CS 354N

Minor Project:- Dog Breed Classification

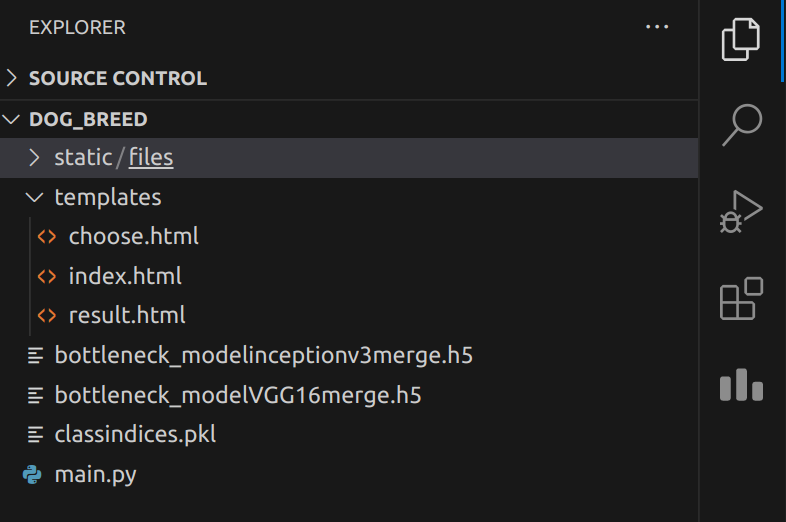
Group members:-

Vansh Kathnawal 200001078

Umang Jain 200001076

Saral Shikhar 200001070

**Implementation of the demo:**

This is the structure of code base

Static folder has some images inside it that contributes in the front end of the website

Create a ‘static’ folder inside that a ‘file’ folder . Then there is a template folder and rest is as shown above.

Templates folder files are as follows

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Home</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

/\* background-color:deeppink; \*/

background-image: url("{{url\_for('static' , filename='bg.jpg')}}");

background-size:100% 100%;

background-repeat: no-repeat;

background-attachment: fixed;

}

h1 {

text-align: center;

font-size: 32px;

margin-top: 40px;

}

h2 {

text-align: left;

font-size: 32px;

margin-top: 100px;

}

form {

display: flex;

flex-direction: column;

align-items: left;

margin-top: 200px;

margin-right: 400px;

margin-left: 40px;

}

label {

font-size: 20px;

margin-top: 20px;

margin-bottom: 10px;

}

input[type="file"] {

border: 2px solid #ccc;

border-radius: 4px;

padding: 10px;

width: 300px;

font-size: 16px;

}

input[type="submit"] {

background-color: #708089;

color: white;

border: none;

border-radius: 4px;

padding: 10px 20px;

font-size: 20px;

margin-top: 20px;

margin-right: 800px;

cursor: pointer;

}

input[type="submit"]:hover {

background-color: #5f6060;

}

#header{

top:0px;

left:0px;

position: fixed;

width:100%;

background-color: rgb(28, 3, 3);

text-align: center;

opacity:0.5;

text-decoration: underline;

text-decoration-color:rgb(0, 0, 0);

font-size: 45px;

}

#footer{

bottom:0px;

left:0px;

background-color: rgb(10, 0, 0);

width: 100%;

text-align: center;

opacity:0.5;

position: fixed;

}

</style>

</head>

<body >

<h1 id="header" style="color :rgba(0, 204, 245, 0.946)"> CS 354N Dog Breed Classification </h1>

<h1 id="footer" style="color :rgb(6, 247, 255)"> Saral Shikhar , Vansh Kathnawal , Umang Jain</h1>

<h2>Insert any Dog image to find it's Breed</h2>

<form method='POST' enctype='multipart/form-data'>

{{form.hidden\_tag()}}

{{form.file()}}

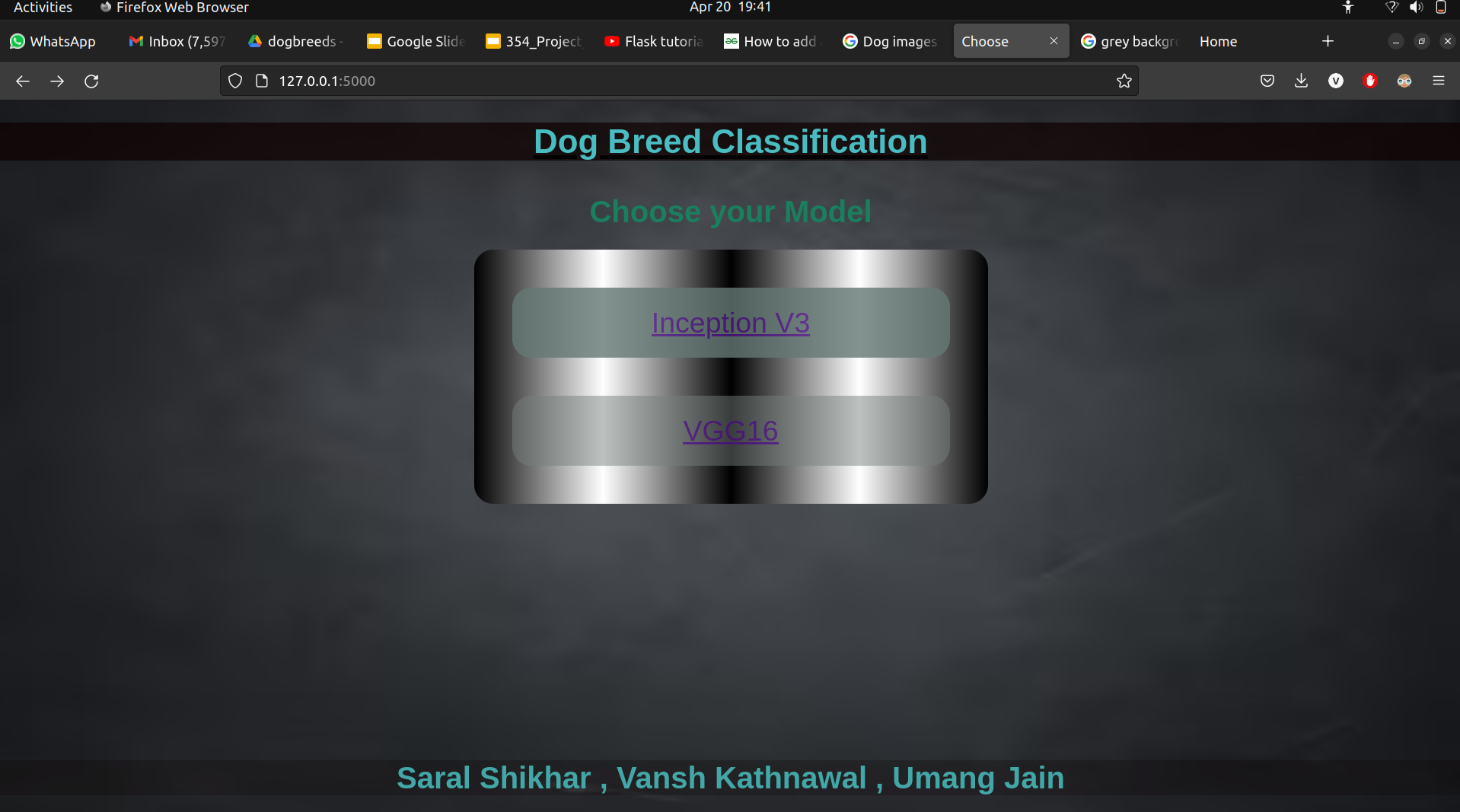
{{form.submit()}}

</form>

</body>

</html>

**Choose.html**

****

<!DOCTYPE html>

<html lang="en" >

<head>

<title>Choose</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

/\* background-color:deeppink; \*/

background-image: url("{{url\_for('static' , filename='grey.jpg')}}");

background-size:100% 100%;

background-repeat: no-repeat;

background-attachment: fixed;

}

#a{grid-area: a1;}

#b{grid-area: b2;}

/\* #c{grid-area: c3;}

#d{grid-area: d4;}

#e{grid-area: e5;} \*/

#maindiv{

/\* display:grid;

width:150%;

grid-template-areas: 'a1 a1 a1 a1 a1 a1'

'b2 b2 b2 c3 c3 c3'

'd4 d4 d4 e5 e5 e5';

background-color:cyan; \*/

display:flex;

flex-direction:column ;

background-image:linear-gradient(90deg,black,white,black ,white,black);

padding:20px;

margin:auto;

width:500px;

border-radius: 200px;

/\* font-family: fantasy; \*/

text-align: center;

/\* text-rendering: geometricPrecision; \*/

}

div{

background-color: rgb(100, 121, 116);

font-size: 30px;

color:rgb(5, 59, 107);

padding:20px;

margin:20px;

border-radius:20px;

}

#a{

opacity:0.8;

}

#b{

background-color: rgba(117, 128, 126, 0.602);

color:black;

opacity:0.8;

}

/\* #e{

background-color: yellow;

} \*/

#header{

top:0px;

left:0px;

position: fixed;

width:100%;

background-color: rgba(13, 0, 0, 0.717);

text-align: center;

text-decoration: underline;

text-decoration-color:rgba(0, 0, 0, 0.933);

font-size: 35px;

}

#footer{

bottom:0px;

left:0px;

background-color: rgba(10, 0, 0, 0.228);

width: 100%;

text-align: center;

position: fixed;

}

.pp{

color:rgb(25, 127, 93)

}

</style>

</head>

<body>

<br>

<br>

<br>

<br>

<h1 align="center" class="pp"> Choose your Model </h1>

<!-- Grid box is combination of rows and columns -->

<div id='maindiv' align="center">

<!-- <div id="a">"Inception V-3"</div><br> -->

<div id="a"> <a href="{{ url\_for('inc') }}">Inception V3</a> </div>

<!-- <div id="b"> "Vgg16 "</div><br> -->

<div id="b"><a href="{{ url\_for('vgg16') }}" >VGG16</a> </div>

<!-- <div id="c"> Audio</div>

<div id="d"> Video</div>

<div id="e"> Status</div> -->

</div>

<h1 id="header" style="color :rgb(73, 189, 193)"> Dog Breed Classification </h1>

<h1 id="footer" style="color :rgb(67, 167, 167)"> Saral Shikhar , Vansh Kathnawal , Umang Jain</h1>

</body>

</html>

**result.html**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Home</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

/\* background-color:deeppink; \*/

background-image: url("{{url\_for('static' , filename='g2.jpg')}}");

background-size:100% 100%;

background-repeat: no-repeat;

background-attachment: fixed;

}

.clss{

text-align:center;

background-color: rgb(100, 121, 116);

font-size: 30px;

color:rgb(5, 6, 7);

padding:20px;

margin:20px;

border-radius:20px;

text-decoration: underline;

text-decoration-color:rgba(0, 0, 0, 0.933);

}

.ig{

align-items:center;

margin-left: 450px;

}

</style>

</head>

<body>

<div class="clss"> {{ txt }}</div>

<div class="ig">

<img src="{{ url\_for('static', filename=img) }}" height="666" width="666">

</div>

</body>

</html>

**—-----------------------------------------------------**

**main.py**

from flask import Flask, render\_template

from flask\_wtf import FlaskForm

from flask import request

from flask import session

from wtforms import FileField, SubmitField

from werkzeug.utils import secure\_filename

import os

from wtforms.validators import InputRequired

#This

from numpy import expand\_dims

from tensorflow.keras.layers import BatchNormalization

from tensorflow import keras

from tensorflow.keras.utils import load\_img

from tensorflow.keras.utils import img\_to\_array

from keras.preprocessing.image import ImageDataGenerator

from matplotlib import pyplot

#This

import numpy as np

from keras.models import Sequential

from keras.layers import Dropout, Flatten, Dense

from keras import applications

from keras.utils.np\_utils import to\_categorical

import matplotlib.pyplot as plt

import math

import cv2

top\_model\_weights\_path\_inc = r'bottleneck\_modelinceptionv3merge.h5'

top\_model\_weights\_path\_vgg = r'bottleneck\_modelVGG16merge.h5'

#useless

import pickle

with open(r'classindices.pkl', 'rb') as f:

class\_dictionary = pickle.load(f)

# model.add(Dense(num\_classes, activation='sigmoid'))

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

app.config['SECRET\_KEY'] = 'supersecretkey'

app.config['UPLOAD\_FOLDER'] = 'static/files'

class UploadFileForm(FlaskForm):

file = FileField("File", validators=[InputRequired()])

submit = SubmitField("Upload File")

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

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

def home():

form = UploadFileForm()

if form.validate\_on\_submit():

file = form.file.data # First grab the file

file.save(os.path.join(os.path.abspath(os.path.dirname(\_\_file\_\_)),app.config['UPLOAD\_FOLDER'],secure\_filename(file.filename))) # Then save the file

img\_rel\_path=os.path.join("files/",secure\_filename(file.filename))

image\_path = os.path.join(os.path.abspath(os.path.dirname(\_\_file\_\_)),app.config['UPLOAD\_FOLDER'],secure\_filename(file.filename))

session["image"]=image\_path

session["img\_rel"]=img\_rel\_path

return render\_template("choose.html",img=image\_path)

# get the prediction label

# print("Image ID: {}, Label: {}".format(inID, label))

# return "The given dog is a {}".format( label)

return render\_template('index.html', form=form)

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

def inc():

image\_path =session.get("image")

rel\_path=session.get("img\_rel")

orig = cv2.imread(image\_path)

image = load\_img(image\_path, target\_size=(224, 224))

image = img\_to\_array(image)

print("[INFO] loading and preprocessing image...")

# image = load\_img(file, target\_size=(224, 224))

image = img\_to\_array(image)

image = image / 255

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

model = applications.inception\_v3.InceptionV3(include\_top=False, weights='imagenet')

# get the bottleneck prediction from the pre-trained VGG16 model

bottleneck\_prediction = model.predict(image)

#inceptionv3

model = Sequential()

model.add(Flatten(input\_shape=bottleneck\_prediction.shape[1:]))

# model.add(Dense(512, activation='relu'))

# model.add(Dropout(0.005))

# model.add(Dense(512, activation='relu'))

# model.add(Dropout(0.05))

# model.add(BatchNormalization())

model.add(Dense(512, activation='relu'))

model.add(Dropout(0.01))

# model.add(BatchNormalization())

model.add(Dense(128, activation='relu'))

model.add(Dropout(0.005))

# model.add(BatchNormalization())

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.005))

# model.add(BatchNormalization())

# model.add(BatchNormalization())

model.add(Dense(70, activation='softmax'))

model.load\_weights(top\_model\_weights\_path\_inc,by\_name=True)

predict\_x = model.predict(bottleneck\_prediction)

classes\_predicted=np.argmax(predict\_x,axis=1)

inID = classes\_predicted[0]

# class\_dictionary = generator\_top.class\_indices

inv\_map = {v: k for k, v in class\_dictionary.items()}

label = inv\_map[inID]

return render\_template("result.html",img=rel\_path,txt="Given dog is a {}".format(label))

@app.route('/vgg16', methods=['GET',"POST"])

def vgg16():

image\_path =session.get("image")

rel\_path=session.get("img\_rel")

orig = cv2.imread(image\_path)

image = load\_img(image\_path, target\_size=(224, 224))

image = img\_to\_array(image)

print("[INFO] loading and preprocessing image...")

# image = load\_img(file, target\_size=(224, 224))

image = img\_to\_array(image)

image = image / 255

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

model = applications.VGG16(include\_top=False, weights='imagenet')

bottleneck\_prediction = model.predict(image)

#inceptionv3

model = Sequential()

model.add(Flatten(input\_shape=bottleneck\_prediction.shape[1:]))

# model.add(Dense(512, activation='relu'))

# model.add(Dropout(0.005))

# model.add(Dense(512, activation='relu'))

# model.add(Dropout(0.05))

# model.add(BatchNormalization())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.05))

model.add(BatchNormalization())

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.05))

# model.add(BatchNormalization())

model.add(Dense(70, activation='softmax'))

model.load\_weights(top\_model\_weights\_path\_vgg,by\_name=True)

predict\_x = model.predict(bottleneck\_prediction)

classes\_predicted=np.argmax(predict\_x,axis=1)

inID = classes\_predicted[0]

# class\_dictionary = generator\_top.class\_indices

inv\_map = {v: k for k, v in class\_dictionary.items()}

label = inv\_map[inID]

return render\_template("result.html",img=rel\_path,txt="Given dog is a {}".format(label))

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

app.run(debug=True)

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Here below is the drive link for the weight files and class indices:**

<https://drive.google.com/drive/folders/13_m1TBZTBrgP4TAlaPJQ7kW0pVDCYN5S>

# **Training of the model:**

<https://colab.research.google.com/drive/1PiqHgJz_7czLe1-6qESAwXtMak8hphMq>

This is the code of the google colab repository where we trained the model and manipulated the data.

# 

# **HOW TO RUN THE PROGRAM:**

As explained above in the dog breed folder create a folder named ‘static’ and inside that a folder named ‘files’.

Then a folder named template

Inside template three files are there choose.html, index.html and result.html

Then there is main.py and weight files which should be there in folder.

Then in terminal just type:**-$ python3 main.py**

**You will get a link in terminal just open it,and you will be directed to the home page**