**SOURCE CODE:**

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

model=Sequential()

model.add(Convolution2D(32,(3,3),input\_shape=(64,64,3),activation="relu"))

model.add(MaxPooling2D(pool\_size = (2, 2)))

model.add(Flatten())

model.add(Dense(init="uniform",activation="relu",output\_dim=120))

model.add(Dense(init="uniform",activation="sigmoid",output\_dim=1))

model.compile(loss="binary\_crossentropy",optimizer="adam",metrics=["accuracy"])

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\acer\Desktop\dataset\train\_dataset',

target\_size = (64, 64),

batch\_size = 32,

class\_mode = 'binary')

x\_test = test\_datagen.flow\_from\_directory(r'C:\Users\acer\Desktop\dataset\test\_dataset',

target\_size = (64, 64),

batch\_size = 32,

class\_mode = 'binary')

history=model.fit\_generator(x\_train,

steps\_per\_epoch = 250,

epochs = 10,

validation\_data = x\_test,

validation\_steps = 63)

model.save("mymodel.h5")

import numpy as np

from skimage.transform import resize

def detect(frame):

try:

img = resize(frame,(64,64))

img = np.expand\_dims(img,axis=0)

if(np.max(img)>1):

img = img/25.0

prediction = model.predict(img)

print(prediction)

prediction = model.predict\_classes(img)

print(prediction)

except AttributeError:

print("shape not found")

import cv2

frame=cv2.imread(r" C:\Users\acer\Desktop\Malaria Application\Dataset\train\_dataset\Infected\C60P21thinF\_IMG\_20150804\_104919\_cell\_140.png")

data = detect(frame)

**FLASK CODE:**

from \_\_future\_\_ import division, print\_function

from flask import Flask, redirect, url\_for, request, render\_template, jsonify

import json

from werkzeug.utils import secure\_filename

from keras.models import load\_model

from keras.preprocessing import image

import numpy as np

import sys

import os

app = Flask(\_\_name\_\_, static\_url\_path='')

def model\_predict(img\_path, model):

img = image.load\_img(img\_path, target\_size=(64, 64))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis=0)

preds = model.predict\_classes(x)

return preds

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

def index():

return render\_template('index.html')

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

def upload():

if request.method == 'POST':

f = request.files['image']

basepath = os.path.dirname(\_\_file\_\_)

file\_path = os.path.join(

basepath, 'uploads', secure\_filename(f.filename))

f.save(file\_path)

model = load\_model('mymodel.h5')

preds = model\_predict(file\_path, model)

print("preds : "+str(preds))

ls=["Malaira","Not-Infected"]

p=preds.flatten()

result = ls[p[0]]

print(result)

return result

return None

if \_\_name\_\_ == '\_\_main\_\_':

port = int(os.getenv('PORT', 8001))

app.run(host='0.0.0.0', port=port, debug=True)

**Templates**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Breast Cancer</title>

<!-- Bootstrap -->

<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">

<link href="styles.css" rel="stylesheet">

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<!-- Include all compiled plugins (below), or include individual files as needed -->

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>

<script src="js/lib/jquery.i18n/jquery.i18n.js"></script>

<script src="js/lib/jquery.i18n/jquery.i18n.messagestore.js"></script>

<script src="js/lib/jquery.i18n/jquery.i18n.fallbacks.js"></script>

<script src="js/lib/jquery.i18n/jquery.i18n.language.js"></script>

<script src="js/lib/jquery.i18n/jquery.i18n.parser.js"></script>

<script src="js/lib/jquery.i18n/jquery.i18n.emitter.js"></script>

<script src="js/lib/jquery.i18n/jquery.i18n.emitter.bidi.js"></script>

<script src="antixss.js" type="text/javascript"></script>

<script>

$( document ).ready(function() {

$.i18n().load( {

en: {

"welcome": "Welcome.",

"name": "name",

"what\_is\_your\_name": "What is your name?",

"hello": "Hello $1",

"added\_to\_database": "Hello $1, I've added you to the database!",

"database\_contents": "Database contents: "

},

ja: {

"welcome": "ようこそ。",

"name": "名前",

"what\_is\_your\_name": "お名前を教えてください。",

"hello": "こんにちは $1",

"added\_to\_database": "こんにちは $1 さん、あなたをデータベースに追加しました。",

"database\_contents": "データベースの内容: "

}

} );

$('body').i18n();

$('#user\_name').attr("placeholder", $.i18n('name') );

});

</script>

</head>

<body>

<div class="container" id="container">

<h1 data-i18n="welcome"></h1> <!- Welcome ->

<div id="nameInput" class="input-group-lg center-block helloInput">

<p class="lead" data-i18n="what\_is\_your\_name"></p>

<input id="user\_name" type="text" class="form-control" aria-describedby="sizing-addon1" value="" />

</div>

<p id="response" class="lead text-center"></p>

<p id="databaseNames" class="lead text-center"></p>

</div>

<footer class="footer">

<div class="container">

<span><a href="https://console.bluemix.net/docs/tutorials/index.html" target="\_blank">Looking for more tutorials?</a></span>

</div>

</footer>

</body>

</html>

<script>

//Submit data when enter key is pressed

$('#user\_name').keydown(function(e) {

var name = $('#user\_name').val();

if (e.which == 13 && name.length > 0) { //catch Enter key

//POST request to API to create a new visitor entry in the database

$.ajax({

method: "POST",

url: "./api/visitors",

contentType: "application/json",

data: JSON.stringify({name: name })

})

.done(function(data) {

if(data && data.name){

if(data.\_id)

$('#response').html($.i18n('added\_to\_database', AntiXSS.sanitizeInput(data.name)));

else

$('#response').html($.i18n('hello', AntiXSS.sanitizeInput(data.name)));

}

else {

$('#response').html(AntiXSS.sanitizeInput(data));

}

$('#nameInput').hide();

getNames();

});

}

});

//Retrieve all the visitors from the database

function getNames(){

$.get("./api/visitors")

.done(function(data) {

if(data.length > 0) {

data.forEach(function(element, index) {

data[index] = AntiXSS.sanitizeInput(element)

});

$('#databaseNames').html($.i18n('database\_contents') + JSON.stringify(data));

}

});

}

//Call getNames on page load.

getNames();

</script>

**Setup**

"""

Hello World app for running Python apps on Bluemix

"""

# Always prefer setuptools over distutils

from setuptools import setup, find\_packages

# To use a consistent encoding

from codecs import open

from os import path

here = path.abspath(path.dirname(\_\_file\_\_))

# Get the long description from the README file

with open(path.join(here, 'README.md'), encoding='utf-8') as f:

long\_description = f.read()

setup(

name='python-hello-world-flask',

version='1.0.0',

description='Hello World app for running Python apps on Bluemix',

long\_description=long\_description,

url='https://github.com/IBM-Bluemix/python-hello-world-flask',

license='Apache-2.0'

)

**JS**

/\*

\* Copyright IBM Corporation 2017

\*

\* Licensed under the Apache License, Version 2.0 (the "License");

\* you may not use this file except in compliance with the License.

\* You may obtain a copy of the License at

\*

\* http://www.apache.org/licenses/LICENSE-2.0

\*

\* Unless required by applicable law or agreed to in writing, software

\* distributed under the License is distributed on an "AS IS" BASIS,

\* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

\* See the License for the specific language governing permissions and

\* limitations under the License.

\*/

"use strict";

class AntiXSS {

static sanitizeInput(str) {

return String(str).replace(/&(?!amp;|lt;|gt;)/g, '&amp;').replace(/</g, '&lt;').replace(/>/g, '&gt;');

}

}

$(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!');

},

});

});

});