import cv2

# Load pre-trained face detection model (Haar cascade)

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

# Load pre-trained face recognition model

face\_recognizer = cv2.face.LBPHFaceRecognizer\_create()

# Load trained face recognition model weights

face\_recognizer.read("face\_recognition\_model.xml")

# Load image or video

input\_source = "input\_video.mp4" # Change this to your input source

cap = cv2.VideoCapture(input\_source)

while True:

# Read frame from video

ret, frame = cap.read()

if not ret:

break

# Convert frame to grayscale for face detection

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# Detect faces in the frame

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

for (x, y, w, h) in faces:

# Draw rectangle around detected face

cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

# Extract face region

face\_roi = gray[y:y+h, x:x+w]

# Recognize face

label, confidence = face\_recognizer.predict(face\_roi)

# Display recognized label and confidence

if confidence < 100:

text = f"Person: {label} Confidence: {round(100 - confidence)}"

else:

text = "Unknown"

cv2.putText(frame, text, (x, y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (36,255,12), 2)

# Display output frame

cv2.imshow('Face Recognition', frame)

# Press 'q' to exit

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Release video capture and close windows

cap.release()

cv2.destroyAllWindows()