import cv2

import mediapipe as mp

import tkinter as tk

from tkinter import Label

from PIL import Image, ImageTk

import threading

# Initialize Mediapipe

mp\_hands = mp.solutions.hands

mp\_drawing = mp.solutions.drawing\_utils

# Global flag to control loop

running = True

class SignLanguageApp:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Sign2Speak - ASL Translator")

# Create UI elements

self.video\_label = Label(root)

self.video\_label.pack()

self.translation\_label = Label(root, text="Predicted Gesture: (placeholder)", font=("Arial", 16))

self.translation\_label.pack(pady=10)

# Start video thread

self.capture = cv2.VideoCapture(0)

self.hands = mp\_hands.Hands(max\_num\_hands=1, min\_detection\_confidence=0.7)

self.video\_loop()

self.root.protocol("WM\_DELETE\_WINDOW", self.on\_closing)

def video\_loop(self):

global running

if running:

ret, frame = self.capture.read()

if ret:

# Flip for selfie-view and convert to RGB

frame = cv2.flip(frame, 1)

rgb = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

results = self.hands.process(rgb)

# Draw landmarks

if results.multi\_hand\_landmarks:

for hand\_landmarks in results.multi\_hand\_landmarks:

mp\_drawing.draw\_landmarks(frame, hand\_landmarks, mp\_hands.HAND\_CONNECTIONS)

# Placeholder gesture prediction logic

# gesture = self.classify\_gesture(hand\_landmarks)

gesture = "A" # mock output

self.translation\_label.config(text=f"Predicted Gesture: {gesture}")

# Convert to ImageTk

img = Image.fromarray(cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB))

imgtk = ImageTk.PhotoImage(image=img)

self.video\_label.imgtk = imgtk

self.video\_label.configure(image=imgtk)

# Repeat

self.root.after(10, self.video\_loop)

def on\_closing(self):

global running

running = False

self.capture.release()

self.root.destroy()

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = SignLanguageApp(root)

root.mainloop()