**Python Training Code for “Sweets From India” Image classification version 1.0.0**

# Import necessary library modules and frameworks into Python.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import os

import glob as gb

import cv2

import tensorflow as tf

import keras

# Check if the machine is ready to use your GPU

from tensorflow.python.client import device\_lib

print('DEVICES', device\_lib.list\_local\_devices())

# Define the paths to your datasets

trainpath='./datasets/sweetsfromindia/train'

testpath='./datasets/sweetsfromindia/test’

# List the number of image files in your datasets

for folder in os.listdir(trainpath):

files=gb.glob(pathname=str(trainpath + '/'+folder + '/\*.jpg'))

print(f"numbers of images {len(files)} in {folder}")

for folder in os.listdir(testpath):

files=gb.glob(pathname=str(testpath + '/'+folder + '/\*.jpg'))

print(f"numbers of images {len(files)} in {folder}")

# Check the image size of each image files in your datasets

size=[]

for folder in os.listdir(trainpath):

files=gb.glob(pathname=str(trainpath + '/'+folder + '/\*.jpg'))

for file in files:

image=plt.imread(file)

size.append(image.shape)

pd.Series(size).value\_counts()

# Define the labels

labels={' kajukatli':0,' gulabjamun':1 ,' Kheer’:2 ,' halwa’:3}

def getcode(n):

for x,y in labels.items():

if n==y:

return x

# Resize the image files to **normalize** the training data

x\_train=[]

y\_train=[]

for folder in os.listdir(trainpath):

files=gb.glob(pathname=str(trainpath + '/'+folder + '/\*.jpg'))

for file in files:

image=cv2.imread(file)

img\_arr=cv2.resize(image,(128,128))

x\_train.append(img\_arr)

y\_train.append(labels[folder])

# Display the training images on your IDE

plt.figure(figsize=(20,20))

for n,i in enumerate(list(np.random.randint(0,len(x\_train),18))):

plt.subplot(3,6,n+1)

plt.imshow(x\_train[i])

plt.axis('off')

plt.title(getcode(y\_train[i]))

# Resize the image files to **normalize** the test data

x\_test=[]

# y\_test=[]

for folder in os.listdir(testpath):

files=gb.glob(pathname=str(testpath + '/'+folder + '/\*.jpg'))

for file in files:

image=cv2.imread(file)

img\_arr=cv2.resize(image,(128,128))

x\_test.append(img\_arr)

# y\_test.append(labels[folder])

# Display the test images on your IDE

plt.figure(figsize=(20,20))

for n,i in enumerate(list(np.random.randint(0,len(x\_test),18))):

plt.subplot(3,6,n+1)

plt.imshow(cv2.cvtColor(x\_test[i], cv2.COLOR\_BGR2RGB))

plt.axis('off')

plt.title(getcode(‘all’))

# Check the size and the number of normalized data

x\_train = np.array(x\_train)

x\_test = np.array(x\_test)

y\_train = np.array(y\_train)

# y\_test = np.array(y\_test)

print(f'X\_train shape is {x\_train.shape}')

print(f'X\_test shape is {x\_test.shape}')

print(f'y\_train shape is {y\_train.shape}’)

# print(f'y\_test shape is {y\_test.shape}')

# Define the training algorythm

sweets=keras.models.Sequential([

keras.layers.Conv2D(200,kernel\_size=(3,3),activation='relu',input\_shape=(128,128,3)),

keras.layers.MaxPool2D((4,4)),

keras.layers.Conv2D(120,kernel\_size=(3,3),activation='relu'),

keras.layers.MaxPool2D((4,4)),

keras.layers.Conv2D(50,kernel\_size=(3,3),activation='relu'),

keras.layers.MaxPool2D((4,4)),

keras.layers.Flatten(),

keras.layers.Dense(120,activation='relu'),

keras.layers.Dense(100,activation='relu'),

keras.layers.Dense(50,activation='relu'),

keras.layers.Dropout(rate=0.5),

keras.layers.Dense(4,activation='softmax'),

])

# Display the summary of the model

sweets.summary()

# Compile the model

sweets.compile(optimizer='adam',

loss='sparse\_categorical\_crossentropy',

metrics=['accuracy’])

# Kick of the training then wait until the training completes

sweets.fit(x\_train, y\_train, epochs=10, batch\_size=64)

# Evaluate the model

sweets.evaluate(x\_train, y\_train)

# Make sure to save the model

sweets.save(‘./models/sweetsfromindia.keras')

# Get prediction ready

y\_pred = sweets.predict(x\_test)

# Select the image test file to perform online prediction

y\_pred = [4]

# Display the test image and get prediction together

Author : Yoshiharu Sato