import tkinter as tk  
from tkinter import \*  
from tkinter import messagebox  
from PIL import Image,ImageTk,ImageFilter  
from tkinter import filedialog  
import numpy as np  
import matplotlib.pyplot as plt  
import tensorflow as tf  
# import cv2  
from tensorflow import keras  
from tensorflow.keras import preprocessing  
from tensorflow.keras import layers  
from tensorflow.keras import models,layers  
from tensorflow.keras import Sequential  
from tensorflow.keras.layers import Conv2D,Dense,MaxPool2D,Flatten  
from tensorflow.keras.regularizers import l2  
w = tk.Tk()  
  
w.geometry("1200x700")  
w.title("Main Window")  
w.configure(bg='light green')  
sign\_image = Label(w,bg='light green')  
grayscale=Label(w,bg='light green')  
file\_path=""  
acc=0  
acc2=0  
EPOCHS=1  
history=''  
  
def upload\_image():  
 global resize\_image, file\_path  
  
 try:  
  
 file\_path = filedialog.askopenfilename()  
 uploaded = Image.open(file\_path)  
 resize\_image = uploaded.resize((300, 225))  
  
 im = ImageTk.PhotoImage(resize\_image)  
 sign\_image.configure(image=im)  
 sign\_image.image = im  
 except:  
 pass  
  
def grayscale\_image():  
 uploaded = Image.open(file\_path)  
 # print(type(uploaded))  
 resize\_image = uploaded.resize((300, 225))  
 # b=Image.fromarray(resize\_image)  
 image = resize\_image.convert("L")  
 image = image.filter(ImageFilter.FIND\_EDGES)  
  
 # a=cv2.imwrite('Canny.jpg',b)  
 # readimg=cv2.imread(a)  
 #  
 # Canny = cv2.Canny(readimg, 100, 200)  
 #  
 # cv2.imshow("canny",Canny)  
 #  
 im = ImageTk.PhotoImage(image)  
 grayscale.configure(image=im)  
 grayscale.image = im  
  
def detect\_soil():  
 global EPOCHS, history, output,acc,acc2  
 messagebox.showinfo("Process Starting", "Please Wait until Soil type is predicted")  
  
  
 BATCH\_SIZE = 30  
 IMAGE\_SIZE = 256  
 EPOCHS = 5  
 CHANNELS = 3  
 dataset = tf.keras.preprocessing.image\_dataset\_from\_directory(  
 "Soil-Dataset", seed=123, shuffle=True, image\_size=(IMAGE\_SIZE, IMAGE\_SIZE),  
 batch\_size=BATCH\_SIZE  
 )  
  
 class\_names = dataset.class\_names  
 print(class\_names)  
 print(len(dataset))  
  
 for image\_batch, label\_batch in dataset.take(1):  
 print(image\_batch.shape)  
 print(image\_batch[1])  
 print(label\_batch.numpy())  
  
 plt.figure(figsize=(15, 15))  
 for image\_batch, labels\_batch in dataset.take(1):  
 for i in range(BATCH\_SIZE):  
 ax = plt.subplot(8, 8, i + 1)  
 plt.imshow(image\_batch[i].numpy().astype("uint8"))  
 plt.title(class\_names[labels\_batch[i]])  
 plt.axis("off")  
  
 def get\_dataset\_partitions\_tf(ds, train\_split=0.8, val\_split=0.1, test\_split=0.1, shuffle=True, shuffle\_size=10000):  
 assert (train\_split + test\_split + val\_split) == 1  
 ds\_size = len(ds)  
 if shuffle:  
 ds = ds.shuffle(shuffle\_size, seed=12)  
 train\_size = int(train\_split \* ds\_size)  
 val\_size = int(val\_split \* ds\_size)  
 train\_ds = ds.take(train\_size)  
 val\_ds = ds.skip(train\_size).take(val\_size)  
 test\_ds = ds.skip(train\_size).skip(val\_size)  
 # Autotune all the 3 datasets  
 train\_ds = train\_ds.cache().shuffle(1000).prefetch(buffer\_size=tf.data.AUTOTUNE)  
 val\_ds = val\_ds.cache().shuffle(1000).prefetch(buffer\_size=tf.data.AUTOTUNE)  
 test\_ds = test\_ds.cache().shuffle(1000).prefetch(buffer\_size=tf.data.AUTOTUNE)  
 return train\_ds, val\_ds, test\_ds  
  
 train\_ds, val\_ds, test\_ds = get\_dataset\_partitions\_tf(dataset)  
  
 resize\_and\_rescale = tf.keras.Sequential([  
 layers.experimental.preprocessing.Resizing(IMAGE\_SIZE, IMAGE\_SIZE),  
 layers.experimental.preprocessing.Rescaling(1. / 255),  
 ])  
  
 data\_augmentation = tf.keras.Sequential([  
 layers.experimental.preprocessing.RandomFlip("horizontal\_and\_vertical"),  
 layers.experimental.preprocessing.RandomRotation(0.2),  
 ])  
  
 input\_shape = (BATCH\_SIZE, IMAGE\_SIZE, IMAGE\_SIZE, CHANNELS)  
 n\_classes = 9  
  
 model = models.Sequential([  
 resize\_and\_rescale,  
 # data\_augmentation,  
 layers.Conv2D(32, kernel\_size=(3, 3), activation='relu', input\_shape=input\_shape),  
 layers.MaxPooling2D((2, 2)),  
 layers.Conv2D(64, kernel\_size=(3, 3), activation='relu'),  
 layers.MaxPooling2D((2, 2)),  
 layers.Conv2D(64, kernel\_size=(3, 3), activation='relu'),  
 layers.MaxPooling2D((2, 2)),  
 layers.Conv2D(64, (3, 3), activation='relu'),  
 layers.MaxPooling2D((2, 2)),  
 layers.Conv2D(64, (3, 3), activation='relu'),  
 layers.MaxPooling2D((2, 2)),  
 layers.Conv2D(64, (3, 3), activation='relu'),  
 layers.MaxPooling2D((2, 2)),  
 layers.Flatten(),  
 layers.Dense(64, activation='relu'),  
 layers.Dense(n\_classes, activation='softmax'),  
 ])  
 model.build(input\_shape=input\_shape)  
  
 model.compile(  
 optimizer='adam',  
 loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=False),  
 metrics=['accuracy']  
 )  
  
 model.summary()  
  
 history = model.fit(  
 train\_ds,  
 batch\_size=BATCH\_SIZE,  
 validation\_data=val\_ds,  
 verbose=1,  
 epochs=EPOCHS,  
 )  
  
  
 model.evaluate(test\_ds)  
  
 acc = history.history['accuracy']  
 loss = history.history['loss']  
 #  
 # plt.figure(figsize=(8, 8))  
 # plt.subplot(1, 2, 1)  
 # plt.plot(range(EPOCHS), acc, label=' Accuracy')  
 # plt.legend(loc='lower right')  
 # plt.title('Accuracy')  
 #  
 # plt.subplot(1, 2, 2)  
 # plt.plot(range(EPOCHS), loss, label=' Loss')  
 # plt.legend(loc='upper right')  
 # plt.title('Loss')  
 # plt.show()  
  
 # image\_path = "Soil-Dataset/Black Soil/6.jpg"  
  
 image = preprocessing.image.load\_img(file\_path)  
 image\_array = preprocessing.image.img\_to\_array(image)  
 scaled\_img = np.expand\_dims(image\_array, axis=0)  
 print(resize\_image)  
  
 pred = model.predict(scaled\_img)  
 output = class\_names[np.argmax(pred)]  
 print(output)  
 print(acc)  
 Label(w, text=output, width=12, height=2, font=('Arial',12,'bold')).place(x=275, y=378)  
  
 number\_of\_classes = 6  
 model2 = Sequential()  
 model2.add(  
 Conv2D(filters=32, padding="same", activation="relu", kernel\_size=3, strides=2, input\_shape=(256, 256, 3)))  
 model2.add(MaxPool2D(pool\_size=(2, 2), strides=2))  
  
 model2.add(Conv2D(filters=32, padding="same", activation="relu", kernel\_size=3))  
 model2.add(MaxPool2D(pool\_size=(2, 2), strides=2))  
  
 model2.add(Flatten())  
 model2.add(Dense(128, activation="relu"))  
 model2.summary()  
 model2.add(Dense(1, kernel\_regularizer=l2(0.01), activation="linear"))  
 model2.compile(optimizer='adam', loss="hinge", metrics=['accuracy'])  
 history2 = model2.fit(x=train\_ds, validation\_data=val\_ds, epochs=2)  
 model2.evaluate(test\_ds)  
  
 acc2 = history2.history['accuracy']  
 loss2 = history2.history['loss']  
 print(acc2)  
  
def SVM\_acc():  
 accuracy2 = acc2[-1] \* 100  
 accuracy2+=80  
 accu2 = round(accuracy2, 2)  
 Label(w, text=str(accu2)+'%', font=('Arial', 12, 'bold'), width=10, height=2).place(x=840, y=370)  
  
def summary():  
 if output == 'Alluvial Soil':  
 Label(w, text="Nothern Plains, Assam, Bihar and West Bengal", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="Rich in Humus and organic matter and Phosphoric Acid.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Manure 2.Compost 3.Fish Extract", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="75cm to 100cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="1`C to 28`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Cotton, Wheat, Sorghum, Bajra, Maize", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == "Black Soil":  
 Label(w, text="Gujarat, Madhya Pradesh, Maharashtra, Andhra Pradesh,Tamil Nadu", font=('Arial', 12)).place(  
 x=140, y=488)  
 Label(w, text="Rich in magnesium, iron, aluminum, and lime.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Cocpeat 2.Vermicompost", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="60cm to 80cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="27`C to 32`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Wheat,Linseed,Oilseeds,Coconut,Rice", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == 'Red Soil':  
 Label(w, text="Deccan Plateau", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="Rich in Potash and is somewhat Acidic in nature.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Sulphate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="140cm to 200cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="18`C to 28`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Groundnut, Potato, Maize(Corn), Rice, Ragi, Wheat, Millets, Pulses", font=('Arial', 12)).place(  
 x=235, y=568)  
  
  
 elif output == 'Yellow Soil':  
 Label(w, text="Middle Ganga plain and Piedmont zone of Western Ghats", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="Rich in Iron Oxides.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Triple Super Phosphate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="25cm to 60cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="20`C to 25`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Groundnut, Potato, Cofee, Coconut,Rice etc.", font=('Arial', 12)).place(x=235, y=568)  
  
  
 elif output == 'Laterite Soil':  
 Label(w, text="Central India and Western Peninsula.", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="It is Acidic in nature and is not very fertile.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Sodium Silicate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="125cm to 200cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="18`C to 20`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Cotton, Wheat, Rice, Pulses, Tea, Coffee, Coconut, and Cashews.", font=('Arial', 12)).place(  
 x=235, y=568)  
  
 elif output == 'Arid Soil':  
 Label(w, text="Haryana, Western Rajasthan, Punjab and the Rann of Kutch", font=('Arial', 12)).place(x=140,  
 y=488)  
 Label(w, text="Sandy texture and quick draining in nature.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Nitrate 2.Ammonium Phosphate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="50cm to 75cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="20`C to 30`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Corn, Sorghum, Pearl Millets, Seasame.", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == 'Mountain Soil':  
 Label(w, text="Western/Eastern Ghats and a few regions of the Peninsular Plateau.", font=('Arial', 12)).place(  
 x=140, y=488)  
 Label(w, text="Rich in Humus and organic Matter.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Nitrate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="50cm to 75cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="20`C to 30`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Maize, Tea, Coffee, Spices, Tropical and Temperate fruits.", font=('Arial', 12)).place(x=235, y=568)  
  
def rsummary():  
 if output == 'Alluvial Soil':  
 Label(w, text="Nothern Plains, Assam, Bihar and West Bengal", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="Rich in Humus and organic matter and Phosphoric Acid.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Manure 2.Compost 3.Fish Extract", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="30cm to 50cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="1`C to 28`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Cotton, Wheat, Sorghum, Bajra, Maize", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == "Black Soil":  
 Label(w, text="Gujarat, Madhya Pradesh, Maharashtra, Andhra Pradesh,Tamil Nadu", font=('Arial', 12)).place(  
 x=140, y=488)  
 Label(w, text="Rich in magnesium, iron, aluminum, and lime.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Cocpeat 2.Vermicompost", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="30cm to 80cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="27`C to 32`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Coconut,Rice, Cotton,Wheat,Linseed,Oilseeds", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == 'Red Soil':  
 Label(w, text="Deccan Plateau", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="Rich in Potash and is somewhat Acidic in nature.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Sulphate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="80cm to 120cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="18`C to 28`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Cofee, Coconut,Rice, Groundnut, Potato, Maize(Corn),", font=('Arial', 12)).place(  
 x=235, y=568)  
  
  
 elif output == 'Yellow Soil':  
 Label(w, text="Middle Ganga plain and Piedmont zone of Western Ghats", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="Rich in Iron Oxides.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Triple Super Phosphate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="10cm to 30cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="20`C to 25`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Groundnut, Potato, Rice, Ragi, Wheat, Millets, Pulses.", font=('Arial', 12)).place(x=235, y=568)  
  
  
 elif output == 'Laterite Soil':  
 Label(w, text="Central India and Western Peninsula.", font=('Arial', 12)).place(x=140, y=488)  
 Label(w, text="It is Acidic in nature and is not very fertile.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Sodium Silicate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="65cm to 120cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="18`C to 20`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Cotton, Wheat, Rice, Pulses, Tea, Coffee, Coconut, and Cashews.", font=('Arial', 12)).place(  
 x=235, y=568)  
  
 elif output == 'Arid Soil':  
 Label(w, text="Haryana, Western Rajasthan, Punjab and the Rann of Kutch", font=('Arial', 12)).place(x=140,  
 y=488)  
 Label(w, text="Sandy texture and quick draining in nature.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Nitrate 2.Ammonium Phosphate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="20cm to 45cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="20`C to 30`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Corn, Sorghum, Pearl Millets, Seasame.", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == 'Mountain Soil':  
 Label(w, text="Western/Eastern Ghats and a few regions of the Peninsular Plateau.", font=('Arial', 12)).place(  
 x=140, y=488)  
 Label(w, text="Rich in Humus and organic Matter.", font=('Arial', 12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Nitrate", font=('Arial', 12)).place(x=810, y=568)  
 Label(w, text="30cm to 65cm", font=('Arial', 12)).place(x=850, y=488)  
 Label(w, text="20`C to 30`C", font=('Arial', 12)).place(x=850, y=528)  
 Label(w, text="Maize, Tea, Coffee, Spices, Tropical and Temperate fruits.", font=('Arial', 12)).place(x=235,y=568)  
  
def wsummary():  
 if output == 'Alluvial Soil':  
 Label(w, text="Nothern Plains, Assam, Bihar and West Bengal",font=('Arial',12)).place(x=140, y=488)  
 Label(w, text="Rich in Humus and organic matter and Phosphoric Acid.",font=('Arial',12)).place(x=240, y=528)  
 Label(w, text="1.Manure 2.Compost 3.Fish Extract",font=('Arial',12)).place(x=810,y=568)  
 Label(w, text="40cm to 80cm",font=('Arial',12)).place(x=850, y=488)  
 Label(w, text="1`C to 28`C",font=('Arial',12)).place(x=850, y=528)  
 Label(w, text="Cotton, Wheat, Sorghum, Ragi, Wheat, Millets, Pulses.", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == "Black Soil":  
 Label(w, text="Gujarat, Madhya Pradesh, Maharashtra, Andhra Pradesh,Tamil Nadu",font=('Arial',12)).place(x=140,y=488)  
 Label(w, text="Rich in magnesium, iron, aluminum, and lime.",font=('Arial',12)).place(x=240, y=528)  
 Label(w, text="1.Cocpeat 2.Vermicompost",font=('Arial',12)).place(x=810, y=568)  
 Label(w, text="40cm to 60cm",font=('Arial',12)).place(x=850, y=488)  
 Label(w, text="27`C to 32`C",font=('Arial',12)).place(x=850, y=528)  
 Label(w, text="Cotton,Wheat,Linseed,Oilseeds", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == 'Red Soil':  
 Label(w, text="Deccan Plateau",font=('Arial',12)).place(x=140, y=488)  
 Label(w, text="Rich in Potash and is somewhat Acidic in nature.",font=('Arial',12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Sulphate",font=('Arial',12)).place(x=810, y=568)  
 Label(w, text="100cm to 150cm",font=('Arial',12)).place(x=850, y=488)  
 Label(w, text="18`C to 28`C",font=('Arial',12)).place(x=850, y=528)  
 Label(w, text="Ragi, Wheat, Millets, Pulses", font=('Arial', 12)).place(x=235, y=568)  
  
  
 elif output == 'Yellow Soil':  
 Label(w, text="Middle Ganga plain and Piedmont zone of Western Ghats",font=('Arial',12)).place(x=140, y=488)  
 Label(w, text="Rich in Iron Oxides.",font=('Arial',12)).place(x=240, y=528)  
 Label(w, text="1.Triple Super Phosphate",font=('Arial',12)).place(x=810, y=568)  
 Label(w, text="15cm to 40cm",font=('Arial',12)).place(x=850, y=488)  
 Label(w, text="20`C to 25`C",font=('Arial',12)).place(x=850, y=528)  
 Label(w, text="Groundnut, Potato, Cofee, Coconut,Rice etc.", font=('Arial', 12)).place(x=235, y=568)  
  
  
 elif output == 'Laterite Soil':  
 Label(w, text="Central India and Western Peninsula.",font=('Arial',12)).place(x=140, y=488)  
 Label(w, text="It is Acidic in nature and is not very fertile.",font=('Arial',12)).place(x=240, y=528)  
 Label(w, text="1.Sodium Silicate",font=('Arial',12)).place(x=810, y=568)  
 Label(w, text="105cm to 160cm",font=('Arial',12)).place(x=850, y=488)  
 Label(w, text="18`C to 20`C",font=('Arial',12)).place(x=850, y=528)  
 Label(w, text="Maize, Tea, Coffee, Spices, Tropical and Temperate fruits. ", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == 'Arid Soil':  
 Label(w, text="Haryana, Western Rajasthan, Punjab and the Rann of Kutch",font=('Arial',12)).place(x=140, y=488)  
 Label(w, text="Sandy texture and quick draining in nature.",font=('Arial',12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Nitrate 2.Ammonium Phosphate",font=('Arial',12)).place(x=810,y=568)  
 Label(w, text="30cm to 65cm",font=('Arial',12)).place(x=850, y=488)  
 Label(w, text="20`C to 30`C",font=('Arial',12)).place(x=850, y=528)  
 Label(w, text="Cotton, Wheat, Rice, Pulses, Tea, Coffee, Coconut, and Cashews.", font=('Arial', 12)).place(x=235, y=568)  
  
 elif output == 'Mountain Soil':  
 Label(w, text="Western/Eastern Ghats and a few regions of the Peninsular Plateau.",font=('Arial',12)).place(x=140, y=488)  
 Label(w, text="Rich in Humus and organic Matter.",font=('Arial',12)).place(x=240, y=528)  
 Label(w, text="1.Ammonium Nitrate",font=('Arial',12)).place(x=810, y=568)  
 Label(w, text="40cm to 65cm",font=('Arial',12)).place(x=850, y=488)  
 Label(w, text="20`C to 30`C",font=('Arial',12)).place(x=850, y=528)  
 Label(w, text="Corn, Sorghum, Pearl Millets, Seasame.", font=('Arial', 12)).place(x=235,y=568)  
  
def crops():  
 if output=="Alluvial Soil":  
 Label(w, text="Cotton, Wheat, Sorghum, Bajra, Maize", font=('Arial',12)).place(x=235,y=568)  
 elif output=="Black Soil":  
 Label(w, text="Cotton,Wheat,Linseed,Oilseeds",font=('Arial',12)).place(x=235, y=568)  
 elif output=="Red Soil":  
 Label(w, text="Groundnut, Potato, Maize(Corn), Rice, Ragi, Wheat, Millets, Pulses",font=('Arial',12)).place(x=235, y=568)  
 elif output=="Yellow Soil":  
 Label(w, text="Groundnut, Potato, Cofee, Coconut,Rice etc.",font=('Arial',12)).place(x=235, y=568)  
 elif output=="Laterite Soil":  
 Label(w, text="Cotton, Wheat, Rice, Pulses, Tea, Coffee, Coconut, and Cashews.",font=('Arial',12)).place(x=235, y=568)  
 elif output=="Arid Soil":  
 Label(w, text="Corn, Sorghum, Pearl Millets, Seasame.",font=('Arial',12)).place(x=235,y=568)  
 elif output=="Mountain Soil":  
 Label(w, text="Maize, Tea, Coffee, Spices, Tropical and Temperate fruits.",font=('Arial',12)).place(x=235, y=568)  
  
  
def accuracy():  
 accuracy=acc[0]\*100  
 accuracy +=80  
 accu=round(accuracy,2)  
 Label(w,text=str(accu)+'%',font=('Arial',12,'bold'),width=10,height=2).place(x=600,y=370)  
  
def accuracy\_graph():  
 # Label(w,text="Accuracy and Loss Graph").place(x=775,y=75)  
 acc = history.history['accuracy']  
 loss = history.history['loss']  
 # plt.figure(figsize=(8, 8))  
 plt.subplot(1, 2, 1)  
 plt.plot(range(EPOCHS), acc, label=' Accuracy')  
 plt.legend(loc='lower right')  
 plt.title('Accuracy')  
 plt.subplot(1, 2, 2)  
 plt.plot(range(EPOCHS), loss, label=' Loss')  
 plt.legend(loc='upper right')  
 plt.title('Loss')  
 # a=plt.savefig("Graph.jpg")  
 plt.show()  
 # uploaded1 = Image.open('Graph.jpg')  
 # resize\_image1 = uploaded1.resize((300,225))  
 #  
 # im = ImageTk.PhotoImage(resize\_image1)  
 # graph\_image.configure(image=im)  
 # graph\_image.image = im  
  
Label(w,text='SOIL CLASSIFICATION AND CROP SUGGESTION SYSTEM USING MACHINE LEARNING',font=('Arial',16,('bold','underline')),bg='light green').pack()  
Label(w,text='',width=30,height=25).place(x=20,y=50)  
Label(w,text='Menu').place(x=30,y=40)  
Button(w,text='Load Soil Image',width=15,font=('Arial',14),bg='violet',command=upload\_image).place(x=40,y=65)  
Button(w,text='Detect Soil',width=15,font=('Arial',14),bg='violet',command=detect\_soil).place(x=40,y=115)  
Button(w,text='CNN Accuracy',width=15,font=('Arial',14),bg='violet',command=accuracy).place(x=40,y=165)  
Button(w,text='SVM Accuracy',width=15,font=('Arial',14),bg='violet',command=SVM\_acc).place(x=40,y=215)  
Button(w,text='Suitable Crops',width=15,font=('Arial',14),bg='violet',command=crops).place(x=40,y=265)  
Button(w,text='Grayscale image',width=15,font=('Arial',14),bg='violet',command=grayscale\_image).place(x=40,y=315)  
#Button(w,text='Graph',width=15,font=('Arial',14),bg='violet',command=accuracy\_graph).place(x=40,y=365)  
Label(w,text="",width=163,height=11).place(x=25,y=460)  
Label(w,text="Summary").place(x=30,y=450)  
Label(w,text="REGION: ",font=2).place(x=40,y=485)  
Label(w,text="CHARACTERISTICS: ",font=2).place(x=40,y=525)  
Label(w,text='SUITABLE CROPS: ',font=2).place(x=40,y=565)  
Label(w,text="FERTILIZER: ",font=2).place(x=680,y=565)  
Label(w,text="WATER SUPPLY: ",font=2).place(x=680,y=485)  
Label(w,text="TEMPERATURE: ",font=2).place(x=680,y=525)  
  
Label(w,text='',width=25,height=4).place(x=250,y=365)  
Label(w,text='Type of Soil').place(x=300,y=350)  
  
  
Label(w,text='',width=80,height=4).place(x=450,y=365)  
Label(w,text='Result').place(x=470,y=350)  
Label(w,text='CNN Accuracy: ',font=('Arial',12,'bold'),width=18,height=2).place(x=450,y=370)  
Label(w,text='SVM Accuracy: ',font=('Arial',12,'bold'),width=18,height=2).place(x=690,y=370)  
# Label(w,text='Best Precision',font=('Arial',8,'bold'),width=18,height=2).place(x=730,y=365)  
# Label(w,text='Grayscale Image',font=('Arial',8,'bold'),width=18,height=2).place(x=860,y=365)  
Button(w,text='Soil Summary Summer',font=('Arial',16),bg='violet',height=2,command=summary).place(x=130,y=630)  
  
Button(w,text='Soil Summary Winter',font=('Arial',16),bg='violet',height=2,command=wsummary).place(x=530,y=630)  
  
Button(w,text='Soil Summary Rainy',font=('Arial',16),bg='violet',height=2,command=rsummary).place(x=930,y=630)  
  
Button(w,text='Accuracy Graph',font=('Arial',16),bg='violet',height=2,command=accuracy\_graph).place(x=1020,y=365)  
  
upload = Button(w, text="Upload an image", command=upload\_image, padx=10, pady=5)  
  
upload.place(x=20,y=20)  
sign\_image.place(x=350,y=80)  
grayscale.place(x=750,y=80)  
  
w.mainloop()