from telegram.ext import Updater, Filters, CommandHandler, MessageHandler

import cv2

from tensorflow.keras.applications.resnet50 import ResNet50

import numpy as np

from labels import lbl

from gtts import gTTS

import os

model = ResNet50()

def start(updater, context):

    updater.message.reply\_text("Welcome to the classification bot!")

def help\_(updater, context):

    updater.message.reply\_text("Just send the image you want to classify.")

def message(updater, context):

    msg = updater.message.text

    print(msg)

    updater.message.reply\_text(msg)

def image(updater, context):=

    photo = updater.message.photo[-1].get\_file()

    photo.download("img.jpg")

    img = cv2.imread("img.jpg")

    img = cv2.resize(img, (224,224))

    img = np.reshape(img, (1,224,224,3))

    pred = np.argmax(model.predict(img))

    pred = lbl[pred]

    print(f"The predicted output is {pred}")

    # Sending text response

    updater.message.reply\_text(pred)

    # Creating and sending voice response

    tts = gTTS(text=pred, lang='en')

    tts.save("response.mp3")

    with open("response.mp3", 'rb') as audio:

        updater.message.reply\_voice(audio)

    # Cleaning up the audio file

    os.remove("response.mp3")

updater = Updater("6792165809:AAFAqJTYTQHc5cpsV04JQ-eoTuGFb6lKAA0")

dispatcher = updater.dispatcher

dispatcher.add\_handler(CommandHandler("start", start))

dispatcher.add\_handler(CommandHandler("help", help\_))

dispatcher.add\_handler(MessageHandler(Filters.text, message))

dispatcher.add\_handler(MessageHandler(Filters.photo, image))

updater.start\_polling()

updater.idle()