Project Title:

PPG Signal Processing for Heart Rate and Abnormality Detection

Objectives

1. **Filter and Enhance PPG Signal** – Apply a bandpass filter to remove noise and enhance useful frequency components.
2. **Detect Peaks and Valleys** – Identify key features (peaks and valleys) from the filtered signal.
3. **Estimate Heart Rate (HR)** – Compute the heart rate from inter-beat intervals (IBI).
4. **Analyze Signal Variance** – Evaluate the variability of the PPG signal for insights into cardiovascular health.
5. **Detect Abnormalities** – Identify unusually high peaks and long inter-beat intervals that may indicate potential health issues like arrhythmia or bradycardia.
6. **Visualize Data** – Display the filtered signal, detected peaks, valleys, and abnormalities using plots for better interpretation.

Procedure

Step 1: Data Collection

- Generate or acquire a raw PPG signal (real-world data from a sensor or a simulated signal).

Step 2: Preprocessing the Signal

- Apply a **bandpass filter (0.5–5 Hz)** to remove unwanted noise and extract relevant frequency components.
- Visualize the **filtered signal** to check the quality.

Step 3: Peak and Valley Detection

- Use **find_peaks()** to detect heartbeats (peaks) and valleys in the PPG waveform.

Step 4: Compute Inter-Beat Intervals (IBI)

- Calculate **time differences** between consecutive peaks to determine IBIs.

Step 5: Estimate Heart Rate

- Convert IBIs into **beats per minute (BPM)** using the formula: *HR=60IBIHR = \frac{60}{IBI}*

Step 6: Analyze Signal Variance

- Compute the **variance** of the signal to understand its stability and fluctuations.

Step 7: Detect Abnormalities

- Identify **abnormally high peaks** (possible noise or signal artifacts).
- Detect **long IBIs (>1.5 sec)** that might indicate bradycardia or arrhythmia.

Step 8: Visualization and Interpretation

- Plot the **filtered PPG signal** along with detected peaks, valleys, and abnormalities.
- Display extracted features such as **heart rate, signal variance, and abnormal intervals**.

Advantages

1. **Non-Invasive Monitoring** – Uses optical sensors, making it safer and more comfortable.

2. **Real-Time Processing** – Can be implemented for live heart rate tracking.

3. **Early Detection of Abnormalities** – Helps identify potential heart conditions early.

4. **Low-Cost Implementation** – Works with affordable hardware (like smartwatches, pulse oximeters).

5. **Improves Health Monitoring** – Useful in fitness tracking and medical diagnostics.

Disadvantages

1. **Sensitive to Motion Artifacts** – Movements can introduce noise and reduce accuracy.

2. **Limited Accuracy in Low Perfusion Conditions** – Poor blood flow affects signal quality.

3. **External Light Interference** – Ambient light can distort readings in optical sensors.

4. **Not a Replacement for ECG** – PPG is less precise for diagnosing severe cardiac conditions compared to electrocardiography (ECG).

from scipy.signal import butter, filtfilt, find_peaks import numpy as np import matplotlib.pyplot as plt

Bandpass filter function

def bandpass_filter(signal, lowcut, highcut, f, order=4): nyquist = 0.5 * f low = lowcut / nyquist high = highcut / nyquist b, a = butter(order, [low, high], btype='band') return filtfilt(b, a, signal)

Generate a synthetic PPG signal (replace with actual data if available)

np.random.seed(0) f = 100 # Sampling frequency (Hz) t = np.linspace(0, 10, f * 10) # 10 seconds of data ppg_signal = 0.6 * np.sin(2 * np.pi * 1.2 * t) + 0.3 * np.sin(2 * np.pi * 0.3 * t) + np.random.normal(0, 0.05, len(t))

Filter the PPG signal

filtered_ppg = bandpass_filter(ppg_signal, 0.5, 5, f)

Plot the filtered signal

plt.figure(figsize=(12, 6)) plt.plot(t, filtered_ppg) plt.title("Filtered PPG Signal") plt.xlabel("Time (s)") plt.ylabel("Amplitude") plt.grid() plt.show()

Step 1: Detect peaks and valleys

peaks, _ = find_peaks(filtered_ppg, height=0.2, distance=f // 2) # Peaks valleys, _ = find_peaks(-filtered_ppg, height=0.2, distance=f // 2) # Valleys

Step 2: Calculate Inter-Beat Intervals (IBI)

ibi = np.diff(peaks) / f # Time between consecutive peaks (in seconds)

Step 3: Estimate Heart Rate

heart_rate = 60 / ibi # Convert IBI to beats per minute (BPM)

Step 4: Compute Signal Variance

signal_variance = np.var(filtered_ppg)

Fault pattern and abnormality detection

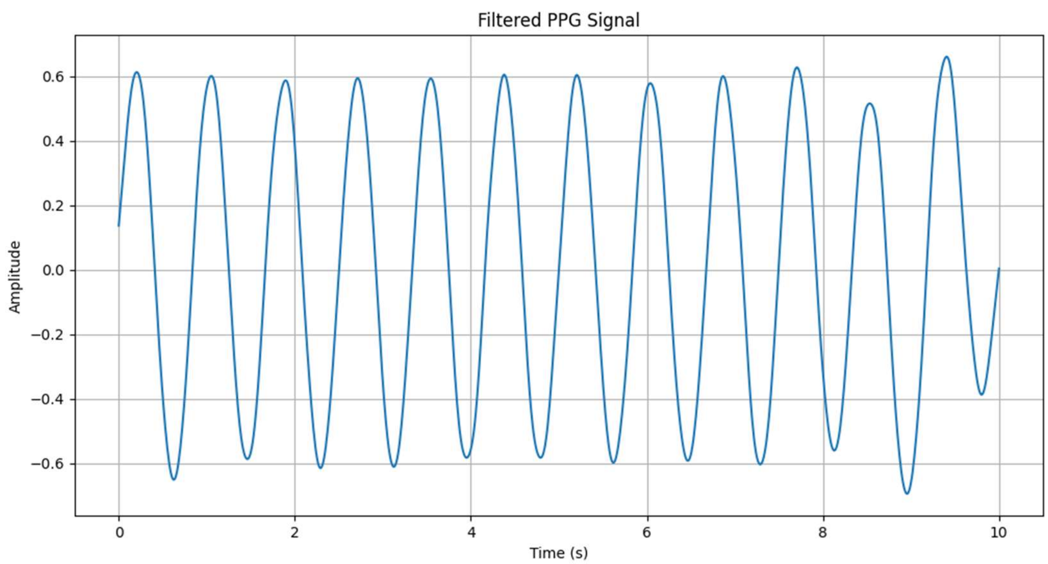
abnormal_peaks = peaks[filtered_ppg[peaks] > 0.8] # Peaks with abnormally high amplitude abnormal_intervals = ibi[ibi > 1.5] # Intervals longer than 1.5 seconds (bradycardia-like behavior)

Plot the filtered signal with detected peaks, valleys, and abnormalities

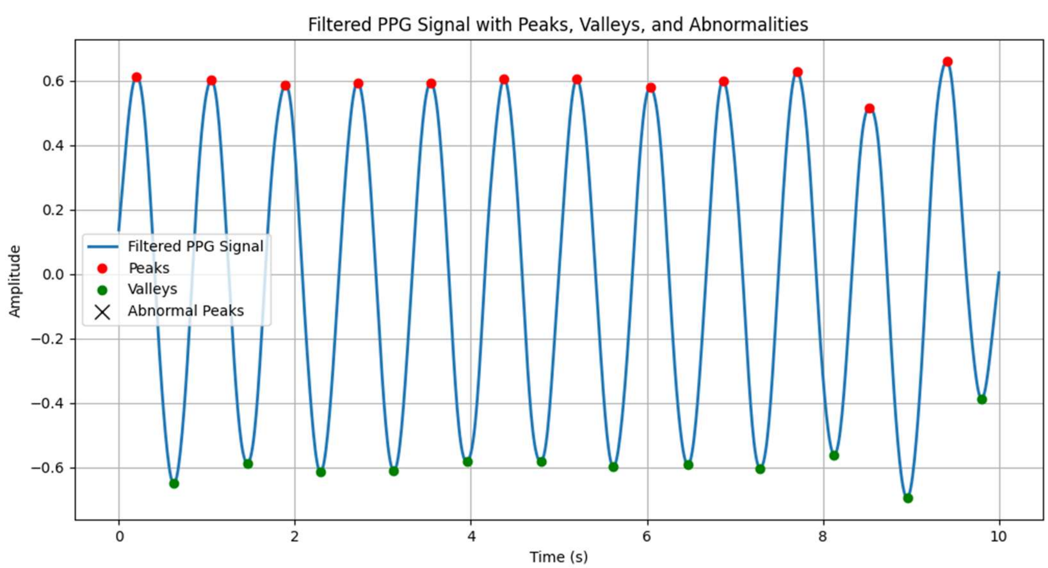
plt.figure(figsize=(12, 6)) plt.plot(t, filtered_ppg, label="Filtered PPG Signal", linewidth=2) plt.plot(t[peaks], filtered_ppg[peaks], "ro", label="Peaks") plt.plot(t[valleys], filtered_ppg[valleys], "go", label="Valleys") plt.plot(t[abnormal_peaks], filtered_ppg[abnormal_peaks], "kx", label="Abnormal Peaks", markersize=10) plt.title("Filtered PPG Signal with Peaks, Valleys, and Abnormalities") plt.xlabel("Time (s)") plt.ylabel("Amplitude") plt.legend() plt.grid() plt.show()

Display extracted features and abnormalities

print("=== Extracted Features ===") print(f"Number of Peaks Detected: {len(peaks)}") print(f"Inter-Beat Intervals (IBI): {ibi} seconds") print(f"Average Heart Rate: {np.mean(heart_rate):.2f} BPM") print(f"Signal Variance: {signal_variance:.4f}") print("\n=== Abnormalities Detected ===") print(f"Number of Abnormal Peaks (High Amplitude): {len(abnormal_peaks)}") print(f"Number of Abnormal Intervals (IBI > 1.5s): {len(abnormal_intervals)}") if len(abnormal_intervals) > 0: print(f"Abnormal Intervals: {abnormal_intervals}") else: print("No abnormal intervals detected.")



Filtered PPG Signal
Amplitude
Time (s)



Filtered PPG Signal with Peaks, Valleys, and Abnormalities
Filtered PPG Signal
Peaks
Valleys
Abnormal Peaks
Amplitude
Time (s)

```
=== Extracted Features ===
Number of Peaks Detected: 12
Inter-Beat Intervals (IBI): [0.85 0.84 0.82 0.83 0.83 0.83 0.83 0.83 0.84 0.82 0.88]
seconds
Average Heart Rate: 71.76 BPM
Signal Variance: 0.1808

=== Abnormalities Detected ===
Number of Abnormal Peaks (High Amplitude): 0
Number of Abnormal Intervals (IBI > 1.5s): 0
No abnormal intervals detected.
```