## DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

WEB-BASED IMAGE CLASSIFICATION

Submitted by

## KEERTHIKA R (927621BEC086)

## KRITHIGA D (927621BEC097)

**Elite Training Project Report**

Submission Date: **21/08/2023**

**Signature of Staff Incharge Signature of HOD**

**Signature of Elite Training Coordinator**

**Dr.D.Pradeep, AP/CSE**

**Summary**

The "Web-Based Image Classification" project aims to create an intuitive and user-friendly web application that utilizes machine learning techniques to classify images into different categories. The web app will provide users with the ability to upload images, which will then be processed by pre-trained image classification models. The application will analyze the visual features of the images and assign them to appropriate labels or classes, effectively automating the categorization process. The project will involve building a frontend interface where users can interact with the application, uploading images and receiving real-time classification results. On the backend, machine learning models will handle the image analysis and classification tasks. The project will not only enhance the understanding of image processing and machine learning but also offer a practical tool for users to organize and manage their image collections efficiently. Through this project, the development of an intelligent and accessible image classification platform on the web will be achieved, demonstrating the power of combining machine learning with web technologies.

**Source Code**

**Web Application code**

<!DOCTYPE html>

<html lang="en">

<head>

<title>IMAGE CLASSFICATION</title>

<meta charset="utf-8">

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

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

</head>

<body>

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url("image bcktnd.png");

background-color: #cccccc;

height: 500px;

background-position: center;

background-repeat: no-repeat;

background-size: cover;

position: relative;

}

</style>

</head>

<body>

</body>

</html>

<div class="container">

<h1 class="jumbotron bg-primary">IMAGE CLASSFICATION</h1>

<br><br>

<form class="form-horizontal" action="/submit" method="post" enctype="multipart/form-data">

<div class="form-group">

<label style="color:white;" class="control-label col-sm-2" for="pwd">Upload Image Here:</label>

<div class="col-sm-10">

<input type="file" class="form-control" placeholder="Hours Studied" name="my\_image" id="pwd">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-success">Click Me </button>

</div>

</div>

</form>

{% if prediction %}

<img src="{{img\_path}}" height="400px" width="400px">

<h1 style="background-color:LightGray;"> The Prediction of image is : <i> </i></h1>

{% endif %}

</div>

</body>

</html>

**Create-API-Using-Flask.py**

import flask

from flask import request, jsonify

#creating ka Flask App

app =flask.Flask(\_\_name\_\_)

phone = [

{'id':0,

'name':'Samsung'},

{'id':1,'name':'iphone'}

]

#Home route

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

def home():

return "<h1>First App<h1>"

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

def api():

return jsonify(phone)

@app.route('/phone/<id>',methods=['GET'])

def api\_id(id):

return jsonify(phone[int(id)])

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

app.run(debug=True)

**python code**

from flask import Flask, render\_template, request

from keras.models import load\_model

from keras.preprocessing import image

app = Flask(\_\_name\_\_)

dic = {0 : 'Cat', 1 : 'Dog'}

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

model.make\_predict\_function()

def predict\_label(img\_path):

i = image.load\_img(img\_path, target\_size=(100,100))

i = image.img\_to\_array(i)/255.0

i = i.reshape(1, 100,100,3)

p = model.predict\_classes(i)

return dic[p[0]]

# routes

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

def main():

return render\_template("index.html")

@app.route("/about")

def about\_page():

return "Please subscribe Artificial Intelligence Hub..!!!"

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

def get\_output():

if request.method == 'POST':

img = request.files['my\_image']

img\_path = "static/" + img.filename

img.save(img\_path)

p = predict\_label(img\_path)

return render\_template("index.html", prediction = p, img\_path = img\_path)

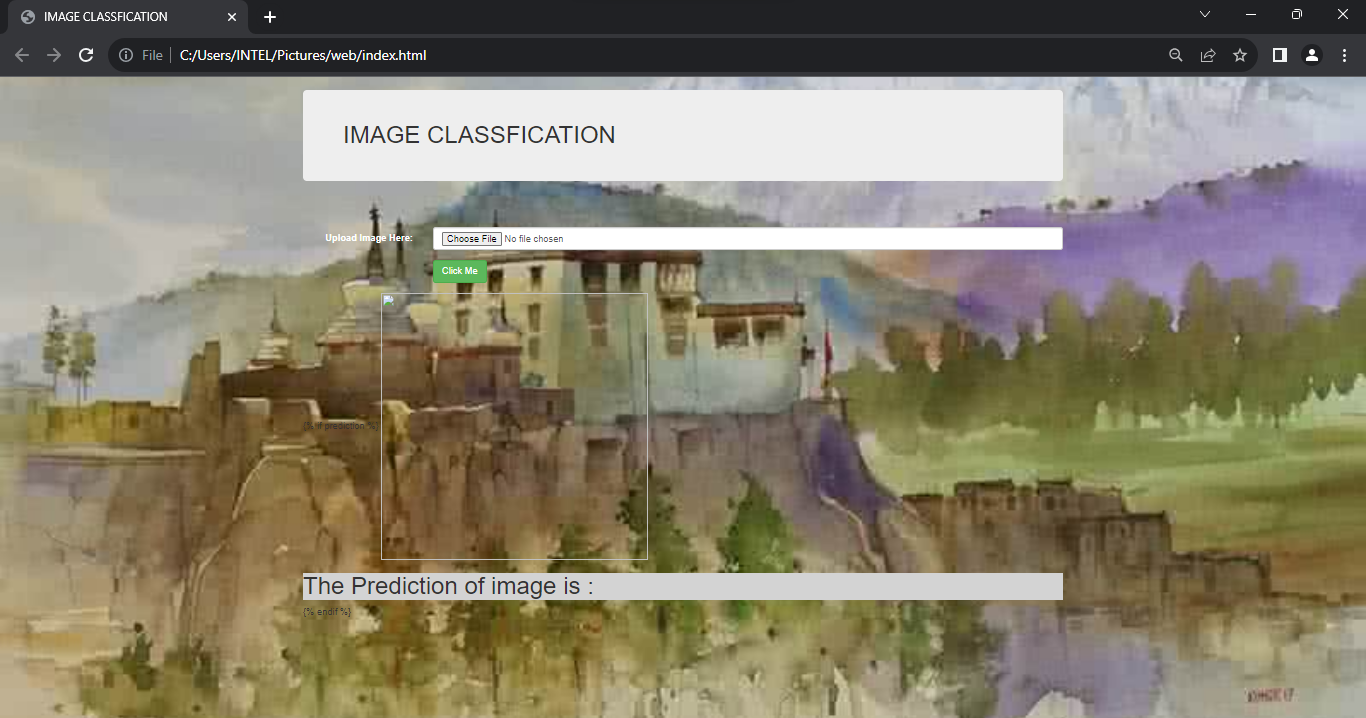
if \_\_name\_\_ =='\_\_main\_\_':

#app.debug = True

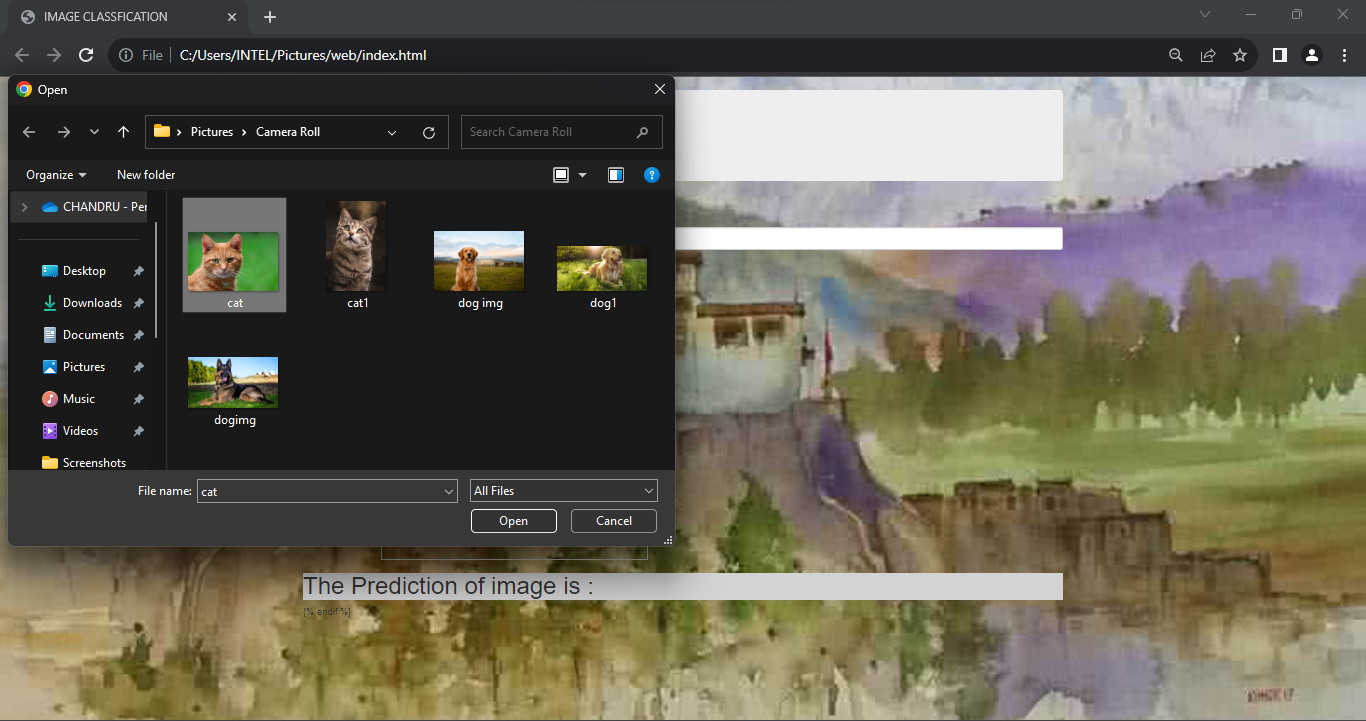
app.run(debug = True)

**Screen shots of our project**:

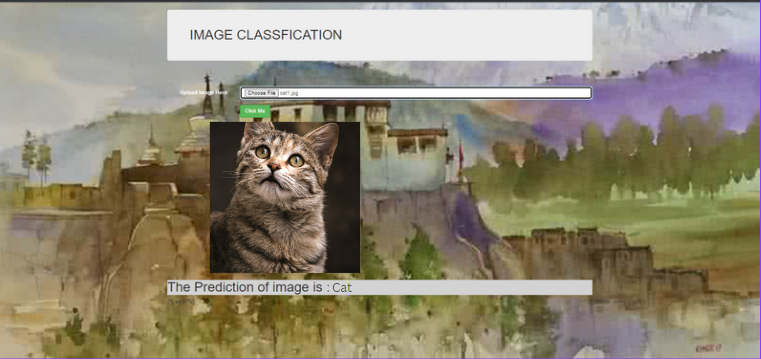
Step 1:



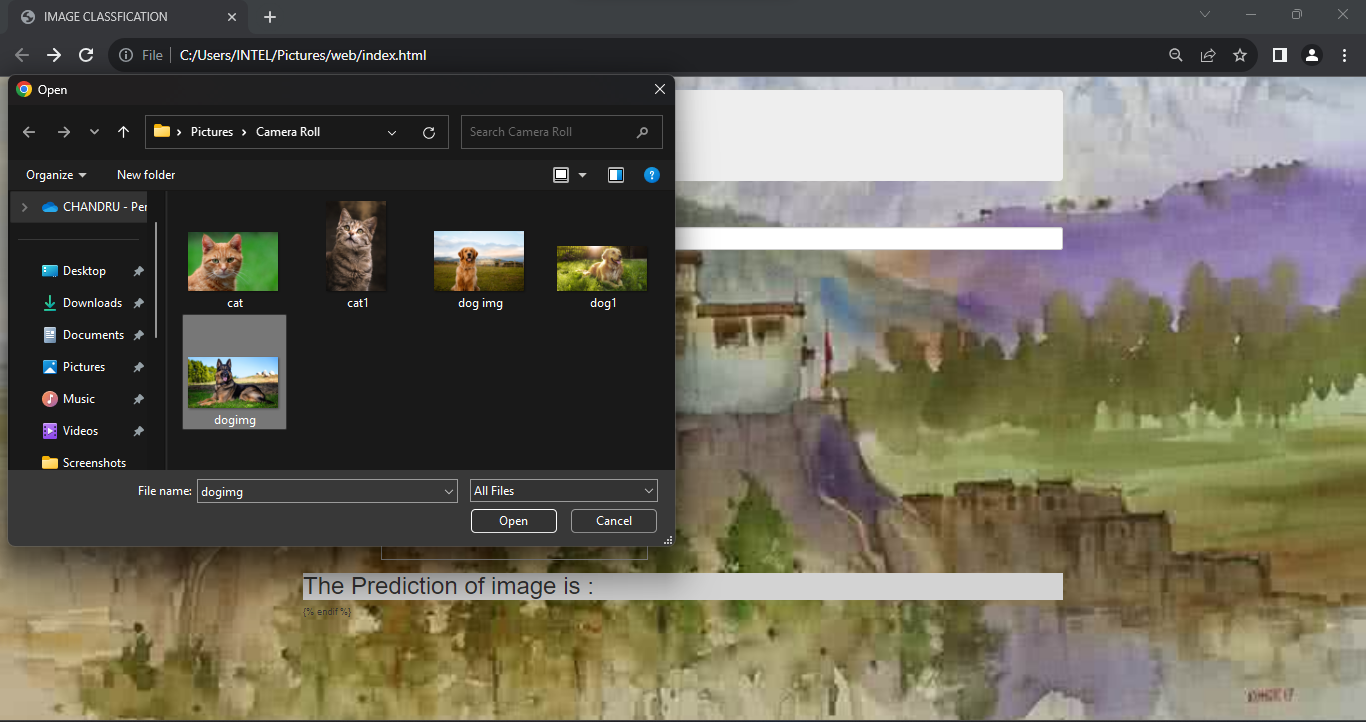
Step 2:



Step 3:



Step 4:



Step 5:

