

RPPG CONTACTLESS

Code:-

```
import cv2
import numpy as np
import pandas as pd

def detect_face(frame):
    face_cascade = cv2.CascadeClassifier('Contactless rPPG
Python/haarcascade_frontalface_default.xml')
    faces = face_cascade.detectMultiScale(frame, 1.3, 5)
    return faces

def calculate_hr(frame, face): # Pass 'frame' as an argument here
    (x, y, w, h) = face
    roi = frame[y:y+h, x:x+w]
    gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    _, thresh = cv2.threshold(gray, 200, 255, cv2.THRESH_BINARY)
    contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)

    hr = 60 * len(contours) / 60
    return hr

def calculate_rr(frame, face): # Pass 'frame' as an argument here
    (x, y, w, h) = face
    roi = frame[y:y+h, x:x+w]
    gray = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    rolling_sum = np.cumsum(gray)
    rr = len(rolling_sum[rolling_sum > 0]) / 60
    return rr

def main():
    cap = cv2.VideoCapture(0)

    while True:
        ret, frame = cap.read()
        faces = detect_face(frame)

        for face in faces:
            hr = calculate_hr(frame, face) # Pass 'frame' to the functions
```

RPPG CONTACTLESS

```

rr = calculate_rr(frame, face) # Pass 'frame' to the functions

(x, y, w, h) = face
cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2) # Draw
a blue rectangle around the face

# Display heart rate and respiratory rate on the frame
cv2.putText(frame, f"Heart rate: {hr:.2f}", (x, y - 20),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)
cv2.putText(frame, f"Respiratory rate: {rr:.2f}", (x, y - 50),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (255, 0, 0), 2)

# Display a constant number on the frame
constant_number = 42 # Change this to the desired constant number
cv2.putText(frame, f"Constant: {constant_number}", (10, 30),
cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)

df = pd.DataFrame({"Heart rate": [hr], "Respiratory rate": [rr]})
df.to_csv("data.csv")

cv2.imshow('frame', frame)

key = cv2.waitKey(1)
if key == ord('Q') or key == ord('q'): # Quit if 'Q' or 'q' is pressed
    break

cap.release()
cv2.destroyAllWindows()

if __name__ == "__main__":
    main()

```

Code Explain:-

Let me explain the code in a diagrammatic form step by step:

1. Import Libraries

- Import the necessary libraries: OpenCV (**cv2**), NumPy (**np**), and Pandas (**pd**).

2. Define Face Detection Function (**detect_face**)

- Load the Haar Cascade classifier for face detection from the XML file.
- The **detect_face** function takes an input frame and detects faces in the frame using the Haar Cascade classifier.
- It returns a list of tuples, where each tuple contains the coordinates (x, y, w, h) of the detected face bounding box.

3. Define Heart Rate Calculation Function (**calculate_hr**)

- The **calculate_hr** function takes the input frame and the coordinates of a detected face.
- It extracts the region of interest (ROI) corresponding to the detected face from the frame.

RPPG CONTACTLESS

- Convert the ROI to grayscale and apply a threshold to segment the heart rate signal.
- Find contours in the thresholded image and calculate the heart rate (beats per minute) based on the number of detected contours.

4. Define Respiratory Rate Calculation Function (`calculate_rr`)

- The **`calculate_rr`** function takes the input frame and the coordinates of a detected face.
- It extracts the region of interest (ROI) corresponding to the detected face from the frame.
- Convert the ROI to grayscale and calculate the respiratory rate (breaths per minute) based on the cumulative sum of the grayscale values.

5. Main Function (`main`)

- Initialize the webcam (capturing device) using OpenCV (**`cv2.VideoCapture(0)`**).

RPPG CONTACTLESS

- Start an infinite loop to continuously read frames from the webcam.
- For each frame, detect faces using the **detect_face** function.
- For each detected face, calculate the heart rate and respiratory rate using the **calculate_hr** and **calculate_rr** functions, respectively.
- Draw a blue rectangle around the detected face and display the heart rate and respiratory rate on the frame using OpenCV functions.
- Display a constant number on the frame (here, the number 42 is used).
- Create a Pandas DataFrame containing the heart rate and respiratory rate data and save it to a CSV file named "data.csv".
- Display the processed frame with rectangles and text using **cv2.imshow**.
- Check for user input; if 'Q' or 'q' is pressed, exit the loop and terminate the program.

RPPG CONTACTLESS

- Release the webcam and close all OpenCV windows when the loop is terminated.
- 6. Call Main Function (`__name__ == "__main__"`)**
- The **main** function is called only if the script is executed directly (not when imported as a module).

Below is a diagrammatic representation of the code flow:

RPPG CONTACTLESS

```
+-----+
|                                     |
|      Import Libraries              |
|                                     |
+-----+
|                                     |
|      V                             |
+-----+
|                                     |
|      Define Face Detection Function |
|                                     |
+-----+
|                                     |
|      V                             |
+-----+
|                                     |
|      Define Heart Rate Calculation Func |
|                                     |
+-----+
|                                     |
|      V                             |
+-----+
|                                     |
|      Define Respiratory Rate Calculation |
|                                     |
+-----+
|                                     |
|      V                             |
+-----+
|                                     |
|      Main Function                 |
|                                     |
+-----+
|                                     |
|      V                             |
+-----+
```

RPPG CONTACTLESS

