import cvzone  
import cv2  
import numpy as np  
from cvzone.HandTrackingModule import HandDetector  
import google.generativeai as genai  
from PIL import Image  
  
genai.configure(api\_key="AIzaSyDTbsVvJFh\_rpE7ReUDh8AdEBj73dUqkC0")  
model = genai.GenerativeModel('gemini-1.5-flash')  
  
# Initialize the webcam to capture video  
# The '2' indicates the third camera connected to your computer; '0' would usually refer to the built-in camera  
cap = cv2.VideoCapture(0)  
cap.set(propId=3,value=1280)  
cap.set(propId=4,value=720)  
# Initialize the HandDetector class with the given parameters  
detector = HandDetector(staticMode=False, maxHands=1, modelComplexity=1, detectionCon=0.7, minTrackCon=0.5)  
  
def getHandInfo(img):  
 # Check if any hands are detected  
 hands,img=detector.findHands(img, draw=True, flipType=True)  
 if hands:  
 # Information for the first hand detected  
 hand1 = hands[0] # Get the first hand detected  
 lmList = hand1["lmList"] # List of 21 landmarks for the first hand  
  
  
 # Count the number of fingers up for the first hand  
 fingers = detector.fingersUp(hand1)  
 print(fingers)  
 return fingers, lmList  
 else:  
 return None  
  
  
def draw(info, prev\_pos, canvas):  
 fingers, lmList = info  
 current\_pos = None  
 if fingers == [0, 1, 0, 0, 0]:  
 current\_pos = lmList[8][0:2]  
 if prev\_pos is None:  
 prev\_pos = current\_pos  
 cv2.line(canvas, prev\_pos, current\_pos, (255, 0, 255), 10)  
 elif fingers==[1,1,1,1,1]:  
 canvas=np.zeros\_like(img)  
 # Update the previous position to the current position  
 return current\_pos, canvas  
def sendToAI(model,canvas,fingers):  
 if fingers==[1,1,1,1,0]:  
 pil\_image=Image.fromarray(canvas)  
 response = model.generate\_content(["Solve this Math Problem",pil\_image])  
  
 print(response.text)  
  
  
prev\_pos=None  
canvas=None  
image\_combined=None  
while True:  
 # Capture each frame from the webcam  
 # 'success' will be True if the frame is successfully captured, 'img' will contain the frame  
 success, img = cap.read()  
 img=cv2.flip(img,flipCode=1)  
  
 if canvas is None:  
 canvas=np.zeros\_like(img)  
  
 info=getHandInfo(img)  
 if info:  
 fingers, imList=info  
 print(fingers)  
  
 prev\_pos= draw(info, prev\_pos,canvas)  
 sendToAI(model,canvas,fingers)  
 image\_combined=cv2.addWeighted(img,0.7,canvas,0.3,0)  
  
 # Check if a second hand is detected  
  
 # Display the image in a window  
 cv2.imshow("Image", img)  
 cv2.imshow("Canvas",canvas)  
 cv2.imshow("image\_combined", image\_combined)  
 # Keep the window open and update it for each frame; wait for 1 millisecond between frames  
 cv2.waitKey(1)