Facial Expression

July 11, 2024

[ ]:
```
#Facial expression recognition in real-time from a web camera_Palanichamy Naveen

#pip install tensorflow opencv-python

import cv2
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.image import img_to_array

# Load the pre-trained model
model = tf.keras.models.load_model('emotion_model.hdf5', compile=False)
model.compile(optimizer='adam', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])

# Define the emotion labels
emotion_labels = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise',␣
 ↪'Neutral']

# Start the video capture from the web camera
cap = cv2.VideoCapture(0)

while True:
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Convert the frame to grayscale
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Load OpenCV's Haar Cascade for face detection
    face_detector = cv2.CascadeClassifier(cv2.data.haarcascades +␣
 ↪'haarcascade_frontalface_default.xml')

    # Detect faces in the grayscale frame
    faces = face_detector.detectMultiScale(gray, scaleFactor=1.3,␣
 ↪minNeighbors=5)
```

```
    # Loop through the detected faces
    for (x, y, w, h) in faces:
        # Extract the region of interest (the face) and resize it to the␣
↪model's input size
        roi_gray = gray[y:y+h, x:x+w]
        roi_gray = cv2.resize(roi_gray, (64, 64))
        roi_gray = roi_gray.astype('float') / 255.0
        roi_gray = img_to_array(roi_gray)
        roi_gray = np.expand_dims(roi_gray, axis=-1)
        roi_gray = np.expand_dims(roi_gray, axis=0)

        # Predict the emotion
        preds = model.predict(roi_gray)[0]
        emotion_probability = np.max(preds)
        label = emotion_labels[preds.argmax()]

        # Draw a rectangle around the face and put the emotion label
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
        cv2.putText(frame, label, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9,␣
↪(255, 0, 0), 2)

    # Display the resulting frame
    cv2.imshow('Facial Expression Recognition', frame)

    # Break the loop on 'q' key press
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the video capture and close windows
cap.release()
cv2.destroyAllWindows()
```

[ ]: