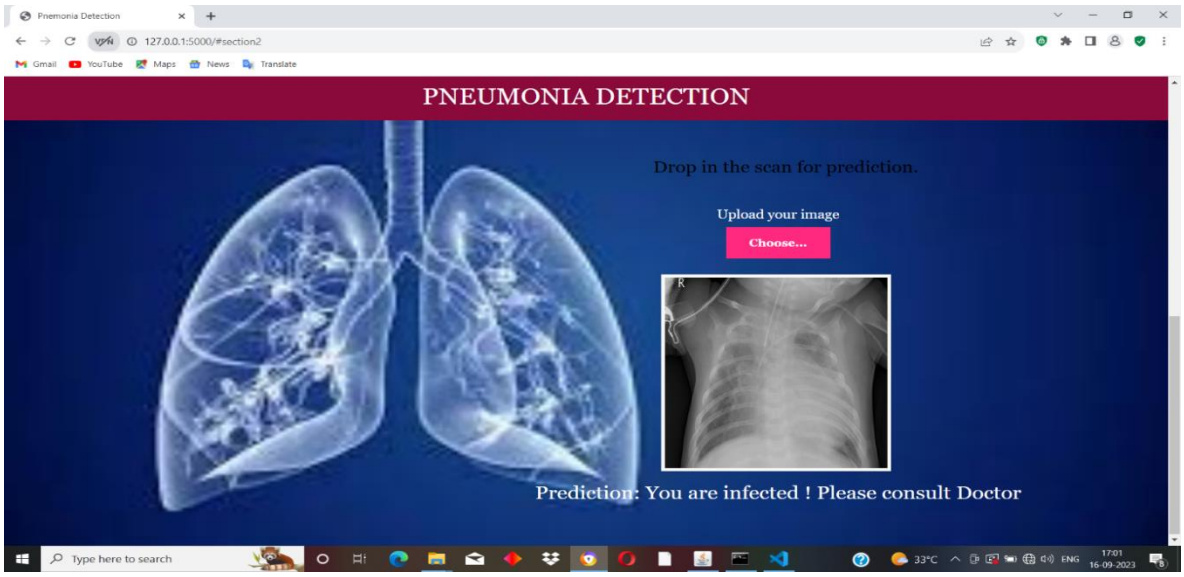


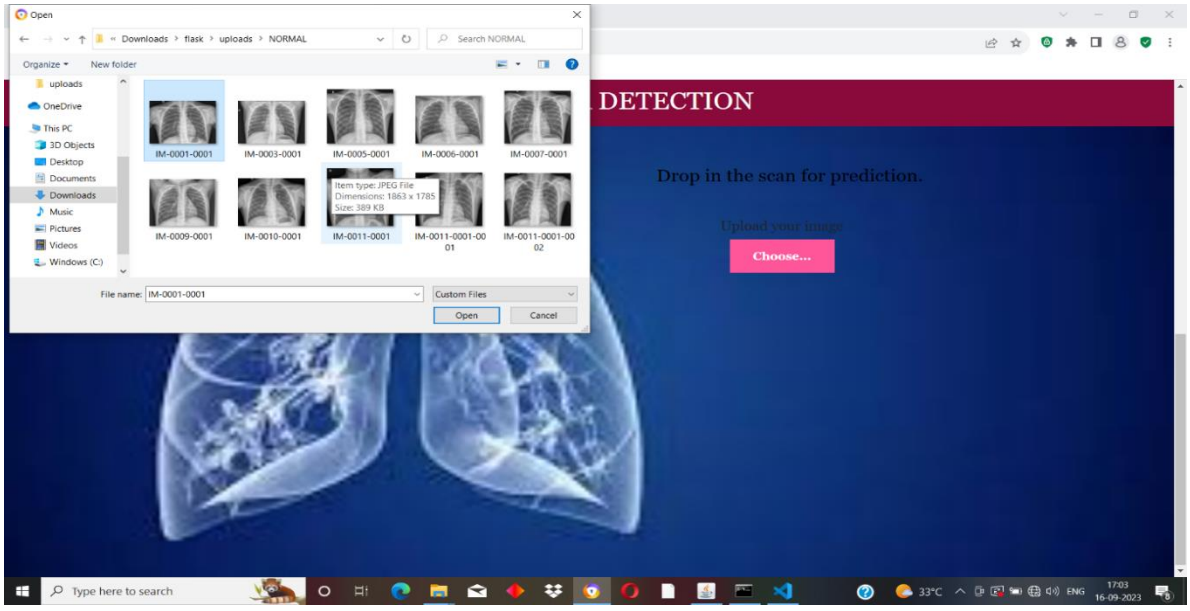
6. RESULT

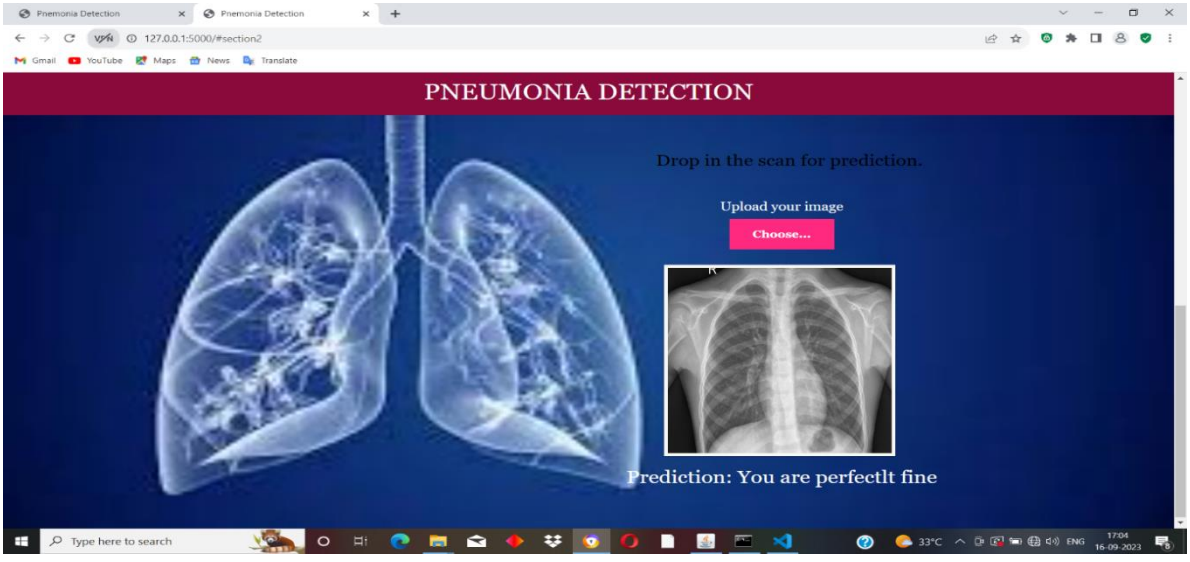

Pnemonia Detection
127.0.0.1:5000
PNEUMONIA DETECTION
The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. The main objective of this project is to help the doctors to predict the pneumonia disease more accurately using a deep learning model. The objective is not only to help the doctors but also to the patients to precisely predict pneumonia
DROP THE SCAN FOR DETECTION!


Pnemonia Detection
127.0.0.1:5000/#section2
PNEUMONIA DETECTION
Drop in the scan for prediction.
Choose...


Open
NORMAL
IM-0001-0001
person1_virus_6
DETECTION
Drop in the scan for prediction.
Upload your image
Choose...
Prediction: You are infected ! Please consult Doctor


Pneumonia Detection
127.0.0.1:5000/#section2
PNEUMONIA DETECTION
Drop in the scan for prediction.
Upload your image
Choose...
Prediction: You are infected ! Please consult Doctor


Open
Downloads > flask > uploads > NORMAL
Search NORMAL
Organize
New folder
IM-0001-0001
IM-0003-0001
IM-0005-0001
IM-0006-0001
IM-0007-0001
IM-0009-0001
IM-0010-0001
IM-0011-0001
IM-0011-0001-0001
IM-0011-0001-0002
File name: IM-0001-0001
Custom Files
Open
Cancel
DETECTION
Drop in the scan for prediction.
Choose...


Pneumonia Detection
127.0.0.1:5000/#section2
PNEUMONIA DETECTION
Drop in the scan for prediction.
Upload your image
Choose...
Prediction: You are perfectlt fine

7. ADVANTAGE & DISADVANTAGES

7.1 ADVANTAGE

- Patients and healthcare providers can access the system from anywhere, making it easier to get preliminary results and opinions without needing to be physically present at a healthcare facility.
- Deep learning models can analyze X-ray images rapidly, providing faster results compared to manual examination.
- Deep learning algorithms provide consistent and objective analysis, reducing the risk of human error and variation in interpretation.

7.2 DISADVANTAGE

- If the application provides incorrect diagnoses, it may lead to legal liability issues for the developers and healthcare providers.
- Depending on the dataset, there may be an imbalance between normal and pneumonia cases, which can affect the model's performance.
- Healthcare professionals may require training to effectively use the web application and interpret its results.

8. APPLICATIONS

- **Clinical Research**: Researchers can use deep learning models for pneumonia detection to analyze large volumes of medical imaging data. This can help in studying disease patterns, treatment outcomes, and the impact of various risk factors.
- **Education and Training**: Medical students and radiology trainees can benefit from deep learning applications as educational tools. These systems can provide case studies and assist in learning how to interpret X-ray images.
- **Triage in Emergency Departments**: In emergency departments, where time is critical, AI-powered systems can quickly analyze X-ray images to prioritize patients with pneumonia for immediate attention.
- **Early Diagnosis**: Deep learning models can assist radiologists and healthcare providers in detecting pneumonia at an early stage. This can lead to prompt treatment and improved patient outcomes.

9. CONCLUSION

Although this project is far from complete, but it is remarkable to see the success of deep learning in such varied real world problems. I have demonstrated how to classify positive and negative pneumonia data from a collection of X-ray images. The model was made from scratch, which separates it from other methods that rely heavily on transfer learning approach. In the future this work could be extended to detect and classify X-ray images consisting of lung cancer and pneumonia. Distinguishing X-ray images that contain lung cancer and pneumonia has been a big issue in recent times, and our next approach should be to tackle this problem.

10. FUTURE SCOPE

This pneumonia detector might not be a market-ready product yet, but it makes me excited to see how easy it is to get started with. With even more iterations, data and layers, I am optimistic that we can have a close to 100% accurate product**.** People will never have to suffer from not being provided with diagnosis or the accurate kind. This will help us diagnose earlier and create treatments that can help save lives.

11. BIBILIOGRAPHY

https://towardsdatascience.com/deep-learning-for-detecting-pneumonia-from-x-ray-images-fc9a3d9fdba8

12. APPENDIX

pnemonia.html

```
<html>
<head><title>Pnemonia Detection</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<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>
<style>
.header {
top:0px;
margin:0px;
left: 0px;
right: 0px;
position: fixed;
background: #8A0A3B;
color: white;
overflow: hidden;
padding-bottom: 10px;
font-size: 2.25vw;
width: 100%;
padding-left:0px;
text-align: center;
padding-top:5px;
font-family:Georgia, serif;
}
.second{
top:60px;
bottom:0px;
margin:0px;
left: 0px;
right: 0%;
```

```
position: absolute;
padding: 0px;
width: 100%;
background-image:url({{url_for('static',filename="images/images.jfif")}});
background-repeat:no-repeat;
background-size: cover;
background-position:center;
background-attachment:absolute;
}
.inside{
top:5%;
bottom:0px;
margin:0px;
left: 5%;
right: 50%;
position: absolute;
padding-left: 40px;
padding-top:8%;
padding-right:5%;
background-color:transparent;
font-family:Georgia, serif;
color:white;
font-size:20px;
text-align:justify;
margin:auto;
overflow:hidden;
}
.myButton{
border: none;
text-align: center;
cursor: pointer;
text-transform: uppercase;
outline: none;
overflow: hidden;
color: #fff;
font-weight: 700;
font-size: 12px;
background-color: #DD135F;
padding: 10px 15px;
margin: 0 auto;
box-shadow: 0 5px 15px rgba(0,0,0,0.20);
}
.predictimg{
background-image:url({{url_for('static',filename="images/images2.jfif")}});
background-repeat:no-repeat;
background-size: cover;
background-position:center;
background-attachment:absolute;
height:100%;
margin-top:49.6%;
```

```
text-align:center;
}
#showcase{
height:300px;
margin-bottom:30px;
}
html {
scroll-behavior: smooth;
}
#main{
float:center;
color:white;
width:100%;
padding:0 30px;
padding-top:7%;
box-sizing: border-box;
font-family:Georgia, serif;
text-align:center;
}
#sidebar{
float:right;
width:50%;
background-color: transparent;
color:white;
font-family:Georgia, serif;
padding-left:0px;
padding-right:0px;
padding-top:1px;
box-sizing: border-box;
}
.img-preview {
width: 300px;
height: 300px;
position: relative;
border: white;
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: 300px 300px;
background-repeat: no-repeat;
background-position: center;
}
input[type="file"] {
display: none;
}
.upload-label{
```

```
display: inline-block;
padding: 12px 30px;
background: #FF2A7F;
color: #fff;
font-size: 1em;
transition: all .4s;
cursor: pointer;
font-weight:bold;
}
.upload-label:hover{
background: #FF5599;
color: white;
font-weight:bold;
}
.loader {
border: white; /* Light grey */
border-top: 8px solid white; /* Blue */
border-radius: 50%;
width: 50px;
height: 50px;
animation: spin 1s linear infinite;
}
@keyframes spin {
0% { transform: rotate(0deg); }
100% { transform: rotate(360deg); }
}
</style>
</head>
<body>
<div class="header">PNEUMONIA DETECTION</div>
<div class="second">
<section id="showcase">
<div class="inside">The risk of pneumonia is immense for many, especially in developing nations where billions face energy poverty and rely on polluting forms of energy. 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 precisely predict pneumonia
<br><br><br>
<div style="margin-left:20%">
<a href="#section2"><button type="button" class="myButton" >Drop the scan for detection!</button></a>
</div>
</div>
</section>
</div>
<div class="predictimg" id="section2" >
<section id="main">
<div style="text-align:left;width:100%;padding-left:56%;">
<p><h3 style=font-size:25px> Drop in the scan for prediction.<br><br></h3></p>
</div>
```

```
</section>
<div style="margin-top:0%;padding-top:0%;padding-left:33%;font-family:Georgia,
serif;width:100%;">
<div>
<h4 style=font-size:19px>Upload your image</h4>
<form action = "http://localhost:5000/" 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;padding-left:35%;">
<div class="img-preview">
<div id="imagePreview">
</div>
</div>
</div>
<div class="image-section" style="display:none;">
<div>
<button type="button" class="btn btn-lg upload-label" id="btn-predict">Predict!</button>
</div>
</div>
<div class="loader" style="display:none;"></div>
<h3>
<span id="result" > </span>
</h3>
</div>
</div></div>
<script>
window.onscroll = function() {myFunction()};
$(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('Prediction: '+data);
console.log('Success!');
},
});
});
});
</script>
</body>
</html>
```

app.py

```
from __future__ import division, print_function
# coding=utf-8
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
```

```
# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
# Define a flask app
app = Flask(__name__)
# Load your trained model
model = load_model("pneumonia2.h5")
print('Model loaded. Check http://127.0.0.1:5000/')
@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('bcancer.html')
@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['image']
        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)
        img = image.load_img(file_path, target_size=(150, 150))
        x = image.img_to_array(img)
        x = np.expand_dims(x, axis=0)
       #  with graph.as_default():
        preds = model.predict(x)
        if preds[0][0]==0:
            text = "You are infected ! Please consult Doctor"
        else:
            text = "You are perfectlt fine"
        text = text
              # ImageNet Decode
        return text
if __name__ == '__main__':
    app.run(debug=False,threaded = False)
```