import os  
  
import cv2  
  
faceProto = "opencv\_face\_detector.pbtxt"  
faceModel = "opencv\_face\_detector\_uint8.pb"  
ageProto = "age\_deploy.prototxt"  
ageModel = "age\_net.caffemodel"  
genderProto = "gender\_deploy.prototxt"  
genderModel = "gender\_net.caffemodel"  
  
MODEL\_MEAN\_VALUES = (78.4263377603, 87.7689143744, 114.895847746)  
ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']  
genderList = ['Male', 'Female']  
  
faceNet = cv2.dnn.readNet(faceModel, faceProto)  
ageNet = cv2.dnn.readNet(ageModel, ageProto)  
genderNet = cv2.dnn.readNet(genderModel, genderProto)  
  
padding = 20  
  
  
def highlightFace(net, frame, conf\_threshold=0.7):  
 frameOpencvDnn = frame.copy()  
 h, w, \_ = frameOpencvDnn.shape  
 blob = cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)  
  
 net.setInput(blob)  
 detections = net.forward()  
 boundBoxes = []  
 for i in range(detections.shape[2]):  
 confidence = detections[0, 0, i, 2]  
 if confidence > conf\_threshold:  
 x1 = int(detections[0, 0, i, 3] \* w)  
 y1 = int(detections[0, 0, i, 4] \* h)  
 x2 = int(detections[0, 0, i, 5] \* w)  
 y2 = int(detections[0, 0, i, 6] \* h)  
 boundBoxes.append([x1, y1, x2, y2])  
 cv2.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0), int(round(h / 150)), 8)  
 return frameOpencvDnn, boundBoxes  
  
  
cap = cv2.VideoCapture(0)  
  
while True:  
 \_, frame = cap.read()  
  
 resultant, boundBoxes = highlightFace(faceNet, frame)  
 if not boundBoxes:  
 print("No face detected")  
  
 for faceBox in boundBoxes:  
 face = frame[max(0, faceBox[1] - padding):  
 min(faceBox[3] + padding, frame.shape[0] - 1), max(0, faceBox[0] - padding)  
 :min(faceBox[2] + padding, frame.shape[1] - 1)]  
  
 blob = cv2.dnn.blobFromImage(face, 1.0, (227, 227), MODEL\_MEAN\_VALUES, swapRB=False)  
 genderNet.setInput(blob)  
 gender\_pred = genderNet.forward()  
 gender = genderList[gender\_pred[0].argmax()]  
 print(f'Gender: {gender}')  
  
 ageNet.setInput(blob)  
 age\_pred = ageNet.forward()  
 age = ageList[age\_pred[0].argmax()]  
 print(f'Age: {age[1:-1]} years')  
  
 cv2.putText(resultant, f'{gender}, {age}', (faceBox[0], faceBox[1] - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8,  
 (0, 255, 255), 2, cv2.LINE\_AA)  
 cv2.imshow("Detection", resultant)  
 if cv2.waitKey(1) & 0xFF == ord('q'):  
 break  
cap.release()  
cv2.destroyAllWindows()