import cv2

from tensorflow.keras.models import load\_model

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

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

# Load pre-trained emotion classification model (TensorFlow CNN)

emotion\_model = load\_model('emotion\_classification\_model.h5')

# Define emotion labels (modify based on model's output classes)

emotion\_labels = ['angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', 'surprise']

def detect\_and\_classify\_emotions(image):

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY) # Convert to grayscale

# Detect faces

faces = face\_cascade.detectMultiScale(gray, 1.1, 4)

# Process each detected face

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

# Extract the face region of interest (ROI)

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

# Preprocess the face ROI for emotion classification model (resize, normalize)

face\_roi = cv2.resize(face\_roi, (48, 48)) # Adjust dimensions as needed by your model

face\_roi = face\_roi.astype('float32') / 255.0

face\_roi = np.expand\_dims(face\_roi, axis=0) # Add extra dimension for batch processing

# Predict emotion using loaded model

predictions = emotion\_model.predict(face\_roi)

predicted\_emotion = emotion\_labels[np.argmax(predictions)]

# Draw bounding box and emotion label on the image

cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 2) # Green bounding box

cv2.putText(image, predicted\_emotion, (x, y - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.7, (0, 255, 0), 2)

return image

# Example

result\_image = detect\_and\_classify\_emotions(image.copy())

# Avoid modifying the original image

cv2.imshow('Emotion Detection', result\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()